

Modelos de Pronóstico para la Monitorización de la Calidad de Servicios Cloud

Un Enfoque basado en el Aprendizaje Automático

Autor: Ximena Guerrón Sandoval
Directoras: Silvia Abrahão Gonzales
Marta Fernández Diego
Diciembre 2025

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ITI INVESTIGATE TO INNOVATE



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Departamento de Sistemas Informáticos y Computación
Tesis Doctoral depositada en cumplimiento parcial de los requisitos para el grado de Doctor en Informática

Modelos de Pronóstico para la Monitorización de la Calidad de Servicios Cloud

Un Enfoque basado en el Aprendizaje Automático

Ximena Guerrón Sandoval

Directoras:
Dra. Silvia Abrahão Gonzales
Dra. Marta Fernández Diego

Agradecimientos

El camino de realización de una tesis doctoral resulta arduo, no siendo posible recorrerlo solo. A continuación, se citan a todas las personas que de alguna forma han contribuido en este trabajo.

A mis tutoras de tesis, Silvia Abrahão y Marta Fernández por su confianza, apoyo y enseñanzas a lo largo de mis estudios de doctorado.
A Emilio Insfrán por su apoyo y enseñanzas enriquecedoras.
A mi esposo Geovanny por acompañarme en todo momento, y ser mi soporte.
A mis hijos Emilia y Joaquín por inspirarme a seguir.
A mis padres, por todo su amor incondicional, confianza, consejo y por ser mi inspiración de trabajo y tenacidad para conseguir mis sueños.
A mis hermanos Lenin, Sandra, Andrea y mi familia por todo su amor y constante apoyo.
A mis amigos Fabián, Marco, Carlos, Patricia y Luisa, por su amistad, apoyo, consejo, complicidad y sentido del humor que hicieron gratos e inolvidables todos estos años.
A Carmela, Carito, Silvia, Kamila, Diego
A la Universidad Central del Ecuador y a mi país, por la confianza depositada en mí y por darme esta oportunidad que ha enriquecido mi vida en todo aspecto.
Y por último y no menos recordadas todas las personas que de una u otra manera han contribuido a hacer posible este trabajo.

Resumen

La calidad de servicio (QoS) en sistemas cloud es un factor crítico debido a la naturaleza dinámica, distribuida y heterogénea de estos entornos. Sin embargo, las prácticas actuales de gestión, aunque apoyadas en acuerdos de nivel de servicio (SLA) y herramientas de monitorización, siguen siendo predominantemente reactivas y se basan en la observación aislada de métricas, lo que limita la detección temprana de anomalías y la prevención de violaciones de SLA. A ello se suma la ausencia de un marco unificado que conecte de manera sistemática las características de calidad abstractas, las métricas de QoS y los métodos predictivos. Esta fragmentación conceptual y operativa constituye un desafío científico y práctico que dificulta avanzar hacia una gestión verdaderamente proactiva y fundamentada de la calidad en servicios cloud.

Esta tesis propone un marco integral para el modelado, la evaluación y el pronóstico de la calidad de servicios cloud, articulado en tres contribuciones principales. En primer lugar, se desarrolla una **taxonomía de métricas de QoS** derivada en una revisión sistemática de la literatura, que organiza y caracteriza métricas heterogéneas desde la perspectiva de SaaS, PaaS e IaaS, proporcionando una base estructurada para su interpretación. En segundo lugar, se define un **modelo de calidad genérico para servicios cloud**, alineado con ISO/IEC 25010 y complementado con un proceso de evaluación conforme a ISO/IEC 25040. El modelo descompone las características en 68 subcaracterísticas y 242 atributos específicos del dominio cloud, y se valida mediante un estudio de caso industrial aplicado a un servicio cloud real, mostrando cómo puede instanciarse, configurarse y adaptarse de forma iterativa para generar información relevante, identificar incumplimientos de objetivos de calidad y priorizar acciones de mejora. En tercer lugar, se propone un **proceso de modelado para pronosticar la calidad** de servicios cloud que guía la construcción de modelos de aprendizaje automático. Este proceso se evalúa mediante un experimento controlado que compara enfoques estadísticos (ARIMA) y de aprendizaje profundo (LSTM y BI-GRU), demostrando la superioridad de los modelos bidireccionales en términos de precisión, estabilidad y capacidad de generalización.

En conjunto, los resultados muestran que el marco propuesto proporciona una visión integrada de la calidad en servicios cloud, al conectar la definición formal del dominio, su evaluación sistemática y el pronóstico basado en datos. Esta integración no solo mejora de manera significativa la capacidad para diagnosticar, interpretar y anticipar la QoS, sino que también establece una base metodológica sólida para el diseño de estrategias de gestión proactiva en entornos cloud. Además, la tesis contribuye a avanzar el estado del arte

al ofrecer un enfoque unificador que abre nuevas oportunidades de investigación orientadas al uso de arquitecturas avanzadas para series temporales, mecanismos de aprendizaje continuo y sistemas auto-adaptativos impulsados por métricas de calidad.

Palabras clave: Servicios cloud, Calidad de servicio, QoS, Modelos de calidad, Evaluación de la calidad, Aprendizaje profundo, Pronóstico de series temporales.

Resum

La qualitat de servici (QoS) en sistemes cloud és un factor crític a causa de la naturalesa dinàmica, distribuïda i heterogènia d'estos entorns. No obstant això, les pràctiques actuals de gestió, encara que recolzades en acords de nivell de servici (SLA) i ferramentes de monitoratge, continuen sent predominantment reactives i es basen en l'observació aïllada de mètriques, la qual cosa limita la detecció precoç d'anomalies i la prevenció de violacions de SLA. A això se li suma l'absència d'un marc unificat que connecte de manera sistemàtica les característiques de qualitat abstractes, les mètriques de QoS i els mètodes predictius. Esta fragmentació conceptual i operativa constituïx un desafiament científic i pràctic que dificulta avançar cap a una gestió verdaderament proactiva i fonamentada de la qualitat en servicis cloud.

Esta tesi proposa un marc integral per al modelatge, l'avaluació i el pronòstic de la qualitat de servicis cloud, articulat en tres contribucions principals. En primer lloc, es desenrotlla una **taxonomia de mètriques de QoS** derivada en una revisió sistemàtica de la literatura, que organitza i caracteritza mètriques heterogènies des de la perspectiva de SaaS, PaaS i IaaS, proporcionant una base estructurada per a la seua interpretació. En segon lloc, es definix un **model de qualitat genèric per a servicis cloud**, alineat amb ISO/IEC 25010 i complementat amb un procés d'avaluació conforme a ISO/IEC 25040. El model descompon les característiques en 68 subcaracterístiques i 242 atributs específics del domini cloud, i es valguda mitjançant un estudi de cas industrial aplicat a un servici cloud real, mostrant com pot crear-se, configurar-se i adaptar-se de manera iterativa per a generar informació rellevant, identificar incompliments d'objectius de qualitat i prioritzar accions de millora. En tercer lloc, es proposa un **procés de modelatge per a pronosticar la qualitat de servicis cloud** que guia la construcció de models d'aprenentatge automàtic. Este procés s'avalua mitjançant un experiment controlat que compara enfocaments estadístics (ARIMA) i d'aprenentatge profund (LSTM i BI-GRU), demostrant la superioritat dels models bidireccionals en termes de precisió, estabilitat i capacitat de generalització.

En conjunt, els resultats mostren que el marc proposat proporciona una visió integrada de la qualitat en servicis cloud, en connectar la definició formal del domini, la seua avaluació sistemàtica i el pronòstic basat en dades. Esta integració no sols millora de manera significativa la capacitat per a diagnosticar, interpretar i anticipar la QoS, sinó que també establix una base metodològica sòlida per al disseny d'estratègies de gestió proactiva en entorns cloud. A més, la tesi contribuïx a avançar l'estat de l'art en ofe-

rir un enfocament unificador que obri noves oportunitats d'investigació orientades a l'ús d'arquitectures avançades per a sèries temporals, mecanismes d'aprenentatge continu i sistemes autoadaptatius impulsats per mètriques de qualitat.

Paraules Clau: Servicis cloud, Qualitat de servici, QoS, Models de qualitat, Avaluació de la qualitat, Aprenentatge profund, Pronòstic de sèries temporals

Abstract

The quality of service (QoS) in cloud systems is critical due to their dynamic, distributed, and heterogeneous nature. However, current management practices, despite being supported by service level agreements (SLAs) and monitoring tools, remain predominantly reactive. These practices rely on the isolated observation of metrics, which limits the early detection of anomalies and the prevention of SLA violations. Moreover, there is a lack of a unified framework that systematically connects abstract quality characteristics, QoS metrics, and predictive methods. This conceptual and operational fragmentation poses a scientific and practical challenge, hindering progress towards proactive, informed quality management in cloud services.

This thesis presents a comprehensive framework for modeling, assessing, and predicting cloud service quality, comprising three key contributions. First, a **taxonomy of QoS metrics** is developed based on a systematic review of the literature. This taxonomy organizes and characterizes heterogeneous metrics from the perspectives of Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). It provides a structured basis for interpreting these metrics. Second, a **generic quality model for cloud services** is defined and aligned with ISO/IEC 25010. This model is complemented by an evaluation process that aligns with ISO/IEC 25040. The model categorizes characteristics into 68 sub-characteristics and 242 cloud-specific attributes, and is validated through an industrial case study applied to a real cloud service. This study demonstrates how the model can be instantiated, configured, and iteratively adapted to generate relevant information, identify quality objective breaches, and prioritize improvement actions. Thirdly, a **modeling process is proposed to forecast cloud service quality**, which in turn guides the construction of machine learning models. The efficacy of this process is evaluated through a controlled experiment comparing statistical (ARIMA) and deep learning (LSTM and BI-GRU) approaches. The experiment demonstrates the superiority of bidirectional models in terms of accuracy, stability, and generalization ability.

The collective findings suggest that the proposed framework provides an integrated perspective on the quality of cloud services by combining the formal definition of the domain, systematic evaluation, and data-driven forecasting. This integration has been demonstrated to enhance the ability to diagnose, interpret, and forecast the quality of service (QoS). Additionally, it establishes a robust methodological foundation for developing proactive management strategies within cloud environments. Moreover, the thesis contributes to the advancement of the state of the art by offering a unifying approach

that opens up new research opportunities focused on the use of advanced architectures for time series, continuous learning mechanisms, and self-adaptive systems driven by quality metrics.

Keywords: Cloud services, Quality of service, QoS, Quality models, Quality assessment, Deep learning, Time series forecasting.

Tabla de contenidos

Agradecimientos I

Resumen III

Resum V

Abstract VII

Tabla de contenidos IX

Tabla de figuras XV

Tabla de tablas XVII

1. Introducción 1
- 1.1. Calidad de servicios cloud . . . 2
- 1.2. Aprendizaje automático para el pronóstico de calidad . . . 3
- 1.3. Planteamiento del problema . . . 5
- 1.4. Objetivos e hipótesis . . . 6
- 1.5. Contexto de la investigación . . . 7
- 1.6. Contribuciones principales . . . 9
- 1.7. Metodología de la investigación . . . 10
  - 1.7.1. Análisis del estado del arte y definición de solución candidata . . . 12
  - 1.7.2. Validación estática mediante estudio de caso . . . 14
  - 1.7.3. Validación dinámica mediante experimento controlado . . . 16
- 1.8. Estructura de la tesis . . . 19

2. Fundamentos 23
- 2.1. Servicios cloud . . . 24
  - 2.1.1. Stakeholders . . . 25
  - 2.1.2. Modelos de servicio . . . 26
  - 2.1.3. Arquitectura . . . 27
- 2.2. Modelos de calidad . . . 31
  - 2.2.1. Concepto de calidad . . . 32

2.2.2. Estándares de calidad del software . . . 32
2.3. Modelos de pronóstico de series temporales . . . 36
2.3.1. Pronóstico vs. predicción en métricas de QoS . . . 37
2.3.2. Tipos de series temporales . . . 37
2.3.3. Modelos clásicos de pronóstico . . . 39
2.3.4. Modelos de pronóstico de aprendizaje automático . . . 40
2.3.5. Modelos de pronóstico de aprendizaje profundo . . . 41
2.3.6. Modelos utilizados en esta tesis . . . 44
2.3.7. Justificación de la no inclusión de alternativas avanzadas . . . 45
2.4. Relación entre servicios cloud, modelos de calidad y pronóstico . . . 46
2.5. Conclusiones . . . 47

3. Estado del Arte **49**
3.1. Métricas de calidad y taxonomías . . . 50
3.1.1. Estudios secundarios existentes . . . 51
3.2. Modelos de calidad . . . 53
3.2.1. Modelos de calidad para servicios cloud . . . 53
3.2.2. Modelos en tiempo de ejecución . . . 59
3.3. Modelos de pronóstico para QoS en servicios cloud . . . 60
3.4. Conclusiones . . . 64

4. Taxonomía de Métricas de Calidad para Servicios Cloud **67**
4.1. Introducción . . . 68
4.2. Diseño del protocolo de revisión sistemática . . . 69
4.2.1. Planificación de la SLR . . . 71
4.2.2. Conducción de la SLR . . . 75
4.2.3. Resultados de la SLR . . . 85
4.2.4. Evaluación del protocolo . . . 86
4.3. Análisis bibliométrico del conjunto de artículos primarios . . . 86
4.4. Construcción de la taxonomía . . . 90
4.4.1. Metamodelo para la taxonomía . . . 91
4.4.2. Extracción de datos . . . 94
4.5. Discusión de resultados . . . 116
4.6. Amenazas a la validez . . . 119
4.6.1. Amenazas a la identificación de estudios . . . 119
4.6.2. Amenazas a la selección de estudios y la extracción de datos . . . 120
4.6.3. Amenazas a la síntesis de datos y los resultados . . . 121
4.7. Implicaciones . . . 121
4.7.1. Implicaciones para investigadores . . . 121
4.7.2. Implicaciones para profesionales de la industria . . . 123
4.8. Conclusiones . . . 124

5. Modelo de Calidad para Servicios Cloud **127**
5.1. Introducción . . . 128
5.2. Metodología . . . 129
5.3. Enfoque mixto aplicado al modelo de calidad . . . 131
5.3.1. Consolidación mediante análisis temático . . . 133
5.4. Definición del modelo de calidad para servicios cloud . . . 135
5.4.1. Determinación de las características y subcaracterísticas . . . 138
5.4.2. Descomposición en atributos de calidad . . . 144
5.4.3. Determinación de métricas . . . 153
5.4.4. Modelo de calidad en *Power BI* . . . 154
5.5. Proceso de evaluación de calidad . . . 156
5.5.1. Establecer los requisitos de calidad . . . 157
5.5.2. Configuración de la evaluación . . . 157
5.5.3. Diseño de la evaluación . . . 158
5.5.4. Operacionalización de la evaluación . . . 159
5.5.5. Discusión . . . 160
5.6. Conclusiones . . . 160

6. Aplicación del Modelo de Calidad en un Entorno industrial **163**
6.1. Introducción . . . 164
6.2. Diseño del estudio de caso . . . 164
6.2.1. Selección del caso . . . 165
6.2.2. Justificación del caso seleccionado . . . 166
6.2.3. Preparación para la recolección de datos . . . 167
6.2.4. Establecimiento de requisitos de calidad . . . 167
6.2.5. Configuración de la evaluación . . . 170
6.2.6. Diseño de la evaluación . . . 175
6.3. Recolección de datos . . . 178
6.4. Análisis de datos . . . 180
6.4.1. Datos cuantitativos . . . 180
6.4.2. Datos cualitativos . . . 186
6.5. Discusión de los resultados . . . 187
6.5.1. Impacto del modelo en la evaluación y gestión de la calidad . . . 187
6.5.2. Acciones de mejora derivadas . . . 189
6.6. Respuesta a las preguntas de investigación . . . 190
6.7. Amenazas a la validez . . . 192
6.7.1. Validez de constructo . . . 192
6.7.2. Validez interna . . . 193
6.7.3. Validez externa . . . 194
6.7.4. Validez de las conclusiones . . . 194
6.8. Implicaciones . . . 194
6.8.1. Implicaciones para investigadores . . . 195
6.8.2. Implicaciones para profesionales de la industria . . . 196

6.9. Conclusiones 197

7. Proceso de Modelado ML para el Pronóstico de QoS **199**
7.1. Introducción 200
7.2. Descripción del proceso de modelado de ML 201
7.2.1. Requisitos del modelo 202
7.2.2. Preparación de datos 203
7.2.3. Construcción del modelo 205
7.2.4. Uso del modelo 209
7.3. Protocolo de validación empírica del proceso de modelado 210
7.3.1. Diseño del procedimiento de validación 210
7.3.2. Entrevista semiestructurada 211
7.3.3. Lista de comprobación 212
7.3.4. Procedimiento de validación 213
7.3.5. Resultados esperados 216
7.4. Conclusiones 216

8. Modelos de Pronóstico de QoS para Servicios Cloud **219**
8.1. Introducción 220
8.2. Experimento controlado 221
8.2.1. Definición 221
8.2.2. Planificación 223
8.2.3. Operación 229
8.3. Resultados 234
8.3.1. Estadística descriptiva y análisis exploratorio de datos 234
8.3.2. Prueba de hipótesis y discusión 241
8.3.3. Retroalimentación preliminar 247
8.4. Amenazas a la validez 250
8.4.1. Validez del constructo 250
8.4.2. Validez interna 251
8.4.3. Validez externa 251
8.4.4. Validez de las conclusiones 252
8.5. Implicaciones 252
8.5.1. Implicaciones para investigadores 252
8.5.2. Implicaciones para profesionales de la industria 253
8.6. Conclusiones 255

9. Conclusiones y Trabajos Futuros **257**
9.1. Contribuciones principales 258
9.1.1. Contribuciones conceptuales 258
9.1.2. Contribuciones metodológicas 259
9.1.3. Contribuciones empíricas 260
9.2. Análisis de la consecución de los objetivos 263

9.3. Síntesis de las contribuciones e implicaciones teóricas . . . 265
9.4. Limitaciones . . . 266
9.4.1. Limitaciones conceptuales . . . 266
9.4.2. Limitaciones metodológicas . . . 266
9.4.3. Limitaciones empíricas . . . 267
9.5. Trabajos futuros . . . 267
9.5.1. Extensión del marco conceptual de calidad . . . 268
9.5.2. Evolución de capacidades predictivas . . . 268
9.5.3. Integración y transferencia . . . 269
9.6. Difusión de resultados . . . 270
9.6.1. Revistas internacionales con índice de calidad relativo . . . 270
9.6.2. Actas de congresos nacionales . . . 270
9.6.3. Capítulos de libro . . . 270
9.6.4. Trabajos en curso . . . 271

Bibliografía **273**

Anexos **288**

A. Anexo 1. Estudios primarios de la Taxonomía de métricas **289**

Tabla de figuras

1.1. Modelo de transferencia tecnológica. 10
1.2. Proceso de la Revisión Sistemática de Literatura. 13
1.3. Proceso experimental con los artefactos generados en las distintas actividades. 18
1.4. Estructura de la tesis. 22

2.1. Responsabilidades del proveedor y consumidor por modelos de servicios. . . 28
2.2. Modelo conceptual de la arquitectura de servicios cloud. 29
2.3. Modelo de calidad del producto de software ISO/IEC 25010 [1]. 34
2.4. Modelo de calidad del producto de software ISO/IEC 25010 [1]. 35
2.5. Modelo de calidad en uso ISO/IEC 25010 [1]. 35
2.6. Estructura LSTM 43
2.7. Estructura BI-GRU. 43

4.1. Fases de la revisión sistemática de la literatura [2]. 70
4.2. Resultado del proceso de selección de estudios primarios [2]. 89
4.3. Número de estudios primarios por año y fuente [2]. 90
4.4. Metamodelo de calidad para servicios cloud [2]. 92
4.5. Taxonomía refinada de métricas para servicios cloud [2]. 95
4.6. Distribución de los estudios primarios según las características de calidad [2]. 96
4.7. Resultados obtenidos al combinar las características de QoS y las fases del ciclo de vida del servicio cloud con el punto de vista de los *stakeholders* y el tipo de servicio. 116
4.8. Resultados obtenidos al combinar las características de QoS con el tipo de métrica, el soporte de la herramienta, el resultado de la medición y el artefacto cloud medido. 118
4.9. Taxonomía de métricas de QoS para servicios cloud[2]. 125

5.1. Proceso de descomposición e integración del modelo de calidad. 134
5.2. Modelo de calidad del servicio cloud: características y subcaracterísticas. . . 137
5.3. Modelo de Calidad en *Power BI*. 156
5.4. Proceso de Evaluación de Calidad. 157
5.5. Configuración de Evaluación de Calidad. 158

6.1. Estudio de caso. . . . 165
6.2. Arquitectura del servicio cloud SAlert. . . . 166
6.3. Componentes de arquitectura de SAlert. . . . 166
6.4. Modelo de calidad para la evaluación del servicio SAlert. . . . 176
6.5. Resultados de medición SaaS. . . . 182
6.6. Resultados de medición PaaS. . . . 184
6.7. Resultados de medición IaaS. . . . 185

7.1. Proceso de modelado de ML, adaptado de Amershi et al.[3] 202
7.2. Etiquetado de series temporales para predicciones univariantes y multivariantes. . . . 205
7.3. Variables de los modelos DL [4]. . . . 208

8.1. Arquitectura DL para modelos de pronóstico de QoS: el modelo BI-GRU (arriba) y el modelo LSTM (abajo). . . . 231
8.2. Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS1–4) — parte 1. . . . 235
8.2. Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS5–8) — parte 2. . . . 236
8.2. Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS9–12) — parte 3. . . . 237
8.2. Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS13–16) — parte 4. . . . 238
8.3. Convergencia del modelo BI-GRU durante el entrenamiento para la métrica *Memoria utilizada %* en la iteración final. . . . 240
8.4. Pronóstico de series temporales con BI-GRU para la memoria utilizada % (*memory used %*). . . . 240
8.5. PCoA con elipses de confianza del 95 % y significación por pares. (1–8)—parte 1. . . . 248
8.5. PCoA con elipses de confianza del 95 % y significación por pares. (9–16)—parte 2. . . . 249

Tabla de tablas

1.1. Plantilla *Goal-Question-Metric* para la definición de experimentos. 17

3.1. Resumen comparativo de los estudios secundarios sobre métricas de calidad y taxonomías. 54
3.2. Resumen comparativo de los estudios de modelos de calidad para servicios cloud. 58
3.3. Resumen comparativo de los estudios de pronóstico de QoS en servicios cloud. 61

4.1. Palabras clave y términos relacionados para la cadena de búsqueda. 74
4.2. Ítems de calidad para evaluar los estudios primarios. 76
4.3. Impacto potencial e influencia de los estudios primarios. 77
4.4. Criterios de extracción de datos de los estudios primarios. 78
4.5. Lista de revistas y conferencias relevantes. 87
4.6. Lista de verificación para la evaluación de la calidad. 88
4.7. Relevancia de los estudios primarios. 89
4.8. Contribución de los estudios primarios a las características de calidad. . . . 96
4.9. Resultados de los criterios de extracción de datos. 97

5.1. Subcaracterísticas de la eficiencia de desempeño. 139
5.2. Subcaracterísticas de fiabilidad. 140
5.3. Subcaracterísticas de seguridad. 140
5.4. Subcaracterísticas de Mantenibilidad. 141
5.5. Subcaracterísticas de portabilidd. 142
5.6. Subcaracterísticas de adecuación funcional. 142
5.7. Subcaracterísticas de usabilidad. 143
5.8. Subcaracterísticas de compatibilidad. 143
5.9. Subcaracterísticas no técnicas. 144
5.10. Atributos de eficiencia de desempeño. 145
5.11. Atributos de fiabilidad. 148
5.12. Atributos de seguridad. 149
5.13. Atributos de mantenibilidad. 150
5.14. Atributos de portabilidad. 150
5.15. Atributos de adecuación funcional. 151

5.16. Atributos de usabilidad. 152
5.17. Atributos de compatibilidad. 152
5.18. Atributos no técnicos. 153

6.2. Atributos de calidad seleccionados del modelo de calidad de SAlert. 170
6.3. Atributos, métricas y operacionalizaciones del Modelo de Calidad para SAlert. 172
6.4. Requisitos, atributos, métricas y operacionalizaciones modificados para SAlert: $\oplus$ añadidos o $\ominus$ eliminados 175
6.5. Mapeo operacional de métricas QoS con criterios de decisión 179
6.6. Resultados de la monitorización de SAlert. 180
6.7. Resultados de monitorización de PaaS (Contenedores). 183

7.1. Lista de comprobación para la evaluación de modelos de pronóstico de QoS 212

8.1. Estadísticas descriptivas del rendimiento de los modelos de pronóstico. . . . 242
8.2. Análisis estadístico inferencial del rendimiento de los modelos de pronóstico. 245

Capítulo 1

Introducción

En este capítulo se contextualiza el trabajo de investigación, se presenta el problema que motiva la tesis, se formulan los objetivos y las contribuciones, se describe el contexto institucional en el que se desarrolla y se detalla la metodología de investigación utilizada. Finalmente, se expone la estructura global del documento.

La Sección 1.1 introduce la importancia de los modelos de calidad y métodos de evaluación en el dominio de los servicios cloud. La Sección 1.2 discute la relevancia del aprendizaje automático para abordar enfoques de pronóstico de la calidad de servicios. La Sección 1.3 presenta el planteamiento del problema que motiva el modelo de calidad y el pronóstico de calidad en los servicios cloud. La Sección 1.4 presenta la estructura de los objetivos de investigación. La Sección 1.5 presenta las contribuciones de la tesis. La Sección 1.7 presenta un resumen de las metodologías de investigación aplicadas en este estudio. La Sección 1.8 presenta la estructura de la tesis junto con un resumen de cada capítulo.

1.1. Calidad de servicios cloud

La creciente necesidad de las organizaciones de agilizar la entrega de sus productos y servicios ha encontrado en la computación en la nube (*cloud computing*) una solución efectiva para responder a sus demandas, como el aprovisionamiento flexible de infraestructura, recursos bajo demanda y servicios personalizados. Los servicios cloud poseen características esenciales que los distinguen de otros modelos tecnológicos, ofreciendo ventajas significativas tanto para proveedores como para consumidores. Entre estas se destacan la elasticidad y la escalabilidad, que permiten aprovisionar los recursos de forma dinámica según la demanda, la alta disponibilidad, el uso eficiente de recursos *hardware*, así como un modelo de facturación flexible basado en el uso del servicio [5].

Sin embargo, la creciente complejidad de estos entornos introduce nuevos desafíos en la gestión de la calidad del servicio (*Quality of Service* - QoS), debido a la naturaleza dinámica del aprovisionamiento de recursos, a la heterogeneidad de los proveedores y la necesidad de garantizar servicios confiables y disponibles en un entorno multiusuario. Por ejemplo, la naturaleza dinámica de los servicios puede provocar variaciones en la calidad en función de la carga del sistema o la ubicación geográfica del usuario. Además, los servicios cloud suelen compartir infraestructura física entre múltiples clientes (multitenencia), lo que introduce variabilidad e interferencias en el rendimiento y la disponibilidad. Otro aspecto relevante es que la calidad no depende únicamente del servicio final, sino también de las capas subyacentes asociadas a la plataforma y la infraestructura cloud.

Existen marcos de calidad generales y algunos específicos para servicios cloud, aunque con limitaciones. La familia de normas ISO/IEC 25000 proporciona modelos para la especificación de requisitos y la evaluación de la calidad de los productos de software. En particular, el estándar ISO/IEC 25010 [1] define modelos genéricos tanto para la calidad del producto como para la calidad en uso. No obstante, estos modelos están enfocados en productos de software genéricos, no en servicios distribuidos ni en multitenencia. Por lo tanto, necesitan ser adaptados y extendidos para capturar las particularidades del dominio cloud. Por otra parte, estándares como la ISO/IEC 20000 [6] proporcionan un marco para la gestión de servicios TI, orientado a los procesos ITIL, que, aunque son útiles para la gestión organizacional, no presentan métricas específicas de elasticidad, escalabilidad o virtualización.

En general, estos estándares genéricos necesitan ser extendidos para proporcionar mecanismos que permitan definir y evaluar las características específicas de los servicios cloud. Dichas adaptaciones deben reflejar no solo la naturaleza dinámica de los servicios, sino también sus múltiples artefactos, los diferentes modelos de despliegue (SaaS, PaaS, IaaS) y las distintas fases de su ciclo de vida del servicio.

La calidad constituye un concepto abstracto que debe descomponerse en atributos de calidad medibles [7]. En los últimos años, varios autores han propuesto modelos de calidad específicos para los servicios cloud [8, 9, 10, 11]. Sin embargo, estas propuestas carecen de una cobertura integral de atributos de calidad con sus respectivas métricas, lo que limita su aplicabilidad a distintos modelos de servicios cloud (SaaS, PaaS, IaaS), artefactos y plataformas cloud. Además, la mayoría no está alineada con estándares ampliamente acep-

tados como la ISO/IEC 25010 [1], dificultando su integración con enfoques estandarizados de aseguramiento de la calidad.

Iniciativas como el *Cloud Service Measurement Index* (SMI), desarrollado por el *Cloud Service Measurement Initiative Consortium* (CSMIC) [12], han contribuido al entendimiento multidimensional de la calidad de servicios cloud, proponiendo seis indicadores clave de rendimiento (*Key Process Indicators* - KPIs): calidad, agilidad, riesgo, capacidad, coste y seguridad. Este marco permite comparar servicios y proveedores cloud a partir de más de 40 características individuales relacionadas con los servicios cloud. Sin embargo, el SMI no proporciona métricas ni metodologías de medición concretas.

Por otra parte, la NIST 500-307 [13] constituye una de las principales referencias normativas desarrolladas por el *National Institute of Standards and Technology* (NIST) para estandarizar la evaluación de la calidad de servicios cloud. Este estándar establece un marco conceptual para la identificación, descripción y categorización de métricas de calidad aplicables a los distintos modelos de servicio (IaaS, PaaS y SaaS) y de despliegue (público, privado, híbrido y comunitario). La guía propone un conjunto de atributos de calidad relevantes (como disponibilidad, rendimiento, escalabilidad, elasticidad, interoperabilidad, portabilidad, seguridad y privacidad) y ofrece descripciones de cada uno, indicando su relación con los acuerdos de nivel de servicio (SLA). No obstante, el estándar no prescribe métodos concretos de medición ni valores de referencia, lo que limita su aplicación práctica y deja abierta la necesidad de desarrollar métricas cuantitativas y procedimientos de evaluación normalizados para la comparación objetiva entre proveedores y servicios cloud.

Una gestión eficaz de la calidad requiere no solo una definición conceptual de la calidad, sino también un método de evaluación que permita cuantificarla. Los modelos existentes suelen carecer de soporte para la evaluación o proponen procedimientos excesivamente abstractos (como la ISO/IEC 25040 [14]), lo que limita su aplicabilidad y puede generar resultados inconsistentes o poco fiables.

Una evaluación eficaz de la calidad del servicio permite identificar deficiencias en los distintos artefactos y en las distintas fases del ciclo de vida del servicio, tanto en el tiempo de diseño como en el de ejecución. Estas acciones resultan fundamentales para monitorizar y garantizar el cumplimiento de los acuerdos de nivel de servicio (*Service Level Agreement* - SLA), los cuales establecen los requisitos de calidad pactados entre proveedores y consumidores, así como las penalizaciones en caso de incumplimiento. Asimismo, los resultados de la medición de la calidad del servicio pueden emplearse para dirigir otras actividades de mejora, como la adaptación del servicio, con el fin de asegurar el cumplimiento de los niveles de calidad esperados.

1.2. Aprendizaje automático para el pronóstico de calidad

La gestión efectiva de los SLAs requiere mecanismos de monitorización robustos y confiables que permitan verificar el cumplimiento del SLA entre proveedores y consumidores. La monitorización continua de los servicios cloud posibilita observar su comportamiento a lo largo del tiempo y proporciona información crítica para apoyar distintas actividades, como la gestión del rendimiento, la planificación de la capacidad, la tolerancia a fallos o

la migración de servicios. En consecuencia, la evaluación y monitorización permanentes se han convertido en una necesidad para comprender el estado actual y pasado de los servicios y verificar su conformidad con la calidad esperada.

No obstante, en entornos cloud, caracterizados por su complejidad, volatilidad y dinamismo, conocer el estado actual o histórico del servicio ya no es suficiente. La posibilidad de utilizar datos históricos como base para anticipar condiciones futuras mediante técnicas de pronóstico representa un avance significativo hacia esquemas de evaluación proactiva. Este enfoque permite a las organizaciones la detección temprana de posibles degradaciones del servicio, anticiparse a violaciones de SLA, aplicar medidas correctivas oportunas y optimizar la planificación y provisión de recursos.

En este contexto, el aprendizaje automático (*Machine Learning* - ML) ofrece un gran potencial para abordar los desafíos del pronóstico de la calidad (QoS) en servicios cloud. Gracias a su capacidad para analizar grandes volúmenes de datos, identificar patrones complejos y modelar relaciones no lineales entre variables, las técnicas de ML permiten anticipar el comportamiento futuro del servicio con mayor precisión que los enfoques tradicionales basados en reglas estáticas con umbrales predeterminados. Además, los modelos de aprendizaje automático pueden ser entrenados y personalizados para ajustarse a las necesidades específicas de cada organización, lo que incrementa su aplicabilidad en la práctica. Su capacidad de ajuste dinámico (*fine-tuning*) frente a nuevos datos o condiciones operativas cambiantes constituye una ventaja clave en entornos donde la variabilidad es constante.

La elección de la técnica de ML más adecuada depende de múltiples factores, como el tamaño del conjunto de datos, la complejidad de las relaciones entre variables, las dependencias temporales y los recursos computacionales disponibles para el entrenamiento. Entre las técnicas más efectivas para abordar tareas de pronóstico en series temporales destacan las redes neuronales recurrentes (*Recurrent Neural Network* - RNN), particularmente las arquitecturas LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*), por su capacidad para procesar secuencias de datos y mantener información contextual a lo largo del tiempo [15].

El aprendizaje profundo (*Deep Learning* - DL) permite capturar y modelar dependencias a largo plazo, así como detectar patrones complejos en datos secuenciales, incluso en presencia de ruido y fluctuaciones de trabajo [16]. Estas redes permiten incorporar múltiples capas de abstracción, lo que facilita la detección de patrones latentes en datos multidimensionales. En el contexto de los servicios cloud, esta capacidad resulta especialmente valiosa, dado que los atributos de calidad están interrelacionados y dependen de múltiples factores externos, como la carga de trabajo, el comportamiento de los usuarios o la asignación de recursos.

A pesar de su potencial, la aplicación de estas técnicas al pronóstico de la calidad de los servicios cloud sigue siendo incipiente. Estudios como los de White et al. [17] y Syu et al. [18] han explorado el uso de modelos LSTM con resultados prometedores en términos de precisión, aunque aún existen brechas en cuanto a la adopción sistemática de estas técnicas, la disponibilidad de conjuntos de datos y la evaluación rigurosa de su capacidad de generalización a diferentes tipos de servicios y métricas de calidad.

En conjunto, el uso de técnicas de ML para el pronóstico de QoS representa un paso hacia una gestión más inteligente, proactiva y eficiente de los servicios cloud, permitiendo no solo describir el estado actual del sistema, sino anticipar comportamientos futuros y favorecer decisiones estratégicas y alineadas con los objetivos del negocio.

1.3. Planteamiento del problema

En la actualidad, los servicios cloud se han consolidado como una solución esencial para responder a las demandas de escalabilidad, elasticidad y eficiencia en la provisión de infraestructura y software. Sin embargo, esta adopción conlleva un desafío crítico: garantizar la calidad de los servicios ofrecidos.

Aunque la ISO/IEC 25010 [1] constituye un referente consolidado para la especificación y evaluación de la calidad de los productos de software, su aplicación al dominio cloud presenta limitaciones significativas debido a su carácter genérico y la falta de consideración de propiedades específicas de estos entornos.

La ISO/IEC 25010 define un modelo de calidad basado en características abstractas que, si bien son ampliamente aplicables, requieren adaptación para abordar adecuadamente los aspectos diferenciales de dominios particulares. En consecuencia, diversos trabajos han propuesto extensiones o especializaciones de este estándar para dominios particulares, como el aprendizaje automático [19], la sostenibilidad del software [20] o la calidad del producto en general [21]. Estas propuestas reconocen la necesidad de operacionalizar las características del estándar mediante métricas y atributos específicos que permitan una evaluación objetiva y contextualizada.

Sin embargo, en el caso particular de los servicios cloud aún no existe un modelo que descomponga su calidad en características y atributos específicos y que proporcione mecanismos para medir estos atributos. Esta carencia impide a los *stakeholders* contar con una aproximación que les permita personalizar, evaluar y mejorar la calidad de sus servicios en escenarios reales.

Otro desafío clave reside en la evaluación dinámica y continua de la calidad. Si bien existen mecanismos de monitorización que permiten observar el comportamiento de los servicios en tiempo real, la gestión efectiva de los SLA requiere también la capacidad de anticipación ante condiciones futuras. Esta dimensión predictiva resulta estratégica en entornos multitenencia y multicloud, donde las condiciones de operación varían rápidamente y la toma de decisiones proactiva puede marcar la diferencia entre cumplir o no un SLA.

Las técnicas de ML ofrecen un enfoque prometedor para anticipar el comportamiento futuro del servicio a partir de datos históricos de monitorización. Modelos como LSTM [16] y BI-GRU [22], especializados en el tratamiento de secuencias temporales, han mostrado buenos resultados en dominios afines (servicios web [23]), aunque su aplicación al pronóstico de QoS en servicios cloud aún es limitada. Asimismo, los modelos estadísticos tradicionales, como ARIMA [24], permiten establecer una línea base de comparación con otros modelos de ML.

En consecuencia, el problema central que aborda esta tesis es la ausencia de un modelo de calidad específico para servicios cloud que permita evaluar la calidad del servicio

tanto en tiempo de diseño como en tiempo de ejecución. Dicho modelo deberá integrar las características de calidad abstractas con los mecanismos de evaluación de calidad específicos (métricas). Además, debe facilitar la evaluación de atributos de calidad desde las fases iniciales del ciclo de vida del servicio, posibilitando una evaluación sistemática de la calidad esperada antes del despliegue.

Asimismo, en la fase de operación, el modelo debería permitir una *evaluación continua*, orientada tanto a la observación reactiva (para identificar variaciones o degradaciones de calidad) como a la *evaluación pronóstica*, que anticipe posibles incumplimientos de los niveles de calidad esperados. Finalmente, el modelo deberá ser flexible e instanciable, de modo que pueda adaptarse a las necesidades específicas de diferentes organizaciones o proveedores de servicios.

1.4. Objetivos e hipótesis

El objetivo principal de esta tesis doctoral es:

> *Definir y evaluar empíricamente una aproximación integral para el modelado, evaluación y pronóstico de la calidad de los servicios cloud, que permita determinar el estado actual del servicio, detectar variaciones o degradaciones y anticipar incumplimientos de los niveles de calidad esperados.*

Para abordar este objetivo, se plantean los siguientes objetivos específicos:

- **O1. Analizar el estado del arte** sobre evaluación de la calidad en servicios cloud y los enfoques existentes para el pronóstico de calidad, identificando limitaciones metodológicas y operacionales y oportunidades de mejora.
- O2. **Elaborar una taxonomía de métricas de QoS** que permita clasificar y estructurar las métricas disponibles en la literatura para su uso en la monitorización de servicios cloud.
- O3. **Proponer un modelo de calidad para los servicios cloud**, alineado con la ISO/IEC 25010 [1], que integre las características de calidad abstractas con las métricas de QoS derivadas de la taxonomía definida previamente. El resultado será un modelo de calidad de servicios cloud genérico que podrá ser instanciado y customizado a distintos servicios y organizaciones.
- O4. **Definir un proceso sistemático de evaluación** basado en el modelo de calidad propuesto.
- O5. **Operacionalizar y evaluar empíricamente el modelo de calidad** y el proceso de evaluación propuestos en un entorno industrial, con el fin de verificar su aplicabilidad y utilidad.

- O6. **Definir un proceso para la construcción de modelos de pronóstico de QoS mediante ML** que guíe desde la definición de requisitos hasta la construcción y el uso del modelo.
- O7. **Construir, validar y comparar modelos de pronóstico de calidad de servicios cloud**, aplicando el proceso propuesto y evaluando su precisión, eficiencia computacional y capacidad de generalización mediante experimentos controlados que utilicen datos de monitorización de servicios cloud en un entorno industrial real.

Estos objetivos se articulan como hipótesis relacionadas con la capacidad del modelo de calidad propuesto para representar de manera fiable las características de los servicios cloud y con la eficacia del proceso de ML para anticipar la QoS.

1.5. Contexto de la investigación

Esta tesis doctoral se ha desarrollado en el contexto del grupo de investigación de Ingeniería del Software y Sistemas de Información (ISSI) del Instituto Universitario Mixto de Tecnología Informática (IUMTI) de la Universitat Politècnica de València (UPV). Los trabajos que han hecho posible el desarrollo de esta tesis se engloban en proyectos de I+D financiados con fondos públicos. En particular, esta tesis ha contribuido a los siguientes proyectos:

- `Adapt@Cloud`: *"Adaptación dinámica de servicios cloud centrada en el usuario"* de la convocatoria de ayudas a proyectos de I+D+i, orientada a los retos de la sociedad del año 2018, financiada por el Ministerio de Economía y Competitividad (España).

  Participantes: Universitat Politècnica de València (España), Universidade Nova de Lisboa (Portugal), Université Catholique de Louvain (Bélgica), Software Engineering Institute, Carnegie-Mellon University (EE. UU.).

  IP: Silvia Abrahão y Emilio Insfrán.

  De enero de 2018 a septiembre de 2021.

  Esta tesis contribuye a un objetivo clave del proyecto relacionado con la evaluación de la calidad, ya que proporciona una aproximación que permite detectar desviaciones de los niveles de calidad de los servicios que son usadas para dirigir la adaptación dinámica del servicio.

- `CALESI`: *"Red de excelencia en calidad y sostenibilidad de software"* de la convocatoria de acciones de dinamización "redes de excelencia" del año 2017 (TIN2017-90689-REDT), financiado por el Ministerio de Economía, Industria y Competitividad (España).

  Participantes: Universitat Politècnica de València, Universidad de Castilla-La Mancha, Universidad Politècnica de Catalunya, Universidad Politécnica de Madrid, Universidad de Murcia, Universidad de Málaga, Universidad de Alicante, Universidad de Sevilla, Universidad de Cádiz, Universidad de Mondragón.

IP: Silvia Abrahão.

De enero de 2018 a octubre de 2021.

En el contexto de esta red, se ha colaborado con la profesora Carme Quer de la Universitat Politècnica de Catalunya en la definición de la primera versión del modelo de calidad para servicios cloud planteado en esta tesis.

- `UCI-Adapt`: *"User-Centered Intelligent software system ADAPTion"* de la convocatoria de ayudas a proyectos de I+D+i del Plan Estatal de Investigación Científica y Técnica y de Innovación, Proyectos Generación de Conocimiento (PID2022-140106NB-I00), financiada por la Agencia Estatal de Investigación, Ministerio de Economía y Competitividad (España).

  Participantes: Universitat Politècnica de València (España), Universidade Nova de Lisboa (Portugal), Université Catholique de Louvain (Bélgica).

  IP: Silvia Abrahão y Emilio Insfrán.

  De septiembre de 2023 a agosto de 2027.

  Esta tesis contribuye a una de las tareas de este proyecto, relacionadas con la definición de un modelo de calidad para servicios cloud, así como en los procedimientos para recopilar los datos de monitorización correspondientes.

- `PAID-06-24`: *"Modelos de pronóstico de la calidad de servicios en la nube: Un enfoque integral basado en aprendizaje profundo"*, Convocatoria de ayudas a primeros proyectos de investigación, financiada por la Universitat Politècnica de València.

  IP: Marta Fernández.

  De enero de 2025 a diciembre de 2026.

  Esta tesis contribuye de manera significativa a este proyecto, ya que constituye uno de los resultados principales del mismo.

- `CALIPRO`: *"Proceso para la Definición y Validación de Modelos de Machine Learning para Pronosticar la Calidad de Servicios Cloud "*, de la convocatoria AICO 2025 (CIAICO/2024/323), financiado por la Consellería de Educación, Cultura, Universidades y Empleo (España).

  IP: Silvia Abrahão y Marta Fernández.

  De septiembre de 2025 a agosto de 2028.

  Esta tesis contribuye de manera significativa a un objetivo importante del proyecto, relacionado con el proceso para la construcción y validación empírica de modelos de pronóstico de calidad de servicios.

Además, el trabajo de investigación de Ximena Guerrón fue financiado por la Universidad Central del Ecuador, Ecuador.

1.6. Contribuciones principales

El desarrollo de esta tesis ha dado lugar a un conjunto coherente de contribuciones científicas y técnicas que avanzan el conocimiento en el ámbito de la calidad de servicios cloud y en la aplicación de técnicas de aprendizaje automático para el pronóstico de atributos de QoS. Estas contribuciones se derivan directamente de los objetivos planteados y se materializan tanto en modelos conceptuales como en procesos sistemáticos y validaciones empíricas. A continuación se sintetizan las aportaciones principales.

En primer lugar, se ha construido una **taxonomía estructurada de métricas de calidad para servicios cloud**, obtenida mediante una revisión sistemática de la literatura y un análisis temático profundo. Esta taxonomía representa el estado del conocimiento del dominio, identifica vacíos en la investigación y proporciona una base sólida para el diseño de modelos de calidad específicos.

En segundo lugar, se ha definido un **modelo de calidad para servicios cloud alineado con ISO/IEC 25010**, que incorpora características, subcaracterísticas y atributos específicos del entorno cloud, así como las métricas y operacionalizaciones asociadas. Este modelo constituye una adaptación formal de los estándares de calidad al contexto particular de la computación en la nube, integrando dimensiones no cubiertas por los modelos genéricos tradicionales.

En tercer lugar, la tesis aporta un **proceso de evaluación de la calidad basado en el modelo propuesto**, formulado conforme a la estructura establecida por ISO/IEC 25040. Este proceso describe las actividades, roles, artefactos y criterios necesarios para evaluar de manera sistemática la calidad de servicios cloud, facilitando su aplicabilidad por parte de organizaciones proveedoras y consumidoras de servicios. En cuarto lugar, se ha realizado una **evaluación industrial del modelo y del proceso de evaluación**, mediante su aplicación en un entorno real de servicios cloud perteneciente a una organización colaboradora. Esta validación permite analizar la utilidad, aplicabilidad y beneficios prácticos del modelo, así como identificar desafíos reales asociados a su uso.

En quinto lugar, se ha desarrollado un **proceso sistemático para la construcción de modelos de pronóstico de atributos de QoS mediante aprendizaje automático**, que guía desde la preparación de datos hasta la evaluación y selección de modelos. Este proceso integra buenas prácticas, consideraciones específicas del dominio cloud y metodologías reproducibles para la experimentación.

Finalmente, como contribución técnica central, se han **construido, evaluado y comparado empíricamente diferentes modelos de pronóstico de QoS** —incluyendo enfoques estadísticos y de aprendizaje profundo— utilizando datos reales de monitorización. Esta comparación experimental permite identificar ventajas y limitaciones de cada enfoque, ofrece evidencia empírica de su desempeño y aporta recomendaciones fundamentadas para su utilización en entornos operativos.

En conjunto, estas contribuciones ofrecen una visión integral del problema de la evaluación y pronóstico de la calidad en servicios cloud, articulando modelos conceptuales, procesos formales y validaciones empíricas que proporcionan tanto valor académico como aplicabilidad industrial.

1.7. Metodología de la investigación

Las actividades de investigación que han dado lugar a esta tesis doctoral están estructuradas siguiendo una extensión del modelo de transferencia de tecnología propuesto por Gorschek et al. [25], el cual promueve la colaboración sistemática entre la academia y la industria mediante un proceso iterativo de formulación, evaluación y validación de soluciones a problemas reales.Este modelo se basa en siete actividades relacionadas donde la búsqueda de soluciones adecuadas se realiza de forma iterativa por medio de la formulación de soluciones candidatas y la correspondiente validación empírica que permite dirigir los esfuerzos hacia una solución realista. Las actividades del modelo, presentadas en la Figura 1.1, son las siguientes:

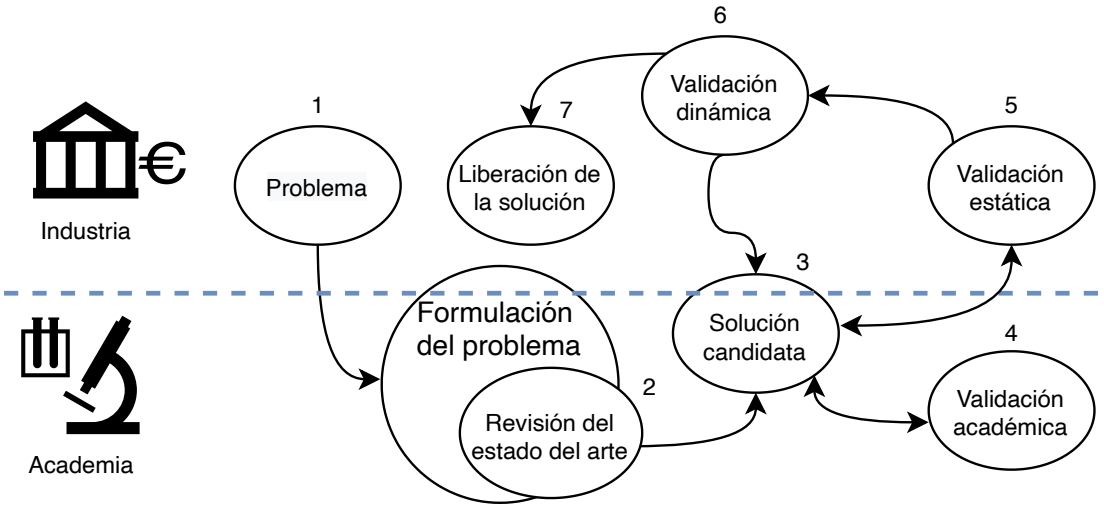


Figura 1.1: Modelo de transferencia tecnológica.

1. **Problema**: Se plantea un problema o desafío que ocurre en la industria. El problema puede ser planteado directamente por una empresa con la que se coopera o ser identificado mediante la observación y el análisis de las prácticas actuales. Existe la necesidad de un marco de trabajo que permita medir la calidad interna y externa de los servicios cloud y adaptarlo a sus requerimientos particulares. Además de abordar mecanismos para el pronóstico de la calidad del servicio que apoyen la toma de decisiones informadas por parte de los *stakeholders*.

2. **Revisión del estado del arte**: Se realiza una revisión crítica de la literatura, así como de las soluciones disponibles (comerciales y de código abierto) para identificar el grado de avance en el problema identificado y como se ha tratado de resolver. Se analizan las iniciativas actuales propuestas para recopilar las métricas de QoS, analizar los modelos de la calidad y los enfoques para el pronóstico de calidad para

los servicios cloud, se llevaron a cabo Revisiones de Literatura (LR). Se exploran, resumen y contextualizan los trabajos preliminares de los temas señalados para identificar las iniciativas y sus carencias.

3. **Solución candidata**: Se proponen una o varias soluciones potenciales al problema identificado. Las soluciones serán evaluadas y depuradas a través de las actividades posteriores (actividades 4, 5 y 6). Para definir una taxonomía de métricas se utiliza una Revisión Sistemática de Literatura (SLR) [26, 27, 28] que hace uso de un metamodelo generado mediante EMF y análisis temático [29] para la clasificación (Objetivo O2). Para proponer el Modelo de Calidad para servicios cloud alineado con la ISO/IEC 25010 [1] y teniendo en cuenta las características específicas del dominio cloud [30, 31, 32, 33], se utilizó la metodología mixta [21] que combina elementos de ambos enfoques descendente (*top-down*) [34] y ascendente (*bottom-up*) [35] (Objetivo O3). El Proceso de Pronóstico ML de QoS para servicios cloud propuesto fue adaptado de Amershi et al. [3] (Objetivo O6) y utilizado para la construcción de modelos de pronóstico de QoS para un servicio cloud (Objetivo O7).

4. **Validación académica**: Se realiza la evaluación de la solución candidata propuesta en un entorno académico, como, por ejemplo, realizando experimentos controlados o estudios de caso involucrando a estudiantes, a expertos del dominio y/o a profesionales de la industria. Esto puede permitir identificar problemas en las soluciones y proponer mejoras antes de aplicarlas en la industria. La definición del proceso de evaluación del modelo de calidad para servicios cloud se basó en el estándar ISO 25040 [14] (Objetivo O4). Evaluamos el rendimiento de los modelos de pronóstico de QoS para un servicio cloud en particular, mediante un experimento controlado [36] (Objetivo O7).

5. **Validación estática**: Consiste en la evaluación de la solución candidata en el contexto de la industria donde profesionales analizan y discuten los resultados del método, técnica o herramienta propuesto con el fin de obtener retroalimentación y realizar las mejoras correspondientes. Para alcanzar el objetivo O5, es decir, evaluar la aplicabilidad del modelo de calidad y su proceso de evaluación, se realizó la operacionalización del modelo de calidad en un entorno industrial mediante un estudio de caso [37]. Expertos en ML de la industria e investigadores evaluaron la propuesta del Proceso de Pronóstico ML de QoS para servicios cloud mediante listas de control con el fin de garantizar su completitud y correcta aplicación (Objetivo O6).

6. **Validación dinámica**: Se realiza la evaluación de la solución candidata en el contexto de la industria, donde profesionales hacen uso del método, técnica o herramienta para resolver un problema determinado usualmente en proyectos piloto. Este resultado es analizado con el fin de obtener retroalimentación y realizar las mejoras correspondientes.

7. **Liberación de la solución**: Se valoran los resultados obtenidos. Las herramientas y el material de entrenamiento son preparados para su uso en contextos industriales.

En esta tesis doctoral nos hemos centrado en las primeras cinco fases de este modelo, debido a que la solución propuesta ha sido desarrollada y evaluada en un contexto académico e industrial. La evaluación académica ha sido exhaustiva, realizando evaluaciones estáticas y dinámicas por medio de estudios de caso y experimento controlado e involucrando a expertos y profesionales de la industria.

A continuación, se describen los métodos empíricos utilizados en cada una de las actividades llevadas a cabo para el desarrollo de esta tesis doctoral.

1.7.1. Análisis del estado del arte y definición de solución candidata

En esta tesis, se ha utilizado el método de **Revisión Sistemática de la Literatura** (*Systematic Literature Review* - SLR) para analizar el estado del arte relacionado con las métricas existentes para evaluar la calidad de los servicios cloud.

Una revisión sistemática de la literatura es una metodología sistemática que permite identificar, analizar y sintetizar el conocimiento existente sobre un tema específico con el fin de establecer el estado del arte, detectar brechas de investigación y fundamentar futuras investigaciones.

En particular, este método ha permitido *desarrollar parte de la solución candidata (Taxonomía de Métricas)* descrita en el modelo de transferencia tecnológica (véase Figura 1.1) Para construir la taxonomía se han utilizado las revisiones sistemáticas para identificar y clasificar las métricas de calidad de los servicios cloud existentes en la literatura. Además, se ha utilizado el método Análisis Temático [29] para soportar la fase de síntesis de los resultados.

Esta metodología fue adaptada a la ingeniería de software por Kitchenham y Charters [26], quienes propusieron guías detalladas para su aplicación en este dominio [28], inspiradas en los métodos de medicina basada en evidencia. También son útiles para los estudiantes de doctorado que deben preparar una descripción general del área temática en la que van a trabajar.

Una SLR es una revisión metodológicamente rigurosa que se planifica formalmente y se ejecuta de forma sistemática y metódica. Las razones más comunes para realizar una SLR son:

- Resumir la evidencia existente relativa a un tratamiento o tecnología, por ejemplo, resumir la evidencia empírica de los beneficios y limitaciones de los modelos de calidad en un dominio específico.
- Identificar las carencias de la investigación actual para sugerir áreas de investigación futura.
- Proporcionar un marco o antecedentes para situar adecuadamente las siguientes actividades de investigación.

No obstante, las SLR también pueden llevarse a cabo para examinar el grado en que las evidencias empíricas apoyan o contradicen las hipótesis teóricas, o incluso para ayudar a generar nuevas hipótesis.

La mayoría de las investigaciones comienzan con algún tipo de revisión bibliográfica. Sin embargo, si la revisión bibliográfica no es exhaustiva ni objetiva, su valor científico es escaso, siendo esta la principal razón para llevar a cabo SLRs. Una SLR sintetiza los trabajos existentes de manera objetiva y que se perciba como tal. De tal forma, que una SLR debe realizarse de acuerdo con una estrategia de búsqueda predefinida y debe permitir evaluar la exhaustividad de la búsqueda. Entre las ventajas de las SLR destacan las siguientes [26]:

- La metodología bien definida hace que sea menos probable que los resultados estén sesgados, aunque no la protege contra el sesgo de publicación en los estudios primarios.
- Pueden proporcionar información sobre los efectos de algún fenómeno en una amplia gama de entornos y métodos empíricos. Si los estudios dan resultados consistentes, las SLR proporcionan evidencia de que el fenómeno es sólido y transferible. Si los estudios dan resultados inconsistentes, se pueden estudiar las fuentes de variación.
- En el caso de los estudios cuantitativos, es posible combinar los datos utilizando técnicas de metaanálisis. Esto aumenta la probabilidad de detectar efectos reales que los estudios individuales más pequeños no pueden detectar.

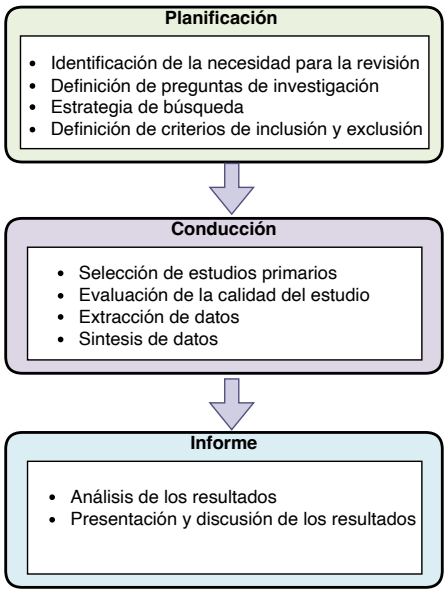


Figura 1.2: Proceso de la Revisión Sistemática de Literatura.

Sin embargo, la principal desventaja de la SLR es que requiere un esfuerzo considerablemente mayor que las revisiones tradicionales. Además, el aumento de la potencia del metanálisis también puede ser una desventaja, ya que es posible detectar pequeños sesgos, así como efectos reales. Algunas de las características que diferencian una revisión sistemática de una revisión tradicional de la literatura por expertos son:

- Las SLRs comienzan definiendo un protocolo de revisión que especifica la pregunta de investigación que se aborda y los métodos que se utilizarán para realizar la revisión.
- Las SLR se basan en una estrategia de búsqueda definida que tiene como objetivo detectar la mayor cantidad posible de literatura relevante.
- Las SLR documentan su estrategia de búsqueda para que los lectores puedan evaluar el rigor, la exhaustividad y la repetibilidad del proceso (teniendo en cuenta que las búsquedas en bibliotecas digitales deben ser adaptadas).
- Las SLR requieren criterios explícitos de inclusión y exclusión para evaluar cada estudio primario potencial.
- Las SLR especifican la información que se obtendrá de cada estudio primario, incluidos los criterios de calidad con los que se evaluará cada estudio primario.
- Una SLR es un requisito previo para el metaanálisis cuantitativo.

Los pasos esenciales del proceso de una SLR son el establecimiento de las preguntas de investigación, la definición del protocolo de revisión, la ejecución del protocolo de revisión y el análisis y la presentación de los resultados (véase la Figura 1.2). En Kitchenham y Charles [28] se pueden encontrar guías completas sobre cómo realizar un SLR.

1.7.2. Validación estática mediante estudio de caso

En esta tesis, se ha utilizado el método de **Estudio de Caso** para la validación empírica de la operacionalización del modelo de calidad y del proceso de pronóstico de QoS basado en ML para los servicios cloud.

Los estudios de caso constituyen un método empírico ampliamente utilizado en la investigación en ingeniería del software, orientado a la comprensión de un fenómeno contemporáneo en su contexto, especialmente cuando es difuso [38, 39, 40]. Este enfoque es recomendable para analizar procesos, modelos o herramientas que involucran interacciones complejas entre factores técnicos, humanos y organizacionales de fenómenos contemporáneos que de manera aislada son difíciles de estudiar [37, 36].

Los estudios de caso centran sus actividades en evaluar las soluciones propuestas en entornos reales, semi-controlados y controlados de la industria, con el fin de obtener retroalimentación contextual, evidencia de aplicabilidad y entendimiento del impacto del cambio tecnológico. Es decir, los estudios de caso permiten analizar cómo las soluciones propuestas se integran en los procesos existentes, cómo los profesionales perciben su aplicabilidad y utilidad, y qué ajustes son necesarios para su adopción efectiva.

En el marco del modelo de transferencia tecnológica, los estudios de caso pueden ser utilizados para la validación del objeto de investigación que conlleva un cambio en el proceso [41, 36], como la validación de artefactos que implican cambios en el proceso o en la práctica profesional, ya que nos permiten observar como las organizaciones perciben, adaptan y aplican estas innovaciones.

Específicamente, este método ha permitido la *validación estática* dentro del modelo de transferencia tecnológica [25]. Se lo utilizó para evaluar la *operacionalización del modelo de calidad* para servicios cloud en un entorno industrial, analizando su aplicabilidad y capacidad para medir la calidad de servicio y su alineación con los requisitos de calidad organizacionales establecidos. Asimismo, el *proceso de pronóstico de QoS para servicios cloud basado en ML* fue examinado por expertos de la industria e investigadores, con el fin de verificar su coherencia y consistencia metodológica.

En definitiva, los estudios de caso permitieron observar el comportamiento de las soluciones en escenarios reales, recoger evidencia empírica sobre su viabilidad y utilidad en la práctica y obtener retroalimentación cualitativa esencial para su mejora continua. Este enfoque fortalece la validez externa de la investigación, al asegurar que los resultados—operacionalización del modelo de calidad y proceso de pronóstico de QoS—son transferibles y aplicables en la industria, cumpliendo el propósito del modelo de transferencia tecnológica al acercar la investigación académica a la práctica profesional.

Un estudio de caso debe ejecutar cinco pasos principales [37, 36]:

1. **Diseño del estudio de caso**: Es necesario planificar la metodología utilizada para la recolección de datos, qué documentos van a analizarse, a qué personas se les va a hacer entrevistas, etc. El plan para un estudio de caso debe contener por lo menos los siguientes elementos:
   - `Objetivo`: qué meta se pretende alcanzar.
   - `El caso`: que se pretende estudiar.
   - `Teoría`: marco de referencia.
   - `Métodos`: cómo recopilar la información.
   - `Estrategia de selección`: donde buscar dicha información
2. **Preparación para la recogida de datos**: normalmente, los datos a analizar provienen de distintas fuentes. Las técnicas de recogida de los datos pueden ser clasificadas en tres niveles:
   - `Primer nivel`: contacto entre investigador y sujetos, la información se recoge en tiempo real.
   - `Segundo nivel`: la información es recogida por el investigador sin actuar con los sujetos durante la recolección.
   - `Tercer nivel`: el investigador analiza los artefactos de forma independiente, es habitual el uso de información agregada.
3. **Recogida de datos**: a través de análisis de datos históricos, entrevistas, métricas u observación.
4. **Análisis de los datos**: a la hora de realizar el análisis de datos se consideran dos aproximaciones distintas, el análisis cuantitativo y el cualitativo. Con respecto al análisis cuantitativo, se utilizan estadística descriptiva para comprender los datos de

una forma informal y modelos predicativos o análisis de correlación para ver cómo se relacionan las distintas actividades del proceso. En cuanto al análisis cualitativo, hay dos vertientes, por un lado, la generación de hipótesis a partir de los datos, como la comparación constante y análisis de casos cruzados [42] y las técnicas de confirmación de hipótesis como triangulación y replicación [42].

5. **Presentación de informes**: es recomendable que los informes sigan ciertas guías para su elaboración, como la [38], aunque si los datos son cualitativos hay cierta libertad a la hora de presentar los informes.

1.7.3. Validación dinámica mediante experimento controlado

En esta tesis, se ha utilizado el **experimento controlado** como método empírico para *construir, validar y comparar modelos de pronóstico de calidad en servicios cloud*, siguiendo el proceso experimental propuesto por Wohlin et al. [36]. La experimentación controlada es esencial en la actividad de *validación académica* en el modelo de transferencia tecnológica [25], ya que permite establecer evidencia empírica sobre la validez, el desempeño y la capacidad de reproducir las soluciones propuestas en un entorno controlado antes de su aplicación en entornos industriales. Este tipo de experimentos posibilita identificar las relaciones causales entre variables, contribuyendo a confirmar hipótesis y validando teorías o modelos bajo condiciones controladas [41, 36]. Este enfoque permite analizar con precisión aspectos como la validación de medidas, la precisión de modelos, la comparación entre enfoques alternativos, garantizando la validez interna de los resultados y minimizando la influencia de factores externos.

El diseño experimental consideró distintos enfoques de modelado—ARIMA, LSTM, y BI-GRU—con el propósito de evaluar su precisión de predicción, eficiencia computacional, y capacidad de generalización. Los experimentos se realizaron utilizando datos reales de monitorización de un servicio cloud en un entorno industrial, lo que permitió obtener evidencia cuantitativa y contextual sobre el rendimiento de cada modelo. Esta aproximación refuerza la validez empírica y la robustez metodológica del proceso de pronóstico propuesto, aportando información relevante para seleccionar el modelo más adecuado y como base antes de su transferencia al entorno industrial.

A diferencia de los experimentos de simulación que emplean únicamente datos sintéticos, el uso de datos reales provenientes de entornos industriales fortalece tanto la validez interna como la externa, al reflejar condiciones auténticas de operación, patrones de carga y variabilidad propia de los servicios cloud. Por ejemplo, las métricas de calidad recopiladas a partir de la monitorización de servicios cloud ofrecen una fuente valiosa de datos históricos que reflejan tales comportamientos.

Esta estrategia conserva las ventajas del control experimental—tales como la replicabilidad, la manipulación de variables independientes y el aislamiento de efectos—al tiempo que la representatividad y transferibilidad de los resultados se incrementan en contextos industriales reales. Así, por ejemplo, es posible analizar con precisión el impacto de distintos algoritmos de pronóstico (variables independientes) sobre métricas de error como RMSE, MAE o MAPE (variables dependientes), bajo escenarios controlados pero con

datos auténticos.

El experimento mantiene los principios fundamentales de la investigación empírica en ingeniería de software, incluyendo la formulación de hipótesis, la definición de métricas de evaluación, el control de amenazas a la validez (interna, externa, constructo y de conclusión) y la aplicación de análisis estadísticos tanto descriptivos como inferenciales. Por tanto, un experimento controlado de simulación con datos reales constituye una estrategia metodológica rigurosa y pragmática para la validación empírica de los modelos de predicción en servicios cloud. Su aplicación proporciona resultados confiables, reproducibles y relevantes en la práctica, especialmente en contextos donde la evaluación directa en producción no es factible por razones de riesgo, seguridad o disponibilidad.

No obstante, la experimentación controlada se complementa mediante la *validación estática* del modelo de calidad y del proceso de modelado ML, donde expertos de la industria evalúan la aplicabilidad y adecuación de la propuesta en contextos reales. Esta combinación de enfoques empíricos —controlado y contextual— asegura una validación integral, que equilibra rigurosidad experimental y relevancia práctica, fortaleciendo la transferencia efectiva de los resultados hacia el entorno industria.

En definitiva, la solución se somete a revisión y aplicación en contextos industriales reales. Esta continuidad metodológica asegura que la investigación trascienda el ámbito experimental, consolidando una transferencia efectiva de conocimiento desde la evidencia empírica hacia la práctica industrial, fortaleciendo así la madurez y aplicabilidad del proceso de pronóstico de QoS en servicios cloud.

La Figura 1.3 muestra las actividades del proceso experimental propuesto por Wohlin et al. [36] y los artefactos generados en cada una de estas actividades.

El detalle de las actividades que se deben realizar se describe a continuación:

1. **Definición del alcance**, en esta actividad, aunque todavía no estén formalmente definidas, sí deberán estar claras las hipótesis del experimento, así como el objetivo y las metas. La meta del experimento es formulada a partir del problema a resolver siguiendo el *framework Goal Question Metric* (GQM) [43] con el esquema de la Tabla 1.1.

Tabla 1.1: Plantilla *Goal-Question-Metric* para la definición de experimentos.

`Analizar`	¿Cuál es el objeto de estudio (producto, proceso, modelo, técnica, teoría, etc.) que se analiza?
`Con el propósito de`	¿Cuál es la intención (evaluar, predecir, controlar, estimar, etc.)?
`Con respecto a`	¿Cuál es el enfoque de calidad (efectividad, coste, mantenibilidad, estabilidad, rendimiento, etc.)?
`Desde el punto de vista de`	¿Cuál es el punto de vista de interés (consumidor, proveedor, gestor de proyectos, usuario, etc.)?
`En el contexto de`	¿Dónde tiene lugar el estudio, sobre qué artefactos y con qué tipo de participantes?

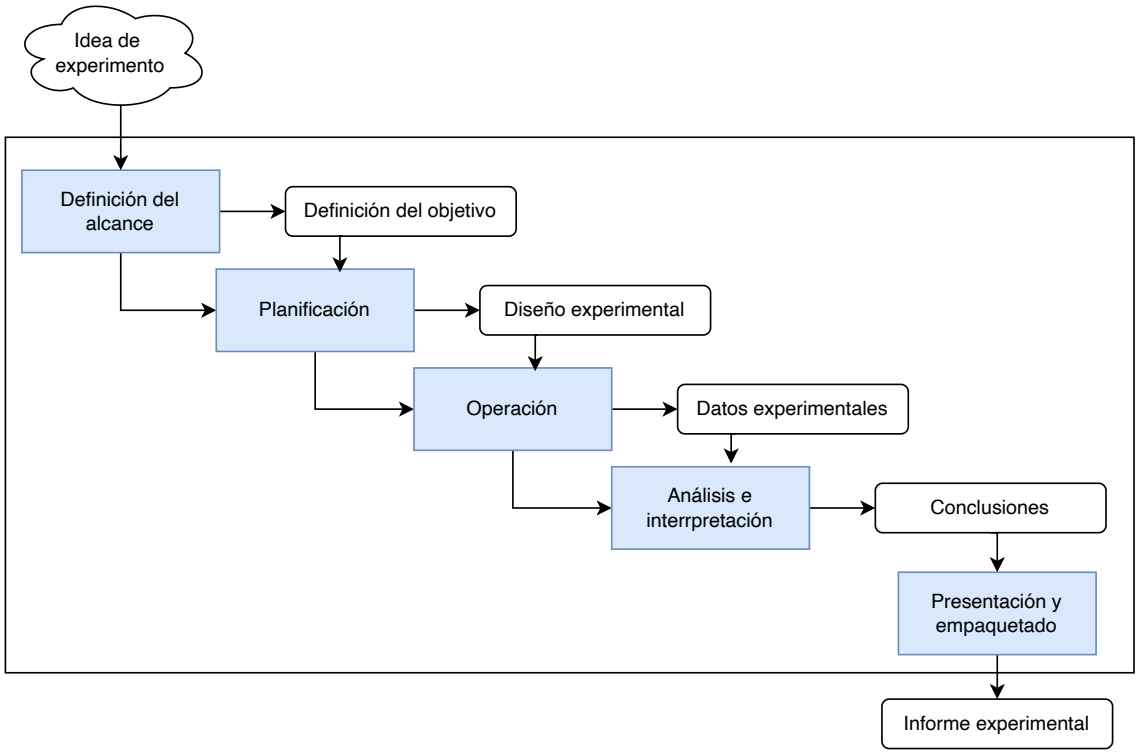


Figura 1.3: Proceso experimental con los artefactos generados en las distintas actividades.

2. **Planificación** es donde se decide el diseño del experimento. En esta actividad se define en profundidad el contexto, que incluye tanto el componente personal (perfiles de los sujetos) como el entorno en el que tendrá lugar. También se especifican formalmente las hipótesis experimentales, incluyendo las hipótesis nulas e hipótesis alternativas, así como las variables, tanto las independientes (entradas) como las dependientes (salidas). Además, se selecciona un diseño experimental y se identifica y prepara la instrumentación a utilizar. El diseño experimental describe, por ejemplo, cómo se llevarán a cabo los ensayos (*online* vs *offline*) o la aleatoriedad de los sujetos. Por último, en la fase de planificación, es donde se ha de considerar la cuestión de la validez de los resultados que cabe esperar.

3. **Operación**, esta es donde se preparan, ejecutan y validan los datos experimentales recogidos. Durante la ejecución, debe asegurarse de llevarla a cabo siguiendo el diseño definido en la fase de planificación y de que se recogen los datos experimentales. Por último, se validan los datos recogidos para asegurar que son correctos y proveen una imagen real del experimento.

4. **Análisis e interpretación**, esta actividad emplea los datos recogidos y validados en la fase de operación. Se emplean estadísticas descriptivas con el fin de entender los datos de manera informal. El siguiente paso es analizar si el conjunto de datos

a considerar debe ser reducido, bien eliminando puntos o eliminando variables, tras analizar si hay variables redundantes que nos ofrecen la misma información. Una vez reducido el conjunto de datos, se llevan a cabo las pruebas de las hipótesis. Se seleccionan las pruebas en función del tipo de escala, de las variables de entrada y del tipo de resultados que se buscan. La interpretación se centra en determinar si, con base en las pruebas, se pueden aceptar o rechazar las distintas hipótesis, esto es, determinar la influencia de las variables independientes sobre las variables dependientes en el caso en que se rechacen las hipótesis nulas.

5. **Presentación y empaquetado**, esta actividad está relacionada con la preparación de la documentación, ya sea un artículo de investigación para difundir los resultados o un paquete de laboratorio con el fin de llevar a cabo replicaciones del experimento.

1.8. Estructura de la tesis

La tesis está organizada siguiendo una secuencia lógica alineada con los objetivos de investigación planteados y con el modelo de transferencia tecnológica adoptado, en el que cada fase—desde la identificación del problema hasta la validación estática—se traduce en un conjunto estructurado de capítulos.

La Figura 1.4 resume gráficamente esta relación mostrando, para cada capítulo, los objetivos específicos que aborda y cómo los resultados obtenidos se integran progresivamente en el marco final de evaluación y pronóstico de la calidad en servicios cloud. A continuación, se detalla esta estructura, explicando cómo cada capítulo contribuye al logro de los objetivos O1–O7:

- **Capítulo 2**. Este capítulo introduce los fundamentos conceptuales necesarios para enmarcar la investigación: servicios cloud, modelos de calidad y técnicas de aprendizaje automático para el pronóstico de series temporales. Aunque este capítulo no da cumplimiento directo a un objetivo específico, proporciona el sustrato teórico sobre el que se construyen los capítulos posteriores y establece la terminología, el alcance conceptual y las motivaciones técnicas que guían la definición del modelo de calidad y del proceso de pronóstico.

- **Capítulo 3**. Este capítulo presenta una revisión del estado del arte en calidad de servicios cloud, métricas y modelos de QoS, así como en técnicas de pronóstico aplicadas a datos de monitorización. Los resultados de este capítulo constituyen la evidencia científica que sustenta el planteamiento del problema y aportan la base necesaria para abordar el **Objetivo O1**, que consiste en identificar, analizar y clasificar las métricas utilizadas en la literatura. Esta revisión inicial permite, además, identificar los desafíos existentes, limitaciones frecuentes y tendencias tecnológicas que justifican la necesidad del modelo de calidad propuesto.

- **Capítulo 4** Este capítulo desarrolla el **Objetivo O2** mediante la construcción de una **taxonomía de métricas de calidad para servicios cloud**. A partir del

análisis temático de la revisión sistemática, se identifican categorías, subcategorías y relaciones entre métricas que reflejan la diversidad del dominio. El capítulo también presenta un **metamodelo de calidad para servicios cloud** que estructura los conceptos relevantes y sus relaciones, proporcionando la base para la definición de la taxonomía. Esta taxonomía no solo organiza el conocimiento existente, sino que también sirve de insumo para la definición del modelo de calidad del Capítulo 5.

- **Capítulo 5**. Este capítulo aborda el **Objetivo O3** mediante la definición de un **modelo de calidad para servicios cloud** alineado con ISO/IEC 25010. En él se presentan las características y subcaracterísticas específicas del dominio cloud, así como sus atributos y métricas asociadas, muchas de las cuales provienen directamente de la taxonomía construida en el Capítulo 4. El modelo resultante constituye una contribución conceptual central de la tesis, basada tanto en evidencia empírica como en estándares internacionales, para representar la calidad de servicios cloud de manera sistemática. Además, este capítulo desarrolla el **Objetivo O4**, ya que se formula un **proceso de evaluación de calidad** basado en la ISO/IEC 25040. Este proceso define las etapas, actividades, roles y artefactos necesarios para evaluar servicios cloud utilizando el modelo propuesto y establece así una metodología reproducible que permitirá su posterior aplicación y validación en entornos reales. De esta manera, el Capítulo 5 integra el modelo conceptual de calidad y su mecanismo de evaluación correspondiente.

- **Capítulo 6**. Este capítulo desarrolla el **Objetivo O5** mediante la evaluación empírica del modelo de calidad y del proceso de evaluación definidos en el capítulo anterior, aplicándolos en un entorno industrial. A partir del modelo de calidad genérico, se construye un modelo personalizado para un servicio cloud específico y se configuran las métricas relevantes para su monitorización. Posteriormente, se aplica el proceso de evaluación con el fin de obtener un diagnóstico del estado actual de la calidad del servicio. La validación industrial realizada aporta evidencia sobre la aplicabilidad y utilidad práctica del marco propuesto, demostrando su capacidad para apoyar la monitorización y evaluación de servicios cloud en contextos organizacionales reales.

- **Capítulo 7**. Este capítulo desarrolla el **Objetivo O6**, mediante la definición de un **proceso sistemático para la construcción de modelos de pronóstico de QoS** basados en aprendizaje automático. Este proceso guía desde la definición de los requisitos del modelo, la preparación de los datos, la selección de algoritmos y arquitecturas, el entrenamiento, la validación y la evaluación del desempeño. Esta contribución metodológica establece las bases necesarias para la experimentación presentada en el capítulo siguiente, que constituye el componente de evaluación dinámica del marco integral de la tesis.

- **Capítulo 8**. Este capítulo da cumplimiento al **Objetivo O7**, al **construir, validar y comparar modelos de pronóstico** aplicando el proceso definido en el Capítulo 7. Mediante un experimento controlado con datos de monitorización procedentes

de un entorno industrial, se evalúan la precisión, la eficiencia computacional y la capacidad de generalización de distintas aproximaciones estadísticas y de aprendizaje profundo. En particular, se desarrollan modelos basados en ARIMA, LSTM, y BI-GRU, empleando un enfoque univariante y de predicción de un solo paso. El capítulo integra los resultados de capítulos previos, ya que utiliza el conjunto de métricas identificadas en el Capítulo 4, se apoya en la comprensión conceptual del dominio presentada en el Capítulo 2 y aplica el proceso metodológico para la construcción de modelos de pronóstico definido en el Capítulo 7.

- **Capítulo 9**. Este capítulo sintetiza los resultados obtenidos, presenta las conclusiones generales, evalúa el grado de cumplimiento del objetivo principal y los objetivos específicos y discute las implicaciones teóricas y prácticas del trabajo. Asimismo, se identifican líneas de investigación futuras que podrían ampliar o extender las contribuciones realizadas.

En conjunto, la tesis sigue una estructura incremental y trazable, en la que cada capítulo aporta elementos conceptuales, metodológicos o empíricos necesarios para construir un marco integral de modelado, evaluación y pronóstico de la calidad de los servicios cloud. La Figura 1.4 refleja esta progresión, mostrando cómo los resultados de cada capítulo alimentan a los siguientes y convergen en una solución completa para anticipar y gestionar la QoS en entornos cloud.

Declaración sobre el uso de inteligencia artificial

Se utilizaron herramientas de inteligencia artificial como apoyo para mejorar la fluidez de la redacción y asistir en el análisis de información, sin que estas intervinieran en el desarrollo metodológico ni en la obtención de resultados. El contenido científico de la tesis es responsabilidad exclusiva del autor.

Adicionalmente, la imagen de la portada fue generada mediante Ideogram AI [44] una herramienta de inteligencia artificial, utilizada exclusivamente con fines gráficos.

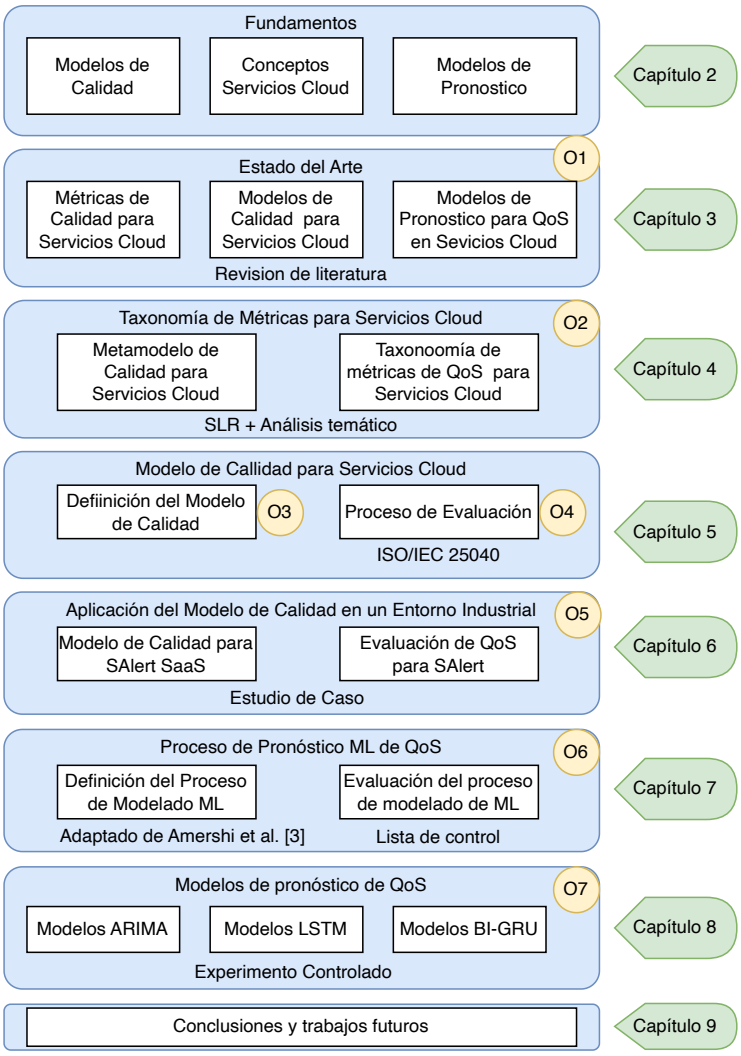


Figura 1.4: Estructura de la tesis.

Capítulo 2

Fundamentos

En este capítulo se presentan los conceptos fundamentales sobre los cuales se asienta esta investigación: los servicios cloud, los modelos de calidad y el aprendizaje automático.

La Sección 2.1 introduce los conceptos fundamentales relacionados con el dominio de los servicios cloud, que incluye definiciones de *stakeholders*, modelos de servicio y arquitectura. La Sección 2.2 introduce los conceptos fundamentales que están relacionados con la calidad, los estándares de calidad de *software* y sus modelos de calidad. La Sección 2.3 introduce los conceptos fundamentales de los modelos de pronóstico con series temporales, la distinción entre pronóstico y predicción, los tipos de series temporales, modelos de pronóstico tradicionales y de aprendizaje automático, además, introduce los modelos de aprendizaje profundo utilizados en esta tesis. La Sección 2.4 presenta la integración conceptual y la relación entre servicios cloud, modelos de calidad y modelos de pronóstico de calidad basados en aprendizaje automático. La Sección 2.5 presenta las conclusiones de los fundamentos conceptuales para esta tesis.

Esta sección presenta los conceptos clave asociados al dominio de la computación en la nube y la calidad, y los relacionados con los modelos de calidad y el aprendizaje automático.

2.1. Servicios cloud

El término computación en la nube ha sido definido de muchas maneras por la academia, la industria, las organizaciones de tecnología, las organizaciones de estandarización y los proveedores de tecnología, reflejando la diversidad de intereses y contextos en los que este paradigma es adoptado. No obstante, pese a la variedad de definiciones existentes, existe un consenso general en torno a la provisión de recursos y funcionalidades de tecnologías de la información como servicios accesibles de manera remota, escalable y bajo demanda.

Una de las definiciones ampliamente aceptadas es la propuesta por el *National Institute of Standards and Technology* (NIST) en el estándar SP 800-145 [30], que define la computación en la nube como "Un modelo para habilitar el acceso ubicuo, bajo demanda por red a un conjunto compartido de recursos computacionales configurables (p. ej., redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente aprovisionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios". Esta definición enfatiza no solo el acceso y la compartición de recursos, sino también la automatización y la reducción de la intervención humana en su gestión.

Por otro lado, desde una perspectiva orientada al mercado, Gartner la define como un estilo de computación en el cual las capacidades escalables, proporcionadas a través de tecnologías de Internet, son provistas masivamente como servicios a múltiples clientes externos [45]. Esta visión pone el énfasis en el modelo de entrega "*como servicio*" y en la capacidad de atender simultáneamente a múltiples consumidores, aspectos clave para la adopción de la nube en entornos empresariales. De forma complementaria, Gens [46] define la computación en la nube como un modelo emergente de desarrollo, despliegue y entrega de productos, servicios y soluciones de TI a través de Internet, destacando su carácter evolutivo y su impacto en los procesos tradicionales de ingeniería de software y operación de sistemas.

Si bien estas definiciones proporcionan una base conceptual de los servicios cloud, no describen de manera explícita sus características que los distinguen. Para abordar esta brecha, NIST SP 800-145 [30] identifica cinco características esenciales para la computación en la nube, que en conjunto permiten comprender el comportamiento y las expectativas asociadas a los servicios cloud:

I. Autoservicio bajo demanda, que permite a los consumidores aprovisionar y administrar de forma independiente los recursos cloud según sea necesario;

II. Amplio acceso a la red, que permite el acceso a los servicios cloud desde cualquier dispositivo a través de la red;

III. Agrupación de recursos, que permite un modelo de tenencia múltiple en el que los recursos informáticos de un proveedor se comparten entre varios consumidores;

IV. Elasticidad rápida, que permite el escalado elástico y automático (hacia arriba o hacia abajo) de los servicios, creando la impresión de recursos ilimitados; y

V. Servicio medido, que permite el control y la medición automáticos del aprovisionamiento, almacenamiento y procesamiento de recursos.

En particular, estas propiedades introducen un alto grado de dinamismo y variabilidad en la provisión de los servicios, lo que tiene implicaciones directas en su comportamiento operativo y en su calidad percibida.

En este contexto, los **servicios cloud** pueden entenderse como la unidad fundamental de consumo de la computación en la nube, encapsulando funcionalidades, recursos y compromisos de calidad ofrecidos a los usuarios finales. Un servicio cloud abstrae la complejidad de la infraestructura subyacente y expone interfaces bien definidas que permiten su utilización, integración y monitorización. Esta abstracción favorece la reutilización y la interoperabilidad, pero también introduce desafíos asociados al control, la previsibilidad y la evaluación del comportamiento del servicio en tiempo de ejecución.

La provisión y el consumo de servicios cloud involucran a múltiples *stakeholders* (véase 2.1.1) con intereses potencialmente divergentes, que demandan niveles de calidad consistentes y predecibles; y, en muchos casos, además, se combinan múltiples servicios cloud para ofrecer soluciones de mayor valor agregado. Para ellos, la calidad del servicio no es únicamente un atributo técnico, sino un factor determinante para la confianza, la satisfacción del usuario y la viabilidad económica del servicio.

2.1.1. Stakeholders

En la computación en la nube, se pueden identificar tres tipos principales de *stakeholders* [31, 32, 33]:

- `Proveedor (CSP)`: la persona, organización o entidad responsable de ofrecer un servicio a los consumidores. Un proveedor construye el software, la plataforma o los servicios de infraestructura solicitados, gestiona la infraestructura técnica necesaria para la prestación de servicios, proporciona servicios a niveles acordados y garantiza la calidad del servicio.

- `Consumidor (CSC)`: la persona u organización que mantiene una relación comercial con los proveedores de servicios cloud y los utiliza. Un consumidor navega por el catálogo de servicios del proveedor, solicita los servicios adecuados, establece contratos de servicio y utiliza el servicio.

- `Facilitador (`*`Broker`*`)`: un consumidor de la nube puede optar por solicitar servicios cloud a un facilitador cloud en lugar de ponerse en contacto directamente con un proveedor de la nube. Un facilitador cloud administra el uso, el rendimiento y la

entrega de servicios y negocia las relaciones entre los proveedores de la nube y los consumidores.

Si bien estos roles específicos de dominio son importantes, para este trabajo, se definen dos roles adicionales que son igualmente relevantes al definir un modelo de calidad para los servicios cloud:

- `Desarrollador`: la persona que contribuye a la prestación del servicio como parte de un equipo o como socio de servicio. Los desarrolladores pueden incluir roles como desarrolladores, integradores, evaluadores y otros.
- `Usuario final`: las personas u organizaciones que son los clientes finales y utilizan el servicio cloud.

2.1.2. Modelos de servicio

La computación en la nube, con varios modelos de servicio, se puede describir como configuraciones funcionales jerárquicas que determinan el grado de abstracción de los recursos y su acceso que ofrecen al consumidor. Así como las responsabilidades específicas que asumen tanto el proveedor como el consumidor del servicio. Estos tipos de servicio son:

- `Infraestructura como Servicio (IaaS)`, el proveedor proporciona los recursos de infraestructura como servicio, es decir, acceso a redes, almacenamiento, procesamiento y virtualización; además, gestiona el centro de datos y los gastos asociados, como calefacción, refrigeración, seguridad física, etc. Por otra parte, los consumidores aprovisionan y utilizan los recursos cloud donde implementan y controlan sistemas operativos, almacenamiento y aplicaciones, además es posible un control limitado de determinados componentes de infraestructura (ej. hosts, firewalls, etc.) [47, 30].
- `Plataforma como Servicio (PaaS)`, agrega una capa sobre la infraestructura cloud, proporcionando un entorno completo de desarrollo. Los proveedores gestionan el ambiente de alojamiento (*hosting*) y su infraestructura, ofreciendo herramientas que incluyen entornos de desarrollo, lenguajes de programación, herramientas de despliegue compatibles con la infraestructura de la nube, parches y aplicaciones propias o adquiridas. Por su parte, los consumidores utilizan estos recursos para implementar aplicaciones aprovechando bibliotecas, entorno de ejecución y servicios compatibles con el proveedor. Esto les permite enfocarse con mayor facilidad en la lógica del negocio y la gestión de datos, pues se descarga de la administración del entorno y de la infraestructura. En resumen, aunque los consumidores no administran ni controlan la infraestructura cloud, sí tienen control sobre las aplicaciones implementadas y la configuración de su entorno de ejecución [47, 30].
- `Software como Servicio (SaaS)` agrega una capa de abstracción que corresponde a las aplicaciones y los datos. Los proveedores proporcionan las aplicaciones listas

para usar, implementadas y operativas sobre una infraestructura cloud que ellos mismos gestionan y controlan. Los consumidores acceden a estas aplicaciones a través de Internet y pagan por su uso. Aunque no gestionan ni la infraestructura subyacente ni la plataforma que soporta las aplicaciones, sí pueden personalizar ciertos aspectos mediante ajustes de configuración disponibles en el propio software. En definitiva, los usuarios utilizan soluciones completas sin preocuparse por la instalación, el mantenimiento o la gestión técnica, limitándose a adaptar la aplicación a sus necesidades operativas mediante configuraciones específicas [47, 30].

- `Función como Servicio (FaaS)`, este modelo promueve una arquitectura sin servidor (*serverless*), impulsada por eventos, lo que permite a los desarrolladores crear, implementar y supervisar aplicaciones de microservicios como funciones individuales sin necesidad de gestionar los recursos fundamentales [48]. El consumidor define la lógica de negocio, los desencadenantes de ejecución (*triggers*) y configura parámetros como el tiempo máximo de ejecución, mientras el proveedor orquesta el resto del ciclo de vida de ejecución. El proveedor de servicios es responsable de toda la infraestructura, aprovisionamiento dinámico, escalado automático, balanceo de carga, gestión del tiempo de ejecución, monitorización, seguridad y facturación por uso real (por invocación y duración de ejecución). Los consumidores despliegan y ejecutan las funciones de software en respuesta a eventos, sin necesidad de gestionar servidores ni recursos subyacentes [47].

Adicionalmente, existe una capa transversal en las arquitecturas de los servicios cloud, que corresponde a la *gestión y orquestación* [33], la cual engloba los mecanismos y servicios necesarios para operar, monitorizar y administrar los servicios cloud (IaaS, PaaS, SaaS, FaaS). Esta capa sustenta varios modelos emergentes, entre los más conocidos están MaaS (*Monitoring as a Service*), realiza la monitorización del estado de los servicios; SLAaaS (*SLA Management as a Service*), realiza la gestión de los niveles de servicio; y otros relacionados con la seguridad, cumplimiento y auditoría, facturación, etc.; que en conjunto dan soporte, habilitan y potencian la entrega efectiva de los servicios cloud.

La Figura 2.1 ilustra la comparación entre IaaS, PaaS, FaaS y SaaS, donde se detallan las capas de abstracción de los modelos de servicio y la división de responsabilidades entre el CSP y el CSC en la gestión de componentes. En particular, este trabajo se centra en los modelos IaaS, PaaS, y SaaS.

2.1.3. Arquitectura

La arquitectura cloud se refiere a la forma en que varios componentes tecnológicos, como el *hardware*, los recursos virtuales, las capacidades de software y los sistemas de red virtual, interactúan entre sí y se conectan para crear entornos de computación en la nube. Sirve como un modelo conceptual que define la forma óptima de combinar recursos para crear un entorno en la nube adaptado a las necesidades empresariales específicas. Esto permite el despliegue distribuido y escalonado de capas de pila de tecnología, que incluyen tanto componentes físicos (p. ej., *hardware*) como componentes abstractos (p. ej.,

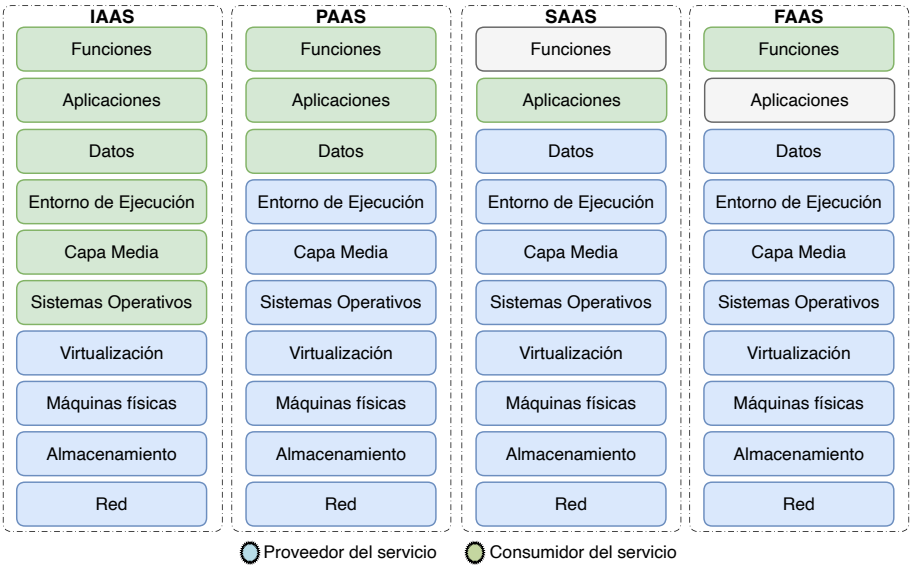


Figura 2.1: Responsabilidades del proveedor y consumidor por modelos de servicios.

aplicaciones o *software*) [30]. Desde una perspectiva funcional, la ISO/IEC 17789 describe la arquitectura como un conjunto de componentes agrupados en capas funcionales, donde cada capa actúa como interfaz con las capas subyacentes [33].

Por lo tanto, la arquitectura de un servicio cloud constituye un elemento fundamental para comprender cómo se materializan las propiedades de calidad y cómo estas pueden ser observadas, evaluadas y gestionadas en la práctica. Esta relación entre arquitectura y calidad es especialmente relevante en entornos cloud, donde la variabilidad, la elasticidad y la distribución de responsabilidades introducen nuevos desafíos para la evaluación sistemática de la calidad.

A diferencia de las arquitecturas monolíticas tradicionales, los servicios cloud se caracterizan por una organización estratificada en capas, una elevada heterogeneidad de componentes y la presencia de mecanismos de control dinámico que permiten adaptar el comportamiento del sistema a condiciones cambiantes de carga, disponibilidad o demanda. En este contexto, disponer de un modelo conceptual de arquitectura resulta clave para establecer una base común que permita razonar de forma sistemática sobre la calidad del servicio.

La Figura 2.2 presenta un **modelo conceptual** de la arquitectura de servicios cloud que abstrae los elementos estructurales y funcionales más relevantes, así como las relaciones entre ellos. En el centro del modelo se sitúa la entidad *CloudArchitecture*, que representa una instancia concreta de arquitectura cloud desplegada en un determinado contexto operativo. Esta entidad recoge información básica sobre el nombre de la arquitectura, su descripción y el contexto de despliegue, permitiendo diferenciar entre distintas configuraciones arquitectónicas de un mismo servicio o entre servicios distintos.

La arquitectura cloud se organiza explícitamente en **capas arquitectónicas** (*Archi-*

tectureLayer), reflejando el modelo clásico de servicios cloud. En concreto, se distinguen tres tipos de capas: ApplicationLayer (SaaS), *PlatformLayer* (PaaS) e *InfrastructureLayer* (IaaS). Esta jerarquía permite capturar la separación de responsabilidades entre la lógica de aplicación, los servicios de plataforma que la soportan y la infraestructura subyacente. La presencia explícita de estas capas en el modelo facilita el análisis de la calidad desde una perspectiva multicapa, permitiendo asociar atributos y métricas de calidad al nivel arquitectónico donde se originan o se manifiestan.

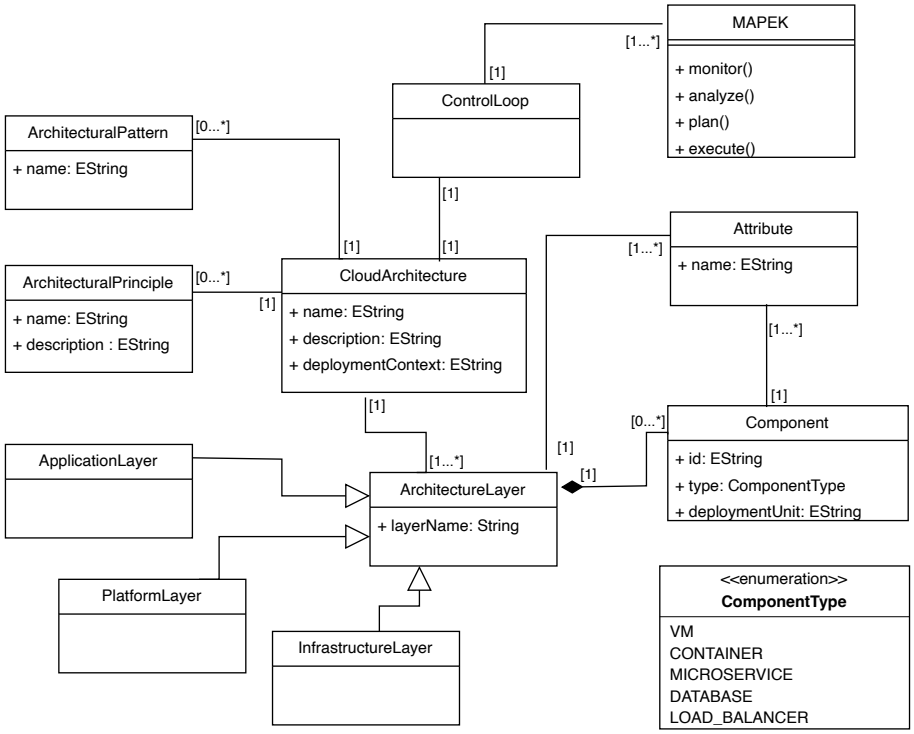


Figura 2.2: Modelo conceptual de la arquitectura de servicios cloud.

Cada capa arquitectónica está compuesta por uno o más **componentes** (*Component*), que representan las unidades técnicas desplegables del servicio, tales como microservicios, contenedores, máquinas virtuales, bases de datos o balanceadores de carga. El tipo de componente se especifica mediante la enumeración *ComponentType*, lo que permite diferenciar de forma explícita la naturaleza y el rol de cada elemento dentro de la arquitectura. Esta distinción resulta especialmente relevante para la evaluación de la calidad, ya que distintos tipos de componentes presentan comportamientos, métricas y requisitos de calidad diferentes.

Los componentes se caracterizan mediante un conjunto de **atributos** (*Attribute*), que representan propiedades observables y potencialmente medibles, como el tiempo de respuesta, la utilización de recursos, la tasa de errores o la disponibilidad. Estos atributos *constituyen el punto de conexión entre la arquitectura del sistema y los modelos de cali-*

dad, ya que permiten vincular los elementos estructurales del servicio con métricas de QoS concretas capturadas mediante herramientas de monitorización. De este modo, el modelo facilita la trazabilidad entre la arquitectura cloud y la evaluación de la calidad del servicio.

El modelo incorpora asimismo la noción de **principios arquitectónicos** (*ArchitecturalPrinciple*) y **patrones arquitectónicos** (*ArchitecturalPattern*), que representan decisiones de diseño de alto nivel que influyen directamente en las propiedades de calidad del sistema. La inclusión explícita de estos conceptos permite razonar sobre cómo determinadas decisiones arquitectónicas (p. ej., el uso de microservicios, replicación o balanceo de carga) afectan a características como la escalabilidad, la fiabilidad o el rendimiento.

Finalmente, el modelo contempla la existencia de **bucles de control** (*ControlLoop*) basados en el paradigma MAPE-K (*Monitor–Analyze–Plan–Execute*), que representan los mecanismos de adaptación y autogestión del servicio cloud. Estos bucles de control se apoyan en la monitorización continua de atributos y métricas para analizar el estado del sistema, planificar acciones correctivas y ejecutarlas de forma automática o semiautomática. La presencia de estos elementos refleja la naturaleza dinámica de los servicios cloud y anticipa la necesidad de enfoques proactivos y predictivos para la gestión de la calidad, que serán abordados en capítulos posteriores.

En conjunto, la Figura 2.2 sintetiza una visión estructurada y coherente de la arquitectura de servicios cloud, sirviendo como base para la definición de modelos de calidad, procesos de evaluación y mecanismos de monitorización y pronóstico. Al capturar explícitamente las capas, los componentes, sus atributos y los bucles de control, este modelo facilita la integración entre la arquitectura técnica del servicio y los enfoques sistemáticos de evaluación y gestión de la calidad desarrollados a lo largo de esta tesis.

Este modelo conceptual resulta coherente con las propuestas de la literatura sobre estilos arquitectónicos cloud, que enfatizan la necesidad de capturar tanto la estructura como los mecanismos dinámicos de adaptación. En este contexto, Pahl et al. [49] definen un estilo de arquitectura cloud, que incluye principios como la orientación a servicios, la virtualización, la adaptación y la incertidumbre, así como patrones como microservicios, modelos de tiempo de ejecución, bucles de control y arquitectura de controladores.

Los *principios* se centran en capturar los factores clave de la arquitectura:

I. `Orientación al servicio`: Habilitada al compartir recursos para optimizar los costes (p. ej., virtualización);

II. `Virtualización`: Gestiona el conjunto de recursos a través de tres capas: infraestructura, es decir, la infraestructura física separada para crear recursos virtuales; plataforma (abstracción a través de la contenerización) y aplicación (las aplicaciones están separadas tanto de la plataforma como de la infraestructura);

III. `Adaptación`: Los sistemas se adaptan para cumplir con los requisitos de QoS en la nube utilizando información del usuario y del entorno; y

IV. `Incertidumbre`: Particularmente en soluciones heterogéneas en la nube, surge la necesidad de medir en diferentes capas y mapear entre los diferentes niveles. Los niveles

superiores representan la aplicación, mientras que los niveles inferiores representan la infraestructura. La monitorización continua del estado de las aplicaciones y los recursos que las alojan es crucial para prevenir posibles desviaciones y reducir la incertidumbre [49].

Por otro lado, los *patrones* están diseñados para mapear los principios de desarrollo e implementación:

I. `Microservicios`: Permiten que las aplicaciones de software se diseñen como servicios desplegables de forma independiente, que también pueden reemplazarse y actualizarse de forma independiente (p. ej., máquinas virtuales, *Docker*, *Kubernetes*). Este enfoque descentraliza las implementaciones y admite la entrega continua desde el desarrollo hasta la operación, a menudo mediante prácticas de DevOps (p. ej., empaquetado, orquestación);

II. `Modelos de ejecución`: Son esenciales para la gestión dinámica de la calidad y permiten la autoadaptación en la nube. Nuestro estudio se centra en la adaptación, donde el modelo de calidad conecta con los requisitos de calidad y permite la personalización según las necesidades del negocio;

III. `Bucle de control`: Normalmente soportado por un controlador, este bucle impulsa la adaptación del tiempo de ejecución en entornos cambiantes de los servicios cloud. Se abordan dos mecanismos clave: la predicción implica el análisis de datos, la identificación de patrones y la reducción del impacto de la incertidumbre, la utilidad implica la identificación de objetivos, y la medición del progreso y la orientación de la toma de decisiones durante el proceso de adaptación; y

IV. `Arquitectura del controlador`: Esto ayuda a vincular y sincronizar las capas y niveles de la arquitectura en función de las preocupaciones de calidad (p. ej., planificar acciones correctivas, garantizar la elasticidad de los recursos y mejorar el intercambio de conocimientos [49]).

En cuanto a los componentes cloud, Bernstein [50] exploró la evolución de los contenedores y su relación con los sistemas cloud. El autor sugiere que los contenedores son la tecnología de implementación central para los entornos cloud. La encuesta de la CNCF [51] confirmó el continuo aumento en la adopción de estas tecnologías, con un 92 % que utiliza contenedores, un 83 % que utiliza *Kubernetes* y otros. Entre las ventajas de estas tecnologías se encuentran el aislamiento de procesos, la virtualización ligera y la implementación simplificada de aplicaciones.

2.2. Modelos de calidad

En el contexto de los servicios cloud, la calidad adquiere un papel fundamental, dado que estos servicios se consumen de forma remota y su comportamiento depende de infraestructuras compartidas, dinámicas y altamente automatizadas. Factores como la elasticidad, la multitenencia y la variabilidad de la carga de trabajo introducen fluctuaciones

en los atributos de calidad observados, lo que dificulta su evaluación y control mediante enfoques reactivos. En consecuencia, la calidad de un servicio cloud no solo debe ser monitorizada, sino también definida y gestionada de manera sistemática a lo largo de su ciclo de vida.

Para abordar esta complejidad, la evaluación de la calidad de los servicios cloud se apoya en marcos de referencia estandarizados que permiten estructurar, medir y comunicar los distintos aspectos de la calidad entre los *stakeholders*. Los estándares de evaluación de calidad propuestos por la ISO/IEC proporcionan un fundamento conceptual ampliamente aceptado para la definición y evaluación de la calidad del producto de *software*.

Esta sección introduce la definición de calidad y los estándares de evaluación propuestos por la ISO/IEC, con especial énfasis en los modelos de calidad. Los modelos de calidad están diseñados para descomponer el concepto abstracto de calidad al proporcionar una definición más operativa de la calidad de un producto de *software* (o servicio cloud) a través de la descomposición jerárquica en elementos observables y medibles.

2.2.1. Concepto de calidad

El estándar ISO 8402 [52] define la calidad como "la totalidad de los rasgos y características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades declaradas o implícitas". Por otro lado, el IEEE 729-83 [53] define la calidad como "el grado en que el cliente o usuario percibe que el software cumple con sus expectativas". En contraste, el DoD 2168 [54] define la calidad como la "capacidad del producto de software para satisfacer los requisitos establecidos". Mientras que la ISO/IEC 25010 [1] define la calidad del software como el "grado en que un producto de software satisface las necesidades declaradas e implícitas cuando se utiliza en condiciones específicas".

A partir de estas definiciones, queda claro que la calidad es relativa y el grado de calidad esperado para un producto dependerá siempre de los requisitos o necesidades que se deseen satisfacer. Por lo tanto, la evaluación de la calidad de un producto siempre implicará una comparación entre los requisitos preestablecidos y las características del producto desarrollado.

2.2.2. Estándares de calidad del software

La familia de normas ISO/IEC 25000 , conocida como SQuaRE (*Systems and Software Quality Requirements and Evaluation*), constituye un marco integral para la especificación, medición y evaluación de la calidad de los productos de software [55]. Esta norma agrupa cinco divisiones que abarcan modelos de calidad, métricas de calidad, requisitos y procesos de evaluación de calidad. Esta norma surge de la integración de los estándares ISO/IEC 9126 [56] e ISO/IEC 14598 [57]. Su principal objetivo es proporcionar una estructura común y coherente que permita evaluar la calidad del software de forma sistemática y replicable.

En este trabajo, se hace especial énfasis en dos estándares de esta familia, los cuales constituyen la base metodológica para el desarrollo y evaluación del modelo de calidad propuesto para servicios cloud:

- `ISO/IEC 25010` [1] define dos modelos de calidad —uno centrado en el producto *software* y otro en su uso— que detallan características y subcaracterísticas fundamentales para la evaluación de la calidad. Este estándar constituye la línea base conceptual para el diseño del modelo de calidad para los servicios cloud propuesto en el presente trabajo (ver capítulo 5).

- `ISO/IEC 25040` [14] presenta un modelo de referencia para llevar a cabo evaluaciones de calidad de software. Este marco metodológico guía la operacionalización del modelo de calidad propuesto, asegurando su aplicabilidad en entornos reales y su alineación con prácticas estandarizadas (ver capítulo 6).

El uso de estos estándares responde a dos motivaciones principales: (1) garantizar la coherencia en la terminología y evitar ambigüedades en la comunicación entre los distintos actores involucrados; y (2) basarse en un conjunto de características de calidad ampliamente reconocido por la comunidad científica e industrial, lo que facilita la aceptación, comparación y reutilización de los modelos desarrollados.

2.2.2.1. Estándar ISO/IEC 25010

El propósito de este estándar es la definición de un modelo de calidad centrado en las propiedades intrínsecas del software, que sirva como un marco de referencia para la evaluación sistemática de productos de software. Este modelo constituye el núcleo estructural sobre el cual se articula todo el proceso de evaluación de calidad, proporcionando una base coherente y estandarizada para su análisis y mejora continua. ISO/IEC 25010 [1] contempla dos modelos complementarios: el modelo de calidad del producto y el modelo de calidad en uso. El primero evalúa propiedades internas y externas del software, tales como mantenibilidad, fiabilidad y seguridad, mientras que el segundo se enfoca en la experiencia del usuario final, valorando aspectos como la efectividad, eficiencia, satisfacción y cobertura del contexto. Juntos, estos modelos permiten una visión integral de la calidad, abarcando tanto las características técnicas del producto como su impacto en el entorno de uso.

En el presente trabajo, el modelo definido por ISO/IEC 25010 [1] se adopta como línea base para construir un modelo de calidad específico para servicios cloud (ver Capítulo 5). No obstante, dado su carácter genérico, el principal desafío radica en la operacionalización y aplicación a este dominio en particular. Muchas de las subcaracterísticas propuestas en el estándar siguen siendo abstractas y no se pueden medir objetivamente, lo que dificulta una aplicación directa. Esta limitación podría explicar, como revelan varias encuestas empíricas [7, 58, 21], por qué menos del 28 % de las empresas utilizan modelos de calidad estándar y la mayoría opta por variantes de estos modelos.

En 2023, se publicó una nueva versión del estándar ISO/IEC 25010 [59], que actualiza y extiende el modelo de calidad propuesto originalmente en 2011, en el contexto de la familia SQuaRE. Esta actualización introduce cambios significativos tanto en la estructura como en el alcance del modelo de calidad. En particular, se incorpora un modelo de calidad específico para los sistemas inteligentes, incluyendo consideraciones específicas para

productos que integran inteligencia artificial (IA), aprendizaje automático (ML) y autonomía adaptativa. Además, se redefinen y reorganizan algunas características existentes para mejorar su aplicabilidad y alineación con prácticas actuales de ingeniería de software. Estas mejoras reflejan una evolución del estándar orientada a dar respuesta a los desafíos tecnológicos actuales y a las demandas emergentes de calidad en sistemas cada vez más complejos, autónomos y dinámicos [59].

Modelo de calidad de producto

La calidad del producto de software se entiende como el grado en que dicho producto satisface los requisitos de sus usuarios, aportando de esta manera un valor. El modelo de calidad del producto mide tanto las propiedades *internas*—como la especificación funcional, el diseño o la arquitectura—, como las propiedades *externas*—como el comportamiento dinámico, rendimiento y fiabilidad—. El modelo se organiza jerárquicamente mediante una estructura multinivel que parte de ocho características principales, las cuales se descomponen en treinta y un subcaracterísticas, como se muestra en la Figura 2.3

Calidad del producto de Software							
Adecuación funcional	Eficiencia de desempeño	Compatibilidad	Usabilidad	Fiabilidad	Seguridad	Mantenibilidad	Portabilidad
Completitud funcional	Comportamiento temporal	Coexistencia	Inteligibilidad	Madurez	Confidencialidad	Modularidad	Adaptabilidad
Corrección funcional	Utilización de recursos	Interoperabilidad	Capacidad de aprendizaje	Disponibilidad	Integridad	Reusabilidad	Instalabilidad
Pertinencia funcional	Capacidad		Capacidad de Operación	Tolerancia a fallos	No repudio	Analizabillidad	Reemplazabilidad
			Protección frente a errores de usuario	Capacidad de recuperación	Autenticidad	Capacidad de modificación	
			Estética de la interfaz de usuario		Responsabilidad	Capacidad de prueba	
			Accesiibilidad				

Figura 2.3: Modelo de calidad del producto de software ISO/IEC 25010 [1].

En cuanto al modelo de calidad del producto, este ha sido extendido y adaptado a tres características—capacidad de interacción, flexibilidad y protección— junto con 13 subcaracterísticas como se muestra en la Figura 2.4. Se agrega la característica de *protección* para abarcar las restricciones operativas, identificación de riesgos, protección ante fallos, advertencia de peligros e integración segura. La usabilidad y portabilidad cambian por *capacidad de interacción* y *flexibilidad*. La capacidad de interacción fortalece el enfoque en accesibilidad universal—con inclusividad y asistencia al usuario— y transparencia—con autodescriptividad—particularmente en sistemas con IA. La *flexibilidad* se amplía y fortalece con escalabilidad y adaptabilidad, con aspectos de adaptabilidad automatizada, automantenimiento y soporte al reentrenamiento en modelos de IA. La *seguridad* se integra con nuevos elementos, como la resistencia a la manipulación algorítmica, privacidad diferencial y mecanismos de control para la toma de decisiones automatizadas.

Modelo de calidad en uso

La calidad en uso es el grado en que un producto o sistema puede ser utilizado por usuarios específicos para satisfacer sus necesidades y alcanzar objetivos específicos con eficacia, eficiencia, ausencia de riesgo y satisfacción en contextos de uso específicos. El

Calidad del producto de Software								
Adecuación funcional	**Eficiencia de desempeño**	**Compatibilidad**	**Capacidad de interacción**	**Fiabilidad**	**Seguridad**	**Mantenibilidad**	**Flexibilidad**	**Protección**
Completitud funcional	Comportamiento temporal	Coexistencia	**Reconocibilidad de adecuación**	**Ausencia de fallos**	Confidencialidad	Modularidad	Adaptabilidad	**Restricción operativa**
Corrección funcional	Utilización de recursos	Interoperabilidad	Capacidad de aprendizaje	Disponibilidad	Integridad	Reusabilidad	**Escalabilidad**	**Identificación de riesgos**
Pertinencia funcional	Capacidad		Capacidad de Operación	Tolerancia a fallos	No repudio	Analizabillidad	Instalabilidad	**Protección de fallos**
			Protección frente a errores de usuario	Capacidad de recuperación	Responsabilidad	Capacidad de modificación	Reemplazabilidad	**Advertencia de peligro**
			Involucración del usuario		Autenticidad	Capacidad de prueba		**Integración segura**
			Inclusividad		**Resistencia**			
			Asistencia al usuario					
			Auto-descriptiviidad					

Figura 2.4: Modelo de calidad del producto de software ISO/IEC 25010 [1].

modelo de calidad en uso mide las propiedades de calidad de un producto de software relacionadas con el resultado de la interacción cuando se utiliza un producto en un contexto concreto. El modelo de calidad en uso se organiza jerárquicamente y se divide en cinco características de calidad: eficacia, eficiencia, satisfacción, ausencia de riesgo y cobertura del contexto, que, a su vez, se descomponen en nueve subcaracterísticas que se muestran en la Figura 2.5.

2.2.2.2. Estándar ISO 25040

El estándar ISO 25040 [14] hace referencia al proceso para llevar a cabo la evaluación de un producto de software, proporcionando directrices precisas sobre las actividades y tareas que deben llevarse a cabo para garantizar una evaluación rigurosa, objetiva y reproducible. Su propósito es asegurar que las evaluaciones sean útiles para los diferentes *stakeholders* durante el ciclo de vida del software. De acuerdo a este estándar, la evaluación de software se estructura en cuatro actividades principales, cada una de las cuales contribuye a garantizar la trazabilidad, validez y utilidad de los resultados obtenidos:

Calidad en uso				
Efectividad	**Eficiencia**	**Satisfacción**	**Ausencia de riesgo**	**Cobertura de contexto**
Efectividad	Eficiencia	Utilidad	Mitigación de riesgo económico	Completitud de contexto
		Confianza	Mitigación de riesgo de salud y seguridad en uso	Flexibilidad
		Placer	Mitigación de riesgo del entorno	
		Confort		

Figura 2.5: Modelo de calidad en uso ISO/IEC 25010 [1].

1. `Establecer el propósito de la evaluación`. Las tareas que conlleva esta actividad son determinar el objetivo de la evaluación, identificar a los *stakeholders* y especificar los requisitos de calidad del producto de software e identificar qué par-

tes del producto serán evaluadas. Este paso es crucial para alinear el proceso de evaluación con los objetivos del producto y el contexto de aplicación.

2. `Especificar la evaluación`. Se seleccionan las métricas e indicadores que se utilizarán para medir las características y atributos de calidad seleccionados. También se establecen los criterios de decisión y umbrales de aceptación asociados a las métricas, permitiendo la interpretación de resultados en términos de cumplimiento o no de los objetivos de calidad establecidos.

3. `Diseñar la evaluación`. Esta actividad está relacionada con la planificación y el diseño de las actividades de evaluación. Incluye la definición del conjunto de pasos, técnicas, herramientas, fuentes de información y recursos necesarios para llevar a cabo la recolección y el análisis de datos. El diseño debe tener en cuenta tanto la viabilidad técnica como la reproducibilidad del proceso de evaluación.

4. `Ejecutar la evaluación`. Completar la evaluación siguiendo el diseño establecido. Esto incluye la recolección y análisis de los datos, la verificación de la calidad y consistencia de los resultados y la elaboración del informe de evaluación. El informe de evaluación debe ser claro y útil para la toma de decisiones, proporcionando evidencia suficiente para sustentar los resultados.

En definitiva, el proceso definido por la ISO/IEC 25040 [14] garantiza una evaluación estructurada, repetible y orientada a la mejora continua, que puede adaptarse a diferentes tipos de productos de software y dominios de aplicación. El estándar ISO 25040 guía el proceso que describe las actividades realizadas para llevar a cabo la evaluación usando el modelo propuesto en este trabajo (ver capítulo 6).

2.3. Modelos de pronóstico de series temporales

La naturaleza dinámica de los servicios cloud hace que muchos de sus atributos de calidad —como la latencia, el tiempo de respuesta, la disponibilidad o el *throughput*— se comporten como **series temporales**, es decir, como secuencias de datos recogidos a intervalos regulares que reflejan la evolución del comportamiento del servicio en el tiempo. Comprender y modelar esta evolución es esencial para anticipar degradaciones, planificar la capacidad, cumplir con acuerdos de nivel de servicio (SLA) y garantizar una experiencia de usuario estable.

A diferencia de los métodos tradicionales, como ARIMA o modelos estadísticos, los algoritmos de aprendizaje automático, incluyendo redes neuronales recurrentes (RNN), máquinas de soporte vectorial (SVM), modelos de árboles de decisión, y arquitecturas basadas en *transformers* han logrado gran precisión al capturar dependencias temporales de largo plazo y manejar grandes volúmenes de datos multivariados provenientes de sensores, dispositivos IoT, o servicios cloud.

La integración de estos enfoques facilita la toma de decisiones proactivas en aplicaciones para la gestión eficiente de servicios cloud, donde la demanda de recursos, los costos

operativos y la calidad del servicio (QoS) dependen de la capacidad de anticipar patrones en datos de monitorización [60].

En esta sección se profundiza en los fundamentos del pronóstico de series temporales y su relación directa con métricas de QoS en entornos cloud. Se presentan también las principales técnicas de aprendizaje automático utilizadas en este contexto.

2.3.1. Pronóstico vs. predicción en métricas de QoS

Es fundamental distinguir entre los conceptos de pronóstico (*forecasting*) y predicción (*prediction*), ya que aunque a menudo se usan de manera intercambiable, responden a objetivos y alcance diferentes. Esta distinción es especialmente relevante en entornos cloud, donde muchas métricas presentan comportamientos recurrentes, tendencias y fluctuaciones asociadas a la carga del sistema, la demanda del servicio o el funcionamiento de la infraestructura subyacente. Por ejemplo, la **latencia** suele aumentar durante picos de tráfico y disminuir cuando la carga baja; el **tiempo de respuesta** puede mostrar patrones diarios (picos diurnos, valles nocturnos); la **disponibilidad** puede presentar fallos intermitentes debidos a reinicios automáticos, tareas de mantenimiento o congestión en la red; el **throughput** está influido por la demanda simultánea de usuarios y por el grado de elasticidad del servicio. Estos fenómenos no son aleatorios, sino que reflejan estructuras temporales que pueden ser modeladas y anticipadas.

El **pronóstico** se centra en realizar predicciones cuantitativas sobre valores futuros de una variable basándose exclusivamente en patrones temporales presentes en sus observaciones pasadas y suponiendo cierta estabilidad en las relaciones subyacentes [60]. Por ejemplo, el pronóstico de métricas como el uso de CPU, el ancho de banda o los costes operativos del servicio, donde métodos tradicionales (ARIMA) y actuales (LSTM, BI-GRU) son ampliamente utilizados. Así, el pronóstico permite capturar no solo variaciones puntuales, sino también tendencias, estacionalidades y patrones cíclicos propios de los servicios cloud.

Por el contrario, como señalan Cerqueira et al. [61], la **predicción** abarca un espectro más amplio, incluyendo la inferencia de eventos discretos (p. ej., fallos de *hardware*) o clases (p. ej., detección de anomalías), donde modelos como redes neuronales convolucionales (CNN) procesan no solo datos temporales, sino también contextuales (logs, métricas multidimensionales). Mientras el pronóstico asume una dependencia temporal explícita, la predicción puede integrar señales no secuenciales, lo cual puede ser útil en diferentes escenarios de la nube, como la calidad del servicio, el balanceo de carga y la asignación dinámica de recursos.

2.3.2. Tipos de series temporales

En el análisis de datos temporales es fundamental comprender los diferentes tipos de series temporales para seleccionar el enfoque de modelado y pronóstico más adecuado. Las series temporales pueden clasificarse en varios tipos según diferentes criterios como su estacionalidad, tendencia y tipo de variable.

2.3.2.1. Estacionaria y no estacionaria

Las *series estacionarias* se caracterizan por mantener constantes sus propiedades estadísticas fundamentales—como la media, la varianza y la autocorrelación—a lo largo del tiempo. Esta estabilidad las convierte en candidatas ideales para el uso de modelos tradicionales de pronóstico, como ARMA o ARIMA, una vez se ha verificado su estacionariedad mediante pruebas como la de Dickey-Fuller. Verificar la estacionariedad es un paso crítico antes de aplicar cualquier modelo estadístico basado en series temporales [60]. En contraste, las *series no estacionarias* presentan propiedades estadísticas que varían con el tiempo, como una media cambiante o una varianza creciente. Este tipo de comportamiento se manifiesta mediante tendencias, estacionalidades o rupturas estructurales en los datos. Para analizarlas adecuadamente, es común aplicar transformaciones como la diferenciación para lograr la estacionariedad [60], mientras que enfoques más recientes, como los modelos de aprendizaje automático, permiten capturar directamente esta complejidad sin requerir transformaciones estrictas [61]. Cerqueira et al. [61] destacan que modelos como SARIMA para estacionalidad o diferenciación (transformación a estacionariedad) son clave, mientras que en escenarios complejos LSTM y Transformers capturan patrones no lineales.

2.3.2.2. Estacional y no estacional

Las *series estacionales* presentan patrones cíclicos que se repiten en intervalos regulares, como días, semanas o meses, lo que refleja comportamientos cíclicos. También puede observarse en horarios que aumentan durante el horario laboral y disminuyen durante las noches y fines de semana. Hyndman et al. [60] destacan que identificar correctamente la estacionalidad mejora sustancialmente la precisión del pronóstico, lo cual es clave en sistemas elásticos donde la anticipación de la demanda permite una planificación eficiente de recursos al ajustar recursos automáticamente, por ejemplo, al aplicar *autoscaling* preventivo. Por el contrario, las *series no estacionales* carecen de patrones regulares y pueden estar influenciadas por factores externos aleatorios, como errores humanos o fallos de *hardware*. En estos casos, la modelación requiere enfoques más robustos y flexibles frente a la aleatoriedad, y técnicas de detección de anomalías pueden complementar al pronóstico tradicional [61].

2.3.2.3. Univariante y Multivariante

Las *series univariantes* contienen observaciones secuenciales de una sola variable, donde las instancias temporales están vinculadas cronológicamente, permitiendo analizar la evolución de la variable a lo largo de periodos consecutivos de tiempo. Su análisis resulta más simple y directo, estas series son útiles para anticipar el comportamiento de indicadores individuales. Hyndman et al [60] explican que los modelos univariantes pueden alcanzar una precisión notable cuando la variable contiene suficiente estructura temporal, aunque su limitación principal es la falta de contexto sobre otros factores que puedan influir.

Las *series multivariantes* registran múltiples variables temporalmente correlacionadas, esto quiere decir que múltiples variables intervienen y varían con el tiempo, generando una

correlación tanto con sus datos históricos como con el resto de las variables. Esta capacidad es especialmente valiosa, ya que permite modelar interdependencias entre variables. Modelos multivariantes, como LSTM multivariado o transformers temporales, pueden capturar estas relaciones y mejorar la precisión del pronóstico. Cerqueira et al. [61] subrayan que los modelos multivariados, si bien más complejos, permiten una comprensión más profunda del sistema y una capacidad predictiva superior en escenarios reales y dinámicos.

2.3.3. Modelos clásicos de pronóstico

Los métodos tradicionales de pronóstico de series temporales han sido ampliamente utilizados en diversas disciplinas debido a su solidez estadística, interpretabilidad y eficacia en contextos donde los patrones temporales son relativamente estables y lineales. Entre estos destacan los modelos autorregresivos (AR), de media móvil (MA), el modelo ARIMA, y sus variantes estacionales (SARIMA), que se basan en principios matemáticos bien establecidos que permiten modelar y extrapolar tendencias históricas. Su fortaleza radica en la capacidad para capturar patrones de dependencia temporal, estacionalidad y tendencia con un bajo requerimiento computacional y sin necesidad de grandes volúmenes de datos.

2.3.3.1. *Auto Regressive Integrated Moving Average* (ARIMA)

El modelo ARIMA es un método estadístico ampliamente utilizado para el análisis y pronóstico de series temporales. Este modelo fue propuesto por Box y Jenkins en su enfoque clásico de modelado de series temporales [62] y es particularmente útil cuando los datos presentan comportamientos no estacionarios, es decir, cuando su media y varianza cambian a lo largo del tiempo.

ARIMA combina tres componentes fundamentales:

- El `componente autorregresivo` (AR (p)) que modela la dependencia lineal entre la observación actual y un conjunto de observaciones pasadas.
- El `componente integrado` (I (d)), que representa el número de veces que se diferencia la serie para volverla estacionaria.
- El `componente media móvil` (MA (q)), que modela la relación entre la observación actual y una combinación lineal de errores pasados (residuos).

La notación $ARIMA(p, d, q)$, indica el número de términos autorregresivos (p), de diferenciación (d) y de media móvil (q), respectivamente. Este modelo pronostica valores futuros combinando estos tres componentes para capturar diferentes patrones en los datos de la serie temporal.

El proceso metodológico basado en el enfoque Box y Jenkins [62], incluye los siguientes:

1. `Identificación de la presencia de tendencias`. Se evalúa si la serie temporal contiene una tendencia, es decir, es estacionaria. Para ello, se aplica comúnmente la prueba de *Dickey-Fuller* Aumentada (ADF). Si la serie no es estacionaria, se

procede a diferenciarla sucesivamente (d veces) hasta alcanzar la estacionariedad. Este proceso transforma una serie temporal no estacionaria en una estacionaria, requisito clave para aplicar modelos ARMA o ARIMA [60].

2. `Identificación del componente estacional de la serie` (periodicidad). Aunque el modelo ARIMA tradicional no maneja estacionalidad explícita, es relevante examinar si existe algún componente periódico en la serie. Esto puede realizarse mediante el análisis espectral, observando la presencia de picos de frecuencia que indiquen ciclos recurrentes.

3. `Identificación del modelo`. Una vez obtenida la serie estacionaria, se exploran combinaciones de los parámetros p, d y q. Por un lado, d se selecciona diferenciando la serie d veces hasta obtener una serie estacionaria (según la prueba ADF), los parámetros p y q se estiman mediante un procedimiento iterativo, basado en el Criterio de Información de Akaike (AIC), cuyo objetivo es determinar el modelo que mejor se ajusta a los datos utilizados para su entrenamiento.

4. `Estimación de parámetros`. Una vez seleccionado el modelo, se estiman los coeficientes de las componentes AR y MA (α_i y β_j) mediante métodos de ajuste, minimizando el error de predicción. Se sigue utilizando el AIC para comparar variantes del modelo y asegurar el mejor ajuste a los datos.

5. `Pronóstico`. El modelo ARIMA ajustado se utiliza para generar pronósticos futuros, extrapolando los patrones identificados en la serie temporal histórica.

El modelo ARIMA representa uno de los enfoques más robustos y utilizados en el pronóstico de series temporales en múltiples dominios, incluyendo el análisis de métricas de QoS para servicios cloud. Su capacidad para convertir series no estacionarias en estacionarias y modelar dependencias lineales lo hace especialmente útil en situaciones donde se requiere una interpretación clara del comportamiento histórico de los datos.

2.3.4. Modelos de pronóstico de aprendizaje automático

En los últimos años, los métodos de aprendizaje automático han ganado protagonismo en el pronóstico de series temporales, especialmente en contextos complejos como la monitorización de servicios cloud, donde los patrones pueden ser no lineales, con ruido y altamente dinámicos. A diferencia de los modelos estadísticos, los modelos de ML no requieren suposiciones estrictas sobre la estructura de datos, lo que les permite capturar relaciones complejas, interacciones entre múltiples variables y dependencias de largo plazo. Algoritmos como *random forest*, redes neuronales recurrentes y sus variantes más avanzadas, como LSTM y GRU, han demostrado ser especialmente eficaces en tareas de predicción de comportamiento del servicio, consumo de recursos o detección de anomalías. Además, estos modelos pueden adaptarse a grandes volúmenes de datos en tiempo real, lo que resulta particularmente valioso en entornos cloud donde la escalabilidad y precisión del pronóstico son críticas [61, 63].

2.3.4.1. *Random Forest*

Random Forest es una técnica de ML basada en el ensamblado de árboles de decisión, propuesta por Breiman [64], que combina múltiples árboles construidos sobre subconjuntos aleatorios de los datos y de las variables. Su principio clave radica en la agregación (*bagging*), donde se entrenan numerosos árboles independientes sobre diferentes particiones del conjunto de datos, y se promedian sus resultados para tareas de regresión o se vota en conjunto para clasificación. Esta estrategia reduce significativamente la varianza del modelo y mejora su capacidad de generalización, lo que la hace especialmente eficaz frente a datos ruidosos o no lineales. En el contexto del pronóstico de series temporales, especialmente en problemas de regresión, Random Forest ha mostrado un desempeño competitivo gracias a su robustez ante valores atípicos, su capacidad para manejar relaciones no lineales y su facilidad para capturar interacciones entre variables. Resultados empíricos han demostrado su eficacia en tareas de predicción multivariada, superando en varios casos a modelos lineales tradicionales, particularmente cuando las relaciones temporales son complejas y no estacionarias [64, 61].

2.3.4.2. *Support Vector Regression* (SVR)

SVR es una técnica de aprendizaje supervisado derivada de las máquinas de vectores de soporte (SVM), diseñada específicamente para problemas de regresión. SVR adapta los principios fundamentales de las SVM, como la maximización del margen y el uso de funciones núcleo (*kernels*), para modelar relaciones no lineales entre variables en contextos donde la predicción precisa de valores continuos es esencial [65]. A diferencia de las SVM, cuyo objetivo es encontrar un hiperplano que separe de manera óptima las clases, SVR busca una función de predicción que se desvíe como máximo ϵ (epsilon) de los valores reales, minimizando al mismo tiempo la complejidad del modelo. Esta formulación permite al modelo tolerar pequeñas desviaciones y centrarse en capturar la estructura general de los datos, lo cual es especialmente útil en presencia de ruido y valores atípicos [66].

La flexibilidad de SVR proviene del uso de funciones núcleo que transforman los datos de entrada a espacios de características de alta dimensión, donde las relaciones no lineales se vuelven linealmente separables o ajustables. Además, al enfocarse en un subconjunto de puntos críticos—los llamados vectores de soporte—SVR logra una representación esparsa del modelo, lo que mejora la generalización y la eficiencia computacional, incluso en contextos de gran escala [67]. Estas propiedades han hecho del SVR una herramienta robusta para tareas de regresión complejas, como el pronóstico de series temporales multivariadas en dominios como el rendimiento de servicios cloud, donde las relaciones entre métricas pueden ser altamente no lineales y ruidosas.

2.3.5. Modelos de pronóstico de aprendizaje profundo

En el contexto del pronóstico de series temporales, los métodos basados en aprendizaje profundo (DL) han demostrado ser particularmente eficaces para modelar relaciones complejas y no lineales entre variables a lo largo del tiempo. El aprendizaje profundo,

como subcampo del aprendizaje automático, se basa en redes neuronales con múltiples capas ocultas (*deep neural networks*, DNN), cuya arquitectura permite extraer representaciones jerárquicas de los datos y capturar patrones temporales complejos que suelen escapar a los modelos tradicionales. En particular, las redes neuronales recurrentes y sus variantes más avanzadas, como las redes de memoria a largo y corto plazo (LSTM) y las unidades recurrentes con compuertas (GRU), han sido ampliamente utilizadas en tareas de pronóstico, debido a su capacidad para modelar dependencias temporales en secuencias de datos. Estas arquitecturas mantienen un estado oculto que permite conservar información de pasos anteriores, facilitando así la predicción de valores futuros incluso en presencia de secuencias largas y dinámicas [22].

2.3.5.1. Redes de memoria a largo y corto plazo (LSTM)

Las redes de memoria a largo y corto plazo (*Long Short-Term Memory*) constituyen una arquitectura especializada de redes neuronales recurrentes, diseñada específicamente para capturar dependencias a largo plazo en secuencias de datos. Su principal ventaja frente a las RNN tradicionales radica en su capacidad para mitigar el problema del desvanecimiento del gradiente, lo que les permite aprender patrones en secuencias temporales extensas sin perder información relevante en el proceso. Esta arquitectura ha demostrado ser eficaz en múltiples dominios como el pronóstico de series temporales, el modelado del lenguaje, el reconocimiento de audio y el análisis de secuencias en general.

Las LSTM incorporan unidades recurrentes avanzadas llamadas *celdas de memoria*, que actúan como mecanismos internos de almacenamiento y transporte de información a través del tiempo, permitiendo la actualización y retención del estado interno durante largos intervalos. Además, integra mecanismos de activación especializados que facilitan el procesamiento y almacenamiento de información en secuencias extendidas, es decir, durante un largo periodo [16].

En la Figura 2.6 representa la estructura de la red LSTM, donde C_{t-1} es la celda de estado, que transporta información de una unidad a otra. Un componente clave de estas celdas son sus tres compuertas de control: una compuerta de entrada, una una compuerta de olvido, y compuerta de salida. h_t es el estado oculto, que representa la memoria a corto plazo de la neurona, y C_t representa la memoria a largo plazo.

La *compuerta de entrada (input gate)* (1), determina qué información debe actualizar la celda de estado. Para ello, usa la función de activación tangente hiperbólica (tanh) en los datos del estado anterior h_{t-1} y los datos de entrada x_t que serán candidatos para ingresar a la celda de estado C_{t-1}. La *compuerta de olvido (forget gate)* (2), decide qué información del estado previo debe descartarse o retenerse. Es decir, utiliza el estado oculto anterior h_{t-1} junto con los datos de entrada x_t para eliminar los datos irrelevantes en la celda de estado. La *compuerta de salida (output gate)* (3), selecciona la parte del estado de la celda que se propagará como salida o como entrada al siguiente paso de tiempo. Es decir, se filtran entrada actual x_t y el estado actual c_t y aplica la función de activación tanh, para generar un vector de activación que representa la salida h_t del LSTM.

Esta estructura especializada permite a las LSTM aprender relaciones temporales com-

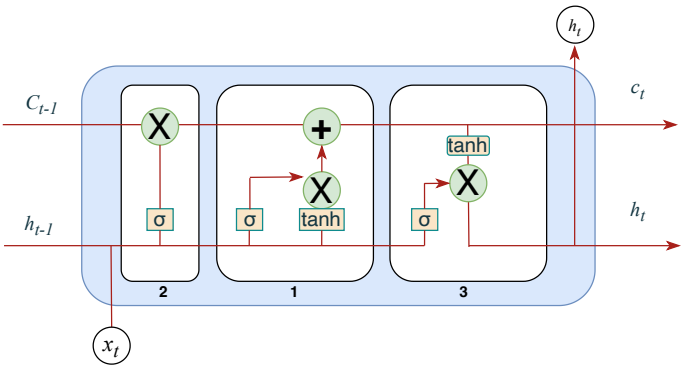
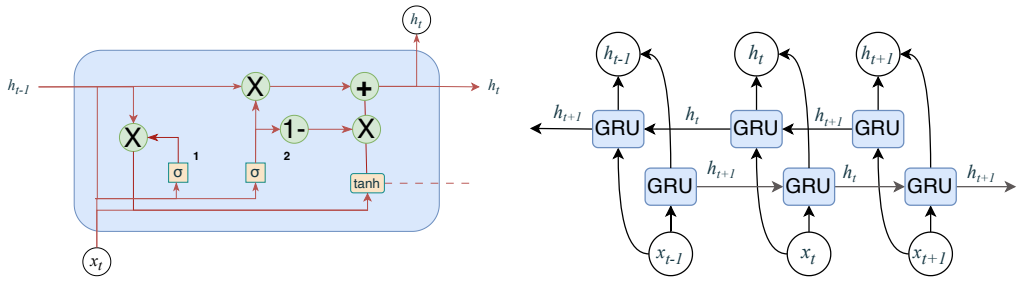


Figura 2.6: Estructura LSTM

plejas, ajustándose de manera dinámica a diferentes horizontes temporales en los datos secuenciales [16].

2.3.5.2. Unidad recurrente con compuerta bidireccional (BI-GRU)

Las redes de Unidad recurrente con compuerta bidireccional *(Bidirectional Gated Recurrent Units)* constituyen una variante avanzada de las unidades recurrentes con compuertas *(Gated Recurrent Units, GRU)*, diseñadas para capturar dependencias secuenciales tanto del pasado como del futuro dentro de una secuencia de datos. A diferencia de las LSTM, las GRU simplifican la arquitectura al prescindir de una celda de memoria independiente y actualizar directamente los estados ocultos. Esta simplificación reduce el costo computacional con un entrenamiento eficiente, al mismo tiempo, mantiene un rendimiento competitivo en tareas como el procesamiento del lenguaje natural y la predicción de series temporales [68].



(a) celda GRU (b) BI-GRU

Figura 2.7: Estructura BI-GRU.

En la Figura 2.7 (a) se presenta la estructura de la celda GRU, la cual utiliza dos compuertas fundamentales: la *compuerta de reinicio (reset gate)* (1), que controla cuánta información previa se debe olvidar, y la *compuerta de actualización (update gate)* (2), que controla la cantidad de información nueva que debe conservarse. Es decir, calcula el estado

oculto h_t en el momento t basándose en la salida de la compuerta de actualización (2), la compuerta de reinicio (1) y la entrada actual x_t.

La Figura 2.7 (b) presenta la estructura BI-GRU que amplía esta capacidad al incorporar dos capas GRU que procesan secuencias de entrada en direcciones opuestas: una hacia adelante en orden cronológico y otra hacia atrás en sentido inverso. Al combinar ambas direcciones, el modelo obtiene una representación enriquecida que captura simultáneamente patrones y dependencias contextuales en ambos sentidos temporales. Esta bidireccionalidad resulta particularmente útil en tareas que requieren una comprensión integral del contexto, como la predicción de series temporales en entornos complejos (p. ej., métricas de QoS de los servicios cloud), el reconocimiento de voz, el análisis de sentimientos o la traducción automática [69, 68].

2.3.6. Modelos utilizados en esta tesis

Considerando el contexto descrito en las subsecciones anteriores, en esta tesis se emplean los siguientes modelos:

- **ARIMA**, como referencia estadística tradicional y ampliamente aceptada en análisis de series temporales;
- **LSTM**, por su eficacia demostrada en la captura de dependencias no lineales y de largo plazo;
- **BI-GRU**, por su eficiencia computacional y su capacidad bidireccional para modelar patrones secuenciales complejos.

ARIMA se utiliza como modelo estadístico tradicional de referencia por varias razones. En primer lugar, constituye uno de los enfoques más establecidos en el análisis de series temporales y ofrece una excelente capacidad para modelar dependencias lineales, estacionalidades simples y tendencias suaves, lo cual permite contextualizar el desempeño de los modelos de aprendizaje profundo. En segundo lugar, su interpretabilidad facilita la comprensión de patrones fundamentales presentes en métricas de QoS, como ciclos de carga o estacionalidad diaria en los tiempos de respuesta. Finalmente, la literatura recomienda incluir modelos estadísticos como línea base para evaluar rigurosamente métodos más complejos, especialmente cuando se trabaja con series univariantes y horizontes de predicción reducidos, como en este estudio.

Por otra parte, la arquitectura **LSTM** se selecciona por su capacidad contrastada para capturar dependencias de largo plazo en series temporales no lineales. Las métricas cloud suelen presentar patrones de acumulación (p.ej., aumentos progresivos de latencia debido a la saturación del sistema) o patrones recurrentes influenciados por la carga del usuario, que requieren mecanismos de memoria ampliada. El uso de LSTM permite modelar este tipo de comportamientos de forma robusta. Asimismo, LSTM es una de las arquitecturas más estudiadas en el contexto de QoS y *cloud computing* [17, 70, 71], lo que facilita la comparación con investigaciones previas y dota de mayor validez externa a los resultados.

Por último, la selección de **BI-GRU** obedece a que esta arquitectura logra un equilibrio óptimo entre precisión predictiva y eficiencia de entrenamiento. Las GRU reducen el número de parámetros frente a las LSTM y aceleran la convergencia sin afectar significativamente la capacidad de memoria, lo cual es beneficioso en escenarios donde el volumen de datos no es masivo o donde la inferencia debe ser rápida. La variante *bidireccional* de la arquitectura procesa y aprende patrones en dos direcciones, tanto previos como posteriores al instante de predicción durante el entrenamiento, mejorando la capacidad de capturar tendencias locales o transiciones abruptas típicas de métricas como el *throughput* o la disponibilidad. Estudios recientes han demostrado que las arquitecturas GRU ofrecen un rendimiento competitivo e incluso superior frente a arquitecturas más complejas en series temporales ruidosas o irregulares [72], como es común en entornos cloud.

Todos estos modelos se aplican a métricas de QoS extraídas de un servicio cloud real, lo que permite capturar comportamientos auténticos, incluyendo ruido, variaciones abruptas y estacionalidad.

2.3.7. Justificación de la no inclusión de alternativas avanzadas

Aunque existen modelos avanzados de última generación (como *Prophet* un modelo aditivo enfocado en la interpretabilidad y la descomposición de series temporales) [73], *Transformers* especializados para series temporales [74], N-BEATS (*Neural Basis Expansion Analysis for Time Series Forecasting*) [75], TCN (*Temporal Convolutional Networks*) [76] o DeepAR (*Probabilistic Forecasting with Autoregressive Recurrent Networks*) [77]) su uso no se considera prioritario en esta tesis por varias razones:

1. **Objetivo metodológico de comparación controlada**.

   El propósito del experimento controlado descrito en el Capítulo 8 es comparar modelos representativos de tres familias: estadísticos clásicos, redes recurrentes profundas y variantes optimizadas de estas. Incluir arquitecturas más complejas introduciría diferencias estructurales difíciles de comparar en condiciones justas bajo un enfoque univariante y *one-step*.

2. **Naturaleza de los datos y del horizonte de predicción.** Los modelos como N-BEATS o *Transformers* tienden a mostrar ventajas cuando existen horizontes multistep largos, series multivariantes o grandes volúmenes de datos. En problemas univariantes y de un paso, su complejidad adicional no garantiza beneficios significativos.

3. **Sobrecoste computacional.** Las arquitecturas más avanzadas requieren considerable potencia de cómputo y tiempos de entrenamiento más prolongados, lo cual no resulta alineado con el objetivo de evaluar métodos aplicables operativamente en entornos industriales.

4. **Trazabilidad y reproducibilidad.** ARIMA, LSTM y GRU son modelos ampliamente documentados y reproducibles, lo cual facilita la validación y transferencia de resultados.

En conjunto, la selección de ARIMA, LSTM y BI-GRU permite establecer una comparación equilibrada entre enfoques tradicionales y modernos, manteniendo la coherencia metodológica, la eficiencia y la aplicabilidad industrial del estudio. Además, estos modelos ofrecen una base sólida para analizar la capacidad de las distintas familias de técnicas de pronóstico para anticipar comportamientos de QoS en servicios cloud.

2.4. Relación entre servicios cloud, modelos de calidad y pronóstico

La computación en la nube se caracteriza por su dinamismo, su naturaleza distribuida y la continua fluctuación de los recursos y cargas de trabajo. Estas propiedades, que aportan flexibilidad y eficiencia operativa, introducen a la vez una complejidad significativa en la gestión y evaluación de la calidad del servicio. La elasticidad, la multitenencia, el escalado automático y la heterogeneidad tecnológica hacen que muchos de los atributos de calidad —como la latencia, la disponibilidad, la capacidad o el rendimiento— se comporten de forma altamente variable en el tiempo. Esta variabilidad no solo dificulta la realización de evaluaciones estáticas de la calidad, sino que también exige una monitorización continua y mecanismos que permitan anticipar comportamientos futuros.

En este contexto, los **modelos de calidad** constituyen un elemento fundamental para organizar, formalizar y descomponer el concepto abstracto de calidad en características, subcaracterísticas y métricas específicas que puedan ser evaluadas de forma sistemática. Los modelos derivados de estándares como ISO/IEC 25010 [1] aportan una estructura conceptual reconocida internacionalmente, pero requieren ser adaptados al dominio cloud para incorporar atributos propios de este entorno, tales como elasticidad, eficiencia en la provisión dinámica de recursos, aislamiento en entornos multitenencia o impacto de la virtualización sobre la calidad percibida. La definición de un modelo de calidad específico permite, por tanto, representar las peculiaridades del dominio, ofrecer un marco común de comunicación entre distintos *stakeholders* y establecer las bases necesarias para la definición de mecanismos de evaluación.

Sin embargo, aunque los modelos de calidad aportan un marco conceptual sólido, su aplicación en contextos reales se ve limitada si no se consideran las propiedades temporales de las métricas asociadas. Muchas métricas de QoS en servicios cloud no presentan un comportamiento estático ni lineal. La latencia puede incrementarse de forma abrupta ante picos de carga; la disponibilidad puede fluctuar en función de fallos intermitentes de red; y el *throughput* puede depender tanto del uso concurrente como del tipo de infraestructura subyacente. En consecuencia, una evaluación basada únicamente en valores puntuales o promedios oculta la riqueza y complejidad de estas señales temporales, dificultando la detección temprana de anomalías y la capacidad para anticipar degradaciones.

Es aquí donde los **métodos de pronóstico basados en aprendizaje automático** adquieren un papel esencial. El tratamiento de las métricas de calidad como series temporales permite capturar su evolución en el tiempo, modelar las relaciones entre observaciones pasadas y futuras, y descubrir patrones que serían difíciles de identificar mediante inspec-

ción manual o técnicas descriptivas tradicionales. Los enfoques de aprendizaje profundo, en particular las redes LSTM, GRU y sus variantes bidireccionales, ofrecen una capacidad notable para modelar dependencias no lineales, efectos acumulativos, estacionalidad y patrones complejos asociados a la dinámica del servicio. De este modo, los modelos de pronóstico complementan la visión estructural proporcionada por los modelos de calidad y permiten abordar no solo la evaluación retrospectiva del servicio, sino también su comportamiento futuro.

La integración de ambos enfoques (modelos de calidad y técnicas de pronóstico) constituye un marco conceptual coherente y necesario para comprender el comportamiento de los servicios cloud. Por un lado, los modelos de calidad proporcionan los elementos conceptuales, estructurales y normativos para identificar qué atributos son relevantes y cómo deben ser medidos. Por otro lado, los modelos de pronóstico permiten anticipar la evolución de dichos atributos y proporcionar información que apoya la toma de decisiones proactiva: detección temprana de anomalías, planificación de capacidad, adaptación de recursos o cumplimiento de acuerdos de nivel de servicio (SLA). En conjunto, ambos pilares permiten ofrecer una visión completa de la calidad de los servicios cloud desde una perspectiva tanto estática como dinámica, garantizando un análisis profundo y robusto de un entorno caracterizado por su variabilidad continua.

Esta síntesis establece, por tanto, la base conceptual para las contribuciones desarrolladas en esta tesis. Las siguientes secciones profundizarán en la revisión de la literatura, la construcción de la taxonomía de métricas, la definición del modelo de calidad específico para servicios cloud y el proceso de modelado para el pronóstico de atributos de QoS, integrando de manera progresiva los elementos presentados en este capítulo.

2.5. Conclusiones

El presente capítulo ha establecido los fundamentos conceptuales necesarios para abordar la investigación desarrollada en esta tesis. En primer lugar, se han descrito las características esenciales de los servicios cloud y los distintos modelos de despliegue y provisión, destacando aquellos elementos estructurales (como la elasticidad, la multitenencia y la heterogeneidad de recursos) que influyen directamente en su comportamiento en tiempo de ejecución y, por tanto, en su calidad.

Posteriormente, se ha revisado el marco normativo y conceptual asociado a la calidad del software, poniendo de relieve la necesidad de adaptar los modelos genéricos, como ISO/IEC 25010 [1], a los requisitos específicos del dominio cloud mediante atributos y métricas cuantificables.

Asimismo, se han introducido los fundamentos del pronóstico de series temporales y las principales técnicas estadísticas y de aprendizaje automático capaces de modelar la evolución de las métricas de QoS. Se ha justificado la selección de los modelos ARIMA, LSTM y BI-GRU como enfoques representativos para comparar capacidades predictivas en escenarios con distintas complejidades temporales.

Finalmente, el capítulo ha articulado la relación entre estos tres pilares (servicios cloud, calidad y pronóstico), estableciendo el marco conceptual que sustenta la necesidad de un

modelo de calidad específico y de técnicas predictivas que permitan anticipar desviaciones. Sobre estos fundamentos se apoya el Capítulo 3, que examina en profundidad el estado del arte y permite identificar las limitaciones existentes que motivan las contribuciones de esta tesis.

Capítulo 3

Estado del Arte

En este capítulo se analizan los estudios previos relacionados con las taxonomías de métricas de QoS, los modelos de calidad y los modelos de pronóstico de QoS en el dominio de los servicios cloud.

La Sección 3.1 revisa los trabajos de investigación que proponen taxonomías de métricas de calidad en el ámbito de los servicios cloud, así como los estudios secundarios relacionados con la evaluación y monitorización de la QoS. La sección 3.2 examina los estudios que presentan modelos de calidad o enfoques de evaluación de la calidad aplicados a los servicios cloud. La sección 3.3 analiza los estudios que utilizan modelos de pronóstico para la monitorización de la QoS en servicios cloud, con especial énfasis en aquellos que utilizan técnicas de *deep learning*. Finalmente, la sección 3.4 sintetiza las principales aportaciones identificadas en el estado del arte y da inicio a la correspondiente discusión crítica.

3.1. Métricas de calidad y taxonomías

En esta sección se analizan los trabajos de investigación que han recopilado métricas de QoS o propuesto taxonomías de métricas de calidad para los servicios cloud. Inicialmente, se realizaron búsquedas en las bibliotecas digitales *IEEE Xplore*, *ACM Digital Library*, *SpringerLink* y *Science Direct*. Los metadatos utilizados para la búsqueda incluyeron el título, el resumen y las palabras clave. La búsqueda se llevó a cabo utilizando la siguiente cadena:

> *(metric OR measure) AND quality AND cloud AND (taxonomy OR ontology OR classification)*

Se identificaron algunos estudios cuya atención se centraba en enfoques específicos o en determinadas características de calidad. Por ejemplo, Li et al. [78] propusieron una taxonomía para la evaluación del rendimiento de servicios cloud comerciales, estructurada en dos dimensiones: característica de rendimiento y función experimental. La primera se descompone en 4 propiedades físicas—procesamiento, memoria, almacenamiento y red—y siete elementos de capacidad (p. ej., disponibilidad, confiabilidad). La segunda dimensión considera cinco escenarios ambientales (como el uso de un único proveedor frente a múltiples proveedores de la nube) y quince escenarios operativos (como la repetición de experimentos bajo interferencia humana). Cada escenario se define como una unidad atómica sobre la cual se construye un experimento completo de evaluación. Si bien esta taxonomía resulta útil para analizar prácticas de evaluación existentes y para diseñar nuevos experimentos, su alcance es limitado, ya que solo contempla siete métricas—una por cada elemento de capacidad—enfocadas exclusivamente en la evaluación del rendimiento (velocidad, tiempo de actividad, latencia, tasa de fallas, rendimiento real, escalabilidad y variabilidad).

Habib et al. [79] propusieron una taxonomía de propiedades de confianza para evaluar la calidad y seguridad de los proveedores de servicios cloud. Basaron su enfoque en el Cuestionario de Iniciativa de Evaluación de Consenso (*Consensus Assessments Initiative Questionnaire*, CAIQ) desarrollado por la Alianza de Seguridad en la Nube (*Cloud Security Alliance*,CSA), con el fin de verificar las propiedades de confianza derivadas de dichos controles y evaluar las declaraciones de los proveedores cloud. Para ello, los autores diseñaron una taxonomía que permite mapear y clasificar los controles del CAIQ en propiedades de confianza, considerando tanto los tipos de control como las autoridades de validación. Además, integraron esta taxonomía en un modelo de decisión que combina la verificación de las propiedades de confianza con los requisitos del consumidor, permitiendo determinar el nivel de confiabilidad de los proveedores. En definitiva, el estudio se orienta principalmente a la evaluación de la confiabilidad de los proveedores cloud, más que a la evaluación directa de la calidad de los servicios que estos ofrecen.

Herbst et al. [80] propusieron una taxonomía de métricas para entornos cloud centrada en cuatro propiedades del sistema: la elasticidad del servicio, el aislamiento y la variabilidad del rendimiento entre inquilinos, la disponibilidad del servicio y el riesgo operativo

asociado a la ejecución de sistemas de producción cloud. La propuesta organiza las métricas en cuatro niveles de abstracción—métricas de rendimiento tradicionales, métricas de infraestructura, métricas de políticas y métricas orientadas a decisiones gerenciales—con el propósito de facilitar la comparación entre diferentes ofertas y tecnologías cloud, así como de establecer un lenguaje común entre proveedores, clientes y usuarios finales. No obstante, aunque los autores presentan una estructura jerárquica coherente para la medición y evaluación, su alcance se limita fundamentalmente a propiedades de rendimiento.

La elasticidad es un factor crítico en los servicios cloud y diversos estudios se han centrado en su medición. Muñoz-Escoí et al. [81] identificaron los requisitos y desafíos asociados a la gestión de recursos elásticos en un proveedor de PaaS, además de proponer posibles soluciones. Por su parte, Coutinho et al. [82] definieron métricas y herramientas orientadas a medir la elasticidad. Ambos estudios analizan esta propiedad de forma aislada. Sin embargo, este enfoque resulta limitado si se considera el impacto que la elasticidad tiene sobre otros atributos de calidad del servicio cloud, como la escalabilidad y la eficiencia [83], quienes, además, proponen una taxonomía que caracteriza la elasticidad como una propiedad dinámica y medible, esencial para optimizar el uso de recursos y mantener la calidad del servicio en tiempo real.

También se han publicado otras encuestas cuyo objetivo ha sido analizar y clasificar las métricas empleadas en la evaluación de servicios cloud. Jelassi et al. [84] realizaron una revisión tomando el rendimiento como principal referencia de análisis y presentando varios parámetros y métodos de QoS aplicables a su medición. Los autores identificaron nueve propiedades clave—como rendimiento,latencia, disponibilidad, fiabilidad, costo y reputación—junto con sus correspondientes enfoques de evaluación cuantitativos y cualitativos, y agruparon los mecanismos existentes en cinco categorías: gestión de recursos, control de admisión, gestión de colas de espera, planificación y monitorización. Aunque la encuesta ofrece una visión amplia y útil de los enfoques actuales orientados a la medición del rendimiento, se trata de un estudio exploratorio de carácter general, por lo que se requiere una revisión más sistemática y comparativa que abarque las métricas relacionadas no solo con el rendimiento, sino también con otras características de la calidad de servicio.

Bardsiri y Hashemi [85] centraron su estudio en analizar la clasificación y aplicación de las métricas de QoS para los principales tipos de servicios (SaaS, PaaS e IaaS). Las métricas se clasificaron en cuatro grupos—rendimiento, economía, seguridad y características generales—, proporcionando ejemplos específicos para cada categoría, como tiempo de respuesta, costo total, autenticación y disponibilidad. Su enfoque permitió obtener un catálogo de métricas desde la perspectiva de los proveedores de servicios, lo que facilita la comparación objetiva entre distintas ofertas. No obstante, el estudio se centró principalmente en enumerar las métricas, sin detallar metodologías precisas para su medición, lo que limita su aplicabilidad práctica y la reproducibilidad de los resultados.

3.1.1. Estudios secundarios existentes

De igual forma, en los últimos años se han propuesto diversos estudios secundarios—incluyendo revisiones sistemáticas y mapeos sistemáticos—enfocados en identificar,

clasificar y analizar las métricas relacionadas con la calidad de los servicios cloud.

Destacan los aportes de Li et al. [86], quienes realizaron una revisión sistemática de la literatura centrada en la identificación de métricas de calidad asociadas a atributos como el rendimiento, la economía y la seguridad de los servicios cloud. Como resultado, establecieron un catálogo de 97 métricas que posteriormente sirvió como base para la definición de un marco orientado a la selección de servicios comerciales de IaaS y PaaS. En un trabajo posterior, Li et al. [87] ampliaron el alcance de su revisión incorporando el uso y el contexto de aplicación de las métricas, al considerar procedimientos de evaluación, propiedades, puntos de referencia y entornos experimentales. Su análisis reveló una notable dispersión y falta de uniformidad en la aplicación de las métricas, especialmente en escenarios comerciales cloud.

Abdelmaboud et al. [88] llevaron a cabo un mapeo sistemático con el fin de examinar los enfoques y modelos utilizados para evaluar la calidad de los servicios cloud, analizando 67 estudios primarios publicados entre 2008 y 2012. Sus resultados mostraron que los enfoques de investigación se centraban principalmente en IaaS (48 %) y SaaS (36 %), mientras que un 10 % se orientaba a proveedores y un 3 % a consumidores. Asimismo, identificaron que predominaban los estudios de validación (64 %), orientados al desarrollo de métodos (48 %) y modelos (32 %). Los autores concluyeron que aún persisten carencias significativas en cuanto a métricas, herramientas y evaluaciones empíricas que permitan garantizar niveles de servicio acordes con las expectativas de los usuarios y los acuerdos de nivel de servicio (SLA).

Por su parte, Lehrig et al. [89] profundizaron en su revisión sistemática abarcando definiciones y métricas relacionadas con la escalabilidad, elasticidad y eficiencia en entornos cloud. Su análisis, centrado en estudios indexados en Google Scholar, permitió identificar conceptos y métricas compartidas para evaluar estos atributos; sin embargo, los autores señalan que persiste una carencia significativa de validaciones empíricas.

En el contexto de FaaS, Scheuner y Leitner [90] realizaron una revisión multivocal que combinó literatura académica y literatura gris. FaaS ofrece un modelo de servicio cloud completamente nuevo que habilita arquitecturas *serveless* basadas en microservicios, aportando ventajas como el aprovisionamiento dinámico de recursos y el escalado automático. En su estudio, los autores examinaron las tendencias actuales, las configuraciones de plataforma y las características de rendimiento, centrándose en cuatro atributos: sobrecarga de la plataforma, concurrencia de la carga de trabajo, duración de las instancias e inspección de la infraestructura. Sin embargo, no se identificaron métricas cuantificables, lo que revela una limitación en la formalización de mecanismos de evaluación de la calidad para modelos *serverless*.

Kashani et al. [91] presentaron una revisión sistemática de los enfoques orientados a la QoS en entornos de computación en la niebla (*fog computing*), considerada una extensión natural de la computación en la nube que introduce nuevos desafíos de evaluación debido a la descentralización de los servicios y recursos hacia el borde de la red, más cerca de los dispositivos finales. El estudio clasificó once factores de QoS—entre ellos rendimiento, plazo, tiempo de respuesta, utilización de recursos, costo, tiempo de ejecución, consumo energético, confiabilidad, disponibilidad, escalabilidad y seguridad—organizados en tres

niveles de gestión: servicio/recurso, comunicación y aplicación. Los autores señalaron que factores como el tiempo de respuesta, el costo y la utilización de recursos son los más utilizados, mientras que la disponibilidad y la escalabilidad son los menos considerados. Asimismo, el trabajo propone una taxonomía que agrupa diversas técnicas y mecanismos de QoS (como asignación de recursos, planificación de tareas y gestión de red), evidenciando la ausencia de métricas estandarizadas y la cobertura parcial de las características de calidad.

En la Tabla 3.1 se presenta un resumen comparativo de los estudios secundarios analizados.

3.2. Modelos de calidad

La necesidad de contar con modelos de calidad específicos y bien definidos se ha vuelto fundamental para garantizar una evaluación sistemática, objetiva y contextualizada de los servicios cloud. Si bien los modelos de calidad genéricos, como ISO/IEC 25010, han proporcionado un marco de referencia inicial para establecer criterios de evaluación, diversos estudios han señalado sus limitaciones para reflejar las particularidades de los servicios cloud, especialmente en lo referente a su naturaleza dinámica y distribuida. Estos modelos adquieren especial relevancia en contextos donde las condiciones del entorno cambian continuamente y los requisitos no funcionales deben garantizarse de forma adaptativa, como ocurre en arquitecturas basadas en microservicios.

Asimismo, han surgido estudios centrados en los modelos de calidad en tiempo de ejecución (*runtime quality models*), los cuales permiten monitorizar, evaluar y adaptar dinámicamente las propiedades de calidad de los servicios durante su operación.

3.2.1. Modelos de calidad para servicios cloud

La computación en la nube ha transformado la manera en que se diseñan, implementan y consumen los servicios cloud, introduciendo nuevas oportunidades, pero también desafíos en términos de la evaluación de la calidad. Los servicios cloud operan en entornos altamente dinámicos, elásticos y distribuidos, lo que ha motivado la necesidad de desarrollar modelos de calidad específicos que respondan a estas particularidades. Se han propuesto diversas iniciativas para evaluar la calidad de los servicios cloud, abarcando los distintos tipos de servicio: Software como Servicio (*Software as a Service*, SaaS) [8, 9], Plataforma como Servicio (*Platform as a Service*, PaaS) [92], Infraestructura como Servicio (*Infrastructure as a Service*, IaaS) [11] o modelos aplicables a todos ellos [11, 93]. Estos modelos de calidad buscan descomponer las características clave en atributos evaluables mediante métricas cuantificables. Sin embargo, las propuestas actuales presentan limitaciones importantes, tales como cobertura parcial de atributos específicos del dominio cloud, falta de operacionalización de métricas, escasa validación empírica, enfoque restringido al punto de vista del proveedor o del consumidor y limitada aplicabilidad en escenarios industriales reales.

Tabla 3.1: Resumen comparativo de los estudios secundarios sobre métricas de calidad y taxonomías.

Opciones	Li et al. [86]	Li et al. [87]	Abdelmaboud et al. [88]	Lehrig et al. [89]	Scheuner y Leitner [90]	Kashani et al. [91]
Período de tiempo	2006−2011	2006−2011	01-2008−12-2012	04-2005−2014	2016−2019	2013−2018
Preguntas de investigación	¿Qué métricas se han utilizado para la evaluación de los servicios cloud comerciales?	¿Qué métricas se han utilizado para la evaluación de los servicios cloud comerciales?	¿Qué temas relacionados con QoS en la computación en la nube se han investigado y en qué medida?	¿Qué métricas se pueden utilizar para comparar la escalabilidad, la eficiencia y la elasticidad de los diferentes servicios cloud, y cómo se miden y utilizan?	¿Qué características generales de rendimiento (p. ej., plataforma) se evalúan comúnmente?	¿Qué enfoques de gestión se aplican en *fog computing* (FC)?¿Cuáles son los factores de QoS que se aplican generalmente a los enfoques de gestión en FC? ¿Qué algoritmos y herramientas admiten los enfoques de QoS en FC?¿Qué tipos de evaluaciones se aplican para medir los enfoques de gestión a la QoS en FC? ¿Cuáles son los retos, desafíos y opciones de investigación futura en los enfoques de gestión de la QoS para FC?
Buscadores	Bases de datos de publicaciones digitales populares (No especificado)	ACM, Google Scholar, IEEE Xplore, Science Direct, Springer Link	ACM, Google Scholar, IEEE Xplore, Science Direct, Scopus, Springer Link	Google Scholar	ACM, Google Scholar, Google Search, IEEE Explore, Hacker News Algolia Search, ISI Web of Science, Medium Search, Reddit Search, Science Direct Springer Link, Scopus, Twitter Search, Wiley	ACM, IEEE Xplore, Sage, Science Direct, Springer Link, Taylor & Francis, Wiley

Tabla 3.1 (continúa)

Opciones	Li et al. [86]	Li et al. [87]	Abdelmaboud et al. [88]	Lehrig et al. [89]	Scheuner y Leitner [90]	Kashani et al. [91]
Cadena de búsqueda	ND	(“cloud computing” OR “cloud platform” OR “cloud provider” OR “cloud service” OR “cloud offering”) AND (evaluation OR evaluating OR evaluate OR evaluated OR experiment OR benchmark OR metric OR simulation) AND (<Cloud provider’s name>OR ...)	(QoS OR “quality of service” OR SLA OR “service level agreement” OR “quality”) AND (“cloud computing” OR “cloud services”)	Phrase Def.: (scalability OR elasticity OR efficiency) AND “cloud computing” Phrase Met.: metric AND (scalability OR elasticity OR efficiency) AND “cloud computing”	(serverless OR FaaS) AND (performance OR benchmark) AND experiment AND lambda. (serverless OR FaaS) AND (performance OR benchmark). serverless. serverless benchmark.	Fog AND (“Quality of Service” OR QoS OR time OR cost OR delay OR latency OR energy OR availability OR reliability OR throughput)
# Documentos recuperados	ND	4017	515	418	1619 (956 académicos y 663 grises)	202
# Estudios primarios	46	82	67 (solo 6 relacionados con métricas)	20	112 (51 académicos y 61 grises)	41
# Métricas	97	80	ND	ND	ND	11

ND= No disponible

A continuación, se presenta un análisis detallado de los modelos de calidad más relevantes propuestos para los servicios cloud y sus respectivas métricas, señalando sus contribuciones, brechas y oportunidades de mejora.

En cuanto a los modelos aplicados a SaaS, Lee et al. [8] propusieron un modelo de calidad que extiende el estándar ISO/IEC 9126 [56], incorporando atributos como reutilización, eficiencia, confiabilidad, escalabilidad y disponibilidad. Este modelo resulta útil tanto para proveedores como para consumidores, al permitir predecir el retorno de la inversión (ROI) del servicio en función del incremento de clientes. No obstante, su alcance es limitado, al omitir atributos clave como seguridad y compatibilidad, y carece de ejemplos aplicados a entornos reales.

Wen y Dong [9] propusieron un modelo de calidad para SaaS basado en tres roles y tres factores. Los tres roles definidos son: la plataforma SaaS, encargada de administrar la infraestructura cloud y los servicios SaaS; el desarrollador de aplicaciones, responsable de crear e implementar servicios en la plataforma SaaS; y el cliente del servicio, que utiliza los servicios cloud. Mientras los tres factores considerados son: la seguridad, que abarca la seguridad del cliente, de las aplicaciones, de la red y de los datos; la calidad del servicio (QoS), que incluye la calidad de la plataforma, la calidad de la aplicación y la calidad de la experiencia (*Quality of Experience*, QoE); la calidad del software, que se refiere a la calidad de uso y la calidad del producto según la ISO/IEC 25010 [1], permitiendo abordar con mayor detalle la estructura del entorno SaaS. Los autores clasifican la evaluación de un servicio SaaS según sus requisitos y responsabilidades en cuatro niveles: básico, estándar, optimizado e integrado. Esto permite que los clientes evalúen la madurez y la calidad de los proveedores de servicios, seleccionando el más adecuado en función de sus necesidades. Sin embargo, la propuesta presenta limitaciones importantes: carece de validación empírica y no incluye un mecanismo concreto de medición, lo que restringe su aplicabilidad práctica.

Otro estudio relevante es el modelo CLOUDQUAL, propuesto por Zheng et al. [10] para SaaS, que incorporó seis características relevantes de calidad: usabilidad, disponibilidad, confiabilidad, capacidad de respuesta, seguridad y elasticidad. Además, definió un conjunto de cinco métricas objetivas y una métrica subjetiva para evaluar la usabilidad, aplicadas a estudios de casos en nubes reales como Amazon S3, Azure Blob y Aliyun OSS,considerando el punto de vista del usuario final. Aunque este modelo se acerca a una evaluación práctica, presenta limitaciones, como la utilización de una única métrica por característica y la ausencia de métricas para aspectos críticos, como la eficiencia del desempeño y la compatibilidad.

Por otro lado, Lichtenthaler y Wirtz [94] adaptaron el metamodelo Quamoco—originalmente propuesto por Wagner et al.[21]—al contexto de aplicaciones cloud nativas, integrando prácticas DevOps y seis características de calidad de la ISO/IEC 25010 [1]: rendimiento, fiabilidad, seguridad, mantenibilidad, portabilidad y compatibilidad. El modelo resultante incluye 88 métricas asociadas a 53 atributos de calidad, considerando tanto la perspectiva del desarrollador como la de la arquitectura. Sin embargo, se excluyeron características como la adecuación funcional y usabilidad, dado que el enfoque se centró en atributos internos del software más que en la experiencia del usuario final. Además, el estudio presenta limitaciones como el desequilibrio en la distribución de métricas—24

atributos cuentan con una sola métrica—y la falta de validación empírica, lo que restringe su aplicabilidad práctica.

Respecto de los modelos aplicados a PaaS, Olokunde et al. [92] identificaron la interoperabilidad, la usabilidad y la portabilidad como características clave y definieron seis métricas orientadas a evaluar atributos como la interoperabilidad, la capacidad de aprendizaje y la facilidad de instalación. El modelo fue validado de forma informal en tres plataformas comerciales—*Google App Engine*, *Windows Azure* y *Engine Yard*—, lo que permitió establecer una clasificación relativa de sus características de calidad. Sin embargo, el estudio presenta limitaciones relevantes, ya que se basó únicamente en la documentación pública disponible en los sitios web de los proveedores, sin acceso a datos experimentales, y omitió características críticas como eficiencia del desempeño, fiabilidad, seguridad, mantenibilidad y usabilidad, lo que restringe su cobertura y aplicabilidad práctica.

En el caso de IaaS, la mayoría de los proveedores definen su calidad de manera interna, sin formalizar un modelo único, lo que dificulta la estandarización de la medición y la evaluación comparativa entre proveedores. Este proceso generalmente requiere colaboración entre el proveedor y el cliente para definir, acordar y mantener los niveles deseados de calidad del servicio. Se prioriza la calidad mediante una infraestructura sólida, medidas de cumplimiento, herramientas de monitorización y acuerdos de nivel de servicio (SLA), con el objetivo de brindar a los clientes servicios cloud confiables, seguros y de alto rendimiento.

Con respecto a los modelos de calidad que abarcan los tres tipos de servicio (SaaS, PaaS e IaaS), destaca la propuesta de Zhou et al. [11], que integra los estándares ISO/IEC 25010 [1] y NIST SP 800-146 [31]. El modelo está estructurado en seis características de calidad—usabilidad, seguridad, confiabilidad, tangibilidad, capacidad de respuesta y empatía—desglosadas en 23 subcaracterísticas. El proceso de evaluación sigue un enfoque ascendente, en el cual los valores de seis métricas (5 objetivas y 1 subjetiva) se combinan mediante una suma ponderada para calcular progresivamente las subcaracterísticas, las características y, finalmente, la calidad del servicio cloud. Aunque el modelo de calidad fue aplicado a un servicio cloud de almacenamiento de un proveedor chino, presenta limitaciones en cuanto al número de métricas y la escasa validación empírica, lo que reduce su generalización y aplicabilidad en contextos más amplios.

Por su parte, Hourani et al. [93] introducen el modelo S3MQual, un marco integral diseñado para evaluar la calidad de los servicios cloud. Este modelo incluye doce características—responsabilidad, disponibilidad, mantenibilidad, escalabilidad y elasticidad, rendimiento, seguridad y privacidad, usabilidad, confiabilidad, funciones y funcionalidad, recuperabilidad, empatía y cumplimiento—y busca alinear las expectativas del usuario final con los objetivos de los proveedores mediante SLAs bien definidos. No obstante, la ausencia de validación empírica limita su adopción en entornos industriales.

Giovanoli [95] propuso un modelo de calidad basado en una revisión sistemática y en la percepción de la calidad de los *stakeholders*—clientes y proveedores—mediante entrevistas. El autor identificó atributos clave de calidad a partir de 46 artículos académicos publicados entre 2009 y 2018 y evaluó la idoneidad de estos atributos mediante una entrevista con 21 expertos en servicios cloud. Aunque el estudio proporciona una visión centrada en el usuario, presenta limitaciones importantes, como el reducido número de entrevistas y la

ausencia de métricas concretas, lo que restringe su aplicabilidad en la práctica.

Otras iniciativas, como el Índice de Medición de Servicios (*Service Measurement Index*, *SMI*) [12], representan un esfuerzo relevante hacia la estandarización multidimensional de la calidad de los servicios cloud, basada en requisitos empresariales y tecnológicos específicos. Su objetivo es permitir la comparación de servicios comerciales—tanto los servicios cloud como los no cloud—, así como entre servicios ofrecidos por distintos proveedores o por un mismo proveedor. El SMI propone un marco jerárquico compuesto por siete categorías—responsabilidad, agilidad, garantía, finanzas, rendimiento, seguridad y privacidad, y facilidad de uso—, desglosadas a su vez en 51 atributos, de los cuales solo 21 cuentan con indicadores clave de rendimiento (KPI) definidos. Esta limitación constituye su principal debilidad, ya que restringe su aplicabilidad como herramienta de evaluación comparativa. No obstante, el SMI se considera uno de los modelos más integrales, al articular dimensiones técnicas, financieras y de gobernanza bajo un enfoque orientado al negocio. Su estructura jerárquica permite evaluar tanto las capacidades del proveedor como las características intrínsecas del servicio, integrando aspectos como la auditabilidad, la elasticidad, la confiabilidad y la usabilidad.

La tabla 3.2 presenta un resumen comparativo de los estudios de modelos de calidad para servicios cloud.

Tabla 3.2: Resumen comparativo de los estudios de modelos de calidad para servicios cloud.

Autor	Tipo de Servicio	Estándar base	Características de calidad	Métricas	Validación empírica
Lee et al. [8]	SaaS	ISO/IEC 9126, IEEE 1061	Reutilización, eficiencia, confiabilidad, escalabilidad, disponibilidad	13	✗No
Wen & Dong [9]	SaaS	ISO/IEC 25010	Seguridad, calidad del servicio, calidad del software	ND	✗No
Zheng et al. [10]	SaaS	IEEE 1061	Usabilidad, disponibilidad, confiabilidad, capacidad de respuesta, seguridad, elasticidad	6	✓Sí (casos de estudio)
Lichtenthaler & Wirtz [94]	SaaS	ISO/IEC 25010, Quamoco	Rendimiento, fiabilidad, seguridad, mantenibilidad, portabilidad, compatibilidad + 53 atributos	88	△ Parcial (literatura)
Olokunde et al. [92]	PaaS	ND	Interoperabilidad, usabilidad, portabilidad	6	△ Parcial (informal)
Zhou et al. [11]	SaaS, PaaS, IaaS	ISO/IEC 25010, NIST SP 800-146	Usabilidad, seguridad, confiabilidad, tangibilidad, capacidad de respuesta, empatía + 23 subcaracterísticas	6	△ Parcial (1 caso de estudio)
Hourani et al. [93]	SaaS, PaaS, IaaS	ND	12 características (rendimiento, seguridad, privacidad, empatía, etc.)	ND	✗No
Giovanoli [95]	SaaS, PaaS, IaaS	Revisión sistemática	Atributos derivados de 46 estudios	ND	✓Sí (entrevistas)
SMI [12]	SaaS, PaaS, IaaS	KPI empresarial	7 categorías, 51 atributos (agilidad, seguridad, etc.)	21 KPI	✗No

ND : No definido

3.2.2. Modelos en tiempo de ejecución

El dinamismo inherente a la demanda de servicios cloud plantea diversos desafíos en la evaluación de su calidad, especialmente en escenarios donde es necesario realizar dicha evaluación en tiempo de ejecución. Los modelos de calidad en tiempo de ejecución permiten observar el estado actual de los atributos de calidad, facilitar la detección oportuna de fallos y guiar la adaptación dinámica del comportamiento de los servicios y sus artefactos. Esta capacidad de monitorización continua resulta esencial para garantizar el cumplimiento de los SLA, así como para mantener la confianza del usuario final en entornos caracterizados por una alta variabilidad, heterogeneidad y coexistencia de recursos.

Diversos estudios han abordado esta problemática desde enfoques parciales. Almorsy et al. [96], por ejemplo, propusieron externalizar la gestión de la seguridad en aplicaciones SaaS mediante modelos capaces de capturar los requisitos de seguridad tanto de los proveedores de servicios como de los inquilinos. Su enfoque permite la evolución dinámica de la configuración de seguridad en tiempo de ejecución, aunque su aplicabilidad se limita exclusivamente a atributos de seguridad y al ámbito de los servicios SaaS.

Por otro lado, Chen y Bahsoon [97] desarrollaron un modelo de calidad en tiempo de ejecución orientado a SaaS, utilizando un enfoque fundamentado en principios de autoadaptación. Su propuesta se centró en cuatro atributos de calidad de servicio —tiempo de respuesta, rendimiento, disponibilidad y confiabilidad—. Para posibilitar la autoadaptación, emplearon técnicas de aprendizaje automático con el fin de predecir la calidad y gestionar la incertidumbre en entornos coalojados (es decir, máquinas virtuales y físicas compartidas). Además, el modelo selecciona dinámicamente las entradas como primitivas —incluida la inferencia de la QoS de servicios vecinos— mediante aprendices híbridos duales. Si bien resulta innovador, el modelo presenta una cobertura limitada de atributos y fue validado únicamente mediante un prototipo aplicado al contexto de videojuegos, lo que restringe su generalización.

Shao et al. [98] abordaron la monitorización cloud mediante un modelo en tiempo de ejecución que representaba las preocupaciones comunes asociadas a la monitorización de servicios cloud. Su propuesta permitió generar un perfil dinámico de ejecución capaz de balancear de forma adaptativa la carga de las tareas de monitorización con la capacidad operativa de los servicios cloud. Sin embargo, el modelo no incorporó métricas específicas ni consideró atributos no funcionales relevantes para los servicios SaaS, como la fiabilidad o disponibilidad. Este trabajo se enmarca dentro del paradigma *Model-Driven-Development* (MDE), extendido hacia la ejecución mediante *runtime models*, los cuales mantienen una representación viva del sistema en operación. Dichos modelos ofrecen una visión abstracta y dinámica del comportamiento del sistema, lo que favorece la adaptabilidad y la toma de decisiones informadas en entornos distribuidos y heterogéneos como los servicios cloud. En este contexto, el *Runtime Model for Cloud Monitoring* (RMCM) proporciona una estructura jerárquica e intuitiva para organizar la información de monitorización procedente de múltiples fuentes. No obstante, la ausencia de métricas cuantitativas y la falta de integración con atributos de calidad limitan su aplicabilidad en escenarios donde se requiere una evaluación integral del desempeño y de la calidad del servicio.

Cedillo et al. [99, 100, 101] propusieron una serie de modelos en tiempo de ejecución orientados a monitorizar el cumplimiento de requisitos no funcionales descritos en los SLA, apoyándose en un *middleware* encargado de alimentar el modelo en tiempo de ejecución. Su enfoque permite analizar los datos en tiempo de ejecución, generar informes de incumplimiento de requisitos y violaciones de SLA, así como aplicar ajustes dinámicos a la infraestructura. No obstante, el modelo de calidad de referencia utilizado limita el análisis a un conjunto reducido de atributos de calidad, restringiendo así la cobertura global de la calidad del servicio.

3.3. Modelos de pronóstico para QoS en servicios cloud

En los últimos años, la creciente complejidad y dinamismo de los entornos de computación en la nube han impulsado numerosos estudios enfocados en la aplicación de técnicas de ML y DL para mejorar la gestión de los SLA, así como la monitorización y el pronóstico de los atributos de QoS en dichos entornos. Estas iniciativas tienen como objetivo anticipar la degradación del servicio, garantizar el cumplimiento de los objetivos de nivel de servicio (SLO) y optimizar la asignación de recursos mediante modelos predictivos precisos.

Para identificar los estudios más relevantes en este dominio, se realizó una búsqueda sistemática en las bibliotecas digitales ACM Digital Library, IEEE Xplore, ScienceDirect y SpringerLink, empleando la siguiente cadena de búsqueda:

> *("cloud service" OR "cloud") AND ("performance" OR "quality") AND ("machine learning" OR "prediction" OR "forecasting")*

A partir de esta búsqueda, se seleccionaron los estudios que abordaban la aplicación de técnicas de ML para la gestión de SLA, la monitorización de métricas de QoS y el pronóstico de su comportamiento en servicios cloud. En esta sección, se analizan los trabajos más relevantes, destacando los enfoques utilizados, las métricas de precisión, los modelos de pronóstico aplicados y las limitaciones experimentales reportadas. La Tabla 3.3 presenta un resumen comparativo de los estudios recopilados.

Entre los estudios iniciales, destaca Cavallo et al. [24], quienes utilizaron la monitorización histórica de QoS para comparar diferentes enfoques de predicción, incluyendo modelos lineales, medias móviles y modelos basados en series temporales como ARIMA. Aunque la propuesta no incorpora técnicas de ML, destaca la efectividad de ARIMA para la detección anticipada de violaciones del SLA, gracias a su capacidad para manejar valores atípicos y modelar la variabilidad temporal. Sin embargo, su estudio se basa en servicios SOAP y en datasets que hoy resultan obsoletos frente al predominio actual de arquitecturas *RESTful* y microservicios.

Con un enfoque más reciente, Chen y Bahsoon [97] enfrentan los retos del modelado de QoS en contextos SaaS sobre infraestructura cloud compartida, distinguiendo entre escenarios *co-located* (software ejecutándose en la misma máquina virtual) y *co-hosted* (máquinas

virtuales alojadas en la misma máquina física). Su enfoque introduce un modelado detallado de QoS, al considerar cuatro atributos de calidad—tiempo de respuesta, *throughput*, fiabilidad y disponibilidad—. La propuesta incorpora inferencia dinámica, selección de primitivas de entrada relevantes y mecanismos de autoadaptación basados en aprendices híbridos. Aunque su modelo es avanzado en términos de su capacidad de adaptación al entorno dinámico de ejecución, su aplicabilidad sigue restringida a contextos específicos *co-located* o *co-hosted* y no contempla la generalización hacia servicios multitenencia (*multi-tenant*) o multi-cloud.

Tabla 3.3: Resumen comparativo de los estudios de pronóstico de QoS en servicios cloud.

Aspecto	Chen y Bahsoon[97]	Hussain et al.[102]	White et al.[17]	Syu et al.[18]	Jirsik et al.[70]	Zhu et al.[71]
Métricas QoS	Tiempo de respuesta, *Throughput*, disponibilidad, fiabilidad	CPU, Memoria, I/O	Tiempo de respuesta	Tiempo de respuesta	Usuarios simultáneos, tiempo de respuesta de la aplicación, transacciones, tiempo de transporte de red, tamaño de transporte	Tiempo de respuesta, Throughput
Conjunto de datos experimentales	SaaS on co-hosted VMs (synthetic + runtime)	AWS EC2 (IaaS)	Web service & Emulated IoT Devices	2 RESTful Web Services	2 Enterprise Web Services	WS-Dream (synthetic QoS)
Periodo del dataset	No reportado	3 días	4 meses WS y 1 mes IoT	3 semanas	1 mes	No reportado
Datos y artefactos disponibles	✗No	✗No	✗No	✓Sí	✓Sí	△ Parcial (datos públicos)
Modelos de predicción[1]	Single learners, Hybrid dual-learners (**Hybrid-V1**)	6 TS approaches, 13 ML approaches (**M5P**)	**LSTM**, RNN, GRU	**GP**, ML, TS	**LSTM**, Holt-Winters, ARIMA	**EMD-PSO-LSTM**, LSTM, ARIMA
Resultados de precisión	SMAPE [0.22–12.67]	RMSE [0.0073–0.702]	RMSE [3.05–8.64]	RMSE [0.02–0.30], MAE [0.01–0.15], MAPE [0.01–0.10], MASE [0.2–0.6]	MAPE [1.61–497.23]	RMSE [0.05–0.20], MAE [0.02–0.10], MAPE [0.01–0.05]
Estadísticas inferenciales	✗No	✗No	✗No	✓Yes (visual + análisis de varianza)	✗No	✗No
Divulgación de hiperparámetros	✗No	✗No	✗No	△ Parcial	△ Parcial	△ Parcial
Validación cruzada	✗No	✗No	✗No	✗No	✗No	✗No
Amenazas a la validez	△ Parcial	✗No	✗No	△ Parcial	✗No	✗No

[1] Los modelos de predicción en **negrita** son los que obtuvieron los mejores resultados y su precisión se reporta justo en la fila de abajo.

Hussain et al. [102] proponen asistir a los proveedores de servicios cloud, en particular a los pequeños y medianos proveedores, en la gestión de los SLA y en la prevención de penalizaciones por incumplimiento mediante modelos predictivos. Los autores realizaron un estudio comparativo entre métodos estadísticos de series temporales (ARIMA) y modelos de predicción basados en ML para la calidad de los servicios cloud, y evaluaron su precisión en nueve intervalos de tiempo. Sin embargo, su estudio se limita a datos procedentes de un servicio IaaS en *Amazon EC2*, utilizando únicamente tres métricas (CPU, memoria y rendimiento de E/S) para representar la QoS. Además, carece de un proceso documentado de ajuste de hiperparámetros y no publica los artefactos experimentales, lo que restringe tanto su aplicación como la reproducibilidad de los resultados.

Otros estudios se centran específicamente en la predicción de QoS en el dominio de los servicios cloud utilizando técnicas de aprendizaje profundo (DL). White et al. [17] emplearon redes neuronales LSTM para pronosticar la QoS en entornos de *Internet of Things* (IoT), comparando su rendimiento con ARIMA y GRU. Sus resultados muestran una mayor precisión predictiva de las LSTM, influenciada especialmente por el tamaño del conjunto de datos y la granularidad de muestreo (intervalos de tiempo). No obstante, el estudio presenta limitaciones relevantes: se centró exclusivamente en el tiempo de respuesta como métrica de QoS, no documentó la arquitectura de los modelos ni los procedimientos de entrenamiento, y no controló la aleatoriedad en los experimentos.

Syu et al. [18] plantean el uso de programación genética (GP), SVR y redes neuronales en un experimento con dos servicios RESTful, utilizando el tiempo de respuesta como única métrica de QoS recopilada durante un período de tres semanas, lo que limita la validez externa de los resultados. Su análisis consideró métricas de error como RMSE, MAE, MAPE y MASE, así como factores experimentales como el tamaño del conjunto de entrenamiento, el horizonte de predicción y los intervalos de muestreo. No obstante, el diseño experimental se basa en una división estática entre los conjuntos de entrenamiento y prueba, carece de validación cruzada, no detalla la configuración de los modelos y no proporciona acceso al código ni a los artefactos, lo que limita tanto la reproducibilidad como la robustez de sus conclusiones.

Por otro lado, Jirsik et al. [70] exploraron el uso de redes LSTM para la predicción de QoS en entornos de redes empresariales. Su trabajo se centró en anticipar valores futuros de atributos de QoS relacionados con la red, considerando métricas como el número de usuarios concurrentes, el tiempo de respuesta de las aplicaciones, el recuento de transacciones, el tiempo de transporte en la red y el tamaño de los paquetes transferidos. Para ello, utilizaron trazas de tráfico reales de dos servicios web durante un mes, con intervalos de muestreo de 5 minutos y 1 hora. La evaluación del modelo utilizó MAPE como métrica de precisión y los autores analizaron el impacto de diferentes longitudes de secuencia y horizontes de predicción. Aunque también exploraron la influencia de parámetros como el optimizador y los pesos iniciales, la descripción técnica proporcionada fue mínima. Una limitación clave radica en el uso de un conjunto de datos específico de una sola red empresarial, lo que reduce la validez externa de los resultados. Además, el entrenamiento se basó en una ventana fija de una semana, sin validación cruzada, lo que afecta tanto la reproducibilidad como la capacidad de generalización del modelo. A pesar de estas li-

mitaciones, el estudio refuerza el potencial de los modelos de aprendizaje profundo, en particular las redes LSTM, para la predicción de QoS con fuerte dependencia temporal en entornos de servicios dinámicos y subraya el desafío continuo de equilibrar precisión predictiva y eficiencia computacional.

En los estudios más recientes, Syu y Wang [23] realizaron una revisión exhaustiva de enfoques basados en series temporales y técnicas de ML para la predicción de QoS, identificando tendencias, brechas y limitaciones metodológicas. Su alcance, sin embargo, se limitó a un único atributo de QoS —el tiempo de respuesta— y a un único dominio de aplicación —los servicios web—, sin explorar su aplicabilidad en contextos más heterogéneos o con múltiples métricas de calidad. Los autores evaluaron un amplio conjunto de modelos bajo distintas configuraciones experimentales, considerando variaciones en los intervalos de muestreo y en las divisiones entre entrenamiento y prueba.

De forma similar, Zhu et al. [71] introdujeron un modelo EMD-PSO-LSTM que integra *Empirical Mode Decomposition* (EMD), *Particle Swarm Optimization* (PSO) y redes LSTM. En este enfoque, EMD descompone las series temporales en modos intrínsecos más estables, facilitando el aprendizaje de subpatrones por parte de LSTM, mientras que PSO optimiza hiperparámetros, tales como el número de unidades ocultas. Sus experimentos utilizaron el dataset SERV1 de WS-DREAM, con métricas de tiempo de respuesta y rendimiento recolectadas a intervalos de 5 minutos. Aunque se trata de un dataset ampliamente utilizado, los autores no especificaron la duración total de la serie temporal ni el número de servicios considerados. La evaluación se realizó mediante RMSE, MAE y MAPE, comparando su enfoque con ARIMA, Holt-Winters, LSTM y EMD-LSTM. Entre las limitaciones del estudio destacan la consideración de solo dos métricas de QoS, la ausencia de validación cruzada, el uso de una única división entrenamiento-prueba (70/30), la falta de acceso al código y a los artefactos experimentales y la documentación incompleta de la configuración del modelo.

En cuanto a los modelos de calidad, se observa una cobertura parcial de las características definidas en los estándares, un bajo grado de validación empírica, una integración limitada de múltiples *stakeholders* (incluyendo *brokers*) y una fuerte dependencia de modelos genéricos que no reflejan adecuadamente las propiedades del entorno cloud. Además, la mayoría de las propuestas se enfoca en la fase de operación del servicio, descuidando etapas tempranas como el diseño arquitectónico o la definición de requisitos. En consecuencia, resulta fundamental avanzar hacia modelos de calidad contextualizados al dominio específico, instanciables por tipo de servicio, que integren la visión de todos los *stakeholders*, aplicables a lo largo del ciclo de vida del servicio cloud y de sus artefactos, y mantengan la independencia de la plataforma. Una propuesta basada en ISO/IEC 25010 [1] y adaptada al dominio cloud permitirá operacionalizar las métricas de calidad, habilitar la trazabilidad, facilitar el análisis de causas raíz y promover la adaptación proactiva y la retroalimentación continua. Todo ello contribuiría a una toma de decisiones informada, al cumplimiento de los SLA y a una adopción efectiva tanto en entornos industriales como académicos.

En relación con los modelos de pronóstico para QoS en servicios cloud, la evidencia empírica confirma que los enfoques basados en *deep learning* superan a los métodos es-

tadísticos de series temporales en términos de precisión y capacidad predictiva [102, 18, 17]. No obstante, persisten carencias metodológicas recurrentes como: (1) la limitada disponibilidad de conjuntos de datos reales y actualizados y de acceso público—ya que en muchos casos los datasets son propietarios o están desactualizados— [18]; (2) el análisis restringido a un número limitado de métricas de QoS; (3) la documentación insuficiente sobre decisiones de arquitectura, estrategias de ajuste de hiperparámetros y protocolos de evaluación; (4) la falta de control de la aleatoriedad en los procesos de entrenamiento y prueba; y (5) la ausencia de comparaciones experimentales robustas con pruebas de significancia y validación cruzada.

Estas deficiencias metodológicas son coherentes con los problemas identificados por Vegas y Elbaum [4], quienes evidencian que, en experimentos con *Deep Neural Networks (DNN)* aplicados a la ingeniería de software, son recurrentes la falta de transparencia en el diseño arquitectónico, la omisión de estrategias formales de ajuste de hiperparámetros y la ausencia de análisis estadísticos rigurosos que respalden las conclusiones. En esta línea, los trabajos de White et al. [17] y Syu y Wang [23] ilustran estas limitaciones, mostrando avances parciales, pero sin alcanzar estándares de reproducibilidad y validez experimental ampliamente aceptables.

Aunque Syu y Wang [23] abordaron parcialmente algunos de los desafíos señalados por Vegas y Elbaum [4] —como la falta de reproducibilidad y la ausencia de artefactos públicos—, persistieron limitaciones importantes, especialmente en la documentación de los métodos de ajuste de hiperparámetros. Asimismo, no se aplicaron pruebas de estadística inferencial ni se consideraron de manera explícita las compensaciones entre precisión predictiva y coste computacional, aspectos críticos para la adopción práctica de estos modelos en entornos industriales.

En este contexto, los estudios revisados ponen de manifiesto la necesidad de enfoques metodológicos más robustos y empíricamente verificables, capaces de superar las limitaciones relacionadas con la reproducibilidad, la disponibilidad de datos reales y la ausencia de marcos sistemáticos que guíen la construcción y validación de modelos predictivos. En particular, el uso del enfoque basado en modelos de *deep learning*, que incorpore un diseño experimental replicable, acompañado de validación estadística, validación cruzada y una arquitectura adaptable a distintos dominios cloud. De forma que mejore la precisión, la aplicabilidad práctica en escenarios de selección de servicios, escalabilidad dinámica y gestión proactiva de los SLA.

3.4. Conclusiones

El presente capítulo presenta el análisis del estado del arte, el cual permitió identificar que, a pesar de los avances en la definición y caracterización de la QoS en servicios cloud, persisten desafíos importantes en la definición, selección y aplicación de métricas para los distintos atributos de calidad. Los estudios revisados proponen métricas y taxonomías heterogéneas que no siempre cuentan con mecanismos claros de medición ni con procedimientos que faciliten su uso en entornos operativos. Esta falta de alineación dificulta la evaluación sistemática de la QoS y limita la comparación entre estudios.

También se observó que los modelos de calidad existentes, aunque en algunos casos se derivan de estándares reconocidos, no presentan una descomposición suficiente en atributos medibles ni una guía precisa sobre cómo operacionalizarlos mediante métricas y sus mecanismos de medición. Esta brecha entre el constructo teórico y su implementación práctica reduce su utilidad para apoyar decisiones sobre rendimiento, capacidad o disponibilidad en servicios cloud en operación.

Por otra parte, la revisión de los métodos de pronóstico de QoS evidencia avances relevantes en el uso de técnicas estadísticas y de aprendizaje automático y profundo, especialmente aquellas basadas en redes recurrentes. Sin embargo, gran parte de los estudios se basa en conjuntos de datos limitados, métricas aisladas y experimentos poco reproducibles. Además, la falta de protocolos experimentales detallados y la escasa disponibilidad de datos reales refuerzan la necesidad de investigaciones que combinen rigurosidad metodológica y validez empírica.

En definitiva, el estado del arte pone de manifiesto la necesidad de contar con una taxonomía de métricas que considere sus mecanismos de medición, un modelo de calidad que incluya su descomposición hasta métricas y operacionalizaciones con el fin de facilitar su aplicación en la práctica, y modelos de pronóstico de calidad del servicio dentro de un marco sistemático y reproducible. Sobre esta base, el siguiente capítulo presenta la taxonomía de métricas de calidad cuyo propósito es organizar, clasificar y fundamentar las métricas de QoS relevantes para los servicios cloud.

Capítulo 4

Taxonomía de Métricas de Calidad para Servicios Cloud

En este capítulo se presenta una taxonomía de métricas de calidad aplicadas a los servicios cloud, cuyo objetivo es la identificación y clasificación de dichas métricas dentro de este dominio, así como proporcionar una terminología común de los conceptos involucrados en la evaluación de la calidad de los servicios cloud.

La Sección 4.1 introduce el propósito e importancia de disponer de una taxonomía de métricas de calidad para servicios cloud, así como su relación con el objeto de estudio. La Sección 4.2 describe el diseño de la revisión sistemática de la literatura empleada para la construcción de la taxonomía. La Sección 4.4 presenta el proceso de construcción de la taxonomía de métricas de calidad para servicios cloud, junto con el metamodelo utilizado para organizar sus elementos y relaciones. La Sección 4.5 incluye una discusión de los resultados obtenidos que conforman la taxonomía propuesta. La Sección 4.6 expone las amenazas a la validez asociadas al mapeo sistemático realizado. La Sección 4.7 discute las implicaciones de la taxonomía tanto para la investigación como para la práctica profesional. Finalmente, la Sección 4.8 presenta las conclusiones derivadas de los resultados obtenidos.

4.1. Introducción

La evaluación de la calidad de los servicios cloud constituye un desafío creciente debido a la diversidad de métricas y enfoques presentes en la literatura. Aunque en los últimos años se ha trabajado en múltiples propuestas para medir y analizar distintos aspectos de la calidad, la mayoría de ellas se centra en aspectos aislados, lo que dificulta alcanzar una visión integral del dominio. Además, las métricas identificadas presentan distintos niveles de madurez, formalización y validez empírica, lo que complica su aplicación práctica y limita la posibilidad de realizar comparaciones consistentes entre estudios.

Frente a este escenario, resulta necesario organizar y clasificar el conocimiento existente mediante un enfoque sistemático que permita consolidar las contribuciones previas y establecer una base conceptual común. Con este propósito, este capítulo presenta una taxonomía de métricas de calidad para servicios cloud, obtenida a partir de una revisión sistemática de la literatura y construida mediante un proceso riguroso de extracción, análisis y síntesis de información.

La taxonomía resultante ofrece un marco estructurado que facilita la identificación, comparación y reutilización de métricas, promoviendo así una comprensión compartida entre investigadores y profesionales del área. Asimismo, constituye un paso hacia la transición desde estándares de calidad de carácter general, como ISO/IEC 25010 [1], hacia modelos operativos y específicos para el dominio de la computación en la nube.

Por lo tanto, las principales contribuciones de este capítulo son:

1. Un metamodelo que sustenta la taxonomía propuesta, representando los diferentes conceptos involucrados y las relaciones existentes entre ellos.

2. Un catálogo de métricas de calidad para servicios cloud, elaborado a partir de la literatura relevante.

3. Una taxonomía de métricas para servicios cloud organizada jerárquicamente de acuerdo con el modelo de calidad ISO/IEC 25010 [1] y los conceptos definidos por los estándares NIST SP 800-145 [30] y NIST SP 800-146 [31]. El propósito de esta taxonomía es clasificar y comparar sistemáticamente las métricas de acuerdo a diferentes criterios.

Estas contribuciones buscan fortalecer la reproducibilidad de los estudios empíricos en ingeniería de software y servicios cloud, proporcionando una estructura clara y reutilizable que facilite la integración del conocimiento y promueva la toma de decisiones basada en evidencia en entornos cloud. Asimismo, el análisis detallado de las métricas identificadas y su organización en categorías coherentes sientan las bases para futuras investigaciones orientadas al diseño de herramientas inteligentes de apoyo a la decisión en la gestión de la calidad de servicios cloud.

4.2. Diseño del protocolo de revisión sistemática

El diseño del protocolo de revisión sistemática constituye un paso fundamental para garantizar la validez y la fiabilidad de los resultados obtenidos. En esta sección se describe de manera sistemática la planificación metodológica adoptada, que incluye la definición de los objetivos y las preguntas de investigación, así como la estrategia de búsqueda y los criterios de inclusión y exclusión. Además, se presentan las técnicas utilizadas para la extracción y el análisis de los datos, con el fin de asegurar la trazabilidad del proceso y la reproducibilidad de los hallazgos. De esta manera, se establece un marco riguroso que respalda la pertinencia de las decisiones metodológicas y la solidez de las conclusiones derivadas del estudio.

El objetivo de la revisión sistemática de la literatura es el siguiente:

> *Identificar y clasificar taxonómicamente la evidencia disponible sobre métricas de calidad para servicios cloud, y proporcionar una taxonomía que permita compararlas, así como identificar sus limitaciones.*

La taxonomía propuesta pretende contribuir al cuerpo de conocimientos sobre la evaluación de los servicios cloud a través de los siguientes aspectos:

1. Proporcionar un conjunto de métricas de calidad para los servicios cloud.

2. Identificar los atributos de calidad medidos por dichas métricas y alinearlos con las características de calidad propuestas en la ISO/IEC 25010 [1], así como con los conceptos de computación en la nube definidos por NIST SP 800-145 [30] y NIST SP 800-146 [31].

3. Detectar las limitaciones de los estudios previos y sugerir una agenda para investigaciones futuras.

Con esta taxonomía se pretende establecer una terminología común para los conceptos involucrados en la evaluación de los servicios cloud. Para ello, se apoya en un metamodelo que representa los diferentes conceptos y las relaciones entre ellos, lo cual facilita la comunicación y permite su posterior reutilización tanto por profesionales como por investigadores.

La identificación de métricas de calidad y la construcción de la taxonomía propuesta se llevan a cabo mediante una revisión sistemática de la literatura *(Systematic Literature Review - SLR)*, siguiendo las directrices planteadas por Kitchenham et al. [26]. La Figura 4.1 describe el proceso e incluye las actividades junto con sus respectivas salidas. Las actividades correspondientes a las fases de diseño de la SLR (planificación, conducción y presentación de resultados) se detallan en esta sección, mientras que el análisis bibliométrico del conjunto de artículos primarios se presenta en la Sección 4.3 y la construcción de la taxonomía en la Sección 4.4.

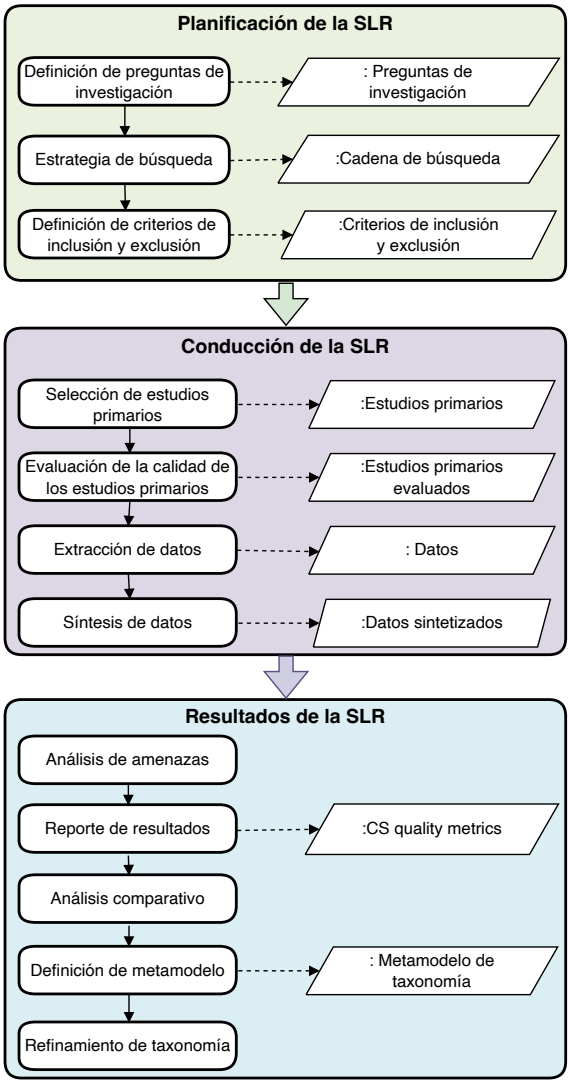


Figura 4.1: Fases de la revisión sistemática de la literatura [2].

4.2.1. Planificación de la SLR

En esta sección se define el protocolo de revisión, el cual proporciona un marco para documentar las decisiones de diseño del estudio necesarias para minimizar el sesgo. Las actividades realizadas para definir dicho protocolo fueron las siguientes: definición de la pregunta de investigación, definición de la estrategia de búsqueda y definición de los criterios de inclusión y exclusión. Estas actividades se describen en las siguientes subsecciones.

4.2.1.1. Pregunta de investigación

La estrategia PICOC, sugerida por Petticrew et al. [103] se utiliza para enmarcar los elementos de la pregunta de investigación con el fin de desarrollar el protocolo de revisión. Los elementos PICOC utilizados en este estudio son:

- `Intervención (I)`: Caracterización, extracción de datos y síntesis.
- `Comparación (C)`: Una comparación llevada a cabo mediante el mapeo de los estudios primarios en una taxonomía (marco de caracterización).
- `Resultado (O)`: Una taxonomía de métricas para servicios cloud alineada con los estándares ISO/IEC 25010 [1], NIST SP 800-145 [30] y NIST SP 800-146 [31].
- `Contexto (C)`: Una revisión sistemática con el fin de consolidar la investigación revisada por pares.

Se siguieron las directrices propuestas por Easterbrook et al. [104] y por Kitchenham y Charters [26] para definir una pregunta de investigación de carácter exploratorio y descriptivo. La principal pregunta de investigación abordada en este estudio es:

> *¿Qué métricas se han utilizado para evaluar la calidad interna y externa de los servicios cloud? ¿cómo se miden y se utilizan?*

La revisión sistemática se limita deliberadamente a las métricas de calidad interna y externa de los servicios cloud porque constituyen los indicadores directamente utilizados para monitorizar, evaluar y gestionar el comportamiento de los sistemas en tiempo real. A diferencia de las métricas de calidad en uso (que dependen de percepciones subjetivas y requieren estudios con usuarios), las métricas internas y externas se obtienen directamente a partir de sistemas de monitorización y son esenciales para el análisis operativo y el pronóstico de QoS.

Este enfoque ofrece varias ventajas. En primer lugar, estas métricas son observables y medibles en tiempo real, lo que facilita la generación de datos consistentes y reproducibles. En segundo lugar, están directamente alineadas con los SLA y con las prácticas de gestión utilizadas por proveedores y operadores de servicios cloud, lo que incrementa la relevancia práctica del estudio. En tercer lugar, permiten identificar y analizar relaciones causales entre condiciones internas del sistema y comportamientos externos, algo

fundamental para construir modelos de calidad y de predicción. Finalmente, su homogeneidad técnica aumenta la comparabilidad entre estudios y evita mezclar constructos heterogéneos, garantizando la coherencia conceptual del análisis.

En conjunto, centrarse en métricas internas y externas asegura un marco sólido y operativo para abordar los objetivos de evaluación y pronóstico de calidad de servicio, mientras que las métricas de calidad en uso quedan fuera del alcance de esta investigación por su naturaleza subjetiva, dependiente del contexto y metodológicamente distinta.

La *calidad interna* está constituida por aquellos atributos de un artefacto de software que pueden medirse únicamente a partir del conocimiento del artefacto [105]. Algunos ejemplos de atributos cloud son:

i) *Capacidad*, que puede medirse en términos de capacidad de almacenamiento de recursos (tamaño en gigabytes, GB) o capacidad de red (número de conexiones disponibles);

ii) *Densidad*, que puede medirse considerando el número de aplicaciones o el número de máquinas virtuales;

iii) *Complejidad*, que puede evaluarse en términos de la estructura del código fuente del servicio cloud o de la complejidad de la definición de la interfaz de servicio (p. ej., interfaces descritas mediante el Lenguaje de Descripción de Servicios Web (WSDL)), pudiéndose medir como el esfuerzo mínimo de refactorización;

iv) *Legibilidad*, que puede medirse mediante métricas de legibilidad aplicadas a las descripciones de servicios web (especificaciones WSDL).

Los atributos de *calidad externa* son aquellos atributos que no pueden medirse únicamente a partir del artefacto de software, sino que requieren considerar el artefacto junto con su entorno y las interacciones entre ambos [105]. Algunos ejemplos de estos atributos son la fiabilidad, el rendimiento, la usabilidad y la mantenibilidad.

Esta pregunta de investigación permitió comprender y sintetizar la evidencia existente respecto a las métricas actualmente disponibles, así como identificar áreas para futuras investigaciones. Dado que la pregunta principal es amplia, se refinó en un conjunto de preguntas más específicas. En particular, se busca explicitar qué características y atributos de calidad se están midiendo, cómo se han utilizado las métricas, a qué tipo de servicio se han aplicado, para qué tipo de *stakeholders* resultan útiles y qué métodos se han empleado para su validación. Las preguntas resultantes son las siguientes:

- **PI1**: ¿Qué características y atributos de calidad se evaluaron?
- **PI2**: ¿Qué tipo de métricas se utilizaron?
  - PI2.1: Si la métrica es base, ¿qué método de medida y unidad se emplearon para calcularla?
  - PI2.2: Si la métrica es derivada, ¿qué función y unidad de medida se emplearon para calcularla?
- **PI3**: ¿Existen herramientas que apoyen el proceso de medición? En caso afirmativo, ¿cuáles son?
- **PI4**: ¿Qué tipo de medición se obtiene a partir de las métricas proporcionadas?

- **PI5**: ¿Durante qué fases del ciclo de vida del servicio cloud se utilizaron estas métricas?
- **PI6**: ¿Para qué tipo de *stakeholders* (roles cloud) son relevantes estas métricas?
- **PI7**: ¿A qué tipo de servicio cloud (es decir, SaaS, PaaS, IaaS) se aplicaron estas métricas?
- **PI8**: ¿Qué artefactos o recursos cloud se midieron?
- **PI9**: ¿Qué método de validación se utilizó para aportar evidencia sobre la validez y utilidad de las métricas?

4.2.1.2. Estrategia de búsqueda

Se definió una estrategia de búsqueda no sesgada y efectiva mediante el establecimiento de un intervalo de tiempo de publicación preciso, la definición de una *cadena de búsqueda* y el empleo de motores de búsqueda automáticos en bibliotecas digitales reconocidas. El período revisado incluye estudios publicados desde 2006 hasta noviembre de 2018. Se eligió 2006 como año de inicio porque Amazon Web Services—uno de los primeros servicios cloud—comenzó a operar en ese año.

El proceso de búsqueda combinó búsquedas automatizadas en bibliotecas digitales seleccionadas con búsquedas manuales adicionales en las conferencias y revistas más relevantes, en caso de que fueran necesarias. Las bibliotecas digitales seleccionadas fueron *IEEE Xplore*, *ACM Digital Library*, *SpringerLink* y *ScienceDirect*, dado que constituyen las fuentes más utilizadas en ingeniería de software [106]. Cabe señalar que *Google Scholar* no fue seleccionado como fuente debido a la baja precisión de sus resultados de búsqueda y a la generación de numerosos resultados irrelevantes [107].

Para construir la cadena de búsqueda se siguieron cuatro pasos:

1. Derivación de términos principales (palabras clave) a partir de la pregunta de investigación;
2. Uso del operador booleano `AND` para vincular los términos principales;
3. Identificación de ortografías alternativas y sinónimos para los términos principales;
4. Uso del operador booleano `OR` para incorporar ortografías alternativas y sinónimos

A continuación, se modificaron y combinaron estos términos de búsqueda para crear un conjunto de cadenas de búsqueda candidatas. Se llevaron a cabo varias búsquedas piloto y la cadena de búsqueda se refinó tantas veces como fue necesario para mejorar la integridad de los resultados y encontrar la cadena de búsqueda más adecuada. La integridad de los resultados se evaluó mediante un conjunto de diez estudios conocidos. En la Tabla 4.1 se presenta la cadena de búsqueda resultante.

Tabla 4.1: Palabras clave y términos relacionados para la cadena de búsqueda.

Palabra clave	***Términos alternativos y sinónimos***
Métrica	*((metric* OR measur*) AND*
Calidad	*(QoS OR "quality of service" OR "quality model" OR "evaluation model" OR "assessment model" "OR "quality in cloud" OR "quality of cloud") AND*
Nube	*(cloud*))*

La búsqueda se llevó a cabo aplicando la cadena de búsqueda a los mismos metadatos (es decir, título, resumen y palabras clave) de cada fuente de datos, lo que implicó adaptar la sintaxis de la cadena de búsqueda a cada biblioteca digital. Las cadenas de búsqueda finales utilizadas en cada fuente están disponibles en el sitio web que acompaña este documento (https://bit.ly/frameworkCloudQuality).

Con el fin de garantizar la calidad del proceso de búsqueda, se verificó que las revistas y conferencias de mayor relevancia (véase la Tabla 4.1) para los dominios de computación en la nube y de calidad del software estuvieran incluidas en las bibliotecas digitales seleccionadas. La lista de revistas se obtuvo a partir del factor de impacto JCR. La lista de conferencias se obtuvo de forma análoga tomando como referencia las conferencias mejor clasificadas según la clasificación de conferencias CORE (http://www.core.edu.au/conference-portal), seleccionándose aquellas con categorías CORE A* y A. En particular, se verificó que todas las ediciones de las actas de cada congreso y cada revista, desde 2006 hasta 2018, estuvieran indexadas en al menos una de las bibliotecas digitales. Posteriormente, se realizó una búsqueda manual para recuperar las ediciones que no estaban disponibles.

4.2.1.3. Criterios de inclusión y exclusión

Se utilizaron los siguientes criterios de inclusión y exclusión para seleccionar los artículos candidatos para nuestro estudio:

- `Criterios de inclusión`:

  1. Estudios publicados como artículos científicos revisados por pares;
  2. Estudios que propongan o utilicen métricas para evaluar la calidad interna o externa de los servicios cloud;
  3. Estudios que introduzcan marcos o métodos para evaluar la QoS de los servicios cloud;
  4. Estudios que reporten investigaciones empíricas cuyo objetivo sea validar la utilidad de las métricas para los servicios cloud.

- `Criterios de exclusión`:

1. Estudios que propongan métricas que no estén relacionadas con la calidad interna o externa de los servicios cloud;
2. Estudios que propongan métricas, pero no expliquen cómo medirlas;
3. Estudios que presenten atributos de calidad, pero no propongan métricas para medirlos;
4. Editoriales, resúmenes o artículos breves (de menos de cinco páginas);
5. Artículos duplicados del mismo estudio encontrados en diferentes fuentes;
6. Estudios no escritos en inglés.

4.2.2. Conducción de la SLR

La conducción de la SLR siguió estrictamente el protocolo definido, se especificó un conjunto de subprocesos a seguir : selección de estudios primarios, evaluación de la calidad de los estudios, extracción de datos y síntesis de datos.

4.2.2.1. Selección de estudios primarios

Para identificar los estudios primarios se utilizó la cadena de búsqueda predefinida, adaptando la búsqueda automatizada a cada biblioteca digital. A continuación, se realizaron tres actividades principales: un escaneo rápido de todos los estudios recuperados tras las búsquedas automatizadas y manuales, una lectura completa de los estudios seleccionados y reuniones de equipo para alcanzar consensos entre los revisores en caso de discrepancias.

1. *Escaneo rápido*: los estudios primarios relevantes se seleccionaron inicialmente examinando el título, las palabras clave y el resumen del artículo. En algunos casos, también se revisaron la introducción y las conclusiones.

2. *Lectura completa*: cuando la decisión sobre la inclusión o exclusión no estaba clara, se realizó una revisión exhaustiva del texto completo.

3. *Reuniones de equipo*: si persistían las dudas, se concertaron reuniones de equipo para discutirlas y llegar a un acuerdo entre los revisores.

Con el fin de evaluar la precisión de la fase de selección de estudios primarios, se realizó un estudio piloto con un experto con el fin de probar y mejorar tanto la estrategia de búsqueda como los criterios de inclusión y exclusión. Asimismo, tres investigadores ejecutaron una segunda iteración sobre una muestra aleatoria de artículos, conformada por diez estudios incluidos en la primera iteración y diez que habían sido excluidos. El nivel de acuerdo entre los investigadores se evaluó mediante el índice *Kappa de Fleiss* [108], cuyo valor oscila entre 0 (sin coincidencia) y 1 (coincidencia total).

4.2.2.2. Evaluación de la calidad

La calidad de los estudios primarios seleccionados es fundamental para obtener resultados fiables. Por ello, se definió una lista de verificación basada en los criterios propuestos por Kitchenham y Charters [26] para evaluar la calidad de los estudios primarios. La calidad de cada estudio se puntuó en función del cumplimiento de los diez ítems de calidad, utilizando una escala predefinida (`S, P, N`): S (sí, cumple plenamente = 1), P (parcialmente = 0,5) y N (no cumple = 0). En la Tabla 4.2 se muestran los ítems de calidad y sus criterios.

Tabla 4.2: Ítems de calidad para evaluar los estudios primarios.

Escala	Pregunta de calidad
`Pregunta 1 (Motivación)` ¿Está claramente especificado el problema de investigación?	
1	Descripción explícita del problema
0.5	Descripción general del problema
0	No hay descripción del problema
`Pregunta 2 (Objetivo)` ¿Están claramente establecidos los fines u objetivos de la investigación?	
1	Fines y objetivos explícitos
0.5	Fines y objetivos generales
0	No se indican fines u objetivos
`Pregunta 3 (Contexto)` ¿Está claramente especificado el contexto del estudio?	
1	Contexto explícito del problema, respaldado por referencias
0.5	Contexto general del problema, respaldado por referencias
0	Descripción del contexto sin relación con el problema
`Pregunta 4 (Datos)` ¿Están claramente definidas las métricas utilizadas para evaluar la calidad de los servicios cloud?	
1	Descripción explícita de la métrica
0.5	Descripción general de la métrica
0	No hay descripción de la métrica
`Pregunta 5 (Datos)` ¿Están claramente definidos los métodos o funciones de medición utilizados para calcular las métricas?	
1	Descripción explícita de los métodos/funciones de medición
0.5	Descripción general de los métodos/funciones de medición
0	No hay descripción de los métodos/funciones de medición
`Pregunta 6 (Utilidad)` ¿Están validadas empíricamente las métricas?	
1	Validación empírica explícita de las métricas
0.5	No hay validación, pero sí una prueba de concepto
0	Sin validación
`Pregunta 7 (Utilidad)` ¿Hay evidencia suficiente que demuestre cómo pueden utilizarse las métricas en la práctica?	
1	Uso explícito de las métricas respaldado por la práctica
0.5	Uso general de las métricas respaldado por la práctica
0	No existe respaldo práctico para el uso de las métricas
`Pregunta 8 (Contribución)` ¿Se discuten las contribuciones o resultados del estudio?	
1	Lista explícita de contribuciones/resultados del estudio
0.5	Discusión general de las contribuciones/resultados
0	No se describen las contribuciones/resultados
`Pregunta 9 (Perspectivas)` ¿Se informan las ideas y lecciones aprendidas del estudio?	
1	Lista explícita de ideas y/o lecciones aprendidas del estudio
0.5	Discusión general de ideas o lecciones aprendidas

Tabla 4.2 (continúa)

Escala	Pregunta de calidad
0	No se describen ideas ni lecciones aprendidas
`Pregunta 10 (Limitaciones)` ¿Se discuten las limitaciones del estudio?	
1	Lista explícita de limitaciones del estudio
0.5	Descripción general de las limitaciones
0	No se mencionan las limitaciones

Un experto contribuyó al refinamiento de la definición de los ítems de calidad utilizados para evaluar los estudios primarios. En particular, recomendó la inclusión de la pregunta 7, destinada a evaluar la utilidad práctica de las métricas, así como la mejora de los criterios de calidad correspondientes a las preguntas 8 y 10, relacionados con la contribución y las limitaciones de los estudios.

La calidad global de un artículo se calcula sumando las puntuaciones de todos los ítems de calidad. Por lo tanto, la puntuación máxima que puede alcanzar un artículo es de 10 puntos, lo que indica que cumple con todos los criterios de calidad. El umbral de calidad se estableció utilizando la media (equivalente al segundo cuartil, 5 puntos) como punto de corte, con el objetivo de minimizar el sesgo y maximizar la validez interna y externa. Esto significa que los estudios que obtengan una puntuación inferior al umbral quedan descartados de la revisión.

Además de la calidad de los estudios primarios, también se evalúa la calidad del lugar de publicación de cada estudio, así como la cantidad de citas que este ha recibido. Estos aspectos permiten estimar el impacto potencial y la influencia de los estudios seleccionados. Resumiendo, se consideran dos criterios:

1. Relevancia de la revista o conferencia donde se publicó el artículo;
2. Número de citas del artículo.

El primer criterio se valora mediante el *ranking CORE-ERA* para artículos de congresos y el factor de impacto del *Journal Citation Reports* (JCR) para artículos de revistas. En cuanto al segundo criterio, se tiene en cuenta el número de citas recibidas por cada estudio primario según *Google Scholar*. Dado que los artículos más recientes suelen tener menos citas y no deberían ser penalizados por ello, seguimos una estrategia similar a la propuesta por Montagud et al. [109] y fijamos el año 2017 como punto de corte para distinguir las publicaciones más recientes. En la Tabla 4.3 se presentan los criterios y la escala utilizada para evaluar el impacto potencial y la influencia de los estudios primarios seleccionados.

Tabla 4.3: Impacto potencial e influencia de los estudios primarios.

Criterio	Descripción	Puntos
Impacto de la publicación		
Muy relevante	Artículos publicados en congresos CORE A* y A, o en revistas indexadas en JCR.	10
Pertinente	Artículos publicados en congresos CORE B.	5

Criterio	Descripción	Puntos
No relevante	Resto de publicaciones.	0
Citas (publicación anterior a 2017)		
Alto	≥ 50 citas.	10
Medio	10–49 citas.	5
Bajo	< 10 citas.	2
No citado	0 citas.	0
Citas (publicación posterior a 2017)		
Citado	≥ 1 cita.	10
No citado	0 citas.	5

4.2.2.3. Estrategia de extracción de datos

La estrategia de extracción de datos proporciona un marco para caracterizar las métricas individuales y nos ayuda a clasificar y comparar taxonómicamente todas las métricas recopiladas. En primer lugar, se recopila información básica de cada publicación: fecha de publicación, tipo de publicación (revista o conferencia), nombre de la publicación, número de citas y nombres y afiliación (país) de los autores. Posteriormente, se recopilan los datos del estudio necesarios para dar respuesta a las preguntas de investigación planteadas mediante la definición de un formulario de extracción de datos y una lista de verificación.

El objetivo de esta estrategia es garantizar una clasificación coherente de todos los estudios primarios y obtener una comprensión clara del estado del arte de las métricas de QoS para los servicios cloud. La evaluación de un experto sugirió incluir el indicador como opción adicional dentro del criterio relativo al tipo de métrica. La Tabla 4.4 resume los criterios de extracción de datos utilizados.

Tabla 4.4: Criterios de extracción de datos de los estudios primarios.

Criterio	Posibles opciones / Información recolectada
Características de QoS	Eficiencia de desempeño, Fiabilidad, Portabilidad, Seguridad, Mantenibilidad, Adecuación Funcional, Usabilidad, Compatibilidad
Atributo de calidad	Nombre del atributo
Tipo de métrica	Base (método de medición), Derivada (función de medición), Indicador (modelo de análisis)
Unidad de medida	Unidad de la métrica
Función de medición	Fórmula de cálculo y explicación sobre cómo se calcula la métrica
Soporte de herramientas	Manual, Automatizado (nombre de la herramienta)
Resultado de la medición	Cualitativo, Cuantitativo, Híbrido
Fases del ciclo de vida	Requisitos, Adquisición, Desarrollo, Integración, Operación, Retiro
Artefacto cloud	Especificación de servicios cloud, Arquitectura cloud, Servicio cloud
Tipo de servicio	SaaS, PaaS, IaaS
Punto de vista de los *stakeholders*	Proveedor, Consumidor, Facilitador, Desarrollador, Usuario final
Procedimiento de validación	Validación teórica, Validación empírica, Sin validación

A continuación se explican con más detalle en las siguientes subsecciones los criterios de extracción de datos resumidos en la Tabla 4.4.

a. **Característica de QoS y atributo de calidad**

El propósito de este criterio es extraer la característica de calidad que se mide mediante la métrica y su atributo de calidad. Cada métrica se clasifica de acuerdo con las siguientes características de calidad internas y externas propuestas por la norma ISO/IEC 25010 [1]:

- `Eficiencia de desempeño`: si la métrica mide un atributo relacionado con la cantidad de recursos utilizados por los servicios cloud en determinadas condiciones;
- `Fiabilidad`: si la métrica mide un atributo relacionado con la capacidad de un servicio cloud para realizar las funciones especificadas cuando se usa bajo ciertas condiciones y en un intervalo de tiempo determinado;
- `Portabilidad`: si la métrica mide un atributo relacionado con la capacidad del servicio cloud para transferirse de manera efectiva y eficiente de un entorno de *hardware*, *software*, operativo o de uso a otro. Esta característica también incluye aspectos relacionados con la escalabilidad y elasticidad del servicio cloud;
- `Seguridad`: si la métrica mide un atributo relacionado con la capacidad del servicio cloud para proteger la información y los datos de modo que personas o sistemas no autorizados no puedan leerlos o modificarlos;
- `Mantenibilidad`: si la métrica mide un atributo relacionado con la capacidad del servicio cloud para modificarse de manera efectiva y eficiente debido a necesidades evolutivas, correctivas o perfectivas;
- `Adecuación funcional`: si la métrica mide un atributo relacionado con la capacidad del servicio cloud para proporcionar funciones que satisfagan las necesidades explícitas e implícitas de los usuarios en condiciones específicas;
- `Usabilidad`: si la métrica mide un atributo relacionado con la capacidad del servicio cloud para ser entendido, aprendido, operado y atractivo para el usuario en el contexto específico de uso;
- `Compatibilidad`: si la métrica mide un atributo relacionado con la capacidad de dos o más servicios cloud para intercambiar información y/o realizar sus funciones requeridas cuando comparten el mismo entorno de *hardware* o *software*.

b. **Tipo de métrica, unidad de medida y función de medición**

Existen tres tipos de métricas: base, derivada e indicador. Para cada métrica se registró su tipo, junto con la información mínima necesaria para su caracterización. Para las métricas base, se documenta el método y la *unidad de medida*; para las métricas derivadas, la *función de medición* y la *unidad de medida* correspondiente; y para los indicadores, el modelo de análisis empleado, que describe un algoritmo o cálculo que combina una o más medidas con criterios de decisión asociados [110]. Un *criterio de decisión* utiliza umbrales, objetivos o patrones para determinar la necesidad de acción o investigación adicional, o para describir el nivel de confianza en un resultado dado [111] (véase las definiciones en la Sección 4.4.1)

c. **Soporte de herramientas**

El objetivo de este criterio es evaluar si la métrica cuenta con una herramienta o algoritmo que facilite su cálculo. Si la métrica dispone de una herramienta que permita su cálculo automatizado o semiautomático, se clasifica como *automatizada* y se registra el nombre y la referencia de la herramienta. En caso contrario, se clasifica como *manual*.

d. **Resultado de la medición**

El resultado de la medición es un número o categoría—junto con cualquier otra información relevante disponible—que se atribuye a una magnitud, es decir, a la propiedad de un fenómeno, cuerpo o sustancia [112]. En este trabajo, el resultado obtenido al realizar una medición es un número o categoría que se asigna a un atributo de calidad de un servicio cloud (definición adaptada de [57]). Por ejemplo, la velocidad de transmisión de datos o el ancho de banda se expresa normalmente en Mbps (Megabits por segundo), y el uso del disco como medida de tamaño en GB (Gigabytes) [113]. El propósito de este criterio es comprender el tipo de resultado de medición que produce la métrica. En particular, una métrica puede medir un atributo de calidad de manera cualitativa, cuantitativa o híbrida:

- Las evaluaciones `cuantitativas` se ocupan de evaluar los atributos mediante valores continuos. Por ejemplo, un atributo como la comunalidad de la función, que mide el promedio de comunalidad de cada característica funcional a partir de la especificación de requisitos de software (*Software Requirements Specification* - (SRS)) en el mismo dominio y definida en un SaaS objetivo [114], se puede medir con valores continuos entre 0 y 1.
- Las evaluaciones `cualitativas` son aquellas que indican cualidades o categorías cualitativas. Por ejemplo, un atributo como la flexibilidad de un servicio cloud que califica la capacidad de agregar o eliminar características predefinidas del servicio para personalizarlo [115], se puede medir como Alto, Medio o Bajo.
- Las evaluaciones `híbridas` combinan evaluaciones cualitativas y cuantitativas [111]. Por ejemplo, la distancia del centro de datos mide la distancia entre la ubicación del centro de datos y la ubicación de servicio esperada y, a continuación, califica al proveedor utilizando la suma de las distancias [116].

e. **Fases del ciclo de vida**

El objetivo de este criterio es identificar las fases del ciclo de vida del servicio cloud en las que se puede aplicar la métrica, pudiendo abarcar una o varias fases. Cada métrica se clasifica según las fases del ciclo de vida propuestas por Schneider et al. [117]:

- `Requisitos`: métricas que se aplican a documentos o especificaciones que describen la decisión del cliente sobre si se utiliza un servicio cloud y en qué medida.

- `Adquisición`: métricas que soportan la evaluación de proveedores y servicios cloud. Por ejemplo, métricas utilizadas para determinar la estimación de la demanda o para evaluar las oportunidades y riesgos asociados.
- `Desarrollo`: métricas aplicables a los artefactos cloud durante actividades como la especificación de requisitos del servicio, el diseño de la arquitectura, la programación, la configuración de *hardware*, las pruebas, la implementación, el despliegue, la orquestación, la gestión de versiones y la integración.
- `Integración`: métricas que evalúan el esfuerzo y la eficiencia con que el cliente configura, conecta y prueba el servicio cloud dentro de un entorno existente. Por ejemplo, la calidad técnica de las conexiones configuradas o la estabilidad operacional del servicio integrado.
- `Operación`: métricas utilizadas para evaluar el servicio en tiempo de ejecución con fines de administración, por ejemplo, para monitorizar la QoS en tiempo real. Incluye métricas para supervisar el cumplimiento de contratos entre consumidores y proveedores y detectar violaciones de SLA. También incluye métricas utilizadas por los proveedores de servicios para el mantenimiento, evolución, soporte y facturación, así como por los consumidores para monitorizar y evaluar el uso del servicio.
- `Retiro`: métricas que garantizan la finalización segura y organizada del servicio o cuando el cliente cambia a otro proveedor. Las métricas del proveedor pueden incluir métricas para asegurar el cumplimiento de la normativa de protección de datos en relación con la conservación o eliminación de datos de los clientes.

f. **Artefacto cloud**

Un artefacto es una representación de servicio cloud que se utiliza para aplicar la métrica y realizar una evaluación de calidad. Estos artefactos pueden corresponder a un estado temprano, intermedio o finalizado del servicio cloud. Cada métrica se clasifica según el artefacto que mide, considerando los siguientes tipos:

- `Especificación del servicio cloud`: documento o modelo que representa la interfaz funcional, enumerando las operaciones y atributos a los que los clientes pueden acceder. Por ejemplo, documentos de requisitos que definen características específicas que se deben cumplir o SLA que reflejan compromisos definidos en términos de QoS entre proveedor y cliente.
- `Arquitectura cloud`: modelo que engloba todos los elementos de un entorno cloud, representando cómo se conectan todos los componentes y capacidades necesarios para ofrecer una plataforma de ejecución de aplicaciones. Por ejemplo, los diagramas de software que encapsulan capas de abstracción de aplicaciones (p. ej., middleware) y los diagramas de red que muestran la conexión de capas de red (p. ej., enrutadores) pueden considerarse instancias de artefactos de arquitectura cloud.

- `Servicio cloud`: el servicio real utilizado en un entorno cloud, incluyendo representaciones de servicio reales. Como ejemplos de artefactos de servicio cloud se pueden considerar: el código fuente desplegado y entregado exitosamente como un servicio versionado, las configuraciones de red o máquinas virtuales (VM) como configuración de *hardware* y *software* para habilitar los servicios de red y la virtualización.

Una métrica puede aplicarse a más de un artefacto. En estos casos, se pueden extraer diferentes operacionalizaciones, es decir, distintas funciones de medición que muestran cómo se calculan las métricas en cada artefacto.

g. **Tipo de servicio**

El objetivo de este criterio es evaluar el tipo de servicio que evalúa la métrica. Una métrica se puede aplicar a más de un tipo de servicio. En este trabajo, se consideran los siguientes tres tipos de servicio: IaaS, PaaS, SaaS (véanse las definiciones en la Sección 2.1.2).

h. **Punto de vista de los *stakeholders***

El punto de vista de los *stakeholders* indica quiénes pueden medir y utilizar la métrica. Cada métrica se ha clasificado de acuerdo con los roles propuestos en la Sección 2.1.1, incluyendo proveedores, consumidores, facilitadores, desarrolladores y usuarios finales.

i. **Procedimiento de validación**

El objetivo de este criterio es obtener el procedimiento utilizado para validar la métrica. Una métrica debe ser validada tanto teóricamente como empíricamente. La *validación teórica* garantiza que la métrica mide el atributo que se supone debe medir y permite confirmar que la medición no viola propiedades necesarias de los elementos de medición. La *validación empírica* proporciona evidencia sobre la utilidad de las métricas en la práctica.

Cada métrica se clasifica según el método o la estrategia de investigación utilizada. Si la métrica está validada teóricamente, se clasifica según el tipo de enfoque utilizado: enfoque basado en la propiedad, enfoque basado en la teoría de la medición, u otro. Si la métrica es validada empíricamente, se clasifica según el método utilizado: experimento controlado, estudio de caso, o encuesta. De lo contrario, la métrica se clasifica como no validada.

En cuanto a las *validaciones teóricas* empleadas para validar las métricas, se considera:

- `Enfoque basado en la propiedad`: permite demostrar que una métrica satisface las propiedades que caracterizan un concepto [118], como tamaño, complejidad o acoplamiento.

- `Enfoque basado en la teoría de la medición`: más riguroso que los enfoques basados en propiedades, ya que prescribe teorías y condiciones para el modelado y la definición de métricas [119, 120]. Esta teoría proporciona una interpretación empírica de los números (de las métricas de software) mediante un hipotético sistema relacional empírico.

En cuanto a las *validaciones empíricas*, se distinguen los siguientes métodos:

- `Estudio de caso`: estudio observacional en el que los datos se recopilan con un propósito específico a lo largo del estudio.
- `Encuesta`: investigación que se realiza en retrospectiva cuando la métrica ha estado en uso durante un cierto período de tiempo.
- `Experimento controlado`: estudio formal, riguroso y controlado, que proporciona un alto nivel de control y es útil a la hora de validar métricas de software. Por ejemplo, puede utilizarse para validar la eficacia de un conjunto de métricas de diseño en la predicción de la usabilidad de los servicios cloud.

La tarea de extracción de datos se facilitó mediante el diseño de una plantilla (libro de cálculo de Excel), acompañada de una guía que detalla cada uno de los criterios. Una de las hojas de la plantilla se utilizó para recopilar información sobre los estudios primarios seleccionados, mientras que otra se destinó a registrar los datos de las métricas del servicio cloud (p. ej., nombre de la métrica, descripción, atributo medido, característica), así como información común, entre la que se incluye: nombres de los autores, título del artículo, detalles de la publicación, nombre de la biblioteca digital, tipo de publicación y número de citas.

Se realizó una prueba piloto con una muestra de artículos para comprobar la comprensibilidad y la corrección de los criterios de extracción de datos y de las hojas de cálculo. Para evaluar la fiabilidad del proceso de extracción de datos, tres investigadores independientes llevaron a cabo una segunda iteración de la clasificación sobre una muestra aleatoria de 30 artículos. El nivel de concordancia entre ellos se evaluó mediante el índice *Kappa de Fleiss* [108].

4.2.2.4. Método de síntesis de datos

Para sintetizar los datos extraídos de los estudios primarios y responder a las preguntas de investigación, se emplearon dos métodos cualitativos de síntesis: síntesis narrativa y análisis temático.

La *síntesis narrativa* presenta los resultados de la revisión sistemática en forma textual. Para apoyar este proceso, se utilizó la herramienta Mendeley [121] para almacenar todos los artículos y anotar las evidencias empleadas en la clasificación de las métricas según los criterios definidos en la sección 4.2.2.3. Asimismo, se analizó el número de artículos encontrados por año en cada fuente bibliográfica y las frecuencias de los estudios clasificados según cada criterio.

Por su parte, el *análisis temático* consiste en identificar y codificar los temas principales o recurrentes en los estudios primarios, agrupando los resultados bajo estos ejes temáticos [29]. Este método se utilizó de manera complementaria a la síntesis narrativa para responder a las preguntas de investigación. Específicamente, se siguieron los pasos que se detallan a continuación:

1. `Lectura de los artículos e identificación de segmentos relevantes`: Se revisó todo el texto asociado a los estudios primarios, identificando los segmentos específicos de texto relevantes para las preguntas de investigación, tales como: características de QoS, atributos de calidad, métricas, fases del ciclo de vida del servicio cloud, artefactos cloud, tipo de servicio, punto de vista de los *stakeholders* y procedimiento de validación. Este proceso permitió formarnos una idea inicial para el análisis.

2. `Generación de códigos iniciales`: Se definieron códigos iniciales para las características de QoS, los atributos de calidad, métricas (junto con su información asociada), fases del ciclo de vida del servicio cloud, artefactos cloud, tipo de servicio, el punto de vista de los *stakeholders* y procedimiento de validación. Estos códigos se asignaron a los segmentos del texto identificados como relevantes. En algunos casos fue necesario revisar nuevamente los estudios para garantizar la correcta codificación.

3. `Búsqueda de temas`: A partir de los códigos iniciales generados, se combinaron aquellos que estaban relacionados para formar temas potenciales.

4. `Análisis, depuración y definición de temas`: Algunos temas pudieron definirse a priori por estar alineados con marcos existentes, como las características de calidad de la ISO/IEC 25010 [1], o los puntos de vista de los *stakeholders* y el tipo de servicio según NIST SP 800-145 [30] y NIST SP 800-146 [31], así como las fases del ciclo de vida del servicio cloud y el procedimiento de validación definidos en trabajos previos. Otros temas emergieron del análisis de los estudios primarios, tales como atributos de calidad, métricas y su información asociada, artefactos cloud y soporte de herramientas.

5. `Revisión y refinamiento de los temas`: Se compararon los atributos de calidad, métricas (incluyendo métodos y funciones de medición, el resultado y la unidad de medición) y artefactos cloud identificados en el paso anterior, con el fin de detectar posibles fusiones o eliminaciones. Por ejemplo, *disponibilidad* y *capacidad de servicio* se fusionaron debido a que compartían el mismo propósito, mientras que *rendimiento* se eliminó por ser redundante, ya que representa el número de solicitudes de servicio atendidas durante el tiempo total de servicio, utilizando la misma función de medición [122, 123, 124]).

   Inicialmente, las métricas se categorizaron considerando las características de QoS, los atributos de calidad y el nombre de la métrica. Posteriormente, se agruparon identificando sinónimos y homónimos mediante el análisis del nombre de la métrica

y su definición, el método de medición, la función o funciones de medición y la unidad de medida (p. ej., milisegundos, requests/min).

Se consideró que dos métricas eran sinónimas cuando tenían nombres distintos pero la misma definición o propósito. Por ejemplo, *disponibilidad* [125] y *capacidad de servicio* [126] emplean las mismas métricas base (es decir, tiempo de actividad y tiempo de inactividad) para medir la disponibilidad del servicio. Del mismo modo, AVAL-CQ [10] y *Uptime* [11] calculan el porcentaje de disponibilidad del servicio a partir de la métrica de tiempo de actividad, como el tiempo total de un período operativo. Se debe tener en cuenta que una métrica puede tener varias operacionalizaciones (diferentes funciones de medición que se pueden utilizar para medir el mismo atributo de calidad). También se consideró que las métricas que tenían el mismo nombre, pero una definición (propósito) diferente eran homónimas. Por ejemplo, Lu et al. [122] mide la fiabilidad usando la tasa de fallas, mientras que en Guerout et al. [127] se mide usando la tasa de éxito.

Este paso permitió definir temas claros y bien estructurados para los atributos de calidad, las métricas (incluyendo su nombre, definición, método y funciones de medición, resultado y unidad de medida), los tipos de servicio, las fases del ciclo de vida del servicio cloud y los artefactos cloud.

Además, se elaboraron diagramas de burbujas [27] con el fin de representar las frecuencias correspondientes a las combinaciones entre los distintos criterios analizados. Un diagrama de burbujas consiste en dos diagramas de dispersión en los ejes x y y, donde, en cada intersección de categorías, se ubica una burbuja cuyo tamaño es proporcional a la frecuencia observada. Este método de síntesis resulta especialmente eficaz para proporcionar una visión rápida y global del tema de investigación, así como para identificar patrones, concentraciones de métricas y relaciones relevantes entre los criterios.

4.2.3. Resultados de la SLR

La fase de obtención de resultados implica la transformación de los datos sintetizados durante la ejecución del protocolo en conocimiento estructurado y evaluado, orientado a responder fundamentadamente a las preguntas de investigación. Para ello se realizaron las siguientes actividades:

1. `Análisis de amenazas`: Se realiza el análisis de las posibles amenazas a la validez de los resultados obtenidos durante el proceso de la SLR.

2. `Reporte de resultados`: Se consolidan e informan los datos provenientes de la extracción, se identifican dimensiones temáticas, patrones, tendencias, similitudes y diferencias relevantes, y brechas en la literatura, estableciendo relaciones entre los hallazgos y las métricas de QoS en servicios cloud.

3. `Análisis comparativo`: Se contrastan enfoques y técnicas empleadas en los estudios primarios. Este análisis permite evaluar la consistencia de resultados, solidez de las definiciones y criterios de calidad.

4. `Definición de metamodelo`: Se define un metamodelo que abstrae los elementos comunes, diferencias, estructuras y enfoques relevantes para crear un marco conceptual unificado.

5. `Refinamiento de la taxonomía`: Como resultado de la mejor comprensión en base al metamodelo y la integración conceptual de los resultados se ajusta la taxonomía definida.

Los resultados no solo reportan la evidencia recopilada, sino que también aportan a la interpretación crítica que orienta el desarrollo de la taxonomía propuesta y fundamenta las decisiones metodológicas adoptadas en este estudio.

4.2.4. Evaluación del protocolo

Como sugiere Brereton et al. [106], el protocolo fue evaluado interna y externamente antes de su ejecución. Esta validación preliminar tuvo como objetivo garantizar que los procedimientos definidos —criterios de selección, evaluación de calidad, extracción y síntesis de datos— fueran completos, no ambiguos y replicables por diferentes investigadores. La evaluación incluyó dos mecanismos complementarios:

- **Validación cruzada**, realizada por el equipo de investigación. Para comprobar la aplicabilidad y consistencia del protocolo, se realizaron iteraciones independientes tanto en la *selección de estudios* como en la *extracción de datos*. Primero, varios investigadores del equipo aplicaron de manera independiente los criterios de inclusión y exclusión sobre una muestra representativa de estudios, comparando posteriormente los resultados. Las discrepancias fueron analizadas y resueltas por consenso, garantizando una interpretación uniforme del protocolo. De la misma manera, la plantilla y el procedimiento de extracción de datos fueron validados mediante una segunda revisión independiente realizada por investigadores del equipo. Esta evaluación permitió detectar ambigüedades, mejorar la definición de algunos campos y asegurar que la información relevante pudiera ser capturada de manera consistente.

- **Revisión externa** Se solicitó a un experto con experiencia en la conducción de la SLR que proporcionará su opinión acerca de la *selección de estudios*, *evaluación de calidad*, y la *extracción de datos*. Sus observaciones permitieron refinar la redacción de algunos criterios, mejorar la precisión de los términos operativos y ajustar la estrategia de extracción de datos, asegurando la generación de resultados relevantes. A partir de esta revisión, se introdujeron ajustes menores que incrementaron la claridad y aplicabilidad del protocolo.

4.3. Análisis bibliométrico del conjunto de artículos primarios

Los resultados de la selección de estudios primarios se presentan sobre la base del análisis de metadatos y la evaluación de la calidad. El 8 de noviembre de 2018 se ejecutó

la cadena de búsqueda en las cuatro bibliotecas digitales consideradas, recuperándose un total de 4333 artículos.

A continuación, se combinaron las listas de publicaciones procedentes de cada base de datos y se eliminaron 147 publicaciones duplicadas. Estos duplicados incluían estudios presentes en más de una fuente, así como estudios publicados tanto en actas de congresos como en revistas. En el primer caso, se seleccionó la publicación una sola vez adoptando el siguiente orden de prioridad: (1) IEEE Xplore, (2) ACM Digital Library, (3) ScienceDirect y (4) SpringerLink. Este orden se justifica porque las bibliotecas digitales especializadas en ciencias de la computación y la ingeniería de software—es decir, IEEE Xplore y ACM Digital Library—ofrecen una cobertura más específica del dominio [128]. ScienceDirect y SpringerLink, al tratarse de bibliotecas multidisciplinarias, se consideraron posteriormente. En el segundo caso, cuando existían múltiples versiones de un mismo estudio (por ejemplo, en congreso y posteriormente en revista), se seleccionó únicamente la versión más completa.

En una fase posterior, se verificó que todas las ediciones de las actas de cada congreso y de las revistas que se muestran en la Tabla 4.5 estuvieran indexadas en al menos una de las bibliotecas digitales. Como no se encontró el *Journal of Cloud Computing: Advances, Systems and Applications* (JoCCASA), se realizó una búsqueda manual en dicha revista, lo que permitió obtener 44 artículos candidatos adicionales.

Tabla 4.5: Lista de revistas y conferencias relevantes.

Tipo	**Nombre**
REVISTAS	IEEE Transactions on Cloud Computing (TCC) Journal of Cloud Computing ACM Transactions on the Web (TWEB) IEEE Internet Computing World Wide Web Journal ACM Transactions on Internet Technology (TOIT) ACM Transactions on Software Engineering and Methodology (TOSEM) Information and Software Technology (IST) Journal of Cloud Computing: Advances, Systems and Applications (JoCCASA) Journal of Systems and Software (JSS) IEEE Software IEEE Transactions on Software Engineering (TSE) Software Quality Journal (SQJ) Empirical Software Engineering (EMSE)
CONFERENCIAS	International Conference on Software Engineering (ICSE) International Conference on Service Oriented Computing (ICSOC) International Conference on Web Services (ICWS) International Conference on Services Computing (SCC) International World Wide Web Conference (WWW) Web Information Systems Engineering (WISE) International Symposium on Empirical Software Engineering and Measurement (ESEM) International Conference on Evaluation and Assessment in Software Engineering (EASE)

El título, el resumen y las palabras clave de cada publicación fueron revisados por dos personas, aplicando el conjunto de criterios de inclusión y exclusión. Cuando fue necesario, también se revisaron la introducción y las conclusiones. Esta fase condujo a la eliminación de 3947 publicaciones. Los principales motivos de exclusión fueron los siguientes: estudios que proponían métricas no relacionadas con la calidad interna o externa, estudios que

describían atributos de calidad sin métricas para medirlos y estudios que proponían o utilizaban métricas sin proporcionar una explicación de cómo medirlas.

Tras esta selección inicial, dos revisores leyeron en su totalidad los artículos restantes para aplicar los criterios de inclusión y exclusión de forma exhaustiva. Esto dio como resultado un conjunto de 195 artículos (194 de las bibliotecas digitales y uno obtenido mediante la búsqueda manual), que fueron almacenados en la herramienta Mendeley [121].

Finalmente, se realizó la evaluación de la calidad de estas publicaciones utilizando los ítems de calidad presentados en la Tabla 4.2. Todas las discrepancias entre los revisores fueron discutidas hasta alcanzar un consenso. La fiabilidad de los resultados se garantizó considerando solo los estudios con una calidad aceptable, es decir, aquellos que obtuvieron más de 5 puntos (equivalente al 50 % de la puntuación máxima). En la Tabla 4.6 se muestran los resultados obtenidos en cada ítem de calidad.

Tabla 4.6: Lista de verificación para la evaluación de la calidad.

		No. de artículos			
Ítem de calidad	**Puntuación**	**S**	**P**	**N**	**Porcentaje (%)**
Pregunta 1	69,0	52	34	2	97,73
Pregunta 2	49,0	32	33	23	75,00
Pregunta 3	81,5	75	11	2	98,86
Pregunta 4	82,5	77	11	0	100,00
Pregunta 5	82,0	76	12	0	100,00
Pregunta 6	28,5	6	45	37	57,95
Pregunta 7	28,5	19	19	50	43,18
Pregunta 8	75,0	64	22	2	97,73
Pregunta 9	24,5	14	21	53	39,77
Pregunta 10	10,5	7	7	74	15,91
Total				88	100,00

Los detalles de la evaluación de cada estudio primario están disponibles en el sitio web asociado a este trabajo: `https://bit.ly/frameworkCloudQuality` . Se excluyeron cuatro artículos que no cumplían con el umbral mínimo de calidad establecido. Al analizar los ítems de calidad, se identificaron varias limitaciones en los estudios seleccionados. En primer lugar, se echa en falta mayor discusión sobre las limitaciones de los estudios ($P10 = 15.91\,\%$). En segundo lugar, se evidenció una falta de documentación sobre ideas y lecciones aprendidas ($P9 = 39.77\,\%$). En tercer lugar, se observó una escasez de evidencia sobre la aplicación de métricas en la práctica ($P7 = 47.18\,\%$). Por último, las métricas reportadas rara vez fueron validadas ($P6 = 57.95\,\%$).

En general, la transferencia de conocimientos relacionados con problemas o resultados negativos es poco frecuente, lo que limita la evaluación retrospectiva de ideas y lecciones aprendidas, ya que la mayoría de los estudios tiende a centrarse en resaltar únicamente las contribuciones y resultados alineados con los objetivos de investigación. Asimismo, las escasas validaciones empíricas de las métricas ponen de manifiesto la necesidad de contar con mayores niveles de evidencia acerca de su utilidad real. En consecuencia, estas carencias en la validación han incidido directamente en la reducida aplicación de las métricas en la práctica.

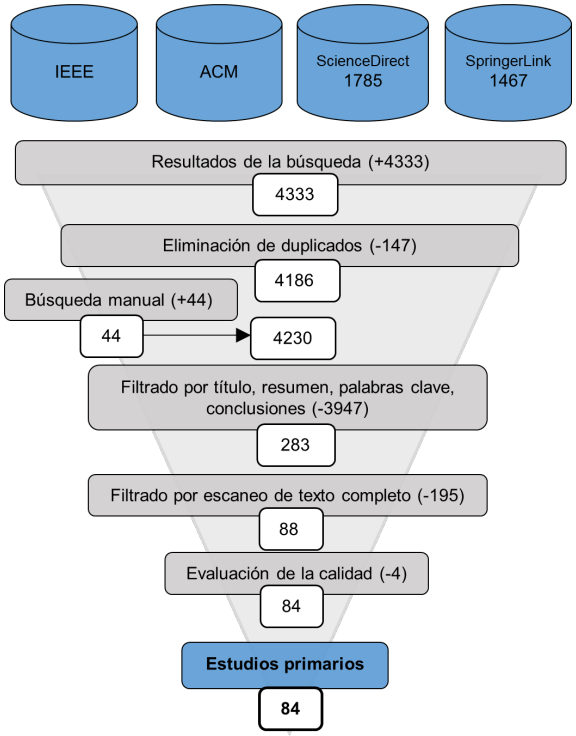


Figura 4.2: Resultado del proceso de selección de estudios primarios [2].

La Figura 4.2 muestra el resultado del proceso de selección de artículos, que condujo a un total de **84** estudios primarios.

Tabla 4.7: Relevancia de los estudios primarios.

Tipo	**Puntos**	**No. Artículos**	**Porcentaje (%)**
Impacto de la publicación	10	24	27.27
	5	24	27.27
	0	40	45.46
	Total	88	100.00
Citas	10	29	32.95
	5	29	32.95
	2	28	31.80
	0	2	2.30
	Total	88	100.00

En la Tabla 4.7 se presenta la calidad de los sitios de publicación de los trabajos seleccionados. Los resultados muestran que más del 55 % de los artículos fueron publicados en *venues* considerados muy relevantes o relevantes. En cuanto al número de citas, se observa que la mayoría de los artículos presentan un volumen alto de citaciones (>= 50 citas) o un nivel medio (entre 10 y 49 citas), representando cada categoría el 32.95 % de

las publicaciones. Además, 30 artículos no registran citas.

La Figura 4.3 muestra el número de estudios acumulados por fuente y año, lo cual puede considerarse un indicador de la creciente importancia que ha adquirido este tema en los últimos años. No es posible extraer conclusiones definitivas sobre cuáles son las mejores fuentes bibliográficas, dado que los artículos que aparecen en varias bases de datos solo se han contabilizado una vez. Sin embargo, la mayoría de los estudios relevantes sobre métricas de calidad para servicios cloud se encontraron en IEEE Xplore (55 artículos), lo que representa el 65 % de los estudios primarios seleccionados. En cuanto al tipo de publicación, 54 artículos (60 %) fueron presentados en conferencias, 6 artículos (7 %) en *workshops* y 28 artículos (33 %) se publicaron en revistas. La lista completa de los estudios seleccionados se incluye en el Apéndice A.

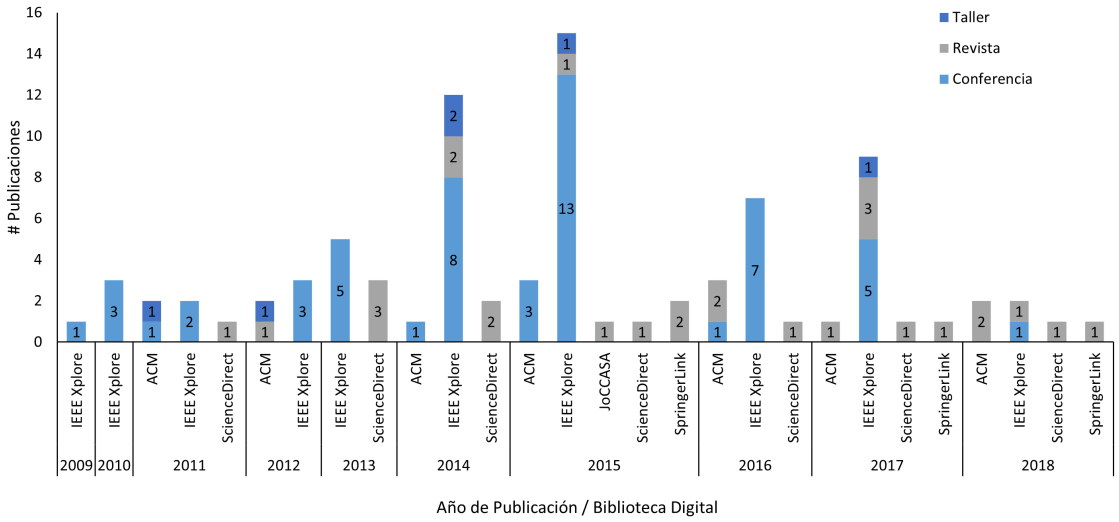


Figura 4.3: Número de estudios primarios por año y fuente [2].

4.4. Construcción de la taxonomía

En esta sección se detalla el proceso de creación y refinamiento de la taxonomía de métricas para los servicios cloud. El proceso inicia con la definición de un metamodelo que establece la línea base conceptual sobre la cual se estructuran los distintos elementos de la taxonomía, conformando la línea base. Estos conceptos están directamente vinculados con los criterios de extracción de datos definidos en la sección 4.2.2.3, lo que garantiza la coherencia metodológica entre las fases del estudio.

Una vez definido el metamodelo, se procedió a la aplicación de métodos de síntesis cualitativa, específicamente, síntesis narrativa y análisis temático. Estas técnicas permitieron integrar de manera sistemática la información extraída de los estudios primarios seleccionados, identificar patrones recurrentes y establecer relaciones significativas entre

los conceptos. El resultado fue la construcción de una taxonomía robusta y fundamentada en la evidencia, que sirve como instrumento de apoyo tanto para investigadores como para profesionales en la evaluación de servicios cloud.

4.4.1. Metamodelo para la taxonomía

Con el fin de organizar los datos recopilados y responder adecuadamente a las preguntas de investigación, se definió un metamodelo que orienta la construcción de la taxonomía de métricas para servicios cloud (véase la Figura 4.4). Su utilidad radica en la capacidad de descomponer y organizar jerárquicamente los elementos de la taxonomía, sobre la base de principios y notaciones de modelado independientes de la tecnología. Esto facilita la representación de los conceptos asociados a las métricas de calidad y promueve un entendimiento compartido, así como una comunicación más clara entre investigadores y profesionales.

La Figura 4.4 muestra el metamodelo propuesto, que estructura la taxonomía en términos de conceptos de calidad, el dominio de la computación en la nube y las relaciones entre ellos. Los conceptos centrales son: características, atributos, métricas y operacionalizaciones. Las relaciones entre ellos permiten capturar la lógica de la taxonomía y asegurar la consistencia en la evaluación de la calidad de los servicios cloud.

Este enfoque permite capturar la complejidad del dominio y, al mismo tiempo, garantizar la trazabilidad y la consistencia en la evaluación de la calidad de los servicios cloud. A continuación, se detallan los principales conceptos representados en el metamodelo, definidos como metaclases, junto con sus propósitos y relaciones.

- `CloudServiceQualityModel`: Agrupa las características de calidad relevantes para el dominio de los servicios cloud y establece las relaciones entre ellas. La calidad de un servicio cloud se entiende como el grado en que este satisface las necesidades explícitas e implícitas de los *stakeholders*, proporcionando así valor (adaptado de ISO/IEC 25010 [1]).

- `Característica`: Según la ISO/IEC 25010 [1], es una propiedad de calidad de alto nivel que se descompone en subcaracterísticas y, posteriormente, en atributos específicos del dominio cloud. Un ejemplo es la eficiencia de desempeño, que refleja el rendimiento del servicio en función del uso de recursos bajo determinadas condiciones.

- `Atributo`: Propiedad física o abstracta, medible en una entidad del servicio cloud [110] (p. ej., red, máquina virtual o contenedor), que puede evaluarse mediante métricas. Un ejemplo es la capacidad de memoria en una máquina física o virtual.

- `Métrica`: Define una escala de medición (nominal, ordinal, intervalo, razón o absoluta) y un enfoque de medición (método o función) que especifica cómo debe realizarse la medición [57]. Ejemplos incluyen el tamaño de la memoria RAM o el tiempo de respuesta a una solicitud. Cada métrica puede tener una o más operacionalizaciones. Se distinguen tres tipos:

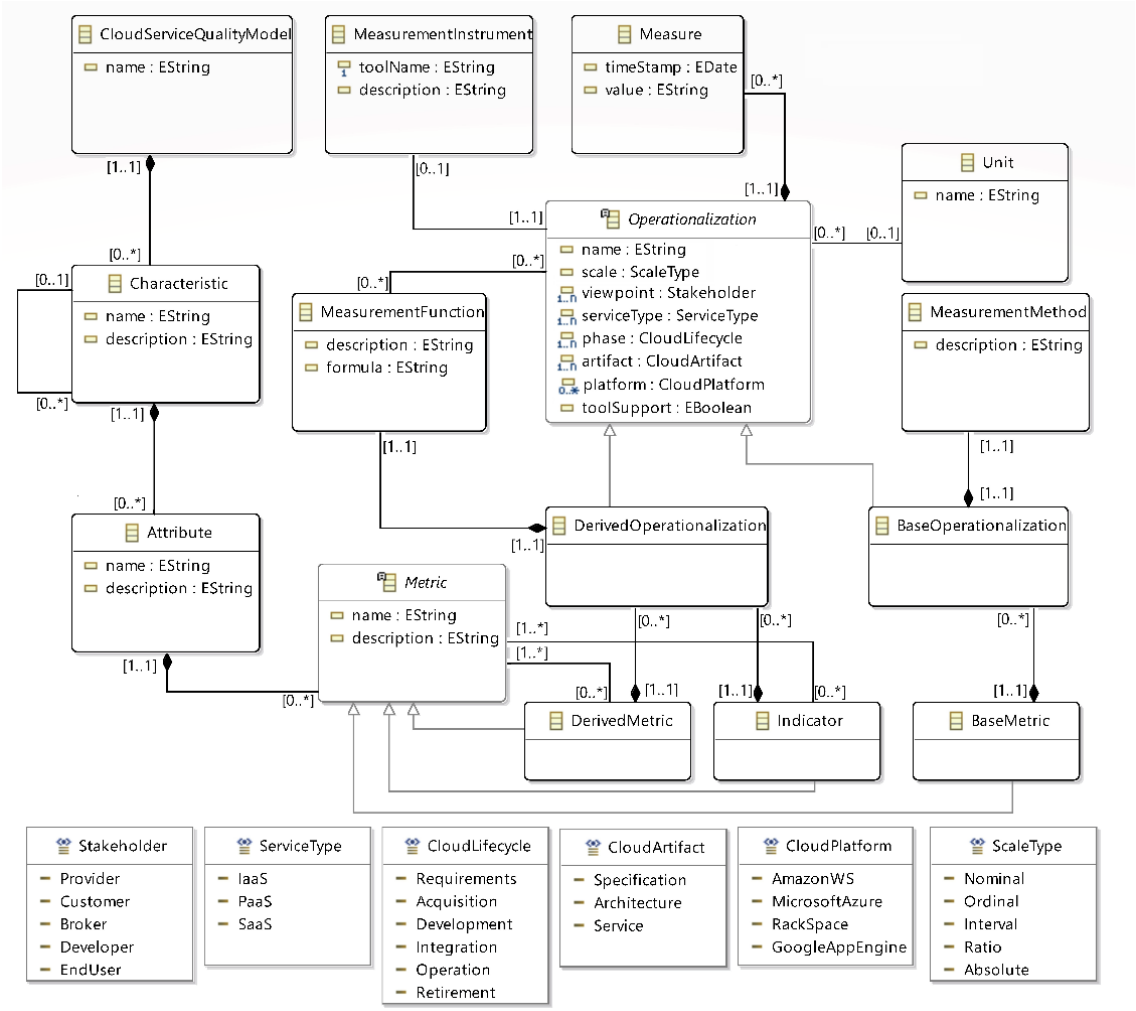


Figura 4.4: Metamodelo de calidad para servicios cloud [2].

- `Métrica base`: Independiente de otras métricas, utiliza un método de medición [111] (p. ej., tamaño de memoria, tiempo de solicitud).
- `Métrica derivada`: Se obtiene a partir de métricas base o derivadas, mediante una función de medición [111] (p. ej., tiempo de respuesta calculado como suma del tiempo de solicitud y de respuesta).
- `Indicador`: Métrica de nivel superior derivada de otras mediante un modelo de análisis [110]. Un ejemplo es la capacidad de servicio, calculada como una suma ponderada de CPU, memoria, almacenamiento y redes.

▪ `Operacionalización`: Define las distintas formas de calcular una métrica. Establece la correspondencia entre la definición genérica de la métrica y el artefacto, plataforma

o entorno cloud en el que se mide. Puede ser base o derivada:

  - `Operacionalización base`: Describe cómo calcular una métrica base mediante un método de medición.
  - `Operacionalización derivada`: Describe cómo calcular una métrica derivada o un indicador mediante una función de medición.

- `Método de medición`: Secuencia lógica de operaciones empleadas para cuantificar un atributo mediante una métrica base [110]. Un ejemplo de método de medición son los pasos para medir el tamaño de almacenamiento en una máquina virtual.

- `Función de medición`: Algoritmo que combina dos o más métricas para cuantificar un atributo (adaptado de ISO/IEC 15939 [111]). Un caso típico es el cálculo del tiempo de respuesta como la diferencia entre la solicitud y la respuesta del servicio cloud.

- `Instrumento de medición`: Herramienta que apoya la aplicación de un método de medición. Ejemplos son *Amazon CloudWatch* para la operacionalización de métricas en AWS, o JCatascopia [113], que mediante su parámetro *rx_bytes* mide los bytes recibidos por un servicio.

- `Medida`: Valor resultante de la aplicación de un método de medición a un atributo de calidad [128], como el número de MB para la memoria o los milisegundos para la latencia.

- `Unidad`: Unidad de medida asociada a operacionalizaciones con escalas de intervalo o razón [109] (p. ej., MB para tamaño de memoria o ms para tiempo de respuesta).

Asimismo, el metamodelo incorpora conceptos propios de la computación en la nube, como los *stakeholders*, tipos de servicio, fases del ciclo de vida del servicio cloud y artefactos cloud (véase la Sección 4.2.2.3). Estas relaciones permiten comprender cómo interactúan los distintos elementos.

El núcleo de la taxonomía es el concepto de métrica y sus posibles operacionalizaciones. La metaclase *Operationalization* incluye atributos que describen propiedades clave, tales como:

- `name`: nombre de la operacionalización.

- `scale`: tipo de escala utilizada.

- `viewpoint`: *stakeholder* al que va dirigida la métrica (proveedor, cliente, facilitador, desarrollador).

- `serviceType`: tipo de servicio (SaaS, PaaS, IaaS) aplicable.

- `phase`: fase del ciclo de vida del servicio cloud en la que se aplica.

- `artifact`: artefacto cloud medido.
- `platform`: plataforma cloud en la que se aplica.
- `toolSupport`: herramienta que facilita el cálculo de la operacionalización.

La taxonomía integra conceptos relevantes para comprender la calidad de los servicios cloud, combinando las características de alto nivel propuestas en la ISO/IEC 25010 [1] con métricas específicas del dominio. Como se analizó en la Sección 3.1, la mayoría de evaluaciones de calidad se centran en características aisladas, lo que da lugar a soluciones fragmentadas. Por ello, resulta necesario un enfoque holístico que integre características, atributos y métricas de calidad del dominio cloud.

En definitiva, esta taxonomía constituye un primer paso hacia la concreción de estándares abstractos en mediciones específicas para la nube. Además, servirá como base para la extracción y clasificación de datos en los estudios primarios, permitiendo consolidar y unificar el conocimiento existente sobre métricas de calidad en servicios cloud.

4.4.2. Extracción de datos

La extracción de datos se llevó a cabo entre marzo a septiembre de 2019. Este proceso permitió responder a las preguntas de investigación y obtener la taxonomía de métricas de calidad para servicios cloud.

Las métricas recopiladas de los estudios primarios fueron categorizadas y organizadas según los criterios definidos en la Sección 4.2.2.3. De esta manera se obtuvo una versión inicial de la taxonomía, representada en una hoja de cálculo, que incluyó un total de 579 métricas extraídas de 84 estudios primarios. Dada la amplitud y diversidad de las métricas recuperadas, se decidió emplear el análisis temático como método de síntesis cualitativa, lo que permitió refinar la taxonomía en cinco pasos (véase la Sección 4.2.2.4):

1. Revisión completa de todos los artículos.
2. Identificación de códigos y citas iniciales (fragmentos de texto) relacionados con las preguntas de investigación, incluyendo atributos de calidad medidos e información detallada sobre las métricas (nombre, definición, métodos y funciones de medición, etc.).
3. Revisión independiente de cada cita por dos investigadores, quienes propusieron categorías (temas) de nivel superior para describir características de QoS, atributos de calidad, métricas, fases del ciclo de vida, artefactos cloud medidos, tipo de servicio, punto de vista de los *stakeholders* y procedimientos de validación.
4. Discusión y consenso sobre los temas identificados: Los temas se definieron con un nombre corto y una descripción. En esta etapa también se analizaron los códigos para reducir solapamientos, se revisaron sinónimos y homónimos, y se consolidaron métricas equivalentes, obteniendo así una primera versión de la taxonomía de métricas.

5. Revisión y refinamiento de los temas y consolidación de la taxonomía, agrupando atributos y métricas en categorías de nivel superior para proporcionar una visión más amplia. En algunos casos fue necesario volver a revisar los artículos originales. Además, se analizó la frecuencia de aparición de cada tema en los estudios primarios.

El resultado de este procedimiento fue la distribución final de los atributos de calidad, métricas y operacionalizaciones, organizada de acuerdo con las ocho características de calidad del estándar ISO/IEC 25010 [1]. La Figura 4.5 resume la estructura refinada de la taxonomía: el anillo interior agrupa 240 atributos, el anillo intermedio contiene 406 métricas únicas (del conjunto inicial de 579) y el anillo externo representa las 470 operacionalizaciones, correspondientes a distintos métodos y funciones de medición. De estas métricas, 156 están asociadas a atributos internos y 314 a atributos externos de los servicios cloud.

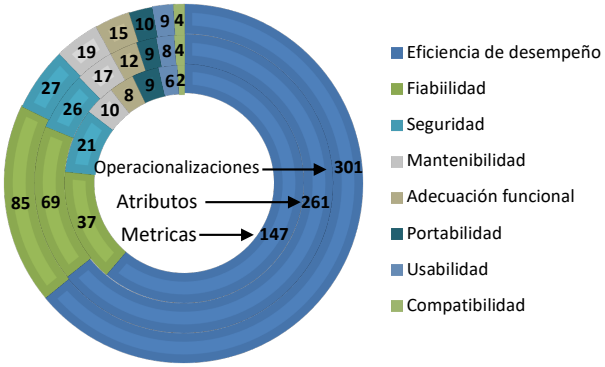


Figura 4.5: Taxonomía refinada de métricas para servicios cloud [2].

Los resultados evidencian que la característica con mayor número de atributos, métricas y operacionalizaciones es la *eficiencia de desempeño*, que concentra más del 50 % del total. Esta característica también registró el mayor número de métricas eliminadas y agregadas durante el proceso de refinamiento, debido a la presencia de métricas sinónimas, como *delay*, *latency*, *throughput* y *response time*. En las siguientes subsecciones se presenta un análisis detallado de estas métricas en relación con las preguntas de investigación.

La Tabla 4.8 presenta la contribución de los estudios primarios a cada característica de calidad del estándar ISO/IEC 25010. Se observa que la *eficiencia de desempeño* concentra la mayor contribución (69 artículos), mientras que la *compatibilidad* registra la menor (3 artículos). Cabe destacar que algunos estudios contribuyeron a más de una característica, lo que coincide con la distribución observada en atributos, métricas y operacionalizaciones.

La Figura 4.6 muestra la cobertura de las características de calidad en los estudios primarios. De los 84 trabajos, 47 se centraron en una sola característica, ninguno abordó las ocho propuestas por la ISO/IEC 25010 [1] y únicamente siete cubrieron al menos la mitad (cuatro) de ellas. La combinación más frecuente fue *eficiencia de desempeño*

y *fiabilidad*, identificada en 32 artículos (p. ej., [8, 129, 130]). Entre los trabajos con un enfoque más holístico destacan el de Singh y Chana [126], quienes proponen una técnica de aprovisionamiento de recursos basada en métricas de QoS con 35 métricas que cubren casi todas las características de calidad (excepto la compatibilidad), y el de Garg et al. [131], cuyo marco SMICloud utiliza 23 métricas que cubren la mayoría de las características (excepto la seguridad).

Tabla 4.8: Contribución de los estudios primarios a las características de calidad.

Característica de calidad	**No. Artículos**	**% Artículos**
Eficiencia de desempeño	69	82
Fiabilidad	42	50
Seguridad	15	18
Adecuación funcional	8	10
Mantenibilidad	7	8
Portabilidad	7	8
Usabilidad	6	7
Compatibilidad	3	4

La Figura 4.9 presenta la taxonomía resultante de métricas de QoS para servicios cloud. Las características de calidad definidas en la ISO/IEC 25010 [1] se desglosan en atributos específicos, asociados a sus respectivas métricas y operacionalizaciones. Entre los estudios centrados en una sola característica, la *eficiencia de desempeño* resultó predominante, ya que factores como el rendimiento, la capacidad y el uso eficiente de los recursos impactan directamente en la percepción de la calidad del servicio desde la perspectiva de los *stakeholders*. Además, esta característica es fundamental para la definición del SLA y la supervisión de los servicios cloud.

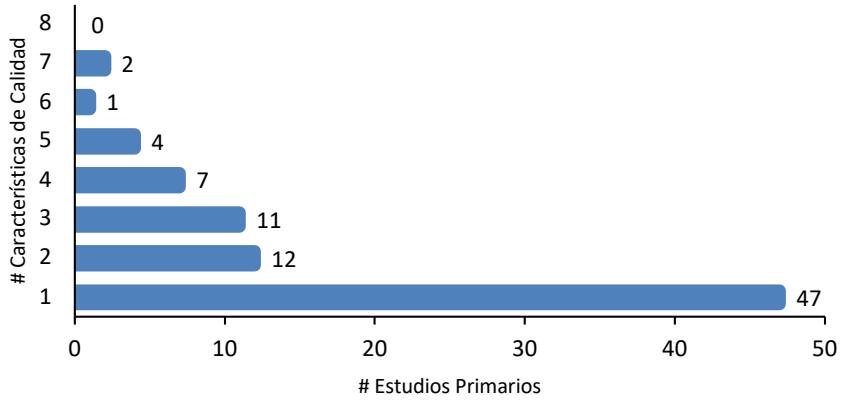


Figura 4.6: Distribución de los estudios primarios según las características de calidad [2].

Con el fin de fomentar la transparencia y la reutilización de los resultados, la clasificación está disponible en el sitio web que acompaña a este trabajo (`https://bit.ly/`

`frameworkCloudQuality`), donde se puede explorar de manera interactiva la distribución de atributos, métricas y operacionalizaciones para cada característica de calidad.

Finalmente, la Tabla 4.9 resume los resultados obtenidos para cada criterio, que se discuten en detalle a continuación en relación con las preguntas de investigación.

Tabla 4.9: Resultados de los criterios de extracción de datos.

Subpreguntas de investigación	**Posibles opciones**	**No. Oper.**	**Porcentaje (%)**
PI 1. Características de QoS	Eficiencia de desempeño	301	64
	Fiabilidad	85	18
	Seguridad	27	6
	Mantenibilidad	19	4
	Adecuación funcional	15	3
	Portabilidad	10	2
	Usabilidad	9	2
	Compatibilidad	4	1
PI 2. Tipo de métrica	Base	108	23
	Derivado	347	74
	Indicador	15	3
PI 3. Soporte de herramientas	Manual	402	85
	Automatizado	70	15
PI 4. Resultado de la medición	Cualitativo	12	2,6
	Cuantitativo	455	96,8
	Híbrido	3	0,6
PI 5. Fase del ciclo de vida del servicio cloud	Requisitos	53	8
	Adquisición	174	26
	Desarrollo	58	9
	Integración	8	1
	Operación	373	56
	Retiro	8	1
PI 6. Artefacto cloud medido	Especificación del servicio	12	2
	Arquitectura cloud	58	11
	Servicio cloud	464	87
PI 7. Tipo de servicio	SaaS	210	35
	Paas	82	14
	IaaS	304	51
PI 8. Punto de vista de los *stakeholders*	Proveedor	374	45
	Consumidor	283	34
	Facilitador	46	6
	Desarrollador	57	7
	Usuario final	69	8
PI 9. Procedimiento de validación			
Validación teórica	Enfoque axiomático	0	0
	Enfoque basado en la teoría de la medición	0	0
Validación empírica	Experimento controlado	39	8
	Estudio de casos	8	2
	Encuesta	0	0
Sin validación	Prueba de concepto	254	52
	Sin validación	188	39

4.4.2.1. Características de QoS

Cada métrica se clasificó de acuerdo con las características de calidad del estándar ISO/IEC 25010. Recopilamos el nombre y la descripción de la métrica, el nombre del atributo de calidad que se midió y alguna información adicional que nos ayudó a comprender el contexto en el que se utilizó la métrica.

1) Eficiencia de desempeño

Los resultados muestran una alta concentración de métricas relacionadas con esta característica, con 301 operacionalizaciones de métricas. Esto puede explicarse por el hecho de que es una de las características esenciales de los servicios cloud —servicio medido— establece que los sistemas cloud deben controlar y optimizar automáticamente los recursos utilizados aprovechando una capacidad de medición adecuada al tipo de servicio (p. ej., almacenamiento, procesamiento, ancho de banda) [78].

Esto significa que el uso de recursos debe ser monitorizado, controlado e informado tanto por el proveedor como por el consumidor del servicio cloud que se utiliza. Por tanto, es muy importante controlar el rendimiento de los servicios cloud y garantizar un tiempo de respuesta adecuado para el cliente.

Un análisis de las métricas recopiladas muestra que la mayoría de ellas, 135 operacionalizaciones de métricas, miden el ***comportamiento temporal*** (*time behaviour*). Esto se debe a que, en la computación en la nube, el rendimiento se expresa por la velocidad con la que se completa una solicitud al servicio cloud. Esa velocidad generalmente se representa en términos de la cantidad de tiempo requerido desde el envío de una solicitud hasta la recepción de la respuesta (es decir, tiempo de respuesta) y la cantidad de solicitudes exitosas dentro de un intervalo de tiempo dado (es decir, rendimiento). El tiempo de respuesta se puede dividir en tiempo de ejecución, que representa el tiempo necesario para procesar una solicitud en el lado del servidor; y latencia, que representa el tiempo necesario para la entrega unidireccional de un mensaje.

Algunos ejemplos de métricas empleadas para medir el comportamiento temporal que recopilamos incluyen el tiempo de respuesta en términos del tiempo necesario para ejecutar una solicitud de servicio [130, 131, 132, 127] o en términos del tiempo de ejecución de una máquina virtual que pertenece a un servicio cloud [127], y la latencia en términos del tiempo que transcurre entre una solicitud y la respuesta correspondiente [124, 125, 127].

También hay una gran cantidad de métricas que miden la ***utilización de recursos*** (*resource utilization*), en total 77 operacionalizaciones de métricas, pero la mayoría de ellas son métricas de bajo nivel, que, en la mayoría de los casos, son proporcionadas por las herramientas de monitorización proporcionadas por las plataformas cloud. Estas métricas permiten medir el nivel de uso de recursos (p. ej., CPU, memoria, disco) o el porcentaje de recursos actualmente ocupados [133, 113]. De estas métricas, 27 fueron propuestas con el propósito de medir la sostenibilidad en términos de consumo de energía. Por ejemplo, en Dou et al. [134], el consumo total de energía de una infraestructura cloud se mide en términos del consumo de energía de los servidores y del consumo de energía requerido para la comunicación entre esos servidores.

También se encontraron 56 métricas que miden la ***capacidad***. La capacidad es el grado

en que los límites máximos de un recurso de un servicio cloud (p. ej., almacenamiento, redes) satisfacen los requisitos esperados. Por ejemplo, Souza et al.[132] proponen varias métricas con las que medir la capacidad del servidor, la capacidad del dispositivo de almacenamiento y la capacidad de la red.

Esta característica también incluye varios atributos y métricas de calidad relacionados con la ***elasticidad*** (29 operacionalizaciones de métricas) y la ***escalabilidad*** (11 operacionalizaciones de métricas), que son factores esenciales que afectan la calidad de los servicios cloud. La escalabilidad y la eficiencia están asociadas con la elasticidad, pero tienen significados distintos, aunque en algunos casos se usan indistintamente.

NIST SP 800-145 define la elasticidad rápida como capacidades que se aprovisionan y liberan rápidamente y elásticamente, en algunos casos de forma automática [30]. Herbst et al. [135] la definen como la capacidad de adaptar el cambio en la carga de trabajo agregando o eliminando recursos automáticamente. Finalmente, en este estudio se adoptó la definición propuesta por Al-Dhuraibi et al. [83] definen la elasticidad además de la escalabilidad. La elasticidad es una automatización del concepto de escalabilidad (auto-scaling), sin embargo, tiene el objetivo de optimizar los recursos lo mejor y más rápido posible en un momento determinado. La escalabilidad se define como la capacidad del servicio cloud para sostener cargas de trabajo crecientes mediante el uso de recursos adicionales y es independiente del tiempo, mientras que la eficiencia describe cómo los recursos cloud se pueden utilizar de manera eficiente a medida que aumentan o disminuyen en escala [83].

La escalabilidad también sustenta la característica esencial de agrupación de recursos de la computación en la nube, durante la cual los recursos informáticos de los proveedores se agrupan para servir a múltiples consumidores utilizando el modelo multitenencia [30]. Su medición se basa en tres métodos básicos, que son replicación, redimensionamiento y migración [133], que están respaldados por las métricas recopiladas. La replicación emplea la escalabilidad horizontal para agregar o eliminar instancias de recursos (p. ej., máquinas virtuales, *Kubernetes*). El redimensionamiento o cambio de tamaño utiliza la escalabilidad vertical para aumentar o disminuir los recursos informáticos (p. ej., tamaño de la RAM, número de núcleos de CPU). Por ejemplo, Souza et al. [132] propuso que la métrica del número de servidores virtuales en el grupo de recursos debería usarse para medir la escalabilidad horizontal del servidor y que las métricas del tamaño de RAM y la cantidad de CPU deberían usarse para medir la escalabilidad vertical de los servidores. Las migraciones transfieren recursos de un servidor a otro (p. ej., contenedores, máquinas virtuales).

Por otro lado, la eficiencia refleja qué tan bien se utilizan los recursos cloud a medida que aumentan o disminuyen. Se puede medir en términos de tiempo de escalamiento de recursos y cantidad de recursos en relación con el coste de optimización. Por ejemplo, Hu et al. [133] utilizaron el tiempo de escalamiento para medir el aumento y la reducción de escalamiento, utilizando el tiempo medio de retraso (*delay*) en el aumento y la reducción de recursos. Por lo tanto, un consumidor puede medir el retraso que se produce al aprovisionar y desaprovisionar un recurso determinado. Estos autores también midieron el coste medio de escalamiento hacia arriba o hacia abajo utilizando el precio por uso de la máquina virtual. Esto significa que cuanto mayor sea la elasticidad, mayor será la eficiencia.

También se encontraron varias métricas que se utilizan para controlar los tres estados de

aprovisionamiento: sobreaprovisionamiento, subaprovisionamiento y justo a la necesidad (just-in-needed). El estado de sobreaprovisionamiento genera costes adicionales e innecesarios al alquilar recursos cloud en exceso. El subaprovisionamiento se produce cuando los recursos proporcionados son inferiores a los necesarios, es decir, insuficientes, lo que puede provocar una degradación del rendimiento y el no cumplimiento de las cláusulas del SLA. Por ejemplo, Hu et al. [133] definieron el sobreaprovisionamiento como el número promedio de recursos sobreaprovisionados y el subaprovisionamiento como el número promedio de recursos insuficientes. En ambos casos, el SLA y la QoS no son óptimos. Por último, el aprovisionamiento justo a la necesidad denota un recurso equilibrado cercano a la demanda real, lo que significa que se gestiona la carga de trabajo y se garantiza la QoS.

2) Fiabilidad

Esta fue la segunda característica de QoS con mayor número de métricas, 85 operacionalizaciones de métricas, lo que representa el 18 % del total de métricas. La fiabilidad siempre ha sido una preocupación importante en los sistemas distribuidos. Proporcionar servicios confiables y de alta disponibilidad en computación en la nube es esencial para mantener la confianza y la satisfacción del cliente y prevenir pérdidas de ingresos. Sin embargo, garantizar la fiabilidad en entornos cloud es un desafío. Un servicio cloud puede constar de cientos de microservicios, cada uno de los cuales se ejecuta en su propio clúster distribuido y contiene sus propias dependencias múltiples, lo que aumenta la cantidad de componentes individuales que pueden fallar. Además, en el entorno de computación en la nube se intercalan varios tipos de fallos, como fallos por desbordamiento, fallos de tiempo de espera, fallos de recursos faltantes, fallos de red, fallos de *hardware*, fallos de *software* y fallos de bases de datos. Esto puede explicar la gran cantidad de métricas propuestas en la literatura para evaluar la fiabilidad.

De todos los atributos de calidad considerados, los más importantes para medir la fiabilidad son la disponibilidad, la recuperabilidad, la tolerancia a fallos y la confiabilidad del servicio. Todos estos atributos son críticos para asegurar la continuidad del servicio cloud.

Un análisis de las métricas recopiladas muestra que la mayoría de ellas (es decir, 34 operacionalizaciones de métricas) miden la disponibilidad. La ***disponibilidad*** es la capacidad de un servicio cloud para ser operativo y accesible cuando sea necesario su uso. Los consumidores valoran un servicio de alta disponibilidad, y debe formar parte de la negociación del SLA con los proveedores de cloud. Por ejemplo, Zheng et al. [10] proponen emplear la métrica AVAL para medir la disponibilidad como el porcentaje del tiempo de actividad de un servicio cloud durante un intervalo de tiempo, donde un mayor tiempo de actividad representa menos interrupciones. En Garg et al. [131], la disponibilidad se mide en términos de accesibilidad al servicio utilizando la relación entre el tiempo de no disponibilidad sobre el tiempo total de servicio. En Rizvi et al. [130], la disponibilidad se mide en términos del estado funcional del servicio en un intervalo de tiempo utilizando el tiempo de actividad (accesibilidad del servicio) y el tiempo de inactividad (servicio en reparación).

También existe una buena cantidad de métricas que miden diferentes atributos rela-

cionados con la recuperabilidad, es decir, 25 operacionalizaciones de métricas. La ***recuperabilidad*** es el grado en que, ante una interrupción o un fallo, un servicio cloud puede recuperar los datos directamente afectados y restablecer el estado deseado del sistema. Este es un factor importante para los consumidores cloud, ya que cualquier pérdida de datos podría ser devastadora para la empresa. Ejemplos de métricas incluyen la cobertura de recuperación de fallos, que se mide como la ratio de fallos solucionados sobre el total de fallos [8]; la durabilidad, que se mide como la probabilidad de pérdida de datos [130]; y la robustez, que se mide como la probabilidad de que el servicio se vea afectado por el fallo de un componente cloud [127].

En particular, la robustez de un servicio IaaS se puede medir por la cantidad de máquinas virtuales afectadas por una falla del host; es decir, maximizar la robustez significa minimizar la cantidad de máquinas virtuales afectadas [127]. De manera similar, la robustez de los servicios multi-nube (*multicloud*) o multitenencia (*multi-tenant*) se puede medir por la cantidad de servicios cloud o la cantidad de inquilinos afectados por un fallo del servicio. Sin embargo, ninguna de las métricas recopiladas aborda la robustez en estos escenarios.

También se encontraron 12 métricas que miden la ***tolerancia a fallos***. Esta es la capacidad del servicio cloud de permanecer accesible y funcionando cuando ocurren anomalías [127]. Pueden ocurrir anomalías como resultado de errores en la máquina física, en la red o en el *software*. Algunos ejemplos de métricas incluyen métricas tradicionales utilizadas para medir la tolerancia a fallos en sistemas de software, como el tiempo medio entre fallas (MTBF), que se mide como el tiempo entre fallos de servicio consecutivos, el tiempo medio de fallo (MTTF), el tiempo medio de reparación (MTTR) y la cobertura de tolerancia a fallos (CFT).

Existe una métrica que mide la madurez. La ***madurez*** es el grado en que un servicio cloud satisface las necesidades de confiabilidad requeridas en condiciones de operación normal. Un servicio es confiable cuando realiza funciones específicas bajo condiciones específicas durante un período de tiempo específico [1]. Estas operacionalizaciones de métricas incluyen la tasa de éxito, que mide la finalización exitosa del trabajo aceptado por el servicio cloud [136].

Finalmente, varios estudios primarios propusieron la fiabilidad del servicio cloud como un indicador que combina otros atributos de calidad relacionados con la fiabilidad (p. ej., disponibilidad, tolerancia a fallos, corrección funcional y recuperabilidad). El propósito de estas 13 operacionalizaciones es asegurar el funcionamiento continuo del servicio cloud sin fallos. Todos los indicadores emplearon el método típico de suma ponderada. Por ejemplo, en Lee et al. [8] se ponderó la tolerancia a fallos, la recuperación de fallos y la corrección funcional para obtener una medida de fiabilidad, mientras que Zheng et al. [10] ponderó el estado libre de fallos del almacenamiento cloud (fallos de *hardware*, fallos de *software* e interrupciones de la red) para proporcionar una medida de la confiabilidad del servicio. Los criterios de decisión de ambos enfoques establecen que cuanto mayor sea el resultado, mayor será la fiabilidad. Esta combinación de medidas permite ponderar de manera diferente las preocupaciones de los *stakeholders*, que también pueden diferir en lo relacionado con el establecimiento de prioridades.

3) Seguridad

Esta característica representó 27 operacionalizaciones de métricas, lo que representa solo el 6 % del total de métricas. Los aspectos de la seguridad cloud incluyen seguridad a nivel de aplicación, seguridad de multitenencia donde diferentes inquilinos comparten una infraestructura común, seguridad de proveedor a inquilino, seguridad de la información y conformidad con los requisitos de seguridad, que incluye pero no se limita a autenticación, autorización, confidencialidad, gestión de identidad, integridad, auditoría, monitorización de seguridad, respuesta a incidentes y gestión de políticas de seguridad [32]. Por lo tanto, uno de los principales desafíos al adoptar servicios cloud está relacionado principalmente con el cumplimiento de la seguridad y la fiabilidad de la infraestructura cloud al ejecutar las aplicaciones de los clientes y almacenar sus datos en la nube. En definitiva, se considera que el número de métricas en la literatura que abordan estos desafíos es bajo.

Algunos de los atributos de calidad que los autores de los estudios primarios seleccionados consideraron más importantes son la integridad, la confidencialidad, la responsabilidad, la autenticidad y la privacidad. Todos estos atributos son críticos para garantizar la seguridad de los servicios cloud y la privacidad de los datos almacenados en la nube. Algunos autores también utilizaron la seguridad del servicio como un indicador que integra otras métricas bases o derivadas de seguridad.

Al analizar las métricas recopiladas, se observó que la mayoría de ellas miden la integridad ,es decir, 8 operacionalizaciones de métricas. La ***integridad*** es la capacidad de impedir el acceso no autorizado o la modificación de aplicaciones o datos. Este es un atributo de calidad importante al adoptar soluciones cloud (es decir, mover datos del negocio a la nube) y debe verificarse a nivel de datos y de cómputo. Por ejemplo, Singh y Chana [126] midieron la integridad de un sistema utilizando la probabilidad de ataque de una amenaza y la probabilidad de repeler un ataque en un tiempo determinado. Mientras Manuel [136] midió la integridad de los datos como una garantía de su preservación y comprobación de que los recursos almacenan los datos correctamente.

Se encontraron 6 métricas que miden atributos relacionados con la confidencialidad. La ***confidencialidad*** es la capacidad de garantizar que los datos sean accesibles sólo para quienes estén autorizados a hacerlo. Esto es importante porque muchas máquinas virtuales en entornos de nube pueden coexistir en la misma máquina física (*co-located*) y pueden adoptar diferentes mecanismos de protección de seguridad. Ejemplos de estas métricas incluyen la cantidad de falsas alarmas monitorizadas y la cantidad de respuestas ineficaces del servicio ante los problemas identificados por la seguridad, como debilidades de control del servicio cloud [126].

Se encontraron 2 métricas que miden la ***responsabilidad***, que se refiere a la capacidad de rastrear las acciones del servicio cloud. En entornos de nube, los controles empleados para auditar y monitorizar la seguridad son esenciales debido a su principio intrínseco de compartir recursos. Algunos ejemplos de estas métricas incluyen los peligros de seguridad identificados proactivamente y los recursos de datos peligrosos que residen en las soluciones [126]. Además, la rápida evolución de la nube y la integración de nuevas tecnologías (p. ej., internet de las cosas (IoT)) hacen aún más importante la existencia de nuevas métricas que permitan mejorar la transparencia en la operación del servicio.

Se encontraron 2 métricas que miden atributos relacionados con la ***autenticidad***, que se refiere a la capacidad de identificar un tema o recurso y demostrar que es lo que dice ser. Esto es fundamental en entornos cloud para abordar el control de acceso a los recursos, que incluyen la infraestructura, aplicaciones y datos, porque generalmente estos se comparten y distribuyen. Algunos ejemplos de estas métricas incluyen la autenticidad para determinar si los usuarios tienen el privilegio de emplear ese servicio cloud y el control de acceso para indicar el estado de acceso de los usuarios a dicho servicio [137].

Algunos estudios primarios han propuesto la seguridad del servicio cloud como un indicador que agrega otras métricas relacionadas con la seguridad (p. ej., integridad, privacidad, disponibilidad, autenticidad). El propósito de estas nueve operacionalizaciones de métricas es garantizar la satisfacción de los requisitos de seguridad. Por ejemplo, Zheng et al. [10] utilizó SECY como función de distribución acumulativa hasta que se produjera la primera violación de seguridad. Esta es la garantía de que los servicios cloud estén libres de virus, intrusiones, *spyware*, ataques y otras vulnerabilidades de seguridad que puedan ponerlos en riesgo.

Las implicaciones de gestionar adecuadamente la privacidad de los datos impactan directamente en la reputación y credibilidad de los proveedores cloud que cumplen con las leyes, normativas y regulaciones (p. ej., protección de datos). Esto establece la seguridad como uno de los desafíos más relevantes a la hora de adoptar los servicios cloud. Por lo tanto, se necesitan más métricas para abordar todo el ciclo de vida de la seguridad de la información y controlar el uso de los datos sensibles. La seguridad también debe evaluarse durante todo el ciclo de vida del servicio cloud para garantizar los requisitos de seguridad desde el diseño y la arquitectura iniciales de los servicios cloud.

Existen otros desafíos relacionados con la seguridad y privacidad de los datos en entornos cloud en los que el acceso amplio a la red [30] es una de las características esenciales. Esto promueve el acceso a los servicios cloud desde diferentes ubicaciones y con diferentes dispositivos (p. ej., teléfonos móviles, estaciones de trabajo), lo que incrementa los aspectos que son necesarios considerar a la hora de garantizar la seguridad. Proteger y controlar el acceso a los datos, se convierte en un verdadero desafío técnico que requiere más métricas y herramientas que lo respalden.

Otro desafío está relacionado con garantizar la seguridad en entornos de multi-nube (es decir, un enfoque de nube compuesto por más de un servicio de nube, proporcionado por más de un proveedor público o privado), en el que el nivel de seguridad general del servicio cloud será una agregación de las propiedades de seguridad de los servicios cloud vinculados. Existe una necesidad particular de métricas que midan la responsabilidad en entornos de multi-nube para permitir una verificación completa de los recursos físicos y virtuales.

4) Mantenibilidad

Esta característica contiene 19 operacionalizaciones de métricas, que representan solo el 4 % del total de métricas. Esto no parece suficiente si se tiene en cuenta que la mayor parte del coste del software (incluido el coste de los servicios cloud) se deriva no de su desarrollo inicial, sino de su mantenimiento continuo. Varias personas diferentes tendrán

que realizar cambios en el servicio cloud a lo largo del tiempo, tanto para mantener su comportamiento actual como para adaptar y/o evolucionar el servicio para hacer frente a nuevos requisitos, y deberían poder realizar estos cambios de manera efectiva y eficiente y tener un mecanismo con el cual comprobarlo.

Algunos de los atributos de calidad considerados más relevantes, al medir la mantenibilidad son la modificabilidad, la reutilización y la modularidad.

Un análisis de las métricas recopiladas muestra que la mayoría de ellas, es decir, 8 operacionalizaciones de métricas, miden la reutilización. La ***reutilización*** se refiere al grado en que un activo se puede utilizar en más de un servicio cloud o para crear otros activos. En entornos cloud, un activo puede ser un componente o un artefacto del servicio cloud. Algunos ejemplos de activos cloud son las máquinas virtuales a nivel de IaaS o los contenedores (*Docker*) y microservicios a nivel de SaaS. Los servicios cloud también se desarrollan comúnmente utilizando metodologías como Agile y DevOps, en las que la reutilización es un factor clave.

Algunos ejemplos de métricas de reutilización para SaaS incluyen:

I. Cobertura de variabilidad, que mide cuántos de los puntos de variación incluidos en el dominio se realizan realmente en el servicio cloud, y

II. Funcionalidad común, que mide una cantidad promedio de la funcionalidad común de cada característica funcional definida en un servicio objetivo [8, 114].

También se encontraron otros indicadores con los que medir la reutilización que combinan diferentes métricas, por ejemplo, puntos funcionales en común (FC), puntos no funcionales en común (NFC) y cobertura de variabilidad (CV) [8]. Todos estos indicadores utilizan el método de la suma ponderada como función de medición.

Hay otras métricas que miden atributos relacionados con la modificabilidad (es decir, 7 operacionalizaciones de métricas). La ***modificabilidad*** es la capacidad de modificar un servicio cloud de manera efectiva y eficiente sin introducir defectos ni degradar la calidad del servicio. Estas métricas se pueden aplicar a artefactos cloud obtenidos en diferentes fases del ciclo de vida (p. ej., diseño, documentación, casos de prueba) que implementan un cambio particular. Algunos ejemplos de estas métricas incluyen fuerza flexible, que mide la facilidad o dificultad con la que se puede cambiar un servicio cloud como respuesta a una solicitud de un cliente [126], y calificar la capacidad de agregar o eliminar funciones predefinidas de un servicio cloud para adaptarse a las preferencias de los usuarios [115].

Se encontró una métrica para medir la ***analizabilidad***, esta mide la cantidad de defectos por servicio cloud [126]. La analizabilidad es la capacidad de evaluar el impacto que tienen los cambios en uno o más de sus recursos en un servicio cloud, de diagnosticar deficiencias o causas de fallo.

Se encontró una métrica para medir la modularidad, ésta mide la proporción del número de elementos sin dependencias externas [114]. La ***modularidad*** es el grado en que un servicio cloud puede estar compuesto por varios componentes, de tal manera que los cambios en un componente tengan un impacto mínimo o representativo en otros componentes.

Por último, se encontró una métrica relacionada con la ***mantenibilidad*** en sí, ésta es el costo de mantenimiento, que calcula la cantidad de capacidad que cada *edge*/nodo necesita para ser restaurado y no puede exceder el presupuesto [138].

Además, un aspecto a considerar es el alto grado de granularidad (varios componentes) de los servicios cloud y su creciente tasa de entrega de versiones de corta duración (diarias o varias por día) significan que las métricas encontradas en la literatura no son suficientes para ayudar a establecer un control adecuado sobre la mantenibilidad de los servicios cloud.

5) Adecuación funcional

Esta característica representó 15 operacionalizaciones de métricas, lo que representa solo el 3 % del total de métricas. La adecuación funcional es el grado en que las funciones realizan o no las funciones básicas. Se centra particularmente en tres tipos de adecuación funcional [1]:

I. *Completitud funcional*, que es el grado en que el conjunto de funciones cubre todas las tareas especificadas y los objetivos del usuario;

II. *Corrección funcional*, que es el grado en que un servicio cloud proporciona los resultados correctos con el grado de precisión requerido, y

III. *Pertinencia funcional*, que es el grado en que las funciones del servicio cloud facilitan el logro de tareas y objetivos específicos.

Los problemas comunes relacionados con la completitud y corrección funcional son principalmente la recuperación de datos incorrecta e ineficaz o las ediciones de datos ineficaces que pueden originar resultados inadecuados. Los problemas asociados a la pertinencia funcional están relacionados con servicios que no son lo suficientemente flexibles para cumplir con los requisitos del negocio o los objetivos de nivel de servicio. La relevancia se debe a las necesidades de los *stakeholders*, que se especifican en los SLAs y deben ser satisfechas por el servicio cloud.

Los resultados indicaron que los atributos de calidad que se consideran más importantes a la hora de medir la adecuación funcional son: corrección, idoneidad, pertinencia y completitud.

Al analizar las métricas recopiladas, se observó que la mayoría de ellas (es decir, 12 operacionalizaciones de métricas) miden la *corrección funcional*. Algunos ejemplos de métricas utilizadas para medirla incluyen la precisión del servicio a nivel de IaaS, que se mide como la frecuencia de incumplimiento del SLA prometido en términos de unidades de cómputo, red y almacenamiento [131]. Por el contrario, la precisión del servicio a nivel de SaaS, que es el grado en que la respuesta a una petición del usuario es correcta [114], y como el ratio entre el servicio cloud que se espera y el que se observa [126].

Finalmente, se encontró una métrica para cada uno de los siguientes atributos: idoneidad, pertinencia y completitud funcional. Estas métricas son la idoneidad de las características no esenciales, que miden el grado en el cual los requisitos de un cliente se cumplen por el proveedor cloud [139]; el valor de pertinencia, que mide el valor de la unidad de

servicio esperado por un usuario y el valor de la unidad de servicio enviado de vuelta a ese usuario [127], y la completitud, que mide el total de servicios cloud existentes sobre el total de servicios cloud solicitados [126].

6) Portabilidad

Esta característica contiene 10 operacionalizaciones de métricas, que representan solo el 2 % del total de métricas. En el contexto de la computación en la nube, la portabilidad se refiere a la capacidad de los clientes para mover y adaptar adecuadamente sus aplicaciones y datos entre sus propios sistemas y sus servicios cloud, y entre los servicios cloud de diferentes proveedores y, potencialmente, diferentes modelos de implementación cloud [139]. Esta es una de las características más cuestionadas debido al alto grado de dependencia del proveedor y el desafío aquí es, por tanto, resolver problemas relacionados con el movimiento de datos o servicios entre proveedores cloud. La portabilidad es relevante en la computación en la nube, ya que los clientes están interesados en evitar quedarse bloqueados (*lock-in*) a un proveedor cuando eligen utilizar servicios cloud.

Se encontraron 6 métricas relacionadas con la instalabilidad. La *instalabilidad* tiene un concepto diferente en la computación en la nube, al ser un modelo de pago por uso, representa el esfuerzo necesario para desplegar o preparar un servicio cloud y que esté listo para ser usado. Por lo tanto, puede definirse como el grado en que un servicio cloud puede migrarse (portarse) desde un sistema de origen a un sistema de destino, o una aplicación de un proveedor cloud a otro, o entre el sistema de un consumidor cloud y un servicio de nube. Como ejemplos, Baranwal y Vidyarthi [116] propusieron métricas dependientes de la aplicación que miden el grado en que el servicio cloud es portable a otras plataformas y lo hicieron empleando el soporte de plataformas, una medida de virtualización y el soporte de sistemas operativos. Esto es útil para identificar la instalabilidad de la aplicación y puede diferir entre las aplicaciones desplegadas en la nube.

Se encontraron 4 métricas relacionadas con la adaptabilidad. La *adaptabilidad* es el grado en que un servicio cloud puede adaptarse de manera efectiva y eficiente al *hardware*, *software* u otros entornos operativos de usos diferentes o en evolución. Tenga en cuenta que la adaptabilidad es diferente de la instalabilidad (portable), ya que la primera representa la capacidad de ajustar o cambiar el servicio según la solicitud del cliente o los cambios tecnológicos. Por ejemplo, en Nadanam y Rajmohan [114] se mide la adaptabilidad en términos de cobertura de variabilidad y completitud del conjunto de variantes para determinar la efectividad de adaptar los servicios al uso de cada aplicación basada en servicios. Garg et al. [131] lo mide como el tiempo necesario para adaptar el servicio cloud a los cambios o actualizarlo a un nivel superior.

Los requisitos esenciales para los consumidores son soluciones personalizadas e independientes del proveedor. Sin embargo, el costo de la personalización suele estar ligado a soluciones propietarias en entornos de nube pública, lo que limita el acceso a servicios cloud entre múltiples proveedores. En definitiva, mover o migrar los datos o servicios de un proveedor a otro representa un desafío. Además, la falta de portabilidad entre los proveedores cloud es un aspecto relevante que resalta la dependencia de los proveedores, lo que hace que los consumidores cloud sean vulnerables a las limitaciones de precios y a la

calidad del servicio proporcionado por una plataforma de nube específica.

7) Usabilidad

Esta característica representó 9 operacionalizaciones de métricas, lo que representa solo el 2 % del total de métricas. Este es un factor importante que se emplea para medir la simplicidad del uso de un servicio cloud y su rápida adopción. Según Stanton et al. [140] los usuarios finales pueden personalizar varios atributos de una solución cloud para toda la organización. Por ejemplo, los servicios cloud deberían ser accesibles para clientes con una variedad de necesidades (accesibilidad), deberían permitir a los consumidores cambiar su interfaz de usuario para adaptarla a sus necesidades (personalización) y deberían garantizar la facilidad de uso mediante la implementación de identidades de acceso múltiple, debido a que los consumidores no son conscientes de la cantidad de pasos de autenticación/autorización que deben seguir para acceder a sus aplicaciones cloud (gestión de identidades). Mientras tanto, los clientes deben tener una sensación de control sobre la funcionalidad del servicio cloud (control) y deben tener propiedad sobre los datos que almacenan en los servicios cloud que utilizan (propiedad de los datos). Esta característica comprende además otros atributos como la capacidad de aprendizaje, que representa el esfuerzo necesario para aprender a utilizar un servicio cloud; y la protección de errores del usuario, que representa el grado en que un servicio cloud protege a los usuarios de cometer errores.

A pesar de la relevancia de la usabilidad en entornos de nube, nuestros resultados muestran que se han definido y utilizado muy pocas métricas para medir ciertos atributos de usabilidad (es decir, operatividad, capacidad de aprendizaje y comprensibilidad). Se encontraron dos operacionalizaciones de métricas para cada uno de estos atributos. Por ejemplo, Garg et al. [131] mide el tiempo promedio que se tomaron los usuarios anteriores para operar, aprender y comprender un servicio cloud; Singh y Chana [126] miden la proporción de operaciones exitosas y el tiempo necesario para aprender el servicio cloud; y, Nadanam y Rajmohan [114] miden la comprensibilidad del servicio como la ratio entre la cantidad de campos que tienen una legibilidad inaceptable y el número total de campos.

Se encontró una métrica que mide la estética de la interfaz de usuario, USAB-CQ, que mide qué tan fácil, eficiente y agradable es usar la interfaz de un servicio cloud, o evalúa la facilidad de invocación cuando la funcionalidad del servicio cloud se muestra en la imagen. forma de API [10]. La estética de la interfaz de usuario es el grado en que la interfaz de un servicio cloud proporciona una interacción agradable y satisfactoria para el usuario.

Sin embargo, la principal preocupación es que la usabilidad generalmente se mide subjetivamente mediante el empleo de medidas cualitativas. Por ejemplo, Ezenwoke et al. [115] lo mide como la facilidad con la que un usuario puede utilizar, aprender, operar, instalar y entender un servicio cloud. Además, los esfuerzos actuales de medición no se centran en los usuarios; las métricas son, en la mayoría de los casos, subjetivas y se basan en estimaciones de experiencias pasadas de los usuarios. Por lo tanto, se necesita más investigación para involucrar a los usuarios y considerarlos como un factor crítico en el éxito de un servicio cloud

8) Compatibilidad

La compatibilidad contiene solo 4 operacionalizaciones de métricas, que representan el 1 % del total de métricas. En la computación en la nube, existe una necesidad intrínseca de que los servicios cloud intercambien e interactúen con otros servicios, independientemente del proveedor. Un desafío aquí son los servicios heterogéneos de infraestructura basados en múltiples cloud (multi-nube) y la integración de aplicaciones de múltiples proveedores y dominios. Estos problemas se vuelven más difíciles de gestionar a medida que los sistemas se vuelven más complejos e interconectados.

Un análisis de las métricas recolectadas muestra que 3 de ellas miden la interoperabilidad. La ISO/IEC 17788 [47] define la *interoperabilidad* como la capacidad de dos o más sistemas o aplicaciones para intercambiar información y utilizar mutuamente la información que se ha intercambiado. En la computación en la nube, existen básicamente dos escenarios de interoperabilidad:

i. La capacidad de una aplicación que se ejecuta en un sistema de consumo para intercambiar información con un servicio cloud y utilizar la información de este; y

ii. la capacidad de un servicio cloud para funcionar con otros servicios cloud.

La interoperabilidad entre ambos servicios cloud está adquiriendo cada vez más importancia. Por ejemplo, Garg et al. [131] lo miden como la relación entre las plataformas proporcionadas por el proveedor y las plataformas requeridas por los usuarios , mientras que Nadanam y Rajmohan [114] lo miden como el nivel de interacciones eficientes entre el servicio cloud y sus servicios dependientes.

Sin embargo, se observa una falta de métricas para medir la coexistencia, que es la capacidad de un servicio cloud para realizar las funciones requeridas mientras comparte un entorno y recursos comunes con otros servicios cloud. Por ejemplo, Nadanam y Rajmohan [114] proponen la componibilidad como un indicador que emplea una suma ponderada de modularidad e interoperabilidad del servicio. Su propósito es garantizar que el servicio sea adaptable a cualquier entorno cloud mediante la incorporación de otros servicios que puedan personalizarse de manera más fácil y eficaz para satisfacer las necesidades específicas de los usuarios.

También faltan métricas que puedan utilizarse para soportar los procesos de migración a la nube, la composición de nuevos servicios cloud a partir de múltiples servicios o la gestión multi-nube en la que se requieran mecanismos que verifiquen y aseguren la compatibilidad de aplicaciones y servicios cloud. Entre los beneficios de la interoperabilidad se incluyen menores costos de integración y un aumento del valor de los servicios cloud al ofrecer nuevas funcionalidades, que se proporcionan al componer servicios cloud [141], por lo tanto, existe la necesidad de más investigación acerca de la evaluación de la compatibilidad en los servicios cloud.

4.4.2.2. Tipo de métrica

Los resultados muestran que existen 108 operacionalizaciones de métricas base, que representan el 23 % del total de métricas. Al realizar el análisis temático, encontramos

43 *métricas base* duplicadas que se fusionaron. También se recopiló el método empleado para medir cada métrica base. La mayoría de estas operacionalizaciones de métricas se definieron con el fin de medir la capacidad de los recursos (físicos o virtualizados) y de la red. Algunos ejemplos incluyen métricas para medir la capacidad de la CPU (p. ej., la frecuencia de la CPU [132]), la memoria (p. ej., el tamaño de la RAM [137]), la red (p. ej., la fluctuación [140], el ancho de banda [133]), las máquinas virtuales (p. ej., el número de núcleos de CPU asignados [132]) y el almacenamiento (p. ej., el disco utilizado [124]). Estas métricas se utilizan habitualmente para medir los atributos de elasticidad y escalabilidad.

Las operacionalizaciones de *métricas derivadas* representan el 74 % del total de métricas. Se trata de medidas de alto nivel que contienen una función de medición que muestra cómo se pueden combinar o agregar métricas base o derivadas. Un ejemplo de agregación de métricas es la capacidad del servicio cloud, que es medida por Baranwal y Vidyarthi como una agregación de la capacidad de los diferentes componentes del servicio, es decir, CPU, memoria y almacenamiento [129].

Los *indicadores* representan el 3 % del total de métricas, también se trata de medidas de alto nivel que utilizan un modelo de análisis y podrían estimar o predecir otra medida. Por ejemplo, Lee et al. [8] miden la eficiencia del servicio mediante una suma ponderada del comportamiento temporal y la utilización de recursos, mientras que Lim y Thiran [142] miden la eficiencia utilizando una suma ponderada de disponibilidad, confiabilidad y tiempo de respuesta. El uso de ponderaciones permite a los *stakeholders* expresar sus principales preocupaciones a la hora de evaluar los servicios cloud.

4.4.2.3. Soporte de herramientas

Los resultados muestran que solo 70 operacionalizaciones de métricas, el 15 %, están respaldadas por una herramienta que ayuda a los *stakeholders* a realizar el proceso de medición. Estas herramientas son en su mayoría herramientas de monitorización comerciales que se han utilizado principalmente para calcular las medidas de rendimiento de los recursos físicos y virtualizados. Por ejemplo, Baranwal y Vidyarthi utilizaron la herramienta CLOUDSLEUTH como una herramienta de visualización casi en tiempo real para calcular la disponibilidad y el tiempo de respuesta de diferentes proveedores de servicios [129].

También se observa el uso común de herramientas de evaluación comparativa para probar las capacidades de carga de trabajo de las plataformas cloud. Varias herramientas han sido desarrolladas por la industria (p. ej., YCSB, TPC-W), por la academia (p. ej., BenchClouds, CloudSuite) u otras áreas para fines específicos, como HiBench para aplicaciones Hadoop, que analiza big data. Hwang et al. [124] utiliza cinco herramientas de evaluación comparativa (YCSB, CloudSuite, HiBench, BenchClouds y TPC-W) para llevar a cabo experimentos controlados con el fin de evaluar el rendimiento en una nube híbrida.

Los resultados de las herramientas de evaluación comparativa se pueden utilizar como evidencia para realizar modificaciones o adaptaciones del servicio cloud de acuerdo con los parámetros informados en cada herramienta (p. ej., las operaciones probadas, la configuración de las instancias o VM, el tamaño de las instancias). Por lo tanto, los resul-

tados pueden conectarse a las herramientas o a la plataforma cloud (p. ej., Amazon EC2, Rackspace) seleccionadas y, en consecuencia, afectan la reproducción de los cálculos de las métricas o las comparaciones coherentes entre los resultados. Otros usos de las herramientas de evaluación comparativa en los estudios primarios son los marcos para la selección de proveedores de servicios cloud, la clasificación de proveedores de servicios cloud y la clasificación de precios.

Los resultados muestran que la mayoría de las operacionalizaciones de métricas (es decir, 402, que representan el 85 % de las métricas) no están respaldadas por herramientas. Esto significa que el proceso de medición se realiza manualmente. Sin embargo, este resultado debe ser visto con cautela debido a que la mayoría de los estudios primarios no mencionaron explícitamente el instrumento o marco utilizado para obtener el valor de la métrica, por lo que fueron clasificados como manuales. Además, es conocido que la mayoría de los proveedores de servicios cloud ofrecen funciones, APIs o aplicaciones con las que monitorizar los recursos cloud (p. ej., *CloudWatch* proporcionado por Amazon o *AzureWatch* proporcionado por Microsoft Azure). Sin embargo, nuestro estudio refleja cómo se han calculado las métricas en los estudios primarios seleccionados.

El bajo nivel de automatización de las métricas recopiladas también podría explicarse por el hecho de que muchas de esas métricas son teóricas, utilizan algoritmos complejos o requieren datos externos o acumulativos que no se pueden implementar de manera fácil o efectiva. Por métricas teóricas, nos referimos a aquellas métricas que utilizan conceptos abstractos en sus funciones de medición, por lo que estos conceptos deben redefinirse en términos de artefactos y plataformas cloud específicas.

4.4.2.4. Resultado de la medición

Con respecto a los resultados de las mediciones, casi todas las métricas proporcionan resultados *cuantitativos*, 455 métricas, lo que supone el 96,8 % de las métricas. Esto es coherente, considerando que la eficiencia de desempeño y la fiabilidad fueron las características de calidad con el mayor número de operacionalizaciones de métricas. En los entornos cloud, estos son factores críticos que necesitan medidas cuantitativas que permitan a los proveedores y a los clientes de servicios cloud supervisarlos y controlarlos. Algunos ejemplos son el tiempo de respuesta [132, 129] y la eficiencia del tiempo de respuesta [142], cuyos resultados de medición son cuantitativos y se proporcionan en unidades de tiempo (p. ej., segundos, milisegundos). Además, los resultados de medición de métricas como el número de procesos monitorizados y el número de conexiones TCP [133] también son cuantitativos, pero sus resultados se proporcionan en valores absolutos.

Solo 12 operacionalizaciones de métricas, 2,6 % del total, presentan resultados cualitativos. Las métricas *cualitativas* se utilizan principalmente para medir la responsabilidad, la agilidad y el costo al comparar diferentes servicios cloud. Algunos ejemplos de atributos que utilizan resultados cualitativos son la continuidad, que utiliza una escala ordinal para calificar el mecanismo de preparación para emergencias [11].

Por último, solo 3 operacionalizaciones de métricas, 0,6 % del total, proporcionaron un resultado de medición *híbrido* (es decir, SLA/seguridad, idoneidad y ubicación del

centro de datos). El objetivo de SLA/Seguridad es evaluar el cumplimiento de los SLA midiendo los atributos de la regulación de seguridad, privacidad o derechos de autor [124]. La idoneidad es el grado en que el proveedor de servicios cloud cumple con los requisitos del cliente y se puede utilizar para cuantificar las características esenciales y no esenciales. Las características esenciales se cuantifican calificando la satisfacción de los requisitos esenciales como uno si se satisfacen todas las características y como cero en caso contrario. Por el contrario, las características no esenciales se cuantifican como una relación entre las características no esenciales proporcionadas y las características no esenciales requeridas [131]. Por último, la ubicación del centro de datos utiliza el número de centros de datos y la distancia entre ellos, lo que significa que el proveedor con la distancia mínima ocupa el primer lugar, y así sucesivamente [116].

En general, se cree que el elevado número de métricas cuantitativas es muy positivo, ya que facilita la evaluación de una serie de atributos de calidad de forma objetiva y el uso de estos resultados de medición para apoyar la adaptación y evolución continua de los servicios cloud con el fin de satisfacer las necesidades de los grupos de interés. Sin embargo, se observa una falta de mecanismos basados en umbrales para ayudar a los *stakeholders* en la interpretación de los resultados de las mediciones. Establecer umbrales adecuados para las métricas de los servicios cloud (p. ej., para medir la elasticidad) no es una tarea fácil. Por ejemplo, la carga de trabajo o el comportamiento de la aplicación cambia, lo que hace que la precisión de los resultados de la métrica sea subjetiva y propensa a la incertidumbre. Por lo tanto, se necesitan nuevos mecanismos basados en umbrales que tengan en cuenta las características específicas de los servicios cloud.

4.4.2.5. Fases del ciclo de vida del servicio cloud

Los resultados muestran que la mayoría de las métricas se aplicaron en la fase de *Operación* con 373 operacionalizaciones de métricas, lo que representa el 56 % del total de métricas. Esto indica que estas métricas se miden principalmente durante el uso real del servicio (p. ej., durante la monitorización del servicio cloud en tiempo de ejecución). La monitorización es un componente clave de la mejora continua del servicio desde la perspectiva del proveedor y los resultados de la medición normalmente se exportan al portal en la nube para permitir a los clientes ver cómo se están desempeñando sus servicios. Algunos ejemplos son métricas como la velocidad de lectura y escritura en disco, las conexiones activas e inactivas, el uso del CPU y la memoria, cuyo objetivo es informar a los *stakeholders* sobre el estado de aprovisionamiento de sus recursos [133].

La segunda fase del ciclo de vida, con el mayor número de métricas, fue *Adquisición*, con 174 operacionalizaciones de métricas, lo que corresponde al 26 % del total. Esta fase es crucial para establecer un SLA entre el cliente y el proveedor del servicio cloud. En esta fase, un cliente potencial puede utilizar las ofertas de servicios publicadas por el proveedor de servicios cloud para comprobar si el servicio cumple con sus requisitos en términos de, por ejemplo, seguridad, protección de datos personales, rendimiento, etc., y ver cómo se compara una oferta con otra en el mercado. Por ejemplo, Rizvi et al. [130] propuso el índice de seguridad con el fin de describir el nivel de seguridad alcanzado por los proveedores de

servicios cloud. Esto significa que los consumidores deciden o no adoptar servicios cloud y requieren métricas para respaldar su proceso de toma de decisiones.

La mayoría de las métricas de esta fase se utilizaron para evaluar las ofertas o comprobar el rendimiento de los proveedores de servicios cloud. Por ejemplo, Abdeladim et al. [143] hizo uso de métricas de cobertura de subaprovisionamiento, sobreaprovisionamiento y escalabilidad para estimar la demanda de recursos (p. ej., CPU, memoria, espacio y rendimiento del disco duro) en la fase de adquisición.

Si bien encontramos un gran número de operacionalizaciones de métricas, pocas pueden ser aplicadas a las fases de *Requisitos* con 53 operacionalizaciones, que corresponden al 8 %; *Desarrollo* con 58 operacionalizaciones, que corresponden al 9 %; e *Integración* con 8 operacionalizaciones, que corresponden al 1 %. La evaluación de la calidad en estas fases del ciclo de vida del servicio cloud es igualmente importante, ya que es ampliamente aceptado que un buen diseño mejora el servicio que se entregará y disminuye los defectos. Sin embargo, nuestros resultados indican que la mayor parte del esfuerzo de evaluación se centra en las últimas fases del ciclo de vida del servicio cloud.

Por último, a pesar de la relevancia de la fase de *Retiro*, que trata sobre la terminación del contrato o la sustitución del servicio cloud, encontramos un número limitado de métricas que se pueden aplicar en esta fase, con 8 métricas que representan el 1 % del total de las métricas. Estas métricas están relacionadas principalmente con el soporte (p. ej., plataformas, sistemas operativos, virtualización y herramientas de software) y la interoperabilidad (p. ej., plataformas y recursos). El número limitado de métricas puede explicarse por el hecho de que esta fase gestiona el cumplimiento de la legislación local y externa (p. ej., La protección de datos) y estas cuestiones pueden ser difíciles de controlar mediante métricas. Este hecho denota que las obligaciones de los proveedores en la custodia de la información de sus clientes no terminan en el momento del cierre contractual ni cuando el proveedor es reemplazado. Por lo tanto, el proveedor sigue siendo responsable del cumplimiento de la legislación vigente. Sin embargo, consideramos que algunas cuestiones relacionadas con la terminación del servicio y la protección de datos podrían controlarse mediante métricas. Respecto a la terminación del contrato del servicio, es necesario contar con métricas para evaluar los problemas relacionados con la dependencia de un proveedor, así como la portabilidad y la compatibilidad de la implementación de las aplicaciones al transferir el servicio a otras plataformas. Respecto a la protección de datos, se necesitan métricas para controlar los problemas relacionados con la transferencia, custodia y eliminación segura de datos para evitar copias no deseadas de datos y métricas para controlar la frecuencia de las copias de seguridad de datos.

Cabe indicar que algunas métricas se pueden aplicar a más de una fase del ciclo de vida del servicio cloud. Esto ocurre, por ejemplo, en contextos multitenencia que son ampliamente utilizados por las aplicaciones SaaS, en los que muchos inquilinos comparten una sola instancia de software y la métrica del número de inquilinos se puede aplicar durante las fases de requisitos, adquisición y operación. En la fase de requisitos, se utiliza para establecer la cuota prevista, mientras que en la fase de adquisición, se utiliza para garantizar la cuota y en la fase de operación, se utiliza para evaluar el cumplimiento de la cuota. Además, dado que el número de inquilinos tiene un impacto en la escalabilidad y

elasticidad del servicio, también debe formar parte del SLA [123].

4.4.2.6. Artefacto cloud medido

Los resultados muestran que la mayoría de las métricas (464 operacionalizaciones de métricas, lo que representa el 87 % del total de métricas) evalúan el *servicio cloud final*. Por ejemplo, Lee et al. [8] usó las métricas de cobertura de recuperación de errores (CFR) y cobertura de tolerancia a errores (CFT) para evaluar la fiabilidad del servicio cloud en ejecución. Esta concentración de métricas para evaluar el servicio cloud es consistente, ya que la mayoría de los estudios se han centrado en la fase de operación del servicio.

El 11 % de las métricas propuestas se aplicaron a la *arquitectura* de los servicios cloud, como la métrica de número de réplicas para IaaS propuesta por Souza et al. [132]. En general, se observa un número reducido de métricas que se pueden utilizar para evaluar las arquitecturas cloud y ayudar a los arquitectos a crear soluciones cloud.

Apenas el 2 % se aplicó a la *especificación* de los servicios cloud, por ejemplo, el plan de emergencia empresarial [11]. Los hallazgos sugieren que existe la necesidad de más métricas que se apliquen en las primeras fases del ciclo de vida del servicio cloud (especificación y arquitectura del servicio) y que se requiere trabajo futuro para abordar estas carencias.

4.4.2.7. Tipo de servicio

Los resultados sugieren que la mayoría de las operacionalizaciones de métricas se aplicaron a los modelos de servicio IaaS y SaaS que representan el 51 % y 35 % del total, respectivamente. *SaaS* proporciona las aplicaciones como servicio a los consumidores y los proveedores son responsables de la implementación, configuración y mantenimiento del servicio. Las métricas de funcionalidad común, funcionalidad no común, cobertura de variabilidad y reusabilidad fueron propuestas por Lee et al. [8] y permiten descubrir el grado de reutilización de la aplicación.

Con respecto a *IaaS*, algunos ejemplos representativos podrían ser métricas como el número de conexiones TCP, el ancho de banda de los flujos de entrada y salida, y el uso de memoria y disco, que proporcionan un estado de lectura de la capacidad de los servidores y también se aplican a las máquinas virtuales [133].

Por último, las métricas de *PaaS* representaron el 14 % del total. En el caso de proveer los entornos de desarrollo de aplicaciones, la preocupación radica en la capacidad de programación o en el uso efectivo de los kits de desarrollo de software (SDK). Algunos ejemplos de estas métricas son el tiempo de prueba, la tasa de autoservicio del usuario y la capacidad de cómputo del recurso [126]. Es importante tener en cuenta que algunas métricas se pueden aplicar a más de un tipo de servicio. Ejemplos de esto son: la escalabilidad expresada como el intervalo dinámico de autoescalado de los recursos con variación de la carga de trabajo, la eficiencia del rendimiento expresada como la aceleración por la ganancia de velocidad mediante el uso de múltiples nodos de procesamiento, y la elasticidad expresada como el tiempo mínimo para cambiar de un estado de bajo aprovisionamiento a uno aprovisionado en el que los recursos disponibles se ajustan lo más cerca posible a la demanda existente [124].

4.4.2.8. Punto de vista de los *stakeholders*

Los resultados muestran que la mayoría de las métricas se utilizaron para ayudar a los proveedores y consumidores de los servicios cloud, con 374 y 283 operacionalizaciones de métricas, respectivamente, que corresponden al 45 % y 34 % del total. Considerando que una métrica puede ayudar a más de un *stakeholder*.

Un *proveedor* de nube realiza diferentes tareas para la prestación de servicios cloud a diferentes niveles (SaaS, PaaS e IaaS). Por ejemplo, a nivel de IaaS, el proveedor es responsable de proporcionar y administrar el procesamiento físico, el almacenamiento, las redes, el entorno de alojamiento y la infraestructura cloud para los consumidores de IaaS [144]. Métricas como la velocidad de transmisión de datos alcanzada para representar la velocidad de transmisión y el retardo en la transmisión propuesto por Saiz et al. [145] se puede utilizar para ayudar a los proveedores a monitorizar la calidad de los servicios e identificar posibles mejoras.

Un *cliente* cloud explora y selecciona un servicio de un proveedor cloud, establece el contrato con el proveedor y utiliza el servicio. Las actividades y los escenarios de uso pueden ser diferentes entre los clientes en función del tipo de servicio solicitado (SaaS, PaaS e IaaS). Algunos ejemplos de métricas que pueden ser utilizadas por los clientes en cualquier tipo de servicio incluyen el porcentaje de tiempo de actividad y la tasa de reparación de accidentes, propuestas por Zhou et al. [11].

Los resultados también indican que los usuarios finales, los facilitadores y los desarrolladores son los roles menos involucrados al medir la calidad de los servicios cloud, con 69 operacionalizaciones de métricas (8 %), 46 operacionalizaciones métricas (6 %) y 57 operacionalizaciones métricas (7 %), respectivamente.

Algunas de las métricas empleadas para ayudar a los *usuarios finales* son el tiempo de respuesta del juego y la puntuación media de opinión del juego, ambas propuestas por Wang y Dey [146]. Una de las métricas utilizadas para ayudar a los facilitadores de servicios cloud a gestionar el rendimiento de los servicios cloud y negociar las cláusulas de servicio entre proveedores y consumidores es la disponibilidad. Este atributo de calidad se puede medir utilizando herramientas de evaluación comparativa en línea como CLOUDSLEUTH o métricas de bajo nivel (p. ej., tiempo de procesador, conexiones actuales, tiempo de actividad) que están disponibles y se pueden extraer de diferentes tipos de servidores (p. ej., hosts de máquinas virtuales, servidores virtualizados, servidores de acceso de clientes). Los *facilitadores* registran el servicio y lo revisan para mantener el historial de un servicio cloud en particular y también recopilan información de los usuarios sobre su experiencia con el servicio. Esta información ayuda en la selección de proveedores de servicios cloud de acuerdo con las necesidades del usuario.

Los *desarrolladores* utilizaron métricas con diferentes propósitos, como evaluar el rendimiento del servicio, mejorar la experiencia del usuario, determinar el entorno de nube adecuado o especificar el umbral superior de usuarios actuales permitidos. Por ejemplo, Wen y Hsiao [147] analizaron la relación entre la QoS y la QoE en el dominio de los servicios cloud de juegos, y consideraron la latencia como un atributo relevante, relevante en este tipo de servicios que manejan las interacciones con el usuario final. Se midió la

latencia utilizando la métrica de latencia de ida y vuelta, que se compone del retardo de transmisión, retardo de procesamiento del lado del servidor y retardo de procesamiento del lado del cliente. Por lo tanto, estos resultados de medición pueden permitir a los desarrolladores de juegos inferir un índice de experiencia de juego mediante el uso de métricas de QoS y evaluar el impacto de su diseño.

En general, nuestros resultados indican que las métricas recopiladas se utilizaron con mayor frecuencia para ayudar a los proveedores a controlar la prestación de servicios cloud con el fin de garantizar su comportamiento, y para ayudar a los consumidores a evaluar la calidad de los mismos servicios y garantizar su cumplimiento con los acuerdos de nivel de servicio.

4.4.2.9. Procedimiento de validación

Este criterio permitió evaluar en qué medida las métricas recopiladas fueron sometidas a algún tipo de validación teórica y/o empírica, es decir, si las métricas medían lo que se pretendía medir y si los resultados eran consistentes con lo esperado.

Los resultados muestran que 443 operacionalizaciones de métricas—el 90 % del total—carecen de cualquier tipo de validación. En consecuencia, solo 47 operacionalizaciones (10 %) fueron validadas, todas ellas mediante métodos empíricos, sin evidencia de validaciones teóricas.

De las métricas validadas, 39 fueron validadas mediante experimentos, aunque la mayoría de los casos no involucraron participación humana, sino que fueron experimentos que compararon los resultados de las métricas con puntos de referencia. Por ejemplo, Hwang et al. [124] llevó a cabo experimentos de evaluación comparativa para validar la eficacia de métricas relacionadas con la elasticidad, considerando la eficiencia y el rendimiento de los servicios cloud (p. ej., la resiliencia representada por una tasa o capacidad para recuperarse de fallos).

Asimismo, se identificaron 8 métricas validadas mediante estudios de caso. Un ejemplo es Zheng et al. [10], quienes realizaron un estudio de caso para evaluar la QoS de distintos servicios de almacenamiento en la nube (Amazon S3, Azure Blob y Aliyun OSS) mediante cuatro operaciones (creación, carga, descarga y eliminación). La evaluación se basó en el estándar IEEE 1061 [148], aplicando criterios de validez como la correlación, consistencia y poder discriminatorio. La correlación evalúa la fuerza de asociación lineal entre un atributo y una métrica; la consistencia determina si una métrica clasifica con precisión un conjunto de servicios en función de una dimensión de calidad; y el poder discriminatorio evalúa si la métrica distingue efectivamente entre servicios de alta y baja calidad. Un ejemplo de una métrica validada es *RESP-Evaluation*, que mide la capacidad de respuesta considerando el tiempo máximo aceptable desde el punto de vista del usuario, en lugar de la perspectiva centrada en el sistema.

Durante el análisis, también se observó que algunos estudios afirmaban realizar estudios de caso cuando, en realidad, solo presentaban pruebas de concepto sobre el posible uso de las métricas. Para que una métrica sea viable, requiere estar bien especificada, lo que permite su evaluación de manera reproducible y repetible. En consecuencia, clasificamos

una métrica como *no validada* cuando no presentaba ningún tipo de validación o cuando la validación se realizaba con un enfoque metodológico incorrecto. Como resultado, el 52 % de las métricas no validadas correspondían a pruebas de concepto. Además, la mayoría de los estudios no describían con suficiente detalle ni el proceso de validación ni los resultados obtenidos, lo que hacía difícil determinar si existía un diseño de validación adecuado o si se trataba únicamente de una prueba de concepto. Esta falta de claridad explica, en gran medida, el elevado número de métricas no validadas.

En definitiva, es necesario avanzar en la validación de las métricas existentes para los servicios cloud. Es especialmente importante aportar evidencia sobre su utilidad como medida de calidad interna, así como sobre su capacidad para predecir atributos de calidad externos, tales como el rendimiento, la seguridad y la mantenibilidad. No obstante, cualquier proceso de evaluación implica costes asociados a la recolección, procesamiento y almacenamiento de los datos requeridos. En entornos cloud, el coste de almacenamiento de datos y la disponibilidad de expertos del dominio deben ser considerados; además, la evaluación de las métricas debe realizarse con un coste aceptable para los *stakeholders* y dentro de un plazo compatible con las decisiones que dichas métricas pretenden respaldar.

4.5. Discusión de resultados

Los resultados de la revisión sistemática permitieron construir y refinar una taxonomía de métricas de calidad para servicios cloud. Esta sección presenta los principales hallazgos organizados en torno a las preguntas de investigación, acompañados de un análisis crítico de las tendencias observadas en la literatura.

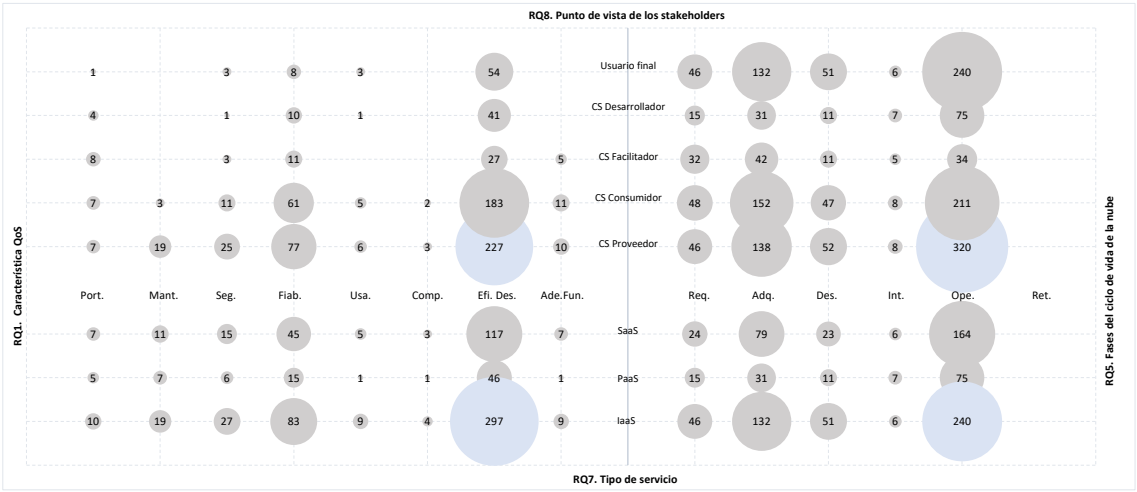


Figura 4.7: Resultados obtenidos al combinar las características de QoS y las fases del ciclo de vida del servicio cloud con el punto de vista de los *stakeholders* y el tipo de servicio.

La Figura 4.7 presenta un gráfico de burbujas de cuatro dimensiones que combina las características de QoS y las fases del ciclo de vida en el eje x, con el punto de vista de los

stakeholders y el tipo de servicio en el eje y. A partir de esta visualización multidimensional, se puede destacar lo siguiente:

- Las métricas asociadas a la eficiencia de desempeño, fiabilidad, seguridad y portabilidad ofrecen cobertura a distintos *stakeholders*, aunque con niveles de atención variables, siendo proveedores y consumidores los actores más beneficiados. Esto refleja el predominio de métricas orientadas a atributos de calidad externos, como el rendimiento y confiabilidad.
- Se observa una notoria escasez de métricas para la compatibilidad, usabilidad y mantenibilidad. Además, pocas resultan de utilidad para ciertos *stakeholders*, como facilitadores, desarrolladores o usuarios finales.
- La fase de operación concentra el mayor número de métricas orientadas a asistir a proveedores y usuarios finales. Le sigue la fase de adquisición, que también presenta una cantidad considerable de métricas relevantes tanto para consumidores como para proveedores.
- Se identificó una marcada carencia de métricas dirigidas a desarrolladores, especialmente en el contexto de servicios PaaS.
- Los servicios IaaS y SaaS son los más evaluados, particularmente en las fases de operación y adquisición, lo que refleja las principales preocupaciones de usuarios y proveedores. En contraste, PaaS recibe menor atención, con métricas limitadas para apoyar su uso en entornos de desarrollo, pruebas o administración de aplicaciones.

En la Figura 4.8 se presenta otro gráfico de burbujas que relaciona el tipo de métrica, el soporte de herramientas, el resultado de medición y el artefacto medido en el eje x con las características de QoS en el eje y. Entre los resultados más relevantes se destacan los siguientes:

- La mayoría de las métricas base se utilizaron para medir atributos de calidad relacionados con la eficiencia de desempeño, mientras que un número más significativo de métricas derivadas proporciona una mayor cobertura tanto de la eficiencia de desempeño como de la fiabilidad, y una menor atención al resto de las características de calidad definidas en el estándar ISO/IEC 25010. Este predominio de métricas derivadas puede deberse a que, al combinar métricas base (y, en algunos casos, otras métricas derivadas), se proporciona información más significativa y útil para los *stakeholders*.
- La medición manual prevalece, aunque los atributos relacionados con la eficiencia de desempeño [113] y fiabilidad [124] presentan mayor grado de automatización gracias al uso de herramientas de monitorización nativas de las plataformas cloud.
- Predominan las métricas cuantitativas y objetivas, lo que facilita un análisis más detallado del comportamiento del servicio en tiempo de ejecución, coherente con su uso frecuente durante la fase de operación.

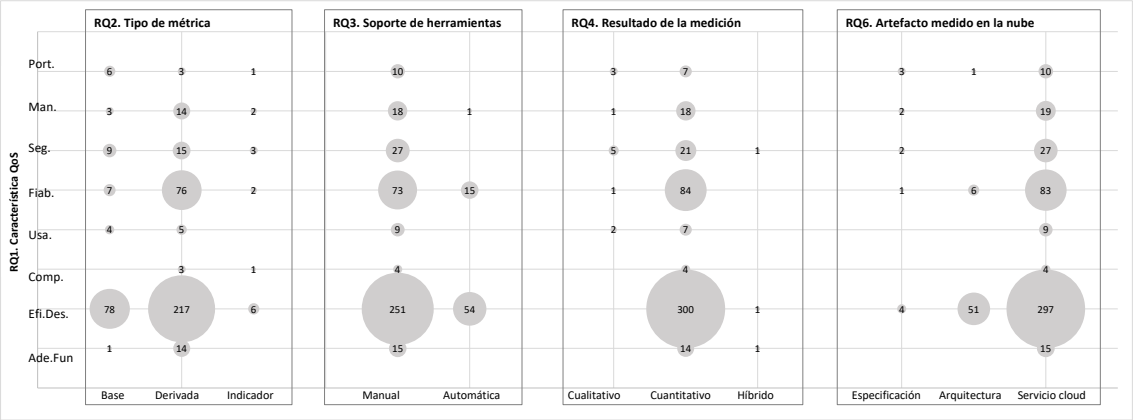


Figura 4.8: Resultados obtenidos al combinar las características de QoS con el tipo de métrica, el soporte de la herramienta, el resultado de la medición y el artefacto cloud medido.

- La mayoría de las métricas se utilizaron para evaluar la eficiencia de desempeño y la fiabilidad de los servicios cloud reales. Este alcance limitado implica que dichas métricas resultan especialmente útiles para apoyar la toma de decisiones, pero solo a corto plazo (acción). No obstante, se evidencia la necesidad de contar con métricas que abarquen múltiples fases e incluso todo el ciclo de vida del servicio cloud, las cuales serían más adecuadas para apoyar decisiones de largo plazo (visión).

- Son escasas las métricas aplicables en fases tempranas (especificación, arquitectura, diseño), pese a que la detección oportuna de desviaciones en estas etapas podría reducir costos y riesgos de manera significativa.

En síntesis, la evidencia muestra una marcada concentración de métricas en la característica de **eficiencia de desempeño**, que aglutina más de la mitad de los atributos, métricas y operacionalizaciones identificadas. Este predominio se explica por la importancia que tienen el rendimiento, la capacidad y la utilización de recursos en la percepción de calidad por parte de los usuarios, así como en la definición de los SLA. Sin embargo, otras características, como la compatibilidad y usabilidad, presentan una representación considerablemente menor, lo que revela un desequilibrio en los enfoques de investigación actuales.

El análisis de los estudios primarios también reveló que la mayoría se enfocan en una única característica de calidad, sin ofrecer una visión integral del problema. Solo un número reducido de investigaciones abordó de forma conjunta cuatro o más características, y únicamente dos estudios consideraron siete de las ocho definidas en el estándar ISO/IEC 25010 [1]. Este hallazgo sugiere que la evaluación de la calidad de los servicios cloud continúa fragmentada, lo que dificulta la adopción de enfoques holísticos que apoyen la toma de decisiones de manera equilibrada y consistente.

Otro aspecto relevante es la limitada validación empírica de las métricas propuestas. Si bien muchos trabajos se enfocan en la definición de métricas, pocos aportan evidencia

sobre su aplicabilidad en entornos reales, lo que limita su adopción práctica.

Finalmente, la taxonomía obtenida permitió establecer correspondencias claras entre atributos, métricas y operacionalizaciones, ofreciendo un marco conceptual que facilita la comparación y la reutilización de métricas. Este resultado contribuye a consolidar la terminología y a mitigar la dispersión conceptual observada en la literatura.

4.6. Amenazas a la validez

A pesar de que se llevó a cabo un proceso riguroso y sistemático, esta SLR puede verse afectada por algunas amenazas a su validez. En esta sección se revisan las acciones implementadas para minimizar posibles sesgos.

4.6.1. Amenazas a la identificación de estudios

Al diseñar las estrategias de búsqueda, el objetivo principal fue recuperar la mayor cantidad posible de literatura disponible para evitar cualquier sesgo. Por ello, el alcance del estudio incluyó trabajos de investigación de distintas comunidades, entre ellas ingeniería de software, sistemas de información y computación en la nube. Estas comunidades utilizan terminologías diferentes para conceptos similares; en consecuencia, se identificaron términos comunes y se combinaron en la cadena de búsqueda para cubrir la mayor cantidad de variaciones posibles y minimizar sesgos.

En definitiva, la evaluación de la calidad de la cadena de búsqueda para la identificación de los estudios primarios es un factor clave para evitar la exclusión de investigaciones relevantes. Con el fin de mitigar la amenaza relacionada con la cadena de búsqueda, se llevaron a cabo las siguientes acciones:

a. Verificación de que las bibliotecas digitales incluyan todas las revistas relevantes y actas de congresos relevantes en los campos de computación en la nube y calidad de software, realizando búsquedas manuales en caso de detectar fuentes faltantes;

b. Definición de la cadena de búsqueda a partir de términos presentes en publicaciones relevantes cuya existencia ya se conocía (p. ej., [8, 10, 11]);

c. Refinamiento de la cadena de búsqueda mediante diferentes combinaciones de términos, eligiendo la configuración que produjera los mejores resultados;

d. Aplicación de la cadena de búsqueda a los mismos metadatos de cada artículo; y

e. Adaptación de la cadena de búsqueda a las particularidades de cada biblioteca digital según las capacidades específicas de cada repositorio para maximizar la precisión de los resultados recuperados, reducir ruido y asegurar pertinencia al dominio de métricas de QoS en servicios cloud.

Por último, para asegurar la identificación imparcial de estudios relevantes, se diseñó un protocolo de revisión. Con todo, la principal limitación de los estudios primarios es que solo

se consideraron publicaciones académicas en revistas y actas de congresos, quedando fuera la literatura gris y los informes no publicados. Esta elección se justificó por la garantía de revisión por pares que ofrecen las publicaciones académicas. Sin embargo, se prevé validar la relevancia de las métricas seleccionadas en contextos industriales para aumentar la aplicabilidad práctica de los resultados.

4.6.2. Amenazas a la selección de estudios y la extracción de datos

Con el fin de validar la selección de los estudios primarios y reducir el juicio subjetivo del equipo de investigación, se implementaron las siguientes acciones durante el proceso de selección:

a. Se registraron las motivaciones para incluir o excluir cada estudio. Dos investigadores mantuvieron discusiones continuas sobre qué artículos debían ser incluidos, resolviendo cualquier discrepancia por consenso. Además, se verificó que se incluyeran diez artículos relevantes de diferentes bibliotecas digitales (p. ej., [8, 10, 11, 127, 146, 131]).

b. Posteriormente, otros tres investigadores realizaron una segunda iteración sobre una muestra aleatoria de 20 artículos (10 incluidos y 10 excluidos) para verificar la correcta aplicación de los criterios de inclusión y exclusión. Las discrepancias se resolvieron por consenso y el nivel de acuerdo entre los investigadores se evaluó mediante el índice *Kappa de Fleiss* [108], obteniéndose una puntuación global de 0.87, lo que indica un buen nivel de acuerdo.

c. Finalmente, se realizó una evaluación de la calidad de todos los estudios seleccionados, descartando aquellos que no alcanzaron el umbral mínimo establecido (cuatro estudios). Este es un paso fundamental, ya que la solidez de las conclusiones depende directamente de la calidad de los estudios primarios incluidos.

Con respecto a la validación de la estrategia de extracción de datos, se llevaron a cabo diversas acciones orientadas a garantizar la consistencia y fiabilidad del proceso:

a. Los criterios de extracción se definieron a partir de las preguntas de investigación y se organizaron mediante una taxonomía y un formulario estructurado que facilitó la recolección sistemática y homogénea de los datos.

b. La extracción inicial de los datos fue realizada por el autor y posteriormente revisada por otro investigador. Al igual que en la fase anterior, se mantuvieron discusiones continuas sobre la clasificación de los artículos, resolviéndose las discrepancias por consenso.

c. La estrategia de extracción fue sometida a una validación externa: un investigador independiente evaluó tanto los criterios de extracción de datos como el procedimiento mediante la clasificación de dos artículos seleccionados al azar. A partir de esta revisión, se introdujeron ajustes menores para mejorar la claridad y reducir posibles ambigüedades.

d. Finalmente, otros tres investigadores realizaron una segunda iteración de extracción de datos sobre una muestra aleatoria de 30 estudios previamente incluidos. Los resultados fueron discutidos en una reunión del equipo y las discrepancias se resolvieron por consenso. La fiabilidad de la extracción se evaluó mediante el índice *Kappa de Fleiss* [108], obteniéndose puntuaciones que reflejan un nivel de concordancia alto y una baja variabilidad entre evaluadores: 0.95 para el criterio 1; 0.86 para el criterio 2; 0.84 para el criterio 3; 0.89 para el criterio 4; 0.85 para el criterio 5; 0.87 para el criterio 6; 0.95 para el criterio 7; 0.86 para el criterio 8; y 0.84 para el criterio 9.

4.6.3. Amenazas a la síntesis de datos y los resultados

Con respecto a la síntesis de los datos, tal como se describe en la Sección 4.2.2.4, se emplearon métodos cualitativos—principalmente síntesis narrativa y análisis temático—para organizar, analizar y consolidar la información extraída. Estos métodos de investigación contribuyeron a mantener una consistencia razonable en la interpretación de los datos. Sin embargo, en algunos casos surgieron dificultades al extraer e interpretar la información, debido a que ciertos artículos no presentaban los detalles necesarios o carecían de suficiente claridad para responder de forma completa a algunas preguntas de investigación. Con el fin de mitigar el sesgo de interpretación, se empleó un esquema de codificación unificado que guía la extracción de los datos. Asimismo, los resultados preliminares fueron sometidos a revisión cruzada por varios investigadores, lo que permitió contrastar interpretaciones y asegurar la coherencia conceptual de la clasificación. La validación externa de la estrategia de extracción, mediante la revisión independiente, reforzó la consistencia del análisis e incrementó la fiabilidad de los resultados. Estas acciones contribuyeron a reducir la subjetividad inherente a los procesos de interpretación cualitativa y garantizaron que las conclusiones derivadas fueran trazables y fundamentadas en la evidencia recopilada.

4.7. Implicaciones

En las siguientes subsecciones, se discuten las implicaciones que la taxonomía de métricas para servicios cloud tiene para los investigadores y profesionales de la industria.

4.7.1. Implicaciones para investigadores

Los hallazgos de este estudio tienen importantes implicaciones para los investigadores que planean desarrollar nuevos trabajos relacionados con la calidad de los servicios cloud. En primer lugar, se identificó que la medición de la calidad en este dominio no ha sido abordada de manera holística en los estudios primarios analizados. La taxonomía propuesta ha permitido reunir e integrar el conocimiento existente sobre métricas de calidad, evidenciando que no todas las propiedades de calidad ni fases del ciclo de vida del servicio cloud relevantes para los *stakeholders* están adecuadamente cubiertas. Un ejemplo notable es la ausencia de métricas para la fase de retiro, a pesar de su importancia para la seguridad y la gestión de la infraestructura en las organizaciones.

Asimismo, se observó que un gran número de métricas identificadas son de bajo nivel y se centran principalmente en la eficiencia de desempeño (p. ej., tiempo de respuesta, uso de CPU), así como en propiedades relacionadas con las infraestructuras cloud. Esto pone de manifiesto la necesidad de desarrollar métricas e indicadores de alto nivel—como aquellas utilizadas para evaluar el cumplimiento de los SLAs)—independientemente de la característica de QoS o tipo de servicio. También se requieren más métricas orientadas a servicios PaaS y SaaS, dado que estos modelos continúan siendo críticos para las organizaciones.

De igual manera, los resultados revelan que varias características de calidad definidas en la norma ISO/IEC 25010 están insuficientemente cubiertas. En particular, la portabilidad, la usabilidad y la compatibilidad representan conjuntamente menos del 5 % del total de métricas (22 de 470). Este déficit resulta especialmente relevante si se considera que la usabilidad es un factor fundamental para la adopción de servicios cloud, mientras que la compatibilidad es fundamental para lograr interoperabilidad entre servicios de distintos proveedores. Además, se evidenció una carencia de métricas que aborden de manera efectiva las preocupaciones de seguridad y privacidad, a pesar de ser aspectos determinantes para la adopción de servicios cloud. Por lo tanto, es necesario realizar un esfuerzo orientado al diseño de métricas que midan de manera rigurosa estas características.

En cuanto a las fases del ciclo de vida, la mayoría de los artículos solo reportan evaluaciones en la fase de operación o bien se centran en una sola fase. Sin embargo, la evaluación continua de la calidad a lo largo de todo el ciclo de vida es esencial para garantizar que el servicio cumpla con los requisitos esperados. Existe, pues, una marcada escasez de métricas aplicables en etapas tempranas del desarrollo, donde la detección oportuna de problemas podría reducir costos y riesgos. Esta situación sugiere que las prácticas actuales aún no aprovechan adecuadamente los artefactos intermedios generados durante el diseño y especificación de los servicios cloud. Nuevas líneas de investigación deberían orientarse a integrar mecanismos de evaluación de la calidad desde etapas tempranas del ciclo de vida, combinando las métricas identificadas en este estudio con otras soluciones técnicas que permitan una visión global y sistemática de la calidad.

Otro hallazgo relevante es que la mayoría de las métricas analizadas se orientan a la evaluación de servicios desplegados en entornos de nube únicos. Sin embargo, buscando mayor flexibilidad de los servicios, las organizaciones están adoptando cada vez más entornos multi-nube (híbridos, federados, públicos y privados), los cuales introducen complejidades adicionales para monitorizar y garantizar la calidad, ya que esta depende de múltiples servicios e infraestructuras interconectadas. Por lo tanto, es necesario avanzar en el desarrollo de métricas que apoyen la evaluación en estos escenarios más complejos, facilitando así la gestión de nuevas prácticas de desarrollo, integración y entrega continua de servicios cloud.

Finalmente, la mayoría de las métricas identificadas en los estudios primarios aún no han sido adoptadas ampliamente en entornos industriales, lo que evidencia la necesidad de validaciones empíricas adicionales. Experimentos controlados o estudios de caso en contextos reales permitirían determinar la utilidad de estas métricas en proyectos de distintas escalas y en diversos dominios.

4.7.2. Implicaciones para profesionales de la industria

Los resultados de este estudio son relevantes para la industria, en particular para las empresas de desarrollo y operación de servicios cloud que buscan mecanismos efectivos para asegurar y gestionar la calidad de sus servicios. El catálogo de métricas elaborado proporciona una base sistemática y estructurada que puede apoyar a distintos *stakeholders*—como clientes, proveedores, facilitadores, arquitectos y administradores de infraestructura—en la selección de métricas adecuadas conforme al contexto específico de aplicación. Dicho contexto puede estar definido por el tipo de artefacto (p. ej., especificaciones de SLA, arquitecturas cloud, servicios en operación), el tipo de servicio (SaaS, PaaS, IaaS) o la fase del ciclo de vida del servicio cloud (p. ej., adquisición, desarrollo, integración, operación). Esto implica que los profesionales pueden seleccionar y adaptar un conjunto de métricas de nuestro catálogo para integrarlas en programas de medición más amplios.

Los clientes y proveedores pueden utilizar el catálogo como guía para especificar objetivos de nivel de servicio, identificar las características y atributos de QoS relevantes para sus necesidades y seleccionar las métricas que deben incluirse en los SLA. Dado que una métrica puede calcularse mediante diferentes funciones de medición, el catálogo de métricas también ayuda a los *stakeholders* a elegir la operacionalización que mejor satisfaga sus objetivos organizacionales o técnicos.

El catálogo de métricas puede servir también para otros propósitos. Por ejemplo, un posible cliente podría emplearlo para evaluar las ofertas de servicios de distintos proveedores y verificar si cumplen con requisitos clave—como seguridad, protección de datos o rendimiento—facilitando comparaciones objetivas entre alternativas de mercado. Asimismo, los clientes actuales podrían utilizar estas métricas para evaluar de forma continua la calidad del servicio adquirido.

Por su parte, los proveedores pueden utilizar el catálogo para múltiples fines: detectar y corregir defectos antes de la prestación del servicio, mejorar características de calidad específicas, garantizar que los clientes reciban el nivel de calidad esperado o incluso fortalecer el posicionamiento de sus servicios en el mercado.

Desde la perspectiva del desarrollo, los equipos técnicos pueden emplear el catálogo de métricas para evaluar y supervisar el rendimiento y otras características de calidad del servicio durante su desarrollo. Esto resulta especialmente útil en entornos de integración y despliegue continuo (CI/CD) o enfoques *DevOps*, donde las métricas proporcionan información continua sobre el estado del servicio desde el punto de vista de los diferentes *stakeholders*, facilitando la toma de decisiones para las acciones correctivas.

En conjunto, el catálogo de métricas constituye un paso importante hacia una mayor transparencia y credibilidad entre los *stakeholders* involucrados en la adquisición, el desarrollo y la operación de servicios cloud. Su adopción sistemática puede contribuir a mejorar la calidad, gestionar riesgos y promover decisiones más informadas en todo el ciclo de vida de los servicios cloud.

4.8. Conclusiones

En este capítulo se presentó el diseño, desarrollo y resultados de un estudio empírico orientado a la construcción de una taxonomía de métricas de calidad para servicios cloud. El proceso metodológico, basado en una revisión sistemática de la literatura, permitió recopilar, clasificar y organizar de forma rigurosa la evidencia disponible, garantizando la trazabilidad y la reproducibilidad de los resultados obtenidos.

La taxonomía propuesta constituye una contribución significativa, ya que integra de manera estructurada conceptos de calidad, atributos, métricas y operacionalizaciones, alineados con las características definidas en la ISO/IEC 25010 [1]. Además de ofrecer una visión consolidada del estado actual de las métricas en servicios cloud, sino también en proporcionar un marco de referencia que facilita la comunicación entre investigadores y profesionales, y promueve la reutilización de métricas en distintos contextos de evaluación.

Los resultados revelaron un marcado predominio de métricas orientadas a la eficiencia de desempeño, frente a la escasa atención prestada a otras características, como la compatibilidad, seguridad y usabilidad. Además, se identificó una limitada validación empírica de las métricas propuestas, lo que pone de manifiesto la necesidad de reforzar la evidencia experimental para favorecer su adopción en escenarios reales. En definitiva, estos hallazgos reflejan la existencia de importantes carencias en la literatura y la ausencia de enfoques holísticos que aborden de manera integral la calidad de los servicios cloud.

A partir de lo anterior, se identifican líneas de investigación futura que resultan necesarias para fortalecer el campo:

- Ampliar la investigación en torno a características de calidad aún insuficientemente exploradas (p. ej., compatibilidad, seguridad, usabilidad).
- Fortalecer la validación empírica de las métricas mediante estudios de caso, experimentos controlados y evaluaciones en entornos industriales.
- Desarrollar marcos metodológicos que integren métricas de múltiples características de calidad, avanzando hacia modelos de evaluación holísticos y representativos de la complejidad inherente a los servicios cloud.

En su conjunto, los resultados obtenidos no solo proporcionan una taxonomía consolidada de métricas, sino que también ofrecen una visión crítica y actualizada del estado de la investigación, permitiendo identificar desafíos y oportunidades que orienten el desarrollo de nuevas propuestas para la evaluación de la calidad en entornos de computación en la nube.

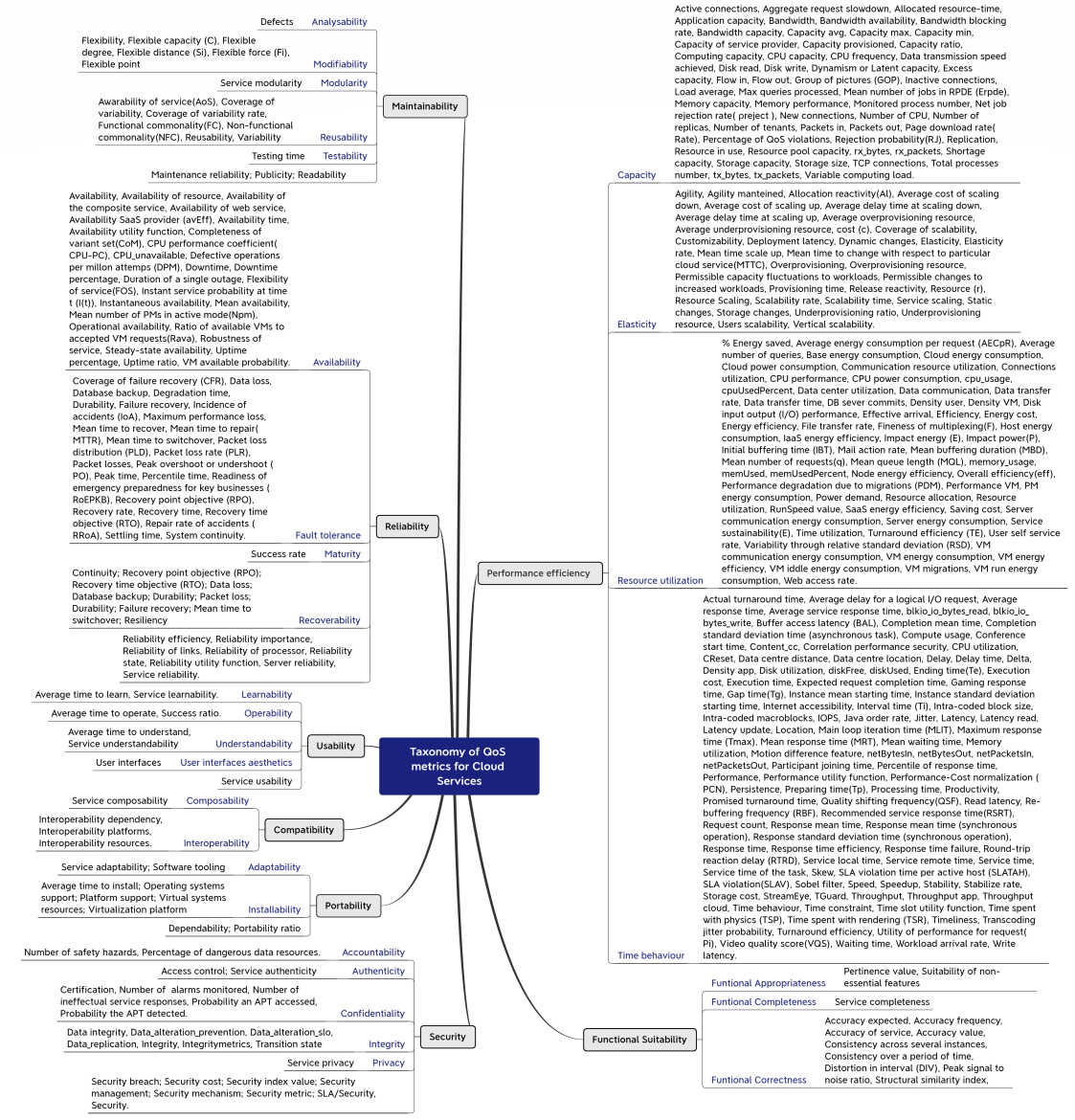


Figura 4.9: Taxonomía de métricas de QoS para servicios cloud[2].

Capítulo 5

Modelo de Calidad para Servicios Cloud

En este capítulo se presenta el diseño y desarrollo de un modelo de calidad para servicios cloud, construido a partir de los hallazgos obtenidos en el capítulo anterior.

En la Sección 5.1 se presenta la motivación para definir un modelo de calidad y las contribuciones de este capítulo. En la Sección 5.2 introduce la metodología que se ha utilizado como base para la definición del modelo de calidad. En la Sección 5.3 se presenta el enfoque mixto empleado para definir el modelo de calidad para los servicios cloud. En la Sección 5.4 se detalla la descomposición jerárquica del modelo de calidad propuesto para los servicios cloud. En la sección 5.5 se presenta el enfoque de evaluación definido para utilizar el modelo de calidad propuesto para realizar evaluaciones específicas. En la Sección 5.6 se presentan las conclusiones y se analizan las implicaciones para la investigación y la práctica de este capítulo.

5.1. Introducción

La evaluación efectiva de la calidad de los servicios cloud es fundamental para comprender sus complejos mecanismos de aprovisionamiento y garantizar el cumplimiento de los SLA [149]. En este contexto, un modelo de calidad no es solo un marco conceptual, sino un instrumento sistemático que permite definir características relevantes, medirlas e interpretar sus resultados, brindando soporte a la toma de decisiones informadas.

A pesar de la importancia crítica de la calidad, su evaluación en entornos cloud se caracteriza por un alcance limitado y la abstracción insuficiente de los modelos existentes. El estándar ISO/IEC 25010 proporciona una línea base genérica, pero su carácter genérico dificulta su aplicación directa ya que no proporciona atributos propios del entorno cloud como la elasticidad, la multitenencia o la gestión de la infraestructura virtualizada. Aunque se han propuesto modelos de calidad específicos (por ejemplo, CLOUDQUAL o S3MQUAL) [10, 93], estos a menudo incurren en fragmentación conceptual y una cobertura limitada de atributos.

En particular, los modelos existentes suelen centrarse en un subconjunto reducido de características (frecuentemente relacionadas con el rendimiento), y en métricas de bajo nivel, sin establecer una correspondencia sistemática con las características de calidad abstractas definidas en estándares como ISO/IEC 25010. Asimismo, muchos de estos enfoques están restringidos a un único nivel del servicio (por ejemplo, SaaS) y no consideran explícitamente la naturaleza multicapa de los entornos cloud. Como consecuencia, resulta difícil analizar la calidad de forma holística, identificar relaciones entre capas (SaaS, PaaS e IaaS) o reutilizar el modelo en distintos contextos organizativos. Estas limitaciones evidencian la necesidad de un modelo más completo, trazable y adaptable, capaz de integrar estándares internacionales con métricas de QoS observables en entornos reales.

El problema se manifiesta, además, en la falta de una operacionalización rigurosa que vincule las características de alto nivel con múltiples métricas y funciones de medición concretas. Esta tesis, por lo tanto, busca cerrar la brecha de trazabilidad y operacionalización, proporcionando un modelo que trascienda la abstracción y sea aplicable a la medición real y proactiva de los servicios cloud.

En este capítulo, se propone un modelo de calidad para servicios cloud que aborda directamente estas limitaciones, proporcionando una solución integral alineada con los principios establecidos por la ISO/IEC 25010 [1], e incorporando tanto la evidencia empírica derivada de la taxonomía de métricas como las propiedades específicas del dominio cloud. Frente a propuestas limitadas como CLOUDQUAL o modelos restringidos a SaaS, el modelo aquí propuesto ofrece una visión holística que integra múltiples características de calidad,y las operacionaliza en un marco que facilita su aplicación práctica y su integración en procesos sistemáticos de evaluación. El modelo captura las ocho características del estándar ISO/IEC 25010, ofrece flexibilidad frente a requisitos cambiantes y resulta aplicable en entornos reales. Como advierten Wagner et al. [21], uno de los principales retos en la práctica es precisamente la brecha entre las características abstractas de calidad y su operacionalización en métricas y evaluaciones concretas, brecha que este trabajo aborda de manera explícita.

Para alcanzar este objetivo, se describe la estructura conceptual del modelo, sus componentes y las interacciones entre ellos. Además, se detalla el proceso de construcción, destacando cómo incorpora atributos, métricas y operacionalizaciones previamente sistematizados. Finalmente, se presenta el enfoque propuesto para su medición en la práctica, alineado con la ISO/IEC 25040 [14], así como su potencial para guiar futuras investigaciones en el ámbito de la evaluación de la calidad en servicios cloud.

Las principales contribuciones de este capítulo son:

1. Un modelo de calidad para servicios cloud, alineado con la ISO/IEC 25010 [1], que descompone de forma jerárquica las características de calidad de alto nivel en subcaracterísticas, atributos y métricas derivadas de una revisión sistemática de la literatura [2]. Para ciertos atributos y métricas se proponen distintas *operacionalizaciones*, definidas como funciones de medición alternativas.

2. Un proceso de evaluación, alineado con la ISO 25040 [14], que permite aplicar el modelo de calidad de servicios cloud en contextos específicos.

Entre los requisitos para su definición, se espera que el modelo de calidad sea independiente de la plataforma cloud y que pueda instanciarse en distintos entornos y modelos de servicio (SaaS, PaaS e IaaS). Asimismo, se espera que sea de utilidad para distintos *stakeholders*, permitiendo definir requisitos de calidad, establecer SLA personalizados, supervisar la prestación de servicios y evaluar el cumplimiento de los compromisos de calidad.

5.2. Metodología

Existen antecedentes relevantes en la literatura sobre la definición de modelos de calidad específicos de dominio a partir de modelos genéricos basados en estándares. Un ejemplo representativo es el modelo de calidad para la selección de paquetes de software comerciales (COTS) propuesto por Franch et al [34]. Este trabajo propone una metodología para definir modelos de calidad específicos de dominio a partir de un modelo estándar, principalmente ISO/IEC 9126-1, con el fin de apoyar la selección de paquetes de software. La metodología consiste en identificar el contexto y propósito de la selección, adaptar el modelo de calidad de referencia mediante la selección y el refinamiento de las características y subcaracterísticas relevantes. Posteriormente, estas se descomponen en atributos de calidad observables y evaluables, que pueden priorizarse según las necesidades del usuario. Este enfoque proporciona un marco común y sistemático para expresar requisitos de calidad, describir productos de software y comparar distintas alternativas de manera estructurada.

Más recientemente, Wagner [21] propuso un *enfoque mixto* para la definición de modelos de calidad, desarrollado en el contexto del modelo de calidad Quamoco. Este enfoque combina los enfoques ascendente y descendente, con el propósito de conciliar la calidad abstracta con la viabilidad práctica de medirla, con el objetivo de crear modelos de calidad específicos para dominios concretos, manteniendo coherencia con estándares reconocidos

y, al mismo tiempo, garantizando su aplicabilidad operativa. Esta propuesta resulta especialmente relevante cuando se busca adaptar el modelo genérico del estándar ISO/IEC 25010 a dominios específicos, como los servicios cloud, sistemas embebidos o aplicaciones de IA.

A continuación se describen los principios y los pasos metodológicos clave del enfoque mixto de Wagner et al. [21], que sirven como base para la definición del modelo de calidad presentado en este trabajo.

Principios del enfoque mixto

La metodología se sustenta en la integración de dos direcciones para asegurar la integridad conceptual con la aplicabilidad técnica del modelo de calidad:

- Aplicar el enfoque `descendente`, se parte de las características de calidad abstractas, generalmente definidas en estándares o derivadas de la experiencia de expertos y estudios previos. Estas características se utilizan como marco conceptual para su descomposición en subcaracterísticas y atributos de calidad relevantes para un dominio específico. Este enfoque facilita la identificación de atributos necesarios que no están explícitamente contemplados en el modelo genérico.

- Aplicar el enfoque `ascendente`, se parte de la evidencia empírica disponible, principalmente métricas y mecanismos de medición utilizados en la práctica o implementados en herramientas existentes. A partir de estas métricas, se definen o refinan atributos de calidad y subcaracterísticas, permitiendo que el modelo refleje las capacidades reales de medición y observación en el dominio analizado.

Proceso metodológico de Quamoco

El modelo Quamoco sigue un proceso estructurado compuesto por seis *pasos*:

1. `Selección de las bases de medición`: identificación de métricas, enfoques de medición e instrumentos de evaluación que faciliten la aplicación práctica del modelo de calidad.

2. `Descomposición ascendente:` a partir de las métricas seleccionadas, se construye una jerarquía de componentes, introduciendo los atributos de calidad como una capa intermedia entre métricas y características. Las métricas se asignan a atributos existentes o motivan la creación de nuevos. Este proceso implica la división, fusión o renombre de atributos, con el fin de mantener la coherencia estructural del modelo.

3. `Definición de impactos:` se establecen relaciones explícitas entre los atributos y las características de calidad, identificando los impactos positivos o negativos de cada atributo sobre una o más características. Estas relaciones deben justificarse explícitamente.

4. `Validación cruzada:` expertos del dominio revisan y consolidan la coherencia entre las descomposiciones descendentes y ascendentes. Este paso permite identificar inconsistencias, reasignar métricas, dividir atributos o asegurar una cobertura adecuada de todas las características de calidad.

5. `Operacionalización del modelo:` se definen concretamente los métodos o funciones de medición asociados a cada métrica, garantizando que el modelo sea ejecutable y aplicable en entornos reales.

6. `Evaluación de la cobertura:` se analiza la completitud del modelo verificando que las métricas disponibles cubran adecuadamente las características de calidad. Este proceso se apoya en reuniones iterativas con expertos de la industria y la academia.

Finalmente, Wagner et al. [21] sostiene que la adaptación efectiva de un modelo de calidad debe combinar tres tipos de métodos:

- `Métodos constructivos`: uso de estándares de calidad existentes, como ISO/IEC 25010, como punto de partida.
- `Métodos empíricos`: recolección de datos mediante SLR, encuestas, entrevistas, estudios de caso o experimentos con el fin de validar la relevancia, aplicabilidad y utilidad de los atributos de calidad seleccionados.
- `Métodos deductivos`: aplicación lógica y sistemática de conceptos y definiciones existentes del dominio para extender o modificar la estructura del modelo de calidad de manera coherente.

5.3. Enfoque mixto aplicado al modelo de calidad

La definición del modelo de calidad para servicios cloud se llevó a cabo mediante una meto2ºdología mixta, explícitamente alineada con el enfoque de Wagner et al. [21], el cual combina un proceso **descendente** (*top-down*) [34] y otro **ascendente** (*bottom-up*) [35], con el objetivo de conciliar las definiciones abstractas de calidad con su evaluación práctica mediante métricas y sus operacionalizaciones. La adopción de esta metodología resulta particularmente pertinente cuando se requiere adaptar un modelo genérico de calidad, como la ISO/IEC 25010 [1], a un dominio específico y altamente dinámico como los servicios cloud.

De acuerdo con los principios metodológicos de Wagner et al. [21], el proceso de construcción del modelo se estructuró como una secuencia de actividades interrelacionadas que integran evidencia conceptual y empírica. En primer lugar, se aplicó un enfoque **descendente**, tomando como punto de partida las ocho características y las treinta y una subcaracterísticas definidas por la norma ISO/IEC 25010, que proporcionan un marco conceptual ampliamente aceptado para la evaluación de la calidad interna y externa del software. No obstante, debido al carácter genérico del estándar, fue necesario llevar a cabo un proceso de reinterpretación y adaptación al dominio cloud, considerando sus particularidades técnicas y organizacionales

Esta adaptación descendente implicó analizar las características del estándar y los conceptos propios del dominio cloud, tales como la virtualización, la elasticidad, la multitenencia y los modelos de servicio (IaaS, PaaS y SaaS). Asimismo, se tuvieron en cuenta

los diferentes tipos de *stakeholders* involucrados —proveedores, consumidores, desarrolladores, facilitadores y usuarios finales—, lo que permitió refinar y reorganizar los atributos de calidad de acuerdo con las responsabilidades y expectativas asociadas a cada stakeholder. Como resultado, se definió una estructura jerárquica preliminar que vincula características, subcaracterísticas y atributos de calidad relevantes para el dominio cloud (véase Sección 2.1)

En el enfoque descendente, se consideraron las ocho características (eficiencia de desempeño, fiabilidad, seguridad, mantenibilidad, portabilidad, adecuación funcional, usabilidad y compatibilidad), desglosadas en 31 subcaracterísticas de la ISO/IEC 25010 para evaluar la calidad interna y externa del software, las cuales se utilizaron como marco de referencia inicial. No obstante, dado el carácter genérico del estándar, fue necesario un proceso de adaptación al dominio cloud, lo que sugiere la posibilidad de reinterpretar, ampliar e incorporar nuevas características, subcaracterísticas y atributos propios del dominio. Este proceso considera conceptos como la virtualización, los modelos de servicio (IaaS, PaaS y SaaS), así como los diferentes tipos de *stakeholders*, lo que orientó la descomposición y el refinamiento de los atributos específicos del dominio (véase Sección 2.1).

El enfoque **ascendente** complementó esta base conceptual al incorporar evidencia empírica derivada de la taxonomía de métricas de calidad cloud presentada en el capítulo anterior [2], obtenida mediante una revisión sistemática de la literatura. Esta taxonomía recopila y organiza 406 métricas, agrupadas en 240 atributos y 470 operacionalizaciones, proporcionando una visión amplia de cómo se aborda la medición de la calidad en estudios primarios. Dado que estas métricas son, en su mayoría, independientes de la plataforma, fue necesario analizarlas y reinterpretarlas para su aplicación en contextos concretos de servicios cloud.

La integración de la evidencia empírica permitió identificar métricas recurrentes, evaluar su frecuencia de uso y analizar su utilidad práctica, apoyándose en la revisión y retroalimentación de expertos del dominio e investigadores. Este proceso garantizó que los atributos de calidad definidos no solo estuvieran alineados con el estándar, sino que también reflejaran prácticas reales de medición utilizadas, al considerar estos aspectos.

1. `Conexión con la práctica`: cada atributo de calidad está respaldado por métricas utilizadas en estudios primarios, proporcionando evidencia observacional de cómo se aborda su medición en la práctica investigadora. No obstante, la mayoría de estas métricas aún requieren una validación empírica más sistemática.

2. `Granularidad y operacionalización`: las métricas y sus operacionalizaciones permiten pasar de definiciones abstractas a formas concretas de medición, incluyendo métodos, funciones y herramientas de monitorización.

La consolidación de los enfoques descendente y ascendente se llevó a cabo mediante un análisis temático sistemático [29]. Este análisis permitió mapear las métricas identificadas en la literatura a las características y subcaracterísticas de la ISO/IEC 25010, eliminando redundancias, resolviendo solapamientos semánticos y estableciendo correspondencias

explícitas entre características, atributos, métricas y sus operacionalizaciones. Como resultado, el modelo garantiza una trazabilidad bidireccional:

- De lo general a lo particular (característica de calidad → subcaracterística → atributo → métrica).
- De lo particular a lo general (métrica concreta → atributo → subcaracterística → característica

Este proceso de integración aportó dos beneficios clave. Por un lado, aseguró una conexión directa con la práctica, al respaldar cada atributo de calidad con métricas documentadas en estudios primarios. Por otro lado, permitió alcanzar un nivel adecuado de granularidad, facilitando la transición desde definiciones conceptuales de calidad hacia mecanismos concretos de medición, incluyendo métodos, funciones y herramientas de monitorización.

El modelo resultante fue sometido a un proceso *iterativo* de validación y refinamiento mediante la revisión de expertos e investigadores, con el fin de verificar su coherencia interna, su alineación con la ISO/IEC 25010 y su pertinencia para el dominio cloud. Este proceso permitió ajustar la estructura jerárquica del modelo y asegurar su capacidad de ser instanciado en distintos modelos de servicio y contextos organizacionales.

Finalmente, el modelo puede ser operacionalizado mediante la definición de instrumentos de evaluación concretos, materializados en métricas y sus respectivas operacionalizaciones, lo que facilita la automatización del proceso de evaluación y la reutilización del modelo en distintos escenarios.

En síntesis, la adopción del enfoque mixto de Wagner et al. [21] permitió estructurar un proceso sistemático, riguroso y replicable para la definición de un modelo de calidad para servicios cloud. Este enfoque fortaleció la robustez metodológica del modelo y contribuyó a superar una de las principales limitaciones de los modelos de calidad existentes: la desconexión entre las definiciones abstractas de calidad y las métricas concretas aplicables en escenarios reales. Al combinar la estructura conceptual de la ISO/IEC 25010 con la evidencia empírica derivada de la taxonomía de métricas, se obtuvo un marco integral, adaptable y orientado a la evaluación práctica de la calidad de los servicios cloud.

5.3.1. Consolidación mediante análisis temático

La consolidación entre el enfoque descendente, basado en la ISO/IEC 25010, y el enfoque ascendente, basado en la taxonomía de métricas, se apoyó en el **análisis temático** [29], un método riguroso y sistemático para organizar y sintetizar información cualitativa heterogénea. Este enfoque permitió identificar, analizar y agrupar patrones (temas) en los datos recopilados. En particular, se codificaron los temas recurrentes para cada característica, subcaracterística y atributo, y se clasificaron las métricas asociadas en consecuencia. El proceso se desarrolló de manera iterativa en tres pasos clave:

1. `Codificación`: identificación de temas iniciales a partir de las métricas recopiladas en la revisión sistemática, incluyendo atributos de calidad, fases del ciclo de vida, artefactos medidos y perspectivas de los *stakeholders*.

2. `Transformación de códigos en temas`: agrupación y alineación de las métricas con las características y subcaracterísticas de la ISO/IEC 25010. Es un proceso iterativo que analiza cada atributo y métrica en relación con su propósito de medición y su correspondencia conceptual con las características de calidad del estándar. Como resultado, se establecieron asociaciones preliminares entre métricas empíricas y atributos de calidad, facilitando la integración entre ambos enfoques metodológicos.

3. `Revisión y refinamiento de temas`: detección de solapamientos, eliminación de redundancias e inconsistencias. Se analizan y comparan los atributos de calidad y las métricas —incluyendo el nombre y la definición de la métrica, el método o la(s) función(es) de medición utilizada(s) y la unidad de medida—, así como los artefactos cloud en los que se aplican las métricas. Este análisis permitió identificar métricas sinónimas —aquellas con distintos nombres pero la misma definición (propósito)— y distinguir métricas homónimas —aquellas que comparten nombre pero difieren en su definición o contexto de aplicación—. El resultado fue un conjunto depurado de temas coherentes, cada uno vinculado de manera explícita con características, subcaracterísticas y atributos de calidad.

Este proceso de análisis temático permitió establecer una jerarquía clara que vincula las características abstractas de la ISO/IEC 25010 con métricas concretas de la literatura, asegurando la trazabilidad bidireccional del modelo. La Figura 5.1 muestra el proceso de análisis temático utilizado, mostrando cómo las métricas recuperadas de los estudios primarios fueron codificadas y agrupadas en temas, que luego fueron asociados con atributos de calidad y, finalmente, integrados en la jerarquía del modelo de calidad.

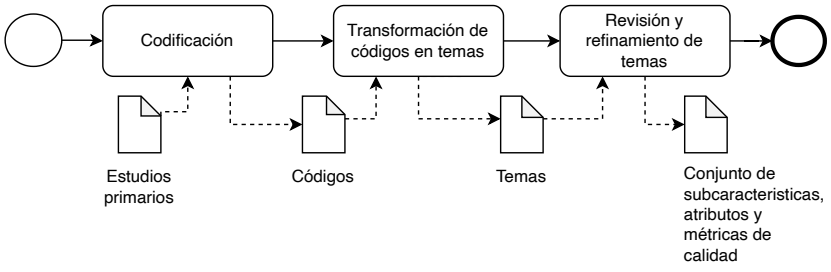


Figura 5.1: Proceso de descomposición e integración del modelo de calidad.

Para organizar y estructurar sistemáticamente estos elementos, se utilizó el metamodelo (véase Figura 4.4) que guía el desarrollo del modelo de calidad. Este metamodelo cumple dos funciones clave:

1. Definir los conceptos esenciales relacionados con la calidad (características, atributos, métricas, operacionalizaciones) y las relaciones entre ellos.

2. Incorporar los conceptos del dominio de los servicios cloud (*stakeholders*, tipos de servicios, fases del ciclo de vida y artefactos cloud) que determinan qué se mide, cómo se mide y quién se beneficia de los resultados de medición.

Como resultado de este procedimiento, se obtuvo un modelo de calidad integral, donde cada característica de la ISO/IEC 25010 se descompone progresivamente hasta alcanzar atributos de calidad y métricas (que pueden incluir distintas operacionalizaciones), aplicables a escenarios prácticos de evaluación de servicios cloud.

Adicionalmente, la *arquitectura propia de los servicios cloud* (caracterizada por su organización en capas funcionales y por la interacción entre componentes heterogéneos) ha influido de manera directa en la definición del modelo de calidad propuesto. Dado que los servicios cloud se estructuran típicamente en distintos niveles (SaaS, PaaS e IaaS), cada uno con responsabilidades, mecanismos de control y métricas específicas, el modelo de calidad refleja explícitamente esta estratificación.

La incorporación de las capas arquitectónicas en la jerarquía del modelo (aunque sea de forma implícita) permite asociar características, atributos y métricas de calidad al nivel donde se originan o se manifiestan, facilitando así una evaluación más precisa y contextualizada del servicio.

Por ejemplo, en el caso de la característica **Eficiencia de Desempeño**, el modelo permite descomponerla en subcaracterísticas y atributos que se materializan de forma distinta según la capa cloud considerada. En el nivel **SaaS**, esta característica puede evaluarse mediante métricas externas, como el tiempo de respuesta percibido por el usuario o el throughput de peticiones atendidas por la aplicación. En el nivel **PaaS**, la eficiencia de desempeño se relaciona con atributos como la latencia de los servicios gestionados, la eficiencia de los mecanismos de escalado automático o el tiempo de provisión de recursos. Finalmente, en el nivel **IaaS**, la misma característica se vincula a métricas de bajo nivel como la utilización de CPU, memoria, I/O de disco o ancho de banda de red. Esta diferenciación permite analizar cómo un problema de rendimiento observado en la capa de aplicación puede tener su origen en ineficiencias de las capas inferiores, favoreciendo así un diagnóstico más preciso de las causas raíz.

Esta alineación entre la arquitectura y modelo de calidad aporta una mayor flexibilidad, ya que permite instanciar, adaptar o extender el modelo de forma selectiva en función de la arquitectura concreta del servicio evaluado, sin comprometer su coherencia global. Asimismo, esta estructura favorece el mantenimiento y evolución del modelo en la práctica, al permitir incorporar nuevas tecnologías, componentes o métricas propias de una capa específica sin necesidad de redefinir el conjunto del modelo. De este modo, la consideración explícita de la arquitectura cloud refuerza la aplicabilidad del modelo y su capacidad de adaptación a escenarios reales y cambiantes.

5.4. Definición del modelo de calidad para servicios cloud

El modelo de calidad propuesto constituye una extensión del estándar ISO/IEC 25010 [1], adaptada al dominio de los servicios cloud e integrado con la evidencia empírica prove-

niente de la taxonomía de métricas presentada en el capítulo anterior. Su objetivo principal es proporcionar un marco estructurado, sistemático y operacionalizable que permita evaluar la calidad de los servicios cloud en escenarios prácticos.

La definición del modelo de calidad siguió el enfoque mixto descrito en la sección 5.3, y su desarrollo se estructuró en las siguientes fases:

1. `Enfoque descendente`: Se identificaron las características y subcaracterísticas derivadas del estándar ISO/IEC 25010 [1], a partir del análisis de sus ocho características y las 31 subcaracterísticas definidas en el modelo de calidad del producto.

2. `Enfoque ascendente`: Con base en la taxonomía de métricas para servicios cloud [2], se reconocieron 8 características y 28 subcaracterísticas de la ISO/IEC 25010, vinculadas con 240 atributos y 406 métricas.

3. `Integración de enfoques`: La información de ambos enfoques fue consolidada mediante un análisis temático, lo que permitió agruparla por temas y adaptarla al contexto de servicios cloud. En esta etapa, se mantuvieron 8 características y 24 subcaracterísticas reinterpretadas del estándar.

4. `Refinamiento de métricas y operacionalizaciones`: Con un enfoque ascendente, se analizaron las métricas y operacionalizaciones, obteniéndose un total de 407 métricas y 468 operacionalizaciones. Durante el proceso, se descartaron 2 operacionalizaciones debido a la insuficiencia en los métodos de medición y se incorporó una métrica para representar un concepto complementario.

5. `Refinamiento de subcaracterísticas y atributos`: Finalmente, se identificaron 44 subcaracterísticas adicionales que ampliaron el estándar, alcanzando una jerarquía de 68 subcaracterísticas. Estas fueron descompuestas en 242 atributos de calidad únicos, adaptados al dominio mediante análisis temático,incluyendo dos atributos adicionales respecto de la taxonomía original. El mapeo entre métricas, atributos y subcaracterísticas fue refinado, asegurando al menos una métrica vinculada a cada atributo y una correspondencia explícita entre subcaracterísticas y atributos.

Este proceso permitió alinear la estructura jerárquica de la ISO/IEC 25010 [1] con la evidencia empírica de métricas, asegurando trazabilidad entre los distintos niveles (característica, subcaracterística, atributo, métrica) y consolidando un modelo robusto y aplicable al dominio de los servicios cloud. El modelo trasciende las definiciones abstractas al sustentarse en métricas verificables y reproducibles.

Un aspecto clave es su **independencia de la plataforma**, que permite instanciar métricas y operacionalizaciones en diferentes entornos y tipos de servicio (IaaS, PaaS, SaaS). Esto favorece su aplicación en escenarios heterogéneos y asegura su pertinencia para distintos proveedores y contextos industriales. Asimismo, el modelo considera a los distintos *stakeholders* involucrados (proveedores, consumidores, desarrolladores, facilitadores y usuarios finales), de modo que cada uno pueda identificar métricas relevantes para sus necesidades específicas.

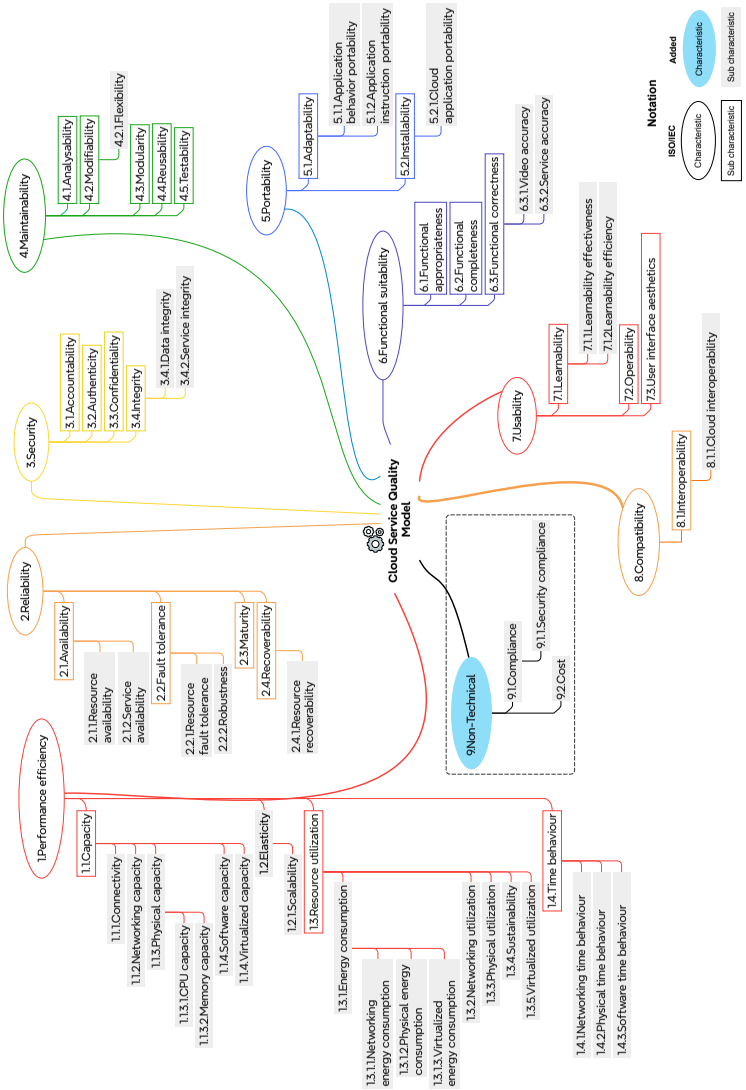


Figura 5.2: Modelo de calidad del servicio cloud: características y subcaracterísticas.

La Figura 5.2 presenta la estructura resultante, mostrando los niveles jerárquicos y las relaciones entre características, subcaracterísticas, atributos, métricas y operacionalizaciones. En ella se evidencia cómo se integran los elementos de la ISO/IEC 25010 [1], la taxonomía de métricas y las extensiones introducidas.

En síntesis, el modelo define un marco para la evaluación de la calidad en servicios cloud, que combina la solidez conceptual del estándar ISO/IEC 25010 [1] con la evidencia empírica recopilada en la literatura. Además, facilita la comparación y reutilización de métricas, fomenta un lenguaje común entre investigadores y profesionales, fortaleciendo la adopción de prácticas sistemáticas de evaluación de calidad en el contexto cloud.

5.4.1. Determinación de las características y subcaracterísticas

La selección de las características y subcaracterísticas incluidas en el modelo no es arbitraria, sino que responde a un proceso de derivación fundamentado en tres criterios: (1) alineamiento explícito con las definiciones del estándar ISO/IEC 25010, (2) adecuación al contexto específico de los servicios cloud identificada a partir del análisis conceptual del Capítulo 2 y de la taxonomía de métricas desarrollada en el Capítulo 4, y (3) evidencia empírica sobre las dimensiones de calidad que influyen de manera más significativa en la prestación de servicios cloud en entornos industriales.

De este modo, se seleccionan únicamente aquellas características y subcaracterísticas cuya presencia se justifica tanto por su relevancia teórica como por su correspondencia con métricas relevantes en sistemas cloud reales.

Este proceso excluye características que, si bien relevantes en otros escenarios de software, no poseen manifestaciones observables ni métricas asociadas en el contexto cloud (p. ej., usabilidad interna o accesibilidad del software). El resultado es un conjunto de características y subcaracterísticas que representan de forma más fiel y operativa los factores que determinan la calidad de un servicio cloud.

El resultado ha sido incorporar todas las características de calidad definidas en la ISO/IEC 25010 [1], junto con una característica adicional denominada *No Técnica*, que permite abordar aspectos más allá de los factores estrictamente relacionados con la calidad. Asimismo, se introducen nuevas subcaracterísticas específicas del dominio, no contempladas en el estándar, que permiten una descomposición más detallada y representan mejor los atributos clave relacionados con la calidad del servicio cloud.

A continuación, se presenta una descripción detallada de estas características y subcaracterísticas.

1. **Eficiencia de desempeño**: esta característica se refiere al grado en que un servicio cloud realiza sus funciones previstas dentro de las restricciones de tiempo y rendimiento especificadas, al tiempo que utiliza de manera eficiente recursos como la CPU, la memoria, el almacenamiento, el ancho de banda de la red y la energía en condiciones especificadas. Analizamos este concepto teniendo en cuenta varias capas de abstracción en modelos de tipo de servicio, como la red, el almacenamiento, el servidor y la virtualización, los sistemas operativos y el tiempo de ejecución, las aplicaciones y los datos. Se identificaron cuatro subcaracterísticas: tres incluidas en

la ISO/IEC 25010 y una subcaracterística recientemente introducida, la elasticidad, que es específica del dominio cloud. En su caso, cada subcaracterística se descompuso aún más, como se muestra en la Tabla 5.1.

Tabla 5.1: Subcaracterísticas de la eficiencia de desempeño.

Subcaracterística	**Definición**
CAPACIDAD	Grado en que los recursos máximos disponibles de un servicio cloud (p. ej., almacenamiento, red) son suficientes para satisfacer los requisitos establecidos.
Conectividad	Habilidad del servicio para mantener canales de comunicación de red estables y con niveles aceptables de calidad, garantizando la transmisión de datos entre aplicaciones y recursos.
Capacidad de red	Disponibilidad de los recursos de red para permitir atender las demandas del servicio cloud.
Capacidad física	Disponibilidad de recursos de *hardware*, como CPU y servidores, necesarios para soportar la operación del servicio.
Capacidad del software	Grado en que las aplicaciones y plataformas del servicio disponen de recursos suficientes para su correcto funcionamiento.
Capacidad virtualizada	Disponibilidad de recursos asignados a entornos virtuales (p. ej., memoria y CPU en máquinas virtuales) para cubrir los requerimientos de uso.
ELASTICIDAD	Grado de automatización con el que un servicio cloud ajusta sus recursos de forma eficiente y rápida en función de las variaciones en la demanda [83].
Escalabilidad	Capacidad del servicio cloud para soportar incrementos en la carga de trabajo mediante la incorporación de recursos adicionales, sin restricciones temporales [83].
UTILIZACIÓN DE RECURSOS	Grado en que los recursos utilizados (en cantidad y tipo) por el servicio cloud son adecuados para ejecutar sus funciones conforme a los requisitos definidos.
Consumo de energía	Eficiencia en el uso de energía y en la reducción de la huella de carbono se descomponen en consumo físico, virtual y de red.
Utilización física	Proporción de recursos de hardware efectivamente usados por el servicio cloud.
Utilización virtualizada	Proporción de recursos asignados a entornos virtuales efectivamente consumidos.
Utilización de redes	Grado de aprovechamiento de la capacidad de transmisión y ancho de banda.
Sostenibilidad	Capacidad del servicio para evolucionar y adaptarse a los requisitos a lo largo del tiempo sin necesidad de cambios sustanciales [127].
COMPORTAMIENTO TEMPORAL	Capacidad de un servicio cloud para cumplir con los requisitos relacionados con el tiempo, como respuesta, procesamiento y rendimiento, durante la ejecución de sus funciones.
Comportamiento temporal físico	Desempeño de los recursos de hardware respecto a los tiempos de operación del servicio cloud exigidos.
Comportamiento temporal del software	Capacidad de las aplicaciones y sistemas de software para cumplir con los requisitos temporales de la operación.
Comportamiento temporal de red	Desempeño de la infraestructura de red en términos de latencia, ancho de banda y tiempos de transmisión.

2. **Fiabilidad**: Esta característica se refiere a la capacidad de un servicio cloud para ejecutar sus funciones especificadas en condiciones definidas y durante un período determinado. Se descompone inicialmente en las cuatro subcaracterísticas de la ISO/IEC 25010 —disponibilidad, tolerancia a fallos, madurez y capacidad de recuperación— que reflejan los aspectos clave de la fiabilidad operativa. Estas se refinan, cuando es pertinente, en subcaracterísticas adicionales que permiten evaluar la confiabilidad de recursos cloud específicos. La Tabla 5.2 presenta la jerarquía resultante, adaptada al dominio cloud con nueve subcaracterísticas.

Tabla 5.2: Subcaracterísticas de fiabilidad.

Subcaracterística	**Definición**
DISPONIBILIDAD	Capacidad de un servicio cloud para permanecer operativo y ofrecer acceso ininterrumpido a sus funcionalidades cuando sea necesario [127].
Disponibilidad de recursos	Nivel en que un sistema o componentes de un servicio cloud (p. ej., servidores, redes, almacenamiento) mantienen un estado operativo y accesible [136].
Disponibilidad del servicio	Grado de disponibilidad en que el servicio cloud, como conjunto integrado de recursos y funciones, asegura la continuidad de uso para el usuario final.
TOLERANCIA A FALLOS	Capacidad del servicio cloud para sostener su funcionamiento y accesibilidad frente a anomalías o fallos en componentes individuales.
Tolerancia a fallos de recursos	Grado en que un sistema o componente específico puede mantener su operación frente a fallos parciales o eventos inesperados.
Robustez	Capacidad de un servicio cloud para evitar la degradación significativa de su funcionamiento pese a errores en uno o varios de sus componentes.
MADUREZ	Nivel en que un servicio cloud demuestra confiabilidad consistente a lo largo del tiempo y bajo condiciones de operación normales.
CAPACIDAD DE RECUPERACION	Habilidad de un servicio cloud para restaurar su estado deseado y recuperar los datos afectados tras una interrupción o fallo en alguno de sus componentes.
Recuperación de recursos	Capacidad de un sistema o componente de un servicio cloud puede recuperarse y restablecer su estado anterior.

3. **Seguridad**: Esta característica describe la capacidad del servicio cloud para proteger la información y los datos frente a accesos o modificaciones no autorizados. En el contexto cloud, la seguridad abarca múltiples dimensiones: a nivel de aplicación; entre inquilinos (*tenant-to-tenant*), cuando varios usuarios comparten la misma infraestructura; y entre proveedor e inquilino (*provider-to-tenant*), donde el proveedor debe garantizar la protección de los datos de sus clientes. Conforme a la ISO/IEC 25010, se descompone en cuatro subcaracterísticas—responsabilidad, autenticidad, confidencialidad e integridad—que capturan dimensiones esenciales de la seguridad. Adicionalmente, en el dominio cloud, la integridad se especializa en dos dimensiones: la integridad de los datos y la integridad del servicio, como se muestra en la Tabla 5.3.

Tabla 5.3: Subcaracterísticas de seguridad.

Subcaracterística	**Definición**
RESPONSABILIDAD	Capacidad de registrar, rastrear y asociar las acciones realizadas en el servicio cloud con los actores que las ejecutan, facilitando la auditoría y la trazabilidad.
AUTENTICIDAD	Capacidad de un servicio cloud para verificar la identidad de entidades (usuarios, dispositivos o recursos) y garantizar que correspondan a quienes afirman ser.
CONFIDENCIALIDAD	Propiedad de un servicio cloud que asegura que los datos y recursos solo estén disponibles para usuarios o sistemas autorizados, mediante mecanismos como control de acceso, cifrado y gestión de credenciales.
INTEGRIDAD	Capacidad de un servicio cloud para preservar la exactitud, consistencia y confiabilidad de los datos y aplicaciones, protegiéndolos contra modificaciones o manipulaciones no autorizadas. Se descompone en:

Tabla 5.3 (continúa).

Subcaracterística	**Definición**
Integridad de los datos	Grado en que un servicio cloud garantiza la no alteración, pérdida o corrupción de los datos durante su almacenamiento, transmisión o procesamiento.
Integridad del servicio	Grado en que un servicio cloud mantiene la confiabilidad de su operación frente a ataques o fallos, asegurando la continuidad y consistencia de sus funciones.

4. **Mantenibilidad**: Esta característica describe la capacidad de un servicio cloud para modificarse de forma eficiente y efectiva con el fin de satisfacer necesidades correctivas, adaptativas o perfectivas. Dado que estos servicios están compuestos por múltiples activos —como máquinas virtuales, instancias de base de datos, microservicios—, los cambios pueden aplicarse a cualquiera de ellos. Según la ISO/IEC 25010 [1], se descompone en cuatro subcaracterísticas: analizabilidad, modificabilidad, reutilización y capacidad de prueba, cada una de las cuales captura un aspecto clave de la mantenibilidad. Para el dominio cloud, la modificabilidad se especializa en dos dimensiones: la flexibilidad para adaptarse a los cambios solicitados por el cliente y el impacto de dichos cambios en los activos interconectados. La Tabla 5.4 presenta esta descomposición.

Tabla 5.4: Subcaracterísticas de Mantenibilidad.

Subcaracterística	**Definición**
ANALIZABILIDAD	Capacidad para evaluar el impacto de cambios en uno o más activos de un servicio cloud, así como para diagnosticar deficiencias o identificar las causas de fallos.
MODIFICABILIDAD	Capacidad de un servicio cloud para incorporar cambios de forma eficaz y eficiente, minimizando la introducción de defectos y evitando la degradación de la calidad del servicio. Se descompone en dos subcaracterísticas.
Flexibilidad	Capacidad de un servicio cloud para adaptarse a cambios solicitados por el cliente.
Modularidad	Grado en el que un servicio cloud está compuesto por activos independientes, de modo que los cambios en uno de ellos tienen un impacto mínimo en los demás.
REUTILIZACIÓN	Grado en que un activo de un servicio cloud puede emplearse en diferentes servicios o servir como base para construir nuevos activos.
CAPACIDAD DE PRUEBA	Facilidad con la que se pueden establecer criterios de prueba y ejecutar verificaciones que confirmen el cumplimiento de los requisitos de calidad del servicio cloud de forma eficiente y efectiva.

5. **Portabilidad**: Se refiere a la capacidad de un servicio cloud para trasladarse de manera efectiva y eficiente entre diferentes entornos de hardware, software, operativos o de uso, independientemente de sus diferencias. Esta característica se descompone inicialmente en dos subcaracterísticas del estándar ISO/IEC 25010 —adaptabilidad e instalabilidad—, mientras que la reemplazabilidad se excluye del modelo propuesto debido a la falta de métricas específicas en la literatura aplicables al dominio cloud. En este contexto, las subcaracterísticas se refinan para evaluar la portabilidad de las instrucciones y la portabilidad del comportamiento de la aplicación, proporcionando una visión más precisa de la migración y adaptación de servicios cloud. Los resultados de esta descomposición se presentan en la Tabla 5.5.

Tabla 5.5: Subcaracterísticas de portabilidd.

Subcaracterística	**Definición**
ADAPTABILIDAD	Grado en el que un servicio cloud puede ajustarse de manera eficiente a diferentes entornos de hardware, software u operativos, así como a entornos en evolución. Se descompone en dos subcaracterísticas.
Portabilidad del comportamiento de la aplicación	Grado en el que la ejecución de un servicio cloud en el entorno de destino mantiene resultados consistentes a los obtenidos en el entorno de origen [141].
Portabilidad de las instrucciones de la aplicación	Grado en que el conjunto de instrucciones de la aplicación se ejecuta correctamente en el entorno de destino [141].
INSTALABILIDAD	Esfuerzo requerido para implementar y preparar un servicio cloud para su uso. Refleja la facilidad con la que un servicio puede ser migrado (portado) desde un sistema de origen a un sistema de destino, o una aplicación entre proveedores de nube, o desde la infraestructura de un consumidor hacia un servicio cloud [141].
Portabilidad de aplicaciones cloud	Capacidad para trasladar una aplicación o cualquiera de sus componentes entre diferentes servicios cloud, o entre el sistema de un cliente y un servicio cloud [139].

6. **Adecuación funcional**: Se refiere a la capacidad de un servicio cloud para ofrecer funciones que satisfagan las necesidades explícitas e implícitas de los usuarios en condiciones específicas. Esta característica se descompone en tres subcaracterísticas de la ISO/IEC 25010 —pertinencia, completitud y corrección funcional—que evalúan si el servicio proporciona un conjunto de funciones completo, relevante y preciso. Para reflejar las particularidades del dominio cloud, la corrección funcional se refina para considerar tanto la satisfacción del usuario con la exactitud del servicio como la precisión de sus salidas, por ejemplo, en los resultados de análisis de precisión de video. La Tabla 5.6 presenta esta descomposición.

Tabla 5.6: Subcaracterísticas de adecuación funcional.

Subcaracterística	**Definición**
COMPLETITUD FUNCIONAL	Grado en que el conjunto de funciones del servicio cloud cubre todas las tareas especificadas y los objetivos definidos por el usuario.
CORRECCIÓN FUNCIONAL	Grado en que el servicio cloud entrega resultados exactos y confiables conforme al grado de precisión requerido. Se descompone en dos subcaracterísticas.
Precisión del servicio	Nivel de cumplimiento de los resultados del servicio cloud respecto a la exactitud establecida en el SLA.
Precisión de video	Nivel de exactitud en los resultados de procesamiento o análisis de vídeo ofrecidos por el servicio cloud.
PERTINENCIA FUNCIONAL	Grado en que las funciones del servicio cloud son relevantes y útiles para la consecución de las tareas y objetivos del usuario.

7. **Usabilidad**: Se refiere a la capacidad de un servicio cloud para ser entendido, aprendido, operado y percibido como atractivo por los usuarios en un contexto de uso específico. Inicialmente se descompone en tres subcaracterísticas de la ISO/IEC 25010: capacidad de aprendizaje, operatividad y estética de la interfaz de usuario. Aunque el

estándar contempla otras subcaracterísticas —accesibilidad, protección contra errores del usuario —, estas no se incluyeron en nuestro modelo debido a la falta de métricas adaptadas al dominio cloud. Las subcaracterísticas seleccionadas permiten evaluar la facilidad de aprendizaje y operación del servicio, así como la efectividad y atractivo de su interfaz. Para capturar mejor los requisitos del dominio, la capacidad de aprendizaje se refina para analizar tanto la eficiencia como la eficacia en el proceso de aprendizaje, como se detalla en la Tabla 5.7

Tabla 5.7: Subcaracterísticas de usabilidad.

Subcaracterística	**Definición**
CAPACIDAD DE APRENDIZAJE	Esfuerzo requerido para aprender y comprender el uso de un servicio cloud, considerando eficacia, eficiencia, reducción de riesgos y satisfacción del usuario.
Eficiencia de aprendizaje	Grado en que el servicio puede ser aprendido y comprendido con el mínimo tiempo, riesgo y esfuerzo.
Eficacia de aprendizaje	Grado en que los usuarios logran aprender y comprender el servicio con éxito.
OPERABILIDAD	Esfuerzo requerido para operar y controlar las funciones del servicio cloud.
ESTÉTICA DE LA INTERFAZ DE USUARIO	Capacidad de la interfaz para ofrecer una experiencia visualmente atractiva y satisfactoria para el usuario.

8. **Compatibilidad**: Se refiere a la capacidad de dos o más servicios cloud para intercambiar información y ejecutar sus funciones en un mismo entorno de *hardware* o *software*. Se descompone en la subcaracterística de interoperabilidad, que captura el grado en que los servicios cloud pueden compartir y utilizar información de manera conjunta. Para responder a las particularidades del dominio cloud, la interoperabilidad se refina en dimensiones específicas que se muestran en la Tabla 5.8. La subcaracterística de coexistencia no se incluyó debido a la falta de métricas aplicables al dominio cloud.

Tabla 5.8: Subcaracterísticas de compatibilidad.

Subcaracterística	**Definición**
INTEROPERABILIDAD	Capacidad de dos o más sistemas o aplicaciones para intercambiar información y utilizarla de manera efectiva [47]. En el dominio cloud, se distinguen dos escenarios principales: *a*) Una aplicación que se ejecuta en un sistema consumidor e intercambia información con un servicio cloud y la utiliza de manera funcional, y *b*) Un servicio cloud que colabora con otros servicios cloud, ya sea del mismo proveedor o de uno diferente. Esta capacidad también se extiende a tres tipos de interfaces del servicio cloud: funcional, administrativa y empresarial, que pueden operar de manera independiente [139].

Tabla 5.8 (continúa).

Subcaracterística	**Definición**
Interoperabilidad en la nube	Capacidad del sistema de servicios cloud de un cliente para interactuar con otros servicios cloud o la capacidad de un servicio cloud para interactuar con otros servicios cloud mediante el intercambio de información [141].

9. **No técnica**: esta nueva característica se refiere a la capacidad del servicio cloud para adaptarse a las características, necesidades o condiciones no técnicas establecidas por el cliente. Si bien estos factores no están directamente relacionados con las especificaciones técnicas del servicio, pueden influir significativamente en el proceso de toma de decisiones al seleccionar un proveedor de nube o medir la calidad de un servicio cloud. Varios factores no técnicos pueden ser relevantes, pero esta característica se descompone en dos subcaracterísticas —cumplimiento y costo—. Los resultados se muestran en la Tabla 5.9. Si bien la calidad generalmente busca satisfacer las expectativas del cliente, el cumplimiento busca cumplir con los requisitos regulatorios. Por el momento, nuestro modelo de calidad se centra en el cumplimiento de la seguridad midiendo la capacidad de un servicio cloud para cumplir con los requisitos de seguridad. El modelo podría ampliarse para incluir otros atributos si fuera necesario. Además, nuestro modelo incluye el coste como una característica no técnica relevante, ya que existe una relación directa entre el nivel de calidad y el coste incurrido en su entrega. Medir el coste de la calidad es esencial para que las empresas obtengan información sobre su desempeño de calidad e identifiquen áreas de mejora.

Tabla 5.9: Subcaracterísticas no técnicas.

Subcaracterística	**Definición**
CUMPLIMIENTO	Capacidad del servicio cloud para cumplir con regulaciones, certificaciones, estándares, marcos o mejores prácticas relacionadas con los sistemas cloud, los servicios cloud o sus componentes. El alcance del cumplimiento puede ser legal, financiero, de seguridad de la información, etc.
Cumplimiento de la seguridad	Capacidad de un servicio cloud para cumplir con las regulaciones, certificaciones, marcos o mejores prácticas recomendadas y relacionadas con la seguridad de los sistemas cloud o sus componentes.
COSTE	Grado en que el servicio cloud es económicamente alcanzable y sostenible. El objetivo es mantener el coste dentro del presupuesto.

5.4.2. Descomposición en atributos de calidad

Esta sección se centra en desglosar los **atributos**, que representan propiedades observables y específicas de las características y subcaracterísticas. Los atributos cumplen un papel fundamental en la transición desde lo abstracto hacia lo medible, ya que describen de manera concreta qué aspecto de la calidad se evalúa, además, garantizan la trazabilidad de las entidades de dominio de servicio cloud para una mejor evaluación de la calidad.

Por ejemplo, la disponibilidad se puede definir como la proporción de tiempo en que un servicio cloud permanece operativo y accesible para los usuarios cuando lo necesitan. Si

se considera una puerta de enlace de API (*gateway*) en la nube que enruta las solicitudes a los servicios de *backend*, su disponibilidad refleja el porcentaje de tiempo que puede aceptar y reenviar con éxito las solicitudes entrantes sin errores.

Las definiciones de los atributos se refinaron cuidadosamente mediante el análisis temático para mejorar la claridad conceptual y garantizar una alineación coherente con las subcaracterísticas identificadas en la sección anterior. Es importante destacar que el modelo de calidad resultante es independiente de la plataforma, lo que significa que se puede aplicar independientemente de la plataforma cloud en la que se implemente un servicio. El conjunto final consta de 242 atributos de calidad medibles relevantes para los servicios cloud. A continuación, presentamos un análisis de los atributos más representativos de cada característica y subcaracterística de calidad. La descomposición completa se puede encontrar en el repositorio que acompaña a este trabajo en `https://bit.ly/frameworkCloudQuality`

1. **Eficiencia del desempeño**: Esta característica incluye el mayor número de métricas y, en consecuencia, una alta concentración de atributos de calidad con 141 identificados, lo que representa el 58.3 % del total. Esta distribución refleja el principio fundamental de que los servicios cloud deben administrar eficientemente los recursos, optimizar el rendimiento y maximizar la capacidad en función del tipo de servicio.

   La distribución de los atributos entre las subcaracterísticas es la siguiente: 50 para el comportamiento temporal, 43 para la utilización de recursos, 28 para la capacidad y 20 para la elasticidad. En la Tabla 5.10 se resumen los atributos más representativos en función de su frecuencia de aparición, e incluye al menos un atributo para cada subcaracterística.

   Se identificó un número particularmente alto de atributos de calidad para el *comportamiento temporal*, ya que el rendimiento en la computación en la nube a menudo se evalúa por la rapidez con la que un servicio cloud puede procesar las solicitudes. Para la *utilización de recursos*, los atributos se centran en el grado de uso de los recursos (p. ej., memoria, CPU, máquinas virtuales, comunicación, infraestructura) y el consumo de energía, un área cada vez más importante para los esfuerzos de *sostenibilidad*. Estos atributos tienden a ser de bajo nivel y, por lo general, se monitorizan mediante herramientas integradas proporcionadas por plataformas cloud, que ofrecen una alta granularidad. En el caso de la *capacidad*, los atributos representan la disponibilidad y adecuación de los recursos para satisfacer las demandas esperadas (p. ej., ancho de banda, capacidad del servidor). En el caso de la *elasticidad*, los atributos reflejan la rapidez y la eficacia con las que se pueden aumentar o reducir los recursos en respuesta a los cambios en la carga de trabajo, incluidos el tiempo necesario para ajustarse y la sobrecarga de recursos en la que se incurre durante el escalado.

Tabla 5.10: Atributos de eficiencia de desempeño.

Subcaracterística	**Atributo**	**Definición**
CAPACIDAD		

Table 5.10 (continúa)

Subcaracterística	**Atributo**	**Definición**
	Capacidad del servicio cloud	Límite de solicitudes de servicio que un servicio cloud puede procesar en un período determinado bajo condiciones de rendimiento específicas.
	Capacidad aprovisionada	Límite de cómputo, almacenamiento o recursos de red asignados explícitamente a una instancia del servicio cloud.
	Capacidad de cómputo	Límite de procesamiento disponible para un servicio cloud sin necesidad de escalar ni reconfigurar.
Conectividad	Conexiones	Límite de clientes o conexiones simultáneas admitidas por el servicio cloud.
Capacidad de red	Ancho de banda	Límite de transferencia de datos disponible o utilizada por un servicio cloud a través de una conexión de red.
	Paquetes de entrada	Límite de paquetes de red entrantes que un servicio cloud puede recibir en una unidad de tiempo definida.
	Paquetes de salida	Límite de paquetes de red salientes que un servicio cloud puede transmitir en una unidad de tiempo definida.
Capacidad física	Capacidad del servidor	Límite de cargas de trabajo o procesos que se pueden ejecutar simultáneamente en un servidor físico dentro de los centros de datos del proveedor.
	Capacidad de almacenamiento	Límites de los recursos de almacenamiento que puede proporcionar un servicio cloud, incluyendo el tamaño, aspectos como el rendimiento de lectura/escritura y las operaciones de E/S por segundo.
Capacidad de software	Capacidad de aplicación	Límite de solicitudes de servicio que una aplicación puede procesar en una unidad de tiempo definida (p. ej., solicitudes por minuto) manteniendo unos niveles de rendimiento aceptables.
Capacidad virtualizada	VM capacidad de CPU	Límite de potencia de procesamiento asignado a una VM, expresado en CPU virtuales (vCPU) o ciclos de procesamiento.
ELASTICIDAD	Aprovisionamiento excesivo	provisión de recursos cloud que supera los requisitos reales de la carga de trabajo.
	Aprovisionamiento insuficiente	provisión de recursos cloud por debajo de los requisitos reales de la carga de trabajo.
Escalabilidad	Escalabilidad del servicio	Capacidad del servicio cloud para aumentar o disminuir sus instancias de servicio disponibles en respuesta a los cambios en la carga de trabajo.
	Escalabilidad de recursos	Capacidad de un servicio cloud para aumentar o disminuir sus recursos disponibles (p. ej., CPU o memoria) en respuesta a los cambios en la carga de trabajo.
	Escalabilidad vertical	Fluctuación máxima permitida de la capacidad asignada a una sola instancia (p. ej., más CPU o memoria).
	Escalabilidad horizontal	Capacidad máxima permitida para aumentar o disminuir el número de instancias de los recursos de un servicio cloud.
	Cobertura de escalabilidad	Capacidad del servicio cloud para adaptarse a las demandas y necesidades de los clientes dentro de los límites de sus capacidades de escalabilidad.
UTILIZACIÓN DE RECURSOS	Eficacia	Comprensión de lo bien que un servicio cloud utiliza los recursos (e.g., cómputo y almacenamiento) para proporcionar la funcionalidad requerida.
Consumo de energía	Rendimiento energético	Capacidad de minimizar el consumo de energía para operar con el mismo rendimiento y funcionalidad de los servicios.
Utilización física	Utilización del disco	Proporción de la capacidad del disco utilizada activamente para operar y mantener el servicio.

Table 5.10 (continúa)

Subcaracterística	**Atributo**	**Definición**
	Transmisión de vídeo	Grado en que un servicio cloud ofrece contenido de vídeo con un nivel de calidad que se aproxima a la resolución máxima y las capacidades de visualización del dispositivo del usuario final, teniendo en cuenta las variaciones en las condiciones de la red, el tamaño de la pantalla del dispositivo y la duración de la reproducción.
Utilización virtual	Utilización de máquinas virtuales	Grado en que los recursos asignados de procesamiento, memoria, almacenamiento y red de una VM se consumen activamente durante el funcionamiento.
Utilización de red	Utilización de conexiones	Proporción de conexiones de red activas en uso en comparación con el total de conexiones admitidas por el servicio cloud.
Sostenibilidad	Sostenibilidad del servicio	Capacidad de un servicio cloud para seguir siendo funcional y reutilizable a lo largo del tiempo sin modificaciones, incluso cuando cambian las condiciones de los requisitos de los usuarios [127].
COMPORTAMIENTO TEMPORAL	Tiempo de respuesta	Tiempo transcurrido entre la recepción de una solicitud por parte de un servicio y la entrega de la respuesta correspondiente, medido únicamente cuando el servicio está operativo y disponible.
	Latencia	Intervalo de tiempo entre la presentación de la solicitud de un usuario y el inicio de su tramitación por parte del servicio competente.
	Tiempo de ejecución	Tiempo que tarda un servicio cloud en procesar una solicitud desde el momento en que comienza la ejecución hasta que se obtiene el resultado, sin incluir los retrasos en la transmisión.
Comportamiento temporal físico	Transferencia de datos	Tiempo que se tarda en transferir datos entre dos servicios o componentes cloud, medido desde el inicio de la transferencia hasta que se entrega con éxito el último byte.
Comportamiento temporal del software	Densidad de aplicación	Número de servidores de aplicaciones que residen en un sistema.
Comportamiento temporal de red	Velocidad de red	Cantidad de datos transmitidos con éxito a través de la red por unidad de tiempo durante un periodo definido.

2. **Fiabilidad**: Esta característica es la segunda más representativa en cuanto al número de atributos de calidad, con 39 atributos identificados, lo que supone el 16.1 % del total. Su importancia radica en el gran énfasis que los *stakeholders* ponen en garantizar que los servicios cloud sean altamente disponibles y confiables. Estos objetivos son fundamentales para mantener la confianza y la satisfacción de los clientes, así como para prevenir posibles pérdidas de ingresos. La distribución de los atributos entre las subcaracterísticas es la siguiente: 11 para la disponibilidad, 10 para la capacidad de recuperación, 9 para la tolerancia a errores, 1 para la madurez y 8 para la fiabilidad propiamente dicha. Un resumen de los atributos más representativos se proporciona en Tabla 5.11.

   La *disponibilidad* es muy valorada por los clientes, ya que garantiza el acceso continuo y el funcionamiento ininterrumpido del servicio cloud. A menudo es un factor clave en las negociaciones de SLA entre los proveedores y los consumidores. La *capacidad de recuperación* es crucial para los consumidores, ya que cualquier tiempo de inactividad del servicio puede tener importantes repercusiones comerciales, como daños a la reputación y pérdidas financieras. La *tolerancia a fallos* desempeña un papel esencial

en la gestión de anomalías o errores que puedan surgir en cualquier componente de la infraestructura del servicio cloud, incluidas las máquinas físicas, las redes o los sistemas de software. Por último, la *madurez* refleja la capacidad del servicio cloud para ofrecer sistemáticamente funciones completas y fiables en condiciones operativas normales.

Tabla 5.11: Atributos de fiabilidad.

Subcaracterística	**Atributo**	**Definición**
DISPONIBILIDAD		
Disponibilidad de recursos	Disponibilidad de VM	Proporción de tiempo en que una VM y sus recursos asignados permanecen operativos y accesibles para ejecutar las cargas de trabajo según lo previsto.
Disponibilidad del servicio	Disponibilidad del servicio	Grado en que un servicio cloud permanece operativo y accesible para los usuarios, permitiendo la ejecución de sus funciones según lo esperado.
	Tiempo de disponibilidad	Duración acumulada durante la cual un servicio cloud está operativo y accesible para realizar las tareas previstas dentro de un periodo de medición.
TOLERANCIA A FALLOS	Tiempo medio entre fallos	Intervalo medio de tiempo durante el cual un servicio cloud funciona sin interrupciones antes de producirse un fallo.
Tolerancia a fallos de recursos	Tasa de pérdida de I-frame	Proporción de I-frames perdidos respecto del total de I-frames enviados en una transmisión de video.
Robustez	Robustez del servicio	Probabilidad de que un servicio cloud mantenga su funcionamiento sin degradación ante fallos de alguno de sus componentes.
MADUREZ	Operación exitosa	Capacidad del servicio cloud para completar correctamente una tarea o trabajo en condiciones normales de operación.
CAPACIDAD DE RECUPERACIÓN	Resiliencia	Grado en que un servicio cloud puede recuperarse y restaurar su funcionamiento tras un fallo o desastre.
Recuperación de recursos	Pérdida de datos	Proporción de datos perdidos que se pueden restaurar con éxito tras un fallo o desastre respecto a los datos afectados.

3. **Seguridad**: La seguridad es una preocupación primordial para las organizaciones que adoptan servicios cloud, ya que está directamente relacionada con el cumplimiento normativo, la protección de los recursos y la fiabilidad general de la infraestructura cloud donde se alojan las aplicaciones y se almacenan los datos de los clientes. Se identificó un total de 19 atributos bajo esta característica, lo que representa el 8 % del total. Estos atributos se distribuyen a través de las subcaracterísticas de la siguiente manera: 5 para la integridad, 4 para la confidencialidad, 3 para la responsabilidad (rendición de cuentas), 3 para la autenticidad y 4 para la seguridad en sí misma. En la Tabla 5.12 se presenta un resumen de los atributos más representativos.

   En el contexto de la adopción de servicios cloud, la *integridad* garantiza que el acceso y las modificaciones a los datos y cálculos se controlen correctamente. Por ejemplo, las empresas que migran a la nube sus servicios deben tener en cuenta amenazas como los ataques a la capa de aplicaciones y la corrupción de datos. La *confidencialidad* requiere mecanismos de seguridad sólidos para proteger los datos confidenciales. La

tenencia múltiple, un modelo común en entornos cloud donde las máquinas virtuales comparten *hardware* físico, puede introducir vulnerabilidades si no se gestiona adecuadamente. La *responsabilidad* es esencial en entornos compartidos, lo que requiere controles efectivos de auditoría y monitorización para rastrear las actividades de los usuarios y garantizar la trazabilidad. Por último, la *autenticidad* desempeña un papel fundamental en la aplicación de controles de acceso a los recursos cloud, como la infraestructura, los datos y las aplicaciones, lo que garantiza que solo los usuarios autorizados puedan acceder a los activos confidenciales en un entorno compartido.

Tabla 5.12: Atributos de seguridad.

Subcaracterística	**Atributo**	**Definición**
RESPONSABILIDAD	Datos peligrosos	Identificación y seguimiento de los recursos de datos sensibles o peligrosos almacenados o procesados en el sistema, garantizando la responsabilidad de su protección.
AUTENTICIDAD	Control de acceso	Mecanismos que garantizan que solo las entidades autorizadas puedan acceder a los datos, servicios o recursos cloud de acuerdo con las políticas definidas.
CONFIDENCIALIDAD	Alarmas monitorizadas	Capacidad del servicio cloud para detectar, generar y supervisar alertas de seguridad que protegen los datos y los servicios contra el acceso no autorizado.
INTEGRIDAD		
Integridad de los datos	Integridad de los datos	Garantía de que los datos de los clientes permanecen completos, precisos y sin alteraciones durante su almacenamiento, procesamiento o transmisión.
Integridad del servicio	Integridad del servicio	Garantía de que los servicios cloud se ejecutan correctamente y ofrecen resultados sin alteraciones ni manipulaciones no autorizadas.

4. **Mantenibilidad**: Esta es una característica clave, ya que los servicios cloud, como cualquier software, requieren un mantenimiento continuo durante todo su ciclo de vida. Esto implica que varios equipos adapten y evolucionen activamente los servicios para cumplir con los requisitos cambiantes o preservar la funcionalidad existente. Se identificó un total de diez atributos de calidad bajo esta característica, lo que representa el 4.1 % del total. Esto es relativamente bajo, teniendo en cuenta que una parte significativa de los costos de software provienen del mantenimiento continuo en lugar del desarrollo inicial. La distribución de los atributos es la siguiente: seis para la reutilización, dos para la modificabilidad y uno para la analizabilidad y la capacidad de prueba. En la Tabla 5.13 se presenta un resumen de los atributos más representativos.

   En términos de *reutilización*, las tecnologías cloud permiten la reutilización de una amplia gama de activos de servicio (es decir, componentes o artefactos). Algunos ejemplos comunes son las máquinas virtuales, los contenedores y las configuraciones de *Kubernetes* en IaaS, así como los microservicios en SaaS. La reutilización es fundamental en las prácticas modernas de desarrollo que se utilizan para el desarrollo

de servicios cloud, como *DevOps, DevSecOps y Agile*, que ponen énfasis en la entrega iterativa y los componentes compartidos. La *analizabilidad* se refiere a la capacidad de evaluar el impacto de los cambios en los recursos cloud y de detectar posibles defectos o fallos, con el objetivo de minimizar las consecuencias no deseadas en todo el sistema. La *modificabilidad* captura la facilidad de realizar cambios en los artefactos cloud, al tiempo que limita las dependencias externas y las interrupciones. Por último, la *capacidad de prueba* está ganando importancia en los contextos cloud debido a la creciente granularidad de los componentes del servicio y la adopción de ciclos de liberación rápidos.

Tabla 5.13: Atributos de mantenibilidad.

Subcaracterística	**Atributo**	**Definición**
ANALIZABILIDAD	Defectos	Capacidad para detectar fallos, errores o cualquier comportamiento inusual durante su funcionamiento.
MODIFICABILIDAD		
Flexibilidad	Flexibilidad	Capacidad para adaptarse a cambios en los requisitos o las condiciones de funcionamiento con un esfuerzo mínimo.
Modularidad	Modularidad del servicio	Grado en que el servicio cloud está estructurado en componentes independientes con un mínimo de dependencias externas.
REUTILIZACIÓN	Variabilidad	Grado en que los componentes del servicio cloud pueden configurarse o ampliarse a través de puntos de variación definidos para admitir diferentes contextos de uso.
CAPACIDAD DE PRUEBA	Tiempo de prueba	Tiempo necesario para diseñar, preparar y ejecutar pruebas para evaluar el servicio cloud.

5. **Portabilidad**: La portabilidad es una preocupación crítica en la adopción de servicios cloud debido al alto nivel de dependencia del proveedor y los riesgos potenciales que introduce. Esta dependencia puede crear una sensación de vulnerabilidad para los clientes, especialmente con respecto a los precios, las limitaciones del servicio y la flexibilidad a largo plazo. La migración de datos y servicios, ya sea entre proveedores cloud o entre sistemas locales y cloud, sigue siendo un desafío complejo que requiere muchos recursos. Se identificó un total de nueve atributos de calidad bajo esta característica, lo que representa el 3.7 % del total. Estos atributos se distribuyen entre las subcaracterísticas de adaptabilidad e instalabilidad, lo que refleja la necesidad de evaluar tanto la facilidad de migración de un servicio como su capacidad para configurarse en un nuevo entorno. En la Tabla 5.14 se presenta un resumen de los atributos más representativos.

Tabla 5.14: Atributos de portabilidad.

Subcaracterística	**Atributo**	**Definición**
ADAPTABILIDAD	Adaptabilidad del servicio	Capacidad para adaptarse a los requisitos del cliente o a los cambios tecnológicos sin pérdida de funcionalidad ni calidad.

Table 5.14 (continua)

Subcaracterística	**Atributo**	**Definición**
Portabilidad del comportamiento de la aplicación	Nivel de adaptabilidad	Grado en que una aplicación puede funcionar correctamente cuando cambia su entorno externo (p. ej., infraestructura, configuración, API).
Portabilidad de las instrucciones de la aplicación	Soporte de herramientas de software	Capacidad de una aplicación para ejecutar sus instrucciones en el entorno de destino utilizando las herramientas de software disponibles sin necesidad de modificaciones.
INSTALABILIDAD		
Portabilidad de aplicaciones cloud	Compatibilidad con sistemas operativos	Capacidad para instalarse o migrarse entre diferentes sistemas operativos sin requerir modificaciones significativas.

6. **Adecuación funcional**: Esta característica es crucial porque refleja directamente el grado en que un servicio cloud cumple con los requisitos y necesidades de los *stakeholders*, en particular, los definidos en los SLA. Las preocupaciones comunes en esta área incluyen resultados inexactos o incompletos de la recuperación o modificación de datos, flexibilidad limitada para adaptarse a las necesidades comerciales cambiantes y incumplimiento de los objetivos de nivel de servicio. Se identificó un total de siete atributos bajo esta característica, lo que representa el 2.9 % del total. En la Tabla 5.15 se presenta un resumen de los atributos más representativos.

Tabla 5.15: Atributos de adecuación funcional.

Subcaracterística	**Atributo**	**Definición**
PERTINENCIA FUNCIONAL		
	Pertinencia del servicio	grado en que los resultados de un servicio cloud se mantienen coherentes con los valores esperados en condiciones específicas.
COMPLETITUD FUNCIONAL		
	Idoneidad	Grado en que un servicio cloud cumple todos los requisitos establecidos por el cliente.
CORRECCIÓN FUNCIONAL		
Precisión del servicio	Precisión del servicio	Grado en que los resultados producidos por el servicio cloud se ajustan a los valores esperados o especificados.
Precisión de video	Degradación de la imagen	Grado de pérdida de calidad visual en una transmisión de vídeo, en particular, las distorsiones que afectan a la información estructural de la imagen.

7. **Usabilidad**: Esta característica es esencial para evaluar la facilidad con la que se pueden entender, aprender y utilizar los servicios cloud, lo que facilita su adopción rápida y exitosa. Se centra en la interacción del usuario y evalúa la eficacia con la que el servicio cloud apoya a los usuarios en la realización de sus tareas. Se identificaron en total siete atributos bajo esta característica, lo que representa el 2.9 % del total. La Tabla 5.16 presenta una selección de los atributos más representativos

Tabla 5.16: Atributos de usabilidad.

Subcaracterística	**Atributo**	**Definición**
CAPACIDAD DE APRENDIZAJE		
Eficiencia de aprendizaje	Esfuerzo de capacidad de aprendizaje	Esfuerzo que debe realizar un usuario para aprender a utilizar el servicio cloud de forma eficaz.
Eficacia del aprendizaje	Comprensibilidad del servicio	Grado en que los usuarios pueden comprender de forma rápida y precisa cómo utilizar el servicio cloud.
OPERABILIDAD	Esfuerzo de operación	Esfuerzo requerido por un usuario para operar y hacer un uso eficaz del servicio cloud.
ESTÉTICA DE LA INTERFAZ DE USUARIO		
	Interfaces de usuario compatibles	Grado en que el servicio cloud proporciona interfaces de usuario que se adaptan a las diferentes necesidades y preferencias de los usuarios.

8. **Compatibilidad**: Esta característica aborda los importantes desafíos involucrados en el intercambio y la interacción de información entre servicios cloud, desafíos que son particularmente importantes en entornos de múltiples nubes, donde las aplicaciones de diferentes proveedores y dominios deben operar juntas sin problemas. Esta complejidad de garantizar la compatibilidad aumenta a medida que los sistemas se vuelven cada vez más interconectados y heterogéneos. Los atributos de calidad bajo esta característica se centran en la interoperabilidad de servicios y la interoperabilidad de plataformas, ofreciendo indicadores medibles para evaluar qué tan bien pueden funcionar juntos los servicios cloud. En total, se identificaron cuatro atributos, que representan el 1.7 % del conjunto general. En la Tabla 5.17 se muestran ejemplos de los atributos más representativos.

Tabla 5.17: Atributos de compatibilidad.

Subcaracterística	**Atributo**	**Definición**
INTEROPERABILIDAD		
Interoperabilidad en la nube	Interoperabilidad del servicio	Capacidad de un servicio cloud para intercambiar información e interactuar correctamente con otros servicios, ya sean del mismo proveedor o de otros diferentes.
	Plataformas de interoperabilidad	grado en que un servicio cloud ofrece las plataformas, interfaces o estándares necesarios que permiten a los usuarios integrarse e interoperar con otros servicios.

9. **No técnica**: Esta característica abarca factores adicionales no cubiertos por las otras subcaracterísticas, pero que pueden influir en la evaluación y selección de los servicios cloud. A menudo se relacionan con aspectos organizativos, legales o regulatorios, que, aunque no son de naturaleza técnica, desempeñan un papel fundamental en la determinación de la idoneidad y la adopción a largo plazo de las soluciones en la nube. Se identificó un total de seis atributos bajo esta característica, lo que representa el 2.5 % del total. La Tabla 5.18 presenta ejemplos de los atributos más representativos relacionados con las cuestiones de cumplimiento y costos.

Tabla 5.18: Atributos no técnicos.

Subcaracterística	**Atributo**	**Definición**
CUMPLIMIENTO Cumplimiento de la seguridad	Certificación de seguridad	Evidencia de que un servicio cloud ha sido evaluado y certificado formalmente según normas de seguridad y mejores prácticas reconocidas (p. ej., ISO/IEC 27001).
COSTE	Coste de escalabilidad	Recursos financieros necesarios para aumentar o disminuir la capacidad de un servicio cloud en respuesta a la demanda.
	Coste de mantenimiento	Recursos financieros necesarios para operar, actualizar y mantener un servicio cloud con la capacidad prevista.
	Coste de seguridad	Recursos financieros necesarios para implementar y mantener medidas de seguridad que protejan el servicio cloud.

5.4.3. Determinación de métricas

La asignación de métricas de la taxonomía a los atributos del modelo se llevó a cabo mediante un proceso sistemático de mapeo conceptual y operacional:

1. *Recopilación de métricas*: Se recopilaron métricas relacionadas con las características de calidad, tanto internas como externas, de los servicios cloud, con base en la taxonomía presentada en el capítulo 4.

2. *Análisis de métricas*: En primer lugar, cada métrica identificada en la taxonomía fue analizada en términos de su propósito, unidad de medida, naturaleza (directa o indirecta) y frecuencia de observación. En segundo lugar, se comparó esta información con la definición formal de cada atributo del modelo para determinar si la métrica representaba de manera directa el comportamiento del atributo o si actuaba como un indicador indirecto. Este mapeo se realizó aplicando el principio de *trazabilidad bidireccional*: una métrica solo se asigna a un atributo cuando existe evidencia conceptual suficiente para justificar la relación, y cada atributo debe estar respaldado por al menos una métrica observada en la práctica. Finalmente, se revisaron las relaciones resultantes mediante una triangulación con la literatura previa y con expertos del dominio industrial, garantizando que el conjunto de métricas asignadas fuese tanto exhaustivo como operacionalizable en un entorno real de monitorización.

3. *Adaptación de las métricas*: En el caso de las métricas destinadas a evaluar los servicios cloud en operación, se verificó la posibilidad de ajustar sus definiciones y funciones de medición para cubrir tanto escenarios independientes de la plataforma como específicos de la plataforma. Por ejemplo, la *tasa de éxito*, como métrica independiente de la plataforma, mide la proporción de tareas u operaciones completadas con éxito en un intervalo de tiempo definido. Al instanciarla en una base de datos como PostgreSQL, puede adaptarse como una métrica específica que evalúa el éxito de las operaciones de consulta o de escritura.

4. *Consolidación de métricas*: Se detectaron casos en los que diferentes atributos hacían referencia al mismo concepto de medición, aunque con diferentes enfoques. Para re-

ducir la redundancia y clarificar su aplicación, estas métricas se integraron mediante análisis temático. El resultado fue una asociación más precisa entre atributos y métricas, cada una con una o más operacionalizaciones. Se descartaron las métricas que carecían de métodos o funciones de cálculo suficientemente detallados para su aplicación práctica.

Como resultado de la aplicación de este proceso, a los atributos de calidad identificados se les asoció al menos una métrica de un total de 407. Por ejemplo, el atributo *precisión del servicio* refleja el grado de alineación entre la funcionalidad real del servicio y los resultados esperados. Este atributo puede cuantificarse mediante métricas como la *precisión esperada* y la *frecuencia de precisión*. La primera calcula la relación entre la precisión esperada y la observada en el servicio [126], mientras que la segunda mide la frecuencia con la que el servicio no cumple con los niveles de precisión definidos por el SLA [131].

Finalmente, se revisaron las relaciones resultantes mediante una triangulación con la literatura previa y con expertos del dominio, garantizando que el conjunto de métricas asignadas fuese tanto exhaustivo como operacionalizable en un entorno real de monitorización.

Este proceso de derivación asegura que el modelo no solo cuente con una base teórica sólida, sino que también sea adecuado para su aplicación práctica, al conectar de forma transparente y reproducible las características de calidad con las métricas observables que las representan.

5.4.4. Modelo de calidad en *Power BI*

Una vez definido el modelo de calidad, se procedió a su implementación en *Power BI* con el fin de facilitar su interpretación, consulta y uso operativo por parte de los distintos stakeholders. La representación del modelo en esta herramienta permite aprovechar sus capacidades de visualización jerárquica, análisis dinámico y vinculación con datos reales de monitorización, lo que incrementa significativamente el valor práctico del marco propuesto.

La integración del modelo en *Power BI* ofrece múltiples ventajas:

1. **Visualización jerárquica y navegable del modelo de calidad**. *Power BI* permite representar la estructura completa del modelo (características, subcaracterísticas, atributos y métricas) mediante *dashboards* interactivos. Esta visualización facilita recorrer la jerarquía desde los niveles abstractos hasta las métricas concretas, lo que permite comprender cómo cada métrica contribuye a los atributos y características superiores.

2. **Actualización dinámica basada en datos reales de monitorización**. El modelo se conecta a fuentes de datos reales del servicio cloud, lo que permite actualizar automáticamente la información y reflejar variaciones en la calidad en tiempo casi real. Esta capacidad es esencial para servicios sujetos a fluctuaciones de carga, eventos inesperados o degradaciones puntuales de QoS.

3. **Agregación, normalización y cálculo de indicadores compuestos**. *Power BI* posibilita transformar métricas heterogéneas en indicadores comparables mediante medidas DAX, escalados, agregaciones temporales y funciones estadísticas. Esto permite generar puntuaciones consistentes para atributos y subcaracterísticas, simplificando la interpretación.

4. **Accesibilidad para distintos perfiles organizacionales** Los *dashboards* pueden ser utilizados por: a) *proveedores del servicio cloud*, que analizan métricas internas y el comportamiento de la infraestructura; b) *clientes*, que interpretan el cumplimiento de los SLAs y las variaciones de calidad; c) *gestores*, que evalúan el impacto global; d) *responsables técnicos*, que profundizan en métricas de bajo nivel.

De este modo, *Power BI* se convierte en un punto de acceso común para todos los *stakeholders*, ofreciendo una plataforma unificada para explorar, comprender y aplicar el modelo de calidad. La herramienta materializa un **modelo de calidad** completo de la estructura propuesta, reproduciendo con detalle la jerarquía definida en el modelo conceptual (características → subcaracterísticas → atributos → métricas). Esta organización se implementó mediante un esquema de tablas relacionadas, donde las claves primarias y foráneas aseguran la coherencia semántica y la trazabilidad entre los distintos niveles del modelo.

El modelo resultante adopta un **enfoque holístico**, integrando no solo la estructura jerárquica del modelo de calidad sino también sus mecanismos de operacionalización. Esta versión está disponible tanto en un archivo de *Excel* como en una instancia interactiva de *Power BI* (véase Figura 5.3), lo que facilita su consulta, análisis y reutilización en distintos contextos organizacionales. La representación incluye varios informes y vistas temáticas que permiten a los *stakeholders* explorar el modelo de forma intuitiva e interactiva. Por ejemplo, los responsables de la toma de decisiones pueden profundizar en modelos de servicio específicos (SaaS, PaaS, IaaS) o en métricas individuales para comprender su definición operativa, sus relaciones y su rol dentro del modelo. Asimismo, las métricas pueden filtrarse o clasificarse según perspectivas relevantes, como los roles de los *stakeholders* (proveedor, cliente), las fases del ciclo de vida o los artefactos cloud evaluados, lo que favorece una interpretación contextualizada y alineada con necesidades reales.

Adicionalmente, se desarrollaron ***dashboards* interactivos** que facilitan la navegación mediante filtros asociados a elementos clave del dominio cloud, como tipo de servicio, *stakeholders*, fase del ciclo de vida y artefactos evaluados. Estos *dashboards* no solo hacen que la exploración del modelo sea más accesible, sino que también constituyen un apoyo concreto para la definición y revisión de SLAs, al vincular cada atributo de calidad con sus métricas e indicadores, así como con los mecanismos de medición específicos. La herramienta también ofrece capacidades de personalización y extensión, permitiendo incorporar nuevas métricas, adaptar el modelo a diferentes contextos organizacionales y generar vistas especializadas orientadas al proveedor, al cliente o a los artefactos cloud en evaluación.

En conjunto, esta implementación refuerza la contribución del modelo de calidad al demostrar su aplicabilidad práctica, su flexibilidad y su potencial para integrarse de manera

operativa en procesos reales de gestión y evaluación de servicios cloud. El archivo completo en *Excel* y la versión interactiva en *Power BI* están disponibles en el repositorio que acompaña esta tesis, en `https://bit.ly/frameworkCloudQuality`, garantizando transparencia, reutilización y replicabilidad conforme a las buenas prácticas de la ingeniería de software empírica.

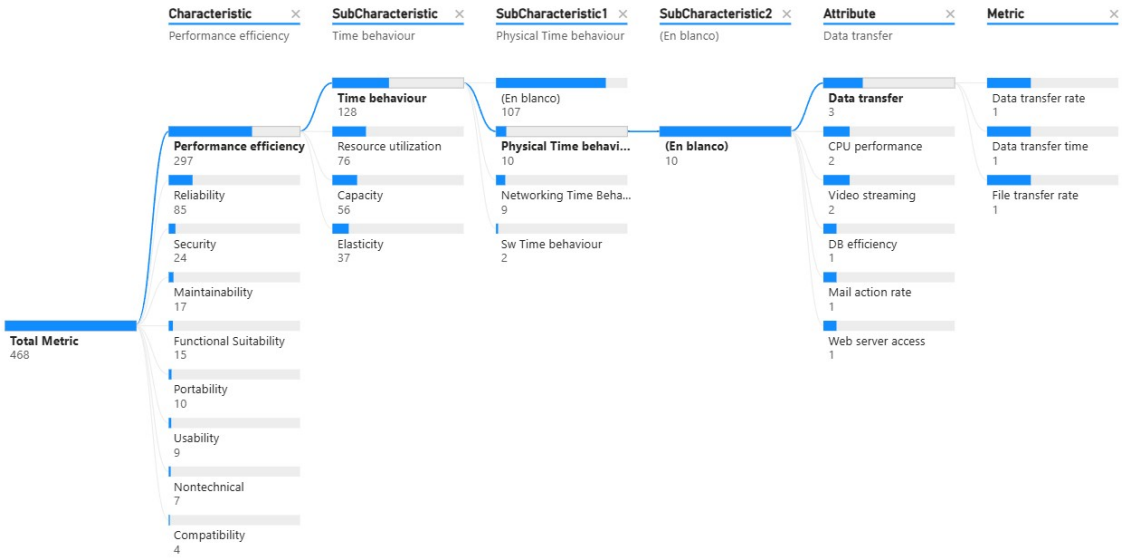


Figura 5.3: Modelo de Calidad en *Power BI*.

5.5. Proceso de evaluación de calidad

El proceso de evaluación propuesto **operacionaliza el modelo de calidad** y proporciona un marco sistemático, alineado con el estándar ISO/IEC 25040 [14], para determinar el grado en que un servicio cloud satisface los requisitos de calidad definidos. Su propósito es guiar a los distintos *stakeholders* (proveedores, consumidores, clientes y desarrolladores) en las actividades de evaluación de la calidad de los servicios cloud. Para ello, además, establece los criterios de medición, los instrumentos de captura y la forma en que los datos brutos se transforman en resultados interpretables y comparables. La Figura 5.4 presenta las actividades principales del proceso, que se describen a continuación:

1. Establecer los requisitos de calidad
2. Configuración de la evaluación
3. Diseño de la evaluación
4. Operacionalización de la evaluación.

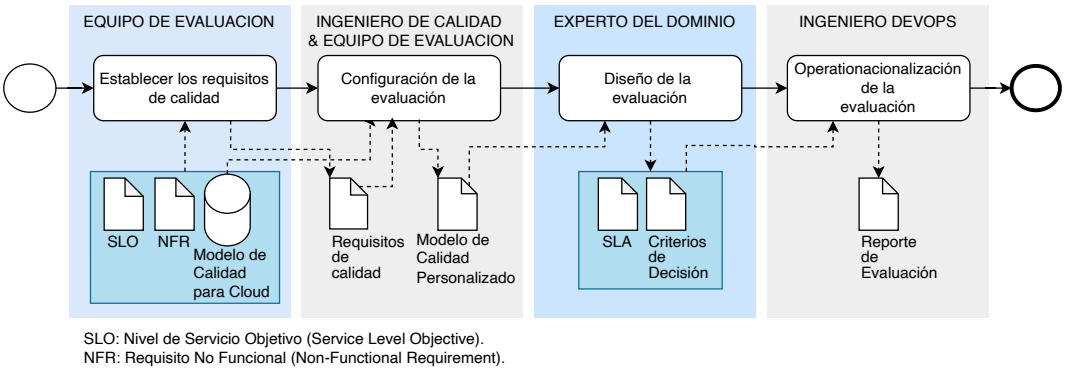


Figura 5.4: Proceso de Evaluación de Calidad.

5.5.1. Establecer los requisitos de calidad

Esta actividad inicial, liderada por el equipo de evaluación conformado por los *stakeholders* clave, se centra en la definición del propósito de la evaluación y de los requisitos de calidad, que deben ser verificables. En esta fase se revisa la documentación relevante del servicio, incluyendo los requisitos funcionales y no funcionales, arquitectura, políticas de operación, historial de incidentes y objetivos de nivel de servicio (SLO), para identificar los atributos de calidad que son prioritarios para la organización.

Una contribución clave de esta etapa es que la formulación de los requisitos se realiza desde el inicio con criterios de medición explícitos: cada requisito se asocia a métricas, parámetros operativos y condiciones específicas de monitorización. Esto implica definir, por ejemplo, metas de disponibilidad mensual esperadas, límites aceptables de latencia, frecuencia de muestreo de las métricas, intervalos temporales de medición y métricas derivadas del SLA o de las políticas internas de la organización. El resultado es un conjunto de requisitos verificables cuya operacionalización es posible desde el inicio del proceso.

5.5.2. Configuración de la evaluación

Para esta actividad, el ingeniero de calidad, en coordinación con el equipo de evaluación, adapta el modelo genérico al contexto particular del servicio evaluado mediante la creación de un *Modelo de Calidad Personalizado*. Esta adaptación se apoya en la estructura jerárquica del modelo, que admite tanto el enfoque descendente como el ascendente.

Como se muestra en la Figura 5.5, el proceso inicia con la selección de las características y subcaracterísticas más relevantes, según los requisitos establecidos. Posteriormente, se eligen los atributos de calidad atendiendo a las prioridades de los *stakeholders*, garantizando que sean medibles. En este punto, se identifican las métricas y las operacionalizaciones más adecuadas,considerando las propiedades del artefacto o plataforma cloud.

Con el fin de fortalecer los criterios de medición, esta etapa implica identificar formalmente las herramientas de monitorización que podrán captar cada métrica. Por ejemplo:

1. servicios nativos del proveedor cloud como AWS CloudWatch, Azure Monitor, Google Cloud Operations,

2. plataformas independientes como New Relic, Prometheus o Datadog,

3. mecanismos internos basados en *logs* de aplicación, agentes del sistema operativo o trazas de Kubernetes.

Asimismo, se especifica el intervalo temporal de cada métrica. Por ejemplo, latencia capturada en percentiles por minuto, disponibilidad evaluada cada 5 o 15 minutos, utilización de CPU cada cinco segundos, y *throughput* agregado por segundo.

Finalmente, se establecen los **umbrales preliminares** de cumplimiento que se utilizarán durante el análisis: valores objetivo del SLA/SLO, rangos aceptables para condiciones de carga normales y límites críticos para la detección de anomalías. De esta forma, el modelo se transforma en un instrumento operativo que define **qué medir, con qué frecuencia, con qué herramienta y bajo qué reglas de decisión**.

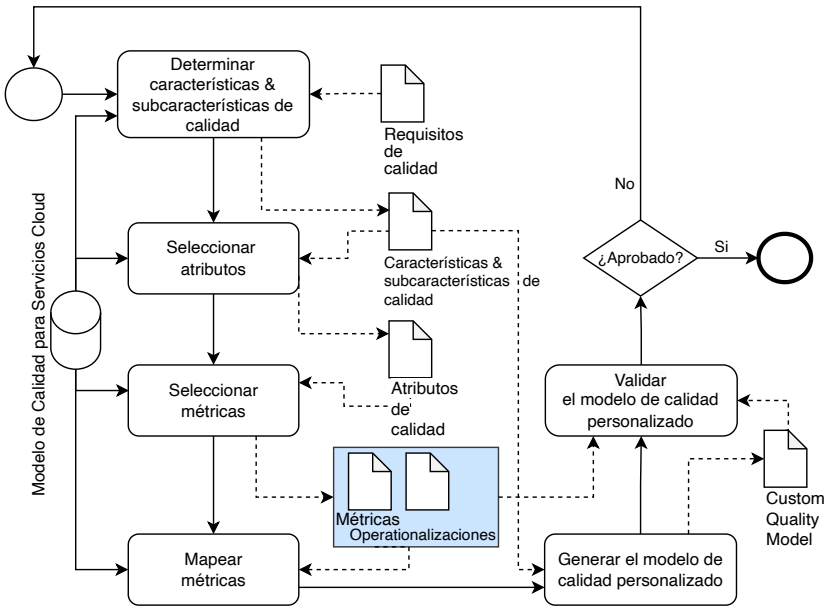


Figura 5.5: Configuración de Evaluación de Calidad.

5.5.3. Diseño de la evaluación

Esta actividad define la estrategia mediante la cual el modelo personalizado será aplicado e interpretado, especificando las entidades objetivo, los artefactos a evaluar y los

resultados esperados. Esta fase adquiere especial relevancia porque determina la forma en que los datos brutos se transforman en indicadores interpretables y comparables.

En primer lugar, se especifican los procedimientos para la agregación y la transformación de las observaciones. Puesto que las *métricas base* suelen capturarse como valores puntuales, se establecen reglas de preprocesamiento que incluyen: el cálculo de medias y medianas móviles, percentiles (p95, p99), tasas de error, detección de picos, normalización de valores y unificación de unidades entre métricas heterogéneas.

En segundo lugar, para las *métricas derivadas* se configuran las funciones que convertirán estas mediciones en valores discretos o normalizados (por ejemplo, escalas entre 0 y 1). Estas funciones permiten que el proceso adopte distintas formas: cumplimiento/no cumplimiento, puntuaciones cuantitativas o la identificación de fortalezas y debilidades, o recomendaciones de mejora.

El equipo de evaluación, con el apoyo de expertos del dominio, establece criterios de decisión para cada métrica, basados en umbrales definidos. Por ejemplo, la métrica *latencia* cumple si el valor de p95 se encuentra por debajo del umbral establecido, la *utilización de CPU* cumple si permanece en un intervalo operativo aceptable durante un periodo predefinido, o la tasa de errores cumple si el valor observado no supera un umbral crítico.

Estos criterios pueden alinearse directamente con el SLA del servicio, describiendo las métricas de QoS más relevantes. El producto final de esta fase es un **plan de evaluación**, que orienta la ejecución de la siguiente etapa.

5.5.4. Operacionalización de la evaluación

La operacionalización corresponde a la ejecución del plan definido. El ingeniero de DevOps es responsable de implementar mecanismos de monitorización que permitan recolectar, validar y procesar los datos reales del servicio conforme a las métricas y operacionalizaciones seleccionadas.

Se implementan los mecanismos de ingesta de datos definidos previamente, ya sea mediante técnicas de extracción periódica (modelo *pull*), agentes que envían métricas automáticamente (modelo *push*) o *pipelines* de datos que consolidan información de múltiples fuentes [100]. Antes de proceder al análisis, se realiza una validación técnica de las mediciones, que incluye la detección de vacíos temporales, la verificación de la consistencia entre diferentes fuentes, la identificación de valores atípicos y la normalización de formatos y unidades.

Los datos recopilados se analizan e interpretan mediante los umbrales definidos y se transforman en puntuaciones que reflejan el nivel de cumplimiento del servicio respecto al atributo de calidad que se está midiendo. Una *adherencia alta* indica conformidad con los requisitos establecidos, *niveles medios o parciales* sugieren áreas de mejora, mientras que una *puntuación baja* indica una desviación significativa entre el servicio evaluado y el atributo de calidad del modelo, sugiriendo la necesidad de replantear la configuración o incluso el diseño del servicio.

Este enfoque sistemático garantiza que los resultados sean comprensibles para los distintos *stakeholders* y que puedan integrarse en decisiones de monitorización, mejora con-

tinua, renegociación de SLA o planificación estratégica.

5.5.5. Discusión

En síntesis, el proceso de evaluación propuesto operacionaliza el modelo de calidad definido previamente, permitiendo su aplicación práctica en distintos contextos de servicios cloud. La secuencia de actividades (establecimiento de requisitos, configuración, diseño y operacionalización) asegura una alineación sistemática entre los objetivos de evaluación, el modelo de calidad personalizado y la evidencia empírica obtenida mediante métricas y monitorización. De este modo, se garantiza la trazabilidad entre los niveles del modelo (características, subcaracterísticas, atributos, métricas) y la aplicabilidad del marco en escenarios reales, facilitando la toma de decisiones fundamentadas por parte de los diferentes *stakeholders*.

5.6. Conclusiones

En este capítulo se presentaron dos contribuciones fundamentales que fortalecen de manera sustantiva la evaluación de la calidad en servicios cloud: el desarrollo de un modelo de calidad genérico adaptado al dominio y la propuesta de un proceso de evaluación sistemático, ambos alineados con la familia de estándares ISO/IEC 25000.

En primer lugar, se desarrolló un **modelo de calidad genérico para servicios cloud**, concebido como una extensión del estándar ISO/IEC 25010. Este modelo mantiene la estructura jerárquica conceptual—características, subcaracterísticas, atributos y métricas—, pero la amplía y adapta al dominio cloud mediante el análisis combinado de una revisión sistemática de la literatura y del marco teórico.

Su diseño se fundamenta en criterios de generalidad, independencia de plataforma y capacidad de operacionalización, lo que garantiza su aplicabilidad a distintos tipos de servicio (IaaS, PaaS, SaaS) y entornos tecnológicos. Su estructura jerárquica soporta un enfoque mixto, tanto descendente (*top-down*)—desde características a métricas concretas—, como ascendente (*bottom-up*)—desde métricas hacia atributos y características—, lo que garantiza flexibilidad y adaptabilidad ante diversas necesidades organizacionales y escenarios de evaluación que incluyen auditorías de SLA, diagnósticos de rendimiento, evaluaciones de resiliencia o análisis comparativos entre proveedores.

En segundo lugar, se propuso un **proceso de evaluación alineado con la ISO/IEC 25040**, que operacionaliza el modelo de calidad mediante un proceso sistemático, compuesto por actividades claramente estructuradas: establecimiento de requisitos, configuración, diseño y operacionalización de la evaluación, para su aplicación en escenarios concretos. Este enfoque proporciona una guía metodológica robusta que evidencia la trazabilidad entre los elementos conceptuales del modelo y la información empírica obtenida mediante métricas y herramientas de monitorización. Al integrar mecanismos explícitos de medición, criterios de decisión y procedimientos de transformación de datos, el proceso ofrece un soporte riguroso para la evaluación práctica en escenarios reales, promoviendo la toma de decisiones basadas en evidencias cuantitativas, reproducibles y comparables.

Estas contribuciones también presentan implicaciones significativas para la investigación y la práctica profesional. Desde la perspectiva investigadora, el modelo y el proceso constituyen una base formal y reutilizable para estudios empíricos sobre calidad en entornos multi-nube, optimización de recursos, ingeniería de rendimiento o elasticidad cloud. Su estructura cuantificable facilita su incorporación en experimentos, simulaciones y análisis predictivos y habilita líneas futuras de investigación que incluyen la selección automatizada de métricas, la generación dinámica de modelos de calidad o la integración del modelo en sistemas de autoscaling basados en políticas adaptativas.

En la práctica, las contribuciones poseen un impacto directo en la operación y gestión de servicios cloud. Para los diferentes actores involucrados (proveedores, consumidores, desarrolladores y usuarios finales), facilita la identificación de requisitos de calidad, la configuración de un modelo de calidad, el diseño del plan de evaluación y la ejecución de la monitorización y evaluación. Además, fortalece la alineación entre los requisitos de negocio y métricas de QoS y permite articular SLA más rigurosos, transparentes y verificables, asegurando la trazabilidad. Para los equipos de despliegue y operaciones (*DevOps*) y *MLOps*, la estructura jerárquica del modelo ofrece una referencia unificada para organizar *dashboards* de monitorización, priorizar alertas, definir umbrales de operación y diseñar estrategias de observabilidad coherentes con los objetivos de calidad. De este modo, se asegura la aplicabilidad del modelo genérico propuesto en escenarios prácticos y heterogéneos, promoviendo la toma de decisiones fundamentadas en evidencia empírica.

Finalmente, la independencia de plataforma y la abstracción de los atributos de calidad permiten que el modelo pueda ser aplicado no solo en entornos cloud, sino también en tecnologías emergentes como *edge computing*, *fog computing* y arquitecturas *cloud-native*. En estos escenarios, caracterizados por distribuciones dinámicas de cargas, heterogeneidad de recursos y requisitos estrictos de latencia y disponibilidad, el modelo ofrece un marco consistente para representar y comparar la calidad a través de diferentes capas de la arquitectura, fomentando la portabilidad conceptual y operativa. Su reutilización facilita la integración entre entornos híbridos y distribuidos, mejorando la coherencia de los mecanismos de evaluación y posibilitando análisis transversales que hasta ahora resultaban difíciles de articular debido a la fragmentación de las métricas y los enfoques.

En conjunto, el modelo de calidad propuesto y el proceso de evaluación constituyen una base conceptual y metodológica sólida para la evaluación de la calidad en servicios cloud. Además de proporcionar una estructura formal para representar y medir la calidad, su diseño facilita la aplicación práctica en entornos reales y habilita tanto la investigación empírica como la evolución hacia mecanismos de monitorización y gestión autónoma. De esta manera, las contribuciones de este capítulo fortalecen la capacidad de los investigadores y profesionales de la industria para comprender, evaluar y optimizar la calidad de los servicios cloud.

Capítulo 6

Aplicación del Modelo de Calidad en un Entorno industrial

En este capítulo se presenta un estudio de caso orientado a validar empíricamente el modelo de calidad genérico propuesto en el capítulo anterior, demostrando su capacidad para integrarse en entornos industriales reales y apoyar la evaluación continua de servicios cloud conforme a la norma ISO/IEC 25040 [14].

En la Sección 6.1 se introduce el propósito y la importancia del estudio de caso para evaluar el modelo de calidad en un contexto industrial. En la Sección 6.2 se presenta el diseño del estudio de caso que hemos utilizado para operacionalizar el modelo de calidad genérico. En la Sección 6.3 se presenta la recolección de datos del estudio de caso mediante la monitorización del servicio En la Sección 6.4 se presenta el análisis de datos para dar a conocer el estado actual de la calidad del estudio de caso. En la Sección 6.5 se presenta la discusión de los resultados obtenidos. En la Sección 6.6 se da respuesta a las preguntas de investigación basadas en los resultados obtenidos de la evaluación. En la Sección 6.7 se presentan las amenazas a la validez del estudio de caso. En la Sección 6.8 se discuten las implicaciones para la investigación y la práctica. En la Sección 6.9 se presentan las conclusiones de los resultados.

6.1. Introducción

La validación empírica de modelos en ingeniería de software requiere metodologías que permitan observar fenómenos en su contexto operativo. En este sentido, los estudios de caso constituyen una estrategia metodológica idónea para evaluar artefactos en condiciones reales, al permitir una exploración profunda y contextualizada de su comportamiento.

En este capítulo se presenta un estudio de caso diseñado para evaluar la aplicabilidad práctica del modelo de calidad genérico propuesto en el capítulo anterior. El estudio se lleva a cabo en un entorno industrial real, siguiendo las guías metodológicas de Runeson et al. [37], con el objetivo de demostrar la utilidad del modelo para la monitorización, evaluación y mejora de los servicios cloud.

La evaluación se desarrolló en un servicio SaaS denominado *Social Alert*, implementado por la empresa latinoamericana Golden S.A., y se aplicó a los tres niveles de servicio cloud: infraestructura (IaaS), plataforma (PaaS) y software (SaaS). Se utilizaron datos cuantitativos obtenidos mediante herramientas de monitorización y datos cualitativos recolectados mediante entrevistas con profesionales del sector.

La principal contribución de este capítulo radica en evidenciar cómo el modelo de calidad puede ser operacionalizado en un entorno industrial, proporcionando métricas significativas para la toma de decisiones y la mejora continua del servicio.

6.2. Diseño del estudio de caso

El diseño del estudio de caso se estructuró conforme al enfoque propuesto por Runeson et al. [37], incluyendo la definición de objetivos, el marco conceptual, las preguntas de investigación, la estrategia de muestreo y los métodos de recolección y análisis de datos.

El *objetivo principal* fue:

> *Demostrar la aplicabilidad práctica del modelo de calidad propuesto mediante su operacionalización en un entorno real, evaluando un servicio cloud en operación.*

El *marco conceptual* se fundamentó en la personalización del modelo genérico, adaptando sus características de calidad a atributos específicos, métricas y operacionalizaciones relevantes para el servicio *Social Alert*.

Las *preguntas de investigación* que guiaron el estudio fueron:

- **PI1**: ¿Se puede configurar un modelo de calidad personalizado a partir del modelo genérico para servicios cloud específicos?
- **PI2**: ¿Las métricas seleccionadas, sus operacionalizaciones, y sus resultados de medición proporcionan información significativa sobre el estado de la calidad del servicio?
- **PI3**: ¿Puede adaptar el modelo de calidad personalizado a requisitos nuevos o cambiantes?

6.2.1. Selección del caso

Se seleccionó un caso relevante en colaboración con expertos en servicios cloud. La unidad de análisis fue el servicio *Social Alert* (veáse 6.1), ofrecido por Golden S.A., con operaciones en México, Argentina, Puerto Rico y Ecuador. La empresa concedió acceso completo al sistema, lo que permitió una evaluación exhaustiva en un entorno controlado.



Figura 6.1: Estudio de caso.

Social Alert es un servicio SaaS de ciencia de datos que emplea técnicas de análisis de sentimientos para interpretar el contenido generado por usuarios en redes sociales. Su arquitectura se compone de microservicios, APIs y bases de datos, desplegados en la infraestructura proporcionada por Linode. La evaluación se centró en los componentes clave de las capas IaaS, PaaS y SaaS, utilizando herramientas de monitorización para recolectar datos cuantitativos y entrevistas para obtener información cualitativa.

La Figura 6.2 presenta la arquitectura de SAlert. El sistema se compone de microservicios, API e instancias de base de datos (DB). Los microservicios realizan procesos clave llamados *downloaders* para extraer datos, un proceso de *match* que utiliza cadenas de búsqueda predefinidas con términos clave para obtener coincidencias y clasificar los datos, y un proceso de *enrichment* para comprender y evaluar las opiniones antes de guardarlas como resultados. Las API permiten la comunicación y exponen la funcionalidad a los usuarios. Como base de datos, utilizamos Redis [150] como almacenamiento en caché de datos en memoria, MongoDB [151] para el almacenamiento de documentos y PostgreSQL [152] para el almacenamiento de datos relacionales. Una API de una aplicación web proporciona la visualización de datos a través de paneles, gráficos e informes para comunicar los resultados a los clientes. Linode [153] proporcionó la infraestructura cloud, con tareas clave que incluyen la creación y configuración de máquinas virtuales (VM), la implementación de recursos en la red (p. ej., contenedores *Docker*) y la adición de volúmenes o cubos de almacenamiento en la nube.

Para evaluar la calidad de SAlert, nos centramos en un subconjunto específico de los componentes de la arquitectura del sistema (véase Figura 6.3), que incluyó los siguientes componentes: infraestructura, como una VM que sirve como capa base; plataforma, como una instancia de base de datos *PostgreSQL* y siete contenedores *Docker*; y software, como una API que representa la capa de aplicación. Estos contenedores alojan instancias de base

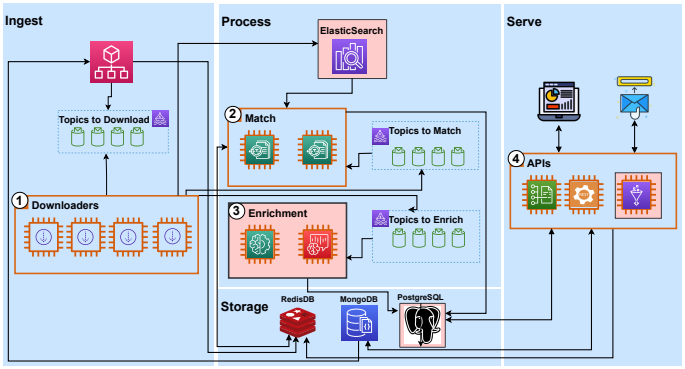


Figura 6.2: Arquitectura del servicio cloud SAlert.

de datos y otros componentes que forman parte del SaaS. Los siete contenedores evaluados son:

1. *Downloader*: microservicio que extrae contenido de redes sociales;
2. *Mongo srv*: una instancia de `MongoDB` para manejar estructuras de datos más complejas;
3. *Redis srv*: una instancia de la base de datos `Redis` para el almacenamiento en caché en memoria;
4. *Autenticación*: gestiona los procesos de autenticación de usuarios;
5. *Gateway*: admite balanceo de carga a través de instancias de contenedores;
6. *Eureka*: se encarga de la detección y el registro de microservicios; y
7. *Interfaz de usuario*: la aplicación web *front-end* para los usuarios finales.

Finalmente, se utilizó una herramienta de monitorización de terceros para recopilar datos de estos componentes para la medición y el cálculo de métricas.

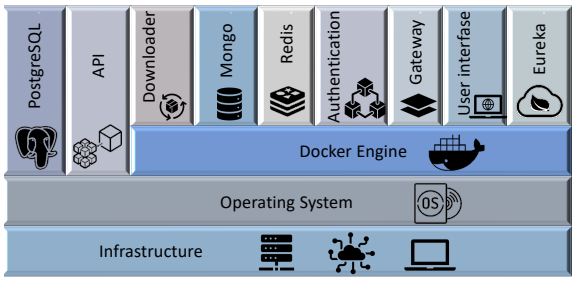


Figura 6.3: Componentes de arquitectura de SAlert.

6.2.2. Justificación del caso seleccionado

El servicio SAlert fue seleccionado como caso de estudio por constituir un escenario representativo y desafiante para la validación del modelo de calidad propuesto en esta tesis. Se trata de un servicio cloud operativo en un entorno industrial real, que combina

características propias de distintos niveles arquitectónicos (SaaS, PaaS e IaaS) y está sujeto a variaciones dinámicas de carga, patrones de uso no estacionarios y requisitos estrictos de disponibilidad y rendimiento. Estas propiedades lo convierten en un caso revelador para evaluar la capacidad del modelo de calidad para capturar y relacionar atributos de calidad en múltiples capas del servicio.

Desde el punto de vista operativo, SAlert presenta un comportamiento típico de servicios cloud modernos, con fluctuaciones en el volumen de peticiones, picos de actividad asociados a eventos externos y una dependencia directa de la infraestructura subyacente y de los mecanismos de escalado y monitorización. Esta variabilidad permite observar cómo métricas de bajo nivel (por ejemplo, utilización de recursos o latencia de red) se reflejan en atributos de calidad percibidos a nivel de servicio, proporcionando un contexto adecuado para analizar la trazabilidad propuesta por el modelo. Asimismo, el hecho de que el servicio esté integrado en una organización proveedora de servicios cloud, con prácticas de observabilidad ya establecidas, facilita la evaluación del modelo en condiciones realistas, más allá de entornos controlados o simulados.

En consecuencia, aunque el estudio se centra en un único servicio, la selección de SAlert responde a criterios de representatividad y complejidad que refuerzan la validez externa del estudio. Las lecciones extraídas sobre la instanciación del modelo, la selección de métricas, la interpretación de resultados y la identificación de áreas críticas de calidad pueden generalizarse a otros servicios cloud con arquitecturas y patrones de uso similares, especialmente en contextos donde la calidad depende de la interacción entre múltiples capas y componentes del servicio.

6.2.3. Preparación para la recolección de datos

La operacionalización del modelo de calidad requirió varias etapas. En primer lugar, el equipo evaluador definió los requisitos de calidad en colaboración con los *stakeholders* clave. Posteriormente, el ingeniero de calidad instanció y personalizó el modelo genérico para el servicio cloud *SAlert*, seleccionando atributos relevantes, mapeando métricas específicas y especificando las operacionalizaciones necesarias para su medición.

Esta personalización incluyó la definición de criterios de decisión que permitieran determinar los niveles de calidad alineados con los objetivos del servicio y los compromisos establecidos en el SLA. El proceso se diseñó para garantizar la trazabilidad entre los requisitos, las métricas y los datos recolectados, facilitando así una evaluación objetiva y reproducible.

6.2.4. Establecimiento de requisitos de calidad

La definición de requisitos de calidad constituye un paso crítico en la configuración del modelo, ya que permite delimitar el alcance de la evaluación y establecer expectativas claras sobre el desempeño del servicio. Este proceso implicó un análisis exhaustivo del dominio de aplicación, considerando tanto requisitos funcionales como no funcionales.

El objetivo de esta evaluación era doble:

> *i) Validar la utilidad del modelo de calidad para evaluar la calidad general del servicio en la nube seleccionado, y ii) identificar posibles áreas de mejora tanto en el modelo de calidad como en el servicio evaluado.*

La identificación de requisitos se realizó de forma colaborativa mediante un taller llevado a cabo en enero de 2023, en el que participaron el propietario del producto, el gerente del proyecto, un experto en el dominio y el ingeniero de calidad (autor del estudio). Durante la sesión se revisaron la arquitectura del servicio, el SLA vigente y el comportamiento observado del servicio en operación.

Los requisitos definidos abarcan aspectos críticos como los riesgos asociados, la urgencia de objetivos específicos, las experiencias pasadas y la utilización de los recursos. A continuación se presenta la lista final de requisitos identificados, cada uno incluye especificaciones técnicas y umbrales cuantitativos que permiten evaluar el cumplimiento de los compromisos establecidos en el SLA.

IaaS

`R1` La capacidad de la CPU de la VM debe ser de al menos 4 núcleos, con una utilización que no supere el 80 % tanto para la VM como para sus contenedores.

`R2` La capacidad de memoria de la VM debe ser de mínimo de 8 GB y su utilización debe mantenerse por debajo del 60 % en condiciones normales de funcionamiento.

`R3` El uso máximo de memoria de la VM antes de activar el balanceador de carga no debe superar el 75 % de la memoria total asignada, lo que garantiza el correcto funcionamiento del host y la disponibilidad de recursos para todos los componentes.

`R4` La capacidad de almacenamiento de la VM debe ser de al menos 120 GB, con un mínimo de 50 GB de espacio libre en disco y un máximo de 70 GB de espacio utilizado en disco.

`R5` La VM debe demostrar una velocidad mínima sostenida de E/S de disco de 20 MBps de lectura y 35 MBps de escritura.

`R6` La VM debe admitir un rendimiento de red mínimo superior a 200 paquetes entrantes por segundo y 35 paquetes salientes por segundo.

`R7` La disponibilidad de la red del servicio debe ser 24/7, con un tiempo de inactividad cero para el mantenimiento planificado mediante implementaciones continuas, logrando al menos un 99,99 % de *uptime* anual.

PaaS

R8 La capacidad de memoria compartida por los contenedores debe ser de 8 GB, con 2 GB reservados para *Mongo srv*. Los límites para el uso de memoria de los contenedores son: i) *Mongo srv*: menos de 1,6 GB; ii) Todos los demás contenedores: menos de 400 MB.

R9 El rendimiento medio mínimo de la red de entrada para los contenedores debe ser: i) *Mongo srv*: 20 MBps; ii) *Redis srv*, *downloaders*, *gateway*: 5 MBps cada uno; iii) Todos los demás contenedores: 0,5 MBps cada uno.

R10 El rendimiento medio mínimo de la red de salida para los contenedores debe ser: i) *Mongo srv*: 50 MBps; ii) *Redis srv*: 5 MBps; iii) Todos los demás contenedores: 0,15 MBps cada uno.

R11 Los contenedores deben admitir al menos 150 procesos simultáneos para *Mongo srv* y 50 procesos simultáneos para la *autenticación*.

R12 El promedio semanal de E/S de disco por contenedor debe cumplir los siguientes umbrales: i) *Mongo srv*: al menos 4 GB de lectura y 10 GB de escritura; ii) *Redis srv*: al menos 2 GB de lectura y 25 MB de escritura.

SaaS

R13 El tiempo medio de respuesta de la API del lado del servidor debe ser inferior a 0,05 segundos, excluyendo la latencia de la red.

R14 Cada instancia de la API debe ser capaz de gestionar al menos 500 solicitudes por segundo de forma simultánea. El sistema debe escalarse horizontalmente para soportar una mayor demanda.

R15 La tasa de éxito de las solicitudes no debe ser inferior al 99 % en condiciones normales de funcionamiento. Los fallos deben registrarse y clasificarse según su gravedad.

R16 La disponibilidad del servicio debe ser 24/7, con un tiempo de actividad de al menos el 99,99 % anual. El mantenimiento planificado debe realizarse sin interrupciones del servicio.

R17 El servicio debe admitir al menos 1500 conexiones de usuarios simultáneas, con al menos un 10 % de ellas activas al mismo tiempo. Los intentos de conexión no válidos o sospechosos deben activar mecanismos de seguridad adaptativos (por ejemplo, bloqueo de IP, limitación de velocidad, detección de anomalías) y notificar de forma proactiva a los administradores las posibles amenazas de seguridad.

R18 Si se detectan más de 10 000 solicitudes por segundo, se debe activar una alarma para evitar posibles ataques de denegación de servicio (DoS).

`R19` Se incluirá un máximo de 10 solicitudes de cambios menores después de la implementación, sin coste adicional. "Menores" se refiere a cambios de configuración o de la interfaz de usuario que no tienen impacto en la base de datos ni requieren un rediseño arquitectónico. Las solicitudes adicionales se facturarán en función de su complejidad y del esfuerzo necesario para su implementación.

`R20` Se debe realizar una evaluación de vulnerabilidades antes de cada implementación. No debe quedar sin resolver ninguna vulnerabilidad crítica o alta . Los problemas deben tratarse y priorizarse para su solución en función de su gravedad.

`R21` El servicio contratado debe proporcionar componentes modulares para la ingesta, el procesamiento y la visualización de datos, lo que permite a los usuarios componer *pipelines* de datos personalizados sin necesidad de pagos adicionales o integraciones obligatorias de terceros.

6.2.5. Configuración de la evaluación

La configuración de la evaluación constituye una etapa clave para garantizar la trazabilidad entre los requisitos de calidad definidos y los datos que se recolectarán durante la monitorización del servicio. Su propósito es establecer los procedimientos para la recolección de datos, definir qué atributos se observarán, qué métricas se utilizarán y cómo se interpretarán los resultados de manera objetiva y reproducible.

Todas las actividades se alinearon con la ISO/IEC 25040 [14], asegurando la coherencia metodológica y rigor técnico. A partir de los requisitos definidos, se seleccionó un subconjunto de atributos del modelo de calidad general , dando lugar a un modelo personalizado para *Social Alert* (veáse 6.4). Este modelo tiene una descomposición jerárquica en características, subcaracterísticas, atributos medibles y métricas asociadas, lo que facilita su implementación en entornos operativos reales.

La selección de atributos se realizó en colaboración con el equipo evaluador, durante sesiones técnicas orientadas a garantizar la cobertura de los requisitos definidos en la sección anterior. La Tabla 6.2 presenta los atributos seleccionados, junto con sus definiciones operativas. Cada atributo está vinculado directamente a uno o más requisitos de calidad, lo que facilita su seguimiento y evaluación.

Tabla 6.2: Atributos de calidad seleccionados del modelo de calidad de SAlert.

No. Req.	**Atributo**	**Definición**
R1	Capacidad de CPU de VM	Núcleos de CPU máximos disponibles por VM.
R1	Uso de CPU del contenedor	Consumo actual de CPU del contenedor.
R1	Uso de CPU de VM	Consumo actual de CPU de la VM.
R2, R3	Utilización de memoria de VM	Consumo actual de memoria de la VM.

Table 6.2 (continua)

No. Req.	Atributo	Definición
R8	Uso de memoria del contenedor	Consumo actual de memoria del contenedor.
R2, R8	Capacidad de memoria	Memoria máxima asignada a VM.
R4	Utilización del disco	Información sobre el espacio libre/utilizado en el disco para operar y mantener el servicio.
R4, R5, R12	Capacidad de almacenamiento	Límites de los aspectos de calidad del almacenamiento (p. ej., lectura, escritura, tasas de E/S y tamaño).
R6, R9	Tasa de red entrante	Volumen de paquetes/datos recibidos por unidad de tiempo.
R6, R10	Tasa de red saliente	Volumen de paquetes/datos enviados por unidad de tiempo.
R7, R9, R10	Rendimiento de red	Velocidad de transferencia de datos sostenida durante un período de tiempo que garantiza el cumplimiento del SLA.
R11	Capacidad de gestión de procesos	Capacidad de los contenedores para gestionar cargas de trabajo (número de procesos).
R13	Tiempo de respuesta	Tiempo medio de finalización de las solicitudes por parte del servidor.
R14	Latencia	Retraso en el procesamiento de la solicitud de extremo a extremo.
R14	Rendimiento (Throughput)	Solicitudes gestionadas por unidad de tiempo por instancia (escalable). Una solicitud puede ser un trabajo, una tarea o una operación (p. ej., WIPS como interacciones web por segundo, OPS como operaciones por segundo, TPS como transacciones por segundo).
R15	Tasa de éxito	Porcentaje de solicitudes exitosas en relación con el total de solicitudes realizadas durante un período de tiempo específico.
R16	Tiempo de disponibilidad	Proporción de tiempo de actividad (99,99 %) durante el cual el servicio es accesible y puede ser utilizado por los clientes.
R17	Límites de conexión	Número máximo de conexiones simultáneas admitidas.
R17, R18	Mitigación de riesgos de seguridad	Detección y prevención proactivas de actividades no válidas o maliciosas e intentos de denegación de servicio.
R18	Alarmas monitorizadas	Detectar, generar y supervisar alertas de seguridad que protegen los datos y los servicios contra el acceso no autorizado.
R19, R20	Gestión de defectos	Priorización y corrección de fallos, errores o cualquier comportamiento inusual.
R21	Modularidad del servicio	Disponibilidad de componentes modulares con dependencias externas mínimas.
R21	Interoperabilidad con servicios dependientes	Nivel de interacciones eficientes entre el servicio y sus servicios subordinados en todos los procesos empresariales en los que participa el servicio.
R21	Composibilidad	Capacidad para componer de forma flexible canalizaciones de datos con módulos integrados.

Posteriormente, se realizó la selección de métricas para cada atributo, considerando tanto métricas independientes como específicas de la plataforma adaptadas al entorno operativo del servicio cloud. Algunas métricas presentaron desafíos de implementación debido a su bajo nivel de abstracción o a la heterogeneidad de las fuentes de datos requeridas. Para abordar estas limitaciones, se definieron operacionalizaciones concretas para cada métrica, incluyendo métodos alternativos cuando era necesario. Estas operacionalizaciones garantizan un mapeo coherente entre la especificación genérica de la métrica y el

concepto real representado en el artefacto cloud evaluado.

La disponibilidad de múltiples operacionalizaciones por atributo permite adaptar la función de medición a las restricciones de la plataforma o de los datos disponibles [101], asegurando flexibilidad sin comprometer la validez del proceso. La Tabla 6.3 muestra el conjunto final de atributos, métricas y operacionalizaciones seleccionados para el estudio de caso, ordenados alfabéticamente por el nombre del atributo. Esta estructura facilita la trazabilidad, la interpretación de los resultados y la replicabilidad del enfoque en otros contextos industriales.

Tabla 6.3: Atributos, métricas y operacionalizaciones del Modelo de Calidad para SAlert.

Atributo	**Métrica**	**Operacionalización**
Alarmas monitorizadas	Alarmas monitorizadas	Número de falsas alarmas supervisadas por Seguridad Corporativa
Capacidad de almacenamiento	Lectura de disco	Velocidad de lectura del disco
	Escritura de disco	Velocidad de escritura del disco
	Capacidad de almacenamiento	Tamaño de los datos almacenados, en MB o GB
Capacidad de CPU de VM	Núcleos de CPU	Número de núcleos de CPU asignados a la VM supervisada.
Capacidad de gestión de procesos	Procesos monitorizados	Número de procesos monitorizados
Capacidad de memoria	Capacidad de memoria	Tamaño del recurso de memoria del servicio, en MB o GB
Componibilidad	Componibilidad	$W * SM + W * ID$, donde SM es la modularidad del servicio, ID es la dependencia de interoperabilidad y W es el peso asignado en función de la importancia. La suma de los pesos debe ser igual a 1.
Fiabilidad operativa	Tasa de éxito (%)	$\frac{\#\ de\ transacciones\ exitosas}{\#\ total\ de\ transacciones\ disponibles\ en\ la\ carga\ de\ trabajo} \times 100$
	Tasa de error (%)	$\frac{\#\ de\ transacciones\ con\ error}{\#\ total\ de\ transacciones\ disponibles\ en\ la\ carga\ de\ trabajo} \times 100$
Gestión de defectos	Defectos	Número de defectos notificados por servicio cloud
Interoperabilidad con servicios dependientes	Dependencia de interoperabilidad	$\frac{\#\ de\ servicios\ dependientes\ con\ interaccion\ satisfactoria}{total\ de\ servicios\ dependientes\ en\ los\ procesos\ participantes}$
Latencia	Latencia por solicitud	Promedio ($tiempo\ de\ respuesta - tiempo\ de\ solicitud$), en ms o ns
Límites de conexión	Conexiones activas	Número de conexiones activas.

Table 6.3 (continua)

Atributo	**Métrica**	**Operacionalización**
	Conexiones bloqueadas	Número de conexiones bloqueadas.
Mitigación de riesgos de seguridad	Riesgos de seguridad	Número de riesgos de seguridad identificados de forma proactiva
Modularidad del servicio	Modularidad	$\frac{\#\ de\ elementos\ que\ no\ dependen\ de\ servicios\ externos}{\#\ de\ elementos\ en\ un\ servicio}$
Rendimiento de red	netBytesIn	Cantidad de bytes que entran por segundo
	netBytesOut	Cantidad de bytes que salen por segundo
	netPacketsIn	Cantidad de paquetes que entran por segundo
	netPacketsOut	Cantidad de paquetes que salen por segundo
Tasa de red entrante	rx packets	Número de paquetes recibidos
Tasa de red saliente	tx packets	Número de paquetes enviados
Tiempo de disponibilidad	Uptime ratio (%)	$\frac{Tiempo\ total\ de\ servicio - Tiempo\ total\ de\ servicio\ no\ disponible}{Tiempo\ total\ de\ servicio}100$ $\frac{Tiempo\ total\ de\ actividad}{Tiempo\ total} \times 100$
Tiempo de respuesta	Tiempo de respuesta	$\sum Tiempo\ de\ procesamiento\ SaaS + \sum Tiempo\ de\ procesamiento\ IaaS + \sum Tiempo\ de\ transmision\ de\ red$ $\frac{\sum Trep_i}{n}$, donde $Trep_i$ es el tiempo transcurrido entre el momento del envío de la solicitud y el momento de la recepción de la respuesta, y n es el número total de solicitudes transmitidas
Throughput	Throughput	$\frac{\#\ de\ solicitudes\ de\ servicio\ atendidas}{tiempo\ total\ de\ servicio}$, como solicitud por unidad de tiempo; donde una solicitud puede incluir trabajos, tareas u operaciones
Uso de CPU del contenedor	Uso de CPU	$\frac{Tiempo\ de\ CPU\ utilizado\ por\ el\ contenedor}{Tiempo\ total\ de\ CPU\ disponible} \times 100$
Uso de CPU de VM	Uso de CPU (%)	$\frac{Tiempo\ de\ CPU\ utilizado\ por\ VM}{Tiempo\ total\ de\ CPU\ asignado\ a\ VM} \times 100$
Uso de memoria del contenedor	Uso de memoria	Recurso de memoria del contenedor ocupado actualmente, en MB o GB
Utilización del disco	Disco libre	Cantidad de capacidad de disco disponible, en MB o GB
	Disco utilizado	Cantidad de capacidad de disco utilizada, en MB o GB
	Utilización del disco (%)	$\frac{Recurso\ de\ disco\ ocupado\ actualmente}{Recurso\ de\ disco\ total} \times 100$

Table 6.3 (continua)

Atributo	**Métrica**	**Operacionalización**
	blkio io bytes leídos	Cantidad de bytes leídos del disco duro
	blkio io bytes escritos	Cantidad de bytes escritos en el disco duro
Utilización de la memoria de VM	Memoria libre	Cantidad de capacidad de memoria disponible, en MB o GB
	Memoria utilizada	Cantidad de capacidad de memoria utilizada, en MB o GB
	Uso de la memoria (%)	$\frac{Recurso\ de\ memoria\ ocupado\ actualmente}{Recurso\ de\ memoria\ total} \times 100$

La primera versión del **modelo de calidad personalizado** fue desarrollada por el ingeniero de calidad, tomando como base el modelo genérico de calidad para servicios en la nube. Esta versión inicial fue sometida a un proceso de validación colaborativa mediante un taller con el equipo evaluador, lo que permitió identificar oportunidades de mejora y realizar ajustes sustantivos.

1. *Identificación de cuellos de botella en la BD.* Durante el taller, la empresa identificó la instancia de la base de datos PostgreSQL como un componente crítico que presentaba limitaciones de rendimiento. En respuesta, se incorporaron nuevos requisitos de calidad (`R22`, `R23`, `R24`) orientados a permitir una monitorización más granular de este componente. Estos requisitos se centraron en la eficiencia de la BD.

2. *Incorporación de atributo "eficiencia de BD".* Aunque el modelo original incluía una métrica relacionada con las conexiones a BD, el equipo evaluador consideró que era insuficiente para capturar la eficiencia operativa. Por ello, se introdujo un nuevo atributo —*eficiencia de la base de datos*— y las métricas específicas obtenidas directamente de la herramienta de monitorización utilizada (New Relic). Estas métricas no estaban contempladas originalmente, lo que evidencia la necesidad de extender el modelo genérico utilizado como base para la selección de métricas.

3. *Simplificación de la métrica de disponibilidad.* Durante la validación del modelo, se acordó que una única implementación de la métrica *uptime ratio* era suficiente para evaluar la disponibilidad del servicio. El método seleccionado calcula la disponibilidad como la diferencia entre el tiempo total de servicio y el tiempo de inactividad.

Estas mejoras se consolidan en la Tabla 6.4, que presenta la refinación y extensión del modelo inicial y que además refleja un mayor grado de alineación con las necesidades operativas de la organización y con las expectativas de los expertos en el dominio.

Además, la estructura del modelo permitió al ingeniero de calidad realizar ajustes dinámicos—como la actualización o eliminación de métricas y operacionalizaciones— según

fuera necesario. Esta flexibilidad contribuyó a minimizar redundancias y a mejorar la relevancia de las mediciones a recolectar, fortaleciendo la utilidad práctica del modelo.

Tabla 6.4: Requisitos, atributos, métricas y operacionalizaciones modificados para SAlert: ⊕ añadidos o ⊖ eliminados

# Req.	**Atributo**	**Métrica**	**Operationalización**
R16	Tiempo de disponibilidad	Uptime ratio (%)	$\ominus \frac{Tiempo\ total\ de\ actividad}{Tiempo\ total} \times 100$
R17	Límites de conexión	Conexiones activas	Número de conexiones activas.
⊕ R22		Conexiones bloqueadas	Número de conexiones bloqueadas.
⊕ R22	⊕ Eficiencia de la BD	⊕ *Commits* de BD	Número de *commits* por segundo
		⊕ *Deadlocks* de BD	Número de *deadlocks* por segundo
⊕ R23		⊕ *Rollbacks* de BD	Número de *rollbacks* por segundo
		⊕ Tasa de eliminación	Número de filas eliminadas por segundo
⊕ R24		⊕ Tasa de inserción	Número de filas insertadas por segundo
		⊕ Tasa de recuperación	Número de filas recuperadas por segundo
		⊕ Tasa de actualización	Número de filas actualizadas por segundo

R22: La base de datos PostgreSQL debe admitir un máximo de 200 conexiones, con menos de 150 conexiones activas y ninguna conexión bloqueada.
R23: La base de datos PostgreSQL debe procesar más de 30 *commits* y evitar *deadlocks*.
R24: Las operaciones de la base de datos PostgreSQL deben tener al menos:
i) más de 1500 filas recuperadas;
ii) más de 10 filas insertadas; y
iii) más de 4 filas actualizadas o eliminadas.
Eficiencia de la BD: Número de operaciones de la base de datos ejecutadas y tiempo empleado.

La Figura 6.4 ilustra la descomposición jerárquica del modelo de calidad personalizado de *SAlert*, mostrando cómo las características de calidad se desglosan en subcaracterísticas, atributos y métricas. Esta representación destaca las prioridades de los diferentes actores: mientras el gerente de producto y los expertos en el dominio valoran enormemente la eficiencia del rendimiento, la fiabilidad y la seguridad, los desarrolladores de software priorizan la mantenibilidad, la usabilidad y la compatibilidad.

6.2.6. Diseño de la evaluación

En esta fase, el experto del dominio definió los **criterios de decisión** necesarios para establecer el nivel de calidad aceptable de cada métrica del caso seleccionado, basándose en su experiencia previa en la monitorización de la calidad de los servicios cloud en la organización. Estos criterios se expresaron como objetivos cuantificables o umbrales específicos para cada métrica y fueron esenciales para interpretar los resultados de las métricas de forma objetiva y reproducible.

Un enfoque clave de esta fase fue el mapeo entre cada métrica seleccionada a sus respectivas fuentes de datos. El objetivo principal fue determinar cómo las métricas in-

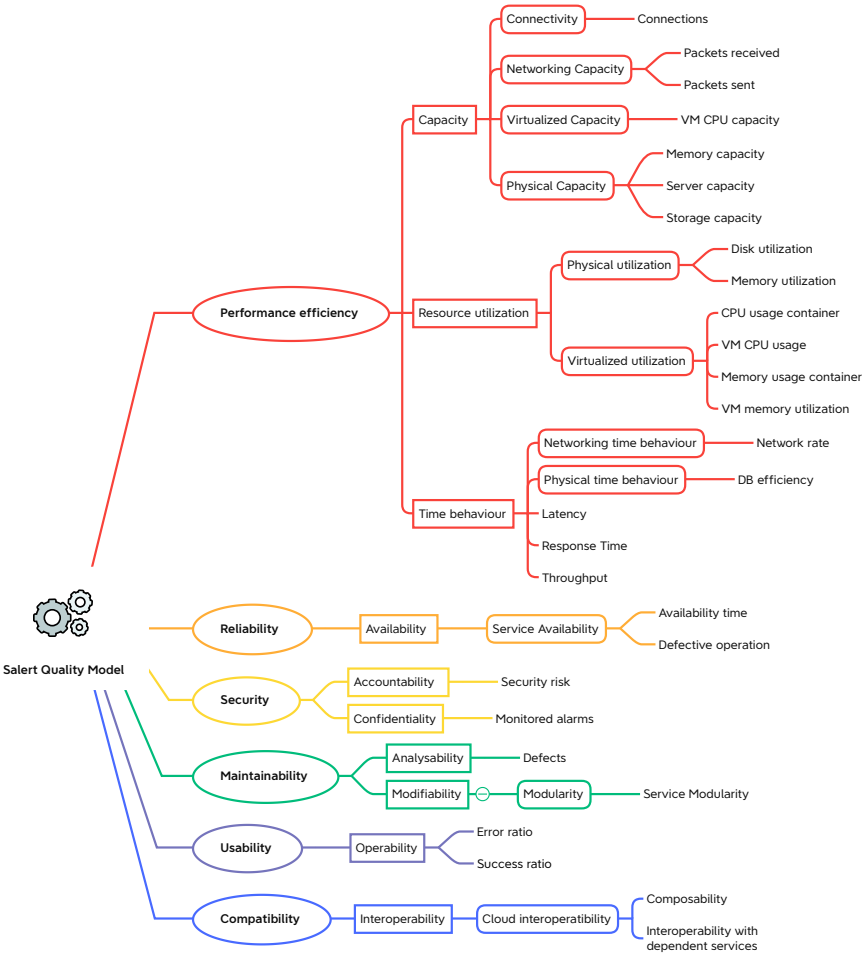


Figura 6.4: Modelo de calidad para la evaluación del servicio SAlert.

dependientes del modelo de calidad podían ser operacionalizadas (calculadas) dentro de una plataforma cloud específica (en este caso, Linode). Para ello se identificaron los métodos más adecuados para la recolección de datos, incluyendo agentes del sistema, APIs, herramientas de monitorización nativas y librerías externas.

Con el fin de garantizar la generalización del modelo de calidad y su adaptabilidad a la plataforma, se amplió la definición de las métricas para incluir funciones o métodos de medición capaces de calcular en diferentes plataformas y artefactos cloud. Esta estrategia permitió que el modelo no dependiera exclusivamente de una plataforma particular, facilitando su aplicación en escenarios diversos.

Por ejemplo, la métrica de *tiempo de respuesta* (RT) en el modelo de calidad para el servicio *SAlert* fue definida con dos operacionalizaciones distintas (6.1) y (6.2) , ambas diseñadas para ser independientes de la plataforma. Una de ellas calcula el tiempo de

respuesta como el intervalo entre el envío de una solicitud por el usuario y la recepción de una respuesta del servicio cloud. Esta definición abarca tanto el tiempo de procesamiento a nivel de SaaS e IaaS, como el tiempo de transmisión de la red (es decir, la latencia de la red).

$$\begin{aligned} RT = &\sum tiempo\ de\ procesamiento\ SaaS \\ &+ \sum tiempo\ de\ procesamiento\ IaaS \\ &+ \sum tiempo\ de\ transmision\ de\ red \end{aligned} \tag{6.1}$$

$$RT = \sum \frac{Trep_i}{n}, \tag{6.2}$$

donde $Trep_i$ es el tiempo transcurrido entre el instante del envío de la solicitud y el instante de la recepción de la respuesta, y n es el número total de solicitudes transmitidas.

Para implementar esto en la plataforma Linode [153], el experto en el dominio realizó las siguientes actividades:

1. Instalación de agentes para monitorización para la recopilación de los datos necesarios que permitan calcular las métricas derivadas, por ejemplo *RT, uptime ratio*,
2. Configuración de una herramienta de monitorización de terceros para recuperar datos, medir las métricas base y calcular las derivadas de cada componente.

Las métricas base utilizadas en estas operacionalizaciones incluyen: tiempo de procesamiento de SaaS, tiempo de procesamiento de IaaS, tiempo de transmisión de red, tiempo de solicitud, tiempo de respuesta y número de solicitudes transmitidas.

Una vez detalladas las operacionalizaciones específicas para cada métrica, se establecieron los criterios de decisión. Estos criterios permiten a los *stakeholders* interpretar los resultados de las mediciones mediante la definición de umbrales u objetivos (como valores numéricos, percentiles o rangos aceptables) que representan un rendimiento aceptable u óptimo. Los umbrales fueron definidos considerando múltiples fuentes: datos históricos del servicio, expectativas de los *stakeholders*, referencias del sector y evaluaciones previas. No obstante, también se tomó en cuenta el dominio específico en el que opera el servicio cloud, lo que garantizó que los criterios reflejaran las particularidades del contexto organizacional.

La definición de estos criterios se realizó de forma colaborativa, con la participación de los expertos del dominio y los *stakeholders* del equipo de evaluación. Este enfoque participativo aseguró que los niveles de calidad aceptables fueran realistas, relevantes y alineados con los objetivos estratégicos de la organización.

Esta fase finalizó con una configuración de evaluación que se ajusta estrechamente al *Plan de Evaluación* descrito en la ISO/IEC 25040 [14]. En ella, se especifican todos los elementos necesarios para monitorizar el servicio cloud durante su ejecución, incluyendo: requisitos de calidad, características, atributos, métricas y operacionalizaciones, así como umbrales de evaluación y ajustes de configuración. Estos elementos, integrados de forma sistemática, respaldan un proceso de evaluación objetivo,adaptable y alineado tanto a la plataforma en la nube como a las necesidades del contexto organizacional.

6.3. Recolección de datos

La recolección de datos constituye una fase crítica en la evaluación empírica, ya que proporciona la evidencia necesaria para validar el modelo de calidad en condiciones operativas reales. En este estudio, se emplearon métodos de recolección indirecta e independiente, conforme a la clasificación propuesta por Runeson et al. [37], priorizando el uso de herramientas automatizadas de monitorización y el análisis documental del servicio evaluado.

Para capturar **datos cuantitativos**, se utilizó una *herramienta de monitorización de terceros (New Relic)* que permitió observar el comportamiento del servicio *SAlert* en tiempo real. Esta herramienta fue configurada para recolectar métricas específicas asociadas a los atributos de calidad definidos previamente, abarcando aspectos de IaaS, PaaS y SaaS. Los datos fueron extraídos directamente de los componentes del sistema, incluyendo máquinas virtuales, contenedores *Docker*, bases de datos y APIs, asegurando una cobertura integral de la arquitectura evaluada.

La configuración de la herramienta incluyó la instalación de *agentes de monitorización*, la *integración con APIs* y la *definición de intervalos de muestreo para cada métrica.* Las métricas recolectadas incluyeron indicadores de rendimiento, disponibilidad, uso de recursos, latencia, tasa de éxito, número de procesos activos y alarmas de seguridad, entre otros. Esta instrumentación permitió capturar datos de alta resolución, representativos del comportamiento del servicio bajo condiciones reales de operación.

Con el fin de reforzar la trazabilidad entre el modelo de calidad definido en el Capítulo 5 y su operacionalización en el estudio de caso, se ha definido un mapeo explícito entre los elementos conceptuales del modelo y las métricas monitorizadas en el servicio SAlert. Este mapeo permite vincular de manera sistemática las subcaracterísticas y atributos de calidad con métricas concretas, especificando, además, su fuente de datos, la granularidad temporal de las mediciones, los umbrales de cumplimiento y las reglas de decisión utilizadas durante la evaluación.

La Tabla 6.5 sintetiza este mapeo operativo, proporcionando una visión compacta y reproducible del proceso de evaluación. En particular, para cada subcaracterística y atributo del modelo de calidad personalizado se identifican: (i) las métricas de QoS empleadas, (ii) las herramientas o fuentes de monitorización utilizadas para su captura, (iii) la resolución temporal con la que se recolectaron los datos, (iv) los umbrales definidos para determinar el grado de cumplimiento, y (v) las reglas de decisión aplicadas para interpretar los resultados. Esta explicitación permite reconstruir de forma precisa el proceso seguido y facilita tanto la replicación del estudio como su adaptación a otros servicios cloud.

De este modo, la evaluación no se limita a la observación aislada de métricas, sino que se apoya en una correspondencia explícita entre el modelo conceptual de calidad y la evidencia empírica obtenida mediante la monitorización. Esta trazabilidad garantiza la coherencia interna del estudio, reduce ambigüedades en la interpretación de los resultados y refuerza la aplicabilidad del marco propuesto en contextos industriales reales.

Tabla 6.5: Mapeo operacional de métricas QoS con criterios de decisión

Subcaracterística	**Atributo**	**Métrica QoS**	**Intervalo y Fuente**	**Umbral**	**Regla de decisión** (Cumple)
Capacidad					
Conectividad	Limites de conexiones	Conexiones activas	1 min / VM New Relic	≤ 150	media(conexiones, 5 min) $\leq$ umbral
Capacidad física	Capacidad de memoria	Memoria asignada	5 min / VM New Relic	≥ 8 GB	media(memoria, 24 h) $\geq$ umbral
Utilización de recursos					
Utilización virtualizada	Uso de memoria	Uso de memoria (%)	5 min / VM New Relic	$\leq 80\,\%$	media(uso_memoria, 5 min) $\leq$umbral
Analizabilidad	Gestión de defectos	defectos	Diaria / servicio Reporte de seguridad	$= 0$	defectos = umbral
Operabilidad	Fiabilidad operativa	Tasa de éxito (%)	5 min / VM New Relic	1 h / servicio $\geq 99\,\%$	p95(tasa_éxito) $\geq$ umbral

El **periodo de monitorización seleccionado** para el estudio de caso comprende *una semana* completa de operación del servicio SAlert. Esta ventana temporal fue elegida de forma deliberada por considerarse representativa del comportamiento operativo habitual del servicio, incluyendo ciclos completos de carga diaria y variaciones propias de los días laborables. Durante este periodo se observaron fluctuaciones de demanda, picos de utilización y episodios de mayor presión sobre los recursos, lo que permitió capturar patrones relevantes de rendimiento y fiabilidad sin introducir sesgos asociados a periodos anómalos o excepcionales.

Si bien una ventana de observación más prolongada podría revelar efectos de estacionalidad a mayor escala o incidentes poco frecuentes, el objetivo principal de este estudio no es caracterizar exhaustivamente el comportamiento a largo plazo del servicio, sino validar la aplicabilidad del modelo de calidad y del proceso de evaluación propuestos en un entorno real. En este contexto, la semana seleccionada proporciona un compromiso adecuado entre representatividad operativa, volumen de datos manejable y viabilidad del estudio, permitiendo evaluar de forma consistente la trazabilidad entre el modelo de calidad, las métricas monitorizadas y los resultados obtenidos.

Los **datos cualitativos**, por su parte, se obtuvieron mediante entrevistas semiestructuradas realizadas al equipo técnico de Golden S.A., integrado por el propietario del producto, el gerente del proyecto y el experto en el dominio. Estas entrevistas permitieron complementar la información técnica con percepciones sobre la utilidad, adaptabilidad y relevancia del modelo de calidad en el contexto organizacional. Las respuestas fueron codificadas y analizadas para identificar patrones de aceptación, desafíos de implementación y oportunidades de mejora.

La triangulación de fuentes (herramientas de monitorización, documentación técnica y entrevistas) permitió fortalecer la validez interna del estudio al contrastar los resultados obtenidos desde distintas perspectivas. Asimismo, se garantizó la trazabilidad entre los datos recolectados y los requisitos de calidad definidos, lo que facilitó el análisis posterior

y la interpretación objetiva de los resultados.

6.4. Análisis de datos

El análisis de datos constituye una etapa fundamental para interpretar los resultados obtenidos durante la monitorización del servicio y evaluar el cumplimiento de los requisitos de calidad definidos.

6.4.1. Datos cuantitativos

Las métricas seleccionadas para *SAlert* (véanse Tablas 6.3 y 6.4) fueron asignadas sistemáticamente a los componentes del servicio que se muestran en la Figura 6.3, de acuerdo con los requisitos de calidad establecidos.

Estas métricas se monitorizaron y evaluaron, y los resultados se presentan en las Tablas 6.6 y 6.7. Estas tablas incluyen información detallada sobre las métricas evaluadas, incluyendo los umbrales definidos por el experto del dominio para cada métrica y que sirven como criterios de evaluación. La columna de *medida* indica el resultado de la medición tras aplicar la métrica al artefacto en la nube. La columna de *unidad* define la unidad de medida, que se refiere a la unidad o escala estandarizada utilizada para cuantificar y expresar un atributo específico [111], mientras que la columna *¿Cumple?* indica si los valores observados satisfacen o no los criterios de calidad establecidos. Esta comparación permite determinar objetivamente el cumplimiento de los requisitos de calidad; además, hay que destacar que algunas métricas están asociadas a múltiples componentes y que sus umbrales pueden variar en consecuencia.

Todas las métricas se han categorizado por tipo de servicio y niveles de recursos. Además, se presentaron gráficos de los datos de medición recopilados durante un período específico (una semana) para proporcionar una visión completa de las métricas clave. Estos gráficos se organizaron en cuadros de mando, diseñados para cumplir con los requisitos de un tipo de servicio cloud y sus respectivos componentes. El cuadro de mando de SaaS se centra en las métricas de rendimiento de las API (Figura 6.5); el cuadro de mando de PaaS abarca las métricas de uso de la instancia de base de datos PostgreSQL (Figura 6.6) y del contenedor; y el cuadro de mando de IaaS se centra en las métricas relacionadas con el uso de VM y la red (Figura 6.7).

Tabla 6.6: Resultados de la monitorización de SAlert.

Métrica	Umbral	Cálculo	Medida	Unidad	¿Cumple?
SaaS (API)					
Alarmas monitorizadas	$= 0$	Zona de decisión, $\alpha = 2$	3	alarmas	**No**
Componibilidad	> 0.80		0.964		S

Tabla 6.6 (continúa)

Métrica	Umbral	Cálculo	Medida	Unidad	¿Cumple?
Defectos	$= 0$	Zona de decisión, $\alpha = 5$	8	defectos	**No**
Dependencia de interoperability	> 0.70		1		S
Modularidad	> 0.70		0.929		S
Riesgos de seguridad	$= 0$	Zona de decisión, $\alpha = 5$	17	riesgos	**No**
Tasa de error (%)	<= 1	P_{95}	0.98	%	S
Tasa de éxito (%)	>= 99	P_{95}	99.02	%	S
Tiempo de respuesta [1]	<= 50	Avg.	26.9	µs	S
Tiempo de respuesta [2]	<= 50	Avg.	88.5	µs	**No**
Throughput	>= 150	Avg.	165	rps	S
Uptime ratio (%)	>= 99.99	P_{95}	99.2	%	**No**
Uso de CPU (%)	<= 50		35	%	S
Uso de memoria (%)	<= 80		51.8	%	S
Uso de memoria	<= 1		0.89	GB	S
PaaS (Instancia de BD PostgreSQL)					
Conexiones activas	<= 150	Avg.	75	conexiones	S
Conexiones bloqueadas	$= 0$	Max	0	conexiones	S
Commits de BD	> 30	Avg.	40	*commits*	S
Deadlocks de BD	$= 0$	Avg.	0	*deadlocks*	S
Lectura de disco	>= 500	Min	549	Bps	S
Rollbacks de BD	< 5		1	*rollbacks*	S
Tasa de actualización	> 4	Avg.	4.2	filas/s	S
Tasa de eliminación	> 4	Avg.	3.9	filas/s	**No**
Tasa de inserción	> 10	Avg.	9	filas/s	**No**
Tasa de recuperación	$> 1,500$	Avg.	1,330	filas/s	**No**
IaaS (VM)					
Almacenamiento asignado	>= 120		120	GB	S
Memoria asignada	>= 8		8	GB	S
Disco libre	>= 60		73.3	GB	S
Disco utilizado	<= 60		47.5	GB	S
Escritura de disco	>= 35		37	GBps	S
Lectura de disco	>= 20		24	GBps	S
Memoria libre	>= 2		3.6	GB	S
Memoria utilizada	<= 6		4.4	GB	S
netBytesIn	>= 200		295	KBps	S
netBytesOut	>= 35		38	KBps	S
netPacketsIn	>= 200		295	pps	S
netPacketsOut	>= 50		58	pps	S
Núcleos de CPU	>= 4		4	núcleos	S
rx packets	>= 35		39	packets	S
tx packets	>= 35		38	packets	S
Uso de CPU (%)	<= 80		32.2	%	S
Uso de memoria (%)	<= 60		56.7	%	S
Utilización del disco (%)	<= 50		30.4	%	S

Zona de decisión = Umbral $+\alpha$, donde α representa la tolerancia operacional definida por los expertos.
rps: solicitudes por segundo; KBps: Kbytes por segundo; pps: paquetes por segundo

En el nivel **SaaS**, se evaluaron 15 métricas de la API, incluyendo tasa de éxito y tiempo de actividad (*uptime ratio*). De estas, cinco métricas no alcanzaron los umbrales establecidos para las métricas —tiempo de respuesta [2], riesgos de seguridad, alarmas

monitorizadas, defectos y uptime ratio—. Estas deficiencias destacaron especialmente las vulnerabilidades de seguridad y los desafíos relacionados con las modificaciones del servicio, que fueron priorizados por el equipo debido a su posible impacto en la mantenibilidad y seguridad del sistema.

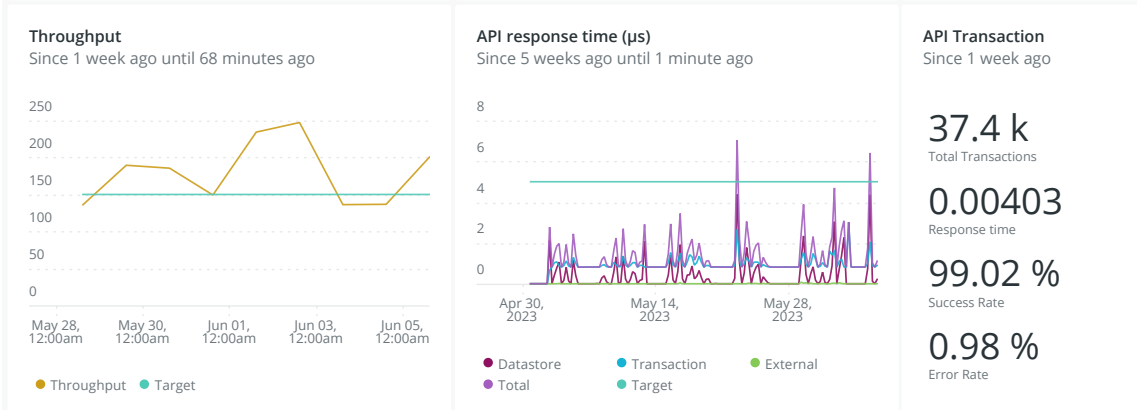


Figura 6.5: Resultados de medición SaaS.

El análisis del modelo SaaS destacó métricas clave como los tiempos de respuesta y las tasas de éxito/error. Como se muestra en la Figura 6.5, durante el período de monitorización de una semana, el *throughput* fluctuó entre 136 y 247 solicitudes por segundo (rps), con un promedio de 165 rps. Si bien esto indica un rendimiento del sistema generalmente eficiente, las fluctuaciones observadas revelan que pequeños cambios en el tráfico podrían provocar caídas por debajo de los umbrales aceptables, lo que sugiere limitaciones en la priorización de solicitudes y una posible sobrecarga durante los períodos pico. El tiempo de respuesta [1] promedio del sistema se calculó en 26,9 microsegundos (µs), medido a partir del tiempo transcurrido entre la solicitud y la respuesta en relación con el número total de solicitudes.

Además, la API alcanza una alta tasa de éxito del 99,02 % en 37 400 transacciones, con una tasa de error correspondiente del 0,98 %, lo que indica casos de fallo aislados. Los resultados obtenidos para estas métricas demuestran la robustez del servicio cloud y su capacidad para satisfacer la demanda dinámica, manteniendo al mismo tiempo la fiabilidad operativa.

En el nivel **PaaS**, se evaluaron dos componentes principales: la instancia de BD PostgreSQL, que incluyó 10 métricas, y los siete contenedores *Docker* incluyendo 14 métricas; de ellas, seis se evaluaron y ocho estaban en observación.

En la instancia de BD PostgreSQL, tres métricas—tasa de eliminación, inserción y recuperación—mostraron incumplimiento en cuanto a la eficiencia de la base de datos. Se registraron un promedio de 75 conexiones activas, utilizando menos del 40 % de la capacidad disponible, además, 40 *commits*, un *rollback* y ningún bloqueo (deadlocks). Las tasas de operación se midieron en *filas/s* y fueron 9 al insertar, 1330 al recuperar, 3,9 al

eliminar y 4,2 al actualizar. De estas, solo la tasa de actualización superó el umbral, lo que sugiere que, si bien el servicio está operativo, existen ciertas limitaciones, especialmente en la configuración de las operaciones de la base de datos. Abordar estas áreas es vital para garantizar la escalabilidad de los recursos y mantener la eficiencia general del rendimiento.

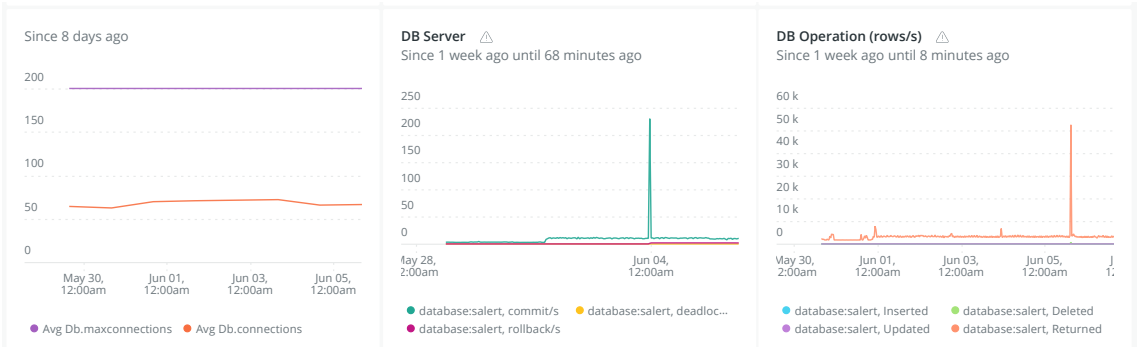
Tabla 6.7: Resultados de monitorización de PaaS (Contenedores).

PaaS (Contenedores)	Condición	Mongo srv		Redis srv		Downloader		Authentication		Gateway		Eureka		User interface			**¿Cumple?**
Métrica		U	M	U	M	U	M	U	M	U	M	U	M	U	M	Unit	
Uso de CPU (%)	≤	20	1.49	20	0.12	100	**103**	20	0.1	20	0.07	20	0.45	20	0	%	S
Capacidad de memoria	≤	2	2	8	0.4	8	0.4	8	0.4	8	0.4	8	0.4	8	0.4	GB	S
Uso de memoria	≤	1.6	1.54	0.4	0.003	0.4	0.22	0.4	0.37	0.4	0.29	0.4	0.26	0.4	0.003	GB	S
netBytesIn	≥	20	36.15	5.0	8.3	5.0	6.1	0.5	2.6	5.0	6.4	5	5.9	0.5	0.29	MBps	S
netBytesOut	≥	50	95.4	5.0	7.5	0.15	1.04	0.15	0.24	0.15	1.56	0.15	0.16	0.15	0.35	MBps	S
Procesos monitorizados	≥	150	197		5		8	50	50		43		72		5	proc.	S
rx packets	*o*		2.31		10.2		8.9		1.13		2.77		3.89		0.96	packets	
tx packets	*o*		22.9		19.3		0.97		1.02		2.81		2.23		0.34	packets	
Lectura de disco	*o*		7.6		4.63		0.08		4.59		0.9		29.2		0.28	GBps	
Escritura de disco	*o*		14.1		0.04		0		0		0.001		0		0	GBps	
blkio io read	*o*		20.1		0.1		0.1		0.98		1		0.8		0.19	KBps	
blkio io write	*o*		31.6		0		0		0		0		0		0	KBps	
Conexiones activas	*o*				9											conn.	
Conexiones bloqueadas	*o*				0											conn.	

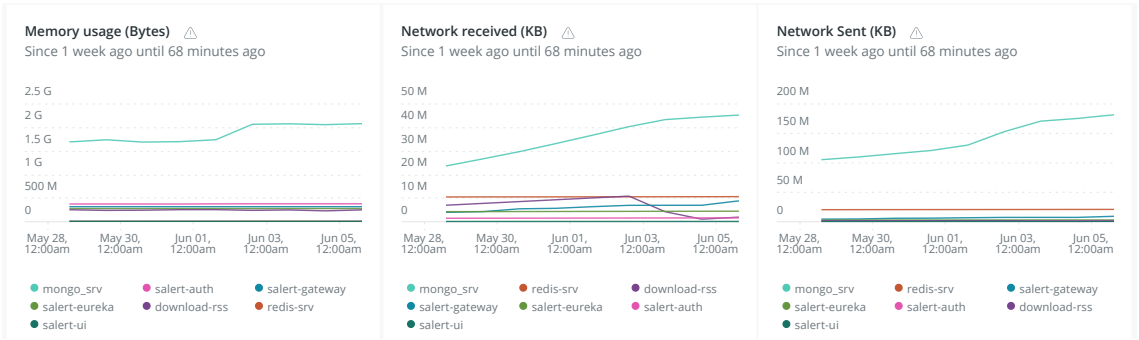
T: umbral; M: medida; *o*: en observación; proc.: procesos; conn.: conexiones

La Figura 6.6.a muestra las métricas clave de PostgreSQL, incluyendo conexiones activas, *commits*, *deadlocks* y *rollbacks*, junto con las métricas recientemente introducidas relacionadas con las tasas de operación de BD (inserción, recuperación, eliminación y actualización). Con base en el análisis, el equipo de evaluación concluyó que sería necesario ajustar los parámetros de la base de datos y corregir las limitaciones identificadas para mejorar la escalabilidad y la eficiencia de los recursos.

En cuanto a los contenedores (*dockers*), todas las métricas evaluadas cumplen con los umbrales establecidos; ciertas métricas, como netBytesIn y netBytesOut, sugieren posibles problemas en la gestión del flujo de datos entrante y saliente, especialmente en el contenedor *Mongo srv*. Si bien las métricas "*en observación*" permanecen bajo supervisión y aún no forman parte de la evaluación formal de calidad, proporcionan un contexto valioso para la mejora continua. En cuanto al rendimiento de red, *Mongo srv* recibió 2,31 paquetes y



(a) PaaS instancia de base de datos.



(b) PaaS contenedores.

Figura 6.6: Resultados de medición PaaS.

transmitió 22,9, mientras que *Redis srv* recibió 10,2 y transmitió 19,3. Estos hallazgos son consistentes con el comportamiento esperado en operaciones de ingesta y almacenamiento de gran volumen, donde *downloader* soporta una mayor carga de procesamiento intensivo. Estos datos son cruciales para comprender la asignación de recursos de los contenedores y optimizar las cargas de trabajo para satisfacer la demanda. *Mongo srv* también registró el mayor consumo de recursos—en términos de uso de memoria—entre todos los contenedores, con un promedio de 1,54 GB durante el período de evaluación, su uso fluctuó entre 1,4 GB y 2 GB, mientras que los demás contenedores se mantuvieron por debajo de los 400 MB.

La Figura 6.6.b muestra el uso de memoria y el flujo de la actividad de la red. *Mongo srv* experimentó picos cercanos a 50 MBps en la tasa entrante y 200 MBps en la tasa saliente de red, correspondientes a la ingesta de datos (p. ej., importaciones masivas de datos) y a los procesos de exportación (p. ej., exportaciones de datos o uso de API). Estas fluctuaciones, observadas a lo largo de la semana, probablemente reflejen eventos programados o variaciones de carga en función del tiempo. *Redis srv*, es utilizado principalmente para el acceso rápido a datos; expuso patrones de actividad de red relacionados con actualizaciones frecuentes y tareas de sincronización. Otros contenedores mantuvieron tendencias de uso constantes y no se detectaron anomalías. Estos resultados confirman el comportamiento

esperado con un alto consumo de recursos, dado el alto volumen de transacciones durante el proceso de ingesta, y enfatizan la importancia de gestionar eficazmente los recursos de los contenedores para evitar su saturación.

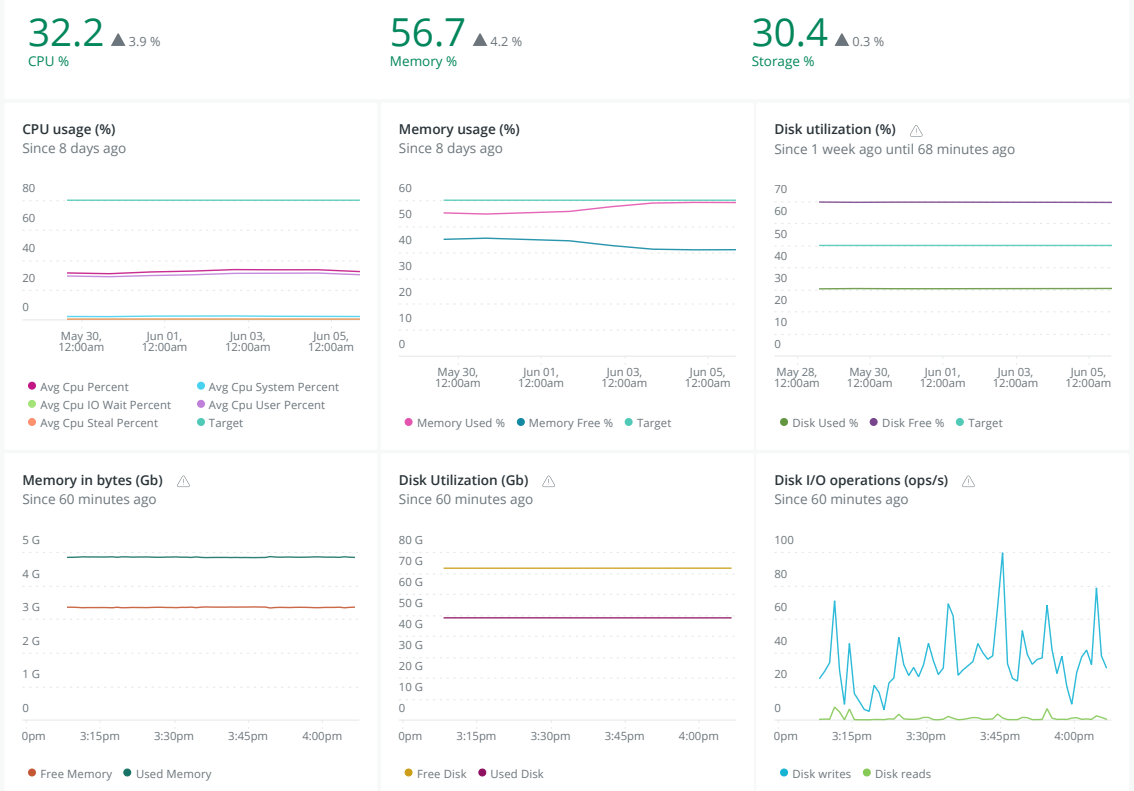


Figura 6.7: Resultados de medición IaaS.

En el nivel **IaaS** se evaluaron 18 métricas relacionadas con máquinas virtuales, almacenamiento y redes. Todas las métricas cumplieron con los umbrales esperados, lo que confirma la capacidad de la infraestructura para soportar de forma fiable servicios de nivel superior. La Tabla 6.7 y la Figura 6.7 resumen los resultados: uso promedio de CPU del 32.2 %, memoria del 56.7 % y disco del 30.4 %, todos dentro de los límites establecidos (80 %, 60 % y 50 %, respectivamente). El uso de recursos mostró una tendencia estable, sin picos ni caídas significativas, aunque se observó una actividad frecuente de escritura en disco, lo que podría afectar el rendimiento bajo condiciones de carga intensiva

En términos de capacidad, se registró una utilización de 4.4 GB de memoria y 47.5 GB de espacio de disco, lo que respalda el cumplimiento de los requisitos de calidad definidos. La memoria mostró una tendencia de crecimiento sostenido, mientras que el uso de CPU y disco se mantuvo estable. Estos resultados confirman que la infraestructura es adecuada para soportar los componentes PaaS y SaaS del servicio SAlert y que su configuración actual permite una operación fiable y eficiente.

6.4.2. Datos cualitativos

Con el fin de complementar la evaluación cuantitativa basada en métricas y reforzar la validez empírica del estudio de caso, se incorporó el *feedback* cualitativo de distintos stakeholders involucrados en la operación y gestión del servicio SAlert. En particular, participaron perfiles con responsabilidades técnicas y de gestión, incluidos ingenieros DevOps, responsables de operación del servicio y personal con funciones de supervisión de la calidad y del cumplimiento de SLA. Estos roles fueron seleccionados por su conocimiento directo del funcionamiento del servicio y su implicación en la toma de decisiones relacionadas con la calidad.

La recogida de *feedback* se llevó a cabo mediante entrevistas semiestructuradas y sesiones de revisión guiadas, apoyadas en el modelo de calidad personalizado y en los resultados de la evaluación obtenidos. Durante estas sesiones se discutieron aspectos como la relevancia de las características y atributos seleccionados, la adecuación de las métricas empleadas, la interpretabilidad de los indicadores generados y la utilidad del modelo para identificar áreas prioritarias de mejora. Como evidencia, se recogieron observaciones recurrentes relacionadas con la necesidad de ajustar determinados umbrales de rendimiento, la conveniencia de priorizar atributos asociados a la disponibilidad y al comportamiento bajo carga, y el valor del modelo para ofrecer una visión integrada de la calidad más allá de métricas aisladas.

El *feedback* obtenido se utilizó para refinar iterativamente el modelo de calidad y el proceso de evaluación. En particular, mediante la revisión de umbrales de cumplimiento, la reformulación de reglas de decisión y la confirmación de la relevancia de determinados atributos en el contexto operativo del servicio. Este enfoque es coherente con el carácter iterativo del proceso de evaluación definido en la ISO/IEC 25040, en el que la retroalimentación empírica constituye un elemento clave para mejorar la adecuación y la utilidad del modelo. Concretamente, se identificó que las reglas binarias aplicadas a ciertas métricas (p.ej., alarmas monitorizadas, defectos y riesgos de seguridad) resultaban demasiado rígidas para entornos cloud reales y podían dar lugar a falsos negativos, al no considerar la tolerancia operativa ni la criticidad contextual de los eventos. A partir de esta observación, se introdujeron zonas de decisión que permiten clasificar el cumplimiento de manera gradual, mejorando la alineación con las prácticas de gestión de riesgos y generando bases para una integración futura con mecanismos de clasificación y predicción usando ML.

Asimismo, el uso de percentiles en las métricas (p.ej., tasa de error, tasa de éxito, *uptime ratio*) fue validado por los *stakeholders* como una práctica adecuada y consistente con la definición de SLAs. Desde esta perspectiva, el incumplimiento del *uptime ratio* no contradice el buen rendimiento general, sino que evidencia una degradación temporal o un evento transitorio. En consecuencia, se plantea incorporar reglas de decisión contextuales que faciliten el análisis longitudinal del comportamiento del servicio y su uso como objetivo en tareas de predicción de la calidad.

En relación con las diferencias en los resultados de las métricas de tiempo de respuesta, el *feedback* permitió descartar inconsistencias en la medición. Por el contrario, estas diferencias evidenciaron la heterogeneidad interna del SaaS, asociada a distintos *endpoints*,

rutas funcionales y componentes internos. Esta interpretación se refuerza por el modelo de arquitectura propuesto (véase 2.1.3, el cual facilita la trazabilidad de la métrica hacia el componente y a la capa arquitectónica correspondiente. Como mejora derivada, se propone enriquecer estas métricas incorporando el contexto o componente evaluado (p. ej., API pública frente a API interna).

Los resultados cualitativos también evidencian que el grado de cumplimiento de los atributos de calidad varía de manera sistemática según la capa arquitectónica analizada. Mientras que la capa de infraestructura (IaaS) presenta un alto nivel de cumplimiento, las desviaciones detectadas en las capas PaaS y SaaS se concentran en atributos relacionados con la fiabilidad, la seguridad y la eficiencia de desempeño.

Este proceso de retroalimentación contribuyó a mejorar la adecuación del modelo a las necesidades reales de la organización y a reforzar su utilidad como instrumento de apoyo a la toma de decisiones. Aunque el *feedback* recogido tiene un carácter cualitativo y exploratorio, su integración proporciona evidencia adicional sobre la aplicabilidad práctica del marco propuesto y su alineación con las expectativas de los *stakeholders* industriales.

6.5. Discusión de los resultados

Los resultados de este estudio proporcionaron una evaluación integral de las capas de infraestructura, plataforma y software del servicio, lo que permitió a la empresa realizar un análisis exhaustivo e identificar posibles problemas en una etapa temprana, antes de que pudieran derivar en fallos críticos, como la no disponibilidad o la degradación del rendimiento del servicio. Esto sentó las bases para una toma de decisiones informada y basada en datos.

6.5.1. Impacto del modelo en la evaluación y gestión de la calidad

Un beneficio clave del modelo de calidad y del enfoque de evaluación propuestos reside en empoderar a los *stakeholders* para que tomen decisiones basadas en evidencia empírica. Al identificar las métricas que no cumplían con las expectativas, la empresa pudo comprender de forma estructurada qué componentes requerían atención prioritaria, como el rendimiento de la API o el comportamiento de la base de datos. Los *stakeholders* que participaron en la evaluación reconocieron este enfoque como una herramienta valiosa para la gestión de la calidad, al proporcionar una visión integrada del estado del servicio y facilitar la detección temprana de riesgos potenciales. Esto apoya directamente tanto la optimización como la mejora continua del servicio cloud.

Como proveedor del servicio, la organización es responsable de garantizar no solo el cumplimiento contractual, sino también la mejora continua de la experiencia del cliente. En este sentido, el caso práctico permitió *identificar áreas específicas de mejora* y *refinar la estrategia de monitorización*, evidenciando el valor del modelo para apoyar el análisis sistemático de la calidad del servicio cloud.

A nivel de **SaaS**, las fluctuaciones observadas en el tiempo de respuesta y el rendimiento pusieron de manifiesto la necesidad de revisar los mecanismos de gestión del tráfico (una

priorización de solicitudes más eficaz) y de mejorar la eficiencia de procesamiento. Las medidas recomendadas incluyen la optimización de algoritmos de respuesta, la introducción de mecanismos de priorización inteligentes y la implementación de estrategias de escalado proactivas para asignar recursos de manera eficaz durante períodos de alta demanda. Estos cambios mejorarían la programación y permitirían una mayor paralelización de procesos. Estos resultados evidencian áreas de mejora potenciales, especialmente en escenarios de alta demanda. Asimismo, en cuanto a los riesgos de seguridad, se identificó la conveniencia de establecer mecanismos de alarma automatizados para detectar anomalías, como picos de tráfico o consumo excesivo de recursos.

A nivel de **PaaS**, los resultados revelaron cuellos de botella en el flujo de datos que afectaban el acceso a la base de datos PostgreSQL, restringían la velocidad operativa y limitaban la escalabilidad, especialmente ante incrementos de carga derivados de la creciente demanda de los usuarios. Para abordar estos problemas, la organización deberá considerar la posibilidad de ajustar los límites de conexión, los búferes de transacciones, el particionamiento de tablas y la indexación de consultas para mejorar el rendimiento general de la base de datos.

De forma análoga, el análisis de los componentes en contenedores permitió identificar desbalances en el uso de recursos, particularmente en términos de tráfico de red y memoria, lo que pone de relieve la importancia de una configuración adecuada de capacidades y políticas de orquestación. En particular, el contenedor *Mongo srv* expuso un alto tráfico de red, probablemente debido a operaciones de uso intensivo de datos, como la ingesta y el almacenamiento a gran escala. Este desbalance podría provocar una saturación del rendimiento y caídas o bloqueos del sistema durante picos de uso, lo que amenaza la integridad de los servicios que dependen de este componente. Para mitigar estos riesgos, se recomienda implementar técnicas de balanceo de carga en los contenedores, priorizar el tráfico de red para garantizar una distribución uniforme de la carga y asignar recursos de forma dinámica. Herramientas de orquestación, como *Kubernetes*, pueden ayudar a automatizar estos procesos. Se espera que la implementación de estas medidas mejore tanto el rendimiento, especialmente en términos de uso de red y memoria, como la resiliencia del sistema al evitar la sobrecarga en contenedores críticos.

Aunque algunas métricas actualmente están *en observación* y carecen de umbrales formales, ofrecen información valiosa sobre los riesgos potenciales. Ignorar estas tendencias puede resultar en la pérdida de señales tempranas de la degradación o saturación de la capacidad del servicio cloud. La organización debe revisar periódicamente estas métricas para determinar si deben incorporarse al modelo de calidad con sus umbrales definidos. Por ejemplo, se podrían incorporar métricas de eficiencia de red o niveles de uso de memoria por contenedor.

A nivel de **IaaS**, las métricas monitorizadas indicaron una estabilidad general de la infraestructura (ya que todas las métricas monitorizadas cumplieron con los umbrales definidos), lo que sugiere una estabilidad general a nivel de infraestructura. Sin embargo, se detectaron patrones de uso intensivo de memoria y operaciones de disco que justifican un análisis más detallado. Por ejemplo, el uso elevado y persistente de memoria puede indicar la ejecución de procesos con un uso intensivo de memoria en las máquinas virtuales,

como aplicaciones específicas o microservicios. En cuyo caso, se recomienda un análisis más detallado del consumo de memoria por proceso para identificar las causas raíz. Además, los cuellos de botella de lectura y escritura de disco podrían afectar la capacidad de respuesta del servicio y la velocidad de procesamiento de datos. Para reducir el riesgo de limitaciones, la organización podría explorar estrategias como la compresión de datos, la optimización del espacio en disco y la provisión de un ancho de banda adecuado durante los picos de actividad.

En conjunto, estos hallazgos aportan valor tanto técnico como estratégico. Desde una perspectiva técnica, respaldan el uso eficiente de los recursos y ayudan a mantener una alta calidad del servicio, garantizando que las demandas futuras puedan gestionarse sin comprometer el rendimiento. Estratégicamente, la información permite a la organización anticipar las necesidades, mejorar la planificación operativa, fortalecer la competitividad e impulsar la innovación mediante la mejora continua de la calidad del servicio cloud.

6.5.2. Acciones de mejora derivadas

A partir de los hallazgos descritos en la subsección anterior, la aplicación del modelo y del proceso de evaluación permitió derivar un conjunto de acciones de mejora concretas orientadas a la gestión del servicio SAlert. Más allá de la identificación de incumplimientos y áreas críticas de calidad, la aplicación del modelo y del proceso de evaluación facilitó la transformación de los resultados de la monitorización en un conjunto de acciones de mejora concretas orientadas a la gestión del servicio SAlert.

El análisis de los resultados permitió la **revisión y el ajuste de los umbrales definidos en el SLA**. En particular, se identificaron métricas cuyo incumplimiento (p. ej., *uptime ratio* o alarmas monitorizadas) no reflejaba necesariamente una degradación del servicio, sino que correspondía a eventos transitorios o picos de carga. A partir de esta evidencia, se propuso la introducción de umbrales contextuales y basados en percentiles, así como zonas de decisión, lo que permite una evaluación más realista del cumplimiento y reduce el riesgo de falsos positivos. El beneficio esperado de esta acción es una mayor alineación entre la evaluación técnica y la percepción de calidad por parte de los *stakeholders*, así como una gestión más efectiva del riesgo de incumplimiento de SLA.

En particular, el mapeo operacional de métricas y los resultados obtenidos facilitaron la **priorización de acciones de optimización en componentes específicos**. Las desviaciones observadas en métricas de fiabilidad, seguridad y eficiencia de desempeño en las capas SaaS y PaaS evidenciaron cuellos de botella localizados en determinados *endpoints* y servicios internos. Debido a la trazabilidad proporcionada por el modelo de arquitectura, estas degradaciones pudieron asociarse a componentes concretos, lo que permitió tomar decisiones focalizadas, como la revisión de configuraciones de la base de datos. El beneficio esperado es una mejora del rendimiento del servicio con un menor coste de intervención, al evitar acciones generalizadas sobre componentes que no presentan degradaciones relevantes.

Los resultados de la evaluación del modelo respaldaron la **revisión de políticas de escalado automático y configuración de recursos**. La identificación de escenarios en

los que el servicio mantenía un alto cumplimiento a nivel de infraestructura (IaaS), pero presentaba degradaciones en capas superiores, puso de manifiesto la necesidad de refinar las políticas de autoescalado para que consideraran métricas funcionales y no únicamente métricas de utilización de recursos. Como acción de mejora, se planteó incorporar métricas de rendimiento y fiabilidad a las reglas de escalado, así como ajustar los parámetros de escalado reactivo ante picos de carga. El beneficio esperado es una respuesta más eficiente ante variaciones de demanda, reduciendo tanto la probabilidad de degradación del servicio como el sobreaprovisionamiento innecesario de recursos

Adicionalmente, la visibilidad proporcionada por el modelo permitió tomar **decisiones preventivas antes del despliegue de cambios de configuración o de nuevas versiones**. Al disponer de un marco estructurado que relaciona atributos de calidad con métricas observables, los responsables del servicio pudieron evaluar el impacto potencial de determinadas modificaciones sobre la calidad del servicio, lo que apoyó la toma de decisiones orientadas a mitigar riesgos antes de su materialización. El beneficio esperado de esta capacidad es la reducción de incidencias post-despliegue y una mayor estabilidad del servicio en el tiempo.

Desde una perspectiva de beneficio esperado, estas acciones contribuyen a mejorar la estabilidad del servicio, reducir el riesgo de incumplimientos de SLA y optimizar el uso de recursos de infraestructura, alineando la operación técnica con los objetivos de calidad definidos por la organización. Aunque el presente estudio no cuantifica de forma exhaustiva el impacto a largo plazo de dichas acciones, los resultados sugieren que el marco propuesto proporciona un soporte efectivo para la toma de decisiones informadas y sistemáticas, superando enfoques reactivos basados en métricas aisladas. De este modo, el modelo y el proceso de evaluación no solo permiten diagnosticar la calidad del servicio, sino también orientar de manera explícita su mejora continua en entornos cloud reales.

6.6. Respuesta a las preguntas de investigación

Después de realizar el estudio de caso en la organización, se ha llevado a cabo una discusión con el equipo que ha participado en el estudio para valorar su percepción en cuanto a los beneficios, limitaciones y lecciones aprendidas, que no ha permitido responder a las preguntas de investigación.

- **PI1**: **¿Se puede configurar un modelo de calidad personalizado a partir del modelo genérico para servicios cloud específicos?**

  Los resultados confirman que el modelo de calidad genérico permite configurar un modelo personalizado adaptado al contexto específico del servicio *SAlert*. La estructura jerárquica del modelo genérico fue utilizada con un enfoque descendente (*top-down*) para obtener una versión ajustada a los requisitos de calidad definidos por la organización.

  El equipo evaluador inició identificando los requisitos de calidad, a partir de los cuales el ingeniero de calidad seleccionó los elementos—subcaracterísticas, atributos,

métricas y operacionalizaciones—relevantes para construir la primera versión del modelo personalizado. Esta versión fue validada en un taller, donde se confirmó la cobertura completa de los requisitos con atributos medibles. El modelo de calidad final para *SAlert* fue aprobado por la organización y reconocido por su utilidad práctica y alineación con el entorno operativo.

La adaptabilidad del modelo se evidenció en su aplicación a las tres capas de los servicios cloud—SaaS, PaaS, IaaS—, evaluando a cada una mediante métricas específicas que reflejan sus particularidades. Además de mantener la trazabilidad entre requisitos, atributos y métricas durante todo el proceso, lo que refuerza la aplicabilidad y utilidad del modelo en entornos de la industria.

- **PI2**: **¿Las métricas seleccionadas, sus operacionalizaciones y sus resultados de medición proporcionan información significativa sobre el estado de la calidad del servicio?**

  La aplicación del modelo personalizado en el caso *SAlert* proporcionó información significativa y procesable sobre el estado actual del servicio. Aunque la plataforma Linode tiene capacidades limitadas de supervisión nativa, la integración con la herramienta New Relic permitió ampliar la recolección de datos, operacionalizar métricas y realizar análisis detallados.

  New Relic fue configurada para recolectar datos de métricas en bruto y calcular métricas derivadas como la tasa de éxito y error, que no podían obtenerse directamente desde la plataforma cloud. Esta capacidad de medición extendida permitió evaluar el cumplimiento de los requisitos de calidad e identificar áreas de mejora.

  Los resultados se han organizado por tipo de servicio y se han presentado en tablas y cuadros de mando visuales. Como se observa en la tabla 6.6, no se cumplieron los criterios de aceptación en 5 de las 15 métricas de SaaS y en 3 de las 10 métricas de la instancia de BD PostgreSQL (PaaS), mientras que todas las métricas de IaaS y los contenedores (PaaS) de la tabla 6.7 sí cumplieron. No obstante, se identificaron 8 métricas que, aunque no se definieron umbrales, ofrecieron información temprana sobre posibles degradaciones futuras.

  Los cuadros de mando facilitaron la identificación de la conformidad de las medidas con los criterios de calidad establecidos, permitiendo a los *stakeholders* una mejor comprensión del estado actual del servicio y la planificación de intervenciones específicas. Este enfoque no solo respalda la toma de decisiones oportuna, sino que también establece una línea base para futuras evaluaciones, promoviendo la mejora continua en la organización.

- **PI3**: **¿Puede adaptar el modelo de calidad personalizado a requisitos nuevos o cambiantes?**

  El modelo de calidad personalizado demostró una alta capacidad de adaptación mediante un proceso iterativo de refinamiento. Aunque inicialmente se construyó con

un enfoque descendente, a partir del modelo genérico y los requisitos de calidad definidos, su estructura permitió incorporar ajustes conforme a nuevos desafíos prácticos que surgieron durante la evaluación.

Un ejemplo es la evaluación de la instancia de PostgreSQL, donde se detectaron cuellos de botella no considerados en los requisitos iniciales. Esto motivó la incorporación de tres nuevos requisitos (R22-R24) centrados en la eficiencia operativa de la BD. Dado que el modelo genérico no incluía atributos ni métricas para cubrir estos aspectos, se adoptó un enfoque ascendente (*bottom-up*), integrando métricas directamente desde New Relic y definiendo un nuevo atributo (eficiencia de la BD).

También se hizo una revisión de las operacionalizaciones para la métrica *uptime ratio*, conservando la función de medición más representativa del comportamiento. Como resultado del análisis, el modelo fue enriquecido mediante la incorporación de tres requisitos, un atributo, siete métricas y la eliminación de una operacionalización.

Este proceso evidenció que el modelo no es estático, sino que puede evolucionar para incorporar nuevas realidades técnicas, configuraciones del servicio y objetivos organizacionales. Su capacidad de adaptación garantiza su relevancia a largo plazo en entornos cloud dinámicos. No obstante, con el propósito de mejorar su usabilidad y facilitar su aplicación sistemática, se determinó la necesidad de desarrollar una herramienta de software que respalde la implementación y gestión del modelo de calidad. Por ello, se ha implementado la primera versión del modelo genérico en *Power BI*, lo que facilita la navegación entre los elementos en ambas direcciones y facilita la trazabilidad. Como parte de los trabajos futuros, se prevé extender la herramienta para incluir la funcionalidad de personalización del modelo y la definición dinámica de los criterios de aceptación y umbrales, lo que permitirá adaptar la evaluación a contextos específicos y fortalecer la toma de decisiones.

6.7. Amenazas a la validez

Esta sección analiza las principales amenazas que podrían afectar la validez del estudio realizado y describe las estrategias de mitigación aplicadas o recomendadas para trabajos futuros.

6.7.1. Validez de constructo

Se refiere al grado en que el estudio mide con precisión lo que pretende medir. Una posible amenaza está en la cobertura incompleta de los atributos de calidad,dado que el modelo de calidad propuesto —tanto genérico como personalizado—no contempla todos los atributos relevantes para los servicios cloud. No obstante, el modelo genérico integra los atributos identificados en la literatura y ha sido diseñado con un enfoque extensible y evolutivo, permitiendo que desarrolladores y expertos del dominio lo amplíen de manera iterativa mediante la incorporación de nuevos atributos y métricas en respuesta a la evolución de los requisitos y tecnologías emergentes.

Respecto a la ausencia de validación empírica de varias métricas adoptadas, si bien varias de ellas fueron seleccionadas en función de su utilidad en estudios previos y en este estudio de caso, muchas aún requieren validación formal. Para mitigar esta limitación, este estudio de caso sirve como prueba de concepto que permite verificar la aplicabilidad de un subconjunto de métricas. En trabajos futuros, se plantea continuar con la operacionalización del modelo de calidad en diferentes contextos, con el objetivo de fortalecer la validación empírica de las métricas y su generalización a diferentes tipos de servicios cloud .

Respecto a la operacionalización de las métricas, en algunos casos, la literatura no ofrece definiciones detalladas ni funciones de medición suficientemente detalladas, lo que exige adaptaciones o la aplicación de métodos de medición indirectos apoyados en las herramientas de monitorización disponibles. Esta situación se abordó documentando claramente todas las suposiciones y empleando mecanismos de recopilación de datos consolidados, como agentes de monitorización y registros del sistema, a fin de garantizar la coherencia y trazabilidad de los resultados. Sin embargo, las limitaciones técnicas de las herramientas, como la falta de disponibilidad del agente de monitorización de contenedores *Mongo srv*, hicieron necesario recurrir a enfoques alternativos— como el análisis de registros—para compensar la falta de datos directos.

6.7.2. Validez interna

Se refiere a la credibilidad de las relaciones causa-efecto inferidas del estudio. En este caso, puede presentarse ambigüedad en la atribución de los resultados, ya que resulta difícil aislar los efectos del modelo de calidad de otros factores organizacionales o técnicos que pueden influir en el comportamiento del servicio cloud. Para mitigar este riesgo, se mantuvo un registro detallado de los cambios e intervenciones realizados durante el período de evaluación, lo que permitió la trazabilidad de las posibles fuentes de variación en los resultados.

Otra posible amenaza es el sesgo introducido por la doble participación de algunos miembros del equipo, quienes intervinieron tanto en el desarrollo como en la evaluación del modelo de calidad. Este riesgo se mitigó mediante la realización de talleres iterativos y sesiones de revisión conjunta, en los que participaron investigadores, profesionales y expertos del dominio. Este enfoque colaborativo permitió incorporar múltiples perspectivas y alcanzar una validación por consenso, fortaleciendo la objetividad del proceso.

Asimismo, aunque se instalaron agentes de monitorización en los distintos componentes de la arquitectura cloud para garantizar la consistencia en la recopilación de datos, el control limitado sobre las variables del entorno, el control limitado sobre variables externas—como las fluctuaciones en la carga de trabajo o las actualizaciones del sistema—pudo haber influido en los resultados observados. Con el fin de mejorar la fiabilidad, se recomienda que las evaluaciones futuras repliquen el estudio en diversas condiciones operativas y en diferentes momentos temporales, de modo que puedan identificar patrones estables y minimizar la influencia de factores contextuales.

6.7.3. Validez externa

Se refiere al grado en que los resultados pueden generalizarse a otros contextos, dominios o configuraciones tecnológicas. Siendo una posible amenaza la naturaleza de caso único del estudio, realizado en colaboración con una empresa que utiliza el servicio cloud SAlert, desarrollado sobre una arquitectura de complejidad moderada. Si bien esta configuración permitió una evaluación exhaustiva que abarcó las capas de IaaS, PaaS y SaaS, es posible que los resultados no sean totalmente generalizables a otros dominios, organizaciones o arquitecturas cloud.

Para mitigar esta limitación, el modelo de calidad propuesto se ha diseñado con un enfoque mixto flexible y extensible, que admite tanto estrategias descendentes como ascendentes. Este enfoque permite a investigadores y profesionales personalizar e instanciar el modelo según sus necesidades específicas. Sin embargo, se recomienda que en trabajos futuros se replicara el estudio en diferentes contextos organizacionales y con configuraciones heterogéneas de servicios cloud, con el fin de confirmar la aplicabilidad más amplia del modelo.

Adicionalmente, la evaluación se realizó en un entorno controlado, correspondiente a una porción del despliegue del servicio en producción, lo que permitió mantener un alto grado de control experimental y minimizar la interferencia de variables externas. No obstante, esta configuración podría no reflejar plenamente las condiciones reales de operación, en las que influyen factores como la variabilidad de la carga, los incidentes operativos y las restricciones de disponibilidad. En consecuencia, estudios futuros podrían considerar observar el comportamiento del modelo bajo condiciones más realistas y exigentes.

6.7.4. Validez de las conclusiones

Se refiere al grado de credibilidad y la solidez de las conclusiones extraídas de los datos. Una posible amenaza proviene de la ausencia de pruebas estadísticas formales, debido a que las conclusiones se basaron principalmente en un análisis descriptivo de los resultados de las métricas. Si bien el uso de herramientas de visualización y los cuadros de mando facilitaron la interpretación de los hallazgos por parte de los *stakeholders*, en evaluaciones futuras se deberán incorporar medidas estadísticas básicas, como tendencias, varianza o correlación, para reforzar las conclusiones. No obstante, este es un estudio exploratorio cuyo propósito principal fue demostrar la factibilidad y aplicabilidad del modelo de calidad propuesto. Estos resultados aportan la base para el desarrollo de estudios que profundicen la validación empírica y estadística del modelo .

6.8. Implicaciones

Los hallazgos de este estudio tienen diversas implicaciones tanto para la comunidad de ingeniería de software como para los profesionales de la industria. Al implementar un modelo de calidad para servicios cloud en un entorno real, esta investigación contribuye

a reducir la brecha entre los modelos de calidad teóricos y su aplicación práctica, proporcionando una base empírica para su evaluación, refinamiento y extensión.

6.8.1. Implicaciones para investigadores

Las principales implicaciones para los investigadores incluyen:

- **Validación de modelos de calidad en contextos industriales:** Este estudio aporta evidencia empírica sobre la aplicabilidad y adaptabilidad de los modelos de calidad jerárquicos en entornos de computación en la nube. La exitosa personalización y el refinamiento iterativo del modelo de calidad en el caso práctico de *SAlert* demuestran su valor como herramienta de evaluación flexible y capaz de evolucionar con los requisitos cambiantes del dominio. El modelo de calidad propuesto reduce la brecha entre las definiciones abstractas de calidad (características) y las métricas concretas y operacionalizables, fortaleciendo su relevancia práctica.
- **Contribución a la evolución del modelo de calidad**: El proceso de adaptación del modelo reveló la necesidad de incorporar nuevos atributos y métricas no presentes originalmente en la literatura, lo que evidencia que los modelos de calidad existentes pueden carecer de profundidad operativa para tecnologías o plataformas específicas (p. ej., BD PostgreSQL u orquestación de contenedores). Este hallazgo resalta la oportunidad de aplicar y extender el modelo propuesto en otros contextos, con el fin de identificar e incorporar nuevas métricas específicas del dominio y enriquecer la comprensión de la calidad.
- **Plan metodológico para futuros estudios**: El estudio de caso ofrece una metodología reproducible que integra el modelado de calidad, la selección de métricas, el análisis de datos y la validación con *stakeholders*. Los trabajos futuros pueden utilizar este enfoque estructurado para evaluar otros servicios cloud o modelos de prestación de servicios, mejorando las comparaciones entre casos y contribuyendo a la validez externa del modelo de calidad propuesto.
- **Validación de métricas:** Si bien la mayoría de las métricas del modelo de calidad propuesto aún no cuentan con validación empírica [2], este estudio ofrece un primer paso para su validación. La aplicación de un conjunto seleccionado de métricas en un entorno industrial demuestra su utilidad y pertinencia para evaluar la calidad de los servicios cloud. Este resultado actúa como una prueba de concepto y sienta las bases para futuras investigaciones orientadas a validar, refinar y ampliar el conjunto de métricas en diferentes contextos y tipos de servicio.
- **Evaluación de la calidad basada en datos:** El uso de herramientas de monitorización, como New Relic, para respaldar la operacionalización y el análisis de métricas sugiere una creciente intersección entre la investigación de la calidad del software y la ingeniería de observabilidad. Esta integración abre nuevas líneas de investigación como la automatización de la evaluación de calidad y la incorporación de técnicas de análisis continuo integradas con los modelos de calidad.

6.8.2. Implicaciones para profesionales de la industria

A partir de la definición del modelo de calidad y su aplicación en un contexto industrial, se identifican algunas implicaciones prácticas relevantes que se detallan a continuación:

- **Mejora de la toma de decisiones basada en la evidencia**: El modelo de calidad personalizado fortalece la toma de decisiones informadas, al establecer una relación entre las métricas observables y los requisitos de calidad definidos. Los cuadros de mando y los mecanismos de evaluación basados en umbrales facilitan la interpretación de resultados y la priorización de acciones por parte de los *stakeholders*, especialmente en entornos dinámicos y de alta variabilidad operativa.
- **Gestión proactiva de la calidad**: La detección temprana de métricas no conformes en las capas de SaaS y PaaS permitió implementar acciones correctivas oportunas, tales como el ajuste de configuraciones de bases de datos o la optimización de las cargas de trabajo en contenedores. Estos resultados demuestran el valor práctico de la monitorización proactiva basada en modelos de calidad, que posibilita anticipar deficiencias o degradaciones del servicio y prevenir interrupciones mayores en la operación del servicio.
- **Aseguramiento de calidad escalable y adaptativo**: La flexibilidad del modelo de calidad facilita su escalabilidad y adaptación continua frente a cambios en la infraestructura o la arquitectura del servicio. Esto resulta especialmente relevante para entornos nativos de la nube, caracterizados por la evolución constante de microservicios, contenedores y recursos distribuidos. En consecuencia, las organizaciones pueden mantener sus niveles de calidad sin comprometer la agilidad ni la capacidad de crecimiento del servicio.
- **Planificación estratégica y optimización del servicio**: El modelo contribuye a alinear la monitorización del rendimiento con los objetivos estratégicos de negocio, estableciendo vínculos claros entre las métricas, los SLA y la satisfacción del cliente. Esta alineación promueve la mejora continua y la innovación sostenida del servicio cloud a largo plazo, integrando la gestión de la calidad en la toma de decisiones organizacionales.
- **Marco de adopción para la industria:** Aunque el modelo de calidad propuesto es genérico e independiente de la plataforma, su estructura extensible permite adaptarlo a diferentes contextos y necesidades. El estudio de caso ofrece un ejemplo concreto de cómo las organizaciones pueden adoptar y personalizar el modelo según sus capacidades y necesidades específicas, aplicando iterativamente enfoques descendentes y ascendentes. De este modo, incluso las organizaciones con recursos de monitorización limitados pueden implementar evaluaciones de calidad integrales utilizando herramientas ya disponibles.
- **Personalización y amplia aplicabilidad**: La flexibilidad del modelo lo plantea como una herramienta de evaluación versátil para distintas organizaciones y contex-

tos operativos. Por ejemplo, las organizaciones que externalizan o contratan servicios cloud pueden emplear el modelo para verificar si la calidad del servicio prestado se ajusta a los términos del SLA. Las organizaciones que dependen de infraestructuras en la nube de terceros pueden usarlo para garantizar que la calidad del servicio contratado no comprometa el rendimiento de sus propios productos o aplicaciones. Las empresas de desarrollo de software pueden utilizarlo como marco de validación independiente de calidad, e incluso como base para procesos de certificación formal de sus servicios cloud.

6.9. Conclusiones

La viabilidad operativa del modelo se demostró mediante un estudio de caso con SAlert, un servicio cloud real que opera en las capas SaaS, PaaS e IaaS. Los resultados del proceso de evaluación evidenciaron la capacidad del modelo para traducir los requisitos de calidad abstractos en métricas concretas y observables, apoyadas por mecanismos de monitorización en tiempo real. Esta operacionalización proporcionó una visión integral de la calidad del servicio, permitió identificar áreas con incumplimiento y ayudó a los *stakeholders* a priorizar las acciones de mejora. Asimismo, la combinación del modelo con herramientas de visualización, como los cuadros de mando, facilitó la trazabilidad y el diagnóstico detallado de los problemas de calidad.

Un hallazgo clave fue la capacidad del modelo para evolucionar y personalizarse conforme a los requisitos de la organización y a los cambios del entorno operativo. Por ejemplo, ante las deficiencias identificadas en una instancia de PostgreSQL, el equipo evaluador aplicó un enfoque ascendente, incorporando nuevas métricas específicas de la plataforma y refinando los atributos de calidad. Este proceso iterativo pone de manifiesto la adaptabilidad y extensibilidad del modelo, así como su potencial para impulsar la mejora continua de la calidad del servicio.

El estudio también proporciona una prueba de concepto empírica para un subconjunto representativo de métricas seleccionadas. Aunque muchas métricas reportadas en la literatura carecen de validación empírica, su aplicación en el estudio de caso confirmó la utilidad del modelo propuesto para generar información significativa y accionable. Este resultado contribuye a los esfuerzos de investigación orientados a la validación empírica de métricas de calidad y fortalece la base conceptual y operativa del modelo.

Más allá de la evaluación descriptiva de la calidad, los resultados obtenidos en este capítulo ponen de relieve la importancia de disponer de métricas de QoS bien definidas, monitorizadas de forma sistemática y contextualizadas dentro de un modelo de calidad estructurado. Estas métricas no solo permiten evaluar el estado actual del servicio, sino que también constituyen una base sólida de datos históricos y conocimiento operativo que puede ser explotada con fines predictivos. En este sentido, el conjunto de métricas seleccionadas, las decisiones sobre granularidad temporal y las lecciones extraídas sobre variabilidad y comportamiento bajo carga alimentan directamente los requisitos y datos de entrada del proceso de pronóstico de QoS que se desarrolla en el Capítulo 7.

Desde una perspectiva práctica, el modelo ofrece un instrumento de evaluación de

calidad para organizaciones que externalizan o dependen de servicios cloud de terceros. Permite a los consumidores verificar si la calidad entregada se ajusta a los compromisos de los SLA y garantizar que el rendimiento de los servicios contratados no comprometa la calidad interna del software. Asimismo, las empresas proveedoras y fábricas de desarrollo de software pueden adoptar el modelo para promover la transparencia y confianza ante sus clientes, e incluso como base para procesos de certificación o auditoría de calidad.

Entre los desafíos observados destacan las limitaciones de ciertas herramientas de monitorización y las definiciones incompletas de algunas métricas en la literatura. No obstante, la integración de herramientas de monitorización nativas de la nube, el uso de análisis temático y los talleres colaborativos contribuyeron a mitigar estas limitaciones, demostrando la resiliencia y flexibilidad del modelo para incorporar nuevos requisitos y métricas de calidad a medida que evolucionan las tecnologías y las expectativas de los *stakeholders*.

El modelo de calidad completo, junto con los resultados del estudio de caso, se encuentra disponible en un repositorio público, con el objetivo de fomentar la replicación, comparación y extensión del trabajo por parte de la comunidad académica y la industria.

Finalmente, el trabajo futuro incluye la realización de nuevos estudios de caso con distintos tipos de servicios, proveedores y arquitecturas cloud, con el fin de evaluar la generalización y robustez del modelo. Asimismo, se plantea la integración sistemática de técnicas de aprendizaje automático para fortalecer la monitorización predictiva de la calidad, permitiendo detectar patrones, anticipar degradaciones y apoyar la toma de decisiones proactiva, línea que se desarrolla en detalle en los capítulos siguientes.

Adicionalmente, se propone ampliar el modelo o definir otro modelo de calidad en uso para incorporar métricas orientadas a la experiencia del usuario, tales como disponibilidad efectiva del servicio, error rate percibido por el usuario o tasa de abandono relacionada con la latencia o el abandono. Esta extensión permitirá una evaluación integral de la calidad, abarcando tanto las dimensiones técnicas como la experiencia de los usuarios, y garantizando que las evaluaciones de calidad se alineen con las expectativas de todos los *stakeholders* involucrados en el entorno cloud.

Capítulo 7

Proceso de Modelado ML para el Pronóstico de QoS

Este capítulo describe el proceso metodológico de modelado basado en aprendizaje automático (ML) propuesto para el pronóstico de la calidad del servicio (QoS) en servicios cloud. El propósito principal es definir, aplicar y validar un proceso estructurado que guíe la construcción de modelos predictivos de QoS, estableciendo las fases, actividades, técnicas y artefactos necesarios para su implementación sistemática.

La Sección 7.1 introduce el propósito y la importancia del proceso de modelado de ML. La Sección 7.2 presenta el proceso de modelado de ML, incluyendo una descripción detallada de los subprocesos y actividades. La Sección 7.3 presenta el protocolo propuesto para validación empírica del proceso. Finalmente, la Sección 7.4 presenta las conclusiones y contribuciones de este capítulo.

7.1. Introducción

La innovación en servicios cloud avanza de forma acelerada, al igual que la complejidad de los entornos en los que operan. Las organizaciones dependen cada vez más de estos servicios para sustentar sus operaciones de TI, lo que plantea nuevos desafíos en la evaluación de la calidad del servicio debido a la asignación dinámica de recursos, la heterogeneidad de las configuraciones de múltiples proveedores y la diversidad de expectativas de los clientes.

Los SLA establecen los compromisos formales entre clientes y proveedores, definiendo los requisitos de calidad, las obligaciones funcionales y las penalizaciones asociadas al no cumplimiento [154]. Estos contratos incluyen la identificación de los participantes (cliente, proveedor), la especificación de las obligaciones funcionales y de calidad, el precio y las penalizaciones [155, 156]. Sin embargo, en entornos altamente dinámicos, la monitorización continua se ha vuelto crucial para verificar el cumplimiento de los SLA y respaldar la toma de decisiones informadas sobre la gestión y optimización de los servicios.

La gestión de los SLA representa una tarea cada vez más compleja, lo que impulsa la necesidad de mecanismos de monitorización automatizados y fiables que permitan medir y analizar las métricas asociadas a la QoS [157]. Los modelos de calidad, como el propuesto en el capítulo 5, proporcionan un marco estructurado para especificar los requisitos de QoS y configurar los sistemas de monitorización y evaluación [158]. No obstante, para su aplicación efectiva en entornos reales, estos modelos deben complementarse con un enfoque proactivo de la calidad. En este contexto, el pronóstico de QoS mediante técnicas de aprendizaje automático (*machine learning*, ML) se presenta como una estrategia fundamental para anticipar degradaciones del servicio, prevenir violaciones de SLA y optimizar la asignación de recursos.

El ML posibilita la identificación de patrones y tendencias complejas en grandes volúmenes de datos, así como la generación de predicciones precisas a partir de series temporales de métricas de desempeño. En particular, el aprendizaje profundo (véase 2.3.5) como las redes neuronales recurrentes (RNN)—como LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*)—han mostrado resultados destacados en el modelado de dependencias a largo plazo, mientras que su variante bidireccional (BI-GRU) combina el análisis de información pasada y futura, mejorando la precisión del pronóstico en secuencias temporales [16, 68].

A pesar de su potencial, la aplicación de estas arquitecturas al pronóstico de la calidad en servicios cloud es aún limitada [17, 18]. Los estudios existentes, por lo general, emplean conjuntos de datos reducidos, consideran un número limitado de métricas de QoS y adoptan procesos experimentales poco reproducibles, lo que dificulta la comparación entre enfoques. Además, se observa la ausencia de una metodología sistemática que guíe de forma integral la construcción, validación y evaluación de modelos de ML aplicados a este dominio.

Con el propósito de abordar estas limitaciones, este capítulo propone un proceso metodológico de modelado de ML reproducible, diseñado para operacionalizar el modelo de calidad propuesto en el Capítulo 5. El proceso establece un marco sistemático que abarca la construcción, el entrenamiento, la evaluación y la validación de modelos de ML orien-

tados a la predicción de la calidad de servicios cloud. Integra principios de ingeniería del software, analítica de datos y gestión de la calidad del servicio y se alinea con las actividades de validación académica del modelo de transferencia tecnológica [25]. A continuación, se describen de manera detallada las fases que componen este proceso de modelado, sus actividades, artefactos y técnicas asociadas, así como los instrumentos utilizados para su validación empírica.

7.2. Descripción del proceso de modelado de ML

Para apoyar a las organizaciones de desarrollo de software en la construcción y validación de modelos ML, se propone un proceso de modelado específicamente adaptado al problema de pronosticar la calidad (QoS) de servicios cloud.

A diferencia de los sistemas tradicionales, los sistemas basados en aprendizaje automático requieren un procesamiento y análisis de datos personalizados, lo que dificulta la reutilización y modularización de componentes, incluso cuando se emplean herramientas especializadas como los cuadernos de desarrollo (*notebooks*). Factores como las normativas de privacidad, la compatibilidad entre plataformas y la naturaleza especializada de los análisis limitan con frecuencia la interoperabilidad entre herramientas y lenguajes.

El desarrollo de un modelo ML eficaz depende del acceso a los datos, lo que implica implementar estrategias rigurosas de anonimización para cumplir con la normativa de protección de datos [159, 160]. Asimismo, la obtención de un rendimiento óptimo exige un proceso de experimentación intenso, que incluye la comparación sistemática de configuraciones, arquitecturas y técnicas de optimización.

Aunque el flujo de trabajo de ML suele representarse de forma lineal, en la práctica comparte similitudes con los enfoques ágiles de desarrollo de software [161], caracterizados por ciclos iterativos de retroalimentación. Estos ciclos abarcan la selección del modelo, el ajuste de hiperparámetros—tanto del modelo como del proceso de entrenamiento—y el refinamiento progresivo de los resultados. Este enfoque iterativo estructura todo el flujo de trabajo, desde la definición de los requisitos del modelo hasta su implementación y la monitorización operativa.

El pronóstico de la calidad de servicio, sin embargo, presenta desafíos específicos del dominio, entre los que destacan:

- La **variabilidad de las métricas**, dado que los patrones de medición pueden fluctuar debido a demandas de recursos inesperadas o a la aparición de anomalías;
- La **heterogeneidad del servicio**, ya que las diferencias entre infraestructuras, plataformas y proveedores dificultan la integración de datos y la comparación de resultados; y
- La **multitenencia**, que incrementa la complejidad por las interacciones entre distintos inquilinos que comparten recursos en entornos cloud.

Para abordar estos desafíos, se adaptó el flujo de trabajo propuesto por Amershi et al. [3], el cual sintetiza las mejores prácticas adoptadas por equipos de desarrollo de ML

de Microsoft. El proceso resultante—representado en la Figura 7.1—está diseñado para la predicción de la calidad de los servicios cloud y se organiza en subprocesos y actividades centradas en el procesamiento de datos y los ciclos de retroalimentación.

El flujo de trabajo se estructura en cuatro subprocesos clave: requisitos del modelo, preparación de datos, construcción del modelo y uso del modelo. En la figura 7.1, las actividades en blanco corresponden al proceso original [3], las actividades en color azul representan las adaptaciones introducidas en el presente trabajo, y las actividades en color verde indican las entradas y salidas específicas del dominio de la calidad de servicio en entornos cloud.

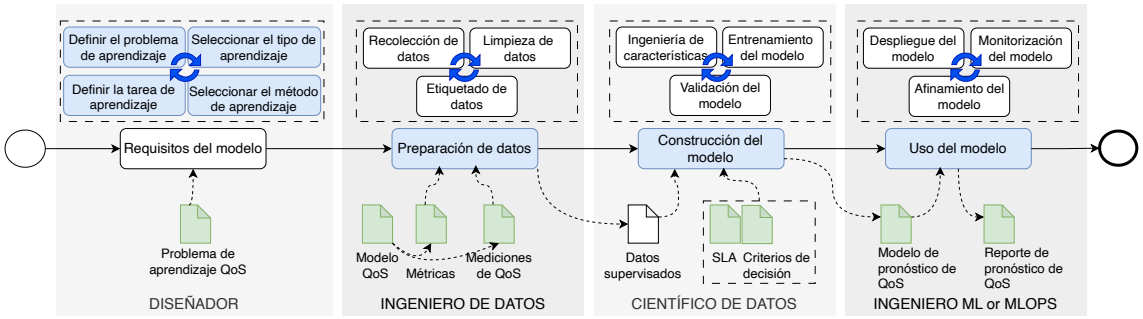


Figura 7.1: Proceso de modelado de ML, adaptado de Amershi et al.[3]

7.2.1. Requisitos del modelo

El objetivo principal de este subproceso, ejecutado por los *diseñadores*, es determinar la viabilidad técnica y conceptual del modelo de ML. En este subproceso se establecen los fundamentos del modelado, que incluyen la definición del problema de aprendizaje, la selección del tipo de aprendizaje, la definición de la tarea de aprendizaje y la selección del método de aprendizaje más apropiado [162].

El **problema de aprendizaje** abordado en este estudio es el pronóstico de la calidad (QoS) para servicios cloud. La QoS constituye un concepto multidimensional definido por un conjunto de características, atributos de calidad y métricas asociadas. Modelar su comportamiento permite identificar tendencias, anticipar posibles degradaciones y apoyar la gestión proactiva de los servicios. En términos prácticos, la predicción de QoS contribuye a reducir las violaciones de SLA mediante la identificación temprana de desviaciones respecto a los umbrales acordados, además de servir como base para implementar acciones preventivas o correctivas que optimicen el desempeño del servicio.

Para abordar este problema, el **tipo de aprendizaje** seleccionado es el *aprendizaje supervisado*, dado que el proceso se apoya en un conjunto de datos etiquetados obtenido a partir de la monitorización continua de los atributos de QoS. En este contexto, las etiquetas se generan estableciendo una correspondencia temporal: los datos monitorizados en un periodo determinado se utilizan como entrada, mientras que los valores del periodo siguiente constituyen la salida objetivo.

La **tarea de aprendizaje** se define como una *regresión* (la alternativa sería la clasificación), cuyo propósito es predecir valores numéricos de las métricas de QoS observadas. Las salidas del modelo—predicciones continuas de variables como tiempo de respuesta o disponibilidad—pueden compararse con los umbrales definidos en los SLA, posibilitando la detección temprana de posibles desviaciones y una toma de decisiones fundamentada.

La selección del **método de aprendizaje** consideró varios factores, entre ellos, el tamaño y la naturaleza del conjunto de datos, la presencia de relaciones lineales y no lineales entre variables, la existencia de dependencias temporales, los recursos computacionales disponibles y el equilibrio entre la complejidad del modelo y su interpretabilidad. En este estudio, se emplean dos estrategias de selección de modelos: intraenfoque y enfoque cruzado.

- La estrategia *intraenfoque* busca alcanzar la máxima precisión posible dentro de un mismo enfoque, identificando el modelo más adecuado entre diversas configuraciones o arquitecturas de modelos pertenecientes al mismo paradigma [18]. Para este problema, adoptamos el enfoque de aprendizaje profundo (DL), específicamente redes neuronales recurrentes, debido a su capacidad para capturar dependencias y patrones de largo plazo en secuencias temporales de métricas de QoS [16].

- La estrategia *enfoque cruzado* explora enfoques heterogéneos, integrando métodos estadísticos y de aprendizaje automático como alternativas de comparación. Este enfoque favorece la evaluación transversal de múltiples enfoques y aumenta la probabilidad de identificar el modelo más robusto y generalizable para el problema en cuestión [18].

Este subproceso sienta las bases del flujo de trabajo al establecer los criterios metodológicos y los parámetros iniciales que guiarán las etapas posteriores de preparación de datos, construcción del modelo y uso.

7.2.2. Preparación de datos

El objetivo de este subproceso, realizado por los *ingenieros de datos*, es preparar y estructurar los datos necesarios para la construcción del modelo de aprendizaje automático. Las actividades clave incluyen: la recolección, la limpieza y el etiquetado de datos, asegurando que el conjunto de datos cumpla con las condiciones requeridas.

La **recolección de datos** consiste en obtener el conjunto de datos que alimentará el proceso de modelado. En los casos donde no existan datos históricos disponibles, es necesario identificar las fuentes de datos relevantes y establecer mecanismos para la recolección de datos sin procesar.

El enfoque propuesto se fundamenta en un modelo de calidad genérico para servicios cloud, el cual incluye características, atributos y métricas de calidad independientes de la plataforma. Este modelo genérico se instancia para crear un modelo de calidad personalizado, ajustado a un servicio y plataforma específicos, que constituye la entrada para la configuración del proceso de monitorización.

El conjunto de datos resultante está compuesto por series temporales de datos recolectadas a intervalos de tiempo definidos. Se recomienda el uso de herramientas de terceros como mecanismos de monitorización, configuradas de acuerdo con el modelo de calidad personalizado. Estas herramientas recopilan mediciones directas para las métricas base y calculan métricas derivadas, que, aunque dependen de la plataforma, mantienen correspondencia con las métricas definidas en el modelo genérico.

El mecanismo de monitorización seleccionado debe permitir el mapeo entre métricas independientes de la plataforma y métricas específicas de la plataforma, así como la extracción de datos en tiempo de ejecución, tal como plantean Cedillo et al. [100]. Debido a la heterogeneidad inherente a los servicios cloud, puede ser necesario el empleo de múltiples mecanismos de monitorización para recopilar y almacenar datos de forma simultánea en distintos medios o repositorios habilitados.

La **limpieza de datos** tiene como propósito detectar y corregir anomalías, además de preprocesar el conjunto de datos con el fin de mejorar su calidad y consistencia. Las anomalías, según Chandola et al. [163], se definen como patrones que se desvían significativamente del comportamiento esperado e incluyen registros inexactos, ruido, sesgos o errores de medición. Desde un punto de vista estadístico, una anomalía se produce cuando una observación se aparta de la distribución esperada de los datos [163].

El preprocesamiento incluye actividades como la eliminación de duplicados, el tratamiento de valores atípicos y la resolución de segmentos con datos faltantes, con el fin de garantizar la coherencia y continuidad de las series temporales. Estas operaciones son esenciales para reducir la varianza, evitar sesgos durante el entrenamiento del modelo y preservar las propiedades estadísticas del conjunto de datos.

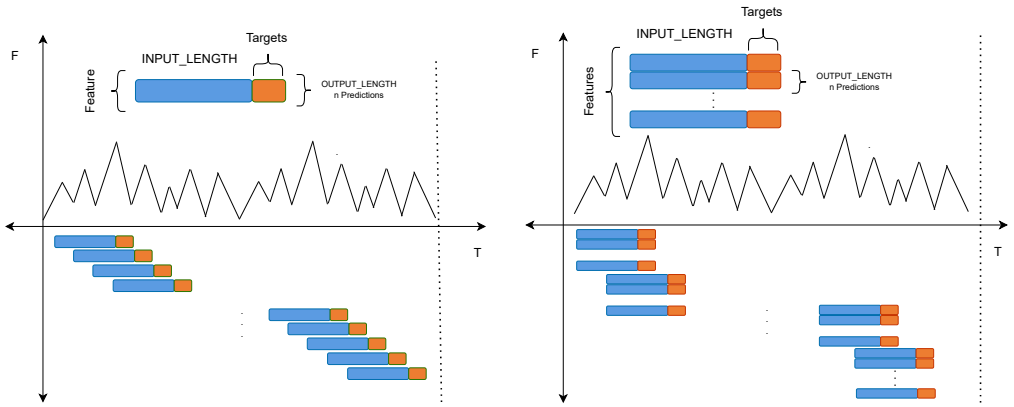
El **etiquetado de datos** constituye una etapa crítica dentro del aprendizaje supervisado, ya que define la relación entre los datos de entrada y las salidas objetivo. Esta actividad puede involucrar tanto a expertos en el dominio como a los miembros del equipo, quienes determinan las reglas para componer los pares entrada-salida en función del contexto del servicio.

El conjunto de datos se transforma en un formato supervisado con el propósito de identificar patrones y tendencias (p. ej., estacionalidad o ciclos de carga) y establecer el intervalo temporal de predicción. La conversión se realiza mediante la generación de vectores o matrices de entrada y salida, donde las entradas corresponden a observaciones históricas y las salidas representan los valores futuros a pronosticar.

Los modelos de pronóstico pueden configurarse según dos enfoques principales: *univariante* o *multivariante*. En las predicciones *univariantes*, cada variable representa una métrica específica de QoS. El vector de entrada x contiene las observaciones históricas (mediciones) de dicha métrica, mientras que el vector de salida y almacena los valores de predicción. En este caso, el proceso de entrenamiento se repite z veces, donde z es el número de métricas de QoS, generando así z modelos independientes. En las predicciones *multivariantes*, múltiples métricas de QoS se consideran de forma simultánea. En este escenario, la matriz de entrada x está compuesta por n variables y sus observaciones históricas, y la matriz de salida y contiene los pronósticos correspondientes para estas n variables (véase la Figura 7.2). Este enfoque permite capturar interdependencias entre

métricas, mejorando la capacidad predictiva del modelo.

Asimismo, es necesario definir el horizonte de predicción. En una predicción de un solo paso (*uni-step*), se predice el siguiente instante de tiempo $(t+1)$ basándose en n valores históricos [63, 164]. Por el contrario, la predicción de múltiples pasos (*multi-step*) genera predicciones para una secuencia de varios instantes de tiempo $(t+1, ..., t+n)$, donde n representa el horizonte de predicción deseado. Este último enfoque, aunque más exigente computacionalmente, permite anticipar patrones de comportamiento a mediano plazo y es especialmente útil en escenarios de gestión de recursos.



(a) Univariante (b) Multivariante

Figura 7.2: Etiquetado de series temporales para predicciones univariantes y multivariantes.

7.2.3. Construcción del modelo

Este subproceso constituye el núcleo del desarrollo de modelos de ML y es ejecutado por los *científicos de datos*. Incluye tres actividades principales: la ingeniería de características, el entrenamiento del modelo y la validación del modelo.

La **ingeniería de características** consiste en la extracción, transformación y selección de las características relevantes del conjunto de datos para su integración en el modelo de ML. Entre las transformaciones más frecuentes se encuentran la normalización y la estandarización de las variables, así como la inclusión de marcas de tiempo. En algunos enfoques, esta actividad puede considerarse parte del subproceso de preparación de datos, dependiendo del método de aprendizaje elegido.

El uso de técnicas de normalización o estandarización resulta esencial, ya que la heterogeneidad en los rangos de valores puede afectar significativamente las actualizaciones de gradientes y, en consecuencia, la eficiencia del aprendizaje [165, p. 133]. Escalar las características a un rango numérico consistente mejora tanto la estabilidad del proceso de entrenamiento como la capacidad de generalización del modelo.

El **entrenamiento del modelo** abarca la definición de la arquitectura del modelo, la configuración y ajuste de sus parámetros, así como la selección de un método de evaluación

del rendimiento adecuado para las características del conjunto de datos. Dado que este estudio plantea el uso de métodos de DL, la construcción y el ajuste de la arquitectura del modelo se alinean con el marco de clasificación de medidas de rendimiento propuesto por Vegas y Elbaum [4], que guía la definición y comparación de modelos de DL y se detalla en la Sección 7.2.3.1.

Para evaluar el rendimiento, se plantea el uso de validación cruzada (*cross-validation* - CV). Sin embargo, la validación cruzada tradicional resulta inapropiada para series temporales, debido a que el muestreo aleatorio rompe la dependencia temporal entre las observaciones, comprometiendo la validez de las estimaciones y la precisión de las predicciones [61, 166, 167]. En su lugar, se propone un *enfoque prequential*, una variante del método fuera de la muestra (*out-of-sample*) que preserva la secuencia temporal, dividiendo el conjunto de datos en múltiples subconjuntos de entrenamiento y prueba (particiones) pero manteniendo el orden cronológico [61].

En definitiva, se define un período inicial para el entrenamiento y un período posterior para la prueba. Cada observación se emplea primero para la validación y posteriormente se incorpora al conjunto de entrenamiento, siguiendo el principio de aprendizaje incremental descrito por Tashman et al. [167]. Para evitar estimaciones sesgadas por la selección aleatoria del conjunto de prueba, los errores se promedian a lo largo de todas las particiones, en un método conocido como encadenamiento hacia adelante (*forward chaining*) [166]. La diferencia clave del enfoque *prequential* es que las particiones de entrenamiento y prueba están predeterminadas por el número de bloques o repeticiones, en lugar de seleccionarse de forma aleatoria [61].

La **validación del modelo** tiene como objetivo evaluar tanto la precisión del modelo o entrenamiento (componente de modelado) como la precisión de la predicción (componente de prueba) [168, 60]. La *precisión del entrenamiento* se obtiene aplicando el modelo entrenado sobre el conjunto de validación, mientras que la *precisión de la predicción* mide la capacidad del modelo para predecir datos no observados previamente utilizando el conjunto de prueba, lo que constituye un indicador directo de su eficacia y capacidad de generalización [60].

Para la evaluación de modelos de pronóstico de series temporales, se proponen tres métricas principales de precisión:

- **Error cuadrático medio** (*Root Mean Squared Error* - RMSE), que penaliza los errores grandes al elevarlos al cuadrado antes de promediarlos (veáse eq.7.1);
- **Error absoluto medio** (*Mean Absolute Error* - MAE), que mide la desviación media en valor absoluto, ofreciendo una interpretación directa del error promedio (veáse eq.7.2); y
- **Error porcentual absoluto medio** (*Mean Absolute Percentage Error* - MAPE), que expresa los errores en términos relativos o porcentuales, facilitando la comparación entre métricas de diferentes servicios [17, 18, 102] (veáse eq.7.3).

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i^*)^2} \tag{7.1}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - y_i^*| \tag{7.2}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - y_i^*|}{y_i} \tag{7.3}$$

Donde N es el tamaño del conjunto de datos que representa el número de observaciones en el período de predicción, y_i es el valor observado y y_i^* denota el valor pronosticado de la métrica de calidad de servicio (QoS) en el instante i.

Los valores obtenidos mediante estas métricas permiten realizar comparaciones consistentes entre diferentes configuraciones o arquitecturas, dado que valores más bajos reflejan un mejor rendimiento predictivo del modelo. El propósito es determinar qué modelo o configuración ofrece el mejor desempeño predictivo y capacidad de generalización, garantizando al mismo tiempo el cumplimiento de los requisitos establecidos de aprendizaje y de las restricciones operativas. Para ello, se lleva a cabo una comparación sistemática entre diferentes modelos, arquitecturas o configuraciones de hiperparámetros.

La evaluación cuantitativa se realiza mediante las métricas de rendimiento calculadas sobre los conjuntos de validación y prueba definidos previamente durante el entrenamiento. Dado que el objetivo es el pronóstico de QoS en servicios cloud, el rendimiento del modelo se mide tanto en términos de precisión como de eficiencia computacional. Las métricas de precisión (RMSE, MAE, MAPE) permiten analizar la magnitud del error y la sensibilidad ante desviaciones en diferentes escalas, ofreciendo una visión integral del comportamiento del modelo [60]. Por su parte, la eficiencia computacional considera aspectos como los recursos necesarios (CPU, GPU, memoria) y el tiempo de inferencia, parámetros especialmente críticos en contextos cloud.

El proceso comparativo para la selección del modelo se fundamenta en las estrategias intraenfoque y enfoque cruzado, introducidas en la Sección 7.2.1.

- En la estrategia **intraenfoque**, se analizan y comparan diferentes configuraciones del mismo tipo de modelo (p. ej., variantes de una arquitectura LSTM con diferentes hiperparámetros).
- En la estrategia de **enfoque cruzado**, se contrastan modelos de naturaleza heterogénea -como los modelos estadísticos (p. ej., ARIMA) frente a los modelos de DL (p. ej., LSTM, GRU o CNN)-con el propósito de identificar el enfoque más adecuado según las características del problema y la disponibilidad de datos [18].

Para garantizar la objetividad en la selección del modelo dentro del estudio de caso, se emplea un protocolo de evaluación uniforme, en el que todas las variantes sean evaluadas bajo condiciones controladas y comparables. Esto implica utilizar los mismos segmentos temporales del conjunto de datos—conjuntos de entrenamiento y prueba——, mismo horizonte de predicción y mismas métricas de desempeño para cada alternativa de

modelo. Este principio, de aplicación sistemática de criterios homogéneos constituye un mecanismo fundamental en la comparación empírica de modelos, para reducir el sesgo en la interpretación de resultados y permite extraer conclusiones fiables [4].

El proceso de selección concluye con la identificación del modelo óptimo, es decir, aquel que alcanza el mejor equilibrio entre rendimiento predictivo, eficiencia y capacidad de generalización. En este punto, puede aplicarse una optimización de hiperparámetros mediante técnicas como búsqueda en cuadrícula (*grid search*), búsqueda aleatoria (*random search*) o algoritmos bayesianos [169].

Finalmente, se documentan las decisiones de diseño, los valores óptimos de los hiperparámetros y las métricas obtenidas. Este registro contribuye a la trazabilidad del proceso de modelado, favorece la utilidad del estudio de caso, y facilita futuras tareas de reentrenamiento, transferencia o adaptación del modelo a nuevas condiciones del servicio.

En definitiva, la evaluación y selección del modelo constituye un punto de convergencia entre el rigor empírico y la aplicabilidad práctica. Permite garantizar que el modelo elegido no solo optimice las métricas de rendimiento, sino que también cumpla con los criterios de confiabilidad, eficiencia y explicabilidad requeridos para su adopción en entornos cloud dinámicos y sensibles a la calidad del servicio.

7.2.3.1. Variables de los modelos de aprendizaje profundo

El entrenamiento de modelos de DL implica ajustar los **parámetros del modelo** mediante un **algoritmo de aprendizaje profundo**, cuyo comportamiento está influenciado por los **hiperparámetros del modelo**, los **hiperparámetros de entrenamiento** y las propiedades del **conjunto de datos** [15].

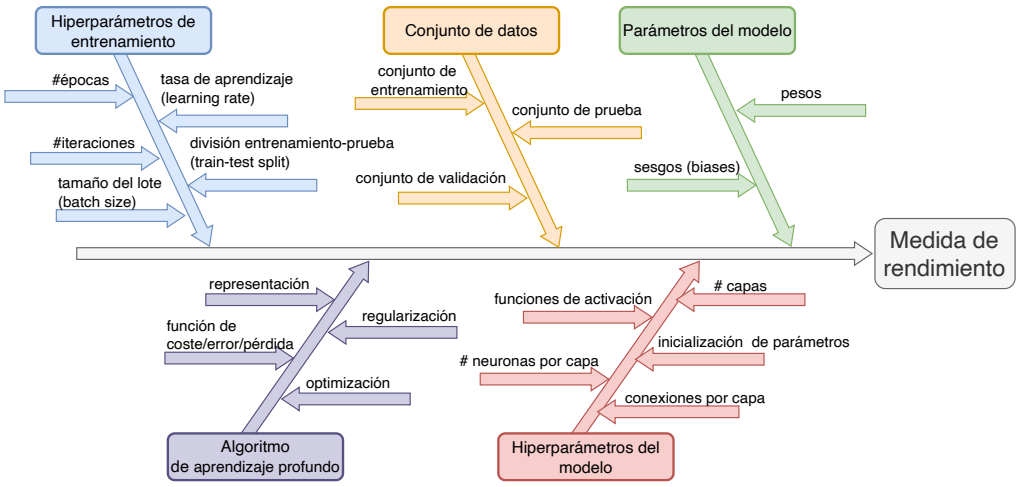


Figura 7.3: Variables de los modelos DL [4].

Los **hiperparámetros del modelo** definen la estructura arquitectónica de la red neuronal e incluyen variables como el *número de capas*, el *número de neuronas por capa*,

los *tipos de capas*—densas, convolucionales o recurrentes—, y las *funciones de activación*.

Por su parte, los **parámetros del modelo**, tales como *pesos* y *sesgos*, se ajustan iterativamente durante el proceso de entrenamiento mediante un *algoritmo de optimización* que busca minimizar una *función de pérdida*.

Las decisiones relacionadas con el **conjunto de datos** son fundamentales. Se recomienda dividir el conjunto de datos en tres subconjuntos: entrenamiento, validación y prueba, con el fin de asegurar una evaluación rigurosa del rendimiento. El *conjunto de entrenamiento* se emplea para optimizar los parámetros del modelo y permitir que la red aprenda patrones dentro de los datos. El conjunto de entrenamiento suele además dividirse en conjunto de entrenamiento y conjunto de validación [170]. El *conjunto de validación* se utiliza durante el ajuste de hiperparámetros para estimar el rendimiento del modelo y detectar problemas de bajo-ajuste (*under-fitting*) o sobre-ajuste (*over-fitting*). Finalmente, el *conjunto de prueba* evalúa la capacidad de generalización del modelo bajo condiciones no observadas.

El **algoritmo de aprendizaje profundo** se caracteriza por cuatro componentes clave: la *representación* que define la estructura de la arquitectura; la *función de coste/error/pérdida* que mide la desviación entre predicciones y valores reales; el procedimiento de *optimización* encargado de actualizar los pesos, como `Adam` o `RMSProp`; y las estrategias de *regularización*, como el *dropout* o la normalización de pesos, orientadas a reducir el sobreajuste y mejorar la generalización [15].

Por último, los **hiperparámetros de entrenamiento** controlan el proceso de aprendizaje e incluyen variables como el tamaño del lote (*batch size*), el *número de iteraciones*, el *número de épocas*, la proporción de *división entre entrenamiento y prueba* y la *tasa de aprendizaje*. El ajuste adecuado de estos hiperparámetros es determinante para la estabilidad, la velocidad de convergencia y el rendimiento final del modelo.

7.2.4. Uso del modelo

Este subproceso, ejecutado por los *ingenieros de ML u Operaciones de Aprendizaje Automático (Machine Learning Operations - MLOps)*, incluye tres actividades: el despliegue, la monitorización y el ajuste (*tuning*) del modelo. Estas actividades completan el ciclo del flujo de trabajo de ML y garantizan la integración, el seguimiento y la mejora continua del modelo en su entorno operativo frente a cambios en los datos o en las condiciones del servicio.

El **despliegue del modelo** consiste en integrar el modelo ML entrenado dentro del entorno de desarrollo o producción. Este proceso puede implicar la exposición del modelo mediante una API, su empaquetado en un contenedor o su incorporación directa en un sistema de software existente. El modelo desplegado opera para generar predicciones en tiempo real o en intervalos definidos, utilizando nuevos datos provenientes del entorno de operación. El despliegue debe incluir, además, mecanismos de gestión de versiones, trazabilidad de artefactos y control de dependencias, con el fin de garantizar la reproducibilidad del proceso de inferencia y la estabilidad del sistema en producción.

La **monitorización del modelo** tiene como propósito verificar la calidad y la fia-

bilidad de las predicciones mediante la comparación continua con los valores observados. Este seguimiento permite detectar posibles desviaciones del modelo—como la degradación del rendimiento—que pueden originarse por cambios en la distribución de los datos, la aparición de nuevos patrones o modificaciones en la infraestructura del servicio cloud. La monitorización también evalúa métricas de la operación, tales como el tiempo de inferencia y el consumo de recursos, que resultan críticas para el cumplimiento de los acuerdos de nivel de servicio (SLA) y para la evaluación continua de la calidad en servicios cloud.

El **ajuste del modelo** (*tuning*) se basa en los resultados de la monitorización y consiste en reentrenar o ajustar el modelo de forma iterativa para mejorar su rendimiento o adaptarlo a nuevas condiciones de operación. Este proceso puede implicar la modificación de hiperparámetros, la incorporación de nuevas capas o la actualización de los conjuntos de entrenamiento con datos recientes [66]. De esta forma, el ajuste del modelo contribuye a mantener la capacidad predictiva y la relevancia del modelo en entornos cloud dinámicos, cerrando así el ciclo de vida del ML orientado al pronóstico de la calidad de los servicios cloud.

7.3. Protocolo de validación empírica del proceso de modelado

La validación empírica tiene como propósito evaluar la aplicabilidad, utilidad y completitud del proceso de modelado de ML propuesto para la predicción de QoS en servicios cloud en un contexto industrial real. Esta fase se encuentra actualmente en ejecución y está orientada a obtener evidencia cualitativa y cuantitativa sobre la utilidad del proceso desde la perspectiva de expertos en ingeniería del software, ciencia de datos y gestión de la calidad de servicios.

La validación utiliza un enfoque mixto que combina técnicas cualitativas (entrevista semiestructurada) y técnicas cuantitativas (lista de comprobación o *checklist*), con el fin de capturar tanto percepciones subjetivas como mediciones objetivas de adherencia al proceso. Las entrevistas buscan comprender la experiencia y los desafíos percibidos por los participantes durante la aplicación del proceso, mientras que las listas de comprobación permiten medir el grado de cumplimiento de los requisitos definidos para cada subproceso del modelo de ML.

7.3.1. Diseño del procedimiento de validación

La validación se desarrolla en dos fases:

1. **Entrevista semiestructurada**, orientada a recabar la experiencia, dificultades y percepciones de los participantes sobre la aplicación del proceso.

2. **Lista de comprobación**, aplicada como instrumento sistemático de evaluación para medir la adherencia del proceso a los subprocesos definidos (definición de requisitos del modelo, preparación de datos, construcción, evaluación y uso del modelo).

Ambos instrumentos se aplican tras la ejecución de los experimentos de pronóstico de QoS y la obtención de los resultados iniciales de los modelos de aprendizaje profundo.

7.3.2. Entrevista semiestructurada

El propósito de la entrevista es obtener retroalimentación cualitativa sobre la comprensión y aplicabilidad del proceso, así como identificar aspectos que podrían requerir ajustes o mejoras. Las entrevistas se realizan individualmente a seis participantes, seleccionados por su experiencia en modelado predictivo, calidad de software o analítica de datos. Dos de ellos participan directamente en el desarrollo o implementación del modelo y han sido invitados a revisar y validar el proceso propuesto. Cada sesión de validación se lleva a cabo de forma individual, es grabada y transcrita, con una duración estimada de entre 45 y 60 minutos.

Este instrumento contiene seis preguntas abiertas orientadas a explorar la percepción de los expertos sobre la claridad, utilidad y facilidad de aplicación del proceso. Las preguntas abordan aspectos como el nivel de conocimiento requerido, las dificultades encontradas, la satisfacción con los resultados y las sugerencias de mejora. Las preguntas formuladas fueron las siguientes:

1. ¿Cómo de importante es tener conocimientos previos en ML para comprender y aplicar el proceso propuesto?
2. ¿Qué dificultades se encontraron al aprender y aplicar el proceso de ML?
3. ¿Son satisfactorios los resultados obtenidos mediante el proceso de ML?
4. ¿Qué agregaría al proceso para que sea más fácil de usar?
5. ¿Qué agregaría al proceso para que sea más útil?
6. ¿Tiene alguna sugerencia de mejora para el proceso?

El formato de entrevista incluye un encabezado para los datos generales del participante, espacios para respuestas y observaciones, y está disponible en formato Word para su impresión o aplicación digital.

Los resultados preliminares muestran una percepción general positiva. Los participantes coinciden en que no es necesario poseer conocimientos avanzados de ML para comprender las actividades del proceso, aunque reconocen la utilidad de conocer las restricciones y relaciones entre los elementos del modelo. Asimismo, los analistas valoran positivamente la estructura secuencial y la claridad de la definición de las actividades, indicando que favorecen la documentación de decisiones y la trazabilidad.

No obstante, se identifican oportunidades de mejora, entre ellas la inclusión de mecanismos de priorización de tareas y una mayor orientación en las etapas de ingeniería de características y ajuste del modelo. Estas observaciones serán incorporadas en una versión refinada del proceso.

7.3.3. Lista de comprobación

Para evaluar el grado de cumplimiento de los criterios definidos para el proceso propuesto, se elabora una lista de comprobación. Este instrumento permite cuantificar el nivel de adherencia del ingeniero de datos o del equipo o proyecto a los subprocesos del modelado ML para el pronóstico de QoS.

La lista de comprobación que se presenta en la Tabla 7.1 está estructurada en 20 preguntas, agrupadas en los cuatro subprocesos del modelo: (1) Requisitos del modelo, (2) Preparación de datos, (3) Construcción del modelo y (4) Uso del modelo.

Tabla 7.1: Lista de comprobación para la evaluación de modelos de pronóstico de QoS

No.	**Descripción**	**Sí**	**No**
`Requisitos del modelo`			
1	¿Está claramente definido el problema de aprendizaje?		
2	¿Se ha especificado el tipo de aprendizaje?		
3	¿Se ha especificado la tarea de aprendizaje?		
4	¿Se ha especificado el método de aprendizaje?		
`Preparación de datos`			
5	¿Está definida la QoS del servicio cloud en función de atributos de calidad?		
6	¿Están definidas las métricas de QoS para medir los atributos del servicio cloud?		
7	¿Están identificadas las fuentes de datos para las mediciones de QoS?		
8	En caso de requerirse acciones de limpieza de datos (tratamiento de valores faltantes, eliminación de duplicados, etc.), ¿se describen las acciones aplicadas y las variables afectadas?		
9	En aprendizaje supervisado, ¿se describe el formato utilizado para el etiquetado (pares entrada-salida)?		
10	¿Se han identificado las variables de entrada del modelo y su formato (univariante, multivariante)?		
11	¿Se han identificado las salidas esperadas del modelo y su formato (unistep, multistep)?		
`Construcción del modelo`			
12	¿Se ha utilizado alguna técnica de ingeniería de características?		
13	¿Se han reportado los parámetros e hiperparámetros del modelo?		
14	¿Está definido el enfoque de validación utilizado (p. ej., validación k-fold, *prequential*)?		
15	Si existe partición de datos, ¿se especifica la división entre entrenamiento y prueba?		
16	¿Se evalúa la varianza del modelo?		
17	¿Están claramente definidos el método de evaluación y las métricas de precisión?		
`Uso del modelo`			
18	¿Está definido el contexto de despliegue del modelo?		
19	¿Se realiza la monitorización del modelo en producción?		

Tabla 7.1 (continúa)

N°	Descripción	Sí	No
20	¿Se especifican los parámetros involucrados en el ajuste del modelo (p. ej., número de capas, hiperparámetros)?		

Cada ítem se evalúa con una escala ordinal de dos niveles, diseñada para captar de manera directa y consistente el grado de cumplimiento de los criterios definidos:

- 0: No cumple
- 1: Cumple

Dado que la escala es binaria, la puntuación máxima posible corresponde a $1 \times N$, donde N es el número total de preguntas. Para expresar el resultado en términos porcentuales se utiliza la fórmula:

$$\text{Score}\,\% = \frac{\text{Suma obtenida}}{N} \times 100 \tag{7.4}$$

Con el fin de interpretar adecuadamente el grado de adherencia al proceso de modelado, se establecen los siguientes rangos de evaluación:

- 85–100 %: Alta adherencia — proceso plenamente aplicable con mínimas mejoras.
- 65–84 %: Adecuado — se recomienda aplicar mejoras específicas.
- 40–64 %: Parcial — requiere revisiones significativas.
- <40 %: Baja adherencia — se recomienda replantear la adopción o realizar una adaptación profunda.

Además del resultado total, se calculan porcentajes parciales por subproceso para identificar aquellas actividades que requieren ajustes metodológicos. Este análisis diferenciado por subprocesos facilita una retroalimentación más precisa y orientada a mejoras incrementales del proceso.

7.3.4. Procedimiento de validación

El propósito de este procedimiento de validación es garantizar que las conclusiones empíricas derivadas del estudio de caso sean fiables, fundamentadas y aplicables dentro del alcance definido. Dado que el estudio se realizó en un entorno industrial real, la validación adquiere un papel central para asegurar que la conexión entre los datos, los métodos empleados y las conclusiones obtenidas sea sólida, consistente y trazable. Conforme a las directrices de validez propuestas por Runeson et al. [37], el procedimiento busca reducir riesgos asociados a interpretaciones erróneas, sesgos contextuales y limitaciones metodológicas propias de los estudios de caso en ingeniería de software.

Los objetivos que dirigen este procedimiento de validación son:

1. Confirmar la corrección técnica de las implementaciones, asegurando que los modelos, *scripts* y procesos operen conforme a lo especificado.

2. Garantizar la integridad, representatividad y coherencia del conjunto de datos, verificando que los patrones observados correspondan a condiciones reales del servicio.

3. Evaluar de forma rigurosa el comportamiento de los modelos de ML, fortaleciendo la solidez de las interpretaciones, inferencias y comparaciones derivadas del análisis empírico.

4. Explorar la robustez de los resultados frente a variaciones operativas, analíticas o contextuales, con el fin de valorar su estabilidad y determinar en qué medida los resultados pueden transferirse a contextos similares o explicarse a partir de las particularidades del entorno industrial.

Para abordar estos objetivos, la validación se estructura en cinco etapas, cada una con un propósito definido y alineado con los objetivos específicos, contribuyendo de manera explícita al fortalecimiento de uno o más tipos de validez.

1. **Preparación**

   Esta etapa establece las condiciones iniciales para una evaluación rigurosa y colaborativa. Se seleccionan los participantes—desarrolladores, ingenieros de datos, científicos de datos y responsables de monitorización (MLOps)—con conocimiento directo del servicio y de su comportamiento operativo. La diversidad de roles contribuye a reforzar la validez contextual, al incorporar múltiples perspectivas sobre la calidad del proceso y del servicio monitorizado.

   A los participantes se les distribuye un conjunto de materiales diseñados para guiar la revisión, que incluye: (i) una guía detallada del proceso de modelado y evaluación, (ii) una lista de comprobación alineada con el protocolo del estudio, y (iii) un formato de entrevista para capturar las valoraciones cualitativas.

   El propósito de esta etapa es asegurar que todos los participantes compartan un entendimiento común de los objetivos, conceptos y criterios de evaluación, reduciendo el riesgo de interpretaciones inconsistentes y reforzando la validez de constructo desde el inicio del proceso.

2. **Aplicación**

   En esta etapa se ejecuta el proceso de validación mediante entrevistas estructuradas y la aplicación conjunta de la lista de comprobación. Las sesiones—presenciales o virtuales— se enfocan en: (i) la revisión de los modelos de ML implementados, (ii) la exploración del conjunto de datos y la trazabilidad entre métricas, (iii) la verificación técnica de las implementaciones—*scripts* o *notebooks*—de preprocesamiento, entrenamiento y evaluación, y (iv) el análisis guiado de las predicciones y su correspondencia con eventos operativos.

Esta etapa permite contrastar la evidencia empírica con el conocimiento experto de los participantes, fortaleciendo tanto la validez de constructo como la validez interna, al verificar las relaciones causales asumidas en el estudio (p. ej., que los resultados no se deben a errores técnicos o a artefactos del preprocesamiento). Asimismo, facilita la identificación temprana de discrepancias entre el comportamiento esperado y los resultados observados, lo cual es fundamental para detectar anomalías antes del análisis formal.

3. **Consolidación**

   Posterior a la aplicación del proceso de construcción de los modelos de ML, se recolectan las puntuaciones de las listas de comprobación y las respuestas cualitativas registradas durante las entrevistas. Esta etapa organiza la evidencia recopilada y la estructura en torno a los subprocesos del protocolo: preprocesamiento, ingeniería de características, entrenamiento, evaluación estadística e interpretación.

   La consolidación fortalece la validez de conclusión, al establecer una base de análisis coherente, sistemática y verificable. También contribuye a la validez interna al facilitar la identificación de factores no controlados, posibles fuentes de error o desviaciones respecto al protocolo que podrían afectar la interpretación de los resultados.

4. **Análisis**

   Esta etapa consiste en evaluar la evidencia recopilada mediante técnicas cuantitativas y cualitativas. Se calcula la puntuación total y por subproceso, se categoriza el nivel de adherencia al protocolo—alto, medio o bajo—, y se interpretan las observaciones cualitativas mediante la triangulación de ambas fuentes de información.

   El objetivo es determinar la consistencia metodológica del proceso e identificar fortalezas, debilidades, desviaciones del protocolo, riesgos para la fiabilidad de los resultados y oportunidades de mejora para futuras aplicaciones.

   En definitiva, esta etapa refuerza principalmente la validez de conclusión, asegurando que las inferencias están respaldadas por patrones coherentes en los datos y en las opiniones de los participantes. También apoya la validez interna al permitir considerar explicaciones alternativas.

5. **Retroalimentación**

   La validación culmina con un proceso formal de integración de las observaciones en una versión revisada del protocolo, de las implementaciones o de las conclusiones del estudio. Entre las posibles implicaciones se incluyen ajustes y aclaraciones en el protocolo, modificaciones en los procedimientos de preprocesamiento, reestructuración parcial del proceso, mejoras en la trazabilidad, refinamiento de las interpretaciones y visualizaciones, así como una mayor explicitación de supuestos y limitaciones.

   Este ciclo iterativo fortalece la validez de constructo y la validez interna, al favorecer la adaptación del proceso y potenciar su transferencia a contextos similares en la industria.

Este procedimiento de validación, alineado con los criterios de validez propuestos por Runeson et al. [37], garantiza que el estudio de caso mantenga el rigor metodológico a pesar de las limitaciones inherentes al contexto industrial. Cada etapa contribuye a reforzar la validez de constructo, interna, de conclusión y contextual, consolidando la confianza en los resultados y en la utilidad práctica del proceso de modelado de ML para el pronóstico de QoS.

7.3.5. Resultados esperados

Como resultado de la ejecución del procedimiento de validación, se espera obtener evidencia clara y razonada que permita sustentar la calidad metodológica del estudio de caso. En primer lugar, se prevé confirmar la corrección técnica de las implementaciones realizadas, verificando que los modelos, *scripts*, *notebooks* y transformaciones se comportan conforme a sus especificaciones. Esto permite reducir amenazas a la validez interna asociadas a errores técnicos o configuraciones defectuosas, asegurando que los resultados proceden de los modelos y no de artefactos del proceso.

Asimismo, se espera obtener evidencia de que los datos utilizados son íntegros y representativos del comportamiento real del servicio cloud. Ello implica demostrar que los patrones observados no provienen de fallos en el proceso de extracción, anomalías del sistema de monitorización o inconsistencias entre métricas. Esta verificación refuerza la validez de constructo al asegurar que las mediciones empleadas reflejan adecuadamente los atributos de calidad del servicio, y también fortalece la validez contextual, pues garantiza que el conjunto de datos describe un comportamiento auténtico del entorno industrial.

Un tercer resultado esperado es la evaluación sólida del comportamiento de los modelos, basada tanto en métricas de desempeño como en interpretaciones cualitativas que permitan contextualizar los valores obtenidos. Este análisis permitirá determinar si las conclusiones sobre la eficacia de los modelos son justificadas, estables y compatibles con la evidencia disponible. En términos de Runeson et al. [37], este resultado contribuye a la validez de conclusión, ya que se fundamentan inferencias sólidas basadas en datos suficientes, consistentes y correctamente interpretados.

Finalmente, se espera identificar claramente las limitaciones del proceso y las oportunidades de mejora, tanto en la construcción de los modelos como en el flujo de análisis y documentación. Este resultado refuerza la validez externa al permitir comprender en qué medida los hallazgos pueden generalizarse a otros servicios cloud y bajo qué condiciones el proceso es transferible. Además, posibilita mejorar el protocolo, haciéndolo más robusto frente a variaciones técnicas, operativas o contextuales.

7.4. Conclusiones

Este capítulo describe el proceso metodológico de modelado basado en aprendizaje automático (ML) propuesto para el pronóstico de la calidad del servicio (QoS) en servicios cloud. Este proceso constituye un aporte metodológico relevante, al proporcionar una guía

estructurada y alineada con los principios de la ingeniería del software para el desarrollo de modelos de pronóstico aplicables a escenarios industriales.

El proceso integra de forma coherente las fases de definición de requisitos, preparación de datos, construcción del modelo y uso del modelo, abarcando desde la identificación de métricas de QoS y fuentes de datos hasta la evaluación y monitorización del desempeño del modelo. Asimismo, se ha incorporado una validación empírica como parte de la validación académica, mediante la aplicación combinada de entrevistas semiestructuradas y listas de verificación. Esta validación permite recopilar evidencia cualitativa y cuantitativa sobre la utilidad, aplicabilidad y coherencia interna del proceso desde la perspectiva de expertos e investigadores del dominio.

Los resultados esperados de esta validación contribuirán a refinar la propuesta metodológica, fortaleciendo su robustez y su potencial de transferencia hacia contextos industriales. Además, proporciona una base sólida para la construcción, comparación y evaluación empírica de modelos de pronóstico de QoS, todos ellos aspectos que se abordan en el siguiente capítulo.

Capítulo 8

Modelos de Pronóstico de QoS para Servicios Cloud

En este capítulo se presenta un experimento controlado siguiendo las directrices de Wohlin et al. [36], cuyo propósito es comparar el desempeño de dos modelos de aprendizaje profundo—BI-GRU y LSTM—frente a ARIMA, un método estadístico ampliamente utilizado en el análisis de series temporales. Los modelos fueron desarrollados con el objetivo de generar pronósticos de la calidad de un servicio cloud en operación dentro de un entorno industrial.

La Sección 8.1 introduce el propósito y la importancia del estudio de caso para evaluar el modelo de calidad en un contexto industrial. La Sección 8.2 escribe el diseño, planificación y operación de un experimento controlado cuyo propósito es evaluar rigurosamente la capacidad predictiva de diferentes modelos de DL y ARIMA aplicados al pronóstico de métricas de calidad de servicio (QoS) en servicios cloud. La Sección 8.3 presenta la estadística descriptiva y un análisis exploratorio de datos de los resultados. Además, se presentan las pruebas de hipótesis para dar respuesta a las preguntas de investigación y la retroalimentación preliminar. La Sección 8.4 presenta las amenazas de la validez del experimento. La Sección 8.5 presenta las implicaciones para la investigación y la práctica. La Sección 8.6 presenta las conclusiones de los resultados del experimento controlado.

8.1. Introducción

El uso de los servicios cloud se ha incrementado de manera sostenida en las organizaciones debido a su flexibilidad, escalabilidad y eficiencia en el uso de recursos. Sin embargo, este crecimiento ha venido acompañado de una mayor complejidad para garantizar la calidad del servicio. Factores como la asignación dinámica de recursos, la heterogeneidad de los entornos cloud y las expectativas variables de los clientes dificultan el cumplimiento de los SLA, los cuales establecen compromisos explícitos sobre atributos de calidad, umbrales de rendimiento, modelos de precios y penalizaciones por incumplimiento [154, 155, 156].

La monitorización continua de la calidad de servicio es fundamental para asegurar el cumplimiento de estos acuerdos. Los modelos de calidad permiten especificar requisitos, estructurar atributos y configurar mecanismos de monitorización [21], constituyendo así un componente central en la gestión eficaz de los SLA. Las métricas bien definidas [2] son esenciales para medir y mantener la calidad del servicio cloud. No obstante, los modelos de monitorización y evaluación son predominantemente reactivos y carecen de capacidades predictivas, resultando insuficientes para satisfacer las necesidades actuales de mantenimiento proactivo, detección temprana de anomalías y prevención de posibles violaciones de SLA.

El aprendizaje automático (ML) emerge como una alternativa prometedora al ofrecer capacidades avanzadas para modelar dependencias complejas y mejorar el pronóstico de la calidad del servicio. Técnicas estadísticas y métodos de ML ampliamente utilizados—como ARIMA, Holt-Winters o la programación genética (GP)—han sido aplicados para predecir métricas de QoS, especialmente en entornos estables o de corto plazo. No obstante, estos métodos presentan algunas limitaciones importantes, entre ellas dificultades para capturar patrones no lineales, dependencias temporales complejas y correlaciones de largo plazo, intrínsecas a los entornos cloud.

El aprendizaje profundo (DL) ofrece oportunidades significativas al permitir el modelado de relaciones temporales en series temporales. Dentro del DL, las redes neuronales recurrentes (RNN) han demostrado gran utilidad al procesar secuencias de longitud variable mientras mantienen un estado oculto que captura dependencias temporales [15]. Entre sus variantes, LSTM y GRU destacan por su capacidad para modelar dependencias de largo plazo, y las variantes bidireccionales aprovechan simultáneamente la información pasada y futura para detectar patrones [16, 68].

Este capítulo propone y evalúa modelos de aprendizaje profundo (BI-GRU y LSTM) para el pronóstico de QoS en servicios cloud en un entorno industrial real. Para ello, se construyó un conjunto de datos multidimensional con cinco meses de métricas provenientes de un servicio cloud en producción, abarcando 16 atributos de QoS que incluyen métricas de utilización de recursos (p. ej., % de CPU usada, % de memoria usada, I/O de disco, tráfico de red), capacidad (p. ej., memoria libre/ocupada, disco libre/ocupado, paquetes Rx/Tx) y disponibilidad (p. ej., ratio de tiempo activo). Adicionalmente, se llevó a cabo un experimento controlado para comparar los modelos BI-GRU y LSTM con ARIMA, con el fin de evaluar su precisión, eficiencia y aplicabilidad práctica.

En síntesis, este capítulo presenta cuatro contribuciones principales:

1. **Protocolo experimental riguroso y reproducible**: se sigue un proceso alineado con las mejores prácticas de ingeniería del software empírica, incluyendo una descripción completa del conjunto de datos, arquitecturas, hiperparámetros y procedimientos de evaluación.

2. **Construcción de un conjunto de datos de QoS**: se presenta un dataset de cinco meses con 16 métricas de QoS provenientes de un servicio cloud real, abordando la limitada disponibilidad de conjuntos de datos actualizados y documentados en este campo.

3. **Comparación experimental de modelos de predicción**: se evalúan empíricamente BI-GRU, LSTM y ARIMA mediante RMSE, MAE y MAPE, considerando además el tiempo de entrenamiento para facilitar un análisis de compensaciones entre precisión y eficiencia.

4. **Paquete de replicación de acceso abierto**: se incluye un paquete con datos, código, configuraciones y scripts, lo que permite la reproducibilidad y extensión del estudio por parte de la comunidad académica.

La siguiente sección detalla esta planificación del experimento, describiendo los criterios adoptados, las etapas del diseño experimental y la forma en que estos elementos guían el desarrollo del estudio.

8.2. Experimento controlado

Esta sección presenta el experimento controlado llevado a cabo para desarrollar modelos de predicción de la calidad de servicio (QoS) y evaluar su rendimiento. El diseño experimental se fundamenta en las directrices para estudios empíricos propuestas por Wohlin et al. [36] y Kitchenham et al.[171], complementadas con adaptaciones inspiradas en el marco experimental de Vegas y Elbaum[4]. Estas adaptaciones responden a las particularidades de los modelos de aprendizaje profundo (*Deep Learning - DL)*, especialmente en lo referente a la selección y ajuste de hiperparámetros (tanto arquitectónicos como de entrenamiento), las configuraciones del algoritmo, las propiedades del conjunto de datos y los parámetros del modelo (véase la Sección 7.2.3.1).

8.2.1. Definición

La fase de definición establece el propósito del estudio, sus motivaciones, el objetivo y las hipótesis de investigación. En el contexto del experimento, esta fase es esencial para garantizar la coherencia entre las preguntas de investigación, la selección de los atributos de QoS, el diseño del experimento y la posterior evaluación de los modelos de pronóstico mediante aprendizaje profundo. Asimismo, proporciona el marco conceptual que orienta las decisiones metodológicas en las etapas posteriores de planificación y operación.

El problema de aprendizaje se define como el pronóstico de la calidad del servicio cloud SAlert en un entorno operativo real. La QoS se concibe como un constructo multidimensional compuesto por características, atributos y métricas que reflejan el comportamiento del servicio. Obtener pronósticos precisos de estos indicadores permite implementar una gestión proactiva, anticipar degradaciones, facilitar la revisión de acuerdos de nivel de servicio (SLA) y reducir la probabilidad de violaciones. Para abordar este problema, se emplea un enfoque de regresión aplicado a series temporales obtenidas mediante mecanismos de monitorización del servicio [97]. El objetivo es estimar los valores futuros de las métricas monitorizadas, que podrán ser comparadas con los umbrales establecidos en los SLA, ofreciendo así una base cuantitativa para la toma de decisiones preventivas.

El pronóstico de QoS aporta beneficios clave tanto para proveedores como para consumidores de servicios cloud. Por un lado, permite seleccionar servicios en función de su rendimiento esperado; por otro, fortalece la gestión de los SLA al minimizar las desviaciones operativas y garantizar su cumplimiento. En este estudio, el rendimiento de los modelos se evalúa a partir de su capacidad para generar predicciones precisas, utilizando un conjunto de prueba independiente, conforme a los sugerido por Mitchell [168] y Hyndman [60]. Esta etapa constituye el fundamento metodológico sobre el cual se desarrollan las fases posteriores del experimento.

8.2.1.1. Objetivo y pregunta de investigación

Siguiendo el paradigma Objetivo-Pregunta-Métrica (*Goal Question Metric* - GQM) [43], el objetivo de este experimento es:

> ***Evaluar empíricamente el rendimiento*** *de modelos basados en DL (específicamente BI-GRU y LSTM) y del modelo ARIMA* ***con el propósito de*** *pronosticar la calidad de servicios cloud,* ***con respecto a*** *su precisión predictiva y su eficiencia computacional.*

La pregunta de investigación (RQ) que guía este estudio es:

> ***RQ.*** *¿Los modelos de pronóstico basados en BI-GRU superan a los modelos LSTM y ARIMA en términos de precisión y eficiencia de entrenamiento al pronosticar la calidad de los servicios cloud?*

Para abordar esta pregunta, se comparan tres modelos con enfoques tanto estadísticos como de DL. Dentro del enfoque estadístico, ARIMA es ampliamente utilizado para el pronóstico de series temporales [60, p.290]. Por otro lado, BI-GRU y LSTM corresponden a modelos de DL, concretamente variantes de RNN, que han mostrado resultados prometedores en diversos problemas de datos secuenciales. Sin embargo, estos modelos requieren una mayor validación empírica en contextos de pronóstico de QoS para servicios cloud.

8.2.2. Planificación

La fase de planificación establece el diseño metodológico que permite ejecutar el estudio de forma rigurosa, sistemática y bajo condiciones controladas. En el contexto de esta investigación, esta fase integra decisiones relativas al conjunto de datos, los modelos a comparar, las métricas de evaluación, los procedimientos de validación analítica y los mecanismos destinados a garantizar la validez del estudio de caso. Su propósito es asegurar que la fase de operación se lleve a cabo de forma reproducible, trazable y alineada con los objetivos formulados en la fase de definición.

8.2.2.1. Contexto y alcance

El experimento se lleva a cabo ***en el contexto de*** Golden S.A., una empresa de TI de tamaño mediano que cuenta aproximadamente con cincuenta empleados y desarrolla sus operaciones en México, Argentina, Puerto Rico y Ecuador. La empresa desarrolla y opera **Social Alert (SAlert)**, un producto SaaS diseñado para el análisis de redes en tiempo real y el análisis de opiniones basado en contenido (véase la Sección8.2.2.6).

Tras un proceso de consulta con varias organizaciones, se seleccionó el servicio de ciencia de datos SaaS Social Alert (SAlert), desarrollado y operado por Golden S.A.. SAlert es un servicio que procesa datos en tiempo real mediante análisis de redes y técnicas basadas en contenido, incluyendo análisis de sentimientos, para extraer, clasificar y evaluar las opiniones de los usuarios sobre eventos sociales, campañas de marketing, productos o tendencias políticas [172]. Esta funcionalidad intensiva en datos convierte al servicio en un escenario adecuado para estudiar la calidad operativa en condiciones reales.

8.2.2.2. Hipótesis

Las ***hipótesis nulas*** del experimento se definen de la siguiente manera:

- $H1_0$: No existen diferencias significativas en la precisión de la predicción de QoS entre los modelos BI-GRU, LSTM y ARIMA.
- $H2_0$: No existen diferencias significativas en la eficiencia del entrenamiento para el pronóstico de QoS entre los modelos BI-GRU, LSTM y ARIMA.

El análisis estadístico busca poner a prueba estas hipótesis. El rechazo de $H1_0$ respaldaría alguna de las siguientes ***hipótesis alternativas***:

- $H1_{a1}$: BI-GRU y ARIMA difieren significativamente en la precisión de la predicción de QoS.
- $H1_{a2}$: BI-GRU y LSTM difieren significativamente en la precisión de la predicción de QoS.
- $H1_{a3}$: LSTM y ARIMA difieren significativamente en la precisión de la predicción de QoS.

Del mismo modo, el rechazo de $H2_0$ respaldaría alguna de las siguientes ***hipótesis alternativas***:

- $H2_{a1}$: BI-GRU y ARIMA difieren significativamente en la eficiencia de entrenamiento para el pronóstico de QoS.
- $H2_{a2}$: BI-GRU y LSTM difieren significativamente en la eficiencia de entrenamiento para el pronóstico de QoS.
- $H2_{a3}$: LSTM y ARIMA difieren significativamente en la eficiencia de entrenamiento para el pronóstico de QoS.

Para comprobar estas hipótesis, se implementaron los modelos BI-GRU, LSTM y ARIMA con el fin de pronosticar la QoS del servicio SAlert y se compararon mediante una evaluación empírica. En las subsecciones siguientes se describen en detalle el diseño y la instrumentación del experimento.

8.2.2.3. Variable independiente

La ***variable independiente*** es el modelo de predicción, con tres tratamientos: BI-GRU, LSTM y ARIMA. Cada modelo realiza pronósticos univariantes de un solo paso para cada una de las métricas de QoS. Aunque el horizonte de pronóstico considerado es de un solo paso, estos modelos pueden aplicarse de forma recursiva para generar pronósticos multi-step y evaluar horizontes temporales extendidos [63].

La selección de los modelos siguió una estrategia estructurada por método/modelo. Se eligió el método de pronóstico, aplicando tanto la selección intraenfoque (*intra-approach*) como la selección entre enfoques (*cross-approach*) propuesta por Syu et al. [18]. Esta estrategia dual de selección de modelos amplía el alcance comparativo del estudio y aumenta la probabilidad de identificar el modelo más adecuado para el problema analizado [18].

En la selección *intraenfoque* se examinaron algoritmos dentro del mismo enfoque de DL, seleccionándose LSTM y BI-GRU por su capacidad para capturar dependencias y patrones de largo plazo en datos secuenciales, lo que favorece predicciones precisas en horizontes temporales prolongados [16, 173, 17]. En la selección *entre enfoques*, se incorporó el modelo ARIMA como representante del paradigma de modelo estadístico.

8.2.2.4. Variables dependientes

Las ***variables dependientes*** reflejan el rendimiento de los modelos tanto desde la perspectiva de la precisión del pronóstico como de la eficiencia del entrenamiento. La precisión del pronóstico evalúa la capacidad del modelo para ajustarse a los datos reales. Para ello, se compara el rendimiento del modelo BI-GRU con los modelos LSTM y ARIMA utilizando tres métricas de precisión: RMSE, MAE y MAPE. Por su parte, la eficiencia computacional se evalúa a través del tiempo de entrenamiento.

El RMSE amplifica el efecto de los valores atípicos al elevar al cuadrado los errores antes de promediarlos. El MAE mide las diferencias absolutas medias, mientras que el MAPE

normaliza los errores como porcentajes, facilitando su interpretabilidad. En conjunto, estas métricas permiten caracterizar de manera integral el error de predicción y proporcionan distintos niveles de sensibilidad frente a valores atípicos [17, 18, 102]. Sus expresiones matemáticas se presentan a continuación:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i^*)^2} \tag{8.1}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - y_i^*| \tag{8.2}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - y_i^*|}{y_i} \tag{8.3}$$

Donde N es el tamaño del conjunto de datos que representa la duración del período de predicción, y_i es el valor observado y y_i^* denota el valor de QoS predicho para el servicio cloud en el instante i.

8.2.2.5. Diseño experimental

El diseño experimental adoptado es de un solo factor, donde dicho factor corresponde al modelo de pronóstico de QoS. Este factor se operacionaliza mediante tres tratamientos (BI-GRU, LSTM y ARIMA) aplicados al mismo objeto experimental: el servicio SaaS SAlert. La comparación se realiza bajo un esquema uniforme de ejecución, asegurando que cada modelo se entrene con las mismas particiones temporales, métricas de entrada y bajo las mismas condiciones de entrenamiento, lo que refuerza la validez interna del estudio.

El alcance del experimento se limita a un único servicio SaaS que operó sobre una infraestructura estable, sin modificaciones arquitectónicas relevantes durante el período de observación. Estas condiciones controladas reducen el ruido asociado a cambios estructurales o a variaciones abruptas de la carga de trabajo, lo cual fortalece la validez interna del estudio a la vez que refleja el comportamiento operativo real de un servicio cloud en un entorno industrial.

Para adaptar el modelo de calidad genérico [158] al contexto operativo de SAlert, se llevó a cabo un taller colaborativo entre la autora y el equipo evaluador de la empresa—conformado por el gerente de TI y dos desarrolladores con experiencia en monitorización—empleando la plataforma *Microsoft Teams*. Durante esta sesión *online*, se seleccionaron atributos y métricas relevantes e independientes de la plataforma, que posteriormente se integraron en la infraestructura corporativa de monitorización mediante agentes y APIs de *New Relic* [174]. Este paso habilitó un proceso de monitorización continua y homogénea, fundamental para garantizar la coherencia metodológica del estudio.

El conjunto de datos obtenido corresponde a cinco meses de monitorización continua del servicio SAlert, configurada con un intervalo de muestreo fijo de 30 minutos. Se recopilaron 16 atributos de QoS, abarcando métricas de utilización de recursos, rendimiento de red, capacidad y disponibilidad. Este conjunto resulta más diverso y robusto que los comúnmente empleados en estudios de pronóstico de QoS, en los que suele predominar la métrica de tiempo de respuesta.

Todas las observaciones fueron capturadas en un entorno productivo real, sin recurrir a datos sintéticos o simulados, lo que incrementa de manera significativa la validez externa del estudio. Las métricas recoplidas se agrupan de la siguiente forma:

- `Utilización de recursos`: porcentaje de uso de memoria, porcentaje de uso de disco, tasas de lectura y escritura, porcentaje de CPU utilizada.
- `Rendimiento de red`: netBytes In/Out, netPackets In/Out, paquetes recibidos (rx packets), paquetes transmitidos (tx packets).
- `Capacidad`: memoria libre, memoria utilizada, disco libre, disco usado.
- `Disponibilidad`: índice de tiempo de actividad (*uptime*).

El diseño experimental aborda las limitaciones recurrentes identificadas en la literatura, tales como la baja diversidad de métricas, el uso de conjuntos de datos reducidos o sintéticos, la escasa claridad metodológica, la falta de reproducibilidad y la ausencia de protocolos de validación rigurosos [4, 171]. Con el propósito de superar estas restricciones, se estableció un entorno experimental riguroso que permitiera evaluar de manera sistemática la capacidad predictiva de los modelos, así como garantizar la validez y la fiabilidad de los resultados.

8.2.2.6. Objeto experimental

El objeto experimental es una instancia del servicio **SAlert** desplegada en el entorno de producción de Golden S.A.. El conjunto de datos comprende observaciones de las métricas capturadas cada 30 minutos en el periodo comprendido entre marzo y julio de 2023. En consecuencia, considerando 48 mediciones por día durante 153 días, el conjunto de datos resultante contiene 7344 observaciones por métrica, representando un volumen sustancial y adecuado para el entrenamiento de modelos de pronóstico.

Todos los datos se obtuvieron a partir de cargas operativas reales, sin aplicar técnicas de aumentación o sintetización de datos, garantizando que los modelos de pronóstico reflejen patrones genuinos de variabilidad y comportamiento del sistema.

El preprocesamiento siguió un procedimiento sistemático y reproducible que incluyó: limpieza de datos, detección de anomalías, normalización y construcción del conjunto de datos supervisado.

La *limpieza de datos* contempló la eliminación de duplicados, la gestión de valores atípicos y el tratamiento de valores perdidos.

La *detección de anomalías* se llevó a cabo mediante la inspección visual de los patrones estacionales, y se definieron como desviaciones significativas de las distribuciones estadísticas esperadas [163]. En particular, se detectó una anomalía estacional en las métricas de disco libre, disco usado y porcentaje de disco usado a partir del 5 de julio, durante un periodo de 27 días, provocada por la interrupción del agente de monitorización debido a una migración planificada de infraestructura. Este segmento fue eliminado para evitar problemas de convergencia durante el entrenamiento, resultando en un total de 6049 observaciones por métrica.

En cuanto a la *normalización*, todos los valores fueron escalados al rango $[-1, 1]$ mediante la técnica `MinMax` que favorece la convergencia de las redes neuronales y mejora la precisión de las predicciones [175, 165].

Para la *construcción del conjunto de datos supervisado*, cada instancia de entrada se emparejó con su valor correspondiente en el siguiente paso temporal (predicción a un paso), generando dos matrices para cada métrica: X (entradas) e Y (salidas). Este procedimiento se aplicó de forma independiente a cada una de las 16 métricas de QoS, dando lugar a 16 tareas paralelas, una por cada métrica de calidad.

8.2.2.7. Entrenamiento y validación

La fase de entrenamiento y validación contempló los tres enfoques (BI-GRU, LSTM y ARIMA) para el pronóstico *one-step-ahead* de 16 métricas de QoS en un entorno SaaS de producción. Durante el entrenamiento, se siguió el marco de clasificación de variables propuesto por Vegas y Elbaum [4], quienes identificaron cinco grupos de variables en experimentos con modelos DNN. En este contexto, se ajustaron los hiperparámetros de los modelos, incluyendo el número de neuronas por capa, el tamaño de lote y la tasa de aprendizaje, buscando un equilibrio entre la precisión y la eficiencia (véase la Sección 7.2.3.1).

Para evaluar la capacidad de generalización de los modelos y detectar posibles problemas de sobreajuste (*overfitting*) o bajoajuste (*underfitting*), se implementó una estrategia de **validación cruzada temporal**. La validación cruzada clásica *k-fold* fue descartada por su incompatibilidad con datos secuenciales, ya que el muestreo aleatorio rompe las dependencias temporales y compromete la validez de las estimaciones [61, 166, 167]. Del mismo modo, la validación *out-of-sample*, aunque respeta la secuencia temporal, se basa en una única partición de los datos, lo que puede generar estimaciones poco representativas o sensibles a segmentos anómalos de la serie [61].

Para mitigar estas limitaciones y asegurar una evaluación más estable y libre de sesgos, se empleó una estrategia de *prequential cross-validation*. Este enfoque preserva la secuencia temporal, generando múltiples particiones de entrenamiento y prueba de igual tamaño, sin solapamientos. De este modo, se evita la fuga de información entre conjuntos y se refuerza la validez interna del estudio.

En concreto, se generaron cuatro bloques temporales equivalentes. Cada bloque actuó secuencialmente como conjunto de prueba, mientras que los bloques anteriores se acumularon para conformar el conjunto de entrenamiento. Adicionalmente, dentro de cada

conjunto de entrenamiento, se aplicó una nueva partición interna 80/20 para separar los datos de entrenamiento y validación, utilizada durante el ajuste de los modelos.

Cada modelo se entrenó en 3 iteraciones para cada una de las 16 métricas de QoS, dando lugar a un total de 48 tareas de predicción. Además, para mitigar la variabilidad introducida por la inicialización aleatoria de los parámetros y por los procesos de entrenamiento, cada experimento se repitió mediante ejecuciones independientes empleando exactamente las mismas particiones de datos.

El rendimiento de los modelos se evaluó utilizando tres métricas de precisión—RMSE, MAE y MAPE—complementadas con una métrica de eficiencia computacional: el tiempo de entrenamiento medido en segundos. Tanto la exactitud predictiva como el coste computacional son especialmente relevantes en contextos industriales, donde la escalabilidad y el uso eficiente de los recursos son factores críticos para la adopción de soluciones de predicción basadas en el aprendizaje automático.

Los modelos de referencia ARIMA se implementaron siguiendo la metodología de Box-Jenkins para optimizar la selección de los parámetros (p, d, q) mediante procedimientos automáticos de minimización del criterio correspondiente.

8.2.2.8. Instrumentación

La instrumentación del experimento requirió una infraestructura de *software* y *hardware* capaz de soportar tanto el entrenamiento intensivo de modelos de aprendizaje profundo como la gestión de múltiples ejecuciones repetidas, necesarias para garantizar la estabilidad de los resultados. Todos los experimentos se llevaron a cabo en *Google Colab* [176], un entorno basado en *Jupyter Notebook* que proporciona recursos de cómputo cloud con aceleración por GPU.

El entrenamiento de los modelos de aprendizaje profundo se ejecutó en una GPU NVIDIA T4 con 16 GB de memoria, mientras que las actividades de despliegue y monitorización se realizaron en una máquina virtual integrada en el entorno operativo de la empresa de TI responsable del servicio SaaS analizado. Esta máquina estaba equipada con una GPU NVIDIA MX130, 8 GB de memoria RAM, 1000 GB de almacenamiento y un procesador Intel i7-7500U a 2.7 GHz.

La instrumentación descrita complementa directamente la planificación experimental presentada anteriormente en la Sección 8.2.2. En esta investigación se emplean de manera conjunta estrategias de comparación intraenfoque y entre enfoques. La comparación intraenfoque permite ajustar y contrastar distintas configuraciones en los modelos basados en aprendizaje profundo, mientras que la comparación entre enfoques permite evaluar el rendimiento relativo de los modelos de DL frente a métodos tradicionales como ARIMA. De este modo, se determina el valor añadido de las técnicas de aprendizaje profundo para la predicción de métricas de QoS en entornos industriales.

Esta articulación entre el diseño experimental y la instrumentación proporciona una base metodológica sólida para el análisis cuantitativo y cualitativo presentado a continuación en la Sección 8.2.3.2.

8.2.3. Operación

Esta fase abarcó la ejecución práctica del experimento siguiendo el plan establecido. Su propósito es garantizar que la recolección de datos, el entrenamiento y validación de los modelos, así como el análisis de resultados, se realicen de manera uniforme, controlada y sistemática. De este modo, la fase de operación transforma el diseño metodológico en actividades observables y reproducibles, asegurando que las conclusiones derivadas se fundamenten en evidencia empírica sólida y en la aplicación rigurosa de los procedimientos de análisis estadístico.

8.2.3.1. Procedimiento de ejecución

La preparación del entorno experimental tiene el propósito de garantizar que la ejecución del estudio se realice bajo condiciones controladas, reproducibles y alineadas con el diseño establecido durante la fase de planificación. Para ello, se estableció una infraestructura computacional homogénea, se automatizaron los procedimientos de recolección y preprocesamiento de datos y se configuraron los modelos de DL de forma consistente para todas las métricas de QoS.

El experimento implicó el entrenamiento de un total de 48 modelos de pronóstico independientes (16 métricas $\times$ 3 enfoques), evaluados a lo largo de las particiones temporales definidas por la validación *prequential*. Cada configuración se ejecutó mediante 10 réplicas independientes, lo que dio lugar a un conjunto amplio de observaciones necesarias para la posterior evaluación estadística. Durante todas las ejecuciones, se registraron de manera sistemática las métricas de precisión (RMSE, MAE y MAPE), así como el tiempo de entrenamiento, con el objetivo de medir el desempeño de cada ejecución y analizar el comportamiento comparativo de los modelos.

El entorno de ejecución se implementó en Python v3.12 utilizando la plataforma Google Colab [176], la cual ofrece aceleración por GPU (NVIDIA T4) y un entorno controlado, adecuado para la experimentación reproducible de modelos basados en DL. Los modelos BI-GRU y LSTM se implementaron utilizando las bibliotecas `TensorFlow` y `Keras`, mientras que el modelo ARIMA se desarrolló mediante la librería especializada `pmdarima`.

Para la obtención de los datos de monitorización se diseñó un procedimiento automatizado de extracción, transformación y carga (ETL) basado en la API *NerdGraph* de *New Relic* [174]. Un *script* en Python ejecutó consultas NRQL programadas para recuperar las 16 métricas de QoS y almacenarlas en archivos CSV con un total de 17 columnas: una columna correspondiente a las marcas temporales y 16 asociadas a las mediciones de las métricas. Este mecanismo garantizó la consistencia temporal, ausencia de duplicados y trazabilidad completa de los datos, aspectos fundamentales para la validez de constructo del experimento.

Las tareas de limpieza, manipulación y preprocesamiento de los datos se implementaron con `NumPy` y `Pandas`. La inspección visual de anomalías, estacionalidad y patrones de variabilidad se apoyó en rutinas desarrolladas con `Matplotlib` y `Seaborn`. Este análisis exploratorio inicial permitió identificar y corregir irregularidades, reforzando la calidad y representatividad del conjunto de datos.

En cuanto a las métricas de precisión, el RMSE se definió mediante una función personalizada implementada sobre operaciones de *TensorFlow*, lo que permitió adaptarla a las necesidades específicas del estudio. Por su parte, el MAE y el MAPE se calcularon utilizando las métricas (`MeanAbsoluteError()` y `MeanAbsolutePercentageError()`) de *tensorflow.keras.metrics*. El uso combinado de métricas personalizadas y métricas estándar de *Keras* permitió evaluar el rendimiento desde múltiples perspectivas, fortaleciendo la validez de la conclusión.

El proceso de entrenamiento de los *algoritmos de aprendizaje profundo* se configuró de manera uniforme con el fin de garantizar la comparabilidad de los resultados. Se empleó el *optimizador* `RMSprop` para gestionar tasas de aprendizaje adaptativas, utilizando el RMSE como *función de pérdida* para guiar el entrenamiento. Los *hiperparámetros de entrenamiento* se fijaron para ambos modelos de DL en 200 *épocas*, un *tamaño de lote* de 256 y una *tasa de aprendizaje* de 5×10^{-5}.

Como se acaba de comentar, los modelos se entrenaron de manera independiente para cada una de las 16 métricas de QoS, preservando la naturaleza univariante del enfoque experimental. Como resultado, se obtuvo un conjunto de modelos especializados por métrica, cada uno capaz de aprender los patrones particulares de su correspondiente serie temporal. Las arquitecturas de los modelos se detallan a continuación y su *representación* se muestra en la Figura 8.1. Las arquitecturas de los modelos de DL constan de *capas* de entrada, capas ocultas y capas de salida, en las que las *neuronas* se representan como unidades de procesamiento.

La arquitectura del modelo **BI-GRU** consta de una *capa de entrada* de 128 neuronas, que acepta una forma de entrada de 1×1, correspondiente a un único paso de tiempo por variable. A continuación, se incluyen dos *capas ocultas* GRU bidireccionales [16] con 128 y 64 neuronas, respectivamente. Estas capas procesan la secuencia de entrada tanto en dirección hacia adelante como hacia atrás, permitiendo capturar información contextual de la secuencia completa y contribuyendo, además, a la *regularización* del modelo. La arquitectura finaliza con una *capa de salida densa* [15] con una sola neurona, encargada de generar un valor continuo de regresión para cada secuencia de entrada.

La arquitectura del modelo **LSTM** inicia con una *capa de entrada* de 128 neuronas y la misma forma de entrada 1×1. A continuación, se apilan tres *capas ocultas* LSTM [16] que utilizan la *función de activación* tangente hiperbólica (*tanh*), intercaladas con una *capa dropout* para la *regularización* [177]. Posteriormente, se incorporan dos *capas densas* [15]. La primera capa LSTM contiene 128 neuronas, seguida de una capa *dropout* con una tasa de 0.1, que desactiva aleatoriamente el 10 % de las neuronas durante el entrenamiento para reducir el sobreajuste. La segunda y tercera capa LSTM cuentan con 128 y 64 neuronas, respectivamente. Las dos capas densas incluyen una primera capa con 16 neuronas y una función de activación ReLU, y una capa de salida final con una sola neurona y activación lineal, destinada a producir la predicción continua.

Los hiperparámetros se ajustaron mediante búsqueda aleatoria, explorando configuraciones de neuronas (32, 64, 128 y 256) y tasas de abandono (0.1, 0.5 y 0.8) [177]. Los mejores resultados se obtuvieron con 128/64 neuronas para BI-GRU y 128/128/64 para LSTM. Aunque también se probaron configuraciones con 256 neuronas, estas no mostra-

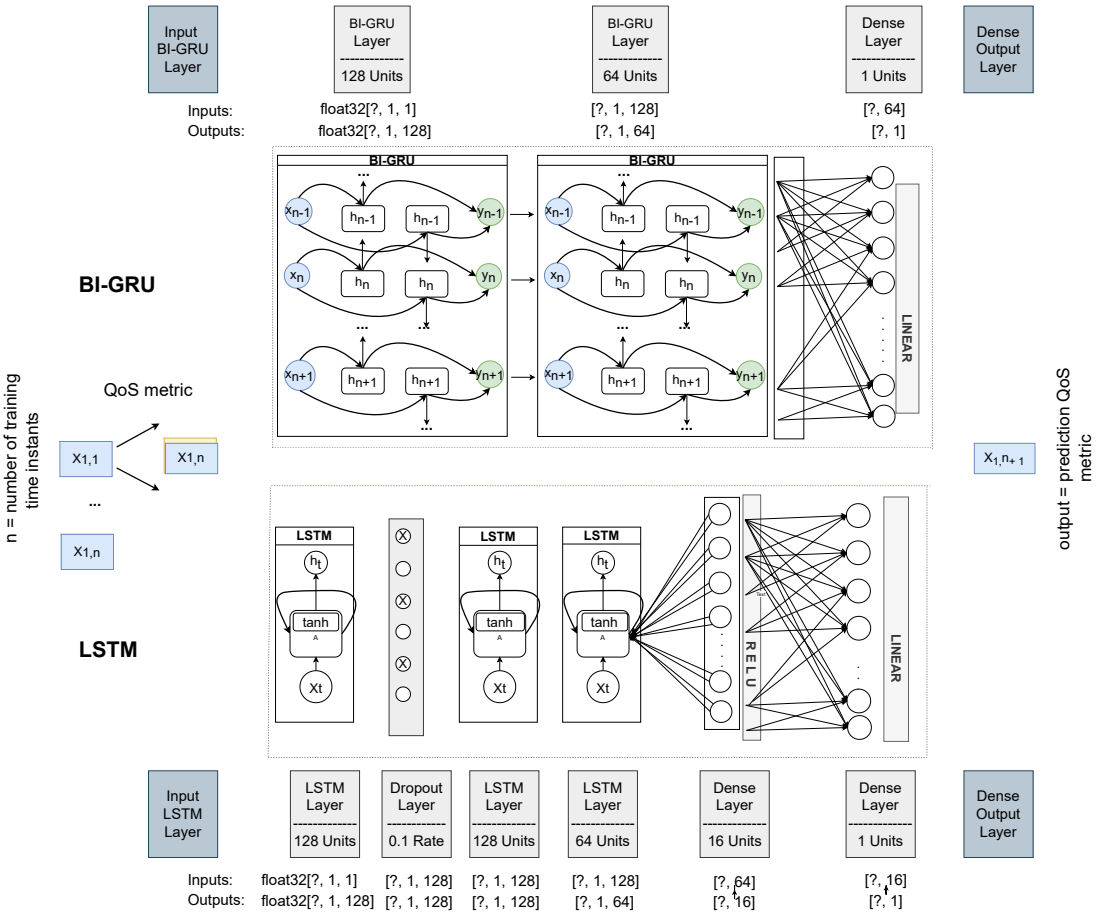


Figura 8.1: Arquitectura DL para modelos de pronóstico de QoS: el modelo BI-GRU (arriba) y el modelo LSTM (abajo).

ron mejoras notables en la precisión y sí implicaron un incremento considerable del coste computacional. La tasa de abandono de 0.1 mejoró la capacidad de generalización, mientras que valores de abandono más altos causaron una pérdida significativa de información. Por último, para las capas densas se exploraron configuraciones de 8, 16 y 32 neuronas, determinándose que la de 16 neuronas ofrecía el mejor equilibrio entre complejidad, rendimiento y coste computacional.

En cuanto a la validación cruzada, el conjunto de datos, compuesto por 6049 observaciones, se dividió en cuatro bloques (*folds*) equivalentes de 1512 observaciones cada uno. Esta decisión se fundamentó tanto en la longitud de la serie temporal como en las restricciones computacionales del entorno experimental. Los mismos cuatro bloques se utilizaron en los tres modelos evaluados (ARIMA, LSTM y BI-GRU), garantizando condiciones de comparación homogéneas. En consecuencia, se realizaron tres iteraciones: la primera iteración utilizó 3025 observaciones (1512 para entrenamiento), la segunda 4537 (3025 para

entrenamiento) y la tercera 6049 (4537 para entrenamiento). En todas las iteraciones, el conjunto de prueba mantuvo un tamaño fijo de 1512 observaciones. Posteriormente, se ejecutaron 10 réplicas independientes por configuración, dando lugar a un total de 480 ejecuciones.

Finalmente, los modelos ARIMA se implementaron utilizando la función `auto_arima` de la librería `pmdarima`, que selecciona automáticamente los parámetros (p, d, q). A diferencia de los modelos de DL, ARIMA no se beneficia de la aceleración por GPU; sin embargo, esta limitación no implicó una sobrecarga computacional significativa ni afectó a la escalabilidad del experimento.

Con el fin de garantizar la transparencia y la reproducibilidad del estudio, todos los recursos empleados—incluyendo el conjunto de datos, los *scripts* de preprocesamiento, los cuadernos de ejecución y los resultados obtenidos para cada una de las 16 métricas de QoS, están disponibles públicamente en el paquete de replicación correspondiente: https://bit.ly/frameworkCloudQuality. Esto facilitará la verificación independiente de los resultados, el análisis secundario y la adaptación del enfoque a otros contextos experimentales.

En conjunto, esta fase estableció un entorno experimental robusto, controlado y replicable, asegurando que la posterior evaluación de los modelos se base en una ejecución metodológicamente consistente y alineada con las guías de experimentación en ingeniería de software.

8.2.3.2. Procedimiento de análisis de datos

La validación empírica se estructuró mediante un enfoque de análisis riguroso, orientado a garantizar la solidez estadística de los resultados y a comparar de manera sistemática los tres modelos de pronóstico evaluados (BI-GRU, LSTM y ARIMA). Para ello, se adoptó una estrategia de análisis en dos fases que combinó estadística descriptiva e inferencial, en coherencia con los principios metodológicos para la experimentación en ingeniería de software propuestos por Wohlin et al. [36]. Todas las pruebas estadísticas se realizaron considerando un nivel de significancia del 5 %, lo que implica una probabilidad máxima de 0.05 de incurrir en un error tipo I; es decir, rechazar la hipótesis nula cuando esta es verdadera [178].

En la **primera fase**, se calcularon *estadísticas descriptivas*—media, desviación estándar, valores mínimo y máximo—para las métricas de QoS como para las métricas de desempeño (RMSE, MAE, MAPE y tiempo de entrenamiento) considerando todas las repeticiones y particiones temporales. Este análisis inicial permitió caracterizar las tendencias centrales, la dispersión y la presencia de posibles valores atípicos, proporcionando una visión exploratoria del comportamiento de los datos. Asimismo, facilitó la identificación de patrones preliminares y diferencias sistemáticas entre modelos que justificaron la aplicación de un análisis inferencial posterior. De este modo, esta fase contribuyó a alinear las observaciones empíricas con los objetivos de evaluación definidos (véase la Sección 8.2.1.1).

En la **segunda fase**, se aplicaron técnicas de *estadística inferencial* para contrastar

formalmente las hipótesis y las diferencias de desempeño entre los modelos de pronóstico. El *factor experimental* principal fue el tipo de modelo (BI-GRU, LSTM, ARIMA), mientras que las particiones generadas mediante validación cruzada *prequential* se incorporaron como una *variable de bloqueo*, con el propósito de controlar la variabilidad asociada a los distintos segmentos temporales. Los bloques (*folds*) fueron modelados como *efectos anidados*, preservando la validez del diseño experimental y evitando posibles sesgos derivados de la dependencia temporal inherente a los datos de series temporales.

Antes de seleccionar las pruebas estadísticas definitivas, se evaluaron los supuestos de normalidad y homogeneidad de varianzas mediante el test de *Shapiro–Wilk* y el test de *Brown–Forsythe Levene–type*, respectivamente. Cuando ambos supuestos se cumplieron, se aplicó un *Multivariate Analysis of Variance* (MANOVA) con interacciones, considerando simultáneamente el tipo de modelo y las métricas de desempeño como factores [36]. No obstante, dado que los datos multivariados de series temporales suelen violar estos supuestos, en los casos correspondientes se empleó *Permutational MANOVA* (PERMANOVA), un método no paramétrico robusto frente a desviaciones de normalidad y heterocedasticidad [179, 61]. En este último caso, el análisis se basó en una **matriz de distancias** construida a partir de las métricas de desempeño, la cual captura las disimilitudes multivariadas entre las observaciones y constituye el insumo fundamental para el cálculo de la estadística de pseudo-F y la evaluación de la significancia estadística mediante permutaciones [179].

Cuando PERMANOVA detectó efectos globales significativos, se interpretaron usando la estadística de pseudo-F y su correspondiente p-valor. Un valor alto de pseudo-F indica que la separación entre los modelos es considerable en relación con la variabilidad de sus resultados, lo que sugiere diferencias sustanciales en el rendimiento. Para establecer la existencia de un efecto global significativo, el p-valor debe ser menor que el nivel de significancia establecido ($\alpha = 0.05$), constituyendo así la primera evidencia estadística de que los modelos no presentan un desempeño equivalente en el conjunto de métricas evaluadas [179].

Una vez identificado un efecto global significativo, se realizaron comparaciones por pares mediante PERMANOVA entre modelos, aplicando un ajuste de p-valores mediante *Bonferroni*, lo que permitió identificar con precisión qué pares de modelos diferían entre sí y proporcionó información precisa sobre la magnitud y localización de tales diferencias. Para garantizan a su vez el control de la tasa de error familiar (*Family-Wise Error Rate*, FWER) asociada a múltiples comparaciones. Este procedimiento de dos pasos —detección global y localización por pares— proporcionó información precisa sobre la magnitud y localización de las diferencias de rendimiento, fortaleciendo la solidez estadística de las conclusiones alcanzadas.

Finalmente, para complementar la interpretación de los resultados inferenciales y facilitar la comprensión de la separación multivariada entre modelos, se implementó un **análisis de coordenadas principales** (*Principal Coordinates Analysis* - PCoA). Esta técnica proyecta la matriz de distancias (disimilitudes) en un espacio bidimensional, e incorporando elipses de confianza al 95 % alrededor de los conglomerados correspondientes a BI-GRU, LSTM y ARIMA. Estas visualizaciones no solo respaldaron la interpretación de los resultados inferenciales, sino que también permitieron observar la magnitud y dirección

de las diferencias de desempeño entre modelos evaluados.

8.3. Resultados

En esta sección se presentan y analizan los resultados del experimento controlado a partir del análisis cuantitativo de los datos y en función de las hipótesis planteadas. El análisis estadístico se llevó a cabo mediante un cuaderno `Jupyter` en `Python v3.12`, y se complementó con un análisis cualitativo preliminar basado en las observaciones del equipo técnico de la empresa participante.

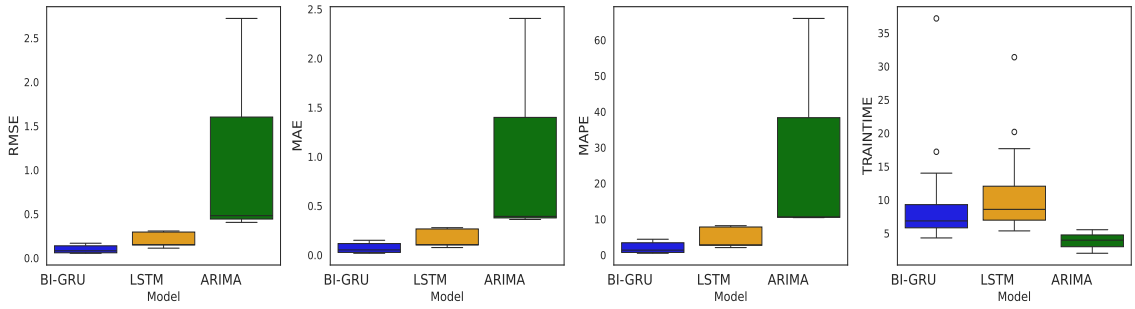
8.3.1. Estadística descriptiva y análisis exploratorio de datos

El análisis comparativo del rendimiento de los tres modelos de pronóstico (BI-GRU, LSTM y ARIMA) se realizó empleando métricas de precisión (RMSE, MAE, MAPE) y eficiencia (tiempo de entrenamiento). Para cada modelo, la Figura 8.2 muestra los diagramas de caja y bigotes de estas métricas de precisión y eficiencia para cada una de las 16 métricas de calidad de servicio (QoS), mientras que la Tabla 8.1 presenta las estadísticas descriptivas correspondientes.

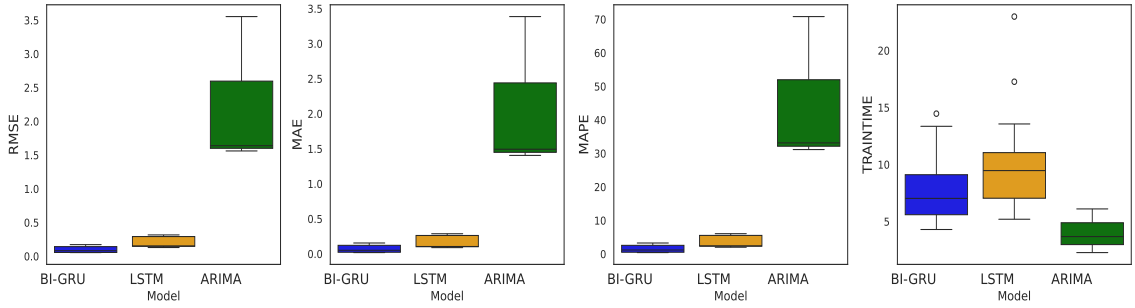
La Tabla 8.1 se organiza en tres bloques. En primer lugar, se incluyen las estadísticas descriptivas de las 16 métricas de calidad de servicio (QoS), donde se reportan la media, la desviación estándar (SD) y los valores mínimo y máximo observados en el conjunto de datos. En segundo lugar, se presentan las estadísticas descriptivas de las métricas de desempeño (RMSE, MAE, MAPE y tiempo de entrenamiento) obtenidas para cada modelo. En esta tabla, las columnas corresponden a las métricas de QoS, mientras que las filas representan las métricas de desempeño para cada modelo. Para cada métrica de desempeño se detallan su media, desviación estándar (SD) y valores mínimo y máximo. En conjunto, estas métricas permiten evaluar el rendimiento predictivo, al capturar tanto la magnitud del error como su variabilidad relativa. Para complementar la interpretación de la precisión de los modelos, la Tabla 8.1 incorpora además un tercer bloque con la ratio RMSE y el índice relativo de mejora predictiva (RPAI), propuesto por [23], que cuantifican la eficiencia y las ganancias relativas respecto al modelo base.

El objetivo principal de la *validación del modelo* es determinar la precisión alcanzada sobre datos no observados [168, 60]. Desde esta perspectiva, los resultados descriptivos confirman que BI-GRU obtuvo la media de RMSE más baja (valor mínimo de 0.0139 en QoS_{16}) en la mayoría de las métricas, superando tanto a LSTM como a ARIMA. Este último presentó sistemáticamente los valores más altos de RMSE, MAE y MAPE, lo cual pone de manifiesto las limitaciones de los modelos estadísticos tradicionales para capturar las dependencias no lineales y la heterogeneidad temporal características de las métricas de QoS en la nube.

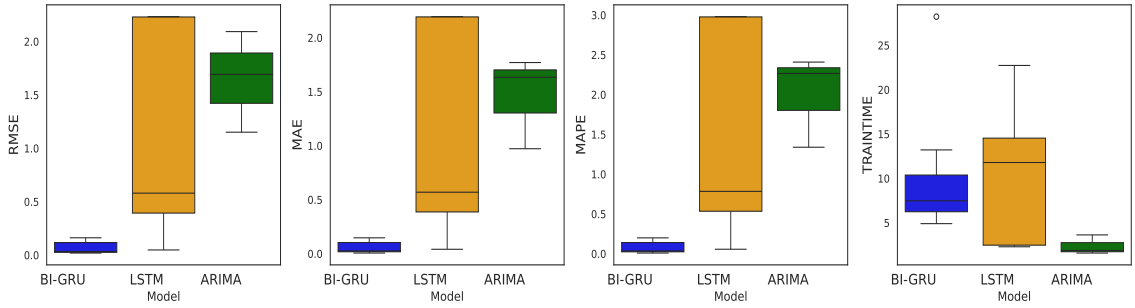
Un análisis detallado de la Tabla 8.1, centrado en los modelos de DL, muestra que el **RMSE promedio de LSTM fue inferior al de BI-GRU en seis de las dieciséis métricas**, en particular, aquellas relacionadas con el tráfico de red (*NetBytes In*, *NetBytes Out*, *NetPackets In*, *NetPackets Out*, *Rx Packets*, *Tx Packets*). Esto sugiere que LSTM



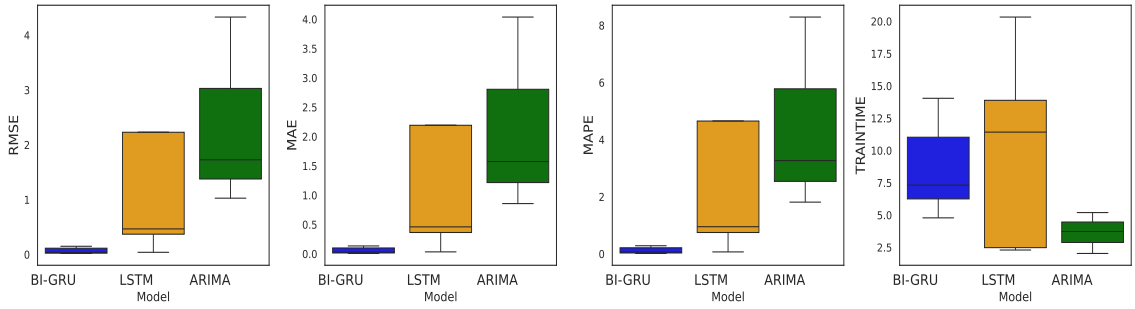
(a) QoS_1 *Free memory*



(b) QoS_2 *Used memory*

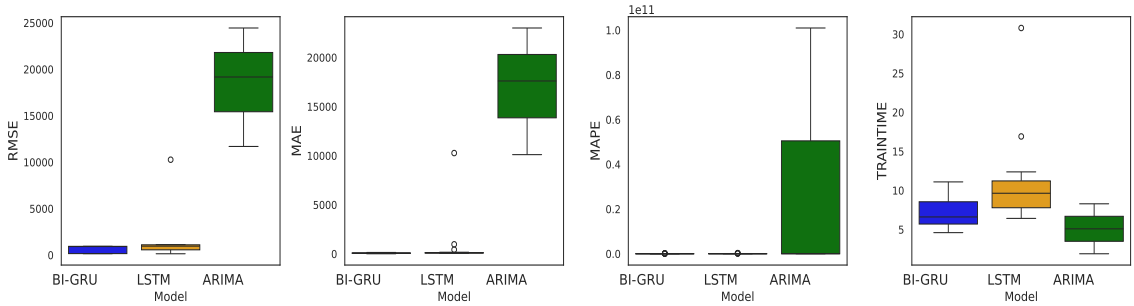


(c) QoS_3 *Free disk*

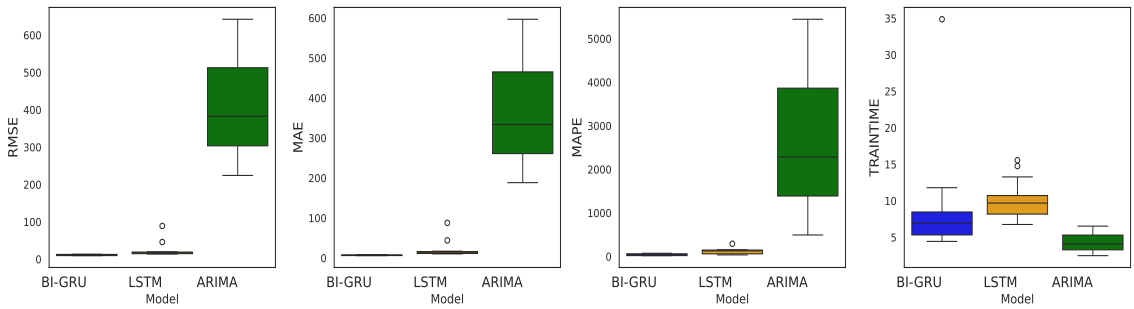


(d) QoS_4 *Used disk*

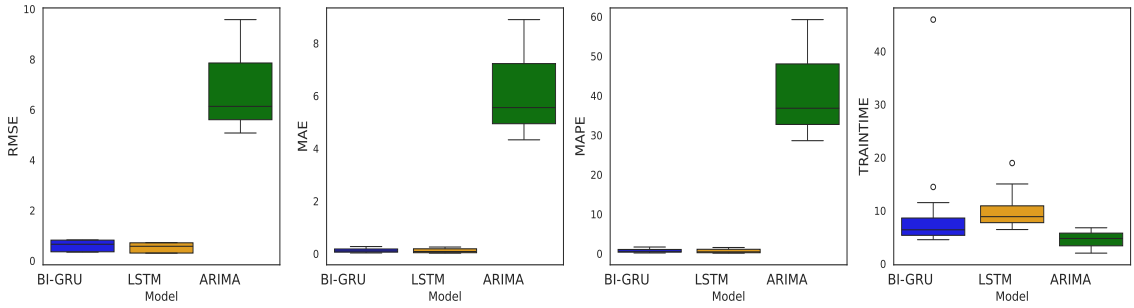
Figura 8.2: Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS1–4) — parte 1.



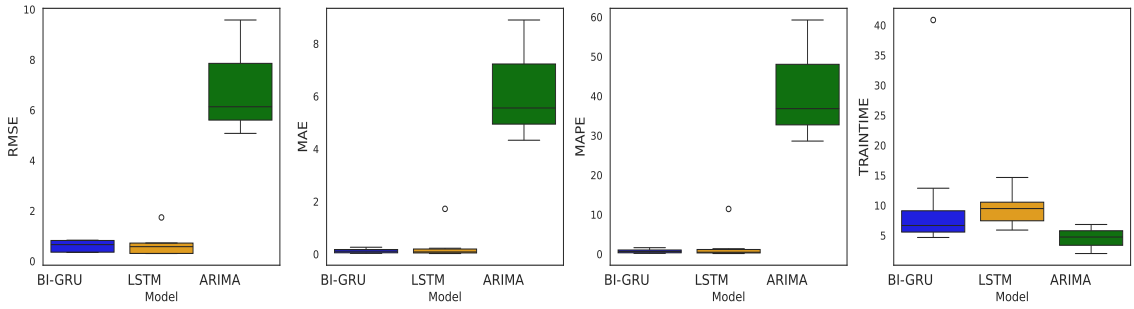
(e) QoS_5 *Read Disk*



(f) QoS_6 *Write disk*

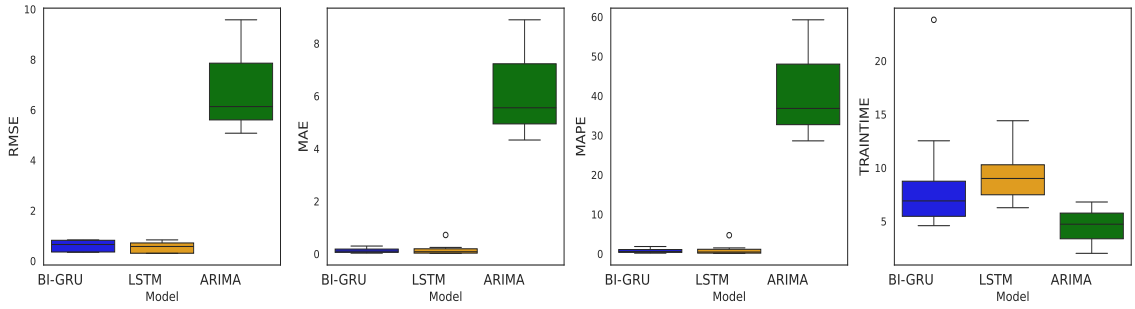


(g) QoS_7 *Netbytes in*

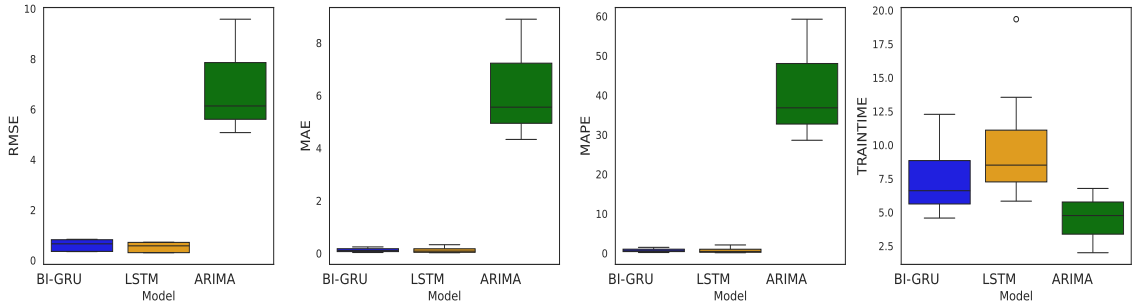


(h) QoS_8 *Netbytes out*

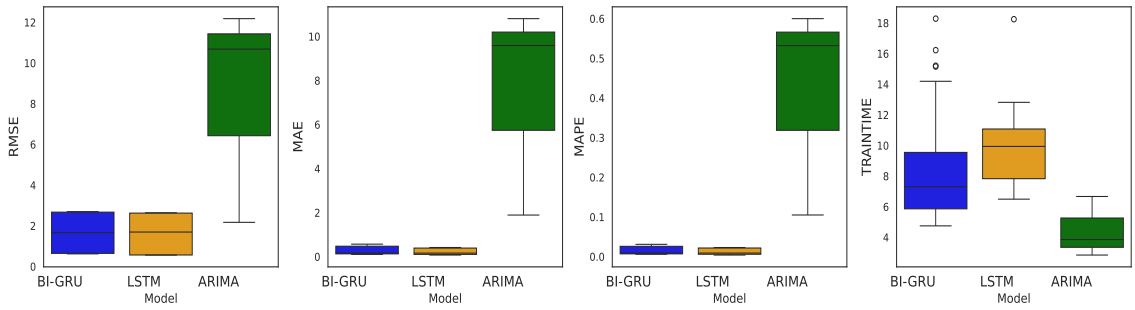
Figura 8.2: Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS5–8) — parte 2.



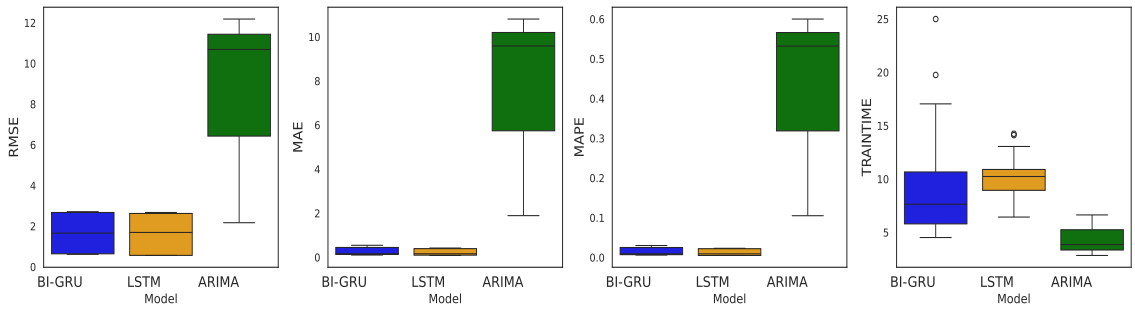
(i) QoS_9 *Netpackets in*



(j) QoS_{10} *Netpackets out*

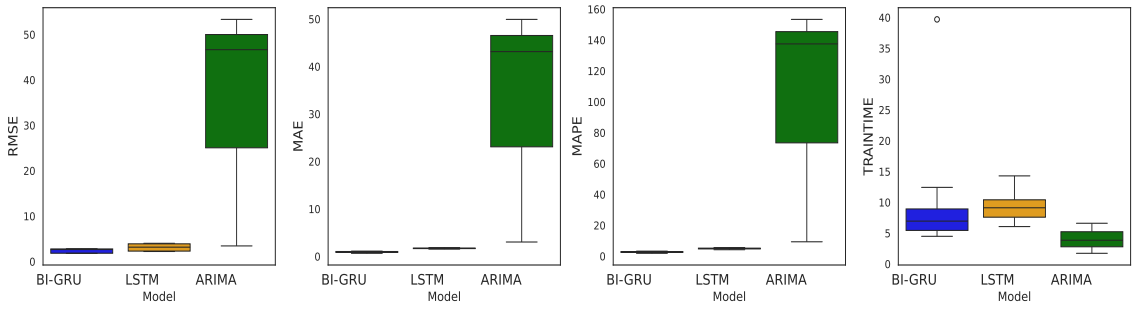


(k) QoS_{11} *Rx packets*

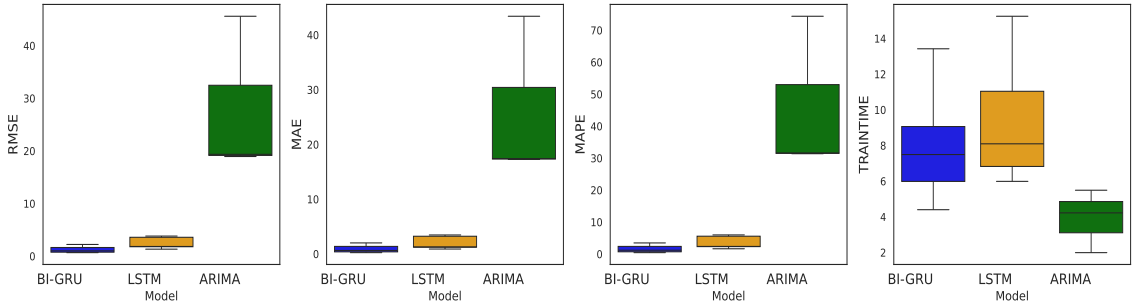


(l) QoS_{12} *Tx packets*

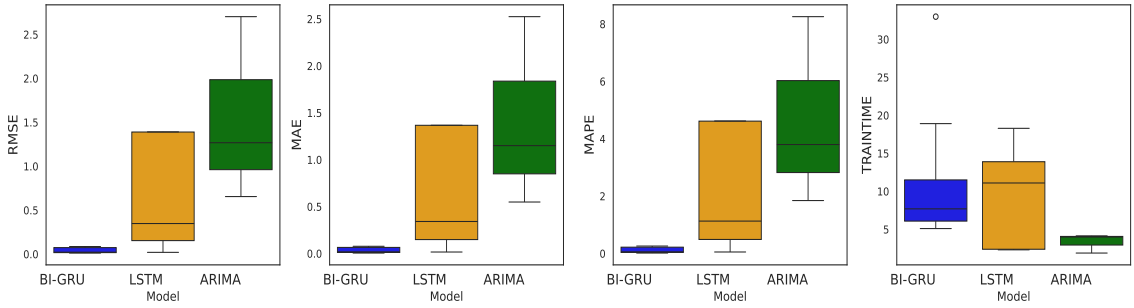
Figura 8.2: Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS9–12) — parte 3.



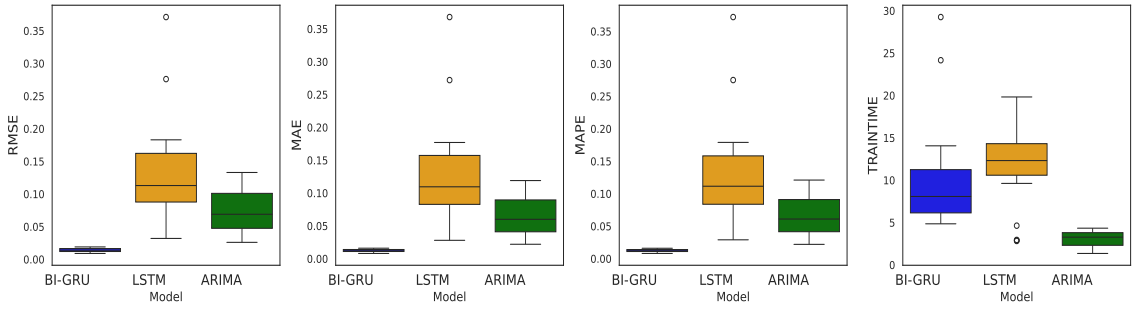
(m) QoS_{13} *CPU used %*



(n) QoS_{14} *Memory used %*



(ñ) QoS_{15} *Disk used %*



(o) QoS_{16} *Uptime ratio*

Figura 8.2: Diagramas de caja y bigotes de las métricas de rendimiento (RMSE, MAE, MAPE, tiempo de entrenamiento) para cada modelo de pronóstico. (QoS13–16) — parte 4.

puede capturar con mayor eficacia patrones de comunicación irregulares o direccionales. No obstante, **BI-GRU superó a LSTM en las diez métricas restantes**, especialmente en atributos relacionados con la utilización y disponibilidad de recursos, lo que evidencia una generalización más robusta frente a la heterogeneidad de las métricas de QoS. En consecuencia, BI-GRU se perfila como un enfoque más adecuado para el pronóstico multidimensional en entornos operativos.

Asimismo, la tabla 8.1 revela que las métricas *Disk Read* y *Disk Write* resultaron particularmente desafiantes para los tres modelos debido a su elevada variabilidad ($SD =$ 1772.22 y 22.77, respectivamente), producto de picos de carga esporádicos. Si bien ARIMA y LSTM mostraron un desempeño limitado ante estas fluctuaciones extremas, BI-GRU produjo errores comparativamente menores, reflejando mayor resiliencia frente a dinámicas altamente volátiles.

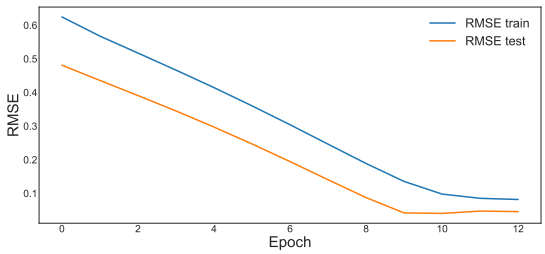
En términos de **eficiencia computacional**, los tiempos de entrenamiento de la Tabla 8.1 muestran que **ARIMA presentó la menor duración media** de entrenamiento (entre 2.9843 y 5.0857 segundos), acorde con su menor complejidad. Los modelos BI-GRU y LSTM requirieron tiempos superiores (7.2198–9.3967 y 9.1642–11.7850 segundos, respectivamente), lo que evidencia la mayor carga computacional de las arquitecturas DL. Sin embargo, esta diferencia se ve compensada por la mejora significativa en precisión—especialmente en el caso de BI-GRU—, sugiriendo una relación costo-beneficio favorable en escenarios productivos.

Una revisión detallada de los resultados del modelo BI-GRU confirma su desempeño sobresaliente: alcanzó valores de RMSE y MAPE inferiores a 1 en 10 de las 16 métricas (62.5 %) y valores de MAE inferiores a 1 en 12 métricas (75 %). Estos resultados respaldan empíricamente que BI-GRU ofrece los pronósticos más precisos y generalizables entre los enfoques evaluados.

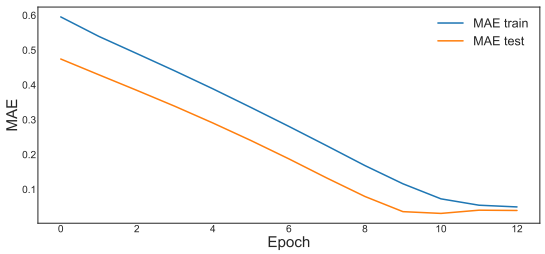
Con el fin de examinar la distribución de errores y detectar posibles valores atípicos, se generaron diagramas de caja y bigotes (*boxplots*) para las métricas de precisión y el tiempo de entrenamiento (Figura 8.2). Estas visualizaciones facilitan la comparación inmediata de la tendencia central, la variabilidad y la presencia de extremos entre los modelos. En particular, confirman nuevamente que ARIMA presenta los menores tiempos de entrenamiento, mientras que LSTM y BI-GRU incurren en mayores costos computacionales debido a su complejidad estructural, compensados, sin embargo, por mejoras sustanciales en la precisión.

Por su parte, la Figura 8.3 muestra las curvas de aprendizaje del modelo BI-GRU para la métrica *Memory Used %*, representando la evolución del error RMSE a través de las épocas. La curva azul representa el error en la precisión del entrenamiento, mientras que la curva naranja representa el error en la precisión de la prueba. Estas curvas revelan estabilidad durante el entrenamiento y una convergencia adecuada, lo que respalda de manera visual la ausencia de sobreajuste y confirma el correcto ajuste del modelo.

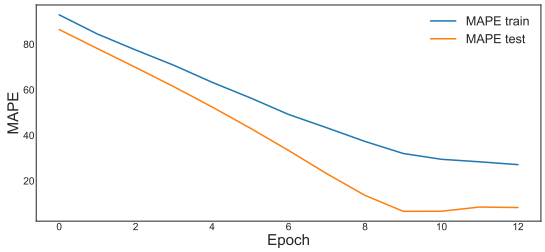
Finalmente, las métricas de comparación relativa—**Ratio RMSE** y **RPAI**—permiten evaluar de manera directa las mejoras obtenidas por cada modelo [23]. La comparación entre BI-GRU y ARIMA mostró ratios RMSE en el rango de 0.0249 a 0.1988, con valores de RPAI superiores al 80 %, confirmando la marcada superioridad de BI-GRU. Frente a



(a) Epochs vs RMSE loss

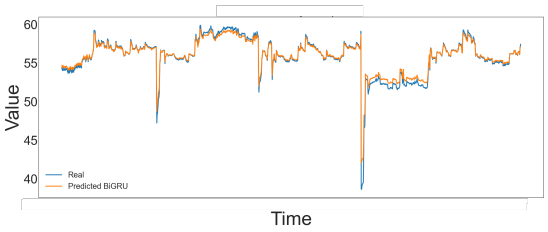


(b) Epochs vs MAE loss

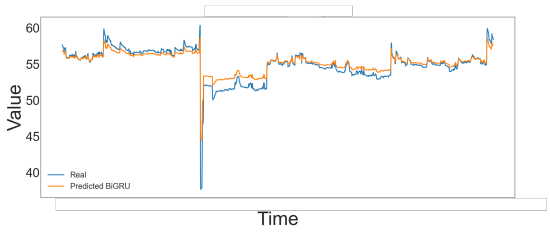


(c) Epochs vs MAPE loss

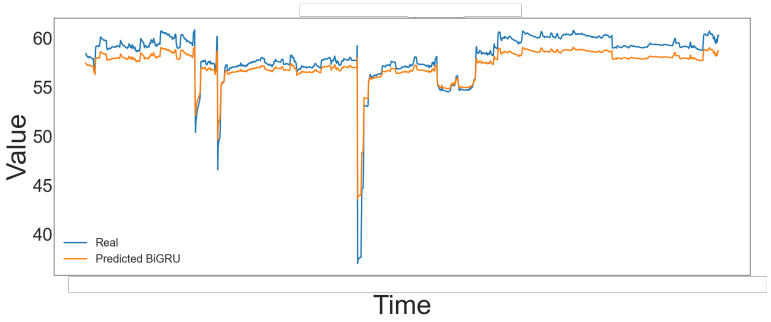
Figura 8.3: Convergencia del modelo BI-GRU durante el entrenamiento para la métrica *Memoria utilizada %* en la iteración final.



(a) Iteración 1



(b) Iteración 2



(c) Iteración 3

Figura 8.4: Pronóstico de series temporales con BI-GRU para la memoria utilizada % (*memory used %)*.

LSTM, BI-GRU obtuvo mejores resultados en 10 métricas (RPAI entre 22.57 % y 97.51 %), mientras que LSTM superó a BI-GRU en seis métricas, aunque con mejoras más moderadas (1.74 %–14.69 %).

En síntesis, los resultados de la Tabla 8.1 confirman la solidez y eficacia del modelo BI-GRU para el pronóstico de múltiples métricas de QoS bajo condiciones reales de operación en la nube, consolidándose como el modelo con el mejor desempeño global. La Figura 8.4 refuerza esta conclusión al mostrar que las predicciones de *Memoria utilizada %* siguen de cerca los valores observados a lo largo del tiempo, lo que evidencia la capacidad del modelo para capturar adecuadamente la dinámica temporal del servicio. La línea azul muestra los valores reales, mientras que la línea naranja muestra los valores predichos.

8.3.2. Prueba de hipótesis y discusión

Con el fin de evaluar las diferencias en el rendimiento predictivo entre los modelos, se contrastaron las hipótesis nulas ($H1_0$ y $H2_0$) mediante una comparación estadística de las métricas de desempeño (RMSE, MAE, MAPE y tiempo de entrenamiento) entre los tres enfoques considerados.

Tabla 8.1: Estadísticas descriptivas del rendimiento de los modelos de pronóstico.

	Métricas de QoS															
	QoS_1	QoS_2	QoS_3	QoS_4	QoS_5	QoS_6	QoS_7	QoS_8	QoS_9	QoS_{10}	QoS_{11}	QoS_{12}	QoS_{13}	QoS_{14}	QoS_{15}	QoS_{16}
Enfoque	Free Memory	Used Memory	Free Disk	Used Disk	Read Disk	Write Disk	NetBytes In	NetBytes Out	NetPackets In	NetPackets Out	Rx Packets	Tx Packets	CPU Used %	Memory Used %	Disk Used %	Uptime Ratio
mean	3.5870	4.6022	73.6012	47.1833	132.4224	31.6785	15.0463	15.0463	15.0463	15.0463	1800.0248	1800.0248	32.7965	56.1983	29.6438	99.1022
SD	0.2503	0.2504	1.4777	1.4777	1772.2175	22.7785	0.4959	0.4959	0.4959	0.4959	1.6960	1.6960	5.0994	3.0571	0.9236	0.1851
min	3.2000	2.6700	71.7400	43.7300	0.0000	4.9300	15.0000	15.0000	15.0000	15.0000	1785.0000	1785.0000	27.8400	32.6000	27.4900	98.7824
max	5.5200	4.9900	77.0500	49.0500	31407.68	781.9100	25.0000	25.0000	25.0000	25.0000	1865.0000	1865.0000	99.9900	60.9600	30.8100	99.4234
ARIMA																
RMSE	1.2030	2.2510	1.6453	2.3613	18437.19	416.5607	6.9280	6.9280	6.9280	6.9280	8.3457	8.3457	34.4807	27.9617	1.5420	0.0760
MAE	1.0537	2.0947	1.4587	2.1560	16906.029	372.9423	6.2620	6.2620	6.2620	6.2620	7.4287	7.4287	32.0373	26.0310	1.4080	0.0670
MAPE	29.0330	45.0360	2.0047	4.4567	33.655E9	2740.6014	41.5553	41.5553	41.5553	41.5553	0.4123	0.4123	100.0853	45.7907	4.6340	0.0680
Training time	3.8380	4.0157	2.3660	3.6513	5.0857	4.3690	4.5157	4.5150	4.5113	4.5120	4.4557	4.4450	4.0617	3.9010	3.3333	2.9843
LSTM																
RMSE																
mean	0.2000	0.2048	1.0257	0.9996	1115.8415	20.3437	**0.5322**	**0.5709**	**0.5408**	**0.5334**	**1.6285**	**1.6304**	3.1218	2.3763	0.6166	0.1309
SD	0.0730	0.0733	0.8633	0.8788	1739.5089	13.9144	0.1707	0.2756	0.1791	0.1711	0.8442	0.8464	0.6754	0.9271	0.5584	0.0666
min	0.1120	0.1290	0.0480	0.0400	139.672	13.909	0.3030	0.3020	0.3030	0.3030	0.5620	0.5610	2.2120	1.3300	0.0200	0.0320
max	0.3070	0.3160	2.2330	2.2310	10267.526	89.027	0.7210	1.7350	0.8370	0.7190	2.6390	2.6740	4.0160	3.8170	1.3910	0.3710
MAE																
mean	0.1574	0.1620	1.0285	1.0023	489.3614	17.0929	0.1130	0.1653	0.1296	0.1181	0.2148	0.2124	1.6998	1.8474	0.6260	0.1272
SD	0.0780	0.0786	0.8564	0.8722	1855.5789	14.6581	0.0762	0.3019	0.1342	0.0804	0.1266	0.1297	0.0724	0.9768	0.5476	0.0675
min	0.0760	0.0890	0.0410	0.0340	60.7190	10.3190	0.0250	0.0260	0.0230	0.0230	0.0740	0.0840	1.5440	0.8920	0.0490	0.0280
max	0.2780	0.2870	2.1950	2.1940	10266.7959	87.8350	0.2550	1.7220	0.7200	0.3280	0.4170	0.4200	1.8580	3.4250	1.3670	0.3680
MAPE																
mean	4.4559	3.5338	1.4024	2.1139	39.828E6	115.7309	0.6593	1.0091	0.7693	0.6933	0.0119	0.0117	4.989	3.3056	2.1018	0.1286
SD	2.4069	1.5894	1.1587	1.8598	76.173E6	56.0506	0.4639	2.0087	0.8714	0.5009	0.0070	0.0072	0.2928	1.6172	1.8572	0.0681
min	2.0710	2.0130	0.0570	0.0690	289.1340	37.3870	0.1400	0.1440	0.1270	0.1230	0.0040	0.0050	4.2140	1.6490	0.1600	0.0290
max	8.1820	6.0560	2.9800	4.6550	30.865E7	293.4440	1.5550	11.4160	4.7100	2.0920	0.0230	0.0230	5.5850	5.9200	4.6190	0.3720
Training time																
mean	10.3248	9.9463	9.9998	9.8331	9.5800	9.7099	9.6024	9.4541	9.0195	9.3115	9.8022	9.9503	9.1647	9.1642	9.4428	11.7850
SD	5.3185	3.6831	5.9380	5.6014	2.2651	1.9201	2.7910	2.4079	2.1801	2.8192	2.3646	1.9449	2.0045	2.7081	5.6704	4.2314
min	5.3590	5.1970	2.2940	2.2960	6.4180	6.7600	6.4460	5.8940	6.2690	5.8310	6.4930	6.4360	6.0670	5.9880	2.3010	2.8240
max	31.3760	22.9780	22.7320	20.3300	16.9090	14.7600	18.9680	14.6490	14.4050	19.3430	18.2440	14.2410	14.2950	15.2450	18.2880	19.8200

Tabla 8.1 (continúa)

Enfoque	Métrica de QoS															
	QoS_1	QoS_2	QoS_3	QoS_4	QoS_5	QoS_6	QoS_7	QoS_8	QoS_9	QoS_{10}	QoS_{11}	QoS_{12}	QoS_{13}	QoS_{14}	QoS_{15}	QoS_{16}
BI-GRU																
RMSE																
mean	**0.0963**	**0.0965**	**0.0635**	**0.0615**	**669.6907**	**11.6651**	0.6104	0.6104	0.6104	0.6111	1.6601	1.6588	**2.4171**	**1.1967**	**0.0384**	**0.0139**
SD	0.0390	0.0414	0.0492	0.0452	365.1343	1.5013	0.1944	0.1944	0.1950	0.1951	0.8340	0.8353	0.4246	0.5164	0.0277	0.0029
min	0.0540	0.0540	0.0190	0.0190	149.4810	9.4960	0.3440	0.3460	0.3450	0.3480	0.6200	0.6160	1.8130	0.6670	0.0120	0.0090
max	0.1680	0.1720	0.1620	0.1500	942.7190	13.3250	0.8270	0.8310	0.8350	0.8360	2.6960	2.7120	2.8230	2.2260	0.0860	0.0190
MAE																
mean	0.0680	0.0683	0.0534	0.0516	68.7575	6.9762	0.1179	0.1194	0.1241	0.1176	0.2600	0.2520	0.9398	0.8533	0.0321	0.0122
SD	0.0445	0.0472	0.0464	0.0427	32.1721	0.8075	0.0641	0.0672	0.0738	0.0667	0.1666	0.1592	0.1194	0.5869	0.0260	0.0021
min	0.0180	0.0180	0.0060	0.0070	9.7030	5.8860	0.0300	0.0300	0.0290	0.0310	0.1040	0.1030	0.6600	0.2310	0.0050	0.0080
max	0.1500	0.1550	0.1470	0.1350	143.1380	8.3570	0.2730	0.2630	0.2990	0.2440	0.5690	0.5450	1.1170	2.0090	0.0770	0.0160
MAPE																
mean	1.9236	1.4929	0.0729	0.1083	2.0213E7	41.5108	0.6299	0.6324	0.6567	0.6294	0.0420	0.0411	2.4637	1.4511	0.0984	0.0122
SD	1.3441	0.9748	0.0629	0.0909	42.724E6	19.5507	0.3826	0.4036	0.4486	0.4007	0.0091	0.0088	0.3711	0.9917	0.0880	0.0021
min	0.4870	0.4140	0.0090	0.0150	877.7070	19.7660	0.1700	0.1700	0.1660	0.1760	0.0060	0.0060	1.8850	0.4360	0.0160	0.0080
max	4.3990	3.2750	0.2000	0.2850	18.370E7	74.9950	1.6770	1.6120	1.8480	1.4840	0.0310	0.0300	3.2840	3.4740	0.2600	0.0160
Training time																
mean	8.4858	7.5739	8.6937	8.4278	7.2245	7.7096	8.0844	8.1434	7.5971	7.2198	8.4436	8.8422	8.0843	7.3268	9.373	9.3967
SD	5.4883	2.6333	4.0031	2.6860	1.9276	4.7971	6.5362	5.7314	3.3500	2.1348	3.4010	4.2611	5.5467	2.3509	5.0727	4.8880
min	4.3050	4.2960	4.9010	4.7850	4.5940	4.4410	4.5350	4.6600	4.5900	4.5750	4.7490	4.5180	4.4880	4.3990	5.0980	4.8480
max	37.1820	14.4610	28.2310	14.0410	11.0900	34.8650	45.9700	40.8670	23.8440	12.2790	18.2810	24.9910	39.6950	13.4210	32.9830	29.2510
RMSE ratio[1]																
BI-GRU vs. ARIMA	0.0800	0.0429	0.0386	0.0260	0.0363	0.0280	0.0881	0.0881	0.0881	0.0882	0.1989	0.1988	0.0701	0.0428	0.0249	0.1829
BI-GRU vs. LSTM	0.4815	0.4712	0.0619	0.0615	0.6002	0.5734	1.1469	1.0692	1.1287	1.1457	1.0194	1.0174	0.7743	0.5036	0.0623	0.1062
Relative Prediction Accuracy Improvement[2] **(RPAI) %**																
BI-GRU vs. ARIMA	**92.00**	**95.71**	**96.14**	**97.40**	**96.37**	**97.20**	**91.19**	**91.19**	**91.19**	**91.18**	**80.11**	**80.12**	**92.99**	**95.72**	**97.51**	**81.71**
BI-GRU vs. LSTM	**51.85**	**52.88**	**93.81**	**93.85**	**39.98**	**42.66**	-14.69	-6.92	-12.87	-14.57	-1.94	-1.74	**22.57**	**49.64**	**93.77**	**89.38**

SD: standard deviation, min: minimum value, max: maximum value
[1] RMSE ratio = $RMSE_{BI-GRU}$ / $RMSE_{[othermethod]}$
[2] RPAI = ($RMSE_{[othermethod]}$ - $RMSE_{BI-GRU}$) / $RMSE_{[othermethod]}$

Dado que las condiciones de normalidad y homogeneidad de varianzas no se cumplieron de manera consistente, se optó por utilizar PERMANOVA, una alternativa no paramétrica basada en matrices de distancia que no exige supuestos de normalidad. El análisis se implementó en Python mediante la función `skbio.stats.permanova` del paquete `scikit-bio`. En primer lugar, se construyó la matriz de distancias euclidianas a partir de las variables dependientes, en este caso, RMSE, MAE, MAPE y tiempo de entrenamiento, para posteriormente aplicar PERMANOVA y evaluar la separación entre grupos.

La tabla 8.2 presenta los resultados inferenciales. Por ejemplo, el PERMANOVA global para la métrica QoS_1, basado en 999 permutaciones aleatorias, arrojó un valor de pseudo-F de 20.521 y un p-valor de 0.001, lo que indica que la variación entre modelos es sustancialmente mayor que la variación dentro de cada modelo.

El uso de 999 permutaciones responde a una práctica ampliamente aceptada en análisis no paramétricos [179], ya que proporciona una estimación estable de la distribución nula y permite alcanzar una resolución mínima del p-valor de 0.001, adecuada para contrastes estadísticos con un nivel de significancia del 5 % [61]. Este resultado proporciona evidencia sólida para rechazar la hipótesis nula y confirma que los enfoques difieren significativamente en su rendimiento conjunto, considerando simultáneamente las cuatro métricas de precisión y eficiencia.

A continuación, se realizó el análisis *PERMANOVA por pares* para identificar qué combinaciones de modelos explican las diferencias observadas. Se aplicó una corrección de *Bonferroni* para ajustar los p-valores debido a la realización de múltiples comparaciones. La estadística pseudo-F refleja la relación entre la variación entre grupos y la variación dentro de los grupos, donde los valores más altos indican una mayor separación. Los valores p-ajustados determinan si dicha separación es estadísticamente significativa.

La tabla 8.2 resume los resultados de estas comparaciones por pares—BI-GRU vs. LSTM, BI-GRU vs. ARIMA y LSTM vs. ARIMA—especificando para cada combinación los valores de pseudo-F, los p-valores originales y los ajustados (p-adj). En términos generales, los resultados pueden sintetizarse así:

- **BI-GRU vs LSTM**: Los valores pseudo-F muestran diferencias moderadas (p. ej., el mínimo de 1.415 para QoS_8, 5.547 en $\text{QoS}_1 4$) y otras más marcadas (31.064 en QoS_3 y el máximo de 66.327 en $\text{QoS}_1 6$). Varias métricas, incluida $\text{QoS}_1 4$, presentan valores p-adj inferiores a 0.05, lo que indica que, aunque BI-GRU y LSTM comparten capacidades similares, existen diferencias estadísticamente significativas en su rendimiento. Es decir, la arquitectura influye de forma observable en los pronósticos.

- **BI-GRU vs ARIMA**: La distancia entre ambos modelos es consistentemente alta, con valores de pseudo-F donde el mínimo en QoS_{16} es 9.778 y el máximo en QoS_2 es 117.711. Los valores p-adj permanecen por debajo de 0.05 en la mayoría de métricas de QoS, lo que confirma una diferencia robusta entre ARIMA y los modelos de BI-GRU. Estos resultados refuerzan las limitaciones de ARIMA para capturar dependencias no lineales complejas que sí son modeladas eficazmente por BI-GRU.

Tabla 8.2: Análisis estadístico inferencial del rendimiento de los modelos de pronóstico.

	Métricas de QoS															
	QoS_1	QoS_2	QoS_3	QoS_4	QoS_5	QoS_6	QoS_7	QoS_8	QoS_9	QoS_{10}	QoS_{11}	QoS_{12}	QoS_{13}	QoS_{14}	QoS_{15}	QoS_{16}
Prueba de hipótesis	Free Memory	Used Memory	Free Disk	Used Disk	Read Disk	Write Disk	NetBytes In	NetBytes Out	NetPackets In	NetPackets Out	Rx Packets	Tx Packets	CPU Used %	Memory Used %	Disk Used %	Uptime Ratio
PERMANOVA																
p-value: 0.001																
número de permutaciones: 999																
pseudo-F	20.521	66.993	20.008	19.665	41.145	49.954	75.959	72.358	82.008	**93.611**	41.754	40.387	37.424	53.539	**19.010**	35.013
Comparación por pares																
BI-GRU vs LSTM																
pseudo-F	6.530	11.598	31.064	26.588	12.140	5.740	1.524	**1.415**	4.565	12.959	3.108	1.545	1.605	5.547	23.836	**66.327**
p-value	0.002	0.004	0.001	0.001	0.001	0.004	0.232	0.276	0.029	0.001	0.072	0.216	0.238	0.009	0.001	0.001
p-adj	0.006	0.012	0.003	0.003	0.003	0.012	0.696	0.828	0.087	0.003	0.216	0.648	0.714	0.027	0.003	0.003
BI-GRU vs ARIMA																
pseudo-F	28.612	117.711	54.106	58.701	76.635	60.382	93.324	92.942	112.111	169.072	55.275	50.349	44.299	78.708	43.719	9.778
p-value	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.002	0.011
p-adj	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.006	0.003	0.003	0.003	0.003	0.003	0.006	0.033
LSTM vs ARIMA																
pseudo-F	19.226	69.619	2.018	4.204	35.319	75.953	200.948	176.003	154.581	112.043	60.413	67.002	55.564	67.461	4.819	4.276
p-value	0.001	0.002	0.138	0.026	0.001	0.001	0.002	0.001	0.001	0.001	0.002	0.001	0.001	0.001	0.007	0.04
p-adj	0.003	0.006	0.414	0.078	0.003	0.003	0.006	0.003	0.003	0.003	0.006	0.003	0.003	0.003	0.021	0.12
Análisis de Coordenadas Principales (PCoA)																
PC1 %	75.97	76.62	**82.27**	81.46	**60.69**	74.58	75.40	75.33	76.10	76.39	74.47	74.61	75.40	78.32	81.82	75.00
PC2 %	23.96	23.36	**17.69**	18.50	24.52	24.40	24.44	24.48	23.73	23.44	22.92	22.78	24.49	21.66	18.15	**24.99**
(PC1+PC2) %	99.94	99.98	99.96	99.97	85.22	98.98	99.84	99.81	99.83	99.83	97.39	97.39	99.89	99.99	99.97	99.99

- **LSTM vs ARIMA**: Los resultados son similares, con valores pseudo-F elevados, donde el mínimo en QoS_3 es 2.018 y el máximo en QoS_9) es 154.581 y valores p-adj consistentemente inferiores a 0.05. En consecuencia, LSTM y ARIMA presentan diferencias notables en el rendimiento de sus pronósticos.

En conjunto, el análisis por pares confirma que todos los modelos difieren significativamente entre sí. Si bien BI-GRU y LSTM presentan un rendimiento cercano, siguen siendo estadísticamente diferentes, y ambos modelos son notablemente superiores a ARIMA. Estos resultados permiten rechazar las hipótesis $H1_0$ y $H2_0$, validando que la arquitectura del modelo influye significativamente en la precisión y la eficiencia de la predicción.

Por último, la Tabla 8.2 también muestra la varianza explicada por el Análisis de Coordenadas Principales (PCoA). El primer componente principal (PC1) captura la mayor parte de la variación en la matriz de distancias—entre el 60.69 % y el 82.27 % según las métricas de QoS—, mientras que el segundo componente (PC2) explica entre el 17.69 % y el 24.99 % adicionales. En conjunto, PC1 y PC2 explican más del 99 % de la variación en la mayoría de los casos, lo que indica que una representación bidimensional es suficiente para visualizar las relaciones multivariadas entre los modelos.

La Figura 8.5 complementa los resultados inferenciales presentados en la Tabla 8.2, al ofrecer una representación visual del comportamiento de los tres modelos evaluados según sus coordenadas principales. Los modelos se representan en azul para BI-GRU, naranja para LSTM y verde para ARIMA. Por ejemplo, en la Figura 8.5.(n), correspondiente a QoS_{14}, el componente principal PC1 explica el 78.32 % de la variación y PC2 explica el 21.66 %, alcanzando un total del 99.99 %. Cada punto en la figura corresponde a una muestra experimental, ubicada según su similitud en el espacio de las métricas de desempeño, mientras que las elipses de confianza al 95 % sintetizan la variabilidad interna y la dispersión de cada modelo. La separación de los clústeres indica diferencias estadísticamente significativas, mientras que su superposición sugiere diferencias más débiles. Los valores pseudo-F cuantifican directamente la magnitud del efecto asociado a estas separaciones.

La visualización evidencia que BI-GRU y LSTM presentan clústeres parcialmente solapados, lo que sugiere desempeños similares con variaciones relevantes principalmente en el eje PC2. En contraste, ARIMA forma un clúster claramente diferenciado, separado principalmente en el eje PC1, que concentra la mayor parte de la variación. Este patrón confirma que ARIMA diverge sustancialmente de los modelos DL, lo que respalda la hipótesis sobre su limitada capacidad para capturar dependencias temporales complejas. Por su parte, las diferencias entre BI-GRU y LSTM son más sutiles pero aún relevantes, en línea con los resultados inferenciales.

En conjunto, los resultados muestran que BI-GRU ofrece una ventaja competitiva en el pronóstico de QoS en servicios cloud, combinando un alto desempeño con menor complejidad y tiempos de entrenamiento más cortos respecto de LSTM. La diferencia significativa observada frente a ARIMA resalta sus limitaciones en escenarios caracterizados por alta variabilidad y relaciones temporales no lineales.

Además, la capacidad de BI-GRU para capturar interdependencias entre atributos de QoS proporciona información útil sobre el comportamiento del servicio, lo cual puede

orientar la toma de decisiones para mejorar la confiabilidad y resiliencia del servicio.

Finalmente, los experimentos controlados demuestran la solidez metodológica y la aplicabilidad práctica del enfoque; la metodología propuesta ofrece lineamientos claros para su replicación y adaptación en distintos entornos cloud, incluyendo procedimientos para el rentreenamiento, la validación y la selección de atributos.

8.3.3. Retroalimentación preliminar

Con el fin de evaluar la aplicabilidad práctica y el valor percibido de los modelos BI-GRU y LSTM para el pronóstico de la calidad del servicio SAlert, se llevó a cabo una sesión de retroalimentación con la participación del equipo de la empresa, conformado por tres profesionales: el gerente de TI y dos analistas responsables de la monitorización del servicio cloud. La discusión se centró en dos objetivos principales:

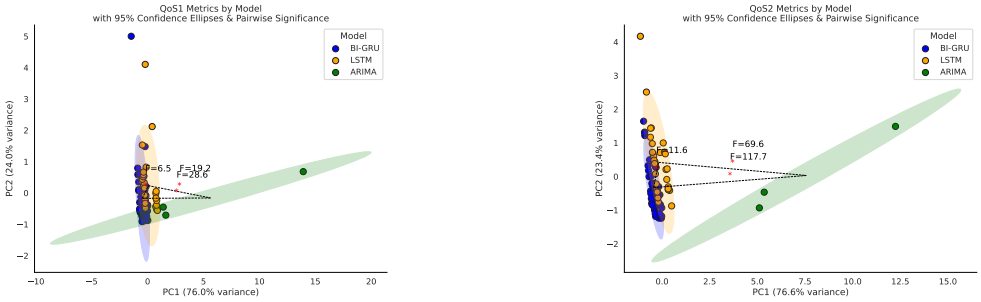
1. Identificar tres aspectos positivos y negativos del modelo BI-GRU observados durante su uso; y
2. Valorar la reproducibilidad del proceso de modelado para su aplicación en otros contextos de servicios cloud sin asistencia directa del equipo de investigación.

La retroalimentación positiva destaca la capacidad de BI-GRU para pronosticar con precisión métricas de QoS, aportando información útil para la toma de decisiones dentro de la empresa. El modelo respaldó las prácticas de gestión de datos, facilitó la optimización del rendimiento del servicio, contribuyó al cumplimiento de los SLA y mejoró la experiencia de los usuarios. Su capacidad para anticipar posibles degradaciones de QoS permitió actuar de forma temprana, reduciendo riesgos y favoreciendo una planificación operativa alineada con la estrategia del servicio.

Asimismo, se valoró la definición integral de QoS adoptada en el modelo, que incorpora múltiples atributos y sus métricas. Esta perspectiva holística proporcionó una visión completa del estado del servicio y permitió una gestión más proactiva del rendimiento y la fiabilidad. Los profesionales señalaron que BI-GRU permitió pronosticar con éxito la mayoría de los picos de demanda, facilitando la asignación óptima de recursos y configuraciones adecuadas de balanceo de carga para mantener niveles aceptables de calidad. Del mismo modo, la detección temprana de posibles violaciones de SLA resultó especialmente útil para priorizar y resolver incidencias críticas con mayor eficacia.

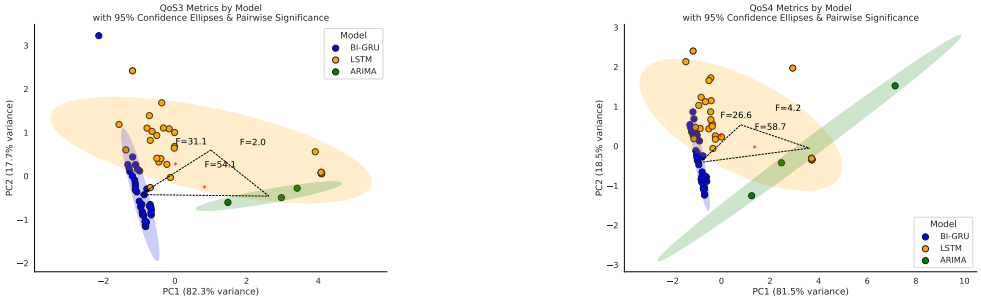
Además, el modelo BI-GRU mostró una notable capacidad para identificar relaciones latentes entre los atributos de QoS. Esta identificación de interdependencias entre métricas aportó información valiosa más allá de los pronósticos puntuales, favoreciendo la mejora continua del servicio y la planificación operativa a largo plazo.

A pesar de la valoración positiva general, también se identificaron algunas limitaciones. Una de ellas es el costo computacional del modelo BI-GRU. Los tiempos y recursos necesarios para el entrenamiento y la inferencia—particularmente con arquitecturas complejas o conjuntos de datos más extensos—pueden generar desafíos de escalabilidad e incrementar los costes operativos, especialmente en organizaciones con acceso limitado a infraestructuras de alto rendimiento.



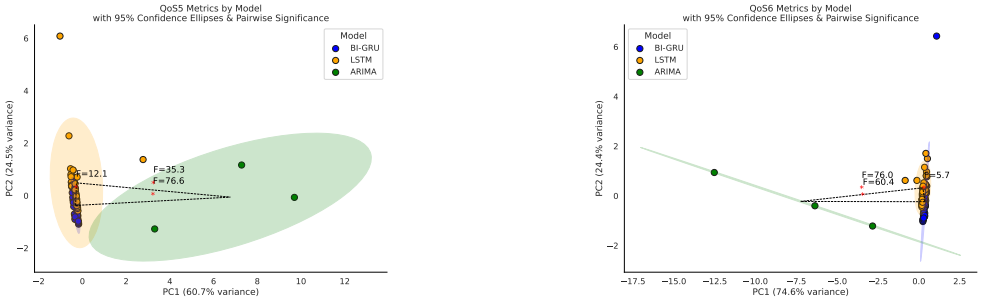
(a) QoS1

(b) QoS2



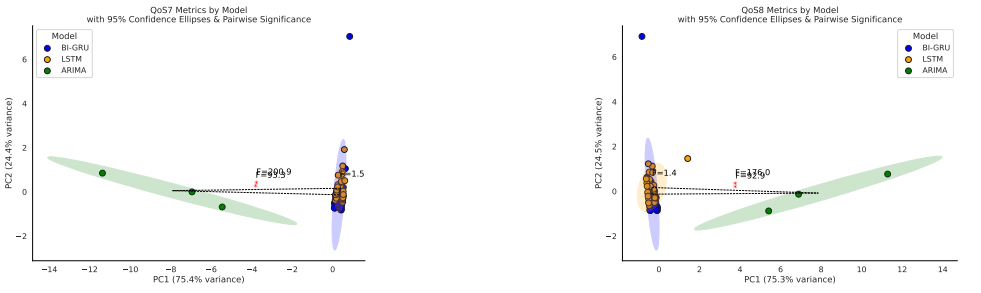
(c) QoS3

(d) QoS4



(e) QoS5

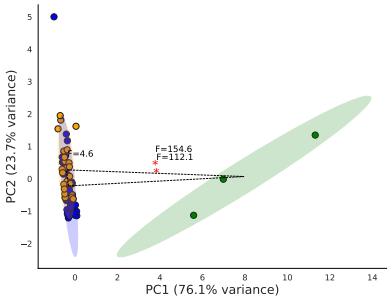
(f) QoS6



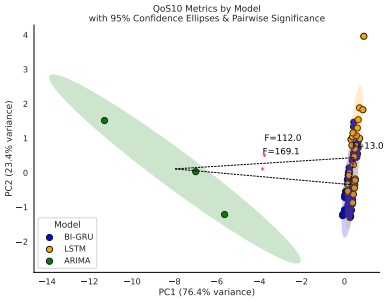
(g) QoS7

(h) QoS8

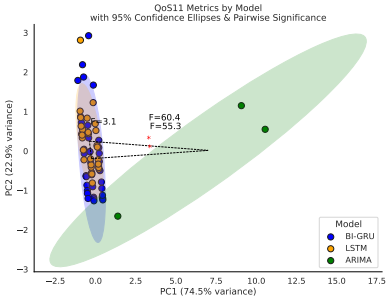
Figura 8.5: PCoA con elipses de confianza del 95 % y significación por pares. (1–8)—parte 1.



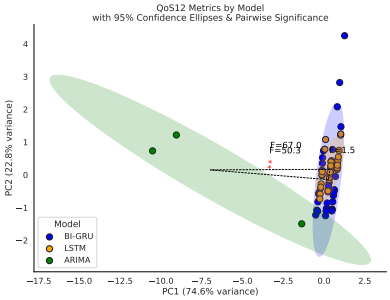
(i) QoS9



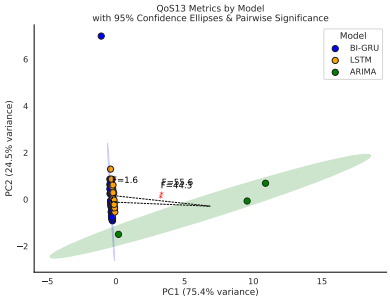
(j) QoS10



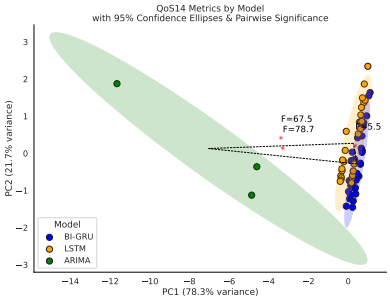
(k) QoS11



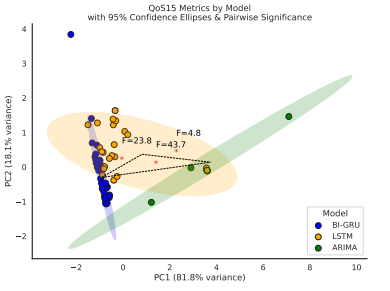
(l) QoS12



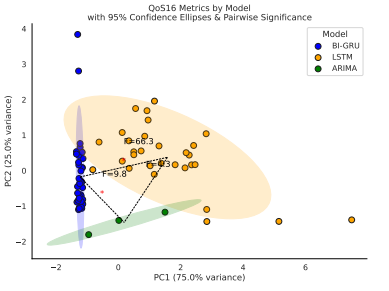
(m) QoS13



(n) QoS14



(ñ) QoS15



(o) QoS16

Figura 8.5: PCoA con elipses de confianza del 95 % y significación por pares. (9–16)—parte 2.

Otro aspecto señalado fue el corto plazo de predicción, dado que la implementación actual se basa en pronósticos de un solo paso. Esto restringe la capacidad para anticipar comportamientos a mediano o largo plazo. Sin embargo, se sugirió la incorporación futura de técnicas de iteración recursiva y enfoques *multi-step* para ampliar el horizonte de pronóstico, como las propuestas por Taieb et al. [180] y Bontempi et al. [63].

Por último, se recomendó mejorar la visualización e interoperabilidad de las salidas del modelo. En particular, se propuso representar las predicciones de QoS en relación con los umbrales definidos en los SLA, con el objetivo de potenciar la monitorización en tiempo real y la toma de decisiones operativas. Un mejor diseño de los formatos de reporte favorecería la adopción de estos modelos tanto por parte de perfiles técnicos como no técnicos.

Respecto a la reproducibilidad del proceso de modelado, los participantes expresaron su confianza en la claridad y exhaustividad del proceso propuesto. Consideraron que se proporciona el nivel de detalle suficiente para guiar la definición y validación de modelos de aprendizaje profundo—como BI-GRU—aplicados a otros servicios cloud de la empresa sin necesidad de apoyo externo.

8.4. Amenazas a la validez

Aunque el experimento se llevó a cabo de forma sistemática, es posible identificar algunos factores que podrían amenazar su validez. Esta sección analiza las principales amenazas siguiendo la clasificación de Wohlin et al. [36] e incorpora consideraciones específicas señaladas por Vegas y Elbaum [4] para experimentos de ingeniería de software basados en aprendizaje profundo. Asimismo, se describen las acciones implementadas para reducir sesgos y mitigar posibles efectos no deseados.

8.4.1. Validez del constructo

En este estudio, la calidad del servicio cloud se operacionalizó mediante dieciséis métricas de QoS que abarcan desde el consumo de recursos hasta la latencia, la disponibilidad y el rendimiento, lo que proporciona un alcance más amplio que estudios previos, en particular aquellos centrados en un único atributo—como el tiempo de respuesta—.

No obstante, los modelos se desarrollaron siguiendo un enfoque de predicción univariante, analizando cada métrica de forma aislada. Aunque esta aproximación facilita la interpretación y reduce la complejidad, omite posibles interdependencias entre atributos de QoS que pueden influir en el cumplimiento del SLA. Trabajos futuros podrían explorar modelos multivariantes o arquitecturas basadas en mecanismos capaces de capturar explícitamente las interdependencias entre métricas.

Asimismo, si bien el uso de métricas como RMSE, MAE y MAPE constituye una práctica ampliamente aceptada para medir errores de predicción, estas no reflejan explícitamente costes asimétricos o penalizaciones a los SLA, aspectos de especial relevancia para la organización. De igual manera, aunque la selección de BI-GRU, LSTM y ARIMA está respaldada por la literatura y su aplicabilidad práctica, la exclusión de modelos más

recientes—como arquitecturas Transformer o enfoques híbridos—limita parcialmente la amplitud del constructo.

8.4.2. Validez interna

La validez interna se refiere al grado en que las diferencias observadas en el desempeño de los modelos pueden atribuirse a las intervenciones experimentales y no a factores no controlados. Una amenaza relevante proviene de la naturaleza inherentemente estocástica del entrenamiento de modelos de aprendizaje profundo (p. ej., inicialización aleatoria de pesos o uso de *dropout*). Para mitigar este riesgo, se realizaron múltiples iteraciones de entrenamiento y se adoptó la validación cruzada *prequential*, que respeta el orden temporal de los datos y aumenta la robustez de la evaluación. Estas prácticas se alinean con las recomendaciones metodológicas para experimentos con series temporales.

Aun así, reconocemos que fuentes menores de no determinismo—como operaciones a nivel de GPU—podrían introducir variabilidad residual. Para reducir su impacto, se emplearon configuraciones de ejecución consistentes y evaluaciones repetidas orientadas a garantizar la fiabilidad.

Otra amenaza interna está asociada al preprocesamiento de datos. En particular, se excluyeron datos anómalos relacionados con las métricas de uso de disco debido a fallos verificados de instrumentación ocurridos durante una transición conocida de infraestructura. Aunque esta decisión garantizó la calidad de los datos, la convergencia del entrenamiento y la estabilidad del modelo, pudo haber introducido sesgos si los datos omitidos representaban escenarios reales de degradación. No obstante, dado que la exclusión se basó en un error documentado y no en una anomalía operacional real del servicio, es poco probable que haya afectado patrones representativos de este.

8.4.3. Validez externa

El estudio se llevó a cabo utilizando datos reales de un servicio SaaS específico (SAlert), operado por una empresa de TI de tamaño medio y monitorizado durante un periodo de cinco meses. Aunque estos datos reflejan el comportamiento real del servicio y aportan relevancia práctica, existen limitaciones respecto a la generalización de los resultados. En particular, los hallazgos no pueden extrapolarse directamente a otros tipos de servicios, plataformas cloud o arquitecturas de implementación, como entornos basados en microservicios.

A pesar de que la inclusión de dieciséis métricas de QoS amplía el alcance de la evaluación, el entorno experimental se mantiene limitado a un único SaaS y proveedor. De la misma manera, el intervalo de muestreo de 30 minutos—acordado con expertos para equilibrar precisión y sobrecarga operacional—responde a un contexto operativo particular. Considerando que el rendimiento de los modelos puede variar según el intervalo temporal, futuras investigaciones podrían explorar diferentes intervalos y condiciones de monitorización para fortalecer la generalización de los resultados.

8.4.4. Validez de las conclusiones

En el estudio se emplearon estadísticas descriptivas e inferenciales, incluyendo intervalos de confianza, el indicador `RPAI` y la inspección visual de distribuciones de error para comparar los modelos de pronóstico. Estas medidas superan los estándares de reporte observados en investigaciones previas. Además, se construyó un modelo independiente para cada métrica y se realizaron comparaciones sobre sus observaciones, analizando la variabilidad propia mediante la desviación estándar, aunque dentro del contexto de un único entorno SaaS. Trabajos futuros deberían incluir la experimentación con otros servicios cloud para aumentar la variabilidad experimental y reforzar la solidez de las conclusiones.

Asimismo, el análisis incorporó pruebas estadísticas multivariantes —en particular PERMANOVA— aplicadas a todos los modelos, complementadas con comparaciones por pares y visualización mediante elipses de confianza. Esto contribuye a una validación rigurosa de las hipótesis evaluadas y respalda la generalización de las conclusiones.

Finalmente, se reportó explícitamente la relación entre precisión de predicción y coste computacional, un aspecto frecuentemente omitido en estudios de pronóstico de QoS y que resulta fundamental en entornos operativos con recursos limitados.

8.5. Implicaciones

Los resultados de este estudio tienen importantes implicaciones tanto para los investigadores como para los profesionales responsables del aseguramiento de la calidad y la gestión de los SLA en servicios cloud. La aplicación de modelos de aprendizaje profundo para el pronóstico de la calidad en un entorno real, similar al de producción, permite comprender con mayor precisión cómo este enfoque puede mejorar la observabilidad de la calidad del servicio y facilitar una gestión proactiva de la misma.

8.5.1. Implicaciones para investigadores

Los hallazgos refuerzan la evidencia de que el aprendizaje automático—y en particular el aprendizaje profundo—constituye un enfoque eficaz para el pronóstico de métricas de QoS en servicios cloud. La comparación empírica entre BI-GRU, LSTM y ARIMA demuestra la superioridad de los modelos basados en DL para capturar patrones temporales no lineales característicos de estos contextos dinámicos.

Además de los resultados de rendimiento, los experimentos permiten identificar consideraciones metodológicas relevantes para futuras investigaciones:

- `Estrategia de validación para series temporales`. La evaluación de modelos de pronóstico de series temporales exige técnicas de validación que preserven la secuencia temporal de los datos. La validación *prequential* empleada en este trabajo ofrece estimaciones más robustas del rendimiento bajo condiciones operativas reales. Futuras investigaciones deberían comparar sistemáticamente diferentes estrategias sensibles al tiempo para establecer guías de mejores prácticas.

- `Especificación de arquitecturas y reporte de hiperparámetros`. Una de las carencias recurrentes en la literatura sobre pronóstico de QoS es la limitada transparencia en la descripción de las arquitecturas de los modelos y los procesos de ajuste. Este estudio contribuye a cerrar esta brecha mediante la documentación detallada del flujo de modelado, explicitando la configuración de arquitecturas, la selección y ajuste de hiperparámetros, los procesos de entrenamiento y artefactos necesarios para la replicación. Este nivel de detalle permite avanzar hacia estándares de reporte más rigurosos y prácticas alineadas con la ciencia abierta, promoviendo la reproducibilidad y contribuyendo a reducir la fragmentación conceptual del dominio.

- `Definiciones de QoS con métricas multidimensionales`. La inclusión de dieciséis métricas de QoS aporta una visión más realista y holística de la calidad del servicio respecto a estudios centrados en un número reducido de métricas, como el tiempo de respuesta. Sin embargo, el enfoque univariante utilizado limita la capacidad de capturar relaciones entre atributos. Futuras investigaciones deberían explorar modelos multivariantes o arquitecturas capaces de aprender interdependencias entre métricas, lo que permitiría una detección temprana y precisa de degradaciones y violaciones de SLA.

- `Integración de conocimiento experto`. La integración de conocimiento experto junto con los pronósticos automáticos constituye una línea de investigación prometedora. El desarrollo de modelos híbridos, sensibles al contexto operativo y a los umbrales definidos por la organización, podría mejorar la interpretabilidad y favorecer una adopción más confiable de las soluciones basadas en ML y DL.

En conjunto, este estudio pone de manifiesto carencias persistentes en las prácticas experimentales actuales, como la limitada adopción de modelos multivariados y la escasa aplicación de técnicas avanzadas que favorezcan la transparencia y explicabilidad. Abordar estos desafíos implica considerar de manera más rigurosa el comportamiento no estacionario de los servicios cloud, un aspecto crítico para mejorar la estabilidad y fiabilidad de los pronósticos. Atender estas limitaciones abrirá líneas de investigación prometedoras orientadas al desarrollo de modelos predictivos más robustos, lo que facilitaría su adopción práctica y fortalecería la confianza de los *stakeholders* en los resultados generados por los modelos de predicción.

8.5.2. Implicaciones para profesionales de la industria

Desde la perspectiva de la industria, la aplicación de modelos DL para el pronóstico de QoS abre oportunidades significativas para la gestión proactiva del servicio, la prevención de incidentes y la mejora del cumplimiento de los SLA. Los resultados demuestran que los modelos BI-GRU y LSTM pueden ofrecer predicciones precisas y fiables sobre diversas métricas operativas, habilitando mecanismos tempranos de alerta y automatización.

Dado que este estudio emplea datos reales de monitorización en un entorno SaaS, los hallazgos tienen una alta relevancia operativa. La evaluación de múltiples atributos de QoS

permite enfoques de gestión más integrales que aquellos basados únicamente en umbrales unidimensionales.

No obstante, la adopción efectiva de modelos de predicción en entornos productivos requiere considerar diversos factores técnicos, organizativos y operativos:

- `Selección del modelo y ajuste de hiperparámetros`. El rendimiento de los modelos de DL depende críticamente de las decisiones arquitectónicas y de los parámetros de entrenamiento. Los equipos técnicos deben planificar procesos sistemáticos de ajuste, empleando técnicas como *grid search*, *random search* o métodos bayesianos, y validarlos sobre datos representativos. De lo contrario, existe el riesgo de obtener modelos sobreajustados o poco adaptables a variaciones temporales.

- `Granularidad temporal y horizonte de predicción`. La precisión del pronóstico está directamente influida por el intervalo de muestreo y el horizonte de predicción. Syu et al. [18] señalan que una mayor granularidad—p. ej., intervalos de un minuto—capturar mejor las fluctuaciones repentinas, pero incrementa el ruido y la carga computacional. Por el contrario, una menor granularidad—como intervalos de una hora—reduce el ruido, pero puede omitir anomalías de corto plazo.

  Así, los profesionales deben ajustar la granularidad temporal y el horizonte de predicción en función de sus objetivos operativos, como la detección temprana de anomalías o la planificación de capacidad a largo plazo, y revisar estos ajustes periódicamente a medida que evoluciona la carga de trabajo.

- `Validación cruzada y generalización del modelo`. La validación cruzada tradicional *k-fold* no es adecuada para series temporales, ya que altera la secuencia temporal. En su lugar, se recomienda emplear técnicas como la *prequential*, que preserva la integridad secuencial. La adopción de estas prácticas mejora la capacidad de generalización del modelo y proporciona resultados más fiables para entornos de producción.

- `Monitorización continua, reentrenamiento e integración de MLOps`. Considerando la naturaleza dinámica de los entornos cloud, las cargas de trabajo, la infraestructura o los propios servicios pueden experimentar variaciones constantes como resultado de actualizaciones, cambios de configuración o fluctuaciones de demanda. Para mantener la precisión y la fiabilidad de las predicciones, resulta esencial incorporar *pipelines* dentro de los procesos de DevOps y MLOps que permitan monitorizar el desempeño del modelo, automatizar ajustes de configuración y activar estrategias de reentrenamiento ante desviaciones significativas. La integración de estos mecanismos en flujos automatizados de despliegue y monitorización favorece el cumplimiento proactivo de los SLA, facilita la escalabilidad y posibilita la adopción temprana de medidas correctivas. Sin estos mecanismos, incluso los modelos mejor diseñados tienden a degradarse con el tiempo.

- `Gestión de recursos y análisis coste-beneficio`. La ejecución y despliegue de modelos DL conllevan una carga computacional significativa, lo que los convierte

en un factor crítico en entornos industriales. Los costes operativos asociados a su mantenimiento pueden limitar su adopción práctica si no se gestionan cuidadosamente. Por ello, las organizaciones deben evaluar si los beneficios obtenidos—como la reducción de violaciones de SLA, la mejora en la fiabilidad del servicio o, en definitiva, un mayor nivel de satisfacción de los usuarios—compensan los costes añadidos. Asimismo, resulta pertinente considerar técnicas que permitan optimizar esta relación coste-beneficio—tales como la compresión de modelos, la inferencia acelerada o la provisión elástica de GPU en la nube—contribuyendo a equilibrar el rendimiento y la eficiencia operativa y permitiendo una implementación sostenible a gran escala.

- `Documentación, control de versiones y reproducibilidad.` La documentación detallada y el control de versiones para datos, código y artefactos del modelo son esenciales para garantizar la trazabilidad, facilitar auditorías y sostener la mejora continua. El enfoque reproducible presentado en este estudio constituye una referencia para equipos que buscan institucionalizar prácticas de modelado robustas en entornos de operación cloud. Este estudio aporta un enfoque de modelización reproducible y directrices prácticas sobre el preprocesamiento de datos, el uso de métricas de precisión y la interpretación de las predicciones, con el fin de apoyar a los profesionales en la implementación de modelos de pronóstico. Para ello, es imprescindible documentar las versiones de los modelos, las transformaciones aplicadas a los datos, las configuraciones de entrenamiento y los procedimientos de evaluación. El mantenimiento de registros versionados promueve la trazabilidad ante posibles regresiones de rendimiento, facilita las auditorías y apoya el aprendizaje organizacional a través de la experimentación iterativa. En conjunto, estas prácticas reducen las barreras para la adopción de técnicas de aprendizaje automático en equipos de operaciones cloud y promueven una cultura de mejora continua y experimentación.

Este estudio evidencia el valor tangible de los modelos de aprendizaje profundo para transformar la gestión de la calidad del servicio, pasando de un enfoque reactivo a una inteligencia proactiva. Al anticipar degradaciones en las métricas de QoS, las organizaciones pueden mitigar su impacto antes de afectar al usuario final, así como optimizar sus procesos de planificación y operación avanzando hacia servicios cloud más fiables, eficientes y resilientes.

8.6. Conclusiones

En este capítulo se propusieron dos modelos de aprendizaje automático orientados al pronóstico de la calidad en servicios cloud, desarrollados específicamente para una empresa de TI. Se evaluaron los modelos BI-GRU y LSTM mediante un experimento comparativo que incluyó también el enfoque estadístico tradicional ARIMA. A diferencia de trabajos previos, este estudio incorporó una amplia variedad de atributos y métricas de calidad, ofreciendo así un escenario de evaluación más representativo de entornos cloud reales.

Con el fin de promover la transparencia y facilitar la replicación, se puso a disposición de la comunidad científica el material experimental, los conjuntos de datos y los resultados

obtenidos. Los modelos desarrollados utilizan 16 métricas de QoS y sus correspondientes mediciones, lo que permite caracterizar de manera integral el comportamiento del servicio cloud analizado.

Los resultados revelan que el modelo BI-GRU alcanza niveles elevados de precisión en la mayoría de las métricas de QoS, como lo evidencian las métricas de precisión y sus intervalos de confianza, los cuales se mantienen ajustados y próximos a cero. Estos hallazgos confirman que BI-GRU constituye una herramienta eficaz para la monitorización continua de servicios cloud en operación, la evaluación de servicios candidatos y la detección temprana de posibles violaciones de SLA. Por lo tanto, su incorporación puede fortalecer la toma de decisiones, particularmente en tareas de recomposición proactiva de servicios o selección de alternativas óptimas.

Aunque algunas métricas de QoS mostraron un rendimiento limitado con BI-GRU, este comportamiento es coherente con la evidencia ampliamente aceptada de que ningún modelo de aprendizaje automático para series temporales domina todas las tareas de pronóstico. En esta línea, futuros trabajos explorarán modelos alternativos de ML y DL, así como variantes que permitan analizar correlaciones entre métricas y dependencias complejas, con el objetivo de mejorar la precisión y robustez de los pronósticos.

El equipo de la empresa valoró positivamente la utilidad del modelo BI-GRU como complemento a su estrategia actual de monitorización cloud. Asimismo, señalaron oportunidades de mejora que orientan investigaciones futuras, como la ampliación del proceso de recolección de datos hacia servicios adicionales, la incorporación de un mayor número de métricas de QoS y el análisis más profundo de las relaciones entre ellas para obtener una visión más completa del comportamiento de los servicios cloud.

En conjunto, los resultados aportan evidencia empírica sólida de que los modelos de aprendizaje automático basados en redes neuronales recurrentes—particularmente BI-GRU—constituyen un enfoque prometedor para el pronóstico de la calidad en servicios cloud. Además, la documentación detallada proporcionada sobre la arquitectura de los modelos y las estrategias de ajuste de hiperparámetros contribuye a la reproducibilidad y puede servir como guía para equipos interesados en adoptar o desarrollar soluciones de pronóstico adaptadas a sus contextos operativos específicos.

Capítulo 9

Conclusiones y Trabajos Futuros

El objetivo general de esta tesis ha sido avanzar en la comprensión, evaluación y pronóstico de la calidad de los servicios cloud, un dominio caracterizado por su complejidad, dinamismo y fuerte dependencia de métricas operativas.

Las contribuciones desarrolladas a lo largo del trabajo constituyen un conjunto coherente de avances conceptuales, metodológicos y empíricos, diseñados para abordar de forma integral las limitaciones detectadas en la literatura y en los entornos industriales.

Este capítulo se estructura como sigue. La Sección 9.1 presenta las contribuciones principales de este trabajo. La Sección 9.2 analiza el grado de consecución de los objetivos planteados que guiaron el desarrollo de la tesis. La Sección 9.3 ofrece una síntesis holística de las aportaciones realizadas, mientras que la Sección 9.4 discute las principales limitaciones del trabajo. La Sección 9.5 propone diversas líneas de investigación futura derivadas de estas limitaciones y resultados. Finalmente, la Sección 9.6 describe las publicaciones y resultados de difusión que sustentan esta tesis.

9.1. Contribuciones principales

Las principales contribuciones de esta tesis se integran en un conjunto coherente de resultados que contribuyen, de manera complementaria, a la caracterización, evaluación y predicción de la calidad en servicios cloud. Dichas contribuciones avanzan el conocimiento teórico, metodológico y empírico en este dominio y proporcionan artefactos conceptuales y herramientas prácticas que pueden ser empleados tanto por los investigadores como por los profesionales de la industria. A continuación se sintetizan las aportaciones principales.

9.1.1. Contribuciones conceptuales

9.1.1.1. Una taxonomía de métricas de QoS rigurosa y estructurada

La taxonomía desarrollada constituye una contribución conceptual significativa porque organiza un dominio extremadamente fragmentado. Las métricas de QoS disponibles en la literatura y en reportes industriales presentan diferencias terminológicas, niveles de granularidad heterogéneos y escasas relaciones explícitas entre ellas. La taxonomía permite superar esta dispersión al clasificar las métricas según criterios operativos y conceptuales alineados con el modelo de capas cloud (SaaS, PaaS, IaaS).

Además, esta estructura contribuye a: i) formalizar el papel de las métricas como indicadores operacionales de calidad; ii) facilitar la selección informada de métricas en procesos de monitorización; iii) proveer un lenguaje común que puede ser utilizado por investigadores o profesionales en entornos industriales; iv) servir de base para la definición de atributos del modelo de calidad

La taxonomía no solo sistematiza métricas existentes, sino que también ofrece criterios para incorporar métricas futuras, lo que la convierte en un instrumento conceptual con valor prolongado.

9.1.1.2. Modelo de calidad alineado con ISO/IEC 25010 para servicios cloud

Aunque existen modelos de calidad específicos para servicios cloud, la mayoría se caracteriza por una cobertura limitada, centrada en un número reducido de atributos y métricas, generalmente de bajo nivel y sin una estructura conceptual que permita asociarlas de forma sistemática con características de calidad más abstractas. Esta fragmentación dificulta su aplicación en evaluaciones integrales y limita su utilidad para organizaciones que necesitan una visión holística de la calidad del servicio.

El modelo propuesto en esta tesis aborda estas carencias al extender de manera rigurosa el estándar ISO/IEC 25010 y adaptarlo explícitamente al dominio cloud. La contribución reside en reorganizar el estándar para: representar cada característica mediante subcaracterísticas relevantes para servicios cloud; asociar cada subcaracterística con atributos específicos y medibles; y garantizar la trazabilidad de cada atributo hacia métricas derivadas de la taxonomía desarrollada previamente. Esta articulación permite conectar definiciones abstractas de calidad con indicadores empíricos observables, algo que los modelos existentes no abordan de manera sistemática.

El resultado es un modelo que aporta claridad conceptual, una granularidad significativamente mayor que la de los enfoques previos, y un rigor formal que facilita su aplicación práctica. Asimismo, proporciona una estructura adecuada para analizar la calidad desde múltiples perspectivas (cliente, proveedor del servicio, arquitectura e infraestructura operativa) y constituye una interpretación sólida y extensible del estándar orientada a sistemas cloud. Este enfoque permite integrar prácticas reales de monitorización con un marco formal de evaluación de calidad plenamente alineado con ISO/IEC 25010.

Además, el modelo se ha materializado en una herramienta interactiva basada en Power BI que facilita su comprensión y uso por parte de distintos perfiles técnicos y no técnicos. Esta herramienta permite navegar por las características, atributos y métricas, visualizar sus relaciones, explorar diferentes niveles de granularidad y adaptar el modelo incorporando nuevas métricas o ajustando atributos según las necesidades del servicio evaluado. Su capacidad para integrar fuentes de datos reales y generar indicadores comparativos refuerza el valor práctico del modelo y contribuye a su aplicabilidad práctica.

En conjunto, esta contribución avanza hacia una *estandarización conceptual más completa de la calidad en servicios cloud*, ofreciendo una estructura formal, granular y extensible que supera las limitaciones de los modelos existentes y conecta la teoría del estándar ISO con las prácticas de monitorización y análisis presentes en entornos cloud modernos.

9.1.2. Contribuciones metodológicas

9.1.2.1. Un proceso de evaluación de la calidad conforme a ISO/IEC 25040

La tesis contribuye metodológicamente al definir un proceso estructurado para evaluar la calidad de servicios cloud. Este proceso actúa como el puente operativo entre la definición conceptual del modelo de calidad y su aplicación práctica, permitiendo realizar evaluaciones objetivas, repetibles y alineadas con las necesidades organizacionales.

El proceso abarca la formulación de requisitos verificables, la configuración del modelo de calidad personalizado, la selección de métricas y herramientas de monitorización, la definición de umbrales y criterios de decisión, y la operacionalización mediante *pipelines* de monitorización. Con ello se aborda una limitación central de la literatura: la ausencia de metodologías reproducibles que permitan aplicar modelos de calidad en entornos cloud reales. Los aportes metodológicos resultantes facilitan la repetibilidad de las evaluaciones, la comparación entre servicios y la realización de auditorías de calidad basadas en datos.

Una característica clave del proceso es su integración explícita con el modelo de calidad propuesto, que especifica atributos y métricas de forma estructurada. Esta integración permite la selección de métricas de manera informada, evitando ambigüedades frecuentes en la literatura y reduciendo la subjetividad en la toma de decisiones. Asimismo, el proceso incorpora prácticas recomendadas para la interpretación de métricas, la generación de informes, el establecimiento de umbrales, la identificación de riesgos y priorización de acciones de mejora, contribuyendo a cerrar la brecha entre el análisis técnico y la gestión estratégica del servicio cloud.

En conjunto, esta contribución proporciona un mecanismo operativo robusto que formaliza cómo evaluar la calidad de un servicio cloud utilizando el modelo propuesto, y

establece las bases metodológicas necesarias para su integración posterior con los modelos de aprendizaje automático, reforzando la coherencia interna del marco propuesto.

9.1.2.2. Un proceso sistemático para el modelado predictivo de QoS

La tesis propone un proceso estructurado que organiza de forma clara, explícita y replicable todas las etapas necesarias para construir modelos predictivos aplicados a métricas de QoS. Esta contribución metodológica permite superar la fragmentación habitual en la literatura, donde los estudios de predicción suelen presentarse como esfuerzos experimentales aislados, con decisiones difíciles de reproducir y con escasa justificación metodológica. El proceso definido en esta tesis formaliza el ciclo completo de modelado, desde la preparación de datos y la selección de ventanas temporales hasta la construcción, validación y ajuste de los modelos, incorporando además prácticas rigurosas como la evaluación prequential, especialmente adecuada para series temporales en evolución.

Un aspecto relevante del proceso es su enfoque univariante, que, aunque acotado, permite aislar y comprender con mayor precisión el comportamiento intrínseco de cada métrica antes de abordar escenarios multivariantes más complejos. Esta aproximación facilita la comparación entre modelos estadísticos y de deep learning, y proporciona criterios claros para justificar la selección de técnicas como ARIMA, LSTM o BI-GRU según las propiedades estadísticas de las series. La inclusión de modelos conceptualmente distintos, junto con una metodología común para evaluarlos, contribuye a generar conclusiones más robustas y comparables que las encontradas en estudios previos.

El proceso propuesto también posee un valor generalizable: puede aplicarse a servicios cloud con diferentes arquitecturas, métricas o patrones de carga, y sienta las bases metodológicas necesarias para incorporar modelos más avanzados o multivariantes en investigaciones futuras. Al estructurar el modelado predictivo como una actividad sistemática y no como un ejercicio ad hoc, la tesis eleva la práctica del pronóstico de QoS hacia un enfoque más científico, transparente y replicable, fomentando la acumulación de conocimiento sólido en un ámbito donde la variabilidad de técnicas y configuraciones dificultaba la comparación entre estudios.

En definitiva, esta contribución proporciona un marco metodológico completo, reproducible y adaptable, que permite a investigadores y profesionales construir modelos de pronóstico de QoS de manera sistemática y fundamentada. Su diseño integra los requisitos conceptuales establecidos por el modelo de calidad, las buenas prácticas para la preparación de datos, la construcción y el uso de modelos de aprendizaje automático. El proceso constituye una herramienta esencial para avanzar hacia evaluaciones predictivas robustas y para fortalecer la capacidad de observabilidad inteligente en servicios cloud.

9.1.3. Contribuciones empíricas

9.1.3.1. Validación industrial del modelo de calidad

La validación industrial del modelo de calidad constituye una de las contribuciones empíricas más relevantes de esta tesis, ya que permitió demostrar su aplicabilidad en un

entorno real. El estudio de caso realizado en una organización proveedora de servicios cloud facilitó la instanciación y adaptación del modelo genérico a un servicio específico, evidenciando su flexibilidad y su capacidad para representar necesidades de calidad concretas.

Durante la validación se seleccionaron características, subcaracterísticas y atributos prioritarios, y se asociaron con métricas de monitorización disponibles en la infraestructura existente. Este proceso mostró que el modelo permite estructurar de forma sistemática la evaluación de calidad, identificar desviaciones respecto a niveles de servicio esperados y generar información útil para la toma de decisiones. Los resultados incluyeron la detección de incumplimientos, la identificación de áreas críticas y la formulación de acciones de mejora orientadas a la gestión del servicio.

La empresa valoró positivamente la capacidad del modelo para ofrecer una visión holística de la calidad del servicio, integrando múltiples dimensiones como disponibilidad, rendimiento, utilización de recursos y comportamiento de red, más allá de las métricas tradicionales basadas únicamente en tiempos de respuesta.

Asimismo, la integración del modelo con herramientas reales de monitorización demostró su viabilidad técnica y su capacidad para coexistir con prácticas de observabilidad ya establecidas. La experiencia adquirida confirma que el marco no solo es conceptualmente sólido, sino también operativo y aplicable en contextos industriales. En conjunto, esta validación aporta evidencia empírica sobre el valor práctico del modelo y su potencial de adopción en organizaciones que gestionan servicios cloud.

Finalmente, la evaluación industrial permitió identificar oportunidades de mejora que enriquecieron el modelo y el proceso, como la necesidad de incorporar métricas específicas para escenarios de carga particulares o ampliar las guías metodológicas para la definición de umbrales de rendimiento. Estos ajustes fortalecieron la coherencia interna del modelo y contribuyeron a refinar sus mecanismos de aplicación práctica.

9.1.3.2. Evaluación experimental de modelos predictivos de QoS

El experimento controlado realizado con ARIMA, LSTM y BI-GRU constituye una contribución empírica relevante, al ofrecer evidencia comparativa sobre la capacidad de distintos enfoques para anticipar métricas de QoS en servicios cloud. La evaluación sistemática de estos modelos permitió observar cómo técnicas estadísticas tradicionales y métodos de *deep learning* difieren en su habilidad para capturar patrones temporales, fluctuaciones abruptas y comportamientos no lineales característicos de métricas operacionales. Los resultados muestran que los modelos basados en redes neuronales, y particularmente los modelos bidireccionales como BI-GRU, son capaces de aprovechar dependencias temporales más complejas, proporcionando predicciones más estables, precisas y con mejor capacidad de generalización que los modelos clásicos.

La aportación del experimento no radica únicamente en los resultados cuantitativos, sino también en el proceso de análisis que permitió interpretar las condiciones bajo las cuales cada enfoque resulta más adecuado. La comparación detallada entre los modelos clarificó, por ejemplo, la sensibilidad de ARIMA a cambios estructurales en la serie, la ne-

cesidad de volúmenes significativos de datos para el entrenamiento de modelos neuronales, y la ventaja de los modelos bidireccionales para capturar relaciones temporales contextuales que los métodos unidireccionales tienden a pasar por alto. Esta reflexión metodológica aporta un valor añadido al estado del arte, ya que proporciona criterios explícitos que pueden guiar la selección de modelos en futuros estudios y aplicaciones prácticas.

Además, la experimentación se realizó dentro de un marco de evaluación riguroso que incorporó prácticas como la validación prequential, la normalización de series, el uso de métricas comparables y la definición de ventanas de predicción consistentes. Este enfoque contribuye a aumentar la transparencia y reproducibilidad del estudio, elementos frecuentemente ausentes en estudios previos sobre pronóstico de QoS. En conjunto, el experimento no solo ofrece evidencia sólida sobre el desempeño de distintos modelos, sino que también establece una base metodológica que puede servir de referencia para investigaciones futuras en pronóstico de métricas para servicios cloud.

9.1.3.3. Artefactos y herramientas reutilizables

Una contribución adicional de esta tesis, tanto de carácter práctico como de transferencia de conocimiento, es la generación de un conjunto de artefactos y herramientas reutilizables que permiten replicar, adaptar y extender los resultados obtenidos. Estos artefactos incluyen scripts de procesamiento y análisis de datos, configuraciones de pipelines de monitorización, especificaciones de métricas, plantillas de modelos personalizados, implementaciones experimentales de modelos predictivos y un *dashboard* interactivo en *Power BI* que materializa el modelo de calidad y facilita su interpretación por parte de los *stakeholders*. En conjunto, estos materiales no solo respaldan la reproducibilidad científica del trabajo, sino que también proporcionan un punto de partida robusto para investigadores y profesionales que deseen aplicar o evolucionar el marco en distintos contextos cloud.

Entre los artefactos desarrollados destacan los **scripts utilizados en el experimento de pronóstico**, implementados en Python y diseñados para permitir una reproducción completa del proceso experimental. Estos scripts incluyen rutinas para la ingesta, limpieza y normalización de datos temporales; la construcción y parametrización de los modelos ARIMA, LSTM y BI-GRU; la definición de ventanas temporales y funciones de evaluación prequential; y la generación de resultados comparables mediante métricas estandarizadas como MAE y RMSE. Además de facilitar la repetición del experimento, estos scripts sirven como base extensible para incorporar nuevos modelos, explorar escenarios multivariantes o analizar métricas adicionales, contribuyendo así al avance metodológico en predicción de QoS.

En cuanto a la evaluación de calidad, se desarrollaron **configuraciones de pipelines de monitorización** que especifican cómo capturar datos desde distintas fuentes, cómo estructurarlos, cómo validar su consistencia y cómo transformarlos en indicadores operativos adaptados al modelo de calidad. Estos pipelines incluyen documentación sobre intervalos de muestreo, formatos de datos y pasos de preprocesamiento que permiten adaptar el proceso a servicios cloud heterogéneos. Este conjunto de materiales facilita la adopción del modelo en entornos empresariales, donde la integración con sistemas existentes de

observabilidad es clave para su uso operativo.

Asimismo, se elaboraron **plantillas y ejemplos de modelos de calidad personalizados**, que ilustran cómo instanciar la estructura genérica del modelo en dominios concretos, cómo seleccionar atributos y métricas relevantes y cómo establecer umbrales de cumplimiento. Este material sirve como guía en organizaciones que deseen aplicar el marco a sus propios servicios, reduciendo el esfuerzo de adaptación y promoviendo buenas prácticas basadas en estándares internacionales.

Finalmente, el *dashboard de Power BI* proporciona un entorno visual que integra el modelo de calidad con los datos de monitorización y permite a los stakeholders navegar entre características, subcaracterísticas, atributos y métricas, visualizar resultados y detectar desviaciones de una forma clara y accesible. Este *dashboard* constituye un puente entre el rigor conceptual del modelo y las necesidades prácticas de quienes deben interpretarlo y tomar decisiones operativas. Su disponibilidad facilita la incorporación del marco en procesos reales de gestión de calidad y promueve su uso en contextos industriales donde la visualización y la interacción con datos son elementos clave para la toma de decisiones.

En conjunto, estos artefactos representan una contribución sustantiva a la comunidad, ya que materializan las propuestas del marco de una manera accesible, reproducible y adaptable. Constituyen, además, una base sólida para futuras investigaciones que busquen extender el modelo, evaluar su comportamiento en nuevos servicios cloud o explorar técnicas predictivas emergentes.

9.2. Análisis de la consecución de los objetivos

El desarrollo de esta tesis se estructuró en torno a siete objetivos específicos que guiaron tanto la construcción conceptual como la elaboración metodológica y la validación empírica del marco propuesto. Su consecución no solo supuso la obtención de artefactos y resultados formales, sino también un proceso de comprensión progresiva sobre la naturaleza de la calidad en servicios cloud y sobre los desafíos asociados a su evaluación y pronóstico. A continuación, se presenta un análisis del grado de consecución de cada objetivo y de los aprendizajes derivados.

- **O1. Analizar el estado del arte sobre evaluación de calidad y pronóstico en servicios cloud.** Este objetivo permitió establecer las bases conceptuales de la tesis, revelando la fragmentación existente en la literatura respecto a definiciones, métricas, procesos y enfoques predictivos. El análisis de la literatura identificó limitaciones persistentes (como la ausencia de marcos unificados, la falta de metodologías reproducibles y la escasa conexión entre métricas y modelos de calidad) que justificaron la necesidad de un enfoque integral. La revisión no solo cumplió su finalidad descriptiva, sino que desempeñó un papel estratégico en la identificación de los desafíos que la tesis abordaría posteriormente.

- **O2. Elaborar una taxonomía de métricas de QoS para su uso en monitorización.** El objetivo se alcanzó mediante la construcción de una taxonomía

rigurosa que clasifica de manera estructurada métricas heterogéneas provenientes de literatura académica e informes industriales. Su desarrollo exigió analizar la operacionalidad de cada métrica, su nivel de abstracción y su relación con distintos niveles de la arquitectura cloud. La taxonomía no fue únicamente un producto clasificatorio: se convirtió en una pieza clave del marco, proporcionando el vocabulario técnico y las relaciones necesarias para sustentar los objetivos posteriores. La experiencia derivada de este proceso confirmó que, sin una estructura conceptual clara, resulta imposible desarrollar modelos de calidad y procesos de evaluación coherentes.

- **O3. Proponer un modelo de calidad alineado con ISO/IEC 25010 adaptado a servicios cloud.** Este objetivo implicó un ejercicio de interpretación profunda del estándar ISO/IEC 25010 y de su articulación con el dominio cloud. La construcción del modelo exigió definir subcaracterísticas específicas, atributos medibles y mecanismos de trazabilidad hacia métricas de la taxonomía. Su logro permitió demostrar que es posible integrar definiciones abstractas de calidad con observaciones empíricas del comportamiento del servicio, reduciendo la distancia entre estándares y práctica industrial. Al finalizar este objetivo, se obtuvo un modelo con la granularidad y extensibilidad necesarias para ser instanciado en múltiples organizaciones.

- **O4. Definir un proceso sistemático de evaluación basado en el modelo de calidad** La consecución de este objetivo permitió operacionalizar el modelo de calidad mediante un proceso conforme la ISO/IEC 25040. Este proceso no solo formaliza actividades como la selección de atributos, la definición de umbrales y la interpretación de métricas, sino que también estructura la participación de distintos stakeholders y garantiza la reproducibilidad de la evaluación. La experiencia de diseño evidenció que la evaluación de calidad requiere integrar decisiones técnicas, organizativas y analíticas, y que un proceso explícito es esencial para que el modelo pueda ser aplicado consistentemente en entornos reales.

- **O5. Operacionalizar y evaluar empíricamente el modelo y el proceso en un entorno industrial** Este objetivo permitió validar tanto la utilidad práctica como la viabilidad técnica del marco en un caso real. La adaptación del modelo y del proceso a un servicio cloud específico reveló de forma directa las fortalezas del enfoque (su flexibilidad, su capacidad de diagnóstico y su alineación con datos reales de monitorización) así como oportunidades de mejora que enriquecieron las iteraciones posteriores. La validación confirmó que el modelo puede integrarse en la infraestructura y prácticas de observabilidad existentes, reforzando así su aplicabilidad industrial. La consecución de este objetivo fue esencial para demostrar que el marco propuesto no se limita al plano conceptual, sino que puede ser adoptado por organizaciones reales para mejorar la gestión de la calidad de sus servicios.

- **O6. Definir un proceso para la construcción de modelos de pronóstico de QoS mediante ML** Este objetivo fue alcanzado mediante el diseño de un proceso metodológico que formaliza todas las etapas necesarias para construir modelos de

pronóstico: desde la definición de requisitos y la preparación de los datos hasta la validación prequential y el ajuste final de los modelos. Este proceso aportó claridad a un campo donde tradicionalmente predominaban enfoques ad hoc y experimentos difíciles de reproducir. Su consecución permitió conectar de forma explícita el modelado predictivo con los elementos conceptuales del marco y sentó las bases para comparar distintos métodos bajo criterios homogéneos.

- **O7. Construir, validar y comparar modelos de pronóstico para QoS** La ejecución del experimento controlado permitió cumplir plenamente este objetivo, proporcionando evidencia sobre el desempeño diferencial de modelos estadísticos y de *deep learning,* así como sobre las condiciones que favorecen a cada tipo de enfoque. El análisis no solo produjo resultados cuantitativos, sino que permitió comprender mejor la dinámica temporal de las métricas de QoS y la idoneidad de técnicas bidireccionales como BI-GRU. La consecución del objetivo O7 demostró que el proceso de modelado predictivo es válido, útil y capaz de generar conocimiento empírico que puede ser transferible a otros escenarios cloud.

En conjunto, los siete objetivos se cumplieron de manera integrada y progresiva. Su consecución permitió avanzar desde la identificación de una carencia conceptual y metodológica hasta la construcción y evaluación empírica de un marco completo para modelar, evaluar y predecir la calidad en servicios cloud. El proceso confirmó que abordar la gestión de la QoS de manera rigurosa requiere articular múltiples niveles (conceptual, metodológico y empírico) y que solo su integración permite ofrecer soluciones robustas, aplicables y alineadas con las necesidades del ámbito empresarial.

9.3. Síntesis de las contribuciones e implicaciones teóricas

Las contribuciones descritas deben interpretarse como partes interdependientes de un marco conceptual que redefine cómo se entiende y gestiona la calidad en entornos cloud. La **taxonomía** proporciona un lenguaje común que permite mapear métricas internas y externas a *dimensiones conceptuales de calidad*. El **modelo de calidad**, por su parte, ofrece la *estructura formal* que integra características abstractas, subcaracterísticas y atributos medibles, permitiendo conectar la teoría con la observación empírica. El proceso de evaluación define cómo debe operacionalizarse esta estructura en la práctica, traduciendo métricas en diagnósticos sistemáticos. Finalmente, el proceso predictivo extiende esta visión hacia un paradigma proactivo, en el que la calidad se entiende no como un estado que se observa, sino como un fenómeno dinámico susceptible de anticipación.

Desde una perspectiva teórica, la tesis demuestra que es posible articular un enfoque integral que conecte *evaluación* y *predicción* en un marco coherente. La capacidad de anticipar variaciones de QoS sugiere que la calidad en servicios cloud debe entenderse no solo como un constructo estático, sino como un proceso evolutivo influido por factores operativos, arquitectónicos y de carga. Esta visión aporta implicaciones relevantes para la Ingeniería del Software, ya que invita a evolucionar desde enfoques centrados en monitorización reactiva hacia modelos de gestión proactivos, adaptativos y basados en evidencia.

9.4. Limitaciones

Aunque esta tesis ha desarrollado un marco integral para el modelado, evaluación y pronóstico de la calidad de servicios cloud, es importante reconocer varias limitaciones que condicionan el alcance de las conclusiones. Estas limitaciones no solo reflejan decisiones metodológicas adoptadas, sino también la complejidad inherente del dominio y la madurez actual de las tecnologías involucradas. Su identificación permite enmarcar adecuadamente los resultados y, al mismo tiempo, señala caminos naturales para la evolución del trabajo.

9.4.1. Limitaciones conceptuales

- **Cobertura parcial del ecosistema de métricas cloud.** La taxonomía propuesta recoge un volumen significativo de métricas presentes en la literatura y documentación pública de proveedores, pero inevitablemente deja fuera métricas propietarias o emergentes asociadas a arquitecturas en rápido desarrollo, como *edge computing* o entornos *serverless*. Esta limitación refleja el alcance de la revisión realizada y el dinamismo del dominio: la naturaleza evolutiva de las métricas cloud hace que cualquier taxonomía sea necesariamente parcial y deba actualizarse de forma continua.

- **Dependencia de la estructura conceptual de ISO/IEC 25010.** El modelo de calidad propuesto extiende este estándar de manera rigurosa, pero sigue condicionado por su marco conceptual, que no contempla de forma explícita propiedades distintivas de los sistemas cloud, como elasticidad horizontal, resiliencia distribuida o autosuficiencia operacional. La adaptación realizada constituye un avance, pero la ausencia de un estándar específico para servicios cloud limita el espacio conceptual desde el cual construir un modelo completamente nativo del dominio.

- **Enfoque predominantemente univariante en el pronóstico de QoS.** Para facilitar el análisis y comparabilidad de modelos, la fase predictiva se centró en métricas individuales. Sin embargo, la calidad cloud es un fenómeno sistémico donde múltiples métricas interactúan entre sí. Esta aproximación simplificada es útil para establecer una línea base, pero limita la capacidad de capturar dependencias complejas que caracterizan escenarios reales de operación.

9.4.2. Limitaciones metodológicas

- **Dependencia de herramientas específicas y decisiones de diseño.** La operacionalización del proceso de evaluación se apoyó en herramientas particulares de monitorización y análisis, así como en intervalos de muestreo y transformaciones de datos que, si bien rigurosos, representan solo una de las posibles configuraciones viables. La replicación del proceso en otros entornos podría requerir ajustes sustanciales, especialmente en organizaciones con infraestructuras heterogéneas o pipelines de observabilidad más complejos.

- **Selección acotada de modelos predictivos.** El estudio experimental comparó modelos representativos —ARIMA, LSTM y BI-GRU— que permiten establecer conclusiones robustas sobre metodologías clásicas y modelos recurrentes. Sin embargo, el panorama actual de series temporales incluye enfoques más recientes basados en arquitecturas profundas especializadas (como N-BEATS, DeepAR o transformadores temporales), cuya exclusión limita el espectro de comparación y deja abierta la posibilidad de modelos más eficientes o precisos.
- **Evaluación en condiciones controladas.** Las métricas utilizadas en el experimento fueron previamente limpiadas, normalizadas y estacionarizadas, lo que favorece análisis comparables pero se distancia del comportamiento natural de métricas cloud en producción, donde los datos suelen presentar ruido, lagunas, eventos abruptos y comportamientos no estacionarios. Esto introduce una brecha entre la robustez experimental y la robustez operativa que debería ser analizada en furutos experimentos.

9.4.3. Limitaciones empíricas

- **Validación industrial limitada a un único caso.** Aunque el estudio de caso realizado permitió evaluar la aplicabilidad del modelo de calidad y del proceso de evaluación en un entorno real, la evidencia empírica sigue siendo circunscrita a las características de un único servicio, arquitectura y organización. Es necesario reconocer que la diversidad del ecosistema cloud puede generar comportamientos distintos en otros servicios, lo que plantea la necesidad de estudios en otros contextos para fortalecer la validez externa de las conclusiones.
- **Horizonte temporal de predicción limitado.** La predicción one-step es fundamental para la detección temprana de degradaciones, pero no aborda necesidades más amplias como la planificación de capacidad o la prevención anticipada de incumplimientos futuros. La ausencia de modelos multi-step o multi-horizon reduce el alcance del análisis predictivo y señala una línea natural de ampliación.

En conjunto, estas limitaciones no invalidan los resultados obtenidos. Más bien, permiten contextualizarlos y reconocer que el marco propuesto constituye un primer paso hacia un enfoque más amplio, capaz de incorporar métricas emergentes, arquitecturas predictivas avanzadas, escenarios multivariantes y evaluaciones en diversos entornos industriales. Precisamente porque la tesis adopta una aproximación integral, sus limitaciones muestran de forma clara las direcciones hacia las que debe avanzar la investigación para completar la visión iniciada en este trabajo.

9.5. Trabajos futuros

El trabajo desarrollado en esta tesis abre múltiples líneas de investigación que pueden fortalecer, extender y generalizar la aproximación propuesta para la evaluación y el pronóstico de la calidad de servicios cloud. A continuación se describen varias direcciones

futuras que representan oportunidades relevantes tanto a nivel científico como industrial. Estos trabajos futuros se estructuran en tres ejes principales que se detallan a continuación.

9.5.1. Extensión del marco conceptual de calidad

- **Extensión del modelo de calidad a nuevos dominios tecnológicos**

  El modelo de calidad presentado puede ser adaptado a contextos emergentes como *edge computing*, *fog computing*, arquitecturas *serverless* o entornos *multicloud*. Estos escenarios presentan características particulares —distribución extrema, latencias ultrabajas, dependencia de redes 5G, variabilidad geográfica, cargas efímeras— que podrían requerir atributos de calidad adicionales, nuevas métricas y relaciones diferentes entre componentes. La adaptación del modelo permitiría comprender mejor cómo se comporta la calidad en estos dominios y validaría su generalización más allá del cloud tradicional.

- **Enriquecimiento de las métricas y fuentes de datos** Aunque la tesis utiliza métricas de QoS clásicas relacionadas con la calidad interna y externa de servicios cloud, existen alternativas que podrían complementar o fortalecer el análisis: métricas de *quality of experience* (QoE) reportadas por usuarios, métricas derivadas de trazas distribuidas (*OpenTelemetry*), logs de aplicaciones, métricas de sostenibilidad (energía, emisiones), métricas económicas (costes por invocación, eficiencia de recursos), métricas de seguridad y resiliencia. Explorar estas fuentes permitiría obtener un modelo más holístico y multimodal de la calidad del servicio.

9.5.2. Evolución de capacidades predictivas

- **Evolución hacia modelos de pronóstico multivariante y multiserie**

  El estudio presentado utiliza un enfoque univariante y de un solo paso (*one-step*). Futuras investigaciones podrían explorar: i) modelos multivariantes que relacionen simultáneamente latencia, throughput, CPU, memoria, I/O y otras métricas de interés, ii) modelos *multi-step* (predicciones a largo plazo), iii) aprendizaje de dependencias cruzadas entre servicios (microservicios). Esto abriría la posibilidad de capturar interacciones complejas entre métricas y producir diagnósticos predictivos más robustos.

- **Incorporación de arquitecturas de última generación para series temporales**

  Modelos recientes como Transformers para series temporales [74], N-BEATS [75], DeepAR [77], TCN [76] y PatchTST [181] han mostrado resultados superiores en distintos benchmarks. Su incorporación permitiría: i) comparar nuevos enfoques con LSTM y BI-GRU, ii) evaluar arquitecturas probabilísticas y deterministas, iii) analizar beneficios en escenarios con más ruido, mayor estacionalidad o mayor horizonte temporal. Además, técnicas como *transfer learning* o preentrenamiento masivo de

series temporales [182] podrían reducir los tiempos de entrenamiento en entornos industriales.

- **Mecanismos de aprendizaje continuo y adaptación del modelo predictivo**

  En entornos cloud la distribución de datos evoluciona con el tiempo debido a cambios de carga, actualizaciones de infraestructura o variaciones en el comportamiento de usuarios. Por ello, sería valioso extender el proceso de modelado de ML con: 1) aprendizaje incremental (*online learning*) [183], ii) detección de *drift* en las series de QoS [184], iii) reentrenamiento automático de los modelos de predicción, iv) actualización dinámica de hiperparámetros. Esto permitiría mantener una precisión estable sin necesidad de construir modelos desde cero.

9.5.3. Integración y transferencia

- **Integración del modelo de calidad en sistemas auto-adaptativos**

  Una línea de investigación natural consiste en combinar el modelo de calidad y el proceso de evaluación con mecanismos de adaptación automática basados en políticas, controladores de retroalimentación o arquitecturas MAPE-K. Esto permitiría cerrar el ciclo monitorización–análisis–decisión–actuación, posibilitando que un sistema ajuste dinámicamente recursos o configuraciones ante desviaciones de QoS. Esta integración tiene un alto valor práctico en sistemas cloud donde la predicción de calidad podría desencadenar acciones proactivas.

- **Evaluaciones a gran escala y replicación en múltiples organizaciones**

  El caso industrial utilizado valida el enfoque, pero futuras investigaciones pueden ampliar la robustez externa mediante: i) evaluaciones en múltiples proveedores cloud (AWS, Azure, GCP), ii) diferentes tipos de aplicaciones (IoT, e-learning), iii) distintos niveles de carga y comportamiento, iv) validación longitudinal durante semanas o meses. Este tipo de estudios consolidaría la generalización del modelo y la validez del proceso predictivo.

- **Desarrollo de herramientas de software basadas en el marco propuesto**

  Finalmente, una línea con alto impacto práctico consiste en desarrollar: i) un prototipo o *framework* que implemente el modelo de calidad, ii) *dashboards* de evaluación y pronóstico, integraciones con plataformas de monitorización (Prometheus, Grafana, DataDog), iv) APIs para desplegar los modelos predictivos en producción, v) recomendaciones automáticas basadas en las predicciones (alertas, optimizaciones, escalado). Un sistema de este tipo permitiría la transferencia directa de esta tesis a organizaciones que gestionan servicios cloud.

9.6. Difusión de resultados

La difusión de los resultados obtenidos en esta tesis ha constituido un componente esencial del trabajo, permitiendo validar las contribuciones, compartir los avances con la comunidad científica y favorecer su transferencia hacia entornos profesionales. A continuación, se describen los diferentes resultados obtenidos durante el desarrollo de la investigación.

9.6.1. Revistas internacionales con índice de calidad relativo

- X. Guerron, S. Abrahao, E. Insfran, M. Fernandez-Diego, F. Gonzalez-Ladron-De-Guevara, *A Taxonomy of Quality Metrics for Cloud Services*, IEEE Access 8 (2020) 131461–131498. doi:10.1109/ACCESS.2020.3009079. **Impact Factor 3.36 (JCR 2020)**.

9.6.1.1. En proceso de revisión

- X. Guerron, S. Abrahao, M. Fernandez-Diego, E. Insfran, S. Vegas, *Building and Validating Deep Learning Models for Forecasting the Quality of Cloud Services*, Automated Software Engineering, 26 November 2025, PREPRINT (Version 1) available at Research Square https://doi.org/10.21203/rs.3.rs-7829797/v1. **Impact Factor 3.1 (JCR 2024)**.

9.6.2. Actas de congresos nacionales

- X. Guerron, S. Abrahao, E. Insfran: *Métricas de Calidad Interna y Externa para Servicios Cloud: Un Mapeo Sistemático*. In: Pérez-Benedí, J. (ed.) Actas de las XXIV Jornadas de Ingeniería del Software y Bases de Datos (**JISBD 2019**). Sistedes (2019). https://hdl.handle.net/11705/JISBD/2019/074

- X. Guerron, S. Abrahao, E. Insfran, M. Fernandez-Diego, E. Díaz, : *Comparación de la precisión de modelos LSTM para la predicción de calidad de servicios cloud: Un cuasi-experimento*. In: Rodríguez Luaces, M. A. (ed.) Actas de las XXVIII Jornadas de Ingeniería del Software y Bases de Datos (**JISBD 2024**). Sistedes (2024). https://hdl.handle.net/11705/JISBD/2024/104

9.6.3. Capítulos de libro

- X. Guerron, S. Abrahao, E. Insfran, C. Quer (2022). *Un modelo de calidad para servicios cloud alineado con el estándar ISO/IEC 25010*. En RA-MA. Calidad y Sostenibilidad de Sistemas de Información en la Práctica.

- P. Cedillo, S. Abrahao, E. Insfran, X. Guerron (2022). *Un caso práctico de monitorización de calidad de servicios cloud*. En RA-MA. Calidad y Sostenibilidad de Sistemas de Información en la Práctica.

9.6.4. Trabajos en curso

- S. Abrahao, X. Guerron, E. Insfran, J. Vanderdonckt, *A Survey of Quality of Experience Metrics for Cloud Services*, ACM Computing Surveys (CSUR) Impact Factor 3.7 (JCR 2023)
- X. Guerron, S. Abrahao, M. Fernandez-Diego, E. Insfran, *Operationalized quality model for cloud services: A model at runtime-based approach*, Journal of Software System, Impact Factor 4.1 (JCR 2024)

Bibliografía

[1] ISO/IEC, Iso/iec 25010:2011 systems and software quality requirements and evaluation (square) - system and software quality models, Tech. rep., International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) (2011).

[2] X. Guerron, S. Abrahao, E. Insfran, M. Fernandez-Diego, F. Gonzalez-Ladron-De-Guevara, A Taxonomy of Quality Metrics for Cloud Services, IEEE Access 8 (2020) 131461–131498. doi:10.1109/ACCESS.2020.3009079.

[3] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, T. Zimmermann, Software engineering for machine learning: A case study, Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019 (2019) 291–300.

[4] S. Vegas, S. Elbaum, Pitfalls in Experiments with DNN4SE: An Analysis of the State of the Practice, in: ESEC/FSE 2023 - Proceedings of the 31st ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 528–540.

[5] J. Rhoton, R. Haukioja, Cloud Computing Architected, British Library Cataloguing-in-Publication Data, 2013.

[6] International Organization for Standardization (ISO), ISO/IEC 20000-1:2018. Information technology – Service management – Part 1: Service management system requirements, Tech. Rep. 20000-1:2018, International Organization for Standardization (ISO) (2018).
URL https://www.iso.org/standard/72361.html

[7] S. Wagner, K. Lochmann, S. Winter, A. Goeb, M. Klaes, Quality models in practice: A preliminary analysis, in: 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, IEEE, 2009, pp. 464–467. doi:10.1109/ESEM.2009.5316003.

[8] J. Y. Lee, J. W. Lee, D. W. Cheun, S. D. Kim, A Quality Model for Evaluating Software-as-a-Service in Cloud Computing, in: Software Engineering Research, Ma-

nagement and Applications, 2009. SERA '09. 7th ACIS International Conference on, 2009, pp. 261–266. doi:10.1109/SERA.2009.43.

[9] P. X. Wen, L. Dong, Quality model for evaluating SaaS service, in: Proceedings - 4th International Conference on Emerging Intelligent Data and Web Technologies, EIDWT 2013, IEEE, 2013, pp. 83–87. doi:10.1109/EIDWT.2013.19.

[10] X. Zheng, P. Martin, K. Brohman, L. D. Xu, CLOUDQUAL: A Quality Model for Cloud Services, IEEE Transactions on Industrial Informatics 10 (2) (2014) 1527–1536. doi:10.1109/TII.2014.2306329.

[11] P. Zhou, Z. Wang, W. Li, N. Jiang, Quality Model of Cloud Service, in: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, IEEE, 2015, pp. 1418–1423. doi:10.1109/HPCC-CSS-ICESS.2015.134.

[12] T. C. S. M. I. C. (CSMIC), Service measurement index framework version 2.1, Tech. rep., The Cloud Service Measurement Initiative Consortium (CSMIC) (2014).
URL http://www.cloudcommons.com/about-smi

[13] F. de Vaulx, E. Simmon, R. Bohn, NIST SP 500-307 Cloud Computing Service Metrics Description, Tech. rep., National Institute of Standards and Technology (2018). doi:10.6028/NIST.SP.500-307.

[14] I. O. F. Standardization, Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) – System and software Quality Requirements and Evaluation (SQuaRE)" ISO/IEC 25040:2011, vol. 2, no. Resolution 937, 2011., Tech. Rep. Resolution 937, International Organization for Standardization and International Electrotechnical Commission (2011).

[15] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[16] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (8) (1997) 1735–1780.

[17] G. White, A. Palade, S. Clarke, Forecasting QoS Attributes Using LSTM Networks, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8. doi:10.1109/IJCNN.2018.8489052.

[18] Y. Syu, C. Wang, Y. Fanjiang, Modeling and forecasting of time-aware dynamic qos attributes for cloud services, IEEE Transactions on Network and Service Management 16 (2019) 56–71.

[19] J. Siebert, L. Joeckel, J. Heidrich, A. Trendowicz, K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, M. Aoyama, Construction of a quality model for machine learning systems, Software Quality Journal 30 (2) (2022) 307–335.

[20] C. Calero, M. Moraga, M. F. Bertoa, Towards a software product sustainability model, arXiv preprint arXiv:1309.1640 (2013).

[21] S. Wagner, A. Goeb, L. Heinemann, M. Klas, C. Lampasona, K. Lochmann, A. Mayr, R. Plosch, A. Seidl, J. Streit, A. Trendowicz, Operationalised product quality models and assessment: The Quamoco approach, Information and Software Technology 62 (1) (2015) 101–123. arXiv:1611.09230, doi:10.1016/j.infsof.2015.02.009.

[22] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference (2014) 1724–1734doi:10.3115/v1/d14-1179.

[23] Y. Syu, C. M. Wang, Qos time series modeling and forecasting for web services: A comprehensive survey, IEEE Transactions on Network and Service Management 18 (2021) 926–944.

[24] B. Cavallo, M. D. Penta, G. Canfora, An empirical comparison of methods to support qos-aware service selection, in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE), Association for Computing Machinery, Cape Town, South Africa, 2010, pp. 210–220. doi:10.1145/1806799.1806831.
URL https://doi.org/10.1145/1806799.1806831

[25] T. Gorschek, P. Garre, S. Larsson, C. Wohlin, A model for technology transfer in practice, IEEE Software 23 (2006) 88–95. doi:10.1109/MS.2006.147.

[26] B. Kitchenham, S. M. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. rep., EBSE (2007).

[27] K. M. Petersen, R. Feldt, S. Mujtaba, Mattsson, Systematic mapping studies in software engineering, Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (2008) 68–77.

[28] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering - a systematic literature review, Information and Software Technology 51 (2009) 7–15. doi:10.1016/j.infsof.2008.09.009.

[29] D. S. Cruzes, T. Dybå, Recommended steps for thematic synthesis in software engineering, in: International Symposium on Empirical Software Engineering and Measurement, IEEE, 2011, pp. 275–284. doi:10.1109/esem.2011.36.
URL http://ieeexplore.ieee.org/document/6092576/

[30] P. Mell, T. Grance, NIST SP 800-145 Definition of Cloud Computing, Tech. rep., National Institute of Standards and Technology (2011). doi:10.6028/NIST.SP.800-145.

[31] L. Badger, T. Grance, R. Patt-Corner, J. Voas, NIST 800-146 Cloud Computing Synopsis and Recommendations, Tech. rep., National Institute of Standards and Technology (2012). doi:10.6028/NIST.SP.800-146.

[32] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, D. Leaf, NIST 500-292 Cloud Computing Reference Architecture, Tech. rep., National Institute of Standards and Technology (2011).

[33] ISO/IEC, ISO/IEC 17789:2014 Information Technology — Cloud Computing — Reference Architecture, Tech. rep., International Organization for Standardization / International Electrotechnical Commission (2014).

[34] X. Franch, J. P. Carvallo, Unsing quality models in software package selection, IEEE Software (2003) 34–41doi:10.3139/9783446453852.013.

[35] R. G. Dromey, SOFTWARE PRODUCT QUALITY : Theory , Model , and Practice, Software Quality Institute, Griffith University, Nathan, Brisbane, QLD 4111 Australia (1998).

[36] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Vol. 9783642290442, Springer, 2012.

[37] P. Runeson, M. Höst, A. Rainer, B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples, John Wiley & Sons, 2012.

[38] C. Robson, Real world research, Vol. 2, Blackwell Oxford, 2002.

[39] R. K. Yin, Case study research: Design and methods, Vol. 5, sage, 2009.

[40] I. Benbasat, D. K. Goldstein, M. Mead, The case research strategy in studies of information systems, MIS quarterly (1987) 369–386.

[41] F. D. Davis, Perceived usefulness, perceived ease of use and user acceptance of information technology, MIS quarterly (1989).

[42] C. B. Seaman, Qualitative methods in empirical studies of software engineering, IEEE Transactions on software engineering 25 (4) (1999) 557–572.

[43] V. Basili, G. Caldiera, D. Rombach, The goal question metric approach, Encyclopedia of Software Engineering., 1994.

[44] Ideogram AI, Ideogram, [Online]. Available: https://ideogram.ai. Accessed: 15-Dec-2025 (2025).
URL https://ideogram.ai

[45] R. Van der Meulen, P. Christy, Gartner Says Cloud Computing Will Be As Influential As E-business (2008).

[46] F. Gens, Defining Cloud Services and Cloud Computing (2008).

[47] ISO/IEC, ISO/IEC 17788:2014 Information technology - Cloud computing - Overview and vocabulary, Tech. rep., International Organization for Standardization/ International Electrotechnical Commission (2014).

[48] M. F. Rahman, Serverless cloud computing: a comparative analysis of performance, cost, and developer experiences in container-level services, Aalto University School of Science (2023).

[49] C. Pahl, P. Jamshidi, O. Zimmermann, Architectural principles for cloud software, ACM Transactions on Internet Technology 18 (2) (2018). doi:10.1145/3104028.

[50] D. Bernstein, Containers and cloud: From LXC to docker to kubernetes, IEEE Cloud Computing 1 (3) (2014) 81–84. doi:10.1109/MCC.2014.51.

[51] C. N. C. Foundation, CNCF Survey 2020, Tech. rep., Cloud Native Computing Foundation (CNCF) (2020).

[52] I. O. for Standardization, Iso 8402: 1994: Quality management and quality assurance-vocabulary, Tech. rep., International Organization for Standardization (1994).

[53] IEEE, Ieee standard glossary of software engineering terminology, Standard IEEE Std 729-1983, IEEE (3 1983). doi:10.1109/IEEESTD.1983.1156879.
URL https://ieeexplore.ieee.org/document/1156879

[54] D. of Defense, Dod-std-2168 defense system software quality program, Tech. Rep. August 1988, Department of Defense (1988).

[55] ISO/IEC, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE, Tech. Rep. ISO/IEC 25000:2020, International Organization for Standardization and International Electrotechnical Commission (2011).

[56] ISO/IEC, ISO/IEC 9126: Software Engineering – Product Quality – Part 1: Quality Model, Tech. rep., International Organization for Standardization / International Electrotechnical Commission (6 2001).

[57] ISO/IEC, ISO/IEC 14598-1:2001 Information technology – Software product evaluation – Part 1: General overview, Technical Report ISO/IEC 14598-1, International Organization for Standardization and International Electrotechnical Commission (2001).

[58] S. Wagner, K. Lochmann, S. Winter, A. Göb, M. Kläs, S. Nunnenmacher, Software quality models in practice, Technischer Bericht Institut für Informatik (2012).

[59] ISO/IEC, Iso/iec 25010:2023 systems and software engineering-systems and software quality requirements and evaluation (square)-product quality model, Tech. rep., International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) (2023).
URL www.iso.org

[60] R. J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2018.

[61] V. Cerqueira, L. Torgo, I. Mozetič, Evaluating time series forecasting models: an empirical study on performance estimation methods, Machine Learning 109 (11) (2020) 1997–2028. arXiv:1905.11744.

[62] G. Box, G. Jenkins, Analysis: Forecasting and control, San francisco (1976).

[63] G. Bontempi, S. Taieb, Y. Le Borgne, Machine learning strategies for time series forecasting, in: Lecture Notes in Business Information Processing, Vol. 138 LNBIP, Springer Verlag, 2013, pp. 62–77.

[64] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32. doi:10.1023/A:1010933404324.
URL https://doi.org/10.1023/A:1010933404324

[65] F. Zhang, L. J. O'Donnell, Support vector regression, Machine Learning: Methods and Applications to Brain Disorders (2020) 123–140doi:10.1016/B978-0-12-815739-8.00007-9.

[66] M. Awad, R. Khanna, Efficient learning machines: theories, concepts, and applications for engineers and system designers, Springer nature, 2015.

[67] C.-H. Ho, C.-J. Lin, Large-scale linear support vector regression, Journal of Machine Learning Research 13 (2012) 3323–3348.
URL http://jmlr.org/papers/v13/ho12a.html

[68] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259 (2014).

[69] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (1997) 2673–2681. doi:10.1109/78.650093.

[70] T. Jirsik, S. Trcka, P. Celeda, Quality of service forecasting with lstm neural network, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, 2019, pp. 250–260.

[71] G. Zhu, D. Wang, J. Niu, Quality of service forecasting using optimized lstm networks based on emd, in: Proceedings - 2022 4th International Conference on Natural

Language Processing, ICNLP 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 466–471. doi:10.1109/ICNLP55136.2022.00085.

[72] A. Kaim, S. Singh, Y. S. Patel, Bglmc: A bidirectional gru-based learning model for cloud-computing resource workloads forecasting, in: Cloud of Things, Chapman and Hall/CRC, 2024, pp. 107–124.

[73] S. J. Taylor, B. Letham, Forecasting at scale, PeerJ Computer Science 2 (2018) e99.

[74] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 11106–11115.

[75] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, arXiv preprint arXiv:1905.10437 (2019).

[76] S. Bai, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271 (2018).

[77] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: Probabilistic forecasting with autoregressive recurrent networks, International journal of forecasting 36 (3) (2020) 1181–1191.

[78] Z. Li, C. Rainbow Cai, H. Zhang, Towards a Taxonomy of Performance Evaluation of Commercial Cloud Services, in: IEEE Fifth International Conference on Cloud Computing, 2012, pp. 344–351. doi:10.1109/CLOUD.2012.74.

[79] S. M. Habib, V. Varadharajan, M. Muhlhauser, A trust-aware framework for evaluating security controls of service providers in cloud marketplaces, in: Proceedings - 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013, 2013, pp. 459–468. doi:10.1109/TrustCom.2013.58.

[80] N. Herbst, A. Bauer, S. Kounev, G. Oikonomou, E. V. Eyk, G. Kousiouris, A. Evangelinou, R. Krebs, T. Brecht, C. L. Abad, A. Iosup, Quantifying cloud performance and dependability: Taxonomy, metric design, and emerging challenges, ACM Transactions on Modeling and Performance Evaluation of Computing Systems 3 (8 2018). doi:10.1145/3236332.

[81] F. D. Muñoz-Escoí, J. M. Bernabéu-Aubán, A survey on elasticity management in paas systems, Computing 99 (2017) 617–656. doi:10.1007/s00607-016-0507-8. URL http://link.springer.com/10.1007/s00607-016-0507-8

[82] E. F. Coutinho, F. R. de Carvalho Sousa, P. A. L. Rego, D. G. Gomes, J. N. de Souza, Elasticity in cloud computing: a survey, Annales des Telecommunications/Annals of Telecommunications 70 (2015) 289–309. doi:10.1007/s12243-014-0450-7.

[83] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, P. Merle, Elasticity in cloud computing: State of the art and research challenges, IEEE Transactions on Services Computing 11 (2018) 430–447. doi:10.1109/TSC.2017.2711009.

[84] M. Jelassi, C. Ghazel, L. A. Saidane, L. A. Saïdane, L. A. Saidane, A survey on quality of service in cloud computing, in: 2017 3rd International Conference on Frontiers of Signal Processing (ICFSP), IEEE, 2017, pp. 63–67. doi:10.1109/ICFSP.2017.8097142.

[85] A. K. Bardsiri, S. M. Hashemi, Qos metrics for cloud computing services evaluation, International Journal of Intelligent Systems and Applications 6 (2014) 27–33. doi:10.5815/ijisa.2014.12.04.

[86] Z. Li, L. O'Brien, H. Zhang, R. Cai, On a catalogue of metrics for evaluating commercial cloud services, Proceedings - IEEE/ACM International Workshop on Grid Computing (2012) 164–173doi:10.1109/Grid.2012.15.

[87] Z. Li, H. Zhang, L. O'Brien, R. Cai, S. Flint, On evaluating commercial cloud services: A systematic review, Journal of Systems and Software 86 (2013) 2371–2393. doi:10.1016/j.jss.2013.04.021.
URL http://dx.doi.org/10.1016/j.jss.2013.04.021

[88] A. Abdelmaboud, D. N. Jawawi, I. Ghani, A. Elsafi, B. Kitchenham, Quality of service approaches in cloud computing: A systematic mapping study, Journal of Systems and Software 101 (2015) 159–179, from Duplicate 2 (Quality of service approaches in cloud computing: A systematic mapping study - Abdelmaboud, Abdelzahir; Jawawi, Dayang N.A.; Ghani, Imran; Elsafi, Abubakar; Kitchenham, Barbara)importante. doi:10.1016/j.jss.2014.12.015.
URL http://dx.doi.org/10.1016/j.jss.2014.12.015

[89] S. Lehrig, H. Eikerling, S. Becker, Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics, QoSA 2015 - Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures, Part of CompArch 2015 (2015) 83–92doi:10.1145/2737182.2737185.

[90] J. Scheuner, P. Leitner, Function-as-a-service performance evaluation: A multivocal literature review, Journal of Systems and Software 170 (2020) 110708. doi:10.1016/j.jss.2020.110708.
URL http://dx.doi.org/10.1016/j.jss.2020.110708

[91] M. H. Kashani, A. M. Rahmani, N. J. Navimipour, Quality of service-aware approaches in fog computing, International Journal of Communication Systems 33 (2020) 1–34. doi:10.1002/dac.4340.

[92] T. Olokunde, S. Misra, A. Adewumi, Quality model for evaluating platform as a service in cloud computing, in: Communications in Computer and Information Science, Vol. 756, 2017, pp. 280–291.

[93] H. Hourani, R. Dawood, L. Sadaqa, M. Abdallah, A. Tamimi, A proposed cloud computing quality model: S3MQUAL (Service measurement metrics model), ICIC Express Letters 13 (11) (2019) 1005–1014. doi:10.24507/icicel.13.11.1005.

[94] R. Lichtenthäler, G. Wirtz, Towards a quality model for cloud-native applications, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 13226 LNCS (2022) 109–117.

[95] C. Giovanoli, Cloud service quality model: A cloud service quality model based on customer and provider perceptions for cloud service mediation, CLOSER 2019 - Proceedings of the 9th International Conference on Cloud Computing and Services Science (2019) 241–248doi:10.5220/0007587502410248.

[96] M. Almorsy, J. Grundy, A. S. Ibrahim, Adaptable, model-driven security engineering for SaaS cloud-based applications, Autom Softw Eng 21 (2) (2014) 187–224. doi:10.1007/s10515-013-0133-z.

[97] T. Chen, R. Bahsoon, Self-Adaptive and online QoS modeling for cloud-based software services, IEEE Transactions on Software Engineering 43 (5) (2017) 453–475. doi:10.1109/TSE.2016.2608826.

[98] J. Shao, H. Wei, Q. Wang, H. Mei, A runtime model based monitoring approach for cloud, in: Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010, Miami, 2010, pp. 313–320. doi:10.1109/CLOUD.2010.31.

[99] P. Cedillo, J. Gonzalez-Huerta, S. Abrahao, E. Insfran, Towards monitoring cloud services using Models@run.time, in: CEUR Workshop Proceedings, Vol. 1270, 2014, pp. 31–40.

[100] P. Cedillo, J. Jimenez-Gomez, S. Abrahao, E. Insfran, Towards a Monitoring Middleware for Cloud Services, in: 2015 IEEE International Conference on Services Computing, IEEE, New York, 2015, pp. 451–458. doi:10.1109/SCC.2015.68.
URL http://ieeexplore.ieee.org/document/7207386/

[101] P. Cedillo, E. Insfran, S. Abrahão, Monitoring cloud services through models at runtime: A case in an ambient assisted living environment, Journal of Object Technology 21 (2022). doi:10.5381/jot.2022.21.04.a1.

[102] W. Hussain, F. Hussain, M. H. O. H. Saberi, E. Chang, Comparing time series with machine learning-based prediction approaches for violation management in cloud slas, Future Generation Computer Systems 89 (2018) 464–477.

[103] M. Petticrew, H. Roberts, Systematic Reviews in the Social Sciences: A Practical Guide, Blackwell Publishing, 2008. doi:10.1002/9780470754887.

[104] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting empirical methods for software engineering research, in: Guide to advanced empirical software engineering, Springer, 2008, pp. 285–311.

[105] N. E. Fenton, J. M. Bieman, Software Metrics: A Rigorous and Practical Approach, 3rd Edition, Taylor and Francis, 2014.

[106] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software 80 (2007) 571–583. doi:10.1016/j.jss.2006.07.009.

[107] L. Chen, M. A. Babar, H. N. Zhang, Towards an evidence-based understanding of electronic data sources, in: 14th International Conference on Evaluation and Assessment in Software Engineering (EASE), 2010, pp. 1–4. doi:10.14236/ewic/EASE2010.17.

[108] T. F. de C. Marshall, J. L. Fleiss, Statistical methods for rates and proportions., The Statistician 25 (1981) 70. doi:10.2307/2988144.
URL https://www.jstor.org/stable/10.2307/2988144?origin=crossref

[109] S. Montagud, S. Abrahão, E. Insfran, A systematic review of quality attributes and measures for software product lines, Software Quality Journal 20 (2012) 425–486. doi:10.1007/s11219-011-9146-7.

[110] F. Garcia, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini, M. Genero, Towards a consistent terminology for software measurement, Information and Software Technology 48 (8) (2006) 631–644. doi:10.1016/j.infsof.2005.07.001.

[111] ISO/IEC, Systems and software engineering — measurement process iso/iec 15939:2007, Tech. rep., International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) (2007).

[112] I. BiPM, I. IFCC, I. IUPAC, O. ISO, The international vocabulary of metrology—basic and general concepts and associated terms (vim), JcGM 200 (2012) (2012).

[113] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, V. Stankovski, Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review, Journal of Systems and Software 136 (2018) 19–38. doi:https://doi.org/10.1016/j.jss.2017.10.033.

[114] P. Nadanam, R. Rajmohan, Qos evaluation for web services in cloud computing, in: 2012 3rd International Conference on Computing, Communication and Networking

Technologies, ICCCNT 2012, IEEE, 2012, pp. 1–8. doi:10.1109/ICCCNT.2012.6395991.

[115] A. Ezenwoke, O. Daramola, M. Adigun, Qos-based ranking and selection of saas applications using heterogeneous similarity metrics, Journal of Cloud Computing 7 (2018) 15. doi:10.1186/s13677-018-0117-4.

[116] G. Baranwal, D. P. Vidyarthi, A cloud service selection model using improved ranked voting method, Concurrency Computation 28 (2016) 3540–3567. doi:10.1002/cpe.3740.
URL http://doi.wiley.com/10.1002/cpe.3740

[117] S. Schneider, A. Sunyaev, Cloudlive: a life cycle framework for cloud services, Electronic Markets 25 (2015) 299–311. doi:10.1007/s12525-015-0205-y.

[118] L. Briand, K. E. Emam, S. Morasca, On the application of measurement theory in software engineering, Empirical Software Engineering 1 (1996) 61–88. doi:10.1007/BF00125812.

[119] S. A. Whitmire, Object-oriented design measurement (1997) 452.

[120] G. Poels, G. Dedene, Distance-based software measurement: necessary and sufficient properties for software measures, Information and Software Technology 42 (2000) 35–46. doi:10.1016/S0950-5849(99)00053-1.

[121] E. Ltd., Mendeley, https://www.mendeley.com/search/, accessed: 2025-06-11 (2025).

[122] L. Lu, Y. Yuan, A novel topsis evaluation scheme for cloud service trustworthiness combining objective and subjective aspects, Journal of Systems and Software 143 (2018) 71–86. doi:10.1016/j.jss.2018.05.004.
URL https://linkinghub.elsevier.com/retrieve/pii/S016412121830092X

[123] D. Preuveneers, T. Heyman, Y. Berbers, W. Joosen, Systematic scalability assessment for feature oriented multi-tenant services, Journal of Systems and Software 116 (2016) 162–176. doi:10.1016/j.jss.2015.12.024.

[124] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, Y. Wu, Cloud performance modeling with benchmark evaluation of elastic scaling strategies, IEEE Transactions on Parallel and Distributed Systems 27 (2016) 130–143. doi:10.1109/TPDS.2015.2398438.
URL http://ieeexplore.ieee.org/document/7027861/

[125] S. S. Wagle, M. Guzek, P. Bouvry, R. Bisdorff, An evaluation model for selecting cloud services from commercially available cloud providers, in: 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, 2015, pp. 107–114. doi:10.1109/CloudCom.2015.94.
URL http://ieeexplore.ieee.org/document/7396144/

[126] S. Singh, I. Chana, Q-aware: Quality of service based cloud resource provisioning, Computers and Electrical Engineering 47 (2015) 138–160. doi:10.1016/j.compeleceng.2015.02.003.

[127] T. Guerout, S. Medjiah, G. Da Costa, T. Monteil, Quality of service modeling for green scheduling in Clouds, Sustainable Computing: Informatics and Systems 4 (4) (2014) 225–240. doi:10.1016/J.SUSCOM.2014.08.006.

[128] H. Zhang, M. A. Babar, P. Tell, Identifying relevant studies in software engineering, Information and Software Technology 53 (2011) 625–637. doi:10.1016/j.infsof.2010.12.010.
URL http://dx.doi.org/10.1016/j.infsof.2010.12.010

[129] G. Baranwal, D. P. Vidyarthi, A framework for selection of best cloud service provider using ranked voting method, in: Advance Computing Conference (IACC), 2014 IEEE International, 2014, pp. 831–837. doi:10.1109/IAdCC.2014.6779430.

[130] S. Rizvi, H. Roddy, J. Gualdoni, I. Myzyri, Three-step approach to qos maintenance in cloud computing using a third-party auditor, Procedia Computer Science 114 (2017) 83–92. doi:10.1016/j.procs.2017.09.014.
URL https://linkinghub.elsevier.com/retrieve/pii/S1877050917318082

[131] S. K. Garg, S. Versteeg, R. Buyya, A framework for ranking of cloud computing services, Future Generation Computer Systems 29 (4) (2013) 1012–1023.

[132] R. H. de Souza, P. A. Flores, M. A. R. Dantas, F. Siqueira, Architectural recovering model for distributed databases: A reliability, availability and serviceability approach, in: 2016 IEEE Symposium on Computers and Communication (ISCC), 2016, pp. 575–580. doi:10.1109/ISCC.2016.7543799.

[133] Y. Hu, B. Deng, Y. Yang, D. Wang, Elasticity evaluation of iaas cloud based on mixed workloads, in: 2016 15th International Symposium on Parallel and Distributed Computing (ISPDC), IEEE, 2016, pp. 157–164. doi:10.1109/ISPDC.2016.28.

[134] W. Dou, X. Xu, S. Meng, S. Yu, An energy-aware qos enhanced method for service computing across clouds and data centers, in: 3rd International Conference on Advanced Cloud and Big Data, CBD 2015, 2015, pp. 80–87, cited By :1Export Date: 23 October 2018. doi:10.1109/CBD.2015.23.

[135] N. R. Herbst, S. Kounev, R. Reussner, Elasticity in cloud computing: What it is, and what it is not, in: J. O. Kephart, C. Pu, X. Zhu (Eds.), Proceedings of the 10th International Conference on Autonomic Computing (ICAC '13), USENIX Association, San Jose, CA, USA, 2013, pp. 23–27.

[136] P. Manuel, A trust model of cloud computing based on quality of service, Annals of Operations Research 233 (2015) 281–292. doi:10.1007/s10479-013-1380-x.
URL http://link.springer.com/10.1007/s10479-013-1380-x

[137] X. Liu, C. Xia, T. Wang, L. Zhong, Cloudsec: A novel approach to verifying security conformance at the bottom of the cloud, in: 2017 IEEE International Congress on Big Data (BigData Congress), 2017, pp. 569–576. doi:10.1109/BigDataCongress.2017.87.

[138] Y. K. Lin, P. C. Chang, Maintenance reliability estimation for a cloud computing network with nodes failure, Expert Systems with Applications 38 (2011) 14185–14189. doi:10.1016/j.eswa.2011.04.230.
URL http://dx.doi.org/10.1016/j.eswa.2011.04.230

[139] C. S. C. Council, Interoperability and Portability for Cloud Computing : A Guide, Tech. rep., Cloud Standards Customer Council (2017).

[140] B. Stanton, M. Theofanos, K. P. Joshi, Framework for Cloud Usability (2015).

[141] ISO/IEC, ISO/IEC 19941 - Cloud computing-Interoperability and portability, Tech. rep., International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) (2017).

[142] E. Lim, P. Thiran, Communication of technical qos among cloud brokers, Proceedings - 2014 IEEE International Conference on Cloud Engineering, IC2E 2014 (2014) 403–409doi:10.1109/IC2E.2014.92.
URL http://ieeexplore.ieee.org/document/6903503/

[143] A. Abdeladim, S. Baina, K. Baina, Elasticity and scalability centric quality model for the cloud, Colloquium in Information Science and Technology, CIST 2015-Janua (2015) 135–140. doi:10.1109/CIST.2014.7016607.
URL http://ieeexplore.ieee.org/document/7016607/

[144] N. I. of Standards, Technology, Nist sp 500-291 cloud computing standards roadmap, Tech. rep., National Institute of Standards and Technology (2013). doi:10.6028/NIST.SP.500-291r2.
URL http://dx.doi.org/10.6028/NIST.SP.500-291r2

[145] E. Saiz, E. Ibarrola, L. Cristobo, I. Taboada, A cloud platform for qoe evaluation: Qoxcloud, Proceedings of the 2014 ITU Kaleidoscope Academic Conference: Living in a Converged World - Impossible Without Standards?, K 2014 (2014) 241–247doi:10.1109/Kaleidoscope.2014.6858471.

[146] S. Wang, S. Dey, Cloud mobile gaming: Modeling and measuring user experience in mobile wireless networks, ACM SIGMOBILE Mobile Computing and Communications Review 16 (2012) 10. doi:10.1145/2331675.2331679.
URL http://dl.acm.org/citation.cfm?doid=2331675.2331679

[147] Z.-Y. Wen, H.-F. Hsiao, Qoe-driven performance analysis of cloud gaming services, in: 2014 IEEE 16th International Workshop on Multimedia Signal Processing

(MMSP), IEEE, 2014, pp. 1–6. doi:10.1109/MMSP.2014.6958835.
URL http://ieeexplore.ieee.org/document/6958835/

[148] IEEE, IEEE standard for a software quality metrics methodology, IEEE std 1061-1998, Tech. rep., IEEE, New York, NY (1998).

[149] S. Singh, I. Chana, R. Buyya, Star: Sla-aware autonomic management of cloud resources, IEEE Transactions on Cloud Computing 8 (4) (2020) 1040–1053.

[150] Redis. Redis [online] (2023).

[151] Mongo. Mongodb: The developer data platform — mongodb [online] (2023).

[152] PostgreSQL. Postgresql: The world's most advanced open source database [online] (2023).

[153] Linode. Cloud computing services — vms, kubernetes, storage — linode [online] (2023).

[154] D. Armstrong, K. Djemame, Towards quality of service in the cloud, in: Proc. of the 25th UK Performance Engineering Workshop, 2009.

[155] S. A. Baset, Cloud slas: Present and future, SIGOPS Oper. Syst. Rev. 46 (2) (2012) 57–66.

[156] M. Comuzzi, G. Jacobs, P. Grefen, Understanding SLA elements in cloud computing, in: IFIP Advances in Information and Communication Technology, Vol. 408, Technische Universiteit Eindhoven, 2013, pp. 385–392.

[157] M. H. Ghahramani, M. Zhou, C. T. Hon, Toward cloud computing QoS architecture: analysis of cloud systems and cloud services, IEEE/CAA Journal of Automatica Sinica 4 (1) (2017) 6–18.

[158] X. Guerron, S. Abrahao, E. Insfran, C. Quer, Un modelo de calidad para servicios cloud alineado con el estándar iso/iec 25010, in: Calidad y Sostenibilidad de Sistemas de Información en la Práctica, RA-MA, 2022.

[159] Jefatura del Estado (España), Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales, publicada en el BOE núm. 294, de 06/12/2018. Entrada en vigor: 7 de diciembre de 2018 (Diciembre 2018).
URL https://www.boe.es/eli/es/lo/2018/12/05/3

[160] Parlamento Europeo y del Consejo de la Unión Europea, Reglamento (ue) 2016/679 del parlamento europeo y del consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la directiva 95/46/ce (reglamento general de protección de datos), entrada en vigor: 25 de mayo de 2018 (Abril 2016).
URL https://eur-lex.europa.eu/eli/reg/2016/679/oj

[161] M. Fowler, J. Highsmith, et al., The agile manifesto, Software development 9 (8) (2001) 28–35.

[162] O. Gheibi, D. Weyns, F. Quin, Applying Machine Learning in Self-Adaptive Systems: A Systematic Literature Review, in: ACM Transactions on Autonomous and Adaptive Systems, 2021, p. 37. arXiv:2103.04112.

[163] V. Chandola, A. Banerjee, V. Kumar, Encyclopedia of Machine Learning and Data Science, 2020.

[164] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow THIRD EDITION Concepts, Tools, and Techniques to Build Intelligent Systems, 2023.

[165] F. Chollet, Deep learning with Python, Simon and Schuster, 2021.

[166] C. Bergmeir, J. Benítez, On the use of cross-validation for time series predictor evaluation, Information Sciences 191 (2012) 192–213.

[167] L. Tashman, Out-of-sample tests of forecasting accuracy: An analysis and review, International Journal of Forecasting 16 (4) (2000) 437–450.

[168] T. Mitchell, Machine Learning, McGraw Hill, 1997.

[169] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, Journal of Machine Learning Research 13 (Feb) (2012) 281–305.

[170] C. Bishop, N. Nasrabadi, Pattern Recognition and Machine Learning, 4th Edition, Springer New York, 2006.

[171] B. A. Kitchenham, D. Budgen, P. Brereton, Evidence-based software engineering and systematic reviews, CRC press, 2015.

[172] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends® in Information Retrieval 2 (1–2) (2008) 1–135. doi:10.1561/1500000011.
URL http://dx.doi.org/10.1561/1500000011

[173] S. Siami-Namini, N. Tavakoli, A. Siami Namin, A comparison of arima and lstm in forecasting time series, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394–1401.

[174] Newrelic. New relic — monitor, debug and improve your entire stack [online] (2023).

[175] T. Fischer, C. Krauss, A. Treichel, Machine learning for time series forecasting - a simulation study, FAU Discussion Papers in Economics (02) (2018) No. 02/2018.

[176] Google, Google colaboratory, https://colab.research.google.com/ (2025).

[177] N. Srivastava, G. Hinton, A. Krizhevsky, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (2014) 1929–1958.

[178] M. R. Hess, J. D. Kromrey, Robust confidence intervals for effect sizes: A comparative study of cohen'sd and cliff's delta under non-normality and heterogeneous variances, in: annual meeting of the American Educational Research Association, Vol. 1, Citeseer, 2004.

[179] M. J. Anderson, A new method for non-parametric multivariate analysis of variance, Austral Ecology 26 (1) (2001) 32–46. doi:10.1046/j.1442-9993.2001.01070.x.

[180] S. Taieb, G. Bontempi, A. Atiya, A. Sorjamaa, A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition, Expert Systems with Applications 39 (8) (2011) 7067–7083. arXiv:1108.3259.

[181] Y. Nie, Z. Han, H. Dan, A time series is worth 64 points: Transformers for time-series forecasting, in: Advances in Neural Information Processing Systems, 2023.

[182] J. Yoon, et al., Ts2vec: Towards universal representation of time series, in: AAAI, 2022.

[183] S. C. Hoi, D. Sahoo, J. Lu, P. Zhao, Online learning: A comprehensive survey, Neurocomputing 459 (2021) 249–289.

[184] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, IEEE transactions on knowledge and data engineering 31 (12) (2018) 2346–2363.

Anexo A

Anexo 1. Estudios primarios de la Taxonomía de métricas

En este apéndice se presentan los estudios primarios que respaldan la revisión sistemática de la literatura expuesta en el capítulo 4. Se puede consultar para revisar los estudios primarios resultantes del proceso de selección, ordenados alfabéticamente por autores,

1. Abd, S. K., Al-Haddad, S. A. R., Hashim, F., Abdullah, A. B. H. J., & Yussof, S. (2017). An effective approach for managing power consumption in cloud computing infrastructure. Journal of Computational Science, 21, 349–360. /https://doi.org/10.1016/j.jocs.2016.11.007

2. Abdeladim, A., Baina, S., & Baina, K. (2014). Elasticity and scalability centric quality model for the cloud. In 2014 Third IEEE International Colloquium in Information Science and Technology (CIST) (pp. 135–140). https://doi.org/10.1109/CIST.2014.7016607

3. Abrahão, S., & Insfran, E. (2017). Models@runtime for Monitoring Cloud Services in Google App Engine. In 2017 IEEE World Congress on Services (SERVICES) (pp. 30–35). https://doi.org/10.1109/SERVICES.2017.14

4. Alam, A. F. B., Soltanian, A., Yangui, S., Salahuddin, M. A., Glitho, R., & Elbiaze, H. (2016). A Cloud Platform-as-a-Service for multimedia conferencing service provisioning. In 2016 IEEE Symposium on Computers and Communication (ISCC) (pp. 289–294). https://doi.org/10.1109/ISCC.2016.7543756

5. Al-Jawad, A., Trestian, R., Shah, P., & Gemikonakli, O. (2015). BaProbSDN: A probabilistic-based QoS routing mechanism for Software Defined Networks. In Network Softwarization (NetSoft), 2015 1st IEEE Conference on (pp. 1–5). https://doi.org/10.1109/NETSOFT.2015.7116128

6. de Oliveira Jr., F. A., & Ledoux, T. (2011). Self-management of Applications QoS for Energy Optimization in Datacenters. In Green Computing Middleware on Pro-

ceedings of the 2Nd International Workshop (pp. 3:1–3:6). New York, NY, USA: ACM. doi:10.1145/2088996.2088999

7. Arumugam, K., & Sumathi, P. (2017). Secure and QoS guaranteed selection resource for storing health care information of cloud users. In 2017 International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1165–1170). https://doi.org/10.1109/ICCMC.2017.8282657

8. Bao, D., Xiao, Z., Sun, Y., & Zhao, J. (2010). A method and framework for quality of cloud services measurement. In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE) (Vol. 5, pp. V5–358–V5–362). https://doi.org/10.1109/ICACTE.2010.5579535

9. Baranwal, G., & Vidyarthi, D. P. (2014). A framework for selection of best cloud service provider using ranked voting method. In Advance Computing Conference (IACC), 2014 IEEE International (pp. 831–837). https://doi.org/10.1109/IAdCC.2014.6779430

10. Baranwal, G., & Vidyarthi, D. P. (2016). A cloud service selection model using improved ranked voting method. Concurrency and Computation: Practice and Experience, 28(13), 3540–3567. https://doi.org/10.1002/cpe.3740

11. Barba-Jimenez, C., Ramirez-Velarde, R., Tchernykh, A., Rodríguez-Dagnino, R., Nolazco-Flores, J., & Perez-Cazares, R. (2016). Cloud based Video-on-Demand service model ensuring quality of service and scalability. Journal of Network and Computer Applications, 70, 102–113. https://doi.org/10.1016/j.jnca.2016.05.007

12. Bardhan, S., & Milojicic, D. (2012). A Mechanism to Measure Quality-of-service in a Federated Cloud Environment. In Proceedings of the 2012 Workshop on Cloud Services, Federation, and the 8th Open Cirrus Summit (pp. 19–24). New York, NY, USA: ACM. doi:10.1145/2378975.2378981

13. Bousselmi, K., Brahmi, Z., & Gammoudi, M. M. (2016). QoS-Aware Scheduling of Workflows in Cloud Computing Environments. In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA) (pp. 737–745). https://doi.org/10.1109/AINA.2016.72

14. Bruneo, D. (2014). A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems. IEEE Transactions on Parallel and Distributed Systems, 25(3), 560–569. https://doi.org/10.1109/TPDS.2013.67

15. Cedillo, P., Jimenez-Gomez, J., Abrahao, S., & Insfran, E. (2015). Towards a Monitoring Middleware for Cloud Services. In Services Computing (SCC), 2015 IEEE International Conference on (pp. 451–458). https://doi.org/10.1109/SCC.2015.68

16. Cervino, J., Rodriguez, P., Trajkovska, I., Mozo, A., & Salvachua, J. (2011). Testing a Cloud Provider Network for Hybrid P2P and Cloud Streaming Architectures. In

Cloud Computing (CLOUD), 2011 IEEE International Conference on (pp. 356–363). https://doi.org/10.1109/CLOUD.2011.52

17. Costa, C. M., Leite, C. R. M., & Sousa, A. L. (2015). Service Response Time Measurement Model of Service Level Agreements in Cloud Environment. In 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity) (pp. 969–974). https://doi.org/10.1109/SmartCity.2015.196

18. de Assunção, M. D., Cardonha, C. H., Netto, M. A. S., & Cunha, R. L. F. (2016). Impact of user patience on auto-scaling resource capacity for cloud services. Future Generation Computer Systems, 55, 41–50. http://dx.doi.org/10.1016/j.future.2015.09.001

19. Dou, W., Xu, X., Meng, S., & Yu, S. (2015). An Energy-Aware QoS Enhanced Method for Service Computing across Clouds and Data Centers. In 2015 Third International Conference on Advanced Cloud and Big Data (pp. 80–87). https://doi.org/10.1109/CBD.2015.23

20. Duggan, J., Cetintemel, U., Papaemmanouil, O., & Upfal, E. (2011). Performance Prediction for Concurrent Database Workloads. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data (pp. 337–348). New York, NY, USA: ACM. doi:10.1145/1989323.1989359

21. Ezenwoke, A., Daramola, O., & Adigun, M. (2018). QoS-based ranking and selection of SaaS applications using heterogeneous similarity metrics. Journal of Cloud Computing, 7(1), 15. https://doi.org/10.1186/s13677-018-0117-4

22. Faragardi, H. R., Shojaee, R., Tabani, H., & Rajabi, A. (2013). An analytical model to evaluate reliability of cloud computing systems in the presence of QoS requirements. In Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on (pp. 315–321). https://doi.org/10.1109/ICIS.2013.6607860

23. Garcia-Pineda, M., Segura-Garcia, J., & Felici-Castell, S. (2018). Estimation techniques to measure subjective quality on live video streaming in Cloud Mobile Media services. Computer Communications, 118, 27–39. https://doi.org/10.1016/j.comcom.2017.08.009

24. Garg, S. K., Versteeg, S., & Buyya, R. (2013). A framework for ranking of cloud computing services. Future Generation Computer Systems, 29(4), 1012–1023. http://dx.doi.org/10.1016/j.future.2012.06.006

25. Ghafari, S. M., Fazeli, M., Patooghy, A., & Rikhtechi, L. (2013). Bee-MMT: A load balancing method for power consumption management in cloud computing. In Contemporary Computing (IC3), 2013 Sixth International Conference on (pp. 76–80). https://doi.org/10.1109/IC3.2013.6612165

26. Ghahramani, M. H., Zhou, M., & Hon, C. T. (2017). Toward cloud computing QoS architecture: analysis of cloud systems and cloud services. IEEE/CAA Journal of Automatica Sinica, 4(1), 6–18. https://doi.org/10.1109/JAS.2017.7510313

27. Gholami, A., & Arani, M. G. (2015). A trust model for resource selection in cloud computing environment. In 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI) (pp. 144–151). https://doi.org/10.1109/KBEI.2015.7436036

28. Ghosh, R., Longo, F., Naik, V. K., & Trivedi, K. S. (2010). Quantifying Resiliency of IaaS Cloud. In Reliable Distributed Systems, 2010 29th IEEE Symposium on (pp. 343–347). https://doi.org/10.1109/SRDS.2010.49

29. Gonzales, D., Kaplan, J. M., Saltzman, E., Winkelman, Z., & Woods, D. (2017). Cloud-Trust—a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds. IEEE Transactions on Cloud Computing, 5(3), 523–536. https://doi.org/10.1109/TCC.2015.2415794

30. Guérout, T., Medjiah, S., Costa, G. Da, & Monteil, T. (2014). Quality of service modeling for green scheduling in Clouds. Sustainable Computing: Informatics and Systems, 4(4), 225–240. https://dx.doi.org/10.1016/j.suscom.2014.08.006

31. Hasan, M. S., Alvares, F., Ledoux, T., & Pazat, J. (2017). Investigating Energy Consumption and Performance Trade-Off for Interactive Cloud Application. IEEE Transactions on Sustainable Computing, 2(2), 113–126. https://doi.org/10.1109/TSUSC.2017.2714959

32. Hassam, M., Kara, N., Belqasmi, F., & Glitho, R. (2014). Virtualized Infrastructure for Video Game Applications in Cloud Environments. In Proceedings of the 12th ACM International Symposium on Mobility Management and Wireless Access (pp. 109–114). New York, NY, USA: ACM. doi:10.1145/2642668.2642679

33. Hecht, G., Jose-Scheidt, B., Figueiredo, C. D., Moha, N., & Khomh, F. (2014). An Empirical Study of the Impact of Cloud Patterns on Quality of Service (QoS). In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on (pp. 278–283). https://doi.org/10.1109/CloudCom.2014.141

34. Heidari, P., Boucheneb, H., & Shami, A. (2015). A Formal Approach for QoS Assurance in the Cloud. In 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom) (pp. 629–634). https://doi.org/10.1109/CloudCom.2015.36

35. Hu, Y., Deng, B., Yang, Y., & Wang, D. (2017). Elasticity evaluation of IaaS cloud based on mixed workloads. In Proceedings - 15th International Symposium on Parallel and Distributed Computing, ISPDC 2016 (pp. 157–164). Beijing Institute of System Engineering, Beijing, China. https://doi.org/10.1109/ISPDC.2016.28

36. Hwang, K., Bai, X., Shi, Y., Li, M., Chen, W.-G., & Wu, Y. (2016). Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies. IEEE Transactions on Parallel and Distributed Systems, 27(1), 130–143. https://doi.org/10.1109/TPDS.2015.2398438

37. Ibrahim, A. A. Z. A., Wasim, M. U., Varrette, S., & Bouvry, P. (2018). PRESEnCE: Performance Metrics Models for Cloud SaaS Web Services. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 936–940). https://doi.org/10.1109/CLOUD.2018.00140

38. Joy, N., Chandrasekaran, K., & Binu, A. (2015). A study on energy efficient cloud computing. In 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1–6). https://doi.org/10.1109/ICCIC.2015.7435661

39. Kaaniche, N., Mohamed, M., Laurent, M., & Ludwig, H. (2017). Security SLA Based Monitoring in Clouds. In 2017 IEEE International Conference on Edge Computing (EDGE) (pp. 90–97). https://doi.org/10.1109/IEEE.EDGE.2017.20

40. Karim, R., Ding, C., & Miri, A. (2015). End-to-End Performance Prediction for Selecting Cloud Services Solutions. In Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on (pp. 69–77). https://doi.org/10.1109/SOSE.2015.11

41. Katsaros, G., Subirats, J., Fitó, J. O., Guitart, J., Gilet, P., & Espling, D. (2013). A service framework for energy-aware monitoring and VM management in Clouds. Future Generation Computer Systems, 29(8), 2077–2091. https://dx.doi.org/10.1016/j.future.2012.12.006

42. Kaur, P. D., & Chana, I. (2014). A resource elasticity framework for QoS-aware execution of cloud applications. Future Generation Computer Systems, 37, 14–25. https://dx.doi.org/10.1016/j.future.2014.02.018

43. Kirsal, Y., Ever, Y. K., Mostarda, L., & Gemikonakli, O. (2015). Analytical Modelling and Performability Analysis for Cloud Computing Using Queuing System. In 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC) (pp. 643–647). https://doi.org/10.1109/UCC.2015.115

44. Lee, J. Y., Lee, J. W., Cheun, D. W., & Kim, S. D. (2009). A Quality Model for Evaluating Software-as-a-Service in Cloud Computing. In Software Engineering Research, Management and Applications, 2009. SERA '09. 7th ACIS International Conference on (pp. 261–266). https://doi.org/10.1109/SERA.2009.43

45. Lim, E., & Thiran, P. (2014). Communication of Technical QoS among Cloud Brokers. In Cloud Engineering (IC2E), 2014 IEEE International Conference on (pp. 403–409). https://doi.org/10.1109/IC2E.2014.92

46. Lin, Y.-K., & Chang, P.-C. (2011). Maintenance reliability estimation for a cloud computing network with nodes failure. Expert Systems with Applications, 38(11), 14185–14189. https://dx.doi.org/10.1016/j.eswa.2011.04.230

47. Liu, M., Dou, W., Yu, S., & Zhang, Z. (2014). A clusterized firewall framework for cloud computing. In 2014 IEEE International Conference on Communications (ICC) (pp. 3788–3793). https://doi.org/10.1109/ICC.2014.6883911

48. Liu, X., Xia, C., Wang, T., & Zhong, L. (2017). CloudSec: A Novel Approach to Verifying Security Conformance at the Bottom of the Cloud. In 2017 IEEE International Congress on Big Data (BigData Congress) (pp. 569–576). https://doi.org/10.1109/BigDataCongress.2017.87

49. Lu, L., & Yuan, Y. (2018). A novel TOPSIS evaluation scheme for cloud service trustworthiness combining objective and subjective aspects. Journal of Systems and Software, 143, 71–86. https://doi.org/10.1016/j.jss.2018.05.004

50. Manuel, P. (2015). A trust model of cloud computing based on Quality of Service. Annals of Operations Research, 233(1), 281–292. https://doi.org/10.1007/s10479-013-1380-x

51. Mastelic, T., Brandic, I., & Jaarevic, J. (2014). CPU Performance Coefficient (CPU-PC): A Novel Performance Metric Based on Real-Time CPU Resource Provisioning in Time-Shared Cloud Environments. In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on (pp. 408–415). https://doi.org/10.1109/CloudCom.2014.13

52. Mesbahi, M. R., Rahmani, A. M., & Hosseinzadeh, M. (2018). Reliability and high availability in cloud computing environments: a reference roadmap. Human-Centric Computing and Information Sciences, 8(1), 20. https://doi.org/10.1186/s13673-018-0143-8

53. Nadanam, P., & Rajmohan, R. (2012). QoS evaluation for web services in cloud computing. In Computing Communication Networking Technologies (ICCCNT), 2012 Third International Conference on (pp. 1–8). https://doi.org/10.1109/ICCCNT.2012.6395991

54. Pedersen, J. M., Riaz, M. T., Junior, J. C., Dubalski, B., Ledzinski, D., & Patel, A. (2011). Assessing Measurements of QoS for Global Cloud Computing Services. In Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on (pp. 682–689). https://doi.org/10.1109/DASC.2011.120

55. Preuveneers, D., Heyman, T., Berbers, Y., & Joosen, W. (2016). Systematic scalability assessment for feature oriented multi-tenant services. Journal of Systems and Software, 116, 162–176. https://doi.org/10.1016/j.jss.2015.12.024

56. Qian, S., Cao, J., Le Mouël, F., Li, M., & Wang, J. (2015). Towards Prioritized Event Matching in a Content-based Publish/Subscribe System. In Proceedings of the 9th

ACM International Conference on Distributed Event-Based Systems (pp. 116–127). New York, NY, USA: ACM. doi:10.1145/2675743.2771823

57. Ran, Y., Shi, Y., Yang, E., Chen, S., & Yang, J. (2014). Dynamic resource allocation for video transcoding with QoS guaranteeing in cloud-based DASH system. In 2014 IEEE Globecom Workshops (GC Wkshps) (pp. 144–149). https://doi.org/10.1109/GLOCOMW.2014.7063421

58. Ravindhren, V. G., & Ravimaran, S. (2017). CCMA—cloud critical metric assessment framework for scientific computing. Cluster Computing. https://doi.org/10.1007/s10586-017-1384-4

59. Ravindran, K. (2013). Self-Assessment and Reconfiguration Methods for Autonomous Cloud-based Network Systems. In Distributed Simulation and Real Time Applications (DS-RT), 2013 IEEE/ACM 17th International Symposium on (pp. 87–94). https://doi.org/10.1109/DS-RT.2013.37

60. Rizvi, S., Ryoo, J., Kissell, J., & Aiken, B. (2015). A Stakeholder-oriented Assessment Index for Cloud Security Auditing. In Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication (pp. 55:1–55:7). New York, NY, USA: ACM. doi:10.1145/2701126.2701226

61. Rizvi, S., Roddy, H., Gualdoni, J., & Myzyri, I. (2017). Three-Step Approach to QoS Maintenance in Cloud Computing Using a Third-Party Auditor. Procedia Computer Science, 114, 83–92. https://doi.org/10.1016/j.procs.2017.09.014

62. Roohitavaf, M., Entezari-Maleki, R., & Movaghar, A. (2013). Availability Modeling and Evaluation of Cloud Virtual Data Centers. In Parallel and Distributed Systems (ICPADS), 2013 International Conference on (pp. 675–680). https://doi.org/10.1109/ICPADS.2013.120

63. Saiz, E., Ibarrola, E., Cristobo, L., & Taboada, I. (2014). A cloud platform for QoE evaluation: QoXcloud. In ITU Kaleidoscope Academic Conference: Living in a converged world - Impossible without standards?, Proceedings of the 2014 (pp. 241–247). https://doi.org/10.1109/Kaleidoscope.2014.6858471

64. Samet, N., Letaïfa, A. Ben, Hamdi, M., & Tabbane, S. (2016). Real-Time User Experience Evaluation for Cloud-Based Mobile Video. In 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA) (pp. 204–208). https://doi.org/10.1109/WAINA.2016.120

65. Singh, S., & Chana, I. (2015). Q-aware: Quality of service based cloud resource provisioning. Computers & Electrical Engineering, 47, 138–160. https://dx.doi.org/10.1016/j.compeleceng.2015.02.003

66. Slivar, I., Skorin-Kapov, L., & Suznjevic, M. (2016). Cloud Gaming QoE Models for Deriving Video Encoding Adaptation Strategies. In Proceedings of the 7th International Conference on Multimedia Systems (pp. 18:1–18:12). New York, NY, USA: ACM. doi:10.1145/2910017.2910602

67. Son, S., & Sim, K. M. (2015). Adaptive and similarity-based tradeoff algorithms in a price-timeslot-QoS negotiation system to establish cloud SLAs. Information Systems Frontiers, 17(3), 565–589. https://doi.org/10.1007/s10796-013-9432-y

68. Sousa, F. R. C., & Machado, J. C. (2012). Towards Elastic Multi-Tenant Database Replication with Quality of Service. In Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on (pp. 168–175). https://doi.org/10.1109/UCC.2012.36

69. Souza, R. H. de, Flores, P. A., Dantas, M. A. R., & Siqueira, F. (2016). Architectural recovering model for Distributed Databases: A reliability, availability and serviceability approach. In 2016 IEEE Symposium on Computers and Communication (ISCC) (pp. 575–580). https://doi.org/10.1109/ISCC.2016.7543799

70. Taherizadeh, S., & Stankovski, V. (2017). Incremental Learning from Multi-level Monitoring Data and Its Application to Component Based Software Engineering. In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 378–383). https://doi.org/10.1109/COMPSAC.2017.148

71. Vedam, V., & Vemulapati, J. (2012). Demystifying Cloud Benchmarking Paradigm - An in Depth View. In 2012 IEEE 36th Annual Computer Software and Applications Conference (pp. 416–421). https://doi.org/10.1109/COMPSAC.2012.61

72. Wagle, S. S., Guzek, M., Bouvry, P., & Bisdorff, R. (2015). An Evaluation Model for Selecting Cloud Services from Commercially Available Cloud Providers. In 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom) (pp. 107–114). https://doi.org/10.1109/CloudCom.2015.94

73. Wang, S., & Dey, S. (2012). Cloud Mobile Gaming: Modeling and Measuring User Experience in Mobile Wireless Networks. SIGMOBILE Mob. Comput. Commun. Rev., 16(1), 10–21. doi:10.1145/2331675.2331679

74. Wen, Z. Y., & Hsiao, H. F. (2014). QoE-driven performance analysis of cloud gaming services. In Multimedia Signal Processing (MMSP), 2014 IEEE 16th International Workshop on (pp. 1–6). https://doi.org/10.1109/MMSP.2014.6958835

75. Wu, X., Liu, G., & Xu, J. (2015). A QoS-constrained scheduling for access requests in cloud storage. In Industrial Electronics and Applications (ICIEA), 2015 IEEE 10th Conference on (pp. 155–160). https://doi.org/10.1109/ICIEA.2015.7334102

76. Xia, Y., Zhou, M., Luo, X., Zhu, Q., Li, J., & Huang, Y. (2015). Stochastic Modeling and Quality Evaluation of Infrastructure-as-a-Service Clouds. IEEE Transactions on

Automation Science and Engineering, 12(1), 162–170. https://doi.org/10.1109/TASE.2013.2276477

77. Xiao, Y., Lin, C., Jiang, Y., Chu, X., & Shen, X. (2010). Reputation-Based QoS Provisioning in Cloud Computing via Dirichlet Multinomial Model. In Communications (ICC), 2010 IEEE International Conference on (pp. 1–5). https://doi.org/10.1109/ICC.2010.5502407

78. Xiong, K., & Chen, X. (2015). Ensuring Cloud Service Guarantees via Service Level Agreement (SLA)-Based Resource Allocation. In 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops (pp. 35–41). https://doi.org/10.1109/ICDCSW.2015.18

79. Xu, H., Qiu, X., Sheng, Y., Luo, L., & Xiang, Y. (2018). A Qos-Driven Approach to the Cloud Service Addressing Attributes of Security. IEEE Access, 6, 34477–34487. https://doi.org/10.1109/ACCESS.2018.2849594

80. Yu, N., Gu, F., Guo, X., & He, Z. (2015). A Fine-grained Flow Control Model for Cloud-assisted Data Broadcasting. In Proceedings of the 18th Symposium on Communications & Networking (pp. 24–31). San Diego, CA, USA: Society for Computer Simulation International. Retrieved from https://dl.acm.org/citation.cfm?id=2872550.2872554

81. Zant, B. El, & Gagnaire, M. (2015). Towards a unified customer aware figure of merit for CSP selection. Journal of Cloud Computing, 4(1), 1–23. https://doi.org/10.1186/s13677-015-0049-1

82. Zheng, X., Martin, P., & Brohman, K. (2013). Cloud Service Negotiation: A Research Roadmap. In Services Computing (SCC), 2013 IEEE International Conference on (pp. 627–634). https://doi.org/10.1109/SCC.2013.93

83. Zheng, X., Martin, P., Brohman, K., & Xu, L. D. (2014). CLOUDQUAL: A Quality Model for Cloud Services. IEEE Transactions on Industrial Informatics, 10(2), 1527–1536. https://doi.org/10.1109/TII.2014.2306329

84. Zhou, P., Wang, Z., Li, W., & Jiang, N. (2015). Quality Model of Cloud Service. In High Performance Computing and Communications (HPCC), 2015