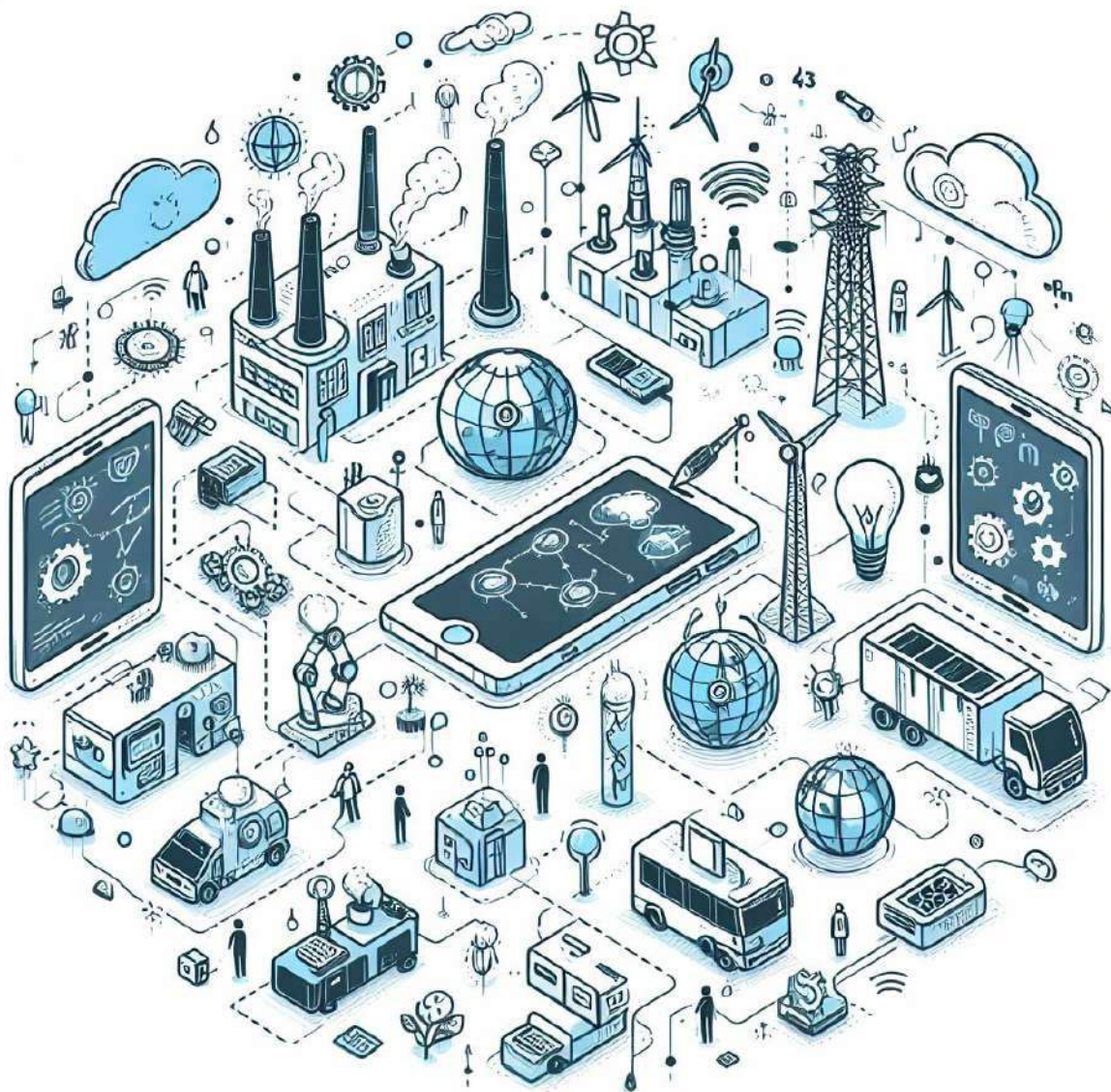




Internet de las Cosas

Juan Guillermo Rivera Berrío
Ramiro Antonio Lopera Sánchez

iCartesiLibri

Internet de las cosas

Juan Guillermo Rivera Berrío



INSTITUCIÓN UNIVERSITARIA
**COLEGIO MAYOR
DE ANTIOQUIA**

Ramiro Antonio Lopera Sánchez
Institución Universitaria Pascual Bravo

Fondo Editorial RED Descartes



Córdoba (España)

2024

Título de la obra:
Internet de las cosas

Autores:
Juan Guillermo Rivera Berrío
Ramiro Antonio Lopera Sánchez

Código JavaScript para el libro: [Joel Espinosa Longi](#), [IMATE](#), UNAM.
Recursos interactivos: [DescartesJS](#)
Fuentes: [Lato](#) y [UbuntuMono](#)
Imagen portada: Diseño del generador de imágenes [Designer](#) de Microsoft

Red Educativa Digital Descartes
Córdoba (España)
descartes@proyectodescartes.org
<https://proyectodescartes.org>

Proyecto iCartesiLibri
<https://proyectodescartes.org/iCartesiLibri/index.htm>

ISBN: 978-84-18834-92-9



Esta obra está bajo una licencia Creative Commons 4.0 internacional: Reconocimiento-No Comercial-Compartir Igual.

Tabla de contenido

Prefacio	7
Contenido multimedia	9
1. Conociendo el IoT	11
1.1 Internet de las cosas	12
1.2 Historia del IoT	14
1.3 IoT en el hogar	17
1.4 Elementos del IoT	19
1.5 Ventajas y desventajas del IoT	21
1.6 Internet del todo	23
2. Sistemas embebidos	29
2.1 Industria 4.0	30
2.2 Sistemas embebidos	36
2.2.1 Entre bits y bytes	39
2.2.2 Los sistemas embebidos	44
2.2.3 Componentes de los sistemas embebidos	46
2.2.4 El Arduino	48
2.2.5 El Raspberry Pi	49
2.3 El Arduino y su impacto en IoT	51
2.3.1 Un poco de historia del Arduino	54
2.3.2 Ventajas del Arduino	56
2.4 El Raspberry Pi y su impacto en IoT	58
2.4.1 Historia del Raspberry Pi	60
2.4.2 Software que se puede usar en una Raspberry Pi	60

3. Programando el IoT	65
3.1 Programación en IoT - Sensores y datos	66
3.1.1 Sensores y datos en un entorno de ciudad	68
3.1.2 Algoritmos	69
3.1.3 Teorema del programa estructurado	72
3.1.4 Estructura secuencial	72
3.1.5 Estructuras selectivas	75
3.1.6 Estructuras iterativas	77
3.2 Computación en la Nube	80
3.2.1 Modelos de computación en la nube	83
3.2.2 Tipos de Nube	84
3.2.3 Ventajas y desventajas de la Nube	86
3.3 Métricas y analítica en IoT	88
3.3.1 Historia de los científicos de datos	90
3.3.2 Herramientas para analítica de datos	92
3.4 Un proyecto inteligente	94
3.4.1 Idea del proyecto inteligente	96
3.4.2 Etapas del proyecto	96
3.4.3 Etapa UNO del proyecto inteligente	97
3.4.4 Sensaciones para la planta inteligente	100
3.4.5 Construyendo el algoritmo	103
4. Aplicaciones IoT	109
4.1 Hablando con la planta inteligente	110
4.1.1 Lenguaje C	110
4.1.2 Estructuras de programación de Arduino	115

4.1.3 Entorno de simulación Wokwi	116
4.1.3.1 Primer proyecto con Wokwi - Luz parpadeante	117
4.1.3.2 Segundo proyecto con Wokwi - uso del protoboard	120
4.1.3.3 Tercer proyecto con Wokwi - uso de sensores	124
4.1.3.4 Cuarto proyecto con Wokwi - sensor y semáforo	127
4.1.4 Proyectos Arduino en Físico	129
4.1.4.1 Instalación de Arduino IDE	130
4.1.4.2 Primer proyecto físico - uso de sensores	132
4.1.4.3 Segundo proyecto físico - planta inteligente	132
4.1.4.4 Tercer proyecto físico - planta inteligente	134

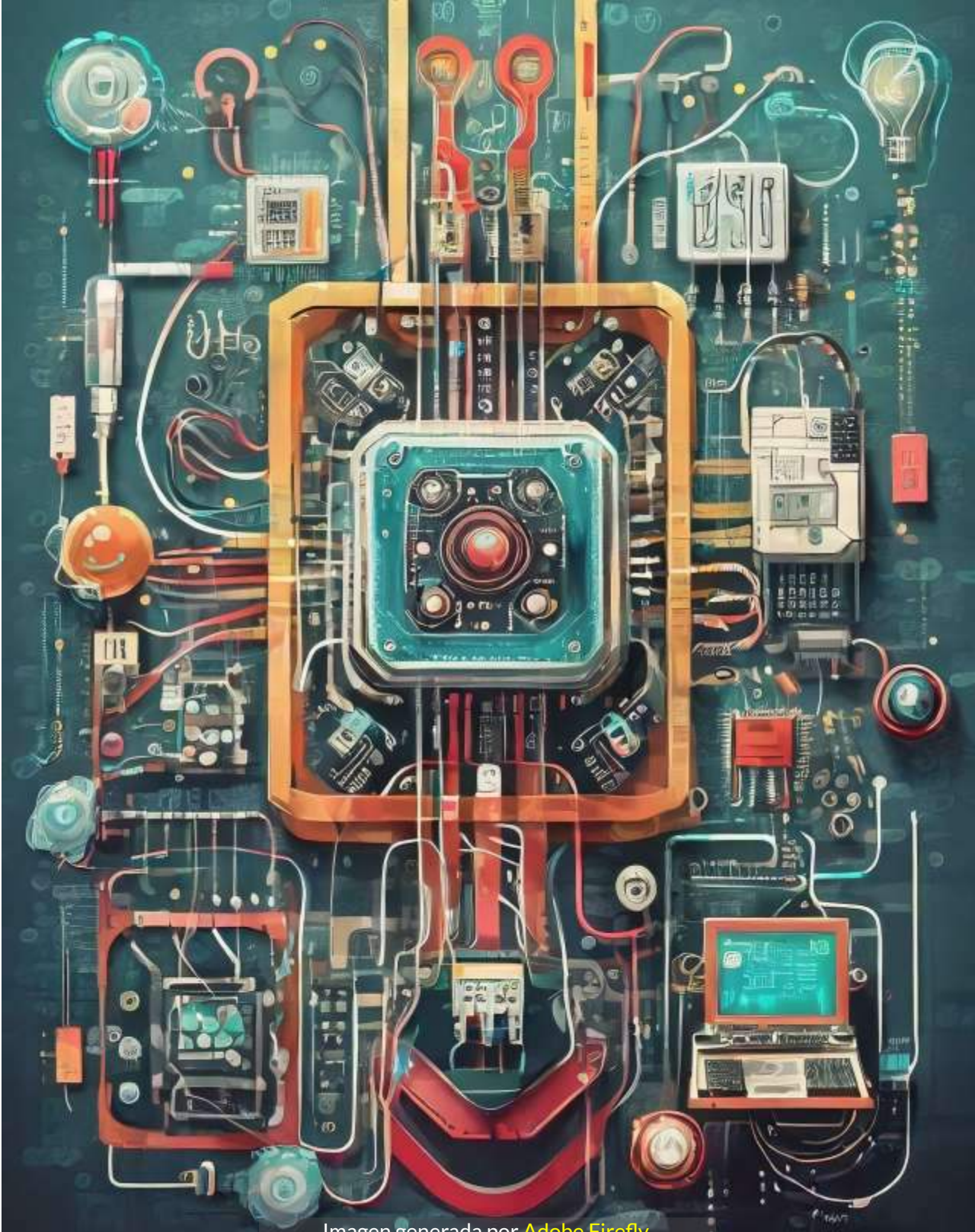


Imagen generada por [Adobe Firefly](#).

Prefacio

El Internet de las Cosas (IoT) está transformando nuestro mundo a un ritmo acelerado. Dispositivos cotidianos, desde refrigeradores hasta automóviles, se están integrando a la red, creando una red interconectada de objetos inteligentes que pueden recopilar, analizar e intercambiar datos. Esta revolución tecnológica está abriendo un sinfín de posibilidades para mejorar nuestras vidas en diversos aspectos, desde la automatización del hogar hasta la gestión de ciudades inteligentes.

En este libro, exploraremos a profundidad el mundo del IoT. Abordaremos los siguientes temas:



Conceptos básicos del IoT: Definición, historia, evolución y tendencias actuales.



Componentes del IoT: Hardware, software, redes y protocolos de comunicación.



Aplicaciones del IoT: Casos de uso en diferentes sectores como la industria, la salud, la agricultura, las ciudades inteligentes y el hogar conectado.



Impacto del IoT: Beneficios, desafíos y riesgos asociados a la implementación del IoT.



El futuro del IoT: Perspectivas sobre la evolución de la tecnología y su impacto en la sociedad.

A lo largo de este libro, nuestro objetivo es proporcionar una comprensión completa del IoT a lectores de diversos niveles de conocimiento. El contenido estará acompañado de ejemplos prácticos, ilustraciones y recursos adicionales para facilitar el aprendizaje.

Este libro está dirigido a estudiantes, investigadores y profesionales que estén interesados en aprender más sobre el IoT. El libro incluye ejemplos y casos de estudio para ilustrar los conceptos y técnicas presentados.

Este libro es una guía para principiantes y para aquellos interesados en aprender más sobre el IoT y sus aplicaciones.

La mayoría de las imágenes y videos están diseñados con IAs generativas, cuyos enlaces se suministran, de tal forma que el lector pueda practicar con estas IAs, incorporando nuevas instrucciones (**prompts**) o reutilizando (**remix**) el prompt usado.

Contenido multimedia

Para ilustrar mejor la información suministrada en este libro, hemos incluido, entre otros elementos multimedia, los siguientes:



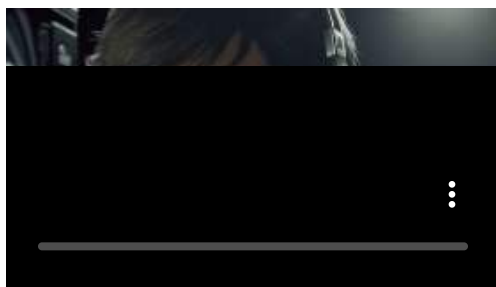
Texto. Además de artículos científicos y de divulgación, publicados en la web, los textos de este libro se soportan en información encontrada en sitios como [Futurepedia](#), [Learn prompting](#), [Al para océanos](#), [GPT for educators](#), [Aprende Machine Learning](#), [Blog NVIDIA](#), [OpenAI blog](#), [Medium Daily Digest](#), [Google DeepMind](#), [OBS Business School](#), [Towards AI](#), [Xavier Mitjana](#), [Salvaje](#) y, obviamente, las IAs generativas de texto, como: ChatGPT, Microsoft Copilot, YOU.com, Perplexity, Gemini, Rytr, Llama 2, Aria, Claude 2.1, entre muchas otras.



Imágenes. Los iconos de las listas, se han obtenido de [Freepik - Flaticon](#). Por otra parte, la mayoría de imágenes se han obtenido de [Pixabay](#), [Pexels](#) y las IAs generativas [lexica.art](#), Leonardo, Ideogram, Stable Diffusion, DALL·E, entre otras.



Videos. Algunos videos son tomados de YouTube con licencia creative commons, otros son solo ilustrativos, como los obtenidos en [Pixabay](#), [Pexels](#) o IAs como Genmo, Fliki o Neiro.



Objetos interactivos. Todos los objetos interactivos, fueron diseñados con el editor DescartesJS, los cuales incluyen presentadores interactivos, puzzles o actividades de evaluación.



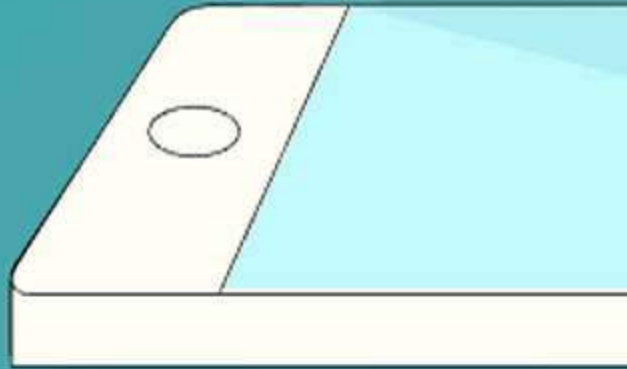
Diseño del generador de imágenes [Designer](#) de Microsoft

Capítulo 1

Conociendo el IoT

1.1 Internet de las cosas

Hoy en día, es una rutina diaria el enviar órdenes a algún dispositivo usando algún tipo de sensor; por ejemplo, usamos un sensor que llamamos "control remoto" para darle órdenes a nuestro televisor, o para subir o bajar la temperatura del aire acondicionado, o para abrir la puerta de un garaje. Este control sobre dispositivos se extiende a otros electrodomésticos como lavadoras, aspiradoras robóticas, hornos y refrigeradores.



Ahora, si en nuestro hogar contamos con servicio de internet y un Wi-Fi, algunos de estos dispositivos o "cosas", los podemos controlar con nuestro celular, práctica que se conoce como "**Internet de las cosas (IdC)**" o, en inglés, "**Internet of Things (IoT)**".



La Internet de los Dispositivos
o Internet de las Cosas,
comúnmente aceptado como
IoT, es la conexión de
dispositivos a Internet.

1.2 Historia del IoT

En la página Web de [Telefónica Tech](#), observarás temas como Ciberseguridad, Cloud, Blockchain y... *AI of Things*, ¿te sorprende el cuarto apartado? A nosotros también, ¿Inteligencia Artificial de las Cosas?, la respuesta la encontramos indagando cómo ha sido la evolución de IoT. por ello, es necesario conocer cómo ha sido la historia del IoT. En esta misma página, puedes consultar el artículo "[Breve historia de la Internet de las cosas](#)", que inicia con este dato:

A finales del siglo XIX se desarrollaron los primeros experimentos de telemetría de la historia. El más antiguo del que se tiene constancia tuvo lugar en 1874, cuando un equipo de investigación francés instaló una serie de dispositivos de información meteorológica en el Mont Blanc, datos eran transmitidos a París a través de un enlace de radio (Ibid).

Parte de esta historia, la narramos a continuación:



Interactivo 1.1. Historia del IoT.

Y en este otro objeto interactivo, puedes identificar los hitos más importantes en la historia del IoT:



Interactivo 1.2. Infografía de la historia del IoT.

Pero, ¿Qué es realmente la Inteligencia Artificial de las Cosas? En principio, es el mismo concepto de IoT, pero aquí el ser humano es reemplazado por una máquina (hardware) que con una combinación de algoritmos (software) se conectan con las cosas (dispositivos). Más

adelante, veremos algunas aplicaciones que usan Inteligencia Artificial con estos propósitos.

Retomando la historia, no podemos dejar a un lado a quien acuñó el término *Internet of Things (IoT)*, se trata del tecnólogo británico Kevin Ashton. A principios de la década del 90 trabajaba para la empresa Procter and Gamble (P&G) y estaba tratando de convencer a sus jefes de implementar las etiquetas de identificación de radiofrecuencia (RFID) y otros sensores en los productos de la cadena de suministro de la empresa [1]. Mientras pensaba en qué nombre darle a una presentación en PowerPoint para P&G, se le ocurrió:



Sabía que quería incluir la palabra 'internet', porque entonces podría obtener algo de aceptación. Todos estos viejos tipos de CEO de tipo blanco estaban muy entusiasmados con internet, pero en ese momento todavía era solo la revolución punto .com... Estaba hablando de que la cadena de suministro es una 'Red de cosas' y que Internet es una 'Red de bits', y de cómo la tecnología de sensores fusionaría los dos. Entonces pensé en un "Internet de las Cosas" y pensé: "Eso servirá, o tal vez incluso mejor". Tenía un timbre. Se convirtió en el título de la presentación (Ibid).

Según Ashton, "si todos los objetos de la vida cotidiana estuvieran equipados con identificadores y conectividad inalámbrica, estos podrían comunicarse entre sí y ser gestionados por las computadoras [2]

1.3 IoT en el hogar

Iniciamos este apartado, mostrando cómo algunas cosas de nuestro hogar las podíamos controlar con sensores u órdenes desde Internet. En la siguiente infografía, puedes observar otros dispositivos (cosas) que también podemos conectar.



Interactivo 1.3. Infografía del IoT en el hogar.

1.4 Elementos del IoT

Este libro lo hemos denominado de tipo "interactivo", pues permite la interacción sujeto - objeto de conocimiento, pero en el mundo del IoT, como lo dijimos al inicio, es posible la interacción objeto (Internet) - objeto (cosa), dispositivo - dispositivo, máquina - máquina.

Han sido dos tendencias las que nos han permitido llegar a este punto (la interconexión): primero, el creciente uso del teléfono inteligente para hacer pagos y transferencias por la red, así como el uso de las tabletas; segundo, el diseño y creación de dispositivos inteligentes como las etiquetas RFID (Identificación por Radiofrecuencia) y los sensores electrónicos, más potentes, rápidos y pequeños [\[3\]](#).

Los elementos que constituyen el IoT se suele llamar "Arquitectura IoT", la cual se puede dividir en siete, cuatro o tres capas. El modelo simplificado de tres capas es el siguiente (ver [Figura 1.2](#)):

- Capa de **Percepción** o detección de objetos (**Sensores y Actuadores**). Incorpora sensores para detectar y recopilar información sobre el entorno como la temperatura, la humedad, la presencia de humo, la presencia de seres vivos (cosas) e, incluso, la presencia de otros objetos inteligentes. Por su parte, los actuadores realizan correcciones en tiempo real, como ajustar el nivel del aire acondicionado.
- Capa de **Red**. Es la responsable de recoger los datos suministrados por los sensores y convertirlos en un formato digital, permitiendo el procesamiento e intercambio de datos a través de una pasarela de Internet bien sea por una WAN¹ inalámbrica (Wi-Fi o celular) o una WAN cableada.
- Capa de **Aplicación**. Servicios de IoT para hogares inteligentes, ciudades inteligentes o fábricas inteligentes.

¹ WAN (Wide Area Network): Red de Área Amplia

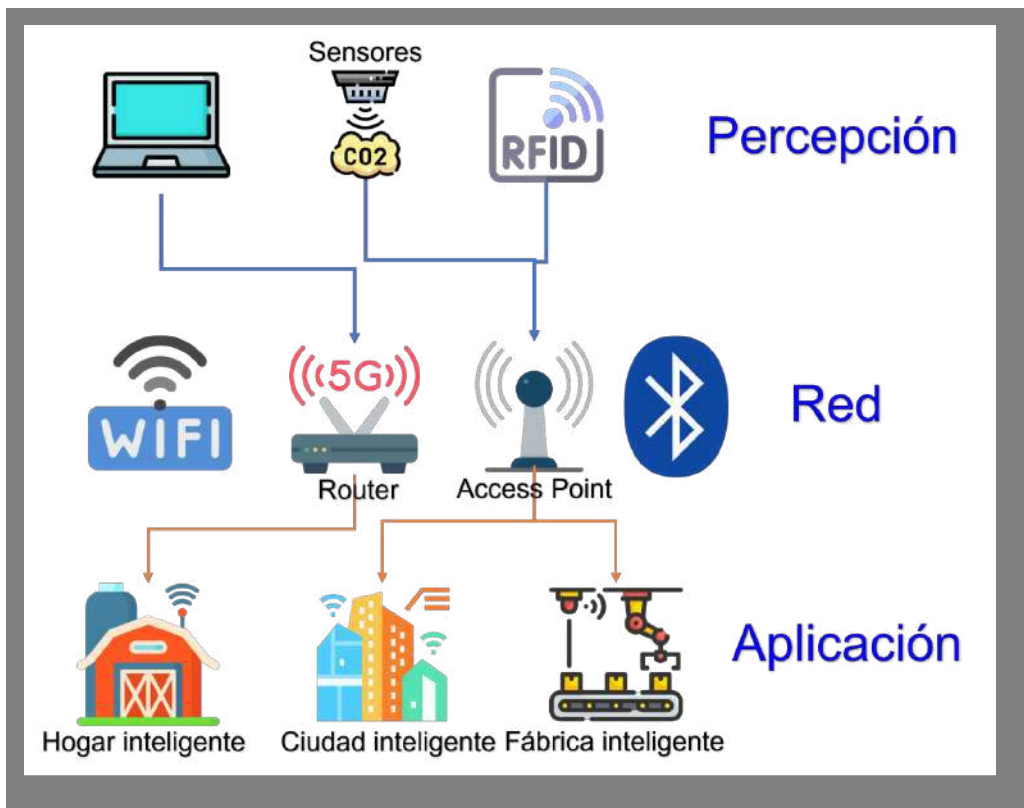


Figura 1.2. Arquitectura IoT, modelo de tres capas (imagen adaptada de Tasneem et al. [4]).

"La capa de aplicación garantiza la autenticidad, integridad y confidencialidad de los datos. En esta capa se logra el propósito de IoT que es la creación de entornos inteligentes" [4].

En el objeto interactivo de la siguiente página, presentamos una infografía de los elementos IoT. Incluimos los sistemas embebidos, pues son sistemas diseñados para cubrir necesidades muy específicas en tiempo real. Los sistemas embebidos son productos que empleamos a diario, por ejemplo en los hornos microondas, tostadoras y lavadoras.



Interactivo 1.4. Infografía de los elementos IoT.

1.5 Ventajas y desventajas del IoT

Como lo hemos visto, son muchas las ventajas que ofrece el IoT; entre ellas, la reducción de costos en la adquisición de datos, mejora en la calidad de vida de las personas, automatización de los procesos de fabricación industrial, identificación y monitoreo de estados de salud,

la detección de amenazas de contaminación, incendios y desastres naturales, seguridad, entretenimiento. En la [infografía del IoT en el hogar](#) mostramos algunas ventajas adicionales.

Pero toda nueva tecnología trae consigo nuevos riesgos. Algunas desventajas del IoT, las presenta Pineda [\[3\]](#):

Desventajas IoT

- *La pérdida de la privacidad y seguridad personal, la conexión de todos los objetos genera una cantidad de información privada y personal propensa a ciberataques por hackers y piratas informáticos. Somos más vulnerables, incluso ante gobiernos y empresas que pueden conocer nuestra vida íntima cotidiana mediante cookies. Y aunque la seguridad física de personas y bienes puede ser mayor, la privacidad de gustos, aficiones, tendencias, anhelos, datos médicos, sensaciones, se vuelve un control en manos ajenas, la nube.*
- *La incompatibilidad, sobre todo si los dispositivos son de distintos fabricantes.*
- *La complejidad de la red misma, que es diversa y cualquier fallo de software o hardware, o un corte eléctrico tendría graves consecuencias.*
- *La pérdida de empleo para los trabajadores menos cualificados ante la automatización. Lo cual requiere mayor educación.*
- *El que la tecnología tome el control de nuestras vidas y nos convierta en dependientes totales de ella, como ya se observa en los jóvenes con el celular.*

Finalmente, te invitamos a realizar la siguiente actividad evaluativa.

En el siguiente objeto interactivo, se presentan dos ejercicios. La actividad evaluativa consiste en señalar las imágenes que cumplen con lo enunciado en el título.



Interactivo 1.5. Evaluación sumativa - Selecccion múltiple con imágenes.

Para una mejor interacción, realiza la actividad en una ventana ampliada, haciendo clic en el botón de la esquina superior derecha.

1.6 Internet del todo

No hace más de una década, en el año 2012, que la empresa **CISCO** hizo una definición del término loE (*internet of everything*) o en español el Internet de Todo o IdT, haciendo alusión a una red de redes en la que se encuentran reunidos elementos como los datos, las

personas, los procesos y, desde luego, las cosas. Todo estos interconectados de una manera valiosa.

Mientras que el IoT permite acceder a dispositivos físicos, usando como canal el internet y haciendo una red global enrutada a las cosas, el IoE se enfoca en conexiones de redes inteligentes y, desde luego, en las tecnologías.

No se trata sólo de "cosas", sino también de cómo se autentican y se comunican (IoE).

El IdT se basa en cuatro pilares: personas, procesos, datos y cosas ([OpenMind](#)). Cada uno de estos pilares tiene un rol importante en el funcionamiento y el valor del IdT.



Personas: se trata de conectar a las personas de formas más relevantes y valiosas, ya sea entre ellas o con las máquinas, los datos y los procesos. Por ejemplo, las redes sociales, las aplicaciones móviles, los dispositivos vestibles y la realidad aumentada son formas de conectar a las personas con el IdT.



Procesos: se refiere a entregar la información adecuada a la persona o máquina adecuada en el momento adecuado, para facilitar la toma de decisiones y la acción. Por ejemplo, los





Imagen generada por la IA [Adobe Firefly](#).

sistemas de gestión, los flujos de trabajo, las alertas y las notificaciones son formas de optimizar los procesos con el IdT.



Datos: se trata de convertir los datos en inteligencia para tomar mejores decisiones. Los datos son el resultado de la captura, el almacenamiento, el análisis y la visualización de la información generada por las personas, las máquinas y las cosas. Por ejemplo, los sensores, los big data, la inteligencia artificial y el aprendizaje automático son formas de aprovechar los datos con el IdT.



Cosas: se refiere a los dispositivos físicos y objetos conectados a Internet y entre sí para la toma de decisiones inteligente. Las cosas pueden tener sensores, actuadores, software y comunicación integrados para interactuar con su estado interno o el entorno externo. Por ejemplo, los electrodomésticos, los vehículos, las máquinas industriales y los edificios inteligentes son formas de integrar las cosas con el IdT.

La conexiones requeridas para que toda esta comunicación, decisiones e interacción sea posible, son:



P2P: personas que se comunican con personas.



M2P - P2M: máquinas que se comunican con personas y viceversa.



M2M: máquinas que se comunican con máquinas.

En el IdT es importante la interacción de las personas en el ecosistema, aportando a través de esta interacción datos importantes y relevantes para que las máquinas y la información generen servicios de calidad y oportunos.

¿Quién quiere ser millonario?

50 x 15



[Documentos de registro](#)

Características del juego

Nº Jugadores	1
Nº Preguntas	≥ 15
Tipo respuesta	4 (A-D)
Tipo de fichero de preguntas	1
Etapas recomendada	5



Jesús M. Muñoz Calle, adaptado por
Juan Gmo. Rivera B.



Jugar

Interactivo 1.6. Juego ¿Quién quiere ser millonario?.

2.1 Industria 4.0

El término Industria 4.0 es adoptado en Alemania en la Feria de Hannover de 2011, entendido como:

la completa digitalización de las cadenas de valor a través de la integración de tecnologías de procesamiento de datos, software inteligente y sensores; desde los proveedores hasta los clientes para poder predecir, controlar, planear y producir de forma inteligente, lo que genera mayor valor a toda la cadena" ([Disruptive Angels](#)).

En la que se destacan cuatro componentes principales: los sistemas Ciber-Físicos (CPS), El Internet de las cosas (IOT), El Internet de Servicios (IOS), las fábricas inteligentes y la manufactura aditiva.

La 4RI comprende un cambio de paradigma de la fabricación automatizada, hacia un concepto de fabricación inteligente. Los mundos físico y virtual crecen juntos, la implementación de la fabricación inteligente hará uso de conceptos como IoT para facilitar este cambio (Thoben et al. [\[5\]](#)).

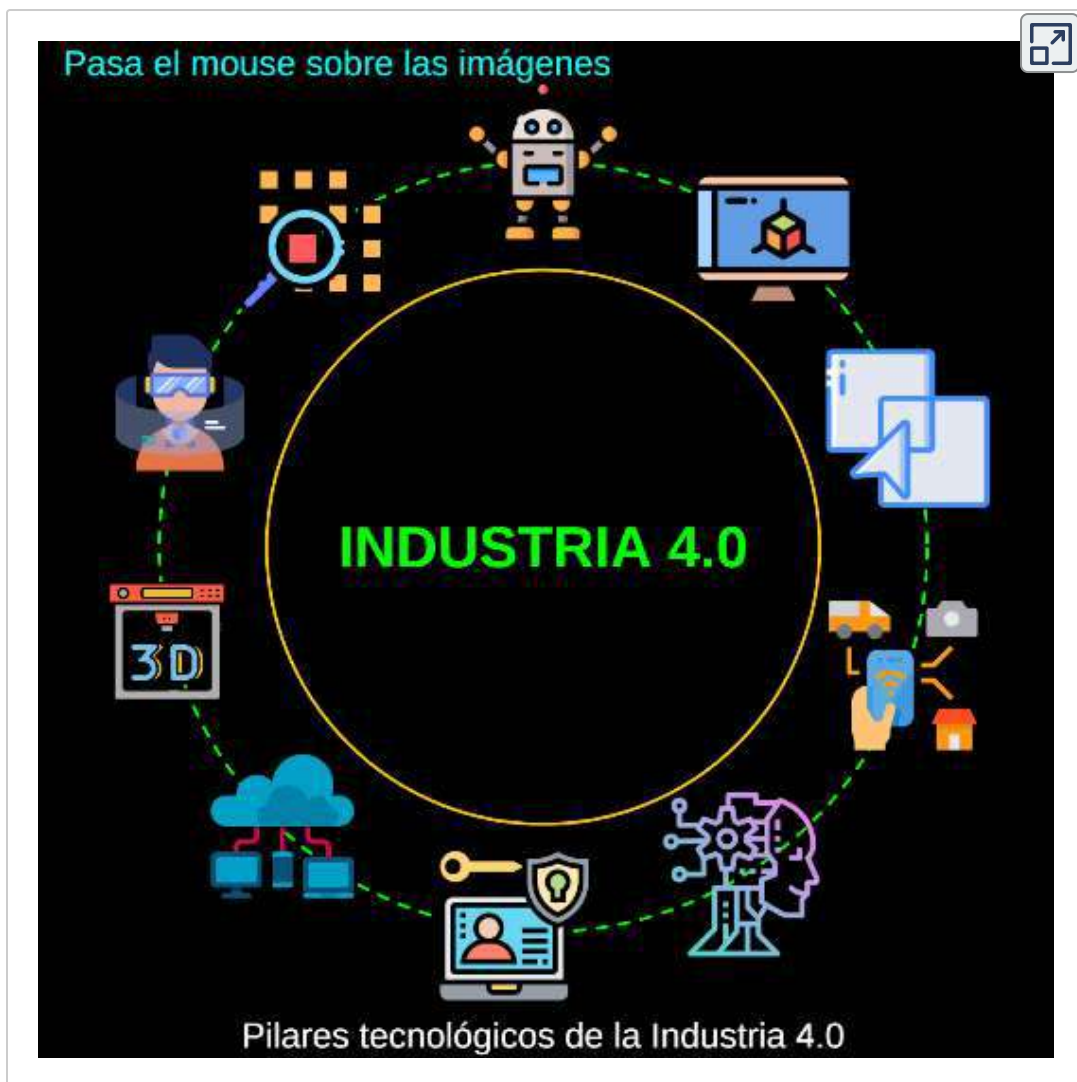
No obstante esta delimitación, existen desarrollos tecnológicos que soportan los sistemas anteriores, entre ellos Basco et al. [\[6\]](#) identifican 10 "pilares tecnológicos de la Industria 4.0", los cuales se presentan en el interactivo 2.1, tales como la robótica, la computación en la nube, el Big Data, la fabricación aditiva, la inteligencia artificial, los sistemas de integración y la ciberseguridad.

La Cuarta Revolución Industrial, entonces, es una etapa en la que la digitalización penetra disruptivamente el sector industrial con tecnologías como las que se muestran en el siguiente interactivo.





Imagen generada por la IA [NightCafé](#).



Interactivo 2.1. Pilares tecnológicos de la Industria 4.0.
Créditos textos: Basco et al. [6], créditos imágenes: [Freepik](#).

Por otra parte, Klaus Schwab, fundador del Foro Económico Mundial, acuña el término "Cuarta Revolución Industrial" en 2016², destacando los avances tecnológicos como la inteligencia artificial, la robótica, el internet de las cosas, los vehículos autónomos, la

² Si quieres conocer sobre las tres primeras revoluciones industriales, consulta el primer capítulo del libro "[Cuarta revolución industrial](#)".

impresión 3D, la nanotecnología, la biotecnología, la ciencia de los materiales y la computación cuántica. Schwab presenta tres razones para justificar la puesta en marcha de la 4RI:



Velocidad. Evolución exponencial como resultado de un mundo interconectado.



Amplitud y profundidad. Basada en la revolución digital que está generando cambios sin precedentes en la economía, los negocios, la sociedad y las personas. "No solo está cambiando el 'qué' y el 'cómo' hacer las cosas, sino el 'quienes somos'" ([7], p.15).



Impacto en los sistemas. Transformación de sistemas complejos entre países, empresas, industrias y la sociedad en general.

De los muchos desafíos diversos y fascinantes que enfrentamos hoy, el más intenso e importante es cómo entender y dar forma a la nueva revolución tecnológica, que implica nada menos que una transformación de la humanidad. Estamos al comienzo de una revolución que está cambiando fundamentalmente la forma en que vivimos, trabajamos y nos relacionamos unos con otros. En su escala, alcance y complejidad, lo que considero la cuarta revolución industrial no se parece a nada que la humanidad haya experimentado antes ([7], p.7).

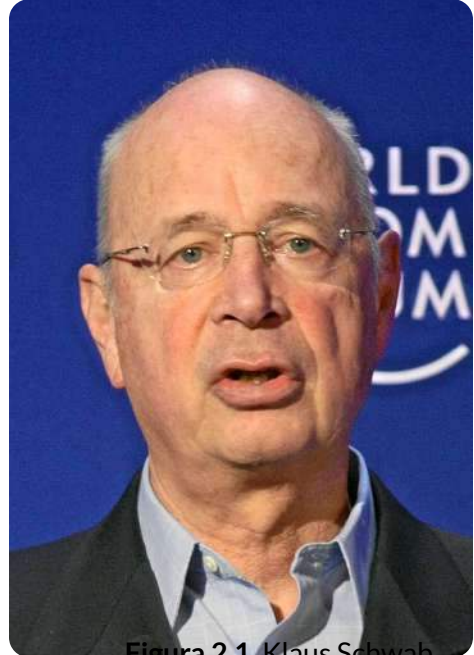


Figura 2.1. Klaus Schwab

Otras tecnologías de la 4RI

La oferta de desarrollos tecnológicos de la 4RI, es más amplia: drones, sensores inteligentes, controladores, plataformas electrónicas abiertas, sistemas de localización, sistemas de autoidentificación y blockchain son algunos de los muchos ejemplos que emergen por la convergencia de los pilares tecnológicos mencionados y juegan un rol igualmente relevante en la amplificación de la matriz tecnológica actual.

Son muchos los desarrollos tecnológicos de esta cuarta revolución industrial, que no sería posible imaginar las ciudades del futuro cercano.

Video



Video 2.1. Las ciudades del futuro (crédito: [Marketeer.co](https://www.marketeer.co) 📺).

Puzle con una imagen de la 4RI

Sugerencia: Arma, primero, el marco de la imagen y, si lo deseas, haz clic en el botón para ampliar el puzle en una ventana más grande.



2.2 Sistemas embebidos

En los hogares actuales se cuenta con gran cantidad de dispositivos que se usan a diario para controlar diferentes espacios y entornos, realizan procesos automáticos controlando diferentes elementos eléctricos o electrónicos mediante comandos de voz, el celular o elementos incorporados, incluso en relojes. Estos dispositivos se llaman "Sistemas Embebidos".

Los sistemas embebidos, hacen parte de una rama de la automatización denominada sistemas domóticos, compuestos por hardware (múltiples sensores, controladores, y desde luego los aparatos domésticos) y también por un software, encargado de proveer capacidad de gestionar las tareas en el hogar o entorno, por ejemplo el control de las luces en horarios estimados, controla luces o riego del jardín si la humedad no es la indicada.

La aplicación de los sistemas embebidos en los hogares se originó en 1934 en la ciudad de Chicago, EEUU, y luego en 1938 en la ciudad de New York, durante las ferias mundiales realizadas en dichas ciudades; además de estas ferias, a través de la historia se han dado diferentes inventos relacionados, tales como la televisión, internet y las computadoras.

Entre 1901 y 1920 surgieron los aparatos eléctricos en el ambiente denominado doméstico, con la idea de reemplazar las tareas del hogar con aparatos relativamente costosos para la época; entre los primeros aparatos en los hogares se encuentra la aspiradora, la plancha, la tostadora y la bombilla. Más tarde la televisión, en el año de 1928, fue presentada en Inglaterra y Estados Unidos.



Imagen generada por la IA [Leonardo](#)

El desarrollo de todos estos dispositivos procuraban que fueran de uso sencillo, que cualquiera los pudiera integrar en diferentes proyectos y además que fueran de bajo costo, así surgieron lo que actualmente se conoce como Sistemas Embebidos.

En el siguiente objeto interactivo, haz clic en cada año, para que conozcas la evolución histórica de los sistemas embebidos.



Interactivo 2.2. Evolución de las tecnologías hacia los sistemas embebidos.

2.2.1 Entre bits y bytes

Antes de adentrarnos en el interesante mundo de los sistemas embebidos, conoce las bases de funcionamiento de los computadores, para poder entender, así, la importancia y alcance de éstos, la funcionalidad que tiene en el quehacer diario y en las interacciones que se realizan mientras se hacen acciones como usar dispositivos inteligentes, crear un sistema para regar las plantas, alimentar las mascotas, desplazarse hasta la universidad en un vehículo o en el transporte público, entre otros.

Video



The video player shows a cartoon turtle on a rock. A speech bubble from the turtle contains the text: "Las Tecnologías de Información y Comunicación están cambiando las maneras en que las Personas hacen las cosas, veamos". The video progress bar shows 0:00 / 3:47.

Video 2.2. Pensamiento Computacional. Una habilidad de la era digital al alcance de todos (crédito: [Ramiro Lopera](#) .

Informática. Para poder entender los tipos de datos y su utilización en la programación de los dispositivos actuales o diferentes sistemas embebidos y los distintos lenguajes, primero hay que conocer y entender los fundamentos principales de la informática. A diario está

la necesidad de utilizar datos, recibir, procesar e intercambiar información, pero no se estás consciente de toda la lógica que ello implica; lo que lleva a una pregunta importante, ¿Qué se define como lógica en la informática? Algunos ingenieros la definen como una serie coherente de ideas y razonamientos; es "la forma más obvia y más fácil de hacer algo".

Toda la información que fluye al interior del sistema informático se basa en un elemento simple denominado **bit**, que es la unidad elemental de información que utilizan las computadoras y que sólo puede tener dos posibles valores 0 o 1, encendido-apagado, verdadero-falso, abierto-cerrado, un conjunto de 8 bits forman 1 byte por regla general, aunque las computadoras antiguas podían usar bytes de 6, 7, 8 o 9 bits. En el siguiente objeto interactivo, en el primer espacio, escribe una cadena con 1 o 0 en una cantidad de 8 dígitos para verificar a cuál carácter corresponde. En el segundo espacio, escriba un texto para convertirlo en código binario.

Activa los interruptores para calcular el valor binario

☐	2^8		☐	2^7		☐	2^6		☐	2^5		☐	2^4		☐	2^3		☐	2^2	
			☐	2^1		☐	2^0													

Video 2.3. Código ASCII.

Por ejemplo, la letra **A** en código ASCII es 65, y en código binario corresponde al **byte** (octeto) 01000001

Código binario. El sistema numérico binario fue el escogido por los ingenieros informáticos para el funcionamiento de los ordenadores, porque era más fácil para el sistema electrónico de la máquina distinguir y manejar solamente dos dígitos, o sea, el "0" y el "1", que componen el sistema numérico binario, en lugar de los diez dígitos (del 0 al 9), que constituyen el sistema numérico decimal.

Todos los programas, instrucciones, textos y órdenes que se introducen en el ordenador, son recibidos en código binario como una cadena de ceros y unos. Cada cero ("0") y cada uno ("1"), representa un **bit** de información. La palabra "**bit**" constituye el acrónimo de **B**inary **digIT**, que significa "dígito binario".

Para formar cada caracter alfanumérico, es decir una letra, número o signo, los ingenieros informáticos, después de realizar muchas pruebas, optaron por combinar ocho bits o cadena de ceros y unos para formar un **octeto** al que denominaron **byte**.

La combinación de este juego de caracteres permitió que los dispositivos informáticos, desde el más sencillo hasta el más completo, pudieran compartir información entre ellos y poder representar la información de manera unificada.

Código ASCII. A cada caracter alfanumérico se le asignó un **byte** de información y se estructuraron 256 valores básicos binarios distintos en un código que llamaron **ASCII** (*American Standard Code for Information Interchange* - Código Estándar Americano para Intercambio de Información).

En el Código ASCII los valores binarios entre 0 y 31 corresponden

a instrucciones, entre 32 y 127 corresponden al alfabeto alfanumérico y entre 128 y 255 a caracteres de otros idiomas y signos menos convencionales. Estos códigos los utilizas a diario en todos los dispositivos electrónicos que manipules.

Haz clic sobre la siguiente imagen, para que veas en pantalla ampliada la tabla ASCII en los sistemas decimal, binario, octal y hexadecimal.

ASCII TABLE															
Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140		
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a	
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b	
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c	
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d	
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e	
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f	
7	7	111	7	[BEL]	55	37	110111	67	7	103	67	1100111	147	g	
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h	
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i	
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j	
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k	
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l	
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m	
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n	
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o	
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p	
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q	
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r	
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s	
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t	
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u	
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v	
23	17	10111	27	[END OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w	
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x	
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y	
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z	
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{	
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174		
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}	
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~	
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]	
32	20	100000	40	[SPACE]	80	50	1010000	120	P						
33	21	100001	41	!	81	51	1010001	121	Q						
34	22	100010	42	"	82	52	1010010	122	R						
35	23	100011	43	#	83	53	1010011	123	S						
36	24	100100	44	\$	84	54	1010100	124	T						
37	25	100101	45	%	85	55	1010101	125	U						
38	26	100110	46	&	86	56	1010110	126	V						
39	27	100111	47	'	87	57	1010111	127	W						
40	28	101000	50	(	88	58	1011000	130	X						
41	29	101001	51	)	89	59	1011001	131	Y						
42	2A	101010	52	*	90	5A	1011010	132	Z						
43	2B	101011	53	+	91	5B	1011011	133	[						
44	2C	101100	54	,	92	5C	1011100	134	\						
45	2D	101101	55	-	93	5D	1011101	135	]						
46	2E	101110	56	.	94	5E	1011110	136	^						
47	2F	101111	57	/	95	5F	1011111	137	_						

Figura 2.2. Tabla ASCII (Fuente: Wikipedia)

En la página siguiente, presentamos un objeto interactivo diseñado por [Monkey Raptor](#), en el que puedes separar los caracteres de una palabra en octetos (**bytes**).



Separando

Escribe aquí

Dividir los caracteres

space	!	"	#	\$	%	&	'	(	)	*
32	33	34	35	36	37	38	39	40	41	42
+	,	-	.	/	0	1	2	3	4	5
43	44	45	46	47	48	49	50	51	52	53
6	7	8	9	:	;	<	=	>	?	@
54	55	56	57	58	59	60	61	62	63	64
A	B	C	D	E	F	G	H	I	J	K
65	66	67	68	69	70	71	72	73	74	75
L	M	N	O	P	Q	R	S	T	U	V
76	77	78	79	80	81	82	83	84	85	86
W	X	Y	Z	[	\	]	^	_	`	a
87	88	89	90	91	92	93	94	95	96	97
b	c	d	e	f	g	h	i	j	k	l
98	99	100	101	102	103	104	105	106	107	108
m	n	o	p	q	r	s	t	u	v	w

Interactivo 2.3. Separando palabras en bytes.

2.2.2 Los sistemas embebidos

Los Sistemas Embebidos están diseñados para funciones particulares y son una integración entre software y hardware, usualmente están contenidos en aparatos domésticos, autos, incluso en teléfonos móviles; generalmente están compuestos por microcontroladores o en algunos casos pueden contener también **microprocesadores**.

Microprocesador

Es un dispositivo procesador que tiene integrada una **Unidad Central de Procesamiento (CPU)** de un dispositivo de cómputo, en un circuito integrado único.

El primer microprocesador, llamado Intel 4004, fue lanzado en 1971, en el que se requerían chips exteriores para generar sistemas funcionales, incrementando sus costos y haciendo más complejo la integración de aplicaciones o limitándose a sectores exclusivos.

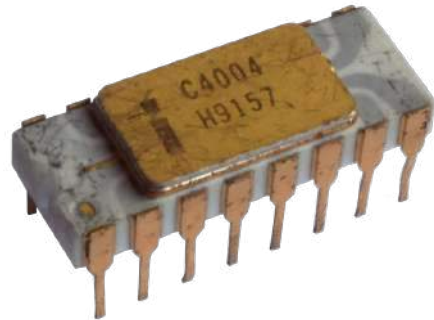


Figura 2.3. Microprocesador Intel 4004
(crédito [pngwing](#))

El microprocesador se encarga de la ejecución de los programas y software en el computador, tiene la novedad de que sólo ejecuta comandos e instrucciones en lenguaje máquina (bits y bytes), procesando cálculos y operaciones aritméticas y lógicas simples, tales como sumar, restar, multiplicar, dividir, también las lógicas binarias y accesos a memoria.

Microcontrolador

Es una **pequeña computadora** en un único circuito integrado, que a su vez integra un procesador, la memoria, y varios periféricos de entrada - salida y tienen la característica de ser programables. Estos se usan en productos que pueden ser **controlados automáticamente**, tales como: máquinas de entornos de oficina, elementos electrónicos médicos, vehículos, y diferentes sistemas automatizados, es decir, están presentes en casi todos los dispositivos electrónicos que se utilizan actualmente.

Si bien el término **Sistemas Embebidos** suena bastante extraño, son dispositivos que se encuentran en interacciones diarias, por ejemplo el dispositivo que calcula el cobro en el taxi, el lector del carnet de estudiante en la universidad, o incluso el teléfono móvil; son importantes para proyectos específicos y pequeños, en el desarrollo de tareas puntuales.

Algunas características de los sistemas embebidos son:



Cuentan con sensores y actuadores



No son equivalentes a sistemas de cómputo como los portátiles o equipos de escritorio



Tienen recursos limitados y aplicaciones específicas que los hacen muy útiles



Tamaños regularmente pequeños



Bajo costo



Consumos de energía bajos



No requieren conocimientos avanzados para su manipulación

2.2.3 Componentes de los sistemas embebidos

Son varios tipos de componentes que podemos encontrar en un sistema embebido. Veamos, primero, los relacionados con el hardware:

 **Microcontrolador.** Es el cerebro del sistema embebido y se encarga de procesar la información y controlar el funcionamiento del sistema. Está compuesto por una unidad central de procesamiento (CPU), memoria RAM, memoria ROM y periféricos.

 **Dispositivos de Entrada/Salida (E/S).** Permiten la interacción con el mundo exterior. Algunos ejemplos son puertos USB, puertos seriales, interfaces de red, etc.

 **Sensores.** Detectan y recopilan información del entorno. Algunos ejemplos son sensores de temperatura, de luz, de movimiento, etc., pueden ser Sensores analógicos o digitales.

 **Actuadores.** Controlan y modifican el entorno en base a la información de los sensores. Algunos ejemplos son motores, LEDs, válvulas, etc.

 **Placas de Circuito Impreso.** Es el componente que conecta eléctricamente y soporta físicamente todos los otros componentes mediante el uso de pistas, pads, y otras características grabadas en láminas de cobre laminadas sobre y/o entre capas de sustrato no conductor.

Los componentes tipo software, pueden ser:

 **Sistema operativo.** Controla la ejecución del software y administra los recursos del sistema.

 **Software de aplicación** Implementa la funcionalidad específica del sistema embebido.

En el siguiente objeto interactivo, pasa el puntero del mouse sobre las imágenes que representan componentes de un sistema embebido.



Interactivo 2.4. Componentes de los sistemas embebidos.

Otros componentes son la fuente de alimentación y una carcasa para proteger los componentes internos del sistema.

El diseño o arquitectura de un sistema embebido, puede ser como una tarjeta electrónica a la que se le conecta diferentes sensores y dispositivos, o más complejo asemejándose a una mini-computadora;

en ambos se cumplen con tareas específicas programadas mediante diferentes lenguajes de programación o sistemas operativos, según sea el caso. Aunque hay muchas placas y dispositivos embebidos en el mercado, hay **dos** que son muy populares en la comunidad de estudiantes y desarrolladores, que suelen llamarse comunidades Maker o DIY (Acrónimo en inglés de Do It Yourself o Hazlo tú mismo), estas son el Arduino y el Raspberry Pi.

2.2.4 El Arduino

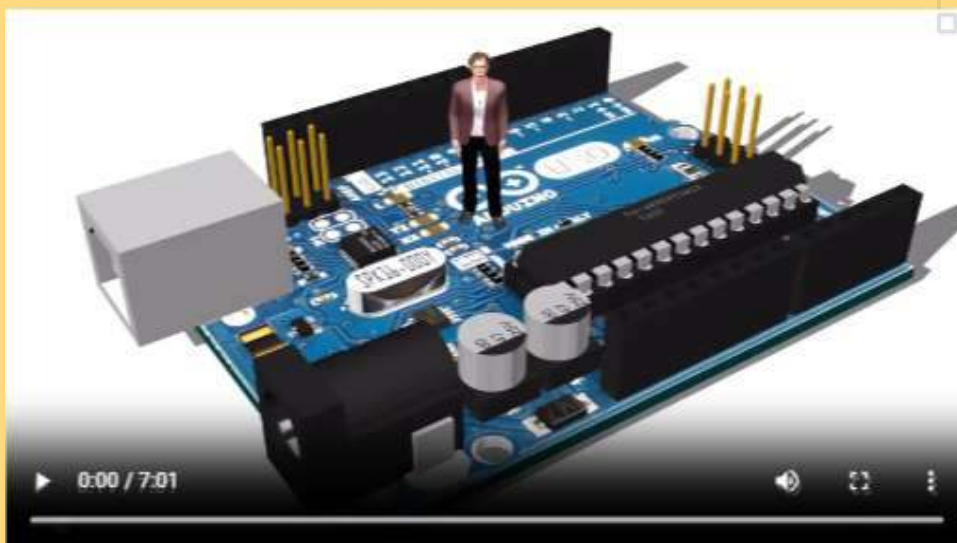
Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.



Una placa electrónica es una PCB (*Printed Circuit Board* o, en español, "Placa de Circuito Impreso"). Las PCBs son superficies planas fabricadas en un material no conductor, la cual consta de distintas capas de material conductor. Una PCB es la forma más compacta y estable de construir un circuito electrónico. Por lo tanto, la placa Arduino no es más que una PCB que implementa un determinado diseño de circuitería interna. De esta forma el usuario final no se debe preocupar por las conexiones eléctricas que necesita el microcontrolador para funcionar, y puede empezar directamente a desarrollar las diferentes aplicaciones electrónicas que necesite.

En el siguiente video se explica qué es Arduino y los diferentes tipos de placas.

Video



Video 2.4. Qué es Arduino y diferentes tipos de placas (crédito: [Asociación de Robótica y Domótica de España](#) , YouTube con Licencia Atribución de Creative Commons).

2.2.5 El Raspberry Pi

La **Raspberry Pi** es una computadora de bajo costo y con un tamaño compacto, del porte de una tarjeta de crédito, puede ser conectada a un monitor de computador o un TV, y usarse con un mouse y teclado estándar. Es un pequeño computador que ejecuta un sistema operativo linux capaz de permitirle a las personas de todas las edades explorar la computación y aprender a programar lenguajes como Scratch y Python. Es capaz de hacer la mayoría de las tareas típicas de un computador de escritorio, desde navegar en internet, reproducir videos en alta resolución, manipular documentos de ofimática, hasta

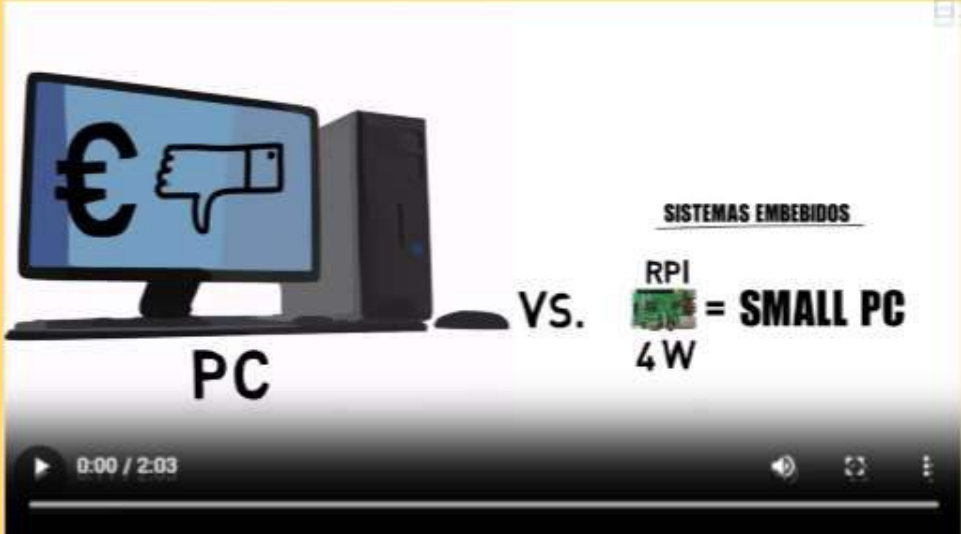


reproducir juegos

Además la Raspberry Pi tiene la habilidad de interactuar con el mundo exterior, puede ser usada en una amplia variedad de proyectos digitales, desde reproductores de música y video, detectores de padres, estaciones meteorológicas hasta cajas de aves con cámaras infrarrojas. La Raspberry Pi puede ser usada por niños y adultos por todas partes del mundo, para aprender a programar y entender cómo funcionan las computadoras ([Raspberry Pi Chile](#)).

Hemos tomado un fragmento del video "El controlador domótico", en el que se muestran algunos modelos de Raspberry Pi:

Video



Video 2.5. Modelos de Raspberry PI (crédito: [Tknika - LH, FP, VET](#) , YouTube con Licencia Atribución de Creative Commons).

Finalmente, los sistemas embebidos han facilitado la utilización de dispositivos sencillos y fácilmente programables para cualquier

persona, sin conocimientos avanzados de electrónica o programación, facilitando tener en la palma de la mano todo tipo de aplicaciones, por ejemplo juegos integrados en consolas personalizadas tipo arcade o integraciones para huertas caseras, alimentadores de mascotas, accesorios para los vehículos, elementos prácticos para bicicletas, entro muchos otros.

2.3 El Arduino y su impacto en IoT

En el apartado anterior, dimos una breve introducción al Arduino. En este apartado, profundizaremos un poco más sobre este sistema embebido y su impacto en IoT.

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra, que permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera sencilla.

Arduino es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Los productos que vende la compañía son distribuidos como Hardware y Software Libre, bajo la Licencia Pública General de GNU (GPL) y la Licencia Pública General Reducida de GNU (LGPL), permitiendo la manufactura de las placas Arduino y distribución del software por cualquier individuo. Las placas Arduino están disponibles comercialmente en forma de placas ensambladas o también en forma de kits, hazlo tú mismo (del inglés DIY: *Do It Yourself*) ([Wikipedia](#)).

Hay una gran cantidad de modelos "Arduino". Se han fabricado diferentes modelos de placas Arduino oficiales, cada una pensada con un propósito diferente y características variadas (como el tamaño físico, número de pines E/S, modelo del microcontrolador, etc). A pesar de las varias placas que existen todas pertenecen a la misma familia (microcontroladores AVR marca Atmel). Esto significa, que comparten la mayoría de sus características de software, como arquitectura, librerías y documentación.

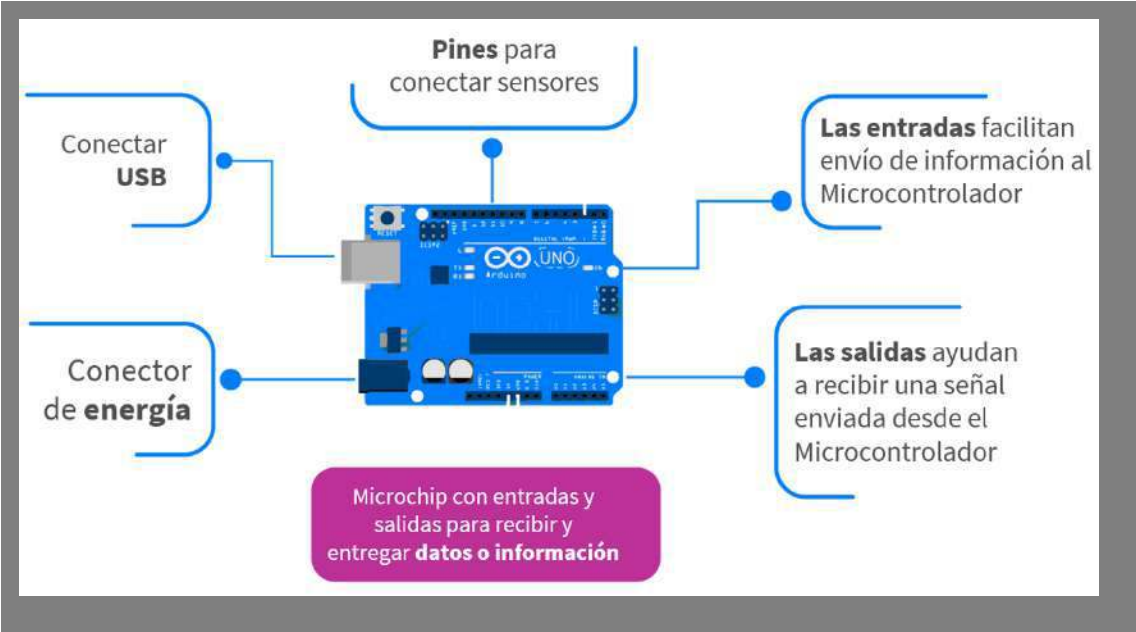


Figura 2.4. Arduino

Modelo	Descripción
Arduino Nano 	Es una placa pequeña, completa y compatible con placas de pruebas, funciona con un cable USB Mini-B en lugar de uno estándar.

Modelo	Descripción
Arduino UNO 	UNO REV 4 ofrece una potencia de procesamiento, capacidad de memoria y funcionalidad general impresionantes. Las placas Arduino UNO REV 4 cuentan con dos versiones, UNO R4 Wi-Fi y UNO R4 Minima. El UNO R4 Minima es rentable sin comprometer las capacidades principales del microcontrolador. El UNO R4 Wi-Fi incorpora un módulo Wi-Fi Espressif S3 para conectividad inalámbrica.
Arduino Mega 	Es una tarjeta basada en el microcontrolador ATmega2560. Tiene 54 pines (Entrada/Salida) digitales, 16 entradas análogas, 4 UARTs (puertos seriales por hardware), oscilador de cristal a 16MHz, conexión USB, jack de energía, conector ICSP y botón de reset.
Arduino Leonardo 	Placa con microcontrolador basado en el ATmega32u4. Posee 20 pines de entrada/salida digital, de los cuales 7 pueden ser usados como salidas PWM, y 12 como entradas analógicas, un oscilador de cristal de 16 Mhz, conexión micro usb, entrada eléctrica, un header ICSP y un botón de reset.
Arduino Yun 	Arduino “Yún”, que significa "nube" en chino, combina la placa Arduino Leonardo existente, basada en el microcontrolador Atmel ATmega32u4 de 8 bits, con un soporte WiFi integrado.

2.3.1 Un poco de historia del Arduino

Arduino nació en el año 2005 en el Instituto de Diseño Interactivo de Ivrea (Italia). Arduino apareció por la necesidad de contar con un dispositivo para utilizar en aulas que fuera de bajo costo. La idea original fue fabricar una placa para uso interno de la escuela. Sin embargo, el instituto se vio obligado a cerrar sus puertas precisamente en 2005. Ante la perspectiva de perder todo el proyecto Arduino en el proceso, se decidió liberarlo y abrirlo al público para que todo el mundo pudiese participar en la evolución del proyecto, proponer mejoras y sugerencias.



Figura 2.5. Instituto de Diseño Interactivo de Ivrea en 2005 ([Designswarmg](#), CC BY-SA 4.0).

Los principales responsables de la idea y diseño de Arduino fueron Massimo Banzi (Italia), Hernando Barragán (Colombia), David Cuartielles (España), David Mellis (Estados Unidos), Tom Igoe (Estados Unidos) y Gianluca Martino (Italia).

Algo curioso con el nombre es su origen en el nombre de un **bar**. En el *Bar di Re Arduino* Massimo Banzi pasaba algunas horas, el cual a su vez viene del nombre de un antiguo rey europeo allá por el año 1002.

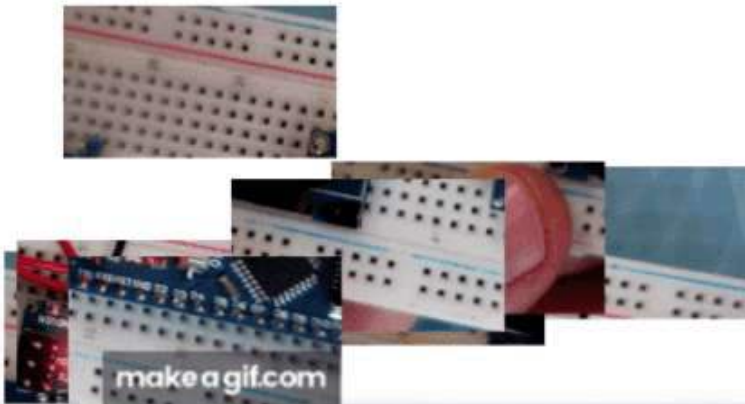
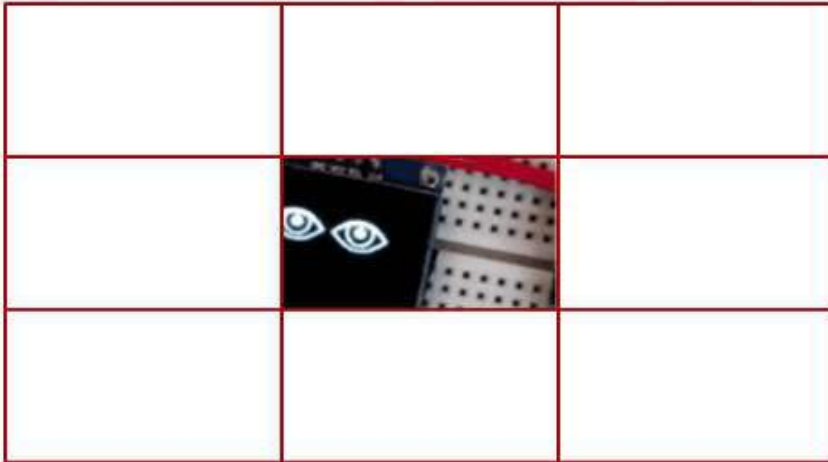
En la feria **Maker Fair** de 2011, se presentó la **primera placa Arduino 32 Bit** para trabajar tareas más pesadas. Entre ellas se presentó la impresora 3D de MakerBot, capaz de imprimir en resina cualquier modelo en 3D.

¡Ya es momento de una pausa lúdica!

Puzle animado



Arma la imagen animada



Procura no montar imágenes

<https://makeagif.com/>

2.3.2 Ventajas del Arduino

En el sitio web de [Arduino](#) se le promueve como un recurso libre y extensible: "Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Está destinado a cualquiera que realice proyectos interactivos". Según esta declaración, cualquiera que desee ampliar y mejorar el diseño hardware de las placas como el entorno de desarrollo, puede hacerlo sin problemas. Esto permite que exista un rico ecosistema de placas electrónicas no oficiales para distintos propósitos y de librerías de software de terceros, que pueden adaptarse mejor a necesidades.

Entre sus principales características:

-  La gran y creciente comunidad de entusiastas del Arduino, que cada día crece y aporta nuevos proyectos, contenidos y documentación.
-  Su lenguaje de programación está basado en C++, con comandos sencillos y con instrucciones de baja complejidad.
-  Su costo está muy por debajo de dispositivos electrónicos programables y tiene además la posibilidad de reprogramarse cuantas veces sea necesario.
-  El entorno de aprendizaje es simple y se hace muy divertido desarrollar proyectos, ya que la curva de aprendizaje es alta.

Para crear proyectos con Arduino no es necesario ser experto en lenguajes de programación o estudiar previamente sobre electrónica, ya que la curva de aprendizaje con esta tarjeta, y su entorno de desarrollo, es alta y está pensada para ser aplicada por usuarios desde edad temprana hasta adultos de diferentes disciplinas.

Arduino se puede utilizar para integrar proyectos interesantes y

dinámicos, tan sencillos como la creación de juguetes, tales como drones, autos a control remoto o la construcción de algo tan complejo como un robot a control remoto hasta un jardín o una casa inteligente, es fácil integrarlo en proyectos como sistemas de riego, iluminación, puertas automáticas, entre otros, todo esto en función de monitorear o controlar dispositivos de manera remota y confiable.

Para ello, hay diferentes plataformas de costo o gratuitas, que facilitan la gestión de dichos proyectos de manera amigable y divertida, apoyando y fortaleciendo la comunidad DIY que ya hemos mencionado y aportando a la cultura MAKER de todo el mundo.

Algunas plataformas conocidas son Arduino, Cayenne, Ubidots, Blynk, Node-RED, ThingsBoard, entre otras más.



Figura 2.6. Entorno de programación Arduino

2.4 El Raspberry Pi y su impacto en IoT

Durante la época de los 80 se vivió una época dorada en la microinformática, pues cualquier persona tenía la oportunidad de adquirir el microprocesador Zylog **z80**, un elemento de bajo costo, para poder desarrollar sus programas y así diseñar y construir proyectos personales más estructurados; sin embargo no fue hasta febrero 25 de 2012 que se oficializó la placa Raspberry Pi.

El Director de Estudios de la Universidad de Cambridge, en el Reino Unido, analizó en el año de 2006 las pocas matrículas hacia los estudios en informática, un porcentaje decreciente en los últimos años de esa década; esta persona era Eben Upton, quien se interesó en fomentar el espíritu creativo en los estudiantes centrándose en los microordenadores que estaban en auge durante los años 80.

En el año de 2009 se constituyó la Raspberry Pi Foundation, una organización sin ánimo de lucro que velaría por los intereses y valores de este importante proyecto muy popular entre las comunidades de informáticos, comunidades Maker y entusiastas de la cultura DIY.

Después de alrededor de dos años realizando una búsqueda de un procesador, Eben Upton y el ingeniero electrónico Pete Lomas, modificaron el procesador chip Broadcom BCM2835, que se podía



Figura 2.7. Eben Upton, cofundador Raspberry Pi Foundation (foto de [Door Jim Killock](#), CC BY-SA 2.0)

integrar en pequeños electrodomésticos y avisos digitales, para definirlo como cerebro de la denominada Raspberry Pi (Por aquellos años estaba de moda bautizar a los sistemas con nombres de fruta, moda inspirada en los Apple, Apricot Computers y Acorn Computers).

Uno de los conceptos que facilitó la popularidad de esta placa fue el sistema operativo, el Linux (Desarrollado en 1991 por Richard Stallman, un genio informático revolucionario y programador del MIT). Una distribución o distro de Linux no es más que una versión personalizada del sistema operativo original, el kernel o núcleo de Linux. ¿Sabías que existen más de 600 distribuciones Linux? Esto se debe al carácter abierto, la condición de código libre de Linux, que cualquiera puede ampliar y modificar.

Video



Video 2.6. ¿Que es Raspberry Pi? (crédito: [Raspberry Pi](#) , YouTube con Licencia Atribución de Creative Commons).

2.4.1 Historia del Raspberry Pi

La Raspberry Pi Foundation se creó en 2009, en el mes de mayo. Con sede en Cambridge, Reino Unido, su propósito es promover el estudio de la informática o ciencias computacionales en las escuelas. Y para hacer esto posible, desarrolla sus ordenadores Raspberry Pi. Precisamente, su clave de éxito son estos pequeños dispositivos. En la página siguiente, puedes ver la evolución histórica de este desarrollo tecnológico.

2.4.2 Software que se puede usar en una Raspberry Pi

Una de las cosas interesantes, prácticas y quizás divertidas de la Raspberry es su funcionalidad en cuanto al software se refiere, pues en pocos segundos puede transformarla en una divertida consola de juegos, una laptop para trabajos del colegio, una plataforma de programación que se puede integrar incluso con arduino, u otra gran cantidad de proyectos interesantes.

Un ejemplo de uso de la Raspberry Pi, es un proyecto de mini computador con Raspberry Pi con elementos impresos en 3D. La Raspberry Pi usa principalmente sistemas operativos GNU/Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse. Aunque uno de los recomendados para estudiantes es la versión Pi-Top OS, muy útil para iniciar en proyectos con Arduino o robótica.

Otra opción recomendada, si no se desea aprender procesos complejos, es el uso de la aplicación NOOBS (*New Out of Box Software*), utilidad que facilita la instalación de diferentes sistemas operativos para Raspberry Pi.

Línea de tiempo del Raspberry Pi

2012

Raspberry Pi 1 Modelo B. La primera versión de Raspberry Pi causó un gran revuelo en el mercado de las pequeñas computadoras de placa única (SBC). Este dispositivo era pequeño, competente y asequible.



Desplaza el punto naranja a lo largo de la línea de tiempo o haz clic en alguno de los puntos.

Zoom



2013

2014

2015

2017

2018

2020



2012

2014

2015

2016

2018

2019

2023

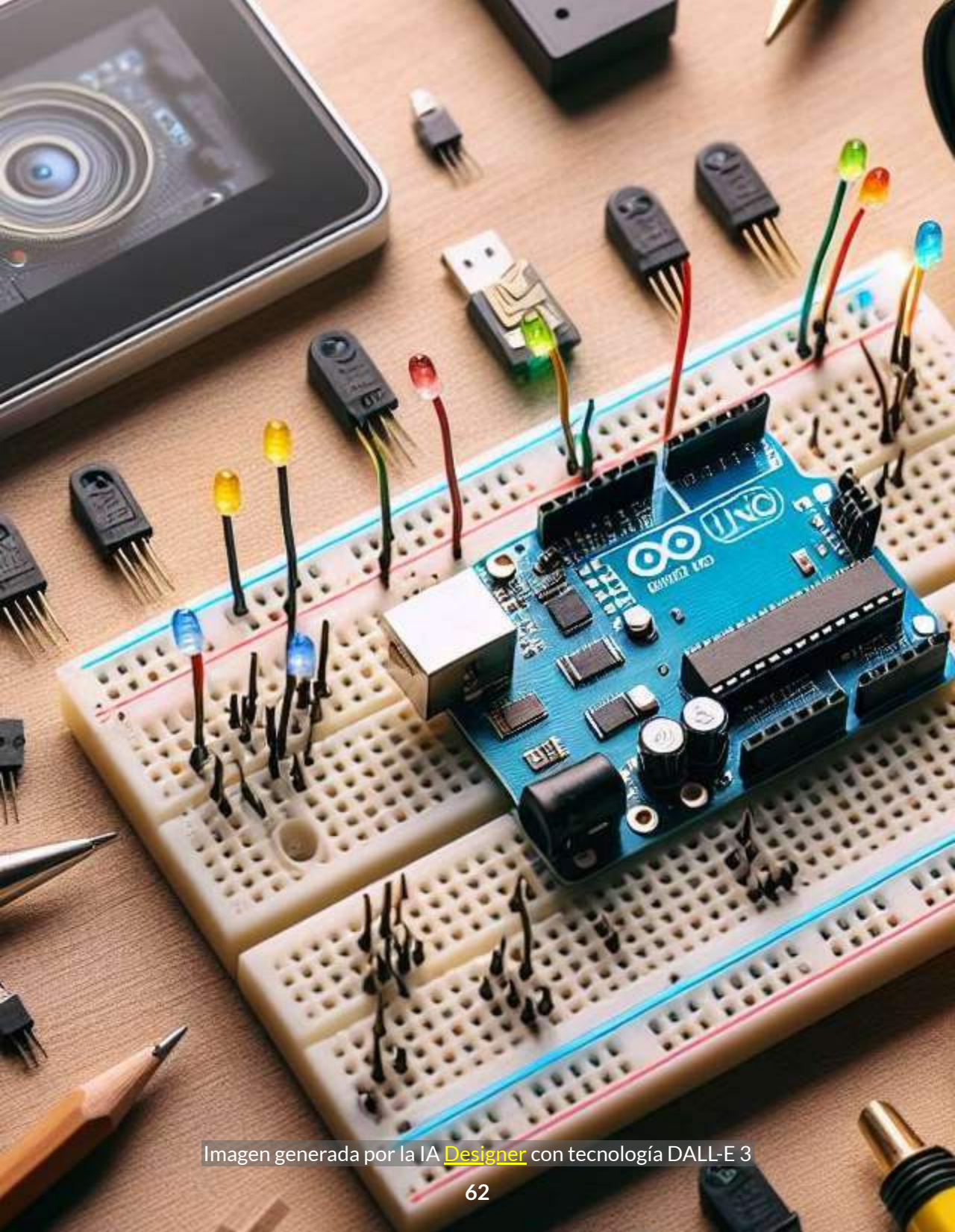


Imagen generada por la IA [Designer](#) con tecnología DALL-E 3



10 preguntas en 40 segundos

Comenzar

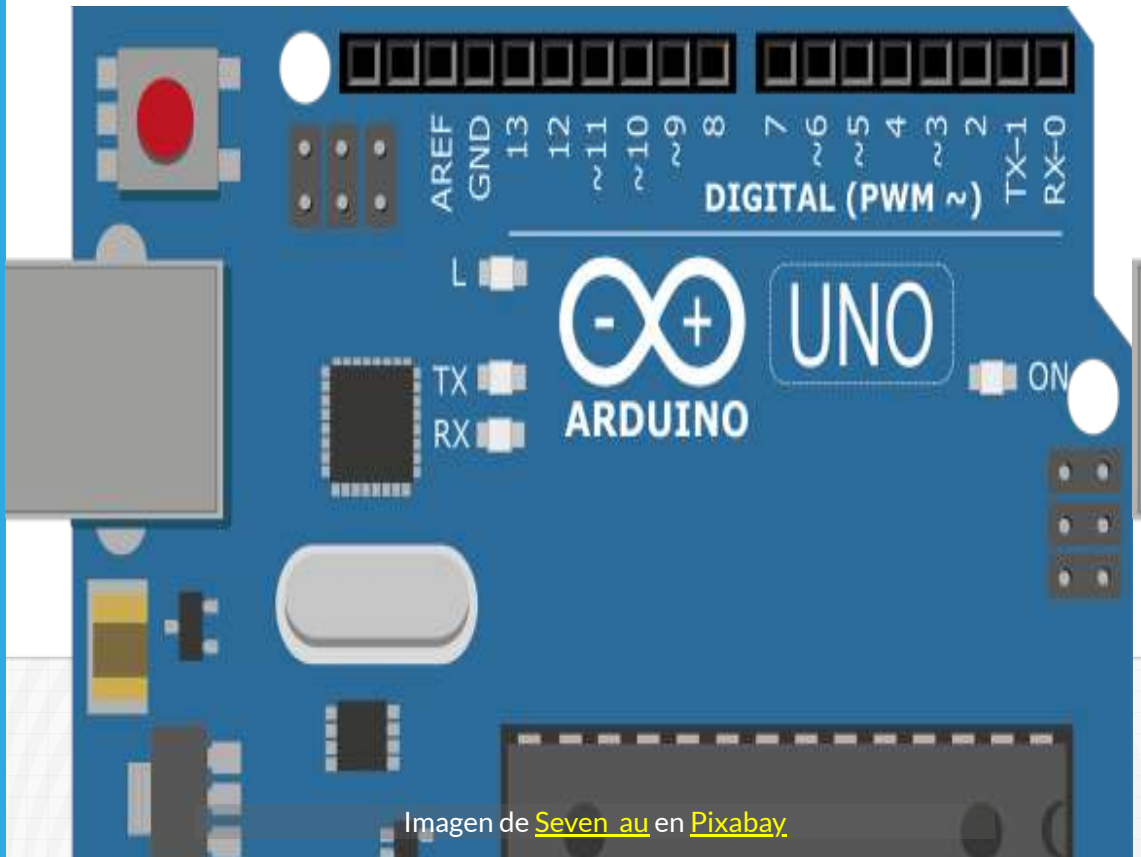


Imagen de [Seven au](#) en [Pixabay](#).

Capítulo 3

Programando el IoT

3.1 Programación en IoT - Sensores y datos

En un mundo en el que los dispositivos interconectados a rutinas pueden ajustarse a decisiones, es importante conocer cómo funcionan, y para ello se requiere entenderlos; por ejemplo: al desactivar la alarma del despertador en la mañana, iniciar la cafetera automáticamente para la preparación del café, considerando las recomendaciones para mantener la salud. O que el reloj inteligente tenga en cuenta la ubicación y avise cuando estés llegando a casa. Hay muchas posibilidades en las que el IoT mejore nuestro estilo de vida, interactuando con las actividades que realizas y transformándose para que prime el bienestar y la seguridad.

Para que todas estas actividades puedan ser realizadas, se requiere medir todos los cambios que se generan en el entorno, esto es posible gracias a los **sensores**, dispositivos que registran los cambios y los envían y almacenan en dispositivos más complejos y estructurados, dando orden a la gran cantidad de datos producidos.

En este capítulo, vas a conocer los fundamentos de estas nuevas tecnologías, sus orígenes y aplicaciones y, en especial, los impactos que afectan positiva o negativamente a la sociedad.





Imagen generada por la IA [Designer](#) con tecnología DALL-E 3

3.1.1 Sensores y datos en un entorno de ciudad

Varias décadas atrás, la gestión de programar los sistemas de cómputo era tarea exclusiva de los equipos informáticos, pero con los avances de la cuarta revolución industrial y la creciente estrategia de formación en la ciudad, en habilidades relacionadas con la programación y algoritmos, se hace cada vez más necesario que las personas, sin importar su profesión, puedan programar. Tal como Steve Jobs dijo: "Todos en el país deberían aprender a programar un ordenador... porque eso te enseña a pensar".



Figura 3.1. Sensores y datos en un entorno de ciudad

Hoy en día es muy común ver a profesionales en todas las áreas dar valor agregado a sus tareas y actividades diarias, mediante la utilización de la programación que se ha convertido en una herramienta que facilita muchas tareas repetitivas.

En el siguiente objeto interactivo, observa algunos conceptos básicos de programación:



Recuerda:

01 02 03

04 05

Pasa el puntero del mouse por las tarjetas de arriba

Un lenguaje de programación es una forma de comunicarnos con una computadora, tablet o celular, e indicarle qué queremos hacer, sin distinción del idioma en que se estén realizando las instrucciones

Interactivo 3.1. Concepto básicos de programación.

3.1.2 Algoritmos

Es un conjunto de pasos organizados que lleva a la solución de un problema específico; estos pasos pueden ser expresados en un lenguaje natural como el español o el inglés. Un algoritmo debe ser preciso, es decir que sus pasos o instrucciones deben ser fáciles de entender y realizar. Debe ser finito, es decir que las instrucciones tengan un inicio y un fin, y debe estar bien definido, que tenga una estructura clara con indicaciones de inicio, un procesamiento de sus datos y un fin o salida de información clara.

Si la solución propuesta es diseñada mediante un sistema informático, esta debe ser descrita mediante un algoritmo.

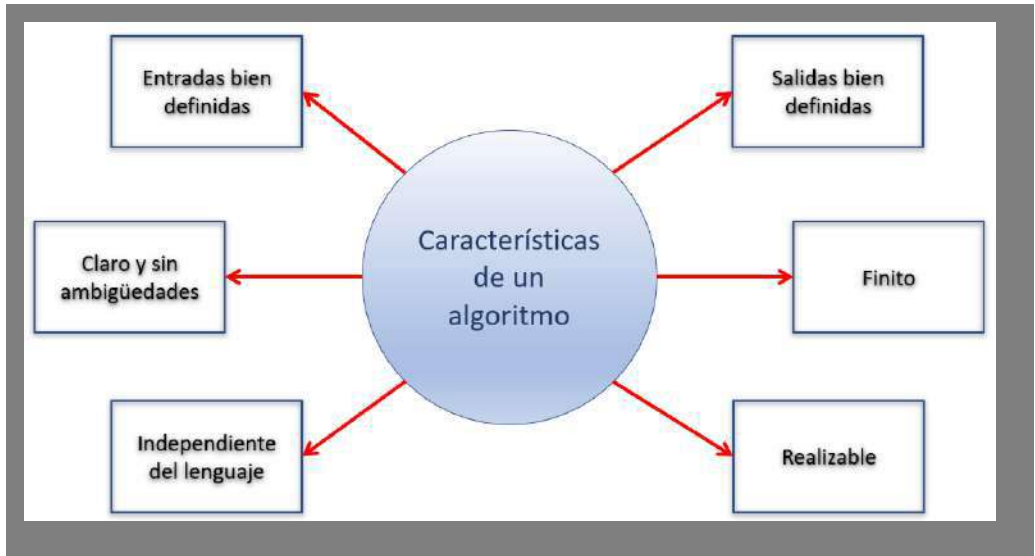


Figura 3.2. Características de un algoritmo

Entre los tipos de algoritmos se encuentran los de tipo:



Cualitativo. Se trata de aquellos en donde los pasos o instrucciones se describen mediante palabras y en donde se relacionan características medibles de un objeto. Por ejemplo, cuando se indica los pasos para cambiar un bombillo, tendrás las instrucciones de modo textual, cómo colocar la escalera, elegir el bombillo, subir a la escalera, girarlo y cambiarlo, bajar de la escalera y activar el interruptor.



Cuantitativo. Son aquellos en donde se requieren cálculos numéricos para definir los pasos del proceso y para las respuestas solicitadas; por ejemplo, para saber la edad de una persona, le solicitas su fecha de nacimiento, calculas contra el año actual, realizas la operación matemática de resta de fechas y presentas el número obtenido en formato de edad.



pseudocódigo. Estos pasos se representan y se organizan en instrucciones más claras. Se trata de una representación NO gráfica en la que NO se usan símbolos y los pasos o instrucciones (sentencias o comandos) se escriben mediante un conjunto de palabras reservadas o propias de este lenguaje. Al escribir un algoritmo en pseudocódigo se debe usar sangría en el margen izquierdo de algunas instrucciones. Este lenguaje es similar al que se usa para codificar los programas en una computadora, por ello se le denomina pseudolenguaje o pseudocódigo. Por ejemplo:

```
Inicio
Entero a, b
Real c
Imprimir "ingresa un número a y un número b en
valor entero"
Leer a,b
Si a > b Entonces
    R ← "a mayor que b"
Sino
    R ← "a no es mayor que b"
FinSi
Imprimir "El resultado es", R
final
```



3.1.3 Teorema del programa estructurado

Quizá algunos se sientan desanimados, pues no esperaban nada de matemáticas y menos de programación. No obstante, como ser humano que eres siempre aplicas algo de lógica en tus decisiones, desde la simple decisión de ir al trabajo a la más compleja que es decidir sobre tu proyecto de vida. No te preocupes que este curso está diseñado para tratar conceptos básicos o, si se prefiere, elementales de programación. Sabías, por ejemplo, que Edsger Dijkstra, un experto en programación (considerado por algunos como el padre de la programación estructurada), nunca usó un ordenador para ejecutar y verificar sus algoritmos... pues, todo se reduce a la lógica... la lógica de programación³.

Para que quedes más tranquilo, algunos científicos de la computación (Dijkstra, Jacopini, Böhm, von Neumann,...), estuvieron de acuerdo en afirmar que todo programa se puede realizar combinando sólo tres estructuras lógicas o de control:

- Estructuras secuenciales
- Estructuras selectivas
- Estructuras iterativas o repetitivas.

3.1.4 Estructura secuencial

Una estructura secuencial es aquella en la que las instrucciones se ejecutan una después de la otra, en el orden en que están escritas, es

³ Aunque parezca irónico, Dijkstra, uno de los mayores desarrolladores de software de su época, evitó el uso de computadores en su trabajo durante décadas. Cuando, finalmente, sucumbió a la tecnología, únicamente utilizó los ordenadores para enviar correos electrónicos y hacer búsquedas en la red. Dijkstra nunca utilizó un computador para realizar ninguno de sus trabajos, todos ellos fueron realizados a mano. (crédito: https://es.wikipedia.org/wiki/Edsger_Dijkstra).

decir, en secuencia. En una estructura secuencial se espera que:

se proporcione uno o varios datos, los cuales son asignados a variables para que con ellos se produzcan los resultados que representen la solución del problema que se planteó (<https://www.uaa.mx>).

Problema 1. A un estudiante se le aplican tres exámenes durante el semestre, cada examen tiene el mismo valor porcentual. Elabora un pseudocódigo para obtener el promedio de calificaciones.

Observa el pseudocódigo para la solución de este problema:

1. Inicio
2. Leer Nota1, Nota2, Nota3
3. Hacer $\text{Suma} = \text{Nota1} + \text{Nota2} + \text{Nota3}$
4. Hacer $\text{Prom} = \text{Suma}/3$
5. Escribir Prom
6. Fin

Observa algunas características del pseudocódigo:

Entrada (input): se utiliza la palabra reservada **Leer**. Este tipo de instrucción permite asignar un valor a una variable a partir de lo que digite un usuario. A esto se le denomina una **asignación externa**.

Salida (output): se utiliza la palabra reservada **Escribir**, **Imprimir** o **Mostrar**. Este tipo de instrucción permite mostrar por pantalla o por impresora los resultados esperados en la solución de un problema.

Asignación interna: corresponde a aquellas instrucciones donde se realizan cálculos que se asignan a variables o valores específicos. En una variable se reconocen dos elementos: Su nombre y su contenido (dato que se almacenará).

La secuencia o instrucción del tipo **Hacer**, como la cuarta instrucción del pseudocódigo anterior (**Hacer Prom = Suma/3**) es una **asignación interna**, que no se debe entender como una igualdad. Su interpretación debe ser de la siguiente forma: "se realiza la operación **Suma/3**, el resultado se **asigna** a la variable **Prom**". La asignación funciona de derecha a izquierda.

Para evitar este tipo de confusiones, algunos programadores prefieren usar la siguiente convención:

Prom ← Suma/3

En lenguajes de programación como Pascal o Delphi: **Prom := Suma/3** o en lenguaje C: **Prom /= 3**, siempre buscando evitar la confusión y comprender que asignaciones como **c = c + 1** son completamente válidas⁴.

Al lado derecho, hemos representado gráficamente las secuencias del pseudocódigo anterior. Esta representación, se conoce como **diagrama de flujo**. Observa que el "flujo" es secuencial.

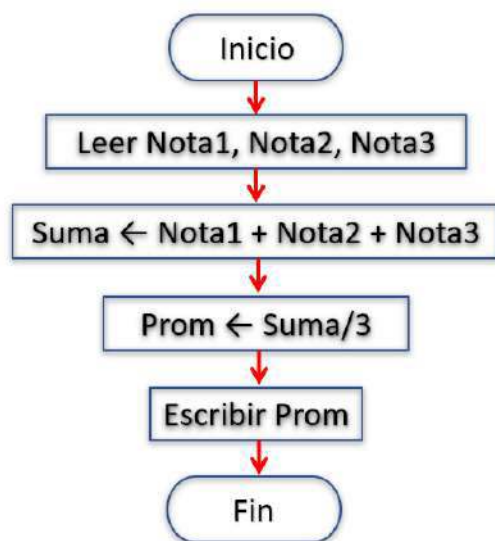


Figura 3.3. Diagrama de flujo

En la página siguiente, hemos diseñado un objeto interactivo que sigue las secuencias anteriores.

⁴ La asignación **c = c + 1** se conoce como **contador**, pues en una estructura iterativa (como veremos más adelante), a la variable **c** se le suma el valor de **1**, **N** veces, es decir, si **c** es inicialmente cero, se le irá asignado los valores **1, 2, 3, ..., N**.



Estructura secuencial

Ingresa tres notas, presionando enter en cada una

Nota1

Nota2

Nota3

Interactivo 3.2. Uso de la estructura secuencial.

3.1.5 Estructuras selectivas

En los algoritmos, las estructuras selectivas o condicionadas son de la forma:

```
Si comparación verdadera
    Entonces
        Hacer acción 1
    Sino
        Hacer acción 2
Fin si
```

En un diagrama de flujo es, quizá, más fácil apreciar el proceso de

selección en una estructura condicionada (observa la **Figura 3.4**).

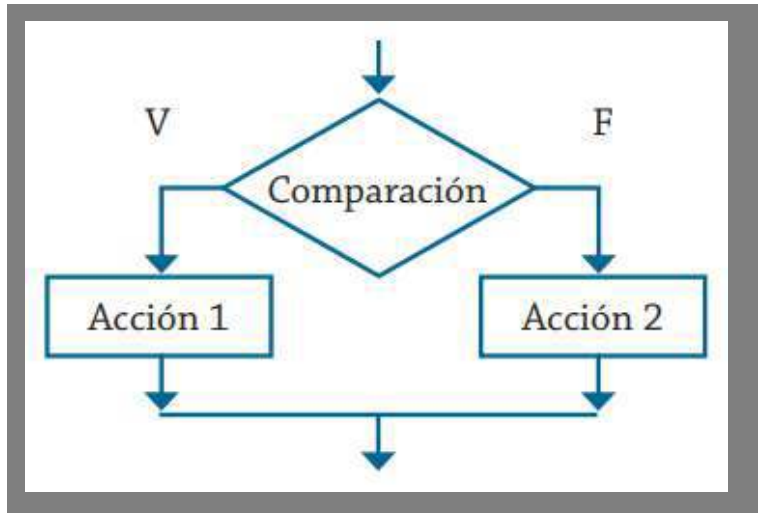


Figura 3.4. Forma de representar el algoritmo de una estructura selectiva (crédito: <https://www.uaa.mx/>).

Problema 2. Dado dos números **distintos**. Elabora un pseudocódigo para obtener el número mayor.

El pseudocódigo para este problema clásico, sería:

1. Inicio
2. Leer dos números A, B
3. Si $A > B$
Entonces
Mayor=A
Sino
Mayor=B
Fin si
4. Escribir número mayor
5. Fin

El diagrama de flujo, sería el siguiente:

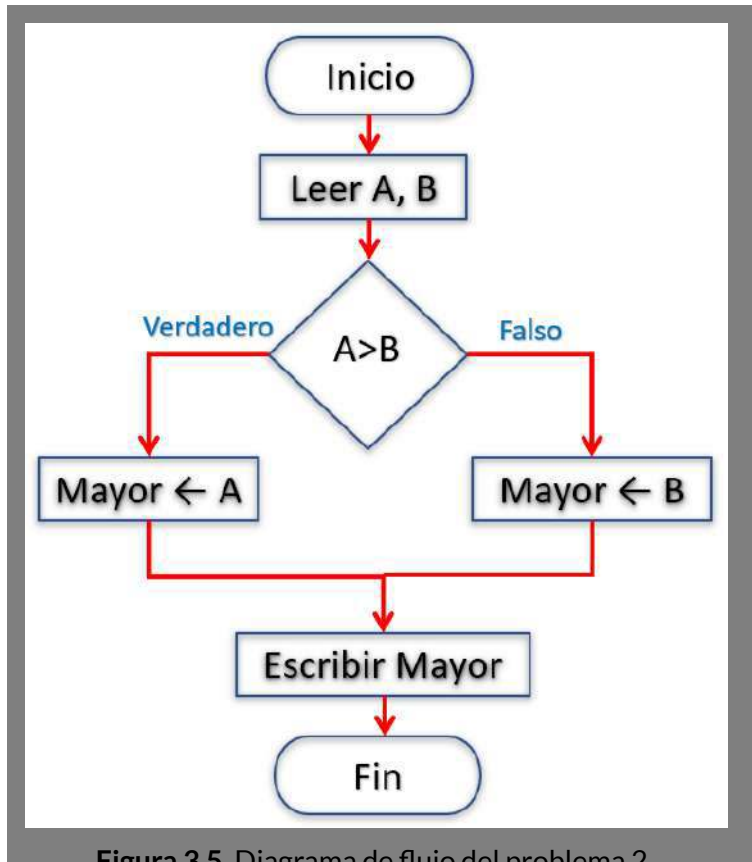


Figura 3.5. Diagrama de flujo del problema 2.

3.1.6 Estructuras iterativas

Una estructura iterativa o de repetición es aquella que permite que un proceso o conjunto de acciones se realice de manera cíclica (loop o bucle). Este proceso repetitivo se realiza **mientras** una condición lógica sea verdadera. Existen cuatro formas comunes de este tipo de estructuras: "Hacer mientras" (Do-While), "Mientras que", "Repite hasta que" y "Desde, hasta" (For-next).

En la **Figura 3.6** se presenta la forma Do-While en un diagrama de flujo.

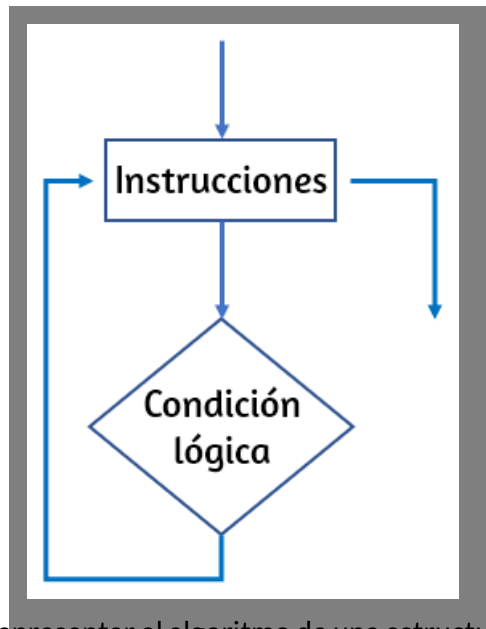


Figura 3.6. Forma de representar el algoritmo de una estructura iterativa Do-While.

Problema 3. Elabora un pseudocódigo para calcular la suma de los 100 primeros números naturales (no uses la fórmula directa).

El pseudocódigo para resolver este problema, es el siguiente:

1. Inicio
2. $c \leftarrow 0$
3. $S \leftarrow 0$
4. Hacer
 - $c \leftarrow c + 1$
 - $S \leftarrow S + c$
 - Mientras $c < 100$
- Fin Hacer
5. Escribir S
6. Fin

Hemos usado el caracter $\leftarrow$ para recordar la instrucción tipo asignación.

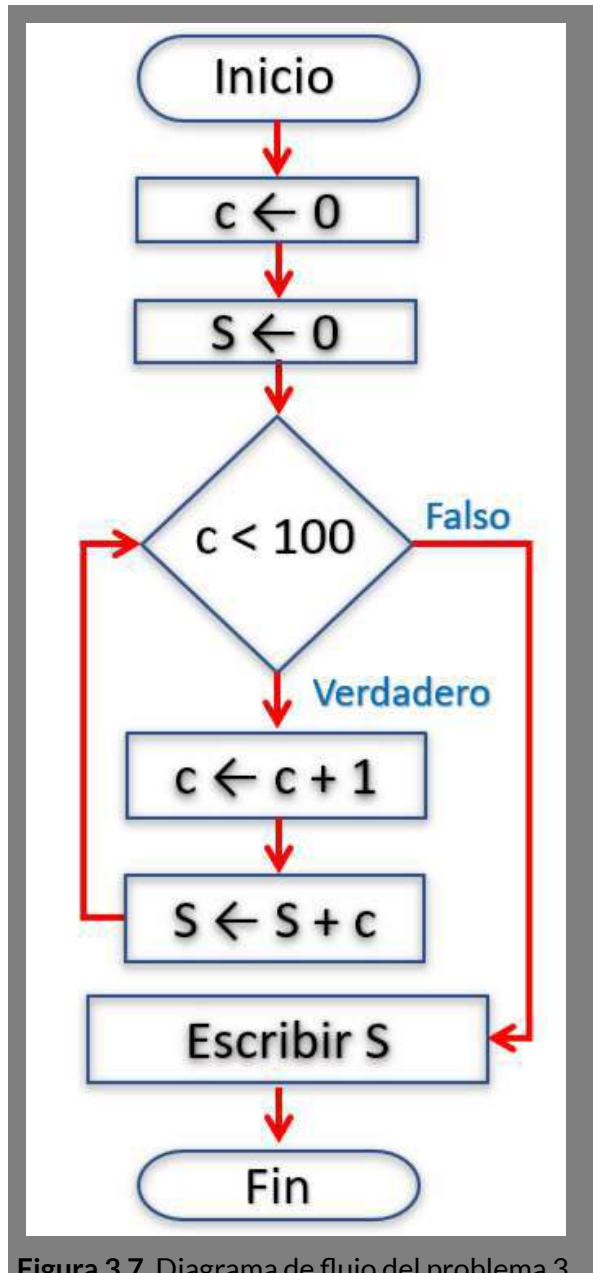


Figura 3.7. Diagrama de flujo del problema 3.

3.2 Computación en la Nube

¿Qué es una nube?

La nube (cloud en inglés) es la forma abreviada de "computación en la nube" (cloud computing), son todas las cosas a las que puedes acceder de forma remota a través de Internet. Cuando algo está en la nube, significa que está almacenado en servidores de Internet en lugar del disco duro de tu computadora.

¿Qué es la computación en la nube?

Según el NIST (*National Institute of Standards and Technology*), es un modelo para permitir el acceso de red ubicuo⁵, conveniente y bajo demanda a un grupo compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar y liberar rápidamente con un mínimo esfuerzo de administración o interacción con el proveedor de servicios ([NIST](#)).

⁵ Ubicuo: que está presente en todas partes al mismo tiempo

Estoy seguro que eres un usuario de la nube pues, al menos, tienes una cuenta de correo electrónico ¿dónde crees que se guardan tus mensajes?... ¡en la nube! Ahora, es posible que ya estés usando algún servicio de almacenamiento ¿identificas alguno en la siguiente nube⁶?



⁶ Animación interactiva diseñada por [Asmit](#).

El origen del término "computación en la nube" no es claro; sin embargo, fue popularizado en la primera década del siglo XXI a través de las primeras ofertas de servicios de computación remota como Salesforce y Amazon Web Services (aws). Estos servicios remotos se hicieron más populares con la incursión de grandes empresas como Apple, Microsoft, IBM, HP, Google y Rackspace, reforzando el concepto de la computación en la nube como proveedores de servicios de Internet a gran escala.



Interactivo 3.3. Servicios en la Nube
Créditos textos: Wikipedia.

3.2.1 Modelos de computación en la nube

Como pudiste observar en la infografía anterior, existen diferentes tipos de servicios, esto ocurre porque "la computación en la nube basa su arquitectura haciendo una separación entre hardware, plataforma y aplicaciones quedando las siguientes capas" [8]:

Software como Servicio. Simplificada como **SaaS** (*Software as a Service*). Está en la capa más alta (Ver [figura 3.4](#)), consiste en la entrega de aplicaciones completas (software) como un servicio, con una distribución del tipo "uno a muchos", por ejemplo el servicio de correo electrónico o el Office 365 de Microsoft.

Plataforma como Servicio. Simplificada como **PaaS** (*Platform as a Service*). Es un servicio de plataforma con todo lo necesario para dar soporte al desarrollo y puesta en marcha de aplicaciones y servicios web; es decir, es útil para desarrolladores para gestionar, distribuir y testear sus aplicaciones de software.

Ejemplos de este modelo son Google App Engine, SAP Cloud Platform, Heroku de Salesforce, Oracle Cloud Platform o AWS Lambda.

Infraestructura como Servicio. Simplificada como **IaaS** (*Infrastructure as a Service*). Es la capa más baja. "La idea básica es la de hacer uso externo de servidores para espacio en disco, base de datos, ruteadores, swiches así como tiempo de cómputo evitando un servidor local y toda su infraestructura" (Ibid), es útil para desplegar aplicaciones web, operar un CRM, realizar análisis de Big Data, backups, etc.

Ejemplos de IaaS son el Apache Cloudstack, Amazon Web Service y Google Cloud.

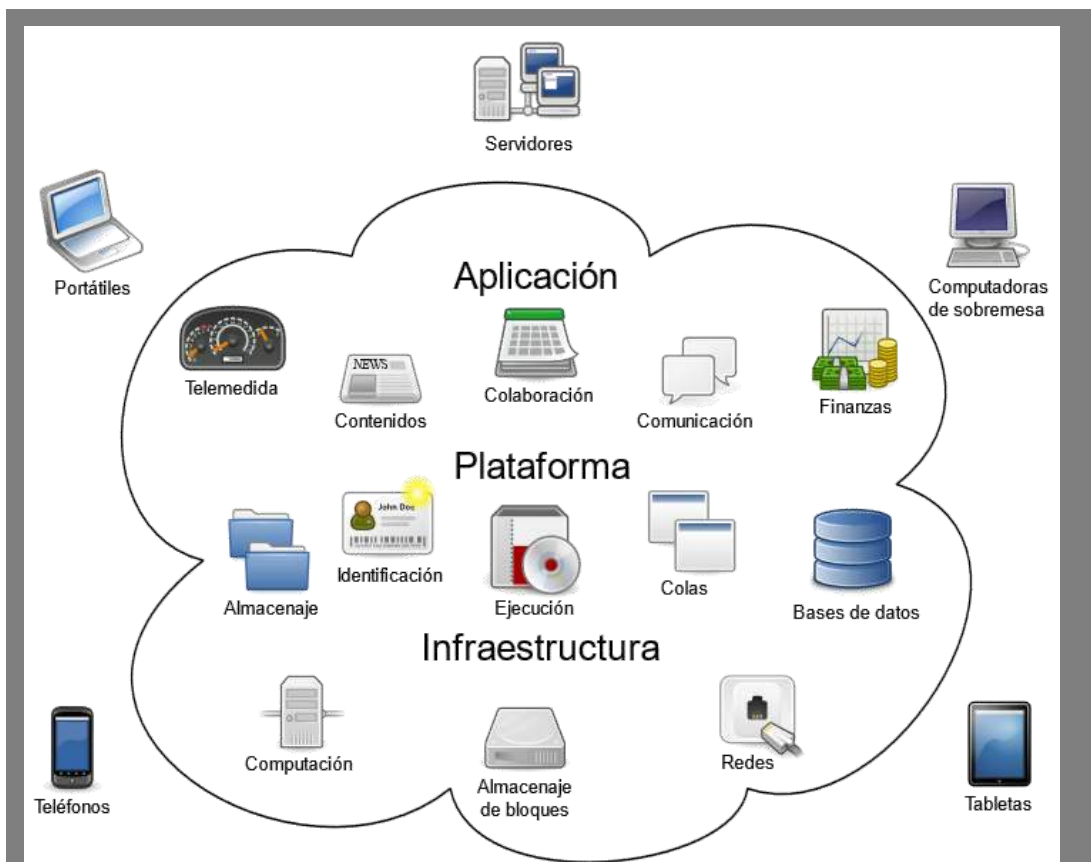


Figura 3.8. Computación en la nube (crédito: Imagen de [Sam Johnston](#), CC BY-SA 3.0).

3.2.2 Tipos de Nube

Según el modelo de implementación, se tienen los siguientes tipos de Nube:

Nube pública. Es aquella que puede ser utilizada por diferentes clientes que necesitan reunirse en servidores, y estos son propiedad y administrados por los proveedores, "El presente modelo de servicio es entendido como un servicio bajo demanda e inmediato, donde los usuarios acceden a él a través de Internet o de redes privadas virtuales (VPN)." Microsoft Azure y Google App Engine son ejemplos [\[9\]](#), [\[10\]](#).

Nube privada. Se fundamenta en la demanda de clientes individuales, proporcionando propiedad sobre los datos, su seguridad y está dedicada al cliente. Es seguro y costoso en comparación con una nube pública, "los recursos ofrecidos pasan a ser propiedad de una sola institución, manteniendo la infraestructura en una red privada de uso, y llegando incluso a ofrecer la posibilidad de alojar los servicios en las propias instalaciones del cliente". Eucalyptus System es el mejor ejemplo de una nube privada (Ibid).

Nube híbrida. Es la composición de los modelos anteriores. "Ambas nubes, tanto pública como privada, componen dos entidades independientes, pero mantienen una interrelación basada en principios de comunicación y portabilidad de aplicaciones y datos" [\[10\]](#). Amazon Web Services es un ejemplo de una nube híbrida.

Nube Comunitaria. Diseñada para una comunidad específica de consumidores de diferentes organizaciones. Puede ser de propiedad, administrado y operado por una o más empresas de la comunidad. Útil en el sector educativo. Facebook es un ejemplo de una nube comunitaria.

Edge computing. No se trata de un nuevo tipo de Nube, lo que permite es acercar (*edge*) "físicamente" la nube al usuario, permitiendo servicios más rápidos y confiables,

Edge computing permite que los datos producidos por los dispositivos de la internet de las cosas se procesen más cerca de donde se crearon en lugar de enviarlos a través de largas recorridos para que lleguen a centros de datos y nubes de computación ([Javier Pastor](#)).

3.2.3 Ventajas y desventajas de la Nube

Es obvio que son muchas las ventajas que ofrecen los servicios de la Nube, tales como:

-  Capacidad de procesar y almacenar grandes volúmenes de información (Big Data).
-  Alta disponibilidad, escalabilidad, flexibilidad, eficiencia. [\[10\]](#).
-  El ahorro, tanto en licencias como en la administración del servicio y en los equipos [\[10\]](#).
-  Implementación rápida y baja en riesgos.
-  Actualizaciones automáticas (Ibid.).
-  Ahorro energético.
-  Copias de seguridad.
-  Trabajo colaborativo. Ofrece la posibilidad de hacer un trabajo conjunto de empleados, estudiantes, investigadores, etc. que se encuentran en puntos geográficos distantes.

En cuanto a las desventajas, se podrían reducir a dos que son de gran impacto:

-  Fallos en la prestación del servicio. Esto puede ocurrir por fallas en los servidores, algo que ya ha ocurrido con los servicios de Google.
-  Amenaza de seguridad. "Tratamiento de la información fuera de los límites manejados por el usuario" [\[9\]](#). Situación que aún genera desconfianza en algunos usuarios, en especial por la pérdida de privacidad, pese a las nuevas tecnologías de ciberseguridad.

A continuación, responde 5 preguntas de selección múltiple. La evaluación es de tipo formativa, puedes realizarla varias veces.



Comprueba tus conocimientos en 5 preguntas

Los servicios remotos o en la Nube se hicieron más populares con la incursión de grandes empresas como:

A

 IBM y Google

B

 Google y Rackspace

C

 Apple y Microsoft

D

 Apple y Marlboro





Interactivo 3.4. Evaluación formativa - Cloud Computing.

Recuerda que puedes hacer clic en el botón de la esquina superior derecha, para realizar la evaluación en ventana completa.

3.3 Métricas y analítica en IoT

A diario se realiza el proceso de medir el entorno, al desplazarse de un lugar a otro se tiene en cuenta los metros o kilómetros recorridos, se mide el tiempo de desplazamiento o, en la realización de una actividad, la capacidad de cargar un objeto el cual es medido en kilos, incluso cuando se va al gimnasio, se mide el esfuerzo realizado, todo esto es medido en diferentes unidades, dependiendo del escenario y características en que se realice.

¿Qué es Medición? La palabra medición deriva del verbo medir que a su vez viene de la palabra latina *metriri*, que significa "comparar un resultado o cantidad con una unidad de medida previa" (significados.com).

De acuerdo al sitio de referencia, la medición sirve para determinar magnitudes de un objeto en relación a otro objeto que sirve de patrón, que es definido antes por un consenso. Hoy en día, estos modelos de comparación que se usan todos los días; por ejemplo, la masa, la temperatura y la longitud, están unificados en lo que se conoce como el **Sistema Internacional** de unidades (SI).

En este sistema, se establecieron las unidades de medición que son usadas para relacionar individual, social y económicamente. En este sentido, la medición es importante porque facilita el intercambio de tiempos, espacios, objetos y teorías.

De acuerdo con lo aprendido en los apartados anteriores, estudiados en este libro sobre IoT, una de las actividades esenciales del Internet de las Cosas, es realizar procesos apropiados de medición de los datos generados por los diferentes sensores que integran el



ecosistema, pasando por la transferencia de estos mediante la red y su presentación en las herramientas apropiadas, dando cuenta de su interpretación con la mejor fidelidad y confiabilidad posible.

En cada minuto que pasa, mientras estás estudiando este apartado, en el mundo se mueven miles de bytes, representados en datos, y estos a su vez se representan en diversos tipos de contenido que te ayudan a interpretar el mundo que te rodea: audios, videos, email, canciones, y muchas otras cosas que visualizas en el día a día.



Muchas personas aún creen que al hablar de datos, métricas y analítica, particularmente en el entorno de IoT, solo se trata de entorno tecnológico o de estadísticas, pero los datos confiables y oportunos ofrecen la ventaja de tener comprensión del mundo cambiante en el que se vive.

En la ciudad se acumula a diario una cantidad de datos, al acumularse suficientes de estos, es posible encontrar patrones de hábitos y comportamiento en la movilidad o el clima, pronósticos de sucesos en las vías o del transporte público, de averías, de fallas, etc, la correcta interpretación de estas permiten anticipar hechos que no se desean que sucedan y entrega mucha información aprovechable; para esto es necesario contar con herramientas de analítica.

Y ahora surge una gran pregunta, ¿qué utilidad tiene que aprenda a interpretar métricas y analíticas en los datos? Una de las oportunidades que se generan para los estudiantes, y futuros profesionales, es proyectarse como científico de datos. Una profesión en la que se debe aplicar habilidades técnicas que se adquieren en los programas afines a los estudios en la Cuarta Revolución Industrial, basadas en administración y análisis de datos para resolver problemas que se presenten en el entorno (crédito foto central: [The International System of Units](#)).

3.3.1 Historia de los científicos de datos

El científico informático, William S. Cleveland, escribió en 2001 un artículo denominado "ciencia de datos: un plan de acción para ampliar el área técnica de la estadística". Este artículo presentó la ciencia de datos como una disciplina de los estadísticos aplicados. Eso fue hace solo un par de décadas, y el mundo de la tecnología y las empresas se han movido rápidamente desde entonces. Como la mayoría de las carreras STEAM (por sus siglas en inglés Science,

Programación

Lenguajes:

- Python,R,Jara
- Bases de datos
- Procesos Cloud

Informática y computación

- Álgebra
- Traducción de algoritmos
- Paquetes de estadística

Conocimientos estadísticos

- Habilidad matemática
- Automatización
- Machine Learning
- Deep Learning e IA

Technology, Engineering, Arts and Maths), los científicos de datos son profesionales muy solicitados.

En 2012, los tecnólogos Thomas H. Davenport y DJ Patil declararon "Científico de datos: el trabajo más sexy del siglo XXI" [\[11\]](#), un eslogan que fue retomado incluso por periódicos de las grandes ciudades como el New York Times. Una década más tarde, lo reafirmaron, afirmando que "el trabajo tiene más demanda que nunca entre los empleadores" ([Harvard HandWiki](#)).



3.3.2 Herramientas para analítica de datos

Las métricas y la analítica en el IoT requieren un buen manejo de la información, y ante todo entender cuál es el propósito y el contexto de lo que está realizando y analizando:

 **La bodega de datos.** Es el espacio donde se almacenan los datos, estos deben guardarse de modo ordenado y lógico, deben estar dispuestos para un proceso de consulta para que sean ubicados ágilmente.

 **Los dashboards o tableros.** Los datos se representan de manera gráfica, visualizando los datos en la bodega. Es una forma de contar la historia de la información de manera más atractiva y asertiva.

**Bodegas
de datos**



Al

Dashboards



 **Las alertas.** Se activan cuando se encuentran anomalías o si se programó en el proyecto IoT anuncios, esto sirve para clasificar los datos y detonarlas cuando se cumplen criterios específicos, ya sea de manera preventiva o informativa.

 **KPIs.** Son los indicadores de desempeño. Miden el rendimiento y/o avance de los procesos IoT, para ello es importante tener unos buenos datos, además permiten mejora continua y optimización de los procesos.

El banner inferior, fue tomado del curso "Internet de las Cosas - IoT" de la plataforma @Medellín e impartido por el Colegio Mayor de Antioquia.



3.4 Un proyecto inteligente

Steve Jobs, uno de los fundadores de la empresa Apple, decía frecuentemente: "todos en el país deberían aprender a programar un computador (...) porque eso te enseña a pensar".

Pero, ¿qué significa esa expresión? ¿Qué implica programar un computador?

Diseñar o construir un programa no solo significa integrar código con características específicas y quedar a la espera que siga una serie de instrucciones. Indiferente al idioma que utilices, siempre es útil y práctico conocer la forma de comunicarte con el componente tecnológico de tu ecosistema en el que habitas.

Unos apartados atrás, se habló sobre Arduino y Raspberry Pi, dos placas divertidas para construir proyectos de domótica, automatización y otros procesos electrónicos, y ha crecido tu curiosidad respecto a qué puedes construir con todo lo aprendido hasta ahora. En la medida que se vaya desarrollando este proyecto encontrarás la aplicación de las habilidades STEAM y todo lo que éstas permiten mejorar el entorno y el ecosistema de ciudad que habitas actualmente.

Imagen derecha generada por la IA Designer de
Microsoft





3.4.1 Idea del proyecto inteligente

La comunidad Arduino ha sido una de las precursoras de la comunidad Maker internacional, gracias a estas mágicas placas electrónicas y su software de código abierto, se le facilita a cualquier persona crear proyectos, compartirlos, transformarlos y aportar a la construcción de una ciudad inteligente e interconectada; por ello, nuestra misión es el pensar y diseñar soluciones para aportar positiva y constructivamente al desarrollo del entorno.

Como idea de referencia, vamos a construir una planta inteligente, integrando sensores y analizando cuáles son las necesidades de nuestra pequeña maceta inteligente, integrando bases de diseño de videojuegos para simular sensaciones y simular un entorno de comunicación con la planta.

3.4.2 Etapas del proyecto

Desarrollaremos este proyecto en tres etapas, durante las cuales aprenderemos generar emociones en la planta y cuidarla como si fuera un videojuego.

Etapas 1. Diseño del algoritmo de la planta inteligente





Etapla 2. Diseño del circuito electrónico y su conexiones.

Etapla 3. Implementación de la planta inteligente.

3.4.3 Etapla UNO del proyecto inteligente

Esta primera etapa consiste en definir las instrucciones, condiciones y tareas que se deben considerar en la planta o maceta inteligente, teniendo en cuenta que cada planta es diferente y requiere cuidados y condiciones específicas para vivir, estar saludable y, si fuera el caso, producir frutos.

¿Qué se necesita para diseñar el primer algoritmo?

Los algoritmos son un elemento fundamental en los procesos de programación de los sistemas de cómputo o también denominados informáticos. Para que puedan ejecutarse las tareas se necesita diseñar, inicialmente, el algoritmo. Esto es, indicar los pasos para enrutar los datos de entrada, que se generan desde los sensores, en datos de salida en una pantalla u otros dispositivos de visualización de información. Una vez sea

diseñado y probado, se llevará a las herramientas respectivas para transformarlo en instrucciones o indicaciones de computador, o lo que se le conoce como **Programa**. Este último está compuesto por el algoritmo, la especificación de los datos y las instrucciones que permiten la comunicación entre los usuarios del programa y el sistema informático.

Diseñar algoritmos requiere realizar una investigación previa de lo que desea construir o inventar, pues su propósito deberá obedecer a una solución, preferiblemente innovadora. Este proceso requiere que desarrollemos habilidades de investigador o, en este caso, de científico de datos, tal como lo aprendimos en el apartado anterior.

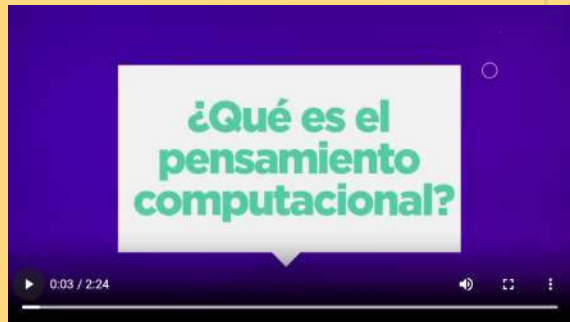
Otra habilidad importante y necesaria para diseñar el algoritmo se denomina **Pensamiento Computacional**. Hay muchas definiciones del término "pensamiento computacional", una de las más conocidas es: [proceso mental utilizado para formular problemas y sus soluciones, de forma que las soluciones se representan en una forma que puede ser llevada a cabo por un agente de proceso de información \[12\]](#)





Te invito a apreciar el siguiente video, para comprender mejor el concepto y su utilidad en el algoritmo a construir:

Video



Video 3.1. ¿Qué es el pensamiento computacional?
(crédito: [Educar Portal](#) , YouTube con Licencia Atribución de Creative Commons).

Para hacer más divertido el proyecto, lo fundamentaremos en un concepto de **personaje digital** (o en algunos casos denominado **mascota virtual**), que fue muy popular a finales de los años 90, uno de ellos fué el **tamagotchi**, un aparato electrónico con una pantalla en blanco y negro pixelada, donde se podía ver una mascota virtual: "Una mascota virtual, también conocida como mascota digital, es un tipo de compañero para humanos, de carácter artificial. Su uso principal es el de compañía y entretenimiento, sustituyendo en algunos casos a las mascotas reales ([Educar Portal](#)).

3.4.4 Sensaciones para la planta inteligente

Primero se definen las sensaciones que deseamos que represente la planta inteligente, teniendo como base los sensores que vamos a instalar inicialmente; estos son: temperatura, humedad, luz... pero, ¿Ya sabes cuál es la cantidad adecuada de cada uno de estos elementos para la planta?

En la definición del algoritmo para la planta, debemos investigar estos factores clave, pues sin ellos el proyecto podría fracasar, ya sea porque se muera por falta de agua o exceso de esta, se congele, se seque por demasiada temperatura, o pueda morir por otros factores.

¡Muy bien! Ahora a diseñar una tabla con estos valores y trataremos de representar o simular algunas sensaciones en la planta, nos ayudaremos con un formato de ejemplo, pero es tu tarea diseñar y representar tu propia tabla de sensaciones.

CONDICIONES				
	TEMPERATURA	HUMEDAD	LUZ	PUNTOS DE VIDA
✓ Ideal (todos los valores son óptimos)	120°C a 25°C adecuada 10	65% a 75% adecuada 10	600 lux a 800 lux adecuada 10	30
✓ Uno de los factores disminuye - Humedad	10	< 65% y > 40% sed 5	10	25
✓ Uno de los factores aumenta - Humedad	10	> 65% ahogo 15	10	35
✓ Uno de los factores disminuye - Temperatura	< 20°C y > 15°C frío 5	10	10	25
✓ ¿Te animas a crear otras condiciones?				

Figura 3.9. Tabla de sensaciones.

Haz clic en la imagen, para verla más grande.

SENSACIONES



Vamos a diseñar las sensaciones de la planta basada en la filosofía de juego virtual del tamagotchi, para ello hay que asignar puntos a cada sensor con límite de 10, es decir que 10 es el valor óptimo, y al sumarlos se obtiene el índice de sensación de la planta, como se puede ver en el ejemplo, la Simpatía tiene un total de 30 puntos.

¡Excelente trabajo! ya se ha construido la base para las sensaciones de la planta, pero puedes construir otras, esa es la siguiente tarea para este apartado.

Algunos ejemplos de sensaciones para inspirarte:

- 😊 simpatía. Las 3 variables estables (30 puntos)
- 😞 exploración. Una de las variables cambia y la planta busca ajustes (25 puntos)
- 😱 sufrimiento. Dos de las variables cambiaron y la inestabilidad de los estímulos ponen en peligro la salud de la planta (entre 15 a 25 puntos ó más de 30 puntos)
- 😌 relajación. Se estabilizan dos variables y se alivia la fatiga de la planta al reducir los estímulos negativos (28 puntos)

Ahora veremos cómo traducir esta información en un algoritmo, planteándose en estructura de pseudocódigo. Recuerda que el pseudocódigo es una manera de definir los algoritmos, es decir, los pasos necesarios para resolver un problema. El pseudocódigo es un paso intermedio entre los diagramas de flujo, que se expresan mediante símbolos, y los lenguajes de programación, que están ligados a una sintaxis bien definida. El principal objetivo del pseudocódigo, es representar la solución a un algoritmo de la forma más detallada posible, y a su vez lo más parecida posible al lenguaje que posteriormente se utilizara para la codificación del mismo.

De este modo, el pseudocódigo contendrá sentencias con estructuras similares a las que se encuentran en los lenguajes de programación, con condicionales, bucles, etc., pero escritas en un lenguaje de fácil lectura para los humanos, sin las restricciones que impone un lenguaje de programación real (<https://desarrolloweb.com/>).

3.4.5 Construyendo el algoritmo

A continuación, presentamos dos ejemplos de algoritmos en versión de diagrama de flujo de la temperatura y la luz, según los rangos ideales diseñados en la tabla de datos de la planta.

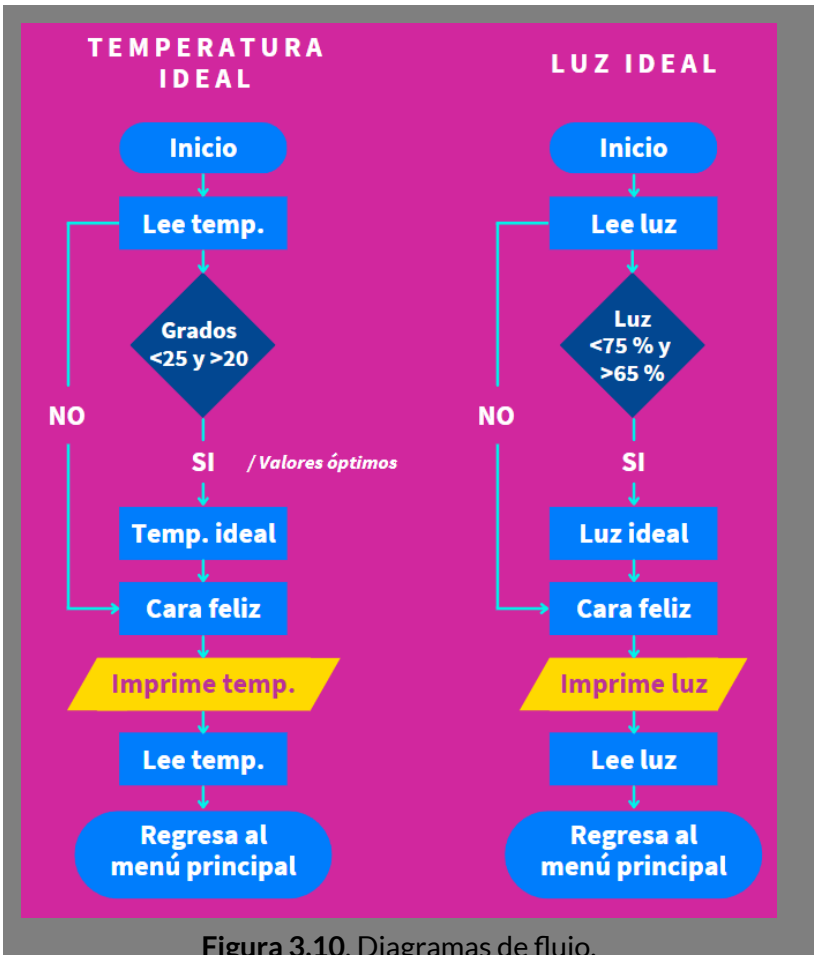


Figura 3.10. Diagramas de flujo.

El **diagrama de flujo** del algoritmo de temperatura inicia tomando la lectura en grados Celsius (nombrado en el algoritmo sólo como grados). Pasa a la primera condición **si la temperatura está entre 20 y 25**, se define como temperatura ideal e imprimirá en pantalla una carita feliz indicando que la planta está en la sensación de felicidad, valida de nuevo el proceso y pasa a la siguiente condición, luego regresa al programa principal para repetir el ciclo.

De esta manera se puede construir la secuencia de instrucciones, una a una de cada sensor, indicando los pasos a seguir para capturar los datos, analizarlos y tomar diferentes decisiones, que están condicionadas por las posibilidades que se generó en la tabla de datos y las sensaciones que se asignen correspondientemente.

Algo muy importante que debes considerar al momento de diseñar el algoritmo de cada sensación: "cada línea de la tabla es necesario evaluarla de manera integral, para ello se enlazan con la palabra de conexión, a este elemento se le conoce como operador lógico".

Los operadores lógicos son usados para concatenar dos o más expresiones con operadores relacionales, se usan comúnmente las letras *Y* y *O*, o las palabras en inglés **AND** y **OR**.

Para que la planta tenga la sensación de felicidad, debe tener al mismo tiempo la humedad en el rango correcto, la temperatura ideal y la luz apropiada, todas a la vez dentro del mismo rango óptimo. Para este caso entonces se define de esta manera:

SI (la temperatura está en un rango mayor o igual a 20°C Y temperatura menor o igual a 25°C) **Y** (la humedad está en un rango mayor o igual a 65% Y humedad menor o igual a 75%) **Y** (la luz está en un rango mayor o igual a 600 Y luz menor o igual a 800) **ENTONCES** la sensación de la planta es **FELIZ**.

En este estado de FELIZ puede integrar varias instrucciones; por ejemplo, encender una luz de color verde, presentar en una pantalla un texto, imagen o emoticón que represente una cara feliz o, incluso, activar un sonido, ¡todo es posible!



Figura 3.11. Planta feliz (Diseñada por la IA Ideogram).

Nota: debes tener presente que al representar porcentajes también se pueden presentar con un número decimal; es decir, 20% sería 0,20 o 25% sería 0,25.

Si se diseña un estado diferente, por ejemplo de **SED**, en las instrucciones correspondientes podría indicar que se encienda una luz azul, activar un pequeño motor que adicione agua a la planta, presentar una cara de disgusto en la pantalla y, si la sed fuera mucha en la planta, avisar que la planta está en riesgo mediante un sonido o mediante un mensaje de texto.

Ya has aprendido a diseñar el primer algoritmo, ahora a practicarlo, para esto diligencia la ficha de la actividad y construye las propias sensaciones adaptadas a las condiciones de la planta, intenta construir también el diagrama de flujo de cada una... y ¡a divertirse!

RAYA VERDE
ALBERTO SASTRIV  INDIOVANDER

CUESTIONARIO 10T

Comprueba tus conocimientos en 15 preguntas

Responde con la mejor opción



Comenzar





Capítulo 4

Aplicaciones IoT

En este capítulo conocerás los fundamentos de las tecnologías de programación IoT, la computación en la nube y las oportunidades que estas representan para el desarrollo de soluciones IoT. En ellas se apropiarán diferentes términos o, en muchos casos, se entenderá la importancia de algunos con los que ya has interactuado.

A continuación, podrás encontrar algunos ejemplos y aplicaciones del IoT. Cómo programar el Sketch y cómo configurar las secciones Setup y Loop.

4.1 Hablando con la planta inteligente

Entramos en la **segunda etapa** de diseño de nuestra planta inteligente, que consiste en convertir el algoritmo, diseñado previamente, en instrucciones de código para que la placa de arduino que hace las veces de cerebro del proyecto interprete las instrucciones y represente las sensaciones que se ha diseñado para la planta inteligente.

Pero, ¿Cómo podemos hablar con la planta, para nos manifieste sus sensaciones?, ¿Qué lenguaje debemos usar?

4.1.1 Lenguaje C

El lenguaje en el que se programa arduino se denomina **lenguaje C**, el cual es un lenguaje de programación (considerado como uno de los más importantes en la actualidad) con el cual se desarrollan tanto aplicaciones, como sistemas operativos y, a su vez, forma la base de otros lenguajes más actuales como Java, C++ o C#.

En 1972, nace el lenguaje de programación C de la mano del científico informático estadounidense Dennis Ritchie (con la ayuda de Ken Thompson). Recibió el nombre de C simplemente por una cuestión alfabética, ya que sustituyó a sus lenguajes predecesores B

y BCPL que codificaban la minicomputadora PDP-7. El desarrollo del lenguaje C toma protagonismo para la creación del sistema operativo UNIX que llevarían consigo los nuevos minicomputadores PDP-11, primeros en conectar el procesador, los periféricos y la memoria entre sí mediante un BUS bidireccional.

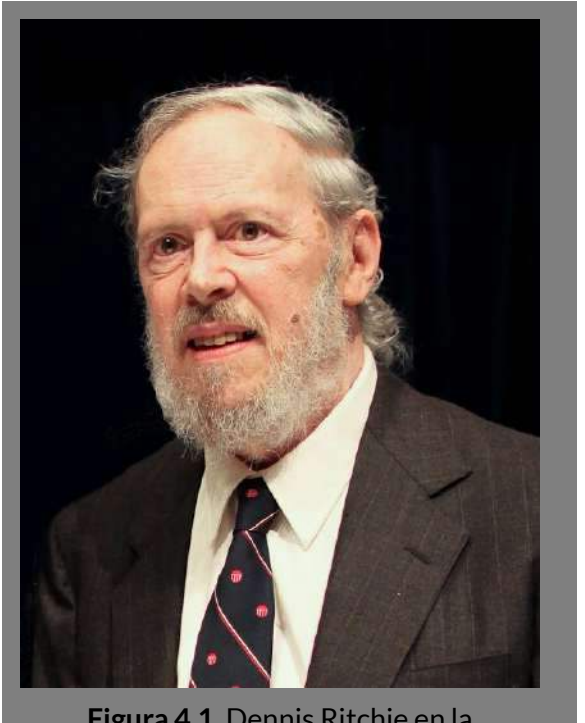


Figura 4.1. Dennis Ritchie en la ceremonia del llamado Premio Japón, en mayo de 2011 ([Wikipedia](#), CC BY 2.0).

Son varias las características de C como se mencionan a continuación:

Estructura de C - lenguaje estructurado: Programación de nivel medio (beneficiándose de las ventajas de la programación de alto y bajo nivel). No depende del hardware, por lo que se puede migrar a otros sistemas. No es un lenguaje para una tarea específica, pudiendo programar tanto un sistema operativo, una hoja de cálculo o un juego.

Adicionalmente, ofrece un control absoluto de todo lo que sucede en el computador. Se tiene organización del trabajo con total libertad. Los programas son producidos de forma rápida y son bastante potentes. Rico en tipo de datos, operadores y variables en C.

Como inconvenientes, siendo también características del lenguaje C, no es un lenguaje sencillo de aprender, requiere una continua práctica y un serio seguimiento si se desea tener el control de los programas, pero... ¡Las **inteligencias artificiales** (IA), te pueden dar una mano, como veremos más adelante!

Sintaxis en C: Todos los programas que son creados tienen diferentes tipos de formato. Este formato es conocido como la sintaxis, una serie de reglas y procesos que lideran la estructura de un programa. Estas reglas tienen que ser entendidas por el compilador para que se pueda crear un programa en C.

La sintaxis básica en C determina la forma en que se agrupan los caracteres para formar tokens, que son la unidad mínima de programación en C. El ejemplo tradicional de "Hola Mundo", usado para representar muchos lenguajes de programación, sería así:

```
#include <stdio.h>
int
main()
{
    printf("Hola Mundo"); // línea sencilla de comentarios
    return 0;
    /*
    comentarios con observaciones
    más amplias
    */
}
```

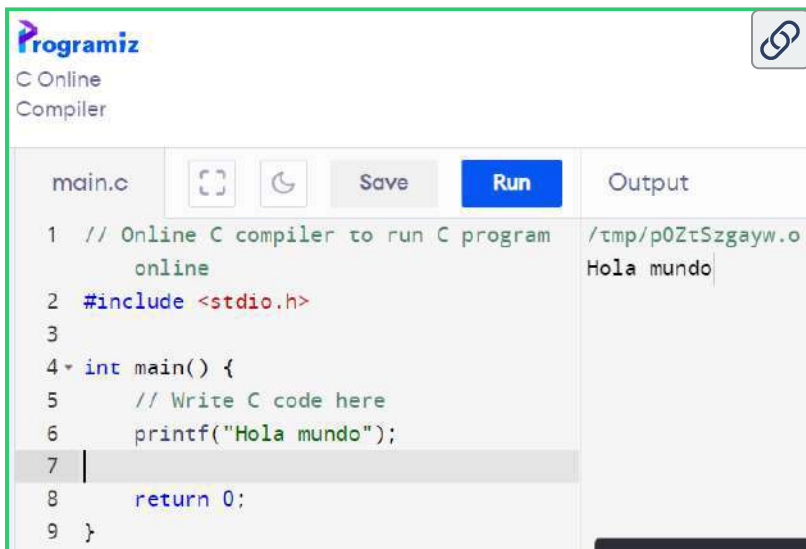
¿Cómo ejecutar este programa?

En la red puedes encontrar varios **compiladores**, para practicar la sintaxis anterior. Haz clic en una de las siguientes imágenes y, luego, en el botón RUN para compilar y ejecutar el programa.



The screenshot shows a web-based C compiler interface. At the top, there is a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a link icon. The main area is a code editor with a dark background, showing a C program named 'main.c'. The code includes a header for stdio.h and a main function that prints 'Hello World' and returns 0. The code is as follows:

```
1- /******  
2-  
3- Online C Compiler.  
4- Code, Compile, Run and Debug C program online.  
5- Write your code in this editor and press "Run" button to compile  
6-  
7- *****/  
8-  
9- #include <stdio.h>  
10-  
11- int main()  
12- {  
13-     printf("Hello World");  
14-     return 0;  
15- }  
16-  
17-
```



The screenshot shows the Programiz online C compiler interface. It has a toolbar with buttons for Run, Save, and a link icon. The code editor shows a C program named 'main.c' that prints 'Hola mundo' and returns 0. The code is as follows:

```
1 // Online C compiler to run C program online  
2 #include <stdio.h>  
3  
4 int main() {  
5     // Write C code here  
6     printf("Hola mundo");  
7  
8     return 0;  
9 }
```

On the right side, there is an 'Output' panel showing the result of the compilation: 'Hola mundo'.

Obviamente, en este libro no te vas a convertir en un experto programador en lenguaje C, pero si vas a usar algunas estructuras mínimas de programación, para nuestra planta inteligente.

Una opción es recurrir a una IA; por ejemplo, a [ChatGPT](#) le pedimos "Escribe un programa en lenguaje C, que pida la edad y responda con la palabra "Joven", si la edad es menor de 30 o con la frase "pasadito de años", si la edad es mayor o igual a 30", este es el código que nos entregó:

```
#include <stdio.h>

int main() {
    int edad;

    // Pide al usuario su edad
    printf("Por favor, introduce tu edad: ");
    scanf("%d", &edad);

    // Comprueba si la edad es menor de 30
    if (edad < 30) {
        printf("Joven\n");
    } else {
        // Si la edad es 30 o mayor
        printf("Pasadito de años\n");
    }

    return 0;
}
```

Figura 4.2. Código en lenguaje C, generado por ChatGPT.

Puedes hacer el ejercicio con una IA como [ChatGPT](#), [Gemini](#), [Claude 3](#) o [Copilot](#) de Microsoft. En cualquiera de ellos, obtendrás el mismo código. Copia ese código en uno de los compiladores anteriores y verifica su funcionamiento.

4.1.2 Estructuras de programación de Arduino

En Arduino, los programas están distribuidos en bucles (loops), esto es, se ejecutan de forma infinita mientras están activados. Pero, antes de ejecutar estos bucles, hay que configurar los parámetros iniciales.

Sección void setup(). Necesaria para realizar las configuraciones en la placa; es decir, establecer cuales serán los pines de entrada o salida, configurar parámetros como velocidad o frecuencias, y todos los demás acuerdos necesarios para establecer las compatibilidades con la placa.

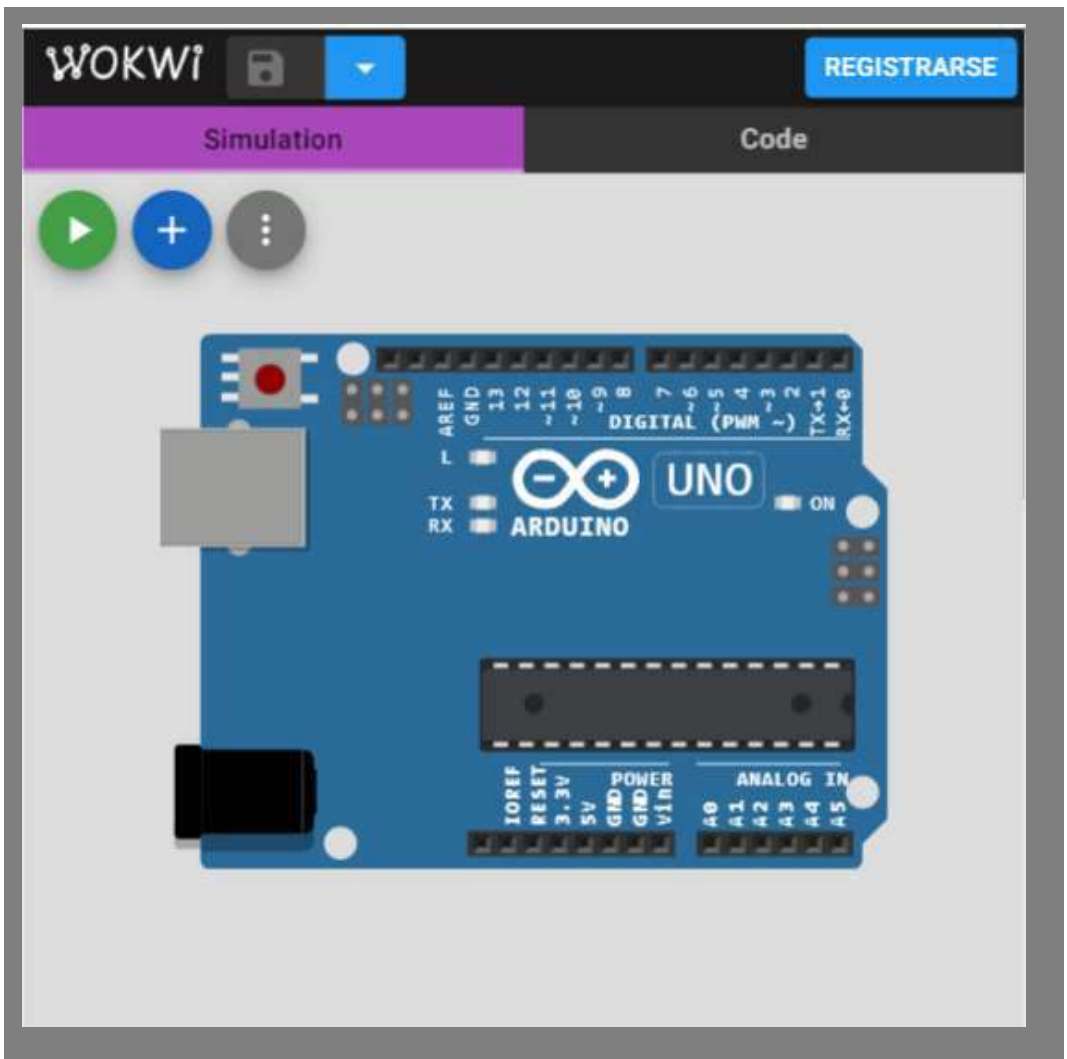
Sección void Loop(). Este es el bucle, tal como su significado en inglés lo sugiere. Se ejecuta de forma infinita y es el espacio donde el código de medición de sensores, decisiones, entre otras tareas, también donde la inteligencia de la planta será ejecutada.

Finalmente, el entorno denominado el IDE (Entorno de Desarrollo Integrado o en inglés *Integrated Development Environment*), el cual consiste en una aplicación que va a permitir escribir el programa, es como un editor de textos pero que tiene la capacidad de verificar si los comandos e instrucciones están correctos, además hace conexión con la placa física de arduino y sube esta información en lenguaje de máquina, para que sean ejecutados de acuerdo a lo que se ha definido, esto quiere decir que la planta inteligente tendrá realmente las instrucciones inteligentes reflejadas en dispositivos electrónicos y acciones físicas correspondientemente. Este entorno lo describimos más adelante; por ahora, usaremos un entorno de simulación.

4.1.3 Entorno de simulación Wokwi

Wokwi es un simulador de proyectos Arduino, que permite simular visualmente componentes, cables, sensores y, también, ejecutar el código y mostrar qué es lo que hace.

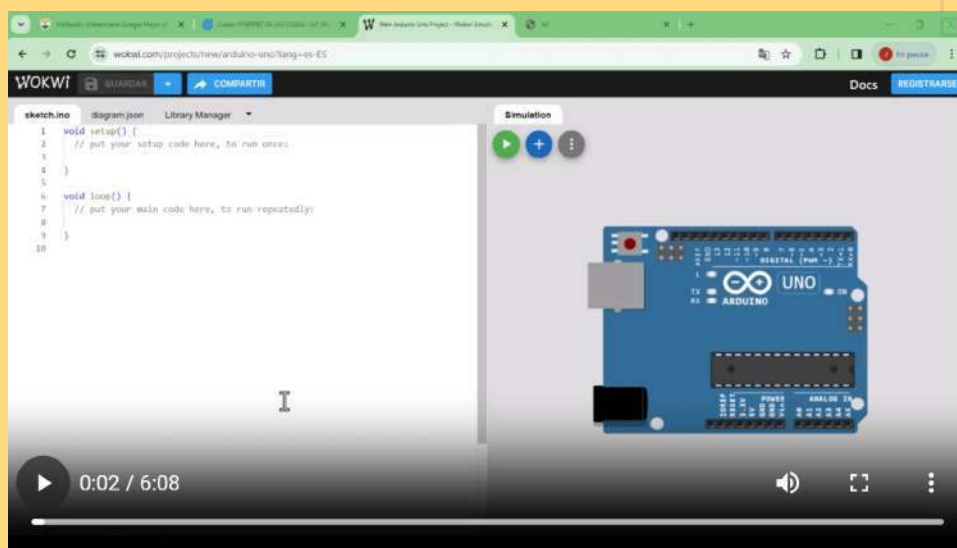
Es usar Arduino pero sin tener la placa (haz clic en la siguiente imagen).



Interactivo 4.1. Entorno Wokwi.

En la página anterior, presentamos el entorno Wokwi, pero puedes acceder a él desde la barra de herramientas de este libro. Una descripción de esta herramienta, la puedes ver en el siguiente video:

Video



Video 4.1. Interactuando con Wokwi.

Como lo dice el video, lo primero que debes hacer es ingresar al sitio web de Wokwi y crear una cuenta de usuario desde el siguiente enlace <https://wokwi.com>, dar clic en la opción **entrar**.

4.1.3.1 Primer proyecto con Wokwi - Luz parpadeante

Como lo dijimos antes, es posible recurrir a una IA para obtener el código de un proyecto Arduino; por ejemplo, le pedimos Gemini "el código en C++ para una placa Arduino, en la que un led se prende y apaga por un segundo, en un loop infinito". Este es el código que nos entregó:

```
// Definimos el pin del LED
const int ledPin = 13;

void setup() {
  // Configuramos el pin del LED como salida
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Encendemos el LED
  digitalWrite(ledPin, HIGH);

  // Esperamos un segundo
  delay(1000);

  // Apagamos el LED
  digitalWrite(ledPin, LOW);

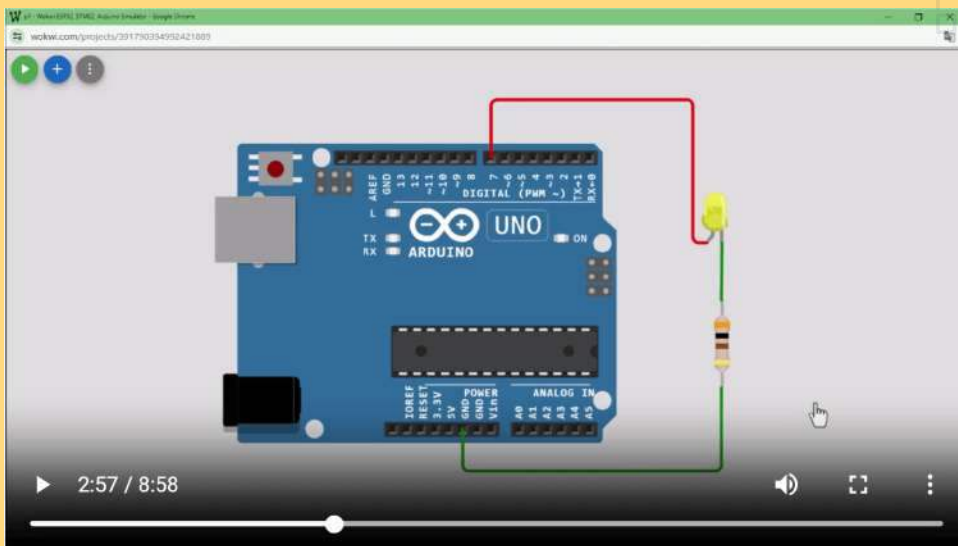
  // Esperamos un segundo
  delay(1000);
}
```

Figura 4.3. Código en lenguaje C, generado por Gemini.

Observa que, inicialmente, define una **constante** con el nombre **ledPin** y valor 13. En la sección **setup void()** se configura un **led** de salida (ouput). Finalmente, se dan cuatro instrucciones en la sección **void loop**, que explicamos en el video 4.2.

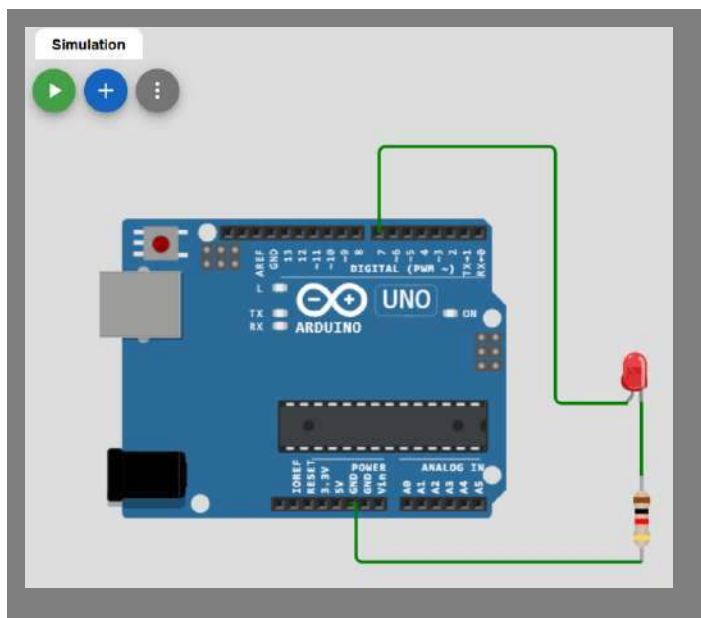
En el video 4.2, hemos usado un código igual al dado por Gemini, cambiando el nombre de la constante y eligiendo el pin 7. Presta atención al video y construye el proyecto siguiendo el procedimiento realizado.

Video



Video 4.2. Proyecto luz intermitente.

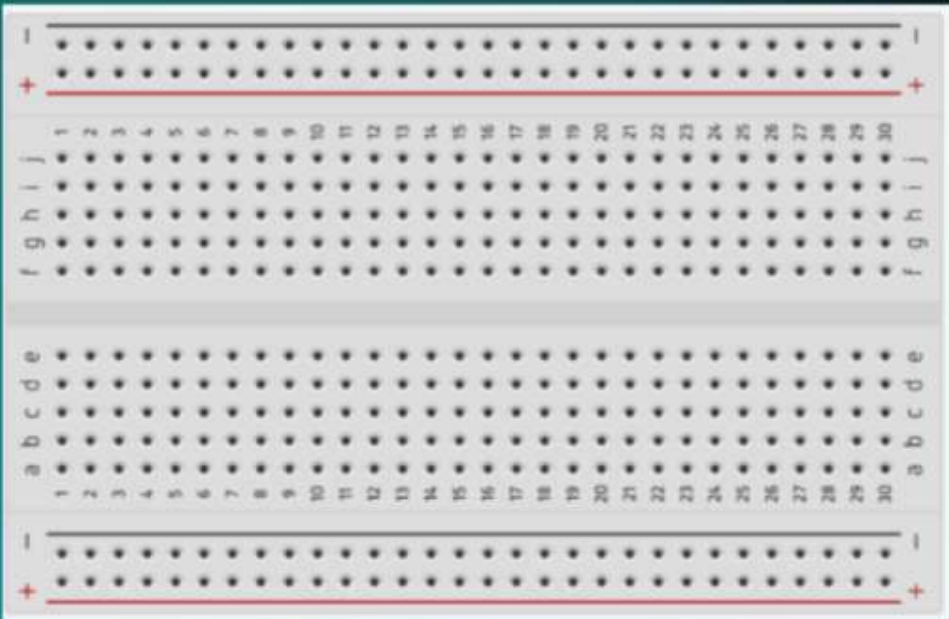
Haz clic en la siguiente imagen, para abrir este primer proyecto:



4.1.3.2 Segundo proyecto con Wokwi - uso del protoboard

Inicialmente, veamos que es un protoboard.

El protoboard



Un protoboard, también conocido como una placa de prototipado, es una placa perforada con una cuadrícula de agujeros, que se utiliza para montar componentes electrónicos y realizar conexiones entre ellos.

La IA Pi.ai nos da una descripción más amplia en el siguiente audio

▶ 0:00 / 0:47

🔊

⋮

Siguiente

Interactivo 4.2. El protoboard.

Es muy importante que hayas entendido cómo funcionan las conexiones en un protoboard, para evitar mal funcionamiento del proyecto o, peor aún, originar cortos en el circuito. En el siguiente video, ampliamos un poco más las conexiones al protoboard:

Video



Video 4.3. Manejo de placas Protoboard (video de [Tecnología Interactiva](#)  en YouTube, Licencia Atribución de Creative Commons).

Ahora, vamos a diseñar un proyecto que incluya una luz LED y un pulsador. Este último, hará que la luz se encienda cuando lo tengamos presionado. Le pediremos a ChatGPT el código en lenguaje C.

Para ello, hemos usado la siguiente indicación: "Escribe el código en lenguaje C para un proyecto Arduino en el que un pulsador enciende una luz LED"

El código generado por ChatGPT, es el siguiente:

```
// Definimos los pines que vamos a utilizar
const int ledPin = 8; //El pin donde conectamos el LED
const int buttonPin = 2; //Pin donde conectamos el pulsador

// Variable para almacenar el estado del botón
int buttonState = 0;

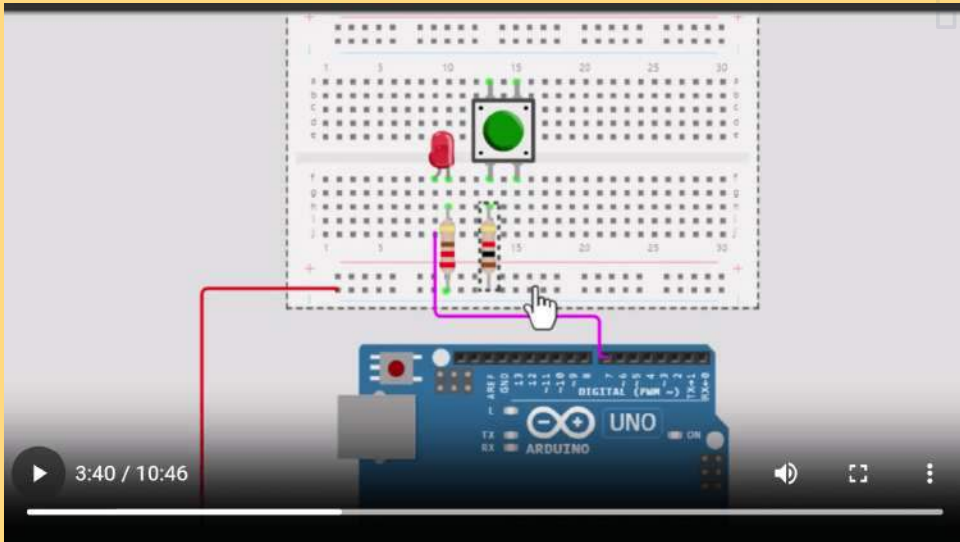
void setup() {
  // Configuramos el pin del LED como salida
  pinMode(ledPin, OUTPUT);
  // Configuramos el pin del botón como entrada
  pinMode(buttonPin, INPUT);
}

void loop() {
  // Leemos el estado del botón
  buttonState = digitalRead(buttonPin);

  // Si el botón está presionado, encendemos el LED.
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    // Si el botón no está presionado, apagamos el LED.
    digitalWrite(ledPin, LOW);
  }
}
```

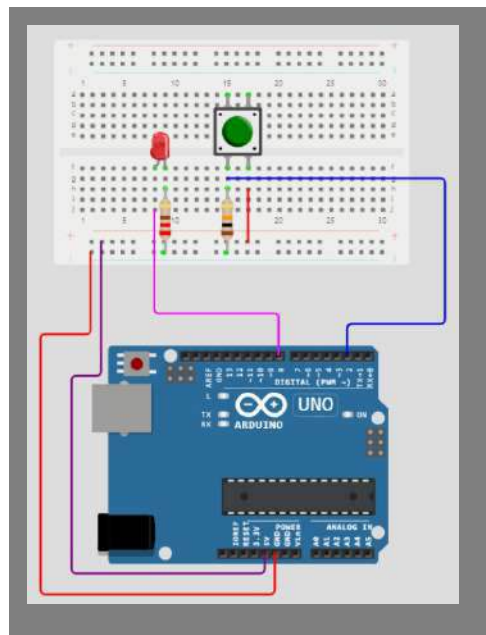
En el siguiente video, podrás ver el paso a paso, para el montaje del proyecto.

Video



Video 4.4. Proyecto luz intermitente.

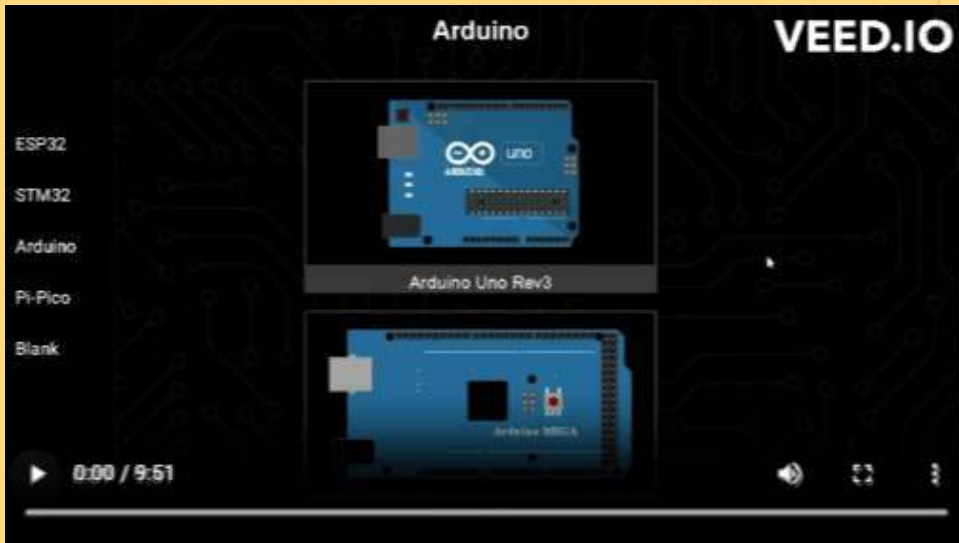
Haz clic en la siguiente imagen, para abrir este segundo proyecto:



4.1.3.3 Tercer proyecto con Wokwi - uso de sensores

En este proyecto usaremos un sensor de temperatura y humedad. En este proyecto, escribiremos el código paso a paso, por lo que te recomendamos prestar bastante atención al siguiente video:

Video



Video 4.5. Proyecto Arduino con sensor de temperatura y humedad.

Como ejercicio, le **indicamos** a ChatGPT 3.5 "Haz un código en lenguaje C para Arduino, que incluya un sensor DHT-22 de temperatura y humedad. El programa debe mostrar en un tabla la temperatura y humedad cada cinco segundos", con el fin de comparar el código con el realizado en el video anterior. Esto respondió ChatGPT:

Claro, te ayudaré a crear un programa para Arduino que utilice el sensor DHT-22 para medir la temperatura y la humedad, mostrando los valores

en el *Serial Monitor* cada cinco segundos. Antes de empezar, asegúrate de tener la biblioteca DHT sensor library de Adafruit instalada en tu IDE de Arduino para poder comunicarte con el sensor DHT-22. Puedes instalarla usando el Gestor de Bibliotecas en el IDE de Arduino (Sketch > Incluir Biblioteca > Administrar Bibliotecas... y busca "DHT sensor library"). Aquí tienes un ejemplo de código:

```
#include "DHT.h"

#define DHTPIN 2 // Definimos el pin donde se
// conecta el sensor
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHT22 Sensor temperatura y humedad");
  dht.begin();
}

void loop() {
  // Esperamos unos segundos entre mediciones.
  delay(5000);

  // Leemos la humedad, la temperatura en Celsius
  float h = dht.readHumidity();
  // Leer temperatura en Celsius
  float t = dht.readTemperature();
```

Continúa el código en la siguiente página

```

// Comprobamos si alguna lectura falló.
if (isnan(h) || isnan(t)) {
    Serial.println("Fallo al leer el sensor DHT22!");
    return;
}
// Mostramos los resultados en el monitor serie
Serial.println("Medida de temperatura y humedad:");
Serial.print("Humedad: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperatura: ");
Serial.print(t);
Serial.println(" *C ");
}

```

Puedes observar que el código es el mismo, con diferencias mínimas en títulos y el pin de conexión.

Una diferencia interesante, es que ChatGPT incluye una estructura de control, que advierte si hay un fallo de lectura del sensor.

Haz clic en la siguiente imagen, para abrir este tercer proyecto:



Ejecuta el proyecto, luego cambia todo el ciclo **loop** por el dado por ChatGPT, vuelve a ejecutar y observa las diferencias.

4.1.3.4 Cuarto proyecto con Wokwi - sensor y semáforo

En este proyecto usaremos el sensor del tercer proyecto, agregando **leds**, que adviertan sobre incumplimientos en las medidas de temperatura o de humedad.

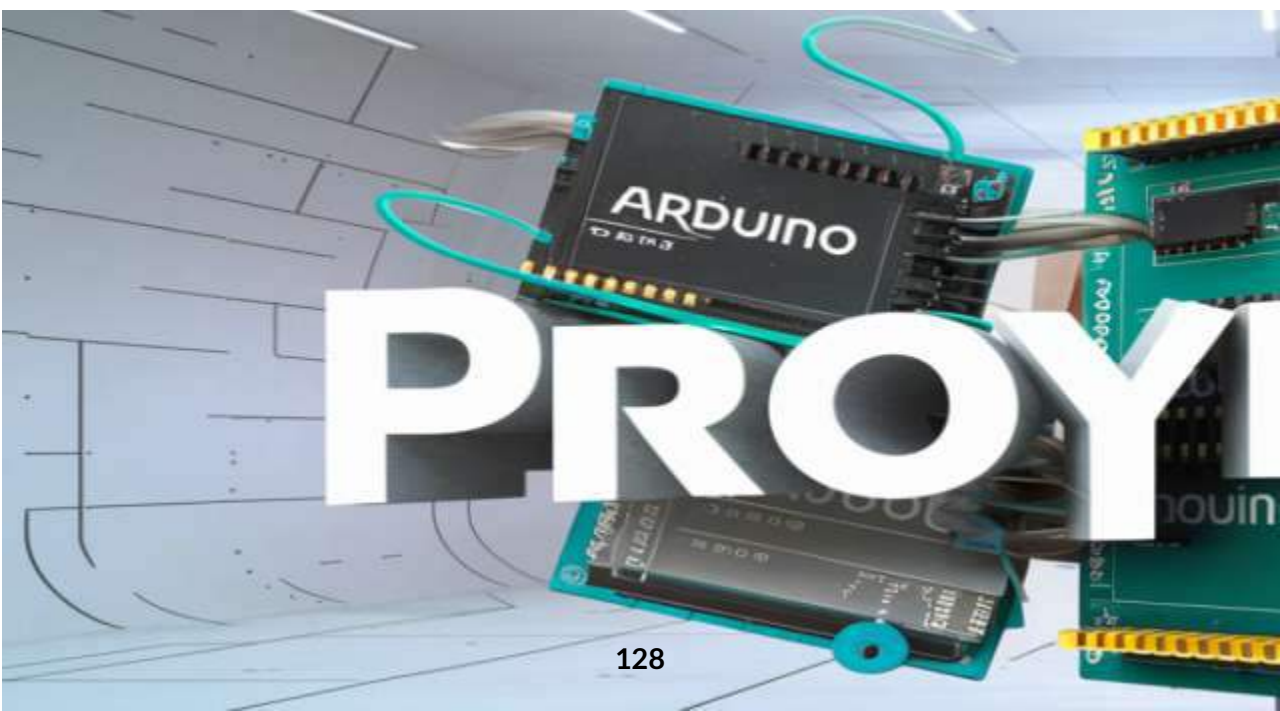
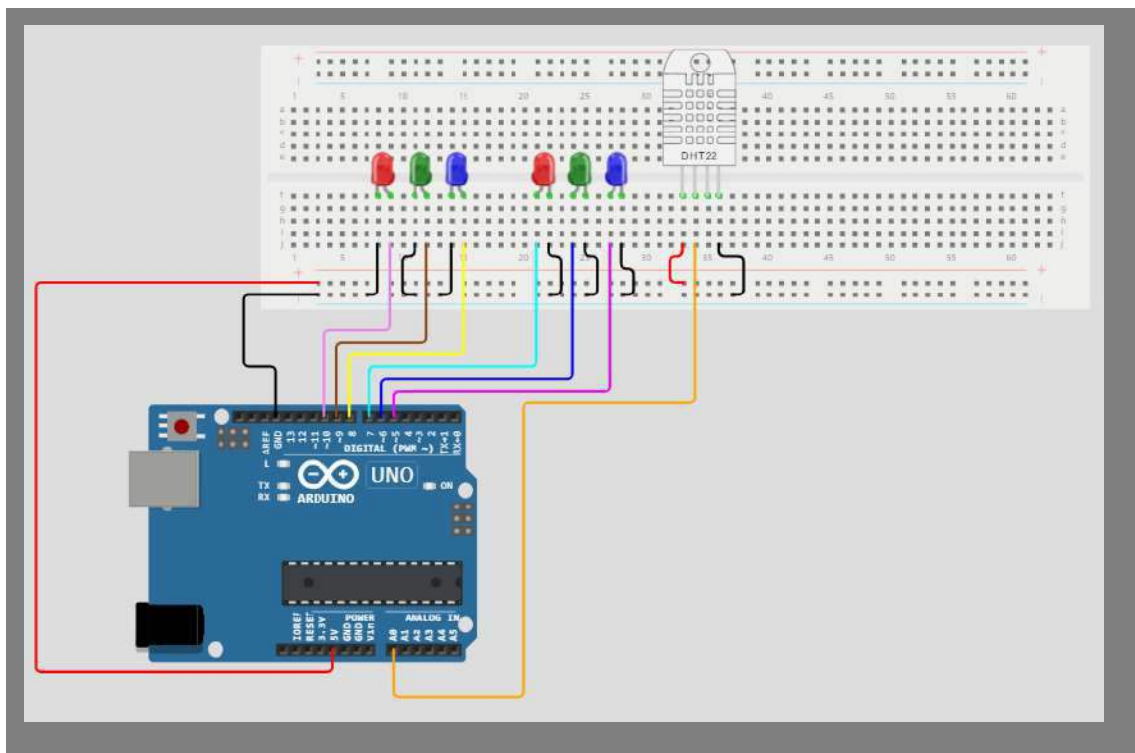
Presta atención al siguiente video, pues de su comprensión depende el diseño del proyecto final de esta capítulo, que un **Planta Inteligente**, en la que se le controla temperatura, humedad e iluminación.

Video



Video 4.6. Sensor y semáforo.

Haciendo clic en la siguiente imagen, puedes acceder al proyecto y ejecutarlo, aunque lo correcto, si quieres aprender, es que lo diseñes de acuerdo a lo explicado en el video.



4.1.4 Proyectos Arduino en Físico

Para estos proyectos, debes contar con los siguientes elementos:

1. Placa Arduino UNO Rev3
2. Cable USB tipoB para Arduino
3. Protoboard tamaño grande o mediana
4. Sensor DHT22 o DHT11
5. Cables 15 cm, macho-hembra, macho-macho y hembra-hembra
6. LEDs de 5mm, colores rojo, verde, azul u otros adicionales
7. Resistencias de 220 o 330 ohm
8. Cables cortos (5 cm)
9. Pantalla 16x2 con I2C

El último elemento (pantalla) es opcional, pues el proyecto de planta inteligente lo vamos conectar a un computador. Pero si deseas saber cómo usar la pantalla 16x2, puedes consultar este [tutorial](#).



Es importante comprender los elementos propios del Arduino; para ello, observa la siguiente imagen:

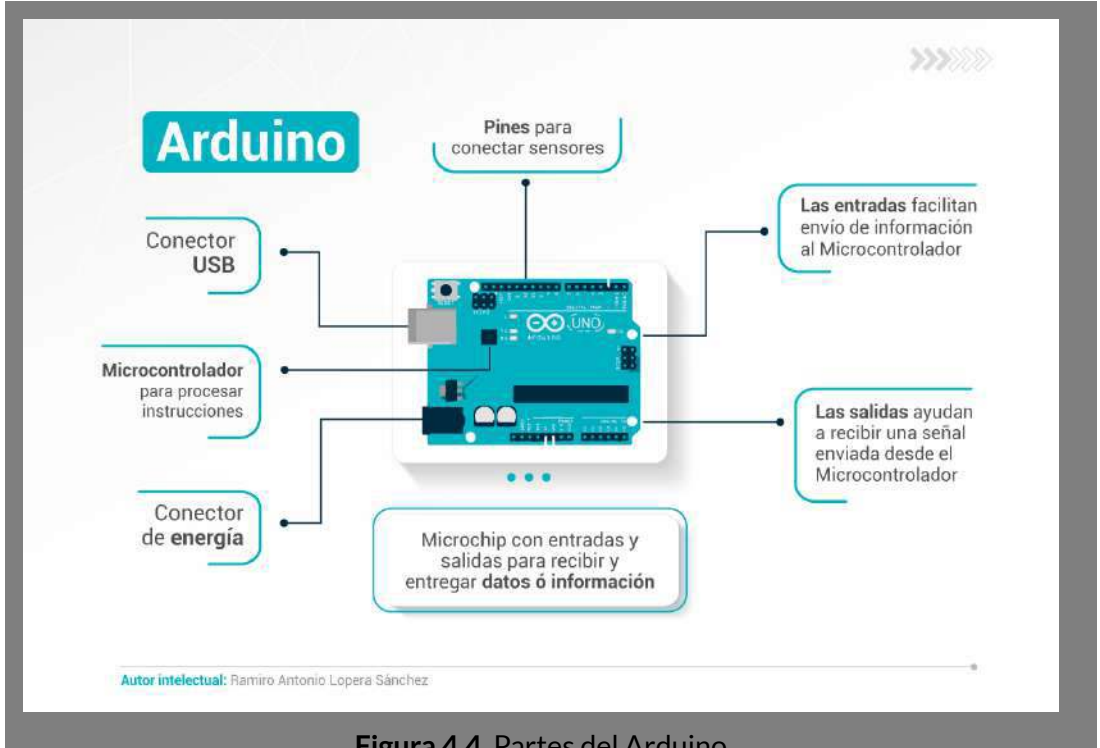


Figura 4.4. Partes del Arduino

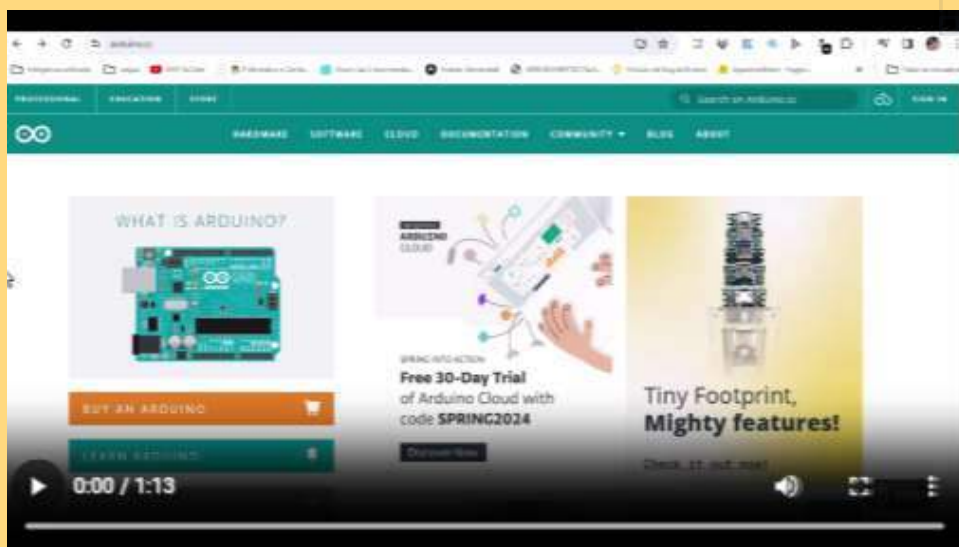
4.1.4.1 Instalación de Arduino IDE

Además de los elementos anteriores, necesitamos una herramienta informática para crear nuestros proyectos físicos; para ello, hay dos opciones. La primera opción es usar una aplicación alojada en la nube, llamada **ARDUINO CLOUD**, a la cual podemos acceder desde este enlace: <https://app.arduino.cc/>. Allí, podemos diseñar nuestros proyectos o, también, conocidos como "bocetos" (*Sketches*)⁷, que es el código con el cual nos comunicaremos con el proyecto físico (Arduino UNO más los elementos conectados).

⁷ Un boceto es el nombre que utiliza Arduino para un programa. Es la unidad de código que se carga y ejecuta en una placa Arduino (<https://docs.arduino.cc/>).

Una segunda opción, que usaremos en nuestros proyectos físicos, es descargar e instalar la aplicación en nuestro ordenador, tal como se indica en el siguiente video:

Video



Video 4.7. Instalación de Arduino IDE.

El software Arduino (IDE) facilita la escritura de código y la carga en la placa sin conexión. Lo recomendamos para usuarios con mala o nula conexión a Internet. Este software se puede utilizar con cualquier placa Arduino.

El IDE 2.3.2 es una nueva versión que es más rápida y más potente que el IDE 1.xx. Además de un editor más moderno y una interfaz más receptiva, incluye funciones avanzadas para ayudar a los usuarios con su codificación y depuración (<https://docs.arduino.cc/>).

4.1.4.2 Primer proyecto físico - uso de sensores

En el apartado 4.1.3.3, diseñamos un proyecto en el simulador Wokwi, el cual hacía uso de un sensor de temperatura y humedad. A continuación, presentamos un video que describe cómo se monta este proyecto **físicamente**.

Video



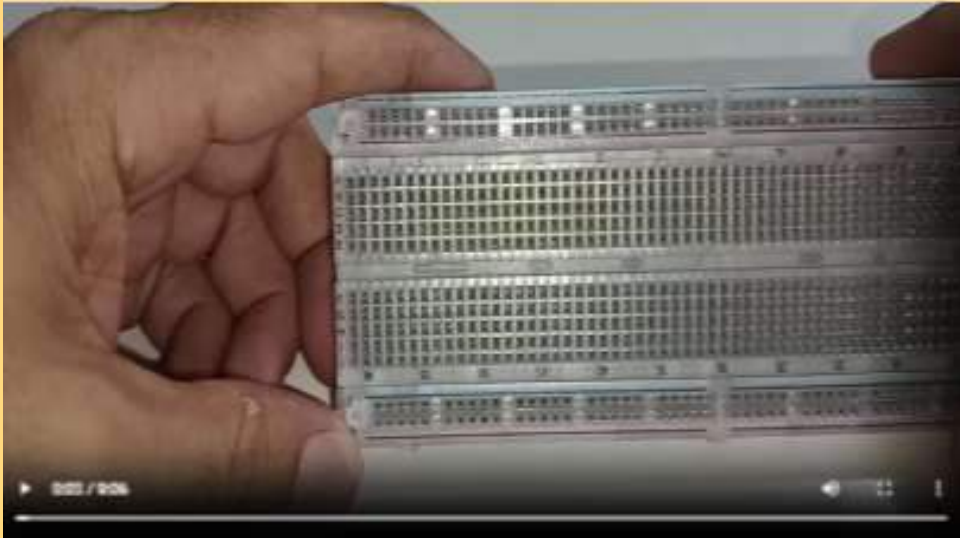
Video 4.8. Arduino con sensor de humedad y temperatura.

Pudiste observar que usamos los siguientes elementos: una placa Arduino UNO Rev3, un cable USB tipoB para Arduino, una protoboard tamaño grande, un sensor DHT11 y cables de 15cm y de 5cm. Te recomendamos realizar este montaje, pues se constituye en un primer paso para llegar a nuestro proyecto final.

4.1.4.3 Segundo proyecto físico - planta inteligente

A continuación, presentamos el montaje de nuestra **planta inteligente**, en forma parcial, pues la idea es que tomes como referencia las instrucciones para que puedas concluir el proyecto.

Video



Video 4.9. Arduino con sensores y leds.

Puedes, también, incluir mensajes de la planta, según sus condiciones de humedad, temperatura e iluminación. Para el caso de humedad, observa un ejemplo en la siguiente imagen.

```
// SECUENCIA DE SENSACIONES DE LA PLANTA INTELIGENTE
// REVISAMOS LAS CONDICIONES DE HUMEDAD DE LA PLANTA
if (humedad >= 80 & humedad <= 90) { // la humedad está en un rango Normal, LEDS apagado
  digitalWrite(ledHumedad, LOW); EstadoPlantaHum = " Mi humedad está perfecta -";
  PuntosHum = 10; // Asignamos 10 puntos de salud
}

if (humedad > 90) { // la humedad está sobre el rango Normal
  digitalWrite(ledHumedad, HIGH); EstadoPlantaHum = " Me estoy ahogando! -";
  PuntosHum = 15; // Asignamos 15 puntos de salud
}

if (humedad < 80) { // la humedad bajo el rango Normal
  digitalWrite(ledHumedad, HIGH); EstadoPlantaHum = " Tengo Sed! -";
  PuntosHum = 5; // Asignamos 5 puntos de salud
}
```

Figura 4.5. Mensajes según las sensaciones de humedad.

4.1.4.4 Tercer proyecto físico - planta inteligente

A continuación, presentamos el montaje definitivo de nuestra **planta inteligente**.

Video



Video 4.10. Arduino con sensores de humedad, temperatura e iluminación.

En la siguiente página, responde las preguntas, que corresponden a la evaluación de este capítulo.



10 preguntas en 120 segundos

IoT

TECHNOLOGY

TIUNOGRAPTCE THE ATT
VISION OT HE FUTURE

Comenzar

Bibliografía

- [1] Elder, Jeff (2019). *Cómo Kevin Ashton nombró el Internet de las Cosas*. [Avasta Blog](#).
- [2] Bonilla, Isaias; Tavizon, Arturo; Morales, Melisa; Guajardo, Luz; Laines, Cristina (2016). IOT, EL INTERNET DE LAS COSAS Y LA INNOVACIÓN DE SUS APLICACIONES. *VinculaTégica*, núm. 1, pp. 2313-2340.
- [3] Pineda, Migdalia (2018). La Internet de las Cosas, el Big Data y los nuevos problemas de la comunicación en el Siglo XXI. *Mediaciones Sociales*, núm. 17, pp. 11-24.
- [4] Yousuf, Tasneem; Mahmoud, Rwan; Aloul, Fadi; Zualkernan, Imran (2015). Internet of Things (IoT) Security: Current Status, Challenges and Countermeasures. *International Journal for Information Security Research*, Vol. 5, núm. 5, pp. 608-616.
- [5] Toben, K.D; Wiesner, S.A.; Wuest, T. (2017). "Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples. *International Journal of Automation Technology*, 11(1): 4-19, Doi: [10.20965/ijat.2017.p0004](#).
- [6] Basco, A.; Beliz, G.; Coatz, D.; Garnero, P. (2018). Industria 4.0: fabricando el futuro. Banco Interamericano de Desarrollo (BID), Buenos Aires.
- [7] Schwab, Klaus. (2016). *The Fourth Industrial Revolution*. World Economic Forum, Ginebra.
- [8] Ávila, Óscar (2011). Computación en la nube. *ContactoS* Núm. 45, pp. 45-52.

- [9] Murty, Bharathi; Shri, Lawanya; Kadry, Seifedine; Lim, Sangsoon (2020). Blockchain Based Cloud Computing: Architecture and Research Challenges. *IEEE Access* Vol. 8, pp. 205190-205205.
- [10] Sánchez, Sergio (2021). *Cloud computing; fundamentos y despliegue de un servicio en la Nube*. Trabajo de grado optar al grado de Ingeniería Informática. Universidad Autónoma de Madrid, España.
- [11] Davenport, T.; Patil, D.J. (2012). Data Scientist: The Sexiest Job of the 21st Century. *Analytics And Data Science*, consultado en [Harvard Business Review](https://hbr.org/2012/06/data-scientist-the-sexiest-job-of-the-21st-century/)
- [12] Cuny, J.; Snyder, L.; y Wing, J. M. (2010). *Demystifying computational thinking for non-computer scientists*. Unpublished manuscript, referenced in <https://www.cs.cmu.edu/>



