Libros de Cátedra

Guía de recomendaciones para diseño de software centrado en el usuario

Javier Díaz
Ivana Harari
Ana Paola Amadeo

exactas

FACULTAD DE
INFORMÁTICA

Edulp
Editorial de la Universidad de La Plata

UNIVERSIDAD NACIONAL DE LA PLATA

GUÍA DE RECOMENDACIONES PARA DISEÑO DE SOFTWARE CENTRADO EN EL USUARIO

Javier Díaz

Ivana Harari

Ana Paola Amadeo

2013

Díaz, Javier

Guía de recomendaciones para diseño de software centrado en el usuario / Javier Díaz ; Ivana Harari ; Ana Paola Amadeo. - 1a ed. - La Plata : Universidad Nacional de La Plata, 2013.

E-Book:ISBN 978-950-34-1030-1

1. Informática. 2. Software. 3. Enseñanza Universitaria. I. Harari, Ivana II. Amadeo, Ana Paola III. Título

CDD 005.3

Fecha de catalogación: 28/10/2013

Diseño de tapa: Dirección de Comunicación Visual de la UNLP

Universidad Nacional de La Plata – Editorial de la Universidad de La Plata

47 N.º 380 / La Plata B1900AJP / Buenos Aires, Argentina
+54 221 427 3992 / 427 4898
editorial@editorial.unlp.edu.ar
www.editorial.unlp.edu.ar

Edulp integra la Red de Editoriales Universitarias Nacionales (REUN)

Primera edición, 2013
ISBN 978-950-34-1030-1
© 2013 - Edulp

Reconocimientos

A la Diseñadora en Comunicación Visual Ariadna Alfano, por su preciosa labor de edición y diseño de portadas, íconos e imágenes que utilizamos en cada capítulo.

A los alumnos que han pasado por la cátedra de Diseño Centrado en el Usuario desde 1992, por sus ideas, iniciativas y desarrollos, varios de los cuales ilustran los ejemplos de este libro.

A la Facultad de Informática de la Universidad Nacional de La Plata junto a la editorial, por respaldar y promover libros de computación que promulgan la importancia de la participación activa de los usuarios en el proceso de ingeniería de software, desde novatos hasta experimentados.

A nuestras familias, por su apoyo permanente e incondicional.

Organización de libro

En este libro se aborda toda la temática relacionada con las interfaces del usuario y su proceso de desarrollo. Se profundiza sobre la etapa de diseño de interfaces específicas, brindando una serie de recomendaciones que permitan alcanzar la calidad de uso del producto. El libro presenta una organización que consiste en seis capítulos, acompañados por ejercitación, casos de estudio y referencias específicas a cada uno.

En el capítulo 1 se introducen los conceptos básicos de la Interacción Hombre – Computadora. Se incluye una breve referencia a su evolución, se aborda el paradigma de independencia de diálogo y métricas de evaluación de una interfaz. El capítulo 2 abarca el proceso de diseño de la interfaz de usuario, haciendo hincapié en las características más importantes, el ciclo de vida de la interfaz de usuario y la prototipación como metodología de desarrollo de una interfaz de usuario. Desde el capítulo 3 al 5, el libro se adentra en recomendaciones de diseño para interfaces específicas como interfaces visuales, icónicas e interfaces para la Web. En el capítulo 6 y último se introducen los conceptos de accesibilidad y las normas internacionales para construir interfaces accesibles.

Al finalizar cada capítulo, se dispone de un conjunto de ejercicios prácticos y un caso de estudio relacionado con la temática. En general, versan sobre la aplicación de las recomendaciones a versiones prototípicas de sistemas desarrollados por los alumnos durante la cursada de la materia. Se han seleccionado para estos casos las mejores entregas de las distintas cohortes del dictado de la materia de Diseño Centrado en el Usuario

Se incluye también un anexo con el glosario de términos y la resolución de las ejercitaciones planteadas en cada capitulo.

Indice Temático — Páginas

Prólogo — pág. 5-6

Organizacion del libro — pág. 7

Acerca de los autores — pág. 7-8

Reconocimientos — pág. 8

introducción — pág. 9

Capitulo I — pág. 10 - 39

Fundamentos en HCI

I- ¿Qué es HCI?
II- Evolución del HCI
III- Tipos de Diálogo hombre-computadora
IV- Clasificación de las Interfaces del Usuario
V- Independencia del Diálogo
- ¿Cómo se logra esta estrategia?
- La aparición de un nuevo rol. El desarrollador del diálogo
- Una nueva interfaz. Diálogo interno y externo
- Complejidades de la independencia de Diálogo

VI- Componentes de la Interfaz del Usuario
VII- Métricas de Evaluación de una Interfaz
- Objetivos de la Interfaz del usuario
- Principios de Nielsen

Caso de Estudio
Ejercitación
Referencias

Capitulo II — pág. 40-62

Proceso de Diseño de la Interfaz del Usuario

I- Introducción
II- Características del Proceso de Desarrollo de la UI
III- Ciclo de Vida de la Interfaz del Usuario
IV- Etapa de Requerimientos de la Interfaz
- Modelo SSOA

V- Etapa de Diseño de la Interfaz
- Características del Proceso de Diseño

VI- La interfaz y la Prototipación
- Características de la Prototipación
- Ventajas de la Prototipación
- Tipos de Prototipos

VII- Etapa de Implementación
- Herramientas para el desarrollo de la Interfaz

VIII- La importancia de la Evaluación de las interfaces del usuario
- Métodos de Indagación
- Métodos de Inspección

Ejercitación
Referencias

Capitulo III — pág. 63-94

Normas de Diseño de las Interfaces Visuales

I -Introducción
II -Características de las Interfaces Visuales
III - Descripción de las Técnicas de Interacción Visual

IV - Las Técnicas de Interacción Visual y sus Aspectos de Diseño
- Múltiples Ventanas
- Menús
- Manipulación Directa
- Gestos
- Feedback Visual
- Dependencias
- Animaciones
- Exploradores de Objetos
- Cajas de Diálogo

Ejercitación
Caso de estudio
Referencias

Capitulo IV — pág. 95-114
Cualidades de una Interfaz Icónica

I -Introducción
II -Características de una Interfaz Icónica
- El Diseño Icónico y los 7 Pasos de Norman
- El Diseño Icónico y la Correspondencia
- El Diseño icónico y la Consistencia

III -Ventajas de una Interfaz Icónica
IV - Complejidad del Diseño Icónico
V - Especificación Formal de una Interfaz Icónica
VI - Ejercitación
VII - Caso de estudio
VIII - Referencias

Capitulo V — pág. 115-141
La Interfaz Web y sus Principios de Diseño

I- Introducción
II- Características Específicas de un Sitio Web
III- Particularidades en el Desarrollo Web
- Actividades previas al Diseño del Sitio
- La Ingeniería de Usabilidad

IV- Principios de Diseño del Sitio Web
- Principios de Diseño a Nivel de Sitio
  - La estructura del sitio Web
  - La navegación del sitio Web
  - Los hipervínculos
- Principios de Diseño a Nivel de Página
  -Diseño general de las páginas Web
  -La página inicial
  -Página con formularios
- Principios de Diseño a Nivel de Contenido
  -Diseño general del contenido
  -Lenguaje empleado

Casos de Estudio
Ejercitación
Referencias

Capitulo VI — pág. 142-172
Diseño de Interfaces Accesibles

I- Introducción
II- Accesibilidad en la Interfaz del Usuario
- Consideración de distintas discapacidades

- Análisis de herramientas para discapacitados

III- Accesibilidad en la Web: su problemática
- Falta de Diseño Universal en la Web
- Falta de Cumplimiento de las Normas de Accesibilidad
- Problemáticas con los Recursos de Adaptación

IV- Legislación y Normas Internacionales sobre Accesibilidad

V- Normas de Accesibilidad en la Web
- Principios generales de Accesibilidad
- Recomendaciones de Accesibilidad de la W3C-WAI
  - -WCAG 1.0
  - -WCAG 2.0

VI- Comprobación de la Accesibilidad Web

Casos de Estudio

Ejercitación

Referencias

CONCLUSIONES GENERALES pág. 173-174

APENDICES pág. 176-187

GLOSARIO DE TÉRMINOS UTILIZADOS pág. 177-180

EJERCITACION RESUELTA pág. 181-187

- Capitulo I
- Capitulo II
- Capitulo III
- Capitulo IV
- Capitulo V
- Capitulo VI

Prólogo

Si usted está a punto de comenzar a desarrollar un sistema de software, - ¡no programe todavía!. Antes de comenzar a escribir las sentencias que constituirán su futuro sistema de software, deténgase a analizar si previamente resolvió cuestiones de diseño fundamentales, para garantizar una mejor calidad de su producto.

Esto no significa únicamente contar con un análisis de requerimientos exhaustivo, con el diseño completo de los algoritmos funcionales y de la representación de los datos que el sistema va a gestionar.

Hay otras cuestiones que deben realizarse antes de la etapa de programación, que es el tratamiento de la interfaz del usuario como una componente del software autónoma tan importante como la parte funcional del sistema.

La misma, debe ser analizada y diseñada con detenimiento, puesto que constituirá el único medio que el usuario tendrá para acceder y alcanzar la funcionalidad subyacente, será su puerta de entrada, su único ambiente de trabajo y de comunicación en donde podrá llevar a cabo sus intenciones y su resolución de problemas.

Hay muchas expresiones por parte de desarrolladores de software, que se han oído en el ámbito informático, ante la experiencia directa con usuarios finales. Expresiones tales como:

-¡Pero... ¿por qué critican mi sistema?, si funciona!;

-¡¿Por qué después de instalarle el sistema, el usuario me llama constantemente para que le explique cómo utilizarlo?!;

-¡¿Por qué el usuario se equivoca tanto?... si el sistema realiza bien todas las funciones que me pidió!, o

-¡Ya es imposible seguir reabriendo y modificando el código por cuestiones superficiales de apariencia y comportamiento de la interfaz!.

El objetivo de este libro, es justamente plantear las recomendaciones de diseño a nivel de interfaz necesarias para que usted justamente no sea el próximo en realizar alguno de estos planteamientos.

Lo que sucede, es que por siempre se consideró al usuario como aquel que especificaba los requerimientos del sistema y, el que luego se encargaba de la carga de datos para el procesamiento. El contacto con ellos era en la etapa inicial y al final, para el uso del producto. Pero, hay que tener en cuenta, que ellos determinan el logro del producto, intervienen en cualquier medida de éxito, como ventas, visitas, quejas, sugerencias. Ellos influyen no sólo en la elección y utilización del software, sino en el grado de permanencia del mismo, en su vida útil.

Es apropiado, entonces, pensar en darle al usuario, otro grado de importancia durante el desarrollo mas que nada de la componente de interfaz, que reúne aspectos subjetivos, tiene en cuenta los factores humanos y las formas de interacción adecuadas al perfil del usuario.

Hoy en día, debe existir una mayor reconciliación, un cooperativismo entre el usuario y el desarrollador, donde el usuario participe, decida, critique, evalúe y apruebe en un marco prototípico, mientras el producto está en vías de desarrollo.

Entonces, a lo largo de esta guía, se van a abordar los temas más importantes referentes a la Interfaz del Usuario, sus características, su proceso de desarrollo y se va a profundizar principalmente en su etapa de diseño, en el análisis e investigación de principios o normas que garanticen una correcta componente de interacción.

Esto constituirá la base para que el desarrollador pueda contar con un diseño completo y comprobado ante el usuario, de la interfaz que se va a proveer en el sistema, garantizando una directa traslación a la codificación y una aprobación final del producto.

Vale aclarar, que mucho del contenido expuesto en este libro, se basa fundamentalmente en la experiencia vivida a lo largo de una vasta carrera docente y profesional, con más de diez años de formación e investigación en temas de Interfaz del Usuario, Interacción Hombre-Computadora, Diseño y Prototipación.

Muchos de los ejemplos y casos planteados como recomendaciones surgen de la actividad académica y profesional, del contacto directo con los alumnos que cursan, o han cursado, las materias Diseño Centrado en el Usuario e Interfaces Avanzadas en la Facultad de Informática de la Universidad Nacional de La Plata, desde el año 1992 a la fecha.

Desde esa fecha, se percibió la necesidad de incluir en la formación de los profesionales informáticos, la perspectiva del usuario y de incluir dentro de la Ingeniería de Software, en permanente evolución, un libro especialmente dedicado a las Interfaces del Usuario.

Muchas gracias.

Prof. Francisco Javier Diaz
Director del área de investigación
de Interfaces del Usuario.
Laboratorio de Investigación
en Nuevas Tecnologías Informáticas.

Lic. Ivana Harari
Profesora de Diseño Centrado
en el Usuario e Interfaces Avanzadas

Lic. Ana Paola Amadeo
Jefe de Trabajos Prácticos de Diseño
Centrado en el Usuario e Interfaces Avanzadas

Introducción

Millones de personas utilizan hoy en día recursos informáticos tales como computadoras, redes de computadoras, tablets, smartphones, para solventar problemas de comunicación, de trabajo o por cuestiones cotidianas.

A medida que el número de usuarios se amplía cada día, incorporando personas de formaciones y culturas dispares, aumenta a la par la demanda de los mismos de adquirir sistemas de software con mayor nivel de asistencia y facilidad de uso.

Sistemas interactivos que provean un estilo de comunicación más simple, que sean cada vez más fáciles de utilizar, de instalar, de aprender, sin requerir ningún tipo de entrenamiento específico en Informática. Además, se pretende que el software brinde mecanismos de interacción más inteligentes y que se adapte al usuario en forma eficiente.

El usuario necesita ver al sistema interactivo que está utilizando en su computadora como una herramienta capaz de ayudarlo a resolver sus problemas, a concretar sus intenciones. Pero esto no siempre es así. Es muy común que se le sume innumerables problemas más al utilizar el software, debiendo solucionar fallos o errores producidos cuyas causas no son claras, aprender cuestiones técnicas imprevistas, incurrir a ciertas artimañas para sortear obstáculos innecesarios, adivinar cómo realizar determinada tarea recurriendo a la técnica de prueba y error.

Aunque la componente funcional esté correcta y completamente desarrollada, existen un sin número de problemas de interacción que pueden empañar el acceso y productividad de la misma, provocando en el peor de los casos, la modificación general del software o el desuso.

Estos problemas de interacción residen en una componente del software denominada “Interfaz del Usuario”, que se encarga de innumerables cuestiones como la entrada y salida de la información, visualización de los datos, soporte, manejo y control del diálogo con el usuario final, entre otros.

La interfaz del usuario o componente de diálogo es una parte del software cuyo diseño afecta el nivel de productividad del sistema general, incide en el grado de satisfacción que pueda percibir el usuario, por lo tanto es determinante para la elección, utilidad y evaluación final del software.

Como la interfaz del usuario se encuentra dentro del sistema de software, su diseño e implementación está en manos del especialista informático, no puede delegarse a un diseñador visual o a cualquier otro profesional.

Es obligación del desarrollador de sistemas atender tanto cuestiones funcionales como de interacción, dando el mismo nivel de importancia a cada una de estas partes y responsabilizándose de su integración, desarrollo y control.

Es necesario entender que el desarrollo de la interfaz del usuario forma parte del proceso de la Ingeniería del Software. Como es la única parte del sistema que interactúa directamente con los usuarios, es fundamental efectuar un estudio completo del comportamiento humano.

Se debe incluir como requerimientos indispensables en la generación del software parámetros tales como ergonometría, facilidad de uso, amigabilidad, simpleza (en términos de reducir el esfuerzo mental del usuario de llevar a cabo en la máquina la tarea en mente o pensada), flexibilidad, naturalidad (en el sentido que el usuario a través de la pantalla vea reflejado o modelado su realidad).

La interfaz del usuario debe ser diseñada teniendo en cuenta principios de diseño propios que no coinciden con la componente de aplicación. Debe ser planteada bajo la consideración de factores específicos que inciden en su gestación, tales como factores humanos principalmente, perfil de los usuarios, elementos del entorno, hardware disponible.

Su diseño parte desde un profundo conocimiento de las/los usuarios finales del software y su contexto, más que desde los términos computacionales y algoritmos de programación.

Debido a que el proceso de diseño está centrado en el usuario, se trabaja con un nivel de incertidumbre y ambigüedad elevados, por lo que hace que el ciclo de vida de la interfaz sea especial y deba ser tratado en forma independiente. Esta componente se construye con la práctica, necesita ser diseñada, evaluada varias veces hasta alcanzar los requerimientos de los usuarios. Se generan versiones prototípicas de la interfaz que son puestas a prueba aunque la misma esté incompleta y aunque aún no tenga acoplada la componente funcional.

En este manual, se va a analizar estas cuestiones, se va a realizar un abordaje desde lo conceptual y metodológico y se va a desplegar una serie de normativas que hacen al buen diseño de la componente de interacción, teniendo en cuenta distintos tipos de interfaces del usuario. Se incluyen ejemplificaciones y se analizan casos de uso prácticos concretos.

Mediante el estudio, exploración y puesta en práctica de estas normas y recomendaciones de diseño, se logra mejorar significativamente los canales de comunicación e interacción con el usuario final, optimizando la calidad del sistema en general.

01

La multiplicidad de aspectos, metodologías y de procesos a considerar para el diseño de una correcta interfaz, que sea óptima desde el punto de vista del usuario y acorde a la lógica del objetivo específico del sistema, hace necesario conocer todos los conceptos y terminologías que estén íntimamente relacionados con el área de HCI.

Temas a Tratar

I- ¿Qué es HCI?
II- Evolución del HCI
III- Tipos de Diálogo hombre-computadora
IV- Clasificación de las Interfaces del Usuario
V- Independencia del Diálogo
- ¿Cómo se logra esta estrategia?
- La aparición de un nuevo rol. El desarrollador del diálogo
- Una nueva interfaz. Diálogo interno y externo
- Complejidades de la independencia de Diálogo

VI- Componentes de la Interfaz del Usuario
VII- Métricas de Evaluación de una Interfaz
- Objetivos de la Interfaz del usuario
- Principios de Nielsen

Caso de Estudio
Ejercitación
Referencias

I - ¿Qué es HCI?

La interacción hombre–computadora, comúnmente referida con la abreviatura HCI por Human Computer Interaction, es el intercambio observable de información, datos y acciones entre un humano y la computadora, y viceversa.

Está constituida por el diálogo, la conversación, la comunicación que pueda fluir en ambas direcciones, entre una persona y el sistema de software que está utilizando.

La interfaz del usuario, en cambio es el medio por el cuál la interacción hombre-máquina es establecida, manejada y controlada. Conforma la parte del software y hardware que permite, que ese intercambio de información y sus distintas secuencias o hilos de diálogo, se produzcan.

Estos dos términos, interacción hombre-máquina e interfaz del usuario, están muy entrelazados en el proceso de desarrollo y en estos casos se los usa como sinónimos. En muchos contextos ambos conceptos se refieren a las entradas del usuario final, su tratamiento localizado de las mismas, la visualización de los estados del sistema y la presentación de las salidas o respuestas.

La transformación funcional o algorítmica de las entradas de la información a las salidas del sistema, pertenece a la componente computacional o de aplicación. Entonces, a través de HCI, se despega la lógica de la interacción como por ejemplo, la lógica del negocio.

Desde una concepción más amplia, se puede definir HCI también como una disciplina. Se la considera un área dentro de las Ciencias de Computación que se encarga del diseño, evaluación e implementación de sistemas de computación interactivos, para el uso humano, incluyendo el estudio de todos los fenómenos concernientes a ello.

Analizando más en profundidad esta última definición, los alcances de esta nueva disciplina son muy difusos y, pueden influir en ella factores muy heterogéneos, desde aspectos tecnológicos vinculados al hardware a aspectos sociales o psicológicos del usuario. Inciden además, cuestiones diversas como facilidad de uso, productividad, eficacia para expresar distintas posibilidades, capacidad de adaptación de acuerdo a distintos usuarios, desde niños a personas de mayor edad, pasando por distintas culturas y características sociales.

Como es un área que se centra fundamentalmente en la interacción entre humanos y máquinas, se pueden plantear muchas situaciones diferentes. Cuando se refiere a “humano”, puede considerarse desde un usuario particular, un grupo de usuarios, una organización o corporación, hasta el mundo entero, como es el caso de las aplicaciones para Internet. Cuando se refiere a “máquina”, puede incluirse desde una simple PC, una workstation, una Intranet o hasta una supercomputadora con máquinas computacionales embebidas.

Además, HCI se convierte en un área interdisciplinaria, que puede nutrirse de nociones provenientes de la Psicología, por la aplicación de teorías de procesos cognitivos y el análisis empírico del comportamiento del usuario; de la Sociología y Antropología, por el estudio de la interrelación entre la tecnología, el trabajo y las organizaciones; del Diseño Industrial, por el estudio y diseño de productos de interacción; del Diseño Visual, por el uso de las formas, colores y otros paradigmas visuales y sus diferentes aplicaciones; entre otros.

Otro punto que no hay que olvidar, son los aspectos relacionados con la Tecnología e Ingeniería, vinculados a la capacidad gráfica de las pantallas y dispositivos, para que el usuario ingrese información, ya sea por teclado, mouse, lápiz óptico, pantalla digital, voz, teléfono, entre otros medios.

Entonces, mientras exista el diálogo entre una máquina y un humano, surgirán innumerables factores relacionados con el HCI, que incluirán todos los aspectos para el diseño y construcción de interfaces del usuario adecuadas.

A continuación, se mostrará una figura que resume las interrelaciones entre los diferentes tópicos que aborda el área de HCI.

Figura 1.1: Variedad de Tópicos que involucra el HCI

Los sistemas de computación existen dentro de un medio social, organizacional y de trabajo. Dentro de este contexto, se encuentran las aplicaciones. Incorporar computadoras en el trabajo implica un proceso de integración.

Además del uso y contexto social de las computadoras, del lado humano debemos considerar el procesamiento humano de la información, la comunicación -el lenguaje- y las características físicas del usuario -ergonometría-.

Desde el punto de vista de la computadora, una variedad de tecnologías han sido desarrolladas para soportar la interacción con humanos, como los dispositivos de entrada y salida. Existen numerosas técnicas para organizar el diálogo y generarlo, las cuáles sirven para implementar cuestiones más específicas del mismo. Ejemplos de estos se encuentran en las técnicas de visualización por computadora, técnicas para diálogo multimedial, para diálogo no visual o conversacional, técnicas de Inteligencia Artificial, de diálogo basado en agentes, técnicas para diálogo multi-usuario, entre otras.

Diálogos complejos o interfaces del usuario con características de avanzada, pueden llevar a consideraciones de arquitecturas de sistemas necesarias para soportar cuestiones como tiempos de respuesta, interconectividad de redes, múltiples ventanas, interfaces multiusuarios cooperativas, utilización de bases de datos, acceso a servicios específicos, entre otros.

Finalmente, hay un proceso de desarrollo que incorpora el diseño del diálogo, herramientas de programación y técnicas de evaluación, cuyos resultados afectan las etapas anteriores, retro alimentándolas, en un proceso de continua mejoría.

II - Evolución del HCI

Debido a que las formas, los medios, los canales de interacción entre los humanos y las computadoras se modifican con el tiempo y evolucionan muy rápidamente, la disciplina de HCI es muy sensitiva a los cambios y tiempos de respuesta.

Un caso concreto fue el descubrimiento del mouse, creado por Douglas C. Engelbart en el año 1968. Este pequeño dispositivo produjo desde el punto de vista del HCI, el surgimiento de un nuevo estilo de interacción y de comunicación, incorporado en las Interfaces Gráficas. El usuario pasó, de expresar sus intenciones mediante la escritura de comandos, a demostrarlas manualmente mediante el señalamiento, el cliqueo, el arrastre de objetos. Además, trajo como consecuencia toda la programación gráfica basada en el mouse y sus eventos.

Entonces, la interacción hombre-máquina o la interfaz del usuario se ve afectada por diversos factores. Puede variar dependiendo de diferentes situaciones, haciendo sumamente complejo el diseño de la misma. Proveer el mejor estilo de diálogo en la interfaz, la visualización y el comportamiento más óptimos no es una tarea fácil, se maneja con información incierta, con un alto nivel de ambigüedad y variabilidad temporal, que hace complicado tener parámetros absolutos o estáticos, sobre el grado de corrección de la misma.

Se requiere de un proceso de evaluación permanente y espíritu crítico para que la interacción sea más natural. Lo que se percibe como “amigable” varía con la introducción de tecnologías y el surgimiento de actitudes distintas, por ejemplo, la simplicidad y naturalidad con que los niños y adolescentes usan los mensajes de texto de la telefonía celular.

Recordar

❒HCI se encuentra íntimamente relacionada con el avance de la tecnología, de las comunicaciones y de los efectos sociales que se producen con el tiempo.

Existen muchos elementos que provocan que el modo de interactuar entre un ser humano y una computadora se modifique y evolucione con el tiempo. Entre los aspectos más influyentes en la evolución del HCI, se pueden mencionar:

Aspectos Influyentes	Explicación
Cuestiones de hardware	■Cada vez el usuario podrá contar con hardware más potente, con mayor capacidad de memoria y velocidad. La interfaz que se diseñe debe tener en cuenta aspectos de rapidez, flexibilidad.
Nuevas tecnologías en dispositivos	■El usuario puede conseguir dispositivos no tradicionales como medio de interacción, como ser cámaras, anteojos especiales, lápiz óptico, o como es el caso del surgimiento del teléfono celular y PALMs para acceder a aplicaciones de Internet.

Tamaño de las pantallas	■Hay disponibles desde monitores muy grandes, finos y livianos, hasta dispositivos como los celulares con pantallas reducidas de hasta 4 a 5 líneas de texto. ■Los primeros, permitirán desarrollo “paper-like”, sistemas de interacción que se basan en una pantalla parecida a un papel electrónico, mientras que los segundos, permitirán portabilidad y movilidad llevando la interfaz en el bolsillo.
Incremento e innovación en técnicas de entrada	■Hay un avance específico en los medios de entrada, con la aparición de la voz, gestos, lápiz ópticos, celulares.
Incremento de usuarios de distintas formaciones	■Se suman día a día potenciales usuarios como niños, adultos mayores, gente con ciertas discapacidades, con distintas formaciones, entrenamiento o preparación. ■La interacción de las mismas con la computadora, presenta características muy particulares a considerar dentro de la interfaz.
Inclusión de la Computación en todos los ambientes y contextos	■Muchos elementos domésticos como microondas, portones, hasta los automóviles vienen con características de programación y el usuario se acostumbra a manejar consolas y visores muy específicos, con interfaces muy distintas entre sí.
Comunicación en cualquier lugar	■Computadoras que se comunican a través de LANs de alta velocidad, nacionalmente mediante WANs y vía computación móvil con tecnologías de infrarrojo, ultrasónico o celular. ■Por lo tanto, servicios de datos y de computación serán accesibles en cualquier lugar y momento, alterando las formas de interacción con la aplicación.
Sistemas con gran componente funcional	■Cada vez más son requeridos servicios de asistencia y búsquedas para facilitar y ayudar el trabajo de los usuarios. ■Los sistemas interactivos deben incluir manuales, tutoriales interactivos para un aprendizaje rápido, ayudas contextuales y procesos de inferencia, ya que la gente no tendrá tiempo de aprender en forma tradicional.
Disponibilidad masiva de “Computer Graphics”.	■Cada vez más se aumentará la capacidad de gráficos por computadora, procesamiento de imágenes, transformaciones de imágenes, animaciones interactivas, que serán disponibles en chips baratos integrables fácilmente en cualquier computadora.
Interfaces para “GroupWare”.	■Las Interfaces “Groupware” permiten a grupos de personas coordinarse, encontrarse y proyectar en forma conjunta, mediante las computadoras. ■Las mismas generan un gran impacto en la naturaleza de organización y en la distribución de labores y responsabilidades.
Multimedia, Hipermedia y Comunidades Virtuales	■La utilización de múltiples medios de comunicación como texto, audio, gráfico, animaciones o video, aplicados en programas multimedia como juegos, enciclopedias, cursos de entrenamientos, hace que la interfaz del usuario deba potenciar estos recursos visuales y brindar mecanismos para operarlos. ■También, el surgimiento de las comunidades virtuales, modifican las características de la interfaz, brindando mecanismos específicos de interacción para que grupos de individuos compartan un interés común vía correo electrónico, salas de chat, sitios de noticias, blogs o wikies.
Computación Ubicua	■La disponibilidad de contar con muchas computadoras a través de un ambiente físico, siendo invisibles al usuario, requiere de cuestiones técnicas esenciales como consumo de energía, conectividad inalámbrica, más una correcta interfaz del usuario. ■Constituye lo que se menciona como el 3er. movimiento en Computación -Third Wave-, donde hay n computadoras por usuario. El 1er.movimiento era n usuarios por computadora mientras que el 2do. fue 1 usuario por computadora.
Acceso ágil a Bases de Datos masivas	■Se requieren de sistemas de visualización y métodos de búsqueda flexibles.
El concepto de “User tailorability”	■Usuarios comunes que tendrán la capacidad de adaptar, personalizar aplicaciones para su propio uso. Podrá generar nuevas aplicaciones basadas en su entendimiento de su propio dominio. Por lo tanto serán como autores de sus propias aplicaciones.

Tabla 1.1: Aspectos que inciden en la Evolución del HCI.

III - Tipos de Diálogo Hombre - Computadora

En el año 1986, E.L.Hutchins, estableció que existen al menos dos metáforas o formas en que los humanos interactúan con una computadora: la “metáfora conversacional” y la del “modelo del mundo”. Estas dos metáforas corresponden unívocamente a dos tipos de diálogo, el secuencial y el asincrónico respectivamente.

En la metáfora conversacional, la interfaz del usuario se basa en representaciones implícitas de los objetos. El usuario debe describir acciones para manipular objetos en forma indirecta. Además, él debe pasar de una etapa de diálogo a la siguiente de una manera predecible y preestablecida por el sistema.

Esta metáfora, permite tanto a los desarrolladores de software como a los usuarios, visualizar una secuencia lógica y específica del comportamiento. Por tal motivo, el diálogo que genera se denomina “Diálogo Secuencial”. Este tipo de diálogo está incluido en la interacción pregunta-respuesta (request-response), lenguajes de comandos, navegación a través de menús y en las entradas de datos.

El diálogo secuencial, está relacionado con la interacción vía el teclado, que generalmente se utiliza en los lenguajes tradicionales donde las sentencias de entrada y salida, están intercaladas con el cómputo del sistema.

En la metáfora del modelo del mundo, la interfaz permite que el usuario final realice las acciones directamente sobre representaciones de los objetos del mundo real. El usuario demuestra sus intenciones, mediante el desplazamiento y manipulación de esas representaciones visuales de los objetos. Por tal motivo, el estilo de interacción característico de esta metáfora, es la “Manipulación directa” y el mouse, es el dispositivo físico apropiado para esto. También, se lo referencia como “Diálogo Asincrónico”, pues es el usuario quien decide cuándo iniciar el diálogo y con qué objeto hacerlo.

A diferencia del diálogo secuencial, donde se presenta ante el usuario una tarea por vez, aunque sea elegir entre varias opciones de un menú, en el diálogo asincrónico se dispone de muchas opciones al mismo tiempo. Es asincrónico desde el punto de vista que la secuenciación de cada camino es independiente de los demás. También al diálogo asincrónico se lo denomina “Diálogo basado en Eventos”, pues son las acciones del usuario que inician la secuencia del diálogos vistas como eventos de entrada.

Asociados a esta metáfora del modelo del mundo, se pueden encontrar los “Diálogos no lineales” o “multi-thread”, que se refieren a la multiplicidad de caminos de diálogo, de hilos de control simultáneos, de acciones alternativas disponibles para el usuario en un determinado momento. Las cajas de diálogo están relacionadas estrechamente con este tipo de diálogo.

Otro tipo de diálogo, es el “Diálogo Concurrente” que es un diálogo multi-thread, en el cuál más de un hilo de diálogo puede ser desarrollado paralelamente. Por ejemplo, un reloj que esté presente mientras que el usuario está dibujando. Además, de existir muchas alternativas de diálogo, éstas están abiertas en forma simultánea.

Entonces, se tienen:

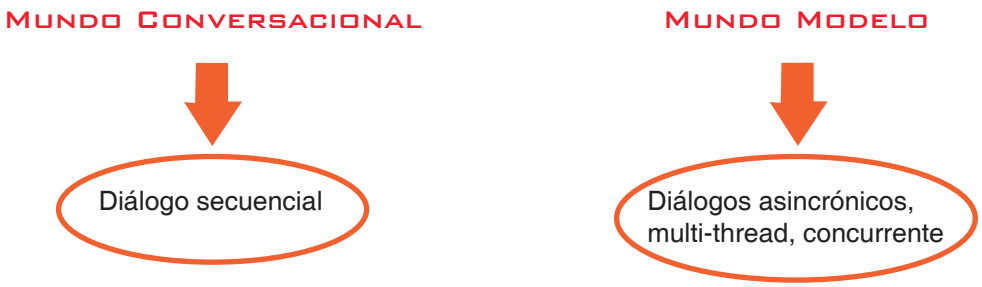


Figura. 1.2: Tipos de diálogos.

En la actualidad, están surgiendo otras formas de interacción, como la “Interacción social o grupal”, donde existe un grupo de usuarios que interactúa en forma conjunta con la interfaz. Este tipo de interacción, es el provisto en las “Interfaces para Groupware”, que se explicará más adelante. Se caracteriza por tener que amoldarse a la complejidad y dinámica del grupo, y por proveer mecanismos específicos para solventar cuestiones de coordinación, comunicación y colaboración entre los miembros del mismo.

También, emergió el concepto de “Interfaz o Interacción basado en Agentes”, infundido por Pattie Maes [Maes P. Wexelblat A., 1996], en donde el diálogo ya no es iniciado y controlado por el usuario como en Manipulación directa, sino que la interfaz tiene la capacidad de hacerlo. La misma puede, tanto empezar una conversación como finalizarla, puede guiar, preguntar, sugerir, enseñar, interrumpir y demás actitudes, que no son admisibles en los tipos de diálogos convencionales. Este tipo de interacción, está basado sobre una metáfora en donde la interfaz se comporta como un interlocutor humano, que tiene por objetivo colaborar con el usuario en la resolución de sus problemas.

El concepto de manipulación directa sigue vigente hoy en día en las aplicaciones que soportan interación gestual como las presentes en smartphones, tablets, juegos para XBOX, Wii y PS Move.

IV - Clasificación de las Interfaces del usuario

Con la llegada de los sistemas de tiempo compartido en la década del 70', el dispositivo de interfaz disponible era el teclado. Esto provocaba el desarrollo de un determinado estilo de interfaz que era totalmente textual, denominado "Interfaz orientada a Comandos".

Un procesador de comandos es un sistema al cuál se le ingresa una cadena de caracteres que luego, será analizada para determinar la función apropiada de la aplicación a invocar.

Este es el caso de la interfaz del usuario más elemental, se caracteriza por ser fácil de desarrollar, se interactúa únicamente con la consola y teclado, las pantallas y las salidas del sistema son tradicionales, con impresiones a cadena de caracteres. Esto tiene ciertas limitaciones, puesto que la entrada desde el teclado provoca mayor tasa de errores, requiere que el usuario recuerde el conjunto de posibles entradas legales, y brinda una visión de la aplicación, del estilo "verbo-objeto" que no resulta amigable, ya que el usuario percibe que es la aplicación la que tiene el control total de la interacción.

Luego, a medida que se incrementó la complejidad del hardware, aparecieron las "Interfaces orientadas a Menús", que se caracteriza por presentar un conjunto de opciones, que pueden ser seleccionadas por los usuarios. Con estas interfaces se permitió abstraer la interfaz de la aplicación, pues fuerza al desarrollador a considerar el espacio de comandos y acciones como una entidad independiente.Se caracteriza por ser fácil de usar y de implementar, la prefirieron la mayoría de los usuarios inexpertos, sin entrenamiento previo.

Estudios sobre factores humanos demuestran que el proceso de lectura y elección de las opciones del menú, resulta ser más sencillo que la invocación a comandos, que requiere recordar la sintaxis de los mismos. Además, alienta a la navegación y exploración del sistema. Pero también, se le atribuyen ciertos inconvenientes como, por ejemplo, en los casos de presentar listas de opciones demasiado extensas donde resulta una técnica engorrosa, o cuando hay demasiados menús anidados que puede provocar la pérdida del contexto.

Tanto las Interfaces basadas en Comandos, como las orientadas a Menús, utilizan al texto como el único medio de representación y de interacción, empleando un diálogo secuencial, por lo tanto están encuadradas en lo que se denomina "Interfaces Textuales".

A mediados de la década de los 80', debido a los grandes avances tecnológicos, la aparición de pantallas con mayor definición, dispositivos de interacción gráficos, como por ejemplo, el lápiz óptico o el mouse, comenzaron a desarrollarse para los sistemas de software, un estilo de interfaz más poderoso que el textual.

Así, surgieron las "Interfaces Gráficas" (GUI) o "Interfaces Visuales", que se caracterizaron por la utilización de recursos visuales para la representación de los objetos y por permitir la manipulación directa de los mismos, mediante un diálogo asincrónico.

Este tipo de interfaces, además de incrementar el poder representativo, también aumentó los costos y la complejidad para gerenciar este tipo de representaciones.

Como una clase específica de las interfaces visuales, se encuentran las "Interfaces Icónicas", que utilizan como medio de interacción y representación visual, exclusivamente al ícono. Se puede mencionar, que el ícono es una imagen, una figura, pero que tiene un significado o semántica subyacente. El mismo está determinado por una imagen que debe ser significativa y fácilmente reconocible por la comunidad de usuarios.

Tanto las "Interfaces Visuales" como las "Icónicas", serán desarrolladas en profundidad, en los Capítulos III y IV, respectivamente.

También, se puede mencionar otro tipo de interfaces que son clasificadas como "Interfaces Inteligentes", puesto que permiten que el comportamiento de la interfaz se acerque aún más al usuario, proveyendo capacidad de razonamiento, de adquisición y aplicación de conocimiento y de comunicación de ideas.

Dentro de las "Interfaces Inteligentes", se pueden citar a las "Interfaces con signos de Adaptación", "Interfaces Evolutivas" e "Interfaces con Inferencia", que presentan características disímiles respectos a sus objetivos particulares, como a su cualidad de "inteligentes".

Las "Interfaces con signos de Adaptación", brindan diferentes modos de interacción que se pueden seleccionar automáticamente de acuerdo al tipo de usuario en cuestión. Son sensibles a los perfiles individuales de los usuarios y a sus estilos de interacción.

Las "Interfaces Evolutivas", tienen la propiedad de cambiar y evolucionar con el tiempo junto con el grado de perfeccionamiento que el usuario particular va adquiriendo con el sistema. Pueden acompañar la evolución o el crecimiento que presenta el usuario ante el uso del sistema, con nuevas ayudas, mensajes más específicos, un estilo de interacción más ágil, entre otras cuestiones.

Las "Interfaces con Inferencia", tienen la capacidad de captar secuencias de acciones que el usuario repite con frecuencia. Una vez registrado esa costumbre y ante la próxima iniciativa del usuario de realizarla nuevamente, el sistema se le adelanta y brinda la posibilidad de completar la secuencia de acciones en forma automática.

La automatización de este tipo de interfaces, se llevaría a cabo mediante el reconocimiento y almacenamiento

de patrones observados en el comportamiento de cada usuario. Esta adquisición de información es transparente para el usuario y conforma una base de conocimiento especializado que, junto con parámetros o criterios de evaluación y de los métodos apropiados para aplicarlo, constituyen un marco apropiado para que la interfaz provea signos de inteligencia, simplificando la labor del usuario.

También, se encuentran las interfaces presentes en los sitios de la World Wide Web, que debido al espectro inmensurable de usuarios al que está dirigida, debe tener consideraciones de diseño y desarrollo especiales. La misma va a ser explicada en detalle, en el Capítulo V.

Existen otras interfaces importantes, como las “Interfaces Accesibles”, “Interfaces para Groupware”, “Interfaces puramente Conversacionales” y las “Interfaces Móviles”.

Las “Interfaces Accesibles”, son aquellas que respetan las normas del diseño universal, para que pueda ser accedida por cualquier usuario, independientemente de sus condiciones físicas o mentales. Se encuentra el Capítulo VI, en donde se va a explayar más este tema.

Las “Interfaces para Groupware”, se caracterizan por interactuar con un grupo de usuarios que tendrán objetivos comunes, recursos a compartir, un ambiente virtual de reunión. La misma deberá tener facultades para solventar la coordinación del grupo, como aspectos de colaboración y comunicación entre sus miembros.

Las “Interfaces puramente Conversacionales”, se montan sobre los sistemas telefónicos y se caracterizan por no contar con una pantalla. El único medio de salida es la voz y, la entrada puede ser a través de la voz, como también, a través de los botones del dispositivo telefónico.

Las “Interfaces Móviles”, son aquellas presentes en los dispositivos inalámbricos como celulares y PALMs, que los usuarios utilizan para acceder a la aplicación. Cuestiones de interrupción, distracción, visualización de una pantalla muy reducida, el apremio por acceder a la información en ese momento y en ese lugar, hacen que la interfaz presente características de diseño muy especiales.

Por ultimo, se pueden citar otras interfaces no tan convencionales como las interfaces hápticas y las multimodales. Las primeras, utilizan dispositivos que permiten al usuario emplear todos sus sentidos. Ellos pueden tocar, oler, oir, sentir o manipular objetos simulados en entornos virtuales y sistemas teleoperados. Además de la sensación de inmersión estas interfáces proporcionan al usuario la posibilidad de interactuar con el medio virtual logrando una transferencia bidimensional.

Las interfaces multimodales intentan adaptar el sistema al usuario, combinando distintas técnicas de entrada y salida, en conjunto con características de interfaces tangibles donde objetos reales del entorno del usuario son convertidos en elementos de interacción digital.

Vale aclarar, que un sistema interactivo puede proveer una interfaz del usuario que posea algunas características de las clases de interfaces definidas anteriormente, y las combine, generando como resultado, una interfaz híbrida.

Recordar

Las Interfaces del usuario se pueden clasificar en:

- Interfaces Textuales, Gráficas e Icónicas
- Interfaces Inteligentes
- Interfaces para la Web y Accesibles
- Interfaces para Groupware
- Interfaces puramente Conversacionales
- Interfaces Móviles, Interfaces Hápticas y multimodales

V - Independencia del Diálogo

La introducción de un sistema informático, siempre implica cambios y modificaciones a la forma de trabajo, los usuarios pueden emplear el tiempo que le resta de realizar sus actividades en forma automatizada, en otros labores, o en distintas tareas adicionales.

La interfaz de un sistema, tiene que lograr potenciar los aspectos que sean pertinentes para cada trabajo, ya sea para simplificar el tipeo, infiriendo lo que se está ingresando, para evitar la invocación errónea de un comando complejo, utilizando buenos mensajes de aclaración y confirmación, entre otras cuestiones.

El concepto es que la introducción de una solución informática, cambia la realidad y que siempre tendrá asociado modificaciones más o menos significativas, que impliquen fundamentalmente adaptar la interfaz del usuario, ya sea el “look & feel” o cambios para mejorar la productividad y eficacia de la interacción con el mismo.

Pero, al principio, esta visión del sistema interactivo no se podía aplicar. La forma tradicional de desarrollar sistemas de software, implicaba que las cuestiones de interfaz y de diálogo con el usuario, estaban fuertemente ligadas con el software computacional. Las sentencias de entrada y salida al usuario, eran tratadas sin distinción dentro del resto del código, podían estar inmersas y entremezcladas dentro de sentencias de control, causando un fuerte acoplamiento entre el flujo de diálogo con el flujo de control del sistema.

Si las cuestiones de interacción con seres humanos, se encuentran inmersas y ocultas dentro de la componente computacional del sistema, esto trae como consecuencia, una resistencia muy fuerte al desarrollo de interfaces del usuario de alta calidad. Esto se debe a que se hace complicado su identificación y tratamiento, el diálogo se convierte en una componente cada vez más resistente a cambios, no se puede profundizar en las necesidades de adaptación a cuestiones del usuario, y principalmente, no permite que la interfaz cumpla con un ciclo de desarrollo especial que la caracteriza, con etapas de diseño, prototipación y evaluación, que se suceden en forma iterativa, a través de un marco donde las correcciones, modificaciones y refinamientos son una constante.

En este contexto, H. Rex Hartson, Roger W. Ehrich, en el año 1981, establecieron una definición formal sobre el concepto de "Independencia de Diálogo", en donde especificaron que un sistema interactivo, debía conformarse fundamentalmente, de dos componentes de software igualmente importantes e independientes, denominadas interfaz del usuario y componente de aplicación, o componente de diálogo y componente de cómputos respectivamente, que deben ser tratadas en forma autónoma, desde el inicio del desarrollo del sistema [Hartson, R. y Hix, D., 1989]

Esta definición, se basa en el reconocimiento de las cuestiones de interacción como una componente de software que debe ser tratada en forma especial y separada del resto del sistema, durante todo el proceso de desarrollo.

La idea de partida, es la definición de una estructura del sistema, en donde se evite la modelización conjunta e intercalada, entre la lógica de los procesos y las operaciones de entrada y salida, desde y hacia el usuario.

En la práctica, esto significa que la componente computacional o los procesos referenciados más recientemente como la lógica del negocio, se desentienden de todo lo que abarca los aspectos de la interfaz, o sea de cómo es su estilo de interacción, cómo son las cuestiones del diálogo y los hilos posibles de conversación, si provee un lenguaje de comandos para comunicarse con el usuario final o si está basada en menús o formularios. Se despreocupa de cómo la entrada es adquirida desde el usuario y de la forma en que le son visualizados los resultados. Directamente focaliza su interés específicamente en el procesamiento de la información.

La independencia del diálogo presenta principalmente dos grandes ventajas. Una, está íntimamente relacionada con la Ingeniería de Software, debido a que esta separación es crucial tanto para modificar fácilmente la interfaz en su proceso iterativo de refinamiento como para su posterior mantenimiento.

Y, otra, es a nivel de factores humanos, debido a que constituye una estrategia en donde las cuestiones específicas del usuario, como sus preferencias, hábitos, costumbres, manera de ser, características de sexo o edad, experiencia particular, entre otros, pueden ser tratadas y evaluadas con énfasis y en forma especial. Esto es, permite diseñar el diálogo de una manera que la computadora se adapte a los requerimientos del usuario, convirtiéndose en una herramienta que se amolda al estilo particular del mismo.

Desde esta perspectiva, el diálogo podrá conjugar la simplicidad de interacción con la optimización en el proceso de validación, proveyendo feedback y ayudas eficientes, para asistir al usuario en su trabajo y lo potencie, guiándolo en su proceso de resolución de problemas.

En este sentido, la Independencia de diálogo da lugar al tratamiento, estudio y evaluación de aspectos del usuario, vinculados a la experiencia, motivación, edad, sexo, formación, capacidades, suministrándole la relevancia necesaria, para poder encarar y desarrollar correctamente la componente de interfaz.

Recordar

❒Independencia de Diálogo es una estrategia que permite que las decisiones de diseño que afectan únicamente a la componente de diálogo, se realicen en forma independiente de aquellas que afectan a la estructura del sistema de aplicación y al corazón funcional.

¿Cómo se logra esta estrategia?

La independencia del diálogo no se implementa simplemente colocando buenos mensajes de error en uno o dos procedimientos, ni tampoco apartando todo lo relacionado con las entradas y salidas, en módulos independientes, separados del resto de las rutinas de cómputo. Esto significa una buena práctica de programación, pero no asegura la independencia del diálogo, debido a que los procedimientos orientados al diálogo, deben estar totalmente desacoplados del código computacional y debe garantizar su modificación en forma independiente del resto del programa. Aún hoy, en muchos casos, el conocimiento relacionado con los detalles del diálogo siguen inmersos dentro del código de la aplicación, y la tarea de desarrollar la interfaz, continúa erróneamente ligada a una tarea de programación.

La independencia del diálogo se soporta a través de la separación del diálogo y el software de la aplicación desde las etapas iniciales del desarrollo. Esto significa que un sistema de aplicación interactivo, se compone de una componente de diálogo, a través de la cual se lleva a cabo toda la comunicación entre el usuario final y el sistema, y una componente computacional, que incluye los mecanismos de procesamiento funcional más específicos del sistema de aplicación, ámbito en donde no hay una interacción directa con el usuario final.

La nueva entidad, el protagonismo y la visibilidad que toma la componente de diálogo en el desarrollo de sistemas, implica un proceso de diseño, rediseño, evaluación y mantenimiento, particular e independiente al de la componente funcional, ligándose en el momento que se requiere una rápida prototipación o ejecución del sistema total.

Como principales consecuencias de la Independencia del Diálogo, se pueden mencionar la aparición de un nuevo rol, que es la del desarrollador del diálogo, quién se va a responsabilizar de todo el proceso de desarrollo de la interfaz, y el surgimiento de un diálogo interno entre la componente de interfaz y de la aplicación, que es el que permitirá el enganche final de ambas partes, conformando el sistema interactivo en su completitud.

La aparición de un nuevo rol: El desarrollador de diálogo

Por mucho tiempo, los dos principales roles involucrados en el desarrollo del software era el programador de la aplicación y el usuario final del sistema.

Estos dos roles diferentes frecuentemente tenían graves problemas en la comunicación. El programador impaciente por llegar a la codificación tenía dificultades en entender necesidades del usuario. Similarmente, el usuario muchas veces no logra comprender los costos o grado de dificultad del desarrollo de determinados requerimientos para el sistema, ni explicaciones técnicas sobre determinado comportamiento efectuado por el mismo.

La aparición del rol de analista de sistemas, proveyó un medio de entendimiento entre el programador, inmerso en un contexto completamente técnico y el del usuario del sistema.

En los últimos tiempos, ante la demanda de interfaces del usuario de alta calidad y cada vez más efectivas, la aparición de especialistas en factores humanos fueron adquiriendo una significante importancia en los equipos de desarrollo de los sistemas de computación, puesto que focalizan su atención y estudio a las necesidades de los usuarios y a la forma de comunicación con los mismos.

Esto llevó al surgimiento de un nuevo rol, denominado desarrollador del diálogo ó también llamado autor, diseñador del diálogo, ó ingeniero de la interfaz del usuario.

El desarrollador del diálogo, es un especialista en factores humanos que se interesa y se encarga de todas las etapas del desarrollo del software de la interfaz del usuario, abarcando el diseño, la implementación y evaluación de la forma, contenido, estilo y secuenciación del diálogo. El está involucrado en todo el ciclo de vida, incluyendo análisis de tareas, especificación de requerimientos del sistema, investigación y aplicación de normas de diseño, cuestiones de factibilidad del desarrollo, dirección de todo el proceso prototípico de la interfaz.

Durante el diseño e implementación del diálogo, este desarrollador hace uso de principios psicológicos y de factores humanos para construir y, paralelamente, evaluar y refinar una interfaz que soporte una comunicación efectiva y simple, entre el ser humano y la máquina.

Debe ser muy sensitivo frente a los requerimientos cognitivos del usuario final y a sus aspectos particulares, debe profundizar por ejemplo situaciones como:

- Formas particulares en que se reacciona a los distintos colores, según diferentes sociedades. Este es el caso del color rojo que en países orientales no está asociado a situaciones de riesgo;
- Los diferentes hábitos de lectura, que puede afectar en el orden de recorrido de la pantalla y posicionamiento de los objetos en la misma;
- Las distintas actividades que implican diferentes posturas y costumbres que condicionan el estilo de la interacción. Por ejemplo, si es mejor la entrada por voz o por teclado, en un entorno industrial o en una carga masiva de datos;
- La postura, el nivel de atención, los horarios de trabajo, afectan interfaces distintas que deben apuntar a mejorar la productividad del usuario;
- La edad y nivel cultural del usuario, que lleva a la utilización de distintas metáforas para construir la interfaz, donde su interpretación depende de grados de madurez y aspectos sociales del mismo.

La Independencia de diálogo, define un marco en donde estas cuestiones como muchas otras más, pueden ser profundizadas y consideradas dentro del diseño de la interfaz, permitiendo el refinamiento y revisión de las mismas, su comprobación y la realización de modificaciones rápidas, por lo que el proceso de evaluación y el ciclo de revisión se puede efectuar múltiples veces.

Recordar

El desarrollador del diálogo tiene el rol y la responsabilidad de:

- ❒Efectuar los modelos necesarios sobre los usuarios, el contexto, la aplicación y aspectos tecnológicos.
- ❒Efectuar y converger al mejor diseño de la interfaz de acuerdo a los factores humanos analizados.
- ❒Evaluar y refinar la interfaz según su impacto directo sobre los usuarios.

Una nueva interfaz. Diálogo interno y externo

El diálogo externo es aquel que abarca toda la interacción entre el usuario final y la componente del diálogo, es la misma interfaz hombre-computadora.

Pero, la Independencia del Diálogo ha provocado la creación de una nueva interfaz que toma lugar entre la componente del diálogo y la computacional, y en consecuencia, esto trae aparejado la aparición de un nuevo tipo de diálogo entre ambas.

La componente funcional que no tiene mecanismos propios para la comunicación directa con el usuario final, requiere de un diálogo interno necesario con la componente de diálogo.

Este diálogo, se basa principalmente en la definición concreta del intercambio de información entre ambas componentes, los tipos, formatos de datos que una espera de la otra, si van a estar ya validados o no, cuáles son las condiciones estipuladas, cuestiones de excepciones o errores que pueden surgir, quién los trata, como también en qué forma y estructura, se entregarán los resultados entre las partes. Hay que estipular claramente, cuáles serán los servicios y las funciones que una componente necesita de la otra y, las cuestiones protocolares y de sincronización necesarias.

Esta nueva interfaz interna, con su diálogo específico, son las bases para la comunicación entre el desarrollador del diálogo y el del software de la aplicación durante el proceso de diseño del sistema y es un punto crucial para llevar a cabo la ligazón del diálogo y el cómputo en los tiempos de ejecución.

El diálogo interno, y su representación formal en la etapa de diseño constituye una clave esencial para lograr la Independencia del Diálogo. Tanto la interfaz del usuario o la aplicación, pueden ser alteradas en forma independiente siempre y cuando, se mantenga la consistencia en la representación del diálogo interno común a ambas.

Complejidades de la Independencia de Diálogo

Lamentablemente la estrategia de la Independencia de diálogo, no está exenta de inconvenientes y dificultades.

Uno de los problemas principales que se le atribuye, es que debido a que se cuenta con un nuevo rol para el desarrollo de la componente de diálogo, aumenta el grado de comunicación e interacción entre los diseñadores y los programadores, sumando problemas de ambigüedades, malas interpretaciones, deslinde de responsabilidades, pugna por toma de decisiones, entre otros inconvenientes.

Es necesario para solucionar esto, fijar claramente desde el inicio, los roles que van a intervenir, especificando puntualmente las responsabilidades de cada uno y sobre qué componente del sistema van a actuar. Como también, la definición precisa sobre el diálogo interno, estipulando cuáles son exactamente los datos, de qué tipo y formato es la información que fluirá entre la componente de interfaz y la de la aplicación, como así los servicios o funciones que se necesitan de cada una.

Otro inconveniente provocado por la Independencia del Diálogo, es que, debido a la existencia de nuevas componentes y módulos, puede causar una disminución en la performance total del sistema, a medida que aumenta la intercomunicación entre las mismas, en tiempo de ejecución. Esto lleva a la consideración de nuevas arquitecturas del sistema que enfatice y facilite por ejemplo, la ejecución concurrente entre la componente de diálogo y la computacional, y considere hardware innovador para el soporte del diálogo.

En el plano de un diálogo secuencial, la separación física entre la componente de interacción y la computacional, desde la etapa de diseño es medianamente directa. Esto se debe a que resulta fácil delinear un patrón de interacción sincrónico que considera las etapas de entrada de datos, requerida al usuario final, del proceso de cómputo de los dichos datos y, de la salida resultante provista por el sistema.

En cambio, en un diálogo con manipulación directa los costos para lograr la separación entre las dos componentes es mucho más alto, porque el usuario final realiza operaciones y acciones en la interfaz sobre las representaciones visuales de los objetos de la aplicación de una manera directa, visual y asincrónica. En este caso, la ejecución del diálogo y del cómputo tienden a estar más interrelacionados y generalmente se comparte representaciones de datos comunes de los objetos de la interfaz y de los de la aplicación subyacente.

En un diálogo con manipulación directa, hay una necesidad de acercamiento entre la interfaz y la semántica de la aplicación, por ejemplo para un simple caso de proveer feedback semántico, y esto hace más difícil la separación de las componentes requerida en la Independencia de diálogo. Esto fuerza a importantes cambios en la arquitectura de los sistemas interactivos.

De todas maneras, a menor o mayor costo, se puede lograr que las decisiones de diseño incluyendo tanto la apariencia como el comportamiento de la interfaz, puedan mantenerse independiente del software que manipula las correspondientes estructuras de datos, y en todos los casos, se comprueba que las ventajas de mantener la independencia del diálogo son mayores que sus posibles dificultades.

VI - Componentes de la Interfaz del Usuario

Uno de los modelos más conocidos sobre la componente de la interfaz del usuario, es el Modelo Seeheim o Seeheim Model, definido en el marco de un Workshop efectuado en Seeheim, Alemania, en el año 1982.

Este modelo, es una descripción estructural que especifica la manera en que la interfaz del usuario se relaciona con el sistema de aplicación, puntualizando la independencia entre ambas partes del sistema. Además, describe a la interfaz, mediante dos componentes: la Componente de Presentación, que conforma su parte estática, y la de Control de Diálogo, que describe la componente dinámica de la misma.

Más tarde, en 1985, el modelo fue modificado por Mark Green, perteneciente a la Universidad de Alberta, donde le introdujo una tercer componente denominada Modelo Interfaz-Aplicación, para referenciar el diálogo interno con la aplicación [Green, M.,1985].

La versión final del modelo, quedó de la siguiente manera:

Figura 1.3: Modelo de Seeheim, Versión modificada por M.Green

La componente de presentación es la responsable de la visualización de la interfaz, especifica las pantallas que se presentarán al usuario y el estilo de interacción. Los colores, ubicaciones de la pantalla, figuras, menús son conceptos relacionados con esta componente. En ella, se definen los siguientes aspectos:

- Se especifica el aspecto y apariencia de la interfaz. O sea, se define la parte del “look”, del concepto de look&feel de la interfaz.
- Se define el estilo de interacción, tipos de diálogo y de interfaz.
- Se identifican los objetos de interacción.
- Se establece el esquema de ventanas.
- Se definen los dispositivos de hardware, asociados a la entrada y salida, a incorporar en la interfaz.

La componente de Control de Diálogo, está a cargo del manejo de la secuencia de eventos y del control de las técnicas de interacción provistas en la interfaz. Establece y define la parte dinámica de la misma. En esta componente, se establecen fundamentalmente, las siguientes cuestiones:

- Se especifica el comportamiento de la interfaz. O sea, la parte del “feel”, del concepto de look&feel de la interfaz.
- Se define el manejo, control, secuenciación y lógica del diálogo.
- Se controlan y se manejan los objetos de interacción presentes en la interfaz.
- Se controla y se maneja el sistema de ventanas.
- Se definen aspectos de navegación e interacción entre las diferentes ventanas o páginas que conformen el sistema.
- Se establece la estructura, secuencia, el proceso del intercambio de información entre el usuario y la aplicación.
- Se identifican, controlan y manejan los objetos computacionales necesarios a nivel de interfaz del usuario, tales como objetos sintácticos.
- Se definen los servicios propios de interfaz del usuario, o sea funciones sintácticas.
- Se controla y se administra el hardware interactivo que incluye la interfaz.

- Se especifica el tipo de asistencia, sistemas de búsqueda, tipo de feedback, que la interfaz va a proveer.
- Se especifican las cuestiones de avanzada a incorporar dentro de la interfaz, como ser aspectos de adaptación, inteligencia, colaboración que deben ser manejados y controlados a través de esta componente.

Finalmente, la componente denominada Modelo de Interfaz-Aplicación es la representación de la aplicación desde el punto de vista de la interfaz. Contiene tanto una visión de la aplicación desde la interfaz como una visión de la interfaz desde la aplicación. Incluye lo siguiente:

- Especifica el diálogo interno entre la interfaz y la aplicación.
- Permite por un lado la abstracción de la forma en que el usuario consulta y provee información a la aplicación y, por otro, independiza las acciones de la aplicación de las implementaciones particulares de las entradas y salidas del sistema.
- Especifica la relación entre los eventos de la interfaz y las funcionalidades de la aplicación.
- Establece la forma de comunicación entre ambas partes. Puede ser vía llamados a procedimientos o métodos, o mediante el uso de estructuras de datos comunes.

VII - Métricas de Evaluación de una Interfaz

Gracias a la Independencia de diálogo, filosofía que establece una separación entre la componente de interacción o interfaz del usuario con la componente de cómputos o aplicación, se puede profundizar más específicamente sobre las normas de diseño y objetivos preliminares que son propios a cada componente.

En el área especial de HCI, existen objetivos particulares que el desarrollador debe tener en cuenta, como así también principios básicos de diseño de interfaz que deben ser incluidos, si se pretende mejorar la calidad de la interacción con el sistema.

Estos aspectos no sólo deben ser considerados en el momento de diseñar una interfaz del usuario sino que además pueden ser utilizados en los procesos evaluativos de la misma, durante su desarrollo o al final, una vez que ya se encuentra en un estado productivo.

Objetivos de la Interfaz del Usuario

Mientras que dentro de la componente de aplicación, se consideran objetivos tales como eficiencia, portabilidad, legibilidad, entre otros aspectos, la interfaz del usuario presenta otros objetivos especiales que están íntimamente relacionados con los factores humanos más que con cuestiones técnicas o requerimientos impuestos por el sistema.

Por ser una componente que dialoga con seres humanos, se le exige cualidades como simpleza, amigabilidad, naturalidad, flexibilidad y otras características que afectan el grado de utilidad de la interfaz y por consiguiente la productividad general del resto del sistema interactivo.

La conjunción de todas las cualidades exigidas a la componente de diálogo es sintetizada con el término de “Usabilidad”. Para referenciar a este término se utilizó en la década del 90’ el término «Amigabilidad», y actualmente en la bibliografía más reciente, se utiliza «Calidad de Uso».

El estándar ISO/IEC 9241 define a la usabilidad como la efectividad, eficiencia y satisfacción con la que un producto alcanza sus objetivos específicos para los usuarios. El ISO/IEC 9126 hace énfasis en las características internas y externas de las aplicaciones que actúan en la usabilidad.

Los objetivos mencionados en este apartado, como los principios que se presentan en las próximas secciones, apuntan a mejorar la calidad de las interfaces provistas por los sistemas interactivos en pos de lograr un alto grado de satisfacción y conformidad por parte del cliente o usuario. Llevando, como consecuencia, a una segura productividad y extensa vida útil del software, finalidad que todo desarrollador de software desearía alcanzar.

Entre los principales objetivos tenemos:

A. Simplicidad:

El sistema interactivo debe ser simple de instalar, de aprender, de usar, de configurar.

Al usuario no se le puede exigir un entrenamiento sobre Computación para poder utilizar el sistema, ni que se lo someta a posteriores cursos de aprendizaje para interactuar con la interfaz.

El sistema interactivo debe proveer un diálogo natural, un ambiente amigable donde el usuario pueda encontrar toda la asistencia necesaria para resolver su problema y llevar a cabo sus intenciones.

Esto satisface a la gran demanda por parte de los usuarios de contar con sistemas de software que realmente funcionen como herramientas destinadas a simplificar el trabajo cotidiano, no a complicarlo aún más con requerimientos físicos del sistema.

b. Confiabilidad:

Este objetivo hace referencia a que la interfaz del usuario debe ser consistente tanto en su forma de expresión como de presentación.

No puede contener diálogos ambiguos, que lleven a falsas interpretaciones o que genere dudas al usuario. Tampoco puede existir desorganización en la visualización de los datos ni heterogeneidad en los mecanismos de utilización de los servicios provistos.

El usuario debe interactuar con un ambiente integral, que presente un diseño coherente, estudiado, con un tratamiento similar en todos sus contextos y transacciones.

Esto ayuda a que los usuarios perciban una interfaz seria, cuidadosamente diseñada, en la que puedan confiar sus datos y delegar el manejo de los mismos.

c. Flexibilidad:

La interfaz del usuario debe comportarse de una manera maleable, dócil, donde el usuario pueda concretar sus intenciones en forma autónoma.

La misma debe ser tolerante a un cierto grado de error que pueda presentar el usuario y condescendiente ante equivocaciones frecuentes efectuadas por él. Puede ayudar brindando mecanismos de sugerencias o de corrección automática, más que informarle un mensaje de error.

No puede comportarse como un agente interlocutor controlador, rígido e inflexible, debe ser abierta y aceptar los tiempos y las formas que el usuario requiera.

Debe permitir que el usuario sea quien decida cuándo comenzar un diálogo y cuándo culminarlo, y sea quien dirija la conversación.

d. Transparencia:

La componente de interfaz no puede actuar como una caja negra, donde el usuario únicamente ingrese los datos y espere los resultados.

En todo momento, la interfaz debe comunicarle al usuario sobre los detalles de la transacción y de los cómputos realizados, debe indicar el porcentaje de realización, índices del progreso e información del tiempo que falta para culminar la función que esté ejecutando.

También debe proveer mecanismos para poder interrumpir, suspender o deshacer lo efectuado por el sistema y alcanzar un estado previo deseado por el usuario.

Esto hace que la interfaz se comporte como un agente abierto, comunicador, que se manifiesta permanentemente, que informa sobre su accionar ante el usuario.

e. Ergonomía:

Este objetivo se refiere a la capacidad que presente la interfaz para amoldarse, adaptarse al estilo propio del usuario, brindando mecanismos de configuración y personalización.

La misma puede estudiar al usuario mientras está sesionando con el sistema en pos de adecuarse a su forma de interacción o también se le puede incorporar modelos computacionales con información sobre los perfiles de los usuarios, previamente investigada en las etapas preliminares del desarrollo.

De esta manera, la interfaz es percibida como un agente inteligente, que conoce al usuario, sus preferencias, sus hábitos y se acondiciona adecuadamente a él.

Recordar

La interfaz del usuario debe cumplir con los objetivos de:

- ❒Simplicidad
- ❒Confiabilidad
- ❒Flexibilidad
- ❒Transparencia
- ❒Ergonomía

La suma de todas estas cualidades genera una interfaz usable, con un alto porcentaje de productividad y principalmente con un elevado grado de satisfacción por parte de los usuarios.

Principios de Nielsen

En este punto, se van a considerar los “Principios de Nielsen”. Consisten en un conjunto de diez reglas de diseño, que estipulan la forma correcta de diseñar el diálogo provisto en una interfaz del usuario, teniendo en cuenta aspectos de usabilidad.

Estos principios, considerados también como “heurísticas”, fueron definidos por Jakob Nielsen, en el año 1990, junto con la colaboración de Rolf Molich [Nielsen, J. y Mack, R., 1994].

Aunque estas normas inicialmente fueron estipuladas para interfaces textuales, sirven de base para el diseño preliminar de cualquier otro tipo de interfaz, desde la más sencilla a la más compleja.

Los principios de Nielsen, se puntualizan de la siguiente manera:

Diálogo simple y natural

En este punto se estipula la forma en que la interacción debe llevarse a cabo. El diálogo presente en los prompts o preguntas del sistema, en los títulos, en los contenidos de las pantallas, en la solicitud de las entradas, como en la muestra de los resultados.

Específicamente, hay que tener en cuenta las siguientes reglas:

Recomendaciones	Explicación
Evitar abuso de abreviaturas	■Es desprolijo que el usuario vea permanentemente sobre las pantallas del software, palabras abreviadas como “cód.”, “impr.”,”art.”, que hacen referencia respectivamente a código, imprimir y artículo.
Evitar uso excesivo de mayúsculas	■Hay estudios realizados en donde demostraron que el diálogo expresado en letras mayúsculas se asemeja al hablar en voz muy alta, como gritando. ■Además de ser intimidatorio, también se le atribuye dificultad en la lectura y por consiguiente en la futura interpretación del texto. ■Si se emplea todo el texto en mayúsculas con la intención de demostrar importancia, en realidad lo que se está provocando es intimidación y complejidad visual.
Evitar errores de tipeo	■El desarrollador del sistema no puede dejar plasmado en las pantallas del software, errores de ortografía, errores gramaticales ni aquellos cometidos por un mal tipeo. ■Esto da la idea de que el sistema no fue revisado, o que no se le dio importancia a su desarrollo.
Distribuir bien la información	■Debe planificarse el diseño general de las pantallas de un sistema. ■Las pantallas deben tener un marco en común, con zonas bien diferenciadas para los títulos, contenidos, comandos, información, mensajes de error y demás componentes de diálogo. ■La consistencia en el diseño visual de estos elementos, reduce la posibilidad de errores de interpretación y facilita encontrar la información.
Analizar la información relevante y evitar la innecesaria	■Los diálogos deberían contener únicamente la información más importante. ■Los datos innecesarios que se encuentran en un diálogo, compiten de alguna manera con la información relevante y esto provoca la disminución de su visibilidad.
El control del diálogo debe estar dirigido por el usuario	■Se debe evitar que el usuario en su interacción tenga que someterse a esquemas de diálogo absurdos exigidos por el sistema. ■Por ejemplo, que el usuario tenga que “apretar dos enter y un escape” para salir de una situación determinada. ■El usuario debe ser dueño de la interacción, debe poder iniciar el diálogo, cancelarlo, reiniciarlo según sus tiempos y necesidades.
Emplear estilos de prompts simples	■Evitar prompts o preguntas del sistema que incluyan frases complejas con operadores lógicos como “o”, “y”, “no”, que dificultan la tarea interpretativa del usuario. ■Por ejemplo, evitar la siguiente frase “Salir sin Guardar”
Unificar el empleo de las funciones predefinidas	■Las teclas funcionales como F1, F2,... o combinaciones de teclas, deben estar utilizadas en forma homogénea en todo el sistema y en toda circunstancia.

Tabla 1.2: Recomendaciones de “Diálogo Simple y Natural”.

Emplear el lenguaje del usuario

Con este principio, se recomienda al diseñador de la interfaz, no trasladar su propio lenguaje, dialecto y costumbres dentro de la componente de interacción. La misma debe utilizar términos, frases o formas de diálogo, que sean familiares al ó a los usuarios del sistema.

Por lo tanto, se recomienda los siguientes puntos:

Recomendaciones	Explicación
Utilizar el lenguaje, idioma, dialecto y jerga según el usuario	■Es necesario emplear el dialecto de las/los usuarios, de acuerdo a su formación, costumbres, hábitos y preferencias.
Evitar uso de lenguaje extranjero	■Palabras como “exit”, “logout”, “loggueate”, no deben ser empleadas.
Evitar uso de lenguaje técnico	■Hay que evitar palabras como “indexar”, “campo”, “registro”, que en el marco del usuario puede tener connotaciones diferentes. ■El diálogo debe referirse a conceptos entendidos por el usuario, más que a la implementación de la operación del sistema.
Evitar el uso del lenguaje regional por parte del programador	■Expresiones como “ché” o “vos” o palabras porteñas[1], cuyo significado puede ser desconocido por algunos usuarios.
Evitar el truncamiento de palabras	■Toda la información que sea cargada por el usuario, debe verse a lo largo de todo el sistema de la misma manera. ■Por ejemplo, al cargar un dato con una longitud permitida de 30 caracteres no debería luego verse truncado a 10.
Permitir al usuario, configurar el grado o cantidad de información que se maneja.	■No se debe utilizar excesivo detalle para experimentados como tampoco escasa información para principiantes. La misma debe ser balanceada.
Permitir cargas parciales de datos	■El orden de ingreso de la información en un formulario debe ser libre. ■El usuario debería poder comenzar a completar un formulario desde cualquier punto, como también tener la posibilidad de suspenderlo, seguirlo después. ■No debe presentar restricciones o condicionamientos sin sentido.

Tabla 1.3: Recomendaciones sobre “Emplear el Lenguaje del Usuario”

Minimizar el uso de la memoria del usuario

En este punto, se pretende evitar que el usuario esfuerce su memoria para interactuar con el sistema.

Esto significa que el usuario no debería recurrir a carpetas manuales para recordar algún código o número de artículo solicitado obligatoriamente por el sistema, ni acudir a un tercero, para obtener asistencia respecto al uso del sistema. Esto implicaría un fracaso para el desarrollador de la interfaz.

Para evitar esto, se debe seguir las siguientes recomendaciones:

Recomendaciones	Explicación
Brindar información de contexto	■Se debe indicar en todo contexto, en dónde se encuentra el usuario, desde dónde vino y hacia dónde puede ir. ■Se puede además mostrar datos importantes cargados en pantallas previas, si es que el usuario debe continuar interactuando con datos relacionados.

[1] Palabras porteñas: se refiere a términos y jerga utilizada en la zona del puerto de Buenos Aires, Capital Federal, Argentina.

Brindar información de la navegación y de la sesión actual	■Se debe mostrar información de la sesión, cuáles fueron las últimas actividades, por qué sectores estuvo trabajando. Así podrá tener presente información sobre la navegación y la interacción efectuada. ■Saber por dónde se llegó a un determinado punto de la interacción, es importante por si se desea volver, revisar puntos anteriores o deshacer pasos. ■El usuario puede ser interrumpido, por ejemplo por teléfono y debe contar con la información necesaria para continuar desde el mismo punto, sin perder datos ni tiempo.
Visualizar los rangos de entrada admisibles por el sistema	■Mostrar en los campos de entrada, los posibles datos que se pueden ingresar y que ya son admisibles por el sistema, es una ayuda importante para el usuario ya que se evitaría el tipeo de los mismos. ■Brindar ejemplos, formatos de entrada esperados, datos por defecto no sólo ayudan al usuario a evitar errores, sino que además le enseñan cómo son los mecanismos de diálogo requeridos.
Brindar mecanismos de asistencia y búsqueda	■El usuario no debería por ningún motivo, acceder a ayuda extra a la otorgada por el sistema o inclusive, utilizar medios manuales para buscar cierta información requerida por el software. ■El sistema debe brindar todos los mecanismos de asistencia y medios de búsqueda necesarios para que el usuario pueda encontrar la información que no recuerda, en forma simple. ■Es importante que el usuario sienta que el sistema está bajo su control y no a la inversa.

Tabla 1.4: Recomendaciones sobre “Minimizar el uso de la memoria del usuario

Consistencia

La consistencia es un punto clave para ofrecer confiabilidad y seguridad al sistema. Se refiere a que tanto el diálogo, el aspecto visual, el aspecto terminológico, el comportamiento del sistema, se presenten ante el usuario en forma homogénea y consistente, sin ambigüedades.

Los usuarios, no deberían tener que adivinar si diferentes palabras, situaciones o acciones, significan lo mismo en distintos contextos. O, por el contrario, que un mismo término adquiera diferentes interpretaciones, a lo largo del sistema. Todas estas situaciones hacen que la interfaz se considere inconsistente, el usuario se desconcierta y no sabe qué actitud tomar.

El poder garantizar consistencia, adquiere mayor importancia, en sistemas extensos o, cuando son desarrollados por distintas personas y en distintos momentos.

A continuación, se mostrará la tabla de recomendaciones respectivas a este punto:

Recomendaciones	Explicación
Debe existir una consistencia terminológica	■El concepto de una función, comando, tecla o botón, debe ser siempre el mismo en todo el sistema. ■Siempre se debe utilizar el mismo término, frase o imagen para un determinado concepto y viceversa. ■Debe conformar una relación 1 a 1, entre la representación y el significado de cada objeto o función. ■De esta manera se evitan falsas interpretaciones.
Debe existir una consistencia visual	■En todas las pantallas debe utilizarse los colores, espacios, distribución, tamaños, ubicaciones, sonidos y formatos en forma coherente y homogénea. ■El usuario no debe notar grandes diferencias visuales ni tratamientos disímiles entre los contextos de un mismo sistema.

Tabla 1.5: Recomendaciones sobre “Consistencia”

Feedback

En este principio, se apunta a que el sistema siempre debería mantener al usuario informado de lo que está sucediendo, más que nada cuando éste manipula los objetos directamente sobre la pantalla.

El mecanismo para lograr la comunicación desde el sistema y hacia el usuario, es utilizando una técnica llamada “feedback” o retroalimentación.

Cuando se provee “feedback”, la interfaz da una respuesta gráfica o textual sobre la pantalla, frente a una acción del usuario. El feedback se refiere al efecto o reacción del sistema, que se produce por alguna operación del usuario.

El feedback, debe diseñarse de acuerdo a las siguientes consignas:

Recomendaciones	Explicación
Brindar avisos informativos	■Se debe informar al usuario, sobre qué puede realizar en un determinado momento, sobre el grado de actualización de la información que se despliega, sobre el tamaño o volumen de datos con el que va a trabajar, entre otras cosas.
Brindar mensajes aclaratorios	■La interfaz del usuario debe servir también como un medio para aclarar, explicar, detallar sobre información o comportamiento esperados del usuario.
Brindar mensajes preventivos	■Al usuario se le debe advertir sobre posibles situaciones indeseables que puedan ocurrir. ❖También, se le debe informar, por ejemplo, tiempos de acciones que demanden varios minutos.
Dar mensajes de confirmación	■El usuario debe tener la posibilidad de confirmar cada transacción o actividad importante que haya realizado, o donde se maneje información sensible.
Brindar mensajes de cierre	■Se debe indicar cómo finalmente culminó una actividad efectuada por el usuario. ■Se debe indicar si una transacción fue realizada exitosamente o no.
Realizar validaciones	■El sistema debe validar la información ingresada por el usuario mientras él está interactuando en una manera natural. ■No sería correcto indicar que existen datos erróneos después de un largo período de tiempo de interacción o cuando el usuario ya se encuentra en otro contexto o haciendo otra cosa. ■Entonces, apenas se pueda dar retroalimentación al usuario, debe hacerse.
Dar información sobre los estados de los procesos	■Se debe informar sobre cómo va el proceso que el sistema está realizando, por dónde va y cuánto falta para culminar. ■Esto es importante en el caso de impresiones, actualizaciones, indexaciones o almacenamiento de gran volumen de información.
Dar información sobre el estado del sistema y del usuario	■Se debe indicar el estado en que se encuentra el usuario o el sistema. Por ejemplo, si está realizando cierta labor o esperando que se corrija un error.

Tabla 1.6: Recomendaciones sobre “Feedback”

Salidas evidentes

Este principio, se refiere a que el usuario en todo momento tenga al alcance una opción de salida de emergencia, que esté fácilmente identificable y accesible.

Esta posibilidad que le brindaría el sistema de salir del estado en que se encuentra, brinda la sensación de seguridad al usuario, pues le permitiría dejar cuanto antes alguna situación no deseada.

Las salidas deben ser siempre evidentes y se deben poder encontrar a simple vista. Las recomendaciones para ellas, son:

Recomendaciones
■Brindar avisos informativos
■Brindar salidas en cada pantalla
■Brindar salidas en cada contexto
■Brindar salidas en cada operación, tarea o transacción
■Brindar salidas en cada estado
■Brindar salidas en cada momento

Tabla 1.7: Recomendaciones sobre “Salidas Evidentes

Las salidas por lo general, se manifiestan como una opción de “Salir” propiamente dicha, pero también como opciones de “Volver”, “Cancelar”, “Deshacer”, “Reestablecer”, “Suspender”, “Modificar”, que deben estar todas o algunas, dependiendo del caso.

Hay que considerar que proveer salidas es importante para que el usuario pueda interrumpir la interacción con el sistema, por ejemplo, ante la presencia de eventos externos que requieran su atención urgente.

Mensajes de error

Eventualmente, cuando los usuarios realizan alguna combinación errónea de funciones provistas por el sistema, éste suele responder con mensajes de error.

Los mensajes de error constituyen un tipo de interacción que fluye de la máquina al usuario. Es el feedback del sistema ante la presencia de un error.

Estos mensajes de error, deben ser expresados en un lenguaje natural, su contenido debe indicar en forma clara y precisa, cuál es el problema, el lugar donde se efectuó, y sugerir alguna solución alternativa.

Hay que considerar que el usuario llegó a un punto crítico del sistema, el mismo percibe inseguridad, incertidumbre por desconocer qué hizo bien, qué hizo mal y en qué grado, el error cometido, puede afectar a los datos.

Los desarrolladores de sistemas, no deberían proveer mensajes de error en forma mecánica e impensada, sino que deben analizar cada situación de error, investigar sus posibles causas y consecuencias.

El primer cuestionamiento que el desarrollador del sistema se debe hacer, es por qué se permitió que el usuario llegue a ese punto. Quizás ante la presencia de mensajes aclaratorios, buena información o más control, el error se hubiera evitado. En principio, se debería prevenir la presencia de errores de gravedad.

Otra cuestión que se debe plantear, es que si el usuario ya se encuentra en ese punto crítico, de qué forma se lo puede ayudar para salir de él, lo más eficientemente posible.

Sobre este punto, se pueden dar las siguientes sugerencias:

Recomendaciones	Explicación
Debe existir mensajes de error	■El sistema debe detectar situaciones de error y poder avisarlas en el debido momento y lugar.
Brindar información del error	■Como mínimo, el mensaje de error debe incluir claramente el motivo del error, explicar la causa del mismo.
Evitar mensajes de error vagos	■No debe mostrarse al usuario mensajes que no dicen nada en concreto o, que presenten ambigüedades. ■Por ejemplo, el empleo de frases como “campo inválido”, “número inexistente”, “formato erróneo”, no indican el tipo de información esperada.
Debe dar alternativas a seguir	■No sólo se debe dar a conocer cómo se llegó hasta una situación de error sino cómo salir de allí. ■Inclusive, se debe brindar comandos para deshacer, volver a un punto anterior seguro, u ofrecer mecanismos de solución automatizados.
No debe existir mensajes de error que intimiden al usuario	■No son aconsejables mensajes que emitan sonidos, o que utilicen un diálogo que alarme al usuario. ■Tampoco se debe utilizar reprimendas, amenazas ni sanciones.
Se debe categorizar los diferentes tipos de mensajes	■Los mensajes de error, deben ser clasificados según sean advertencias, o si dan información sobre errores cometidos, sobre errores severos o críticos, entre otras posibilidades.
Se debe determinar el grado de intervención del sistema.	■Hay que determinar claramente, a qué tipo de acción se le van a vincular los mensajes de error. Si se lo asocia a acciones elementales, compuestas o a transacciones más generales. ■Esto afectará al grado de interrupción y de control determinado por el sistema.

Tabla 1.8: Recomendaciones sobre “Mensaje de error”

Prevención de errores

A veces, es preferible que en vez de proveer buenos mensajes de error, se brinde un diseño más cuidadoso que prevenga problemas, para que directamente estos no ocurran, así la persona no se encuentra en situaciones inesperadas.

En este principio, se establece la necesidad de evitar que el usuario llegue a una instancia de error. El desarrollador del sistema, deberia saber cuáles son las partes más críticas del software, con mayor posibilidad de error, y por lo tanto, debería afianzar los mecanismos de asistencia y feedback, para que estos no se produzcan.

Generalmente, la prevención debe estar presente en la carga de datos como también, en las transacciones más relevantes y críticas, como ser eliminaciones, transferencias y modificaciones.

Algunos mecanismos de prevención son:

Recomendaciones
■Brindar rangos de entradas posibles, para que el usuario seleccione y no tipee. ■Mostrar ejemplos, valores por defecto, formato de entradas admisibles. ■Brindar entradas de datos preformateadas, con caracteres fijos como barras “/”, “@”, “-”, o cualquier símbolo nescesario para evitar el tipeo de los mismos. ■Brindar mecanismos de corrección automática en el ingreso de los datos. ■Ser flexibles, aceptando una letra o por un cero, una letra i por un 1 y demás errores que un usuario puede cometer con frecuencia. ■Cuando se busca por un término, que el sistema busque expresiones similares, a fin de darle al usuario más información relacionada, evitándole molestias y pérdida de tiempo.

Tabla 1.9: Recomendaciones sobre “Prevención de errores”

Atajos

Los atajos, son mecanismos que provee el sistema, para reducir el número de pasos necesarios para disparar ciertas acciones.

Aquí, se pide que el sistema muestre ciertos aspectos de adaptación, que considere la posibilidad de dialogar con usuarios que pueden tener distinto grado de conocimiento, formación y experiencia en el uso de aplicaciones.

Por ello, la interfaz debería proveer alternativas de manejo para que resulte cómodo y amigable, tanto para usuarios novatos como para los muy experimentados, proveyendo de esta manera mecanismos de adaptación a través de atajos o mediante la definición de macros.

Para esto, se sugiere lo siguiente:

Recomendaciones
■Brindar caminos alternativos para acelerar la interacción con el sistema. ■Brindar posibilidad de reorganizar barras de herramientas, menús, colocando sus ítems, de acuerdo a la necesidad del usuario. ■Brindar mecanismos de macros, atajos, definición de teclas de función, para acelerar el acceso a las funciones.

Tabla 1.10: Recomendaciones sobre “Atajos”

Ayudas

Las ayudas son componentes de asistencia muy importantes para el usuario, y esto debe ser tenido en cuenta por el desarrollador del sistema. Muchas veces no se incorpora ningún tipo de ayudas, o se construye a la ligera sin prestar demasiada atención a las mismas.

Existe mucho análisis sobre cómo deben ser diseñadas las ayudas de un software, más aún porque están constituidas por texto o información, que debe ser interpretada en forma rápida y eficiente.

Cuando un usuario ingresa a una ayuda, lo hace porque duda, desconoce, quiere consultar, o teme por realizar alguna acción errónea. Tiene la presión de entrar a la misma para buscar, aprender, leer, interpretar, entender sobre la problemática que le aqueja. Por lo tanto, un mal diseño de las mismas puede llegar a

entorpecer, y dificultar este proceso, más que esclarecerle o asistirlo realmente.

Actualmente, los sistemas de ayuda son considerados el manual en línea del software, y tienen un objetivo explicativo, ilustrativo y de capacitación, que excede el marco de ayuda puntual, para integrarse más al sistema como un todo.

En el momento de diseñar las ayudas del sistema, se debe considerar las siguientes recomendaciones:

Recomendaciones	Explicación
Las ayudas deben existir	■El sistema interactivo debe ofrecer ayudas tanto a nivel del sistema en general –presente en la portada del sistema- como en cada pantalla, contexto, transacción o ante una instancia de diálogo complicada. ■La misma debe indicar la información necesaria para que el usuario pueda llevar a cabo sus intenciones en forma correcta.
Brindar diferentes tipos de ayuda	■Debe existir tipos de ayuda apropiadas para cada caso, como ser “Tips” para ciertos botones, el botón “?” para ciertas cajas de diálogos, asistencia contextual en línea, entre otros.
Diseñar bien el contenido de las Ayudas	■Las ayudas deben incluir información sintáctica –propia del uso de sistema de software- como información semántica –propia del aplicativo en cuestión.
Organización del contenido	■El contenido de las ayudas, debe ser organizado mediante una estructura simple, que facilite la navegación y exploración del mismo. ■La estructura de organización de la ayuda, debe ser lógicamente planificada y representada visualmente en forma coherente. ■Se deben evitar grafos hiper-conectados de páginas de información, ciclos, estructuras no balanceadas, entre otros.
Proveer diferentes formas de lectura	■Proveer mecanismos de acceso a través de la navegación como de acceso directo. Permitir también una lectura secuencial.
Diseño global de las ayudas	■Considerar la forma de organización del texto, al identación, el desplazamiento de páginas, el tipo de letra apropiado, los tamaños de los párrafos. ■Se debe brindar formas visuales eficientes para resaltar los puntos más importantes, el tamaño de las páginas
Brindar diferentes servicios de comunicación asistencial	■Brindar diferentes mecanismos de asistencia, como búsquedas, soporte en línea, comunicación con el soporte técnico mediante el correo o fax, acceso a las preguntas frecuentes o FAQs por Frequently Asked Question, entre otros.

Tabla 1.11: Recomendaciones sobre “Ayudas”

Recordar

❒Los Principios de Nielsen son normas de diseño aplicables a cualquier tipo de interfaz.
❒Los mismos pueden ser utilizados también, como heurísticas de usabilidad, en un marco de evaluación de sistemas de software, ya desarrollados y en pleno uso.

Caso de Estudio 1 - Fundamentos en HCI

En esta sección presentaremos un caso de estudio realizado por alumnos de la cátedra Diseño centrado en el Usuario, materia correspondiente al plan de estudios de la carrera Licenciatura en Informática, de la Facultad de Informática de la Universidad Nacional de La Plata http://www.info.unlp.edu.ar

Objetivo

El objetivo de este caso de estudio es presentar una práctica donde los estudiantes analizan la interfaz de usuario de un sistema implementado por su pares antes de cursar la materia, a la luz de los nuevos conceptos adquiridos al momento. Reflexionar y evaluar requiere de un nivel de comprensión de la temática imprescindible para continuar con los nuevos temas a abordar.

Características del Caso de estudio

La práctica consistió en la evaluación de un sistema de gestión de la Comisión Universitaria sobre Discapacidad de la Universidad Nacional de La Plata. La aplicación permite administrar integrantes, reuniones y eventos organizados por comisión y obtener los correspondientes reportes. Este trabajo se desarrolló como trabajo final en el dictado de la materia Proyecto de Software, materia correlativa de Diseño Centrado en el Usuario. El informe que se presenta fue desarrollado por Juan Manuel Castro (su perfil en Facebook es juan.m.castro.7186) y José Luis Larroque en la cursada del año 2010.

Desarrollo del Caso de estudio

Al comenzar la instalación del sistema, se encuentran problemas de diálogo simple y natural. Al crear una cuenta para el administrador se encuentran 2 opciones en las cuales no se puede interpretar a que hace referencia: la opción donde se debe elegir un color y otra para elegir si el panel lo queremos ubicar a la derecha o izquierda, como se puede observar en la figura 1.1

Figura 1.1: Pantalla de instalación del software con problemas de diálogo simple y natural

Cuando se está creando la cuenta no es posible identificar a que panel hace referencia, ni tampoco a qué aplicaría el color seleccionado. Este error estaría incluido en el principio 1 de diálogo simple y natural dado que estas características deberían poder configurarse una vez que el usuario haya tenido acceso al sistema y pueda saber qué es lo que va a modificar.

Cuando se ingresa como invitado en la sección del panel de de configuración se brinda la posibilidad de realizar algunos cambios. Sin embargo, si se intenta cambiar el color o la orientación del panel se despliega un mensaje de error: “Los datos no son correctos”, como se observa en la figura 1-2.

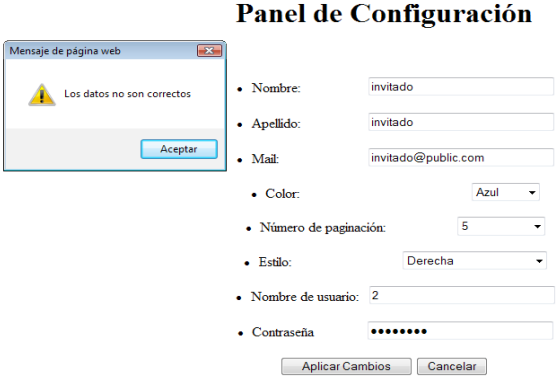


Figura 1.2: Problemas al actualizar información como usuario invitado con mensajes de error poco claros.

Este error estaría incluido en el principio 8 de proveer buenos mensajes de error, ya que el mensaje podría ser mas especifico. También en el 9 para prevenir mediante un buen diseño la aparición de problemas ya que el invitado no puede realizar ciertas modificaciones. Una forma de evitar el error seria deshabilitar esta sección cuando se loguea como invitado.

En la sección de Contacto existe un caja de diálogo (Para) donde se debería escribir el nombre de la persona a quien va dirigido el mensaje. Pero como se ve en la figura 1.3, es posible escribir cualquier caracter en el campo de entrada y es aceptado. Una posibilidad de solución sería poder elegir del una lista de destinatarios y eliminar este campo, validando las entradas ingresadas por le usuario. Este error estaría incluido en el principio 3 de minimizar la memoria del usuario.

Cuando se ingresa como invitado en la sección del panel de de configuración se brinda la posibilidad de realizar algunos cambios. Sin embargo, si se intenta cambiar el color o la orientación del panel se despliega un mensaje de error: “Los datos no son correctos”, como se observa en la figura 1-2.

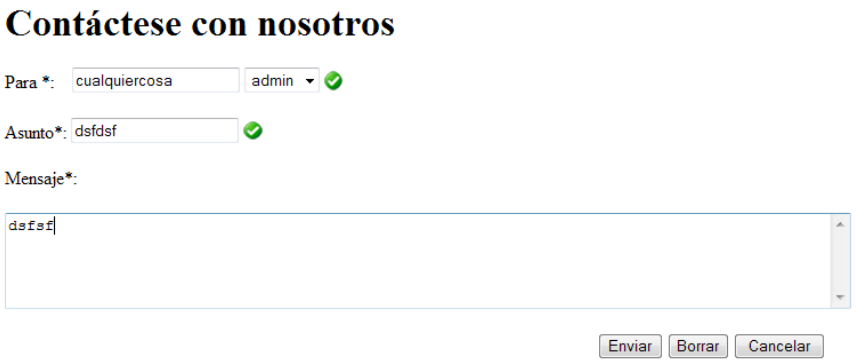


Figura 1.3: Problemas de validación de los campos de entrada.

La pantalla para agregar integrantes también presenta problemas. El botón Editar agrega al integrante cuando los datos están completos. No edita, sino que cumple la misma función del botón Agregar lo cual no es consistente.

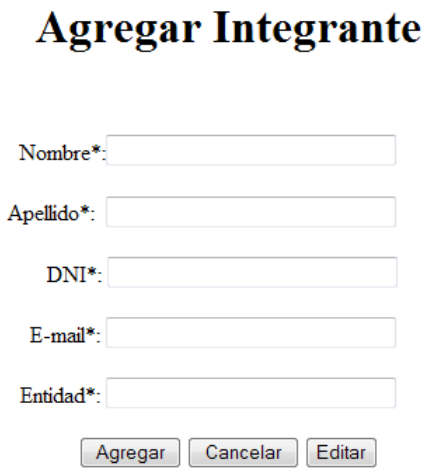


Figura 1.4: Problemas de consistencia ya que el botón Agregar y Editar poseen la misma funcionalidad

Este error estaría incluido en el principio 1 de diáolog simple y natural, donde la información debe tener un orden natural y lógico, y el principio 4 referido a la consistencia visual y terminológica.

La página para agregar reuniones también presenta problemas de diseño. Cuando se agrega una reunión es posible ingresar cualquier valor en los campos de entrada, como se puede observar en la figura 1.5.

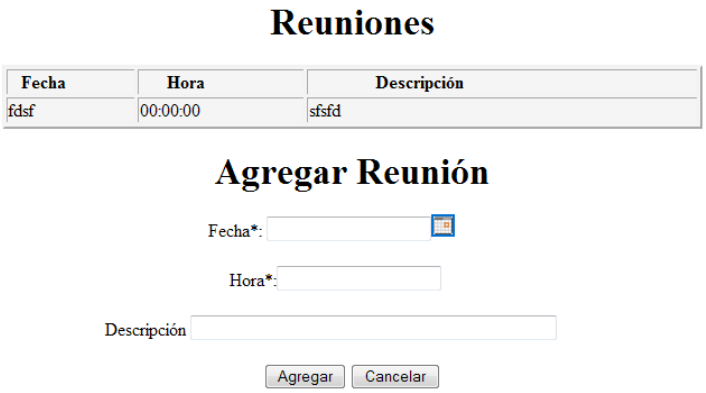


Figura 1.5: Problemas de validación de los campos de entrada

Este error estaría incluido en el principio 9 de prevención de errores. Se podría evitar este tipo de error desplegando un calendario para seleccionar la fecha. Esto mismo podría aplicarse a la selección del horario.

En la sección eventos encontramos el mismo error.

Otro problema se encuentra en la página de Agregar una Reunión está relacionado con el botón Cancelar. Este botón en realidad vuelve a la sección anterior, no cancela la operación que estamos realizando. Este error estaría incluido en el principio 6 de salidas evidentes. Se debe proveer claramente la manera de cancelar la tarea actual. Además, debería estar separada la función cancelar del botón volver.

Otro error detectado relacionado con el feedback adecuado. Registrar reuniones y eventos no se actualizan en el calendario en forma automática. En las figuras 1.6 y .17 se pueden observar el registro de 2 eventos a realizarse en el mes de abril.

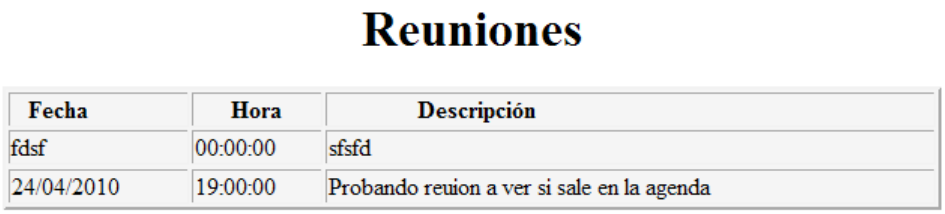


Figura 1.6: Reunión registrada en el mes de abril

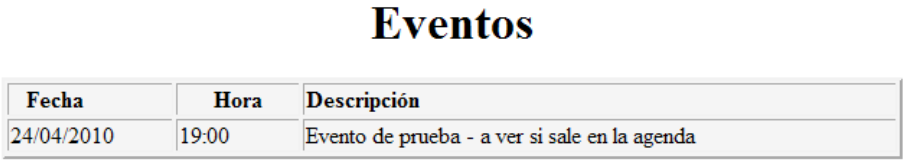


Figura 1.7 Evento registrado en el mes de abril

En la figura 1.8 se puede observar el calendario sin actualizaciones.

Figura 1.8: Problemas de feedback en la visualización del calendario

Finalmente en la sección de Búsqueda, como usuario invitado, no distingue entre el título y el resultado obtenido, como se puede observar en la figura 1.9 La información resultante no es clara, infringiendo el principio 1 de diálogo simple y natural, presentando además información irrelevante.

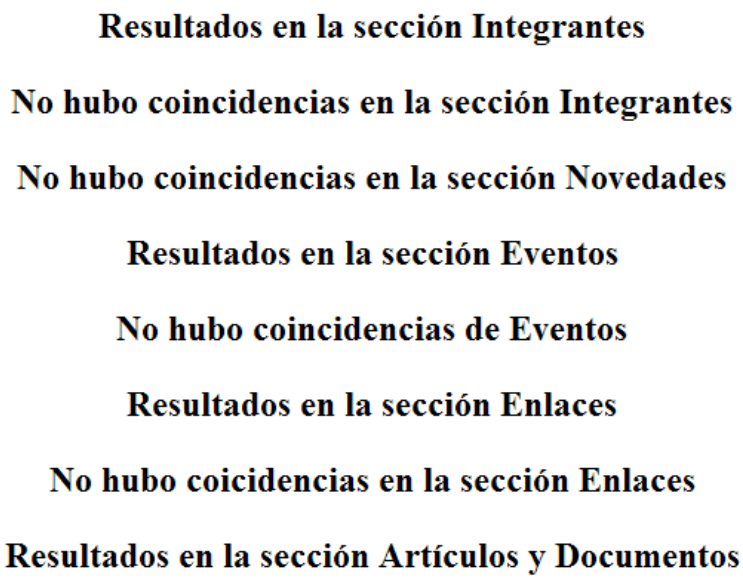


Figura 1.9: Problemas con la forma de visualizar la información y presencia de información irrelevante.

Podemos concluir entonces que un sistema desarrollado sin tener en cuenta los principios básicos de diseño de interfaces trae aparejados problemas de usabilidad que se pueden prevenir facilmente. La evaluación de pares es una actividad productiva y muy valorada en las encuestas finales de los distintos cursos.

Ejercitación

A - Indicar si las siguientes expresiones son verdaderas o falsas. Tachar lo que no corresponda:

- La disciplina de HCI trata solamente cuestiones de software. V F
- La construcción de una interfaz del usuario es responsabilidad exclusivamente del profesional informático. V F
- En una interfaz del usuario, puede existir un único tipo de diálogo. V F
- La interfaz icónica es una interfaz visual. V F
- El rol del desarrollador del diálogo, incluye la organización, planteo y realización de los modelos de usuarios, del contexto y de aspectos tecnológicos, que conforman la etapa de requerimientos de la interfaz. V F
- La independencia entre la componente de interfaz y la de aplicación, sólo se debe aplicar en la etapa de programación de ambas. V F
- Los principios de Nielsen constituyen normas de diseño que pueden aplicarse en un proceso evaluativo de la interfaz. V F

B - Indicar específicamente a qué componente de Green se está refiriendo en cada caso:

Particularidades de la UI	Componente de Green
Definición de las dependencias entre objetos de interacción.	
Diseño de las tarjetas magnéticas que permitirán el ingreso al sistema.	
Definición de las máscaras y formatos de entrada.	
Definición de las condiciones que llevan a la aparición de un mensaje de error.	
Diseño de las tablas de bases de datos donde se registraran los perfiles de usuario.	

C - Explicar y justificar la relación entre los principios de Nielsen y los objetivos de la interfaz del usuario.

Referencias

❒Baber, C. (1997). Beyond the Desktop: Designing and Using Interaction Devices, San Diego, CA: Academic Press.

❒Baecker, R. M.; Grudin, J.; Buxton, W. A. S. y Greenberg, S. (1995). Readings in Human-computer Interaction: Toward the Year 2000. Morgan Kaufmann Publishers: San Francisco, CA.

❒Barnard, P. J. (1987). Cognitive resources and the learning of human-computer dialogs. Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. MIT Press, Cambridge, MA.

❒Brad A. Myers (1994). Challenges of HCI Design and Implementation. ACM Interactions. vol. 1, no. 1.

❒Butter, Keith A. (1996). Usability Engineering Turns 10. Interactions.

❒Hartson, Rex, Hix, Deborah (1989). Human-computer interface development: concepts and systems for its management. ACM Press, New York, NY, USA.

❒Mark Green, (1985). Report on Dialogue Specification Tools. User Interface Management Systems, Berlin: Springer-Verlag, pp. 9-20.

❒Nielsen, Jakob (1989). Coordinating User Interfaces for Consistency.

❒Nielsen, Jakob (1994). Usability Engineering.

❒Nielsen, Jakob. (2000). Designing Web Usability: The Practice of Simplicity.

❒Nielsen, Jakob; Mack, Robert (1994). Usability Inspection Methods.

❒Norman, D. (1991). Cognitive Engineering in User Centered System Design.

❒Norman, D. A. (1983). Some observations on mental models. Lawrence Erlbaum Associates, Hillsdale, NJ.

❒Olsen, S. E. (1987). CATOOL: A computer-based tool for investigations of categorical information in mental models. Human-Computer Interaction INTERACT ‘87. North-Holland, Amsterdam.

❒Reason, J. (1988). Framework models of human performance and error. En Tasks, Errors and Mental Models. Taylor & Francis, London.

❒Reason, J. (1990). Human Error. Cambridge University Press, Cambridge, England.

❒Reisner, P. (1981). Formal grammar and human factors design of an interactive graphics system. IEEE Transactions on Software Engineering, SE-7(2).

❒Rouse, W. B. (1981). Human-computer interaction in the control of dynamic systems. ACM Computing Surveys, 13 (1).

❒Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, MA.

❒Wahlström, B. (1988). On the use of models in human decision making. En Tasks, Errors and Mental Models, cap. 10. Taylor & Francis, London.

❒William Buxton, (1983). Lexical and Pragmatic Considerations of Input Structures. Computer Graphics, (17)1.

❒Pattie Maes, Alan Wexelblat, (1996). Software Agents. CHI 96 Tutorial.

❒Douglas Engelbart (1963). Conceptual framework for the augmentation of Man’s Intellect.

Proceso de Desarrollo de la Interfaz del Usuario

02

Proceso de Desarrollo de la Interfaz del Usuario

La interfaz del usuario requiere de un proceso de desarrollo especial. Se forma con la práctica, con sucesivas modificaciones y testeos frente a los usuarios potenciales del producto. Así, se garantizará la calidad de la misma.

Temas a Tratar

I- Introducción
II- Características del Proceso de Desarrollo de la UI
III- Ciclo de Vida de la Interfaz del Usuario
IV- Etapa de Requerimientos de la Interfaz
- Modelo SSOA

V- Etapa de Diseño de la Interfaz
- Características del Proceso de Diseño

VI- La interfaz y la Prototipación
- Características de la Prototipación
- Ventajas de la Prototipación
- Tipos de Prototipos

VII- Etapa de Implementación
- Herramientas para el desarrollo de la Interfaz

VIII- La importancia de la Evaluación de las interfaces del usuario
- Métodos de Indagación
- Métodos de Inspección

Ejercitación
Referencias

I - Introducción

Las aplicaciones informáticas abarcan cada vez más áreas, es decir, es difícil encontrar disciplinas que no cuenten con herramientas computacionales que faciliten el trabajo cotidiano.

A raíz de que el espectro de usuarios se amplía, es necesario utilizar estrategias que favorezcan el uso de las aplicaciones proveyendo un estilo de comunicación más simple, que reduzca el tiempo de aprendizaje, que sea fácil de usar sin requerir ningún entrenamiento específico en informática.

Por lo tanto, la componente de Interfaz del usuario se convierte en un factor decisivo tanto en la elección de estos sistemas de software como en su posterior utilización y aprobación. Ella afecta en forma directa al grado de satisfacción de los usuarios.

En pos de proveer una interfaz del usuario de calidad, es necesario entender que su construcción y diseño, forma parte de la Ingeniería de software y que requiere un proceso de desarrollo especial, que está centrado en el usuario, en todas sus etapas de gestación. La temática de HCI y el tratamiento de toda la problemática involucrada dentro de la componente de Interfaz, figura explícitamente en la currícula de ACM (The ACM SIGCHI Curricula for Human-Computer Interaction), que es una recomendación internacional de sumo prestigio en las Ciencias de Computación.

Como es una porción del sistema que debe interactuar directamente con los usuarios, es fundamental efectuar un estudio completo del comportamiento humano. Se debe incluir como requerimientos indispensables en la generación del software, parámetros tales como ergonomía, facilidad de uso, amigabilidad, simpleza, en término de reducir el esfuerzo mental del usuario de llevar a cabo en la computadora la tarea que tenía en mente, flexibilidad y naturalidad en el sentido que el usuario a través de la pantalla vea reflejado o modelado su realidad.

A pesar de esto, la mayoría de los diseñadores de software están más compenetrados con las computadoras que con el aspecto humano, están más capacitados en resolver el problema en términos computacionales, buscar los mejores algoritmos de programación, las mejores estructuras de datos, que en proveer un buen estilo de interacción.

Además, resulta muy difícil saber el significado de lo que es proveer una interfaz del usuario «óptima» de acuerdo a los factores humanos. No se cuenta aún con metodologías teóricas que abarquen todo el desarrollo y diseño de las mismas, y no hay todavía métricas exactas establecidas para su evaluación.

Las interfaces del usuario, se construyen mediante un proceso iterativo de diseño. Ellas se forman con la práctica, necesitan ser diseñadas, modificadas, prototipadas y evaluadas varias veces hasta alcanzar los requerimientos de los usuarios.

En este capítulo, se describirán las características del ciclo de vida de la interfaz del usuario, cómo se compone y en qué consiste cada etapa de desarrollo.

II - Características del Proceso de Desarrollo de la UI

Diseñar, por lo general, es un proceso difícil y, más aún si se diseñan cada vez sistemas más complejos, más avanzados tecnológicamente.

Ahora bien, diseñar bajo la conciencia en HCI, también es complicado. Los diseñadores de la interfaz del usuario, se deben preocupar sobre los factores humanos y ergonómicos, deben estudiar sobre los efectos del sistema en los usuarios, grupo de trabajo, empresa, o en el contexto particular, y deben investigar sobre las mejores representaciones y formas de interacción que va a proveer el sistema.

Además, la interfaz del usuario, al constituir una parte del software, requiere para su construcción, de todas las etapas de desarrollo tales como análisis, diseño, implementación y evaluación. Como es una componente que va a ser usada y manipulada en forma explícita por parte de usuarios y va a interactuar directamente con ellos, necesita que estas etapas se desarrollen de una manera no tradicional. La misma requiere por ejemplo, que los bosquejos de diseño sean probados ante los usuarios en forma prematura y aunque estén incompletos, llevando a lo que se denomina el desarrollo de prototipos, que se explicarán detalladamente luego en este mismo capítulo.

En el momento de emprender el desarrollo de una interfaz, el informático debe entender y conocer las características de esta componente de software y tratar toda su complejidad mientras se encuentra en vías de desarrollo. Esto le permitirá encarar correctamente este proceso y poder sobrellevar las problemáticas que presenta.

Entonces, se puede decir que el proceso de desarrollo de la interfaz se caracteriza por las siguientes cuestiones:

a. Constituye un Proceso Iterativo:

El desarrollo de la interfaz conforma un proceso constante e iterativo de refinamientos sucesivos.

Como se va a explicar en la siguiente sección, la interfaz del usuario no presenta una naturaleza de

desarrollo analítica ni secuencial.

La misma requiere para su construcción, ser evaluada frente a usuarios potenciales y modificada en forma sistemática. Esto implica que se pueden saltear o reiterar las etapas de desarrollo.

b. Presenta una Etapa de Requerimientos incompleta:

La etapa de requerimientos de la interfaz del usuario por lo general es ausente e incompleta. A los usuarios le resulta más fácil expresar las especificaciones computacionales que los requerimientos de visualización, diálogo hombre-computadora.

Se dan muy pocos casos, en que los usuarios o clientes logran especificar claramente lo que desean a nivel de elementos de interfaz, objetos de interacción a utilizar, tipos de feedback, entre otros aspectos. Entonces, es inusual obtener de ellos un enunciado fehaciente sobre los requerimientos a nivel de la componente de diálogo.

c. Presenta una Naturaleza Bottom-up:

A diferencias de la componente de aplicación, en donde el desarrollador tiene una visión global del problema y a partir de allí, intenta refinarlo en problemas más pequeños, la interfaz del usuario se va originando a partir de diferentes puntas de información muy detalladas.

Inicialmente, el desarrollador de la interfaz contará con mucha información heterogénea tales como características del usuario, del ambiente, de la aplicación, información de los estándares del mercado, de los aspectos tecnológicos, de dispositivos para interactuar con el sistema, entre otras cuestiones.

Esto significa, que el mismo deberá comenzar a trabajar con información muy puntualizada y disímil, por lo que deberá tratarla realizando generalización, filtrado, abstracción y sintetizado de datos, en pos de converger a una versión de diseño de la interfaz apropiada.

d. Requiere un Mecanismo de Representación Especial:

La componente de interfaz necesita ser evaluada y probada por diferentes tipos de usuarios. Entre ellos, se encuentran los usuarios finales que son los destinatarios del software, los programadores que la van a desarrollar, los evaluadores externos que la van a criticar, como también otros profesionales idóneos en temas afines, a saber sociólogos, diseñadores visuales, ingenieros electrónicos.

Además, se trabaja con diversas visiones del tema que son complementarias. Aunque cada aspecto tiene aristas que le son propias, también comparte aspectos comunes a coordinar.

Por tales motivos, la interfaz requiere un mecanismo de representación entendido por todos los roles, necesita de notaciones o esquemas representativos más simples, no tan formales ni tan técnicos. No es lo mismo que la componente de la aplicación, cuya representación o diseño debe ser entendido e interpretado por el programador, por lo tanto se puede utilizar diagramas, notaciones técnicas tradicionales.

e. Requiere de Métodos de Diseño Específicos:

La interfaz del usuario está conformada, como se explicó en el Capítulo I, por componentes tales como la de presentación, la componente de control de diálogo y la de modelo interfaz-aplicación. Las cuáles involucran muchos aspectos diferentes que deben diseñarse y desarrollarse.

Es completamente distinto, diseñar las cuestiones de visualización que los mecanismos de interacción, o la componente de control de diálogo con el enlace con la aplicación.

Por la diversidad de aspectos que aborda la interfaz del usuario, será necesario emplear varios métodos de diseño que sean específicos a cada área. Por ejemplo, los métodos de análisis de tareas, describen acciones del usuario y conducta de la interfaz, pero no sirven para diseñar el control del diálogo. En cambio, la técnica de diagramas de transición y estados o de eventos, describen la interacción y control de diálogo, pero no ofrecen técnicas de representación para las pantallas del sistema.

f. Existe una Anticipación de las Etapas de Desarrollo:

El desarrollo de la interfaz del usuario, es un proceso en donde se anticipan las etapas de desarrollo.

Este es el caso de la etapa de diseño de la interfaz, que no se puede completar hasta que no se implemente o se desarrollen prototipos de la misma. La misma necesita información futura, que no se puede obtener de ninguna forma hasta que no se ponga en marcha el/los bosquejos de diseño.

Lo mismo sucede con la etapa de requerimientos o de análisis. ¿Cómo saber qué es lo que pretende o necesita el usuario a nivel de interfaz, sino se le muestra bosquejos de diseño para discutirlo?

Entonces, se da el caso en que se requiere predecir el estado futuro de un producto basado en información actual, pero donde esta información corriente se obtiene poniendo en marcha o en práctica, las predicciones realizadas.

El motivo de este fenómeno, se debe fundamentalmente a que, por tratarse de una componente en contacto

directo con seres humanos, permanentemente se trabaja con información imprecisa, incierta, subjetiva, versátil, durante todo el proceso de desarrollo, y es necesario entonces, anticipar las etapas de diseño o de evaluación para concretar, redondear dicha información.

G. Prelación de la Etapa de Evaluación:

La etapa de evaluación, en el caso del desarrollo de la componente de interfaz, es fundamental para probar la corrección y completitud de etapas previas, como las etapas de requerimientos o la de diseño.

A diferencia de la componente de aplicación, donde la evaluación se puede dejar al final del proceso de desarrollo, en el caso de la interfaz, la misma se aplica en todas las etapas. Se la utiliza para completar la etapa de requerimientos, para comprobar lo correcto de los bosquejos de diseño, para evaluar las versiones prototípicas, como también para revisar los aspectos de implementación.

La etapa de evaluación, permite adecuar la interfaz, cuando no se está tan condicionado por la estructura del sistema. Permite encarar e incorporar adecuadamente, todas las modificaciones que sugieren los usuarios cuando prueban la interfaz, mientras ésta aún se encuentra en vías de desarrollo.

H. Requiere Participación Activa de los Usuarios:

Por una cuestión que los factores humanos deben ser considerados en todo el proceso de desarrollo, el usuario debe participar en forma efectiva y sistemática.

El mismo se debe sentir comprometido con las decisiones de diseño, se le debe consultar y se lo debe incluir en la mayoría de las pruebas de usabilidad efectuadas en el sistema.

El usuario se convierte en coartífice o coautor del diseño de la interfaz que él va a utilizar, por lo tanto se garantiza un alto grado de satisfacción personal. El usuario va a disminuir su desconfianza ante un sistema que no sea rígido y se adecue más a sus preferencias.

I. Es un Proceso Interdisciplinario:

La creación de la interfaz del usuario constituye un proceso interdisciplinario, con expertos en Ingeniería de Software, Factores Humanos, Diseño Visual, Expertos en Multimedia, Psicología, Comunicación Social, entre otros. Cada uno de los cuáles, tiene su propio lenguaje, sus diferentes visiones, interpretaciones, y esto puede complicar la comunicación entre los mismos.

Es necesario organizar un marco de trabajo con pautas claras de participación y toma de decisiones, con una firme definición de roles y una eficiente coordinación.

Importante

- El desarrollo de la interfaz del usuario, es un proceso especial, en donde las etapas de desarrollo se suceden en forma iterativa. Se las necesita evaluar y refinar para lograr la completitud y veracidad de las mismas.
- Se requiere utilizar una metodología apropiada, que soporte esta concurrencia y anticipación de etapas.
- También, es necesario encontrar las técnicas o métodos de diseño adecuados, para delinear las características de visualización, de presentación, como de interacción y control del diálogo que incluye la interfaz del usuario.
- Muchos diseñadores expertos han efectivamente desarrollado sus propios métodos de diseño basados en sus experiencias personales con ciertos dominios, técnicas y situaciones. Por lo general, se basan en sentido común, en conocimientos que están implícitos y en experiencia personal.
- El empleo de métodos informales suelen ser difíciles de transmitir y no permiten la reusabilidad de casos existosos.

III - Ciclo de Vida de la Interfaz del Usuario

Algunos desarrolladores de software posponen la construcción de la interfaz solamente a la etapa de implementación, pero como se explicó en la sección anterior, existen descripciones, detalles, consideraciones presentes en la interfaz que intervienen y afectan a todo el proceso de ingeniería del sistema y que deben ser encaradas desde el principio.

Tanto la interfaz del usuario como la parte del cómputo, se desarrollan en etapas que consisten de una fase de especificación, diseño, implementación y de un proceso de evaluación. Pero, la forma en que se

manifiestan estas etapas en la interfaz difiere radicalmente con la de la aplicación.

Mientras que, según la Ingeniería de Software clásica [Pressman Roger,1996], la componente de la aplicación requiere, que cada etapa de desarrollo culmine en forma correcta y completa antes que comience la siguiente, la interfaz del usuario necesita que las mismas se realicen en forma simultánea y concurrente, y que estén en constante estado de evaluación y modificación.

La diferencia reside, en que en el diseño centrado en el usuario, el usuario juega un rol preponderante en todas las etapas de desarrollo, desde la etapa inicial donde se lo modela, hasta participar en evaluaciones del producto, o en sugerir aspectos de diseño.

El ciclo de vida de la interfaz, requiere de un proceso iterativo de refinamientos sucesivos, donde las etapas se solapan, se adelantan y se regeneran, constituyendo lo que se denomina “Ciclo de Vida Prototípico” [Reilly J.P., 1996], a diferencias del ciclo de vida tradicional en cascada.

En el ciclo de vida prototípico, las etapas iniciales de requerimiento y de diseño se desarrollan en forma incompleta y se implementan parcialmente generando lo que se denominan “prototipos del sistema”.

Como el prototipo comienza mostrando la confección parcial del producto, se los utiliza para proveer una visión temprana del sistema, y así se lo puede poner a prueba anticipadamente frente a usuarios. Esto permitirá obtener información más concreta sobre las necesidades del usuario, preferencias, modos de trabajo, errores que comete, dificultades que presenta y demás cuestiones observables, que completarán y corregirán los requerimientos y diseños iniciales.

A medida que al prototipo se lo corrige, y se le van agregando en forma gradual nuevos elementos de interfaz, comportamiento, servicios y hasta, en el caso de los prototipos evolutivos, componentes funcionales de la aplicación, se van generando nuevas versiones prototípicas de mejor calidad de uso y que cuentan con la aprobación garantizada de los usuarios.

También, se podrían utilizar prototipos descartables o revolucionarios, donde una vez que se determina lo que los usuarios realmente esperan y desean de la componente de interfaz, son empleados solamente como un medio de representación del bosquejo final de la misma, y luego, son descartados.

Todo este proceso se desarrolla mediante un esquema metodológico denominado “Prototipación”, que consiste en la generación sistemática de versiones prototípicas del software, que se van evaluando mediante la participación activa de los usuarios y que se van mejorando progresivamente.

El ciclo de vida culmina, cuando las evaluaciones de las versiones avanzadas del prototipo van arrojando resultados positivos, es decir, cuando el usuario o demás roles evaluadores, aceptan el diseño expresado y simulado a través del prototipo.

Importante

Es importante decir, que los requerimientos iniciales de una interfaz del usuario, son por lo general incompletos, y que esta técnica tiende a anticipar problemas, obteniendo mayores precisiones de lo que el usuario espera del sistema.

A continuación, se ilustra el ciclo de vida prototípico de la interfaz, teniendo en cuenta un prototipado tipo evolutivo, en donde la última versión prototípica converge al sistema de software completo, totalmente culminado y aprobado. Se muestra aquí, cómo ocurren las diferentes etapas de desarrollo de la interfaz, a través del tiempo:

Figura 2.1: Ciclo de Vida Prototípico de una Interfaz del Usuario.

Como se muestra en el gráfico, ni bien se obtiene algunas especificaciones iniciales de lo que se pretende de la interfaz, constituyendo una etapa de requerimientos o de análisis incompleta, se comienza con el diseño de la misma.

Estos bosquejos de partes parciales de la interfaz, son implementados en forma de prototipos para ser evaluados ante el usuario, y poder así obtener más información de él, frente a un material concreto.

Con la información obtenida de las evaluaciones, se completa y corrige aún más las etapas de análisis y diseño. Por ello, estas etapas se establecen en forma casi simultánea.

Una vez que se desarrolle el prototipo final evolutivo, donde se incluye tanto la componente de diálogo como la funcional, y los testeos del mismo arrojen resultados positivos, culminarían en forma conjunta todas las etapas cruciales de desarrollo.

La etapa de entrenamiento, comienza desde las versiones prototípicas iniciales, donde el usuario comienza a conocer, probar, usar el producto y familiarizarse con él desde el principio.

El tiempo asociado al entrenamiento se puede extender un poco más, una vez que el producto haya sido terminado. Hay que considerar, que por lo general, no se trabaja con todos los usuarios, sino con una muestra de ellos, y en esta etapa, se extiende la capacitación a todos los usuarios involucrados.

Las etapas de conversión e instalación, se refieren a la puesta a punto del producto sobre las máquinas que se destinan para la utilización del software. En el caso que se mejora o se reemplaza un producto de software anterior, hay un proceso de transferencia, adaptación de datos e información, que constituyen la etapa de conversión.

Existen otros esquemas ilustrativos, del ciclo de vida de una interfaz, como el Modelo en Espiral [Barry W. Boehm, 1988] y el Modelo de Estrella [Deborah Hix y J.Hartson, 1989], que se muestran a continuación:

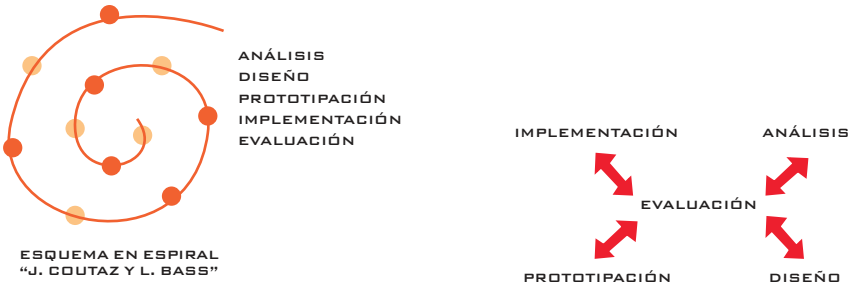


Figura 2.2: Modelo en Espiral y Modelo de Estrella.

Ambos diagramas, expresan la filosofía prototípica, ya que en el caso de la espiral, cada giro de la misma está constituido por el diseño, desarrollo y evaluación de una versión prototípica. En el caso de la estrella, muestra al proceso de evaluación como eje central, a partir del cuál se refinan las demás etapas de desarrollo.

Recordar

- El diseño centrado en el usuario, implica además de conocer al usuario y modelarlo, hacerlo partícipe de muchas actividades de desarrollo de la interfaz.
- El usuario juega un rol fundamental en la modificación, evaluación y diseño de la interfaz, y es artífice de que el ciclo de vida prototípico se logre adecuadamente.

IV - Etapa de Requerimientos de la Interfaz

En el desarrollo de un sistema de software, la componente de la interfaz del usuario debe ser tratada en forma independiente. La misma presentará particularidades y características propias, que deben ser tratadas y consideradas en forma autónoma, si se pretende brindar una componente de interacción de calidad.

La interfaz del usuario debe ser analizada especialmente, su forma y su comportamiento dependerán de múltiples factores que la harán única e irrepetible.

La etapa de requerimientos es esencial para descubrir e investigar dichos aspectos y todo tipo de información que pueda ser útil e influir en el futuro diseño de la componente de diálogo.

Esta etapa no se puede saltear, debe ser responsabilidad del diseñador de la interfaz encararla correctamente para averiguar los requerimientos del usuario a nivel de interacción, por lo que no basta con tener un enunciado completo de lo que se pretende a nivel funcional.

Puntualmente, el desarrollador de la interfaz del usuario, debe realizar las siguientes actividades antes de comenzar con el diseño de la misma:

a. Análisis Funcional del Sistema:

El desarrollador de la componente de interfaz del usuario no puede comenzar a diseñar, si no analiza previamente las características de la componente funcional.

Se debe tener claro conocimiento de los requerimientos funcionales, el objetivo del sistema de software y las características que se pretenden de la aplicación.

Hay que tener en cuenta, que no es lo mismo diseñar la interfaz de un sistema educativo que la de un sistema informativo o, por ejemplo, la interfaz de un sistema telefónico que la de un sistema de transacciones comerciales.

El tipo de aplicación es, entonces, otro aspecto que afectará en el diseño de la interfaz que se va a proveer.

El desarrollador de la interfaz puede, obteniendo del analista de sistemas el enunciado completo de los requerimientos funcionales, realizar sus propios modelos funcionales, prestando interés solamente en aquellas funciones que va a ser utilizada, accedida o percibida por el usuario.

Cualquier método de análisis, tal como diagramas de Entidades y Relaciones (E-R Diagrams), de flujo de datos (DFD), o modelo de tareas, pueden ser de utilidad para que el diseñador de la interfaz comprenda las futuras actividades que el usuario tendrá en la vida útil del software.

b. Análisis del Usuario:

Es fundamental que el diseñador conozca al usuario e investigue sus características individuales, si pretende brindarle una interfaz adecuada para él.

Esta etapa, constituye la esencia del Diseño Centrado en el Usuario. Muchos autores formulan este concepto, mencionando que los sistemas interactivos deben ser "centrados en el usuario" [Norman D.,1986], o que "conocer al usuario" significa que el diseñador pueda tener la capacidad de decidir qué nivel de soporte requerirá el usuario, tanto semántico como sintáctico [Schneiderman B.,1987].

Es importante estudiar la clase de usuarios que van a interactuar con el sistema y categorizar a los mismos. Se debe considerar si son usuarios novatos o con experiencia, si han utilizado computadoras con anterioridad o no, qué edad tienen. Si son tímidos, curiosos o ansiosos. Si prefieren el aprendizaje mediante el uso de documentaciones y manuales o, a través de pruebas y testeos con el propio software. Si necesitan ser guiados y controlados o, por el contrario, encaran los inconvenientes con sus propios medios, entre otras características.

Por ejemplo, la interfaz del usuario usada por niños debe tener un diseño completamente distinto a la que es utilizada por adolescentes. En el primer caso, las letras deben ser redondeadas, grandes, de colores vivos como rojo, azul, verde y amarillo, y debe emplearse un lenguaje claro y sencillo. Mientras que, en el caso de los adolescentes, se utilizan otro tipo de tipografías más modernas, un lenguaje más informal y con colores más oscuros.

Entonces, hay que realizar estudios y modelos sobre la comunidad de usuarios a la que está dirigido el sistema, teniendo en cuenta factores humanos, sus conocimientos sintácticos y semánticos, su experiencia y formación. Para esto, se construye un "modelo del usuario" que es una descripción de las características y propiedades individuales de la interacción entre el usuario y la máquina, permitiendo registrar y modelar sus hábitos y sus reacciones.

A grandes rasgos, se puede identificar dos tipos de modelos del usuario: aquellos que son "conceptuales" y los "cuantitativos".

Modelos	Explicación	
Modelo conceptual	■Los modelos conceptuales del usuario se caracterizan por representar los procesos cognitivos humanos e identificar las estrategias y estructura involucrada en la interacción humano-computadora. ■Se compone de elementos conceptuales, las operaciones que se ejecutarán sobre los mismos y también de sus relaciones y restricciones. ■Codifica lo que el usuario conoce o entiende de la aplicación y del uso de sistemas interactivos en general, o sea determina el grado de conocimiento semántico y sintáctico, a saber:	
	Conocimiento Semántico incluye:	■El conocimiento dependiente del dominio que es inherente a la aplicación específica en cuestión. ■El conocimiento dependiente de los sistemas interactivos, inherente a los conceptos generales del funcionamiento de los sistemas de software.
	Conocimiento Sintáctico incluye:	■Las convenciones lingüísticas y modales que el operador debe conocer para poder comunicarse con el sistema.
Modelo Cuantitativo	■Los modelos cuantitativos tratan con la representación numérica del rendimiento del usuario frente a la computadora. ■Varias características del usuario pueden ser modeladas: desde sus gustos o preferencias, su experiencia, hasta sus motivaciones u objetivos, y en cada caso será de información útil para diseñar la interfaz del usuario apropiada.	

Tabla 2.1: Tipos de Modelos de Usuarios.

Existen ciertas técnicas de cómo adquirir el conocimiento del usuario [Gould J. D., 1995], de cómo organizarlo, registrarlo e interpretarlo para posteriormente, desarrollar una modelación del mismo, que constituirá una descripción, tanto de sus características y propiedades individuales, como de su interacción con la máquina.

Entre las técnicas de recolección de datos más comunes, se encuentran, las entrevistas, cuestionarios, observación directa de la interacción del usuario frente a un software, filmaciones (video-taping) del usuario en su ámbito de trabajo, "Video Tracking", sesiones "Thinking-aloud", donde se insta al usuario a explayarse sobre los problemas e inconvenientes que tienen con el sistema vigente, entre otros.

Recordar

- Es imprescindible conocer al usuario destinatario del sistema, su personalidad, su experiencia, su formación. Se debe analizar y entender la representación mental que él conlleva.
- Conocer al usuario, es la única manera de proveer el diseño de la interfaz, más adecuada para él.

c. Investigación del Ambiente:

El ambiente o contexto, que rodeará al usuario mientras interactúa con el sistema, también debe ser estudiado cuidadosamente y afectará notablemente el diseño de la interfaz.

Se tienen que realizar estudios sobre:

El Ambiente	Explicación
Características del Ambiente Físico	■Hay que realizar un modelo del ambiente, estudiar el lugar físico donde se va a montar el sistema. ■Hay que analizar el grado de concurrencia, ruido, interferencias, nivel de distracción, cuestiones de luminosidad. ■Por ejemplo, no se podría aprovechar una interfaz con sonido, si el sistema se va a instalar en un lugar público, donde concurre mucha gente y hay mucho ruido.
Características de la Organización y del Entorno Grupal	■Se debe modelar al usuario dentro de un marco organizativo, que puede ser su oficina, su estudio, su laboratorio, su casa, una empresa. ■Se debe estudiar su posición, su rol en la organización, sus responsabilidades. ■Se debe estudiar el grupo que lo rodea, su formación, el nivel de conocimiento de los demás. ■Por ejemplo, la interfaz del usuario debería ofrecer conocimiento y asistencia más detallada, si el usuario es el único que sabe Computación dentro de su entorno.
Características Técnicas	■Se debe investigar las características técnicas y los recursos que el o los usuarios cuenten en el momento que se instale el sistema. ■Se debe inventariar el tipo de máquinas que poseen, el tipo de monitor, resoluciones posibles, tipos de conexiones que tienen y demás aspectos tecnológicos. ■Se debe considerar los dispositivos de hardware que se van a destinar para la interacción hombre-computadora. ■Por ejemplo, es inútil planificar el diseño de una interfaz muy cargada de imágenes, si el usuario cuenta con máquinas de poca capacidad gráfica.

Tabla 2.2: Actividades involucradas en la Modelación del Ambiente.

Todo este análisis, debe ser realizado cada vez que se encare el desarrollo de un software. Debe ser desarrollado en forma extensiva y completa, conformando una etapa de requerimientos específica a la interfaz del usuario.

Este proceso determina los pre-requisitos básicos para recién comenzar con la etapa de diseño, donde se decidirá en base a ello, qué tipo de interfaz se va a proveer y qué tipo de diálogo será necesario brindar.

Importante

- Es imprescindible diseñar la interfaz del usuario en base a los modelos efectuados de los usuarios, del contexto y de la aplicación.
- El diseñador puede tener su estilo, pero el diseño de la interfaz debe estar acorde a los modelos analizados anteriormente.
- La estandarización, el estilo de interfaz que se encuentre en la mayoría de los sistemas en el mercado, puede ser considerado como un punto de partida, pero no como el objetivo del diseño de la interfaz.

A partir de aquí, el diseñador empezará con la etapa propiamente de diseño, donde será fundamental, analizar las políticas de diseño que se deben emplear y aplicarlas según el tipo de interfaz que se desee proveer.

Modelo SSOA

El modelo SSOA [Schneiderman B., 1992], es el Modelo Sintáctico-Semántico del conocimiento del usuario. Provee un marco de definición sobre las diferentes formas y grado de conocimiento que un usuario puede emplear al interactuar con un sistema.

La distinción entre las sintaxis y semántica, surgió a partir de la definición de compiladores, donde se separó el proceso de compilación de la entrada de texto o comando, denominado análisis sintáctico, del proceso de interpretación del mismo, denominado análisis semántico.

Los diseñadores de los sistemas interactivos también pueden contar con un modelo sintáctico semántico, pero aplicado al conocimiento del usuario.

Como se explicó en la fase de requerimientos, el conocimiento sintáctico se refiere al entendimiento de detalles dependientes al uso de los dispositivos. El conocimiento semántico, incluye información conceptual concerniente a la aplicación -objetos y acciones del dominio de la tarea- y al uso general de los sistemas interactivos.

Por ejemplo, puede existir el caso de una persona que sea experta en conceptos computacionales pero novata en conceptos propios de la aplicación.

De acuerdo al grado de conocimiento semántico y sintáctico que el usuario posea, se puede realizar una clasificación general de los mismos en tres categorías:

Clasificación SSOA de Usuarios	Explicación
Usuario Novato	■No tienen ningún conocimiento sintáctico. ■Cuentan con un conocimiento semántico pobre sobre el uso de sistemas interactivos. ■Medianamente tienen un entendimiento de la aplicación.
Usuario Intermedio	■Tienen buen conocimiento semántico tanto del uso de sistemas de software como del dominio de tareas. ■Presentan inconvenientes a nivel de cómo se llevan a cabo las tareas en el sistema, con problemas en recordar detalles de conocimiento sintáctico.
Usuario Experto	■Están muy familiarizados con todos los aspectos sintácticos y semánticos del sistema.

Tabla 2.3: Modelo SSOA

Es necesario aclarar que los conceptos de conocimiento sintáctico y semántico tienen una correspondencia directa con las capas que presenta el software. El corazón funcional procesa información semántica mientras que el procesamiento de información sintáctica pertenece a las capas propias de la interfaz del usuario.

En la figura siguiente se muestra cómo el conocimiento semántico es transformado en formas y especificaciones sintácticas para ser presentadas y manipuladas por el usuario final.

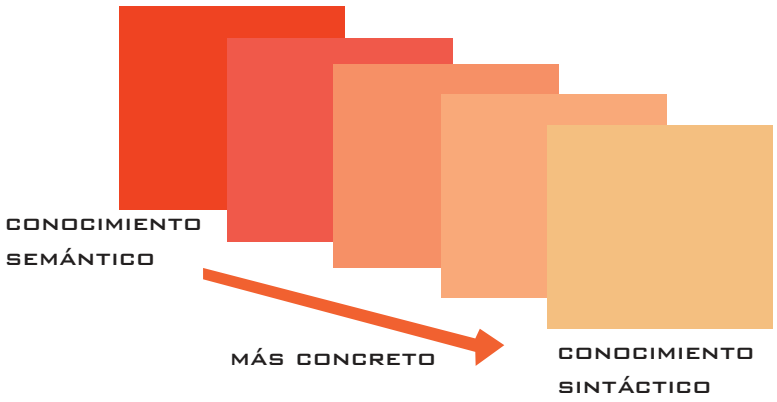


Figura 2.3: El conocimiento semántico y sintáctico de acuerdo a las capas de abstracción del software de la Interfaz del Usuario.

Importante

El modelo SSOA, puede ser aplicado y considerado por los diseñadores, si los mismos no pueden realizar una modelización y clasificación de usuarios finales del sistema, más detallada o precisa.

V - Etapa de Diseño de la Interfaz

El objetivo de la etapa de diseño es expresar los resultados abstractos del análisis, en especificaciones computacionales tangibles. El propósito inicial no es tan sólo solidificar el análisis, sino también estimar su validez. Las especificaciones son necesarias para estimar tanto las posibilidades técnicas, como los costos de desarrollo.

Si el modelo conceptual del usuario es considerado por el diseñador, entonces podrá definir para la interfaz, un claro mapeo entre las tareas y los objetos semánticos con los representados en la pantalla, y permitirá que el conocimiento requerido al usuario para utilizar el sistema, sea lo más mínimo posible.

Más aún, si el diseñador trabaja no sólo con el modelo del usuario sino con los demás estudios realizados, tanto sobre el ambiente como sobre la aplicación, podrá lograr soluciones a nivel de interacción muchas más adecuadas y coherentes.

Sabiendo que, por lo general, no se podrá contar con especificaciones a nivel de interfaz completas, entonces en esta fase, el diseñador deberá partir con un proceso de análisis, integración, refinación, conjugación de los modelos que se crearon en la etapa de análisis, en pos de lograr una aproximación inicial de la misma.

Entonces, en este proceso de diseño se integran y se refinan los modelos del usuario, del sistema y del entorno, adquiridos en la fase anterior.

La interfaz, especificada a través de diferentes medios como prototipos, escenarios, expresará las nociones extraídas de dichos modelos, y permitirá la evaluación por parte de los usuarios, de la usabilidad y legibilidad del diseño.

Como consecuencia del proceso de evaluación de la interfaz, se puede requerir modificaciones tanto en el diseño de la misma como en los modelos definidos. En estos casos, es el rol del diseñador de la interfaz, refinar el modelo del usuario y los aspectos de diseño, mientras que, si se requiere modificar algún aspecto de la funcionalidad, será necesario acudir al diseñador del sistema.

Este constante refinamiento, es influenciado por las interacciones que el usuario va teniendo con el sistema en el proceso de evaluación, formando un proceso en espiral, donde nuevas modificaciones permitirán volver a redefinir etapas anteriores.

En cada momento, se re-analiza la interfaz diseñada, se re-diseña y se re-evalúa analizando los costos y beneficios de la interacción. En cada paso, se puede descubrir nueva información que fue desconocida o pasada por alto en pasos previos, que debe ser incluida afectando decisiones de diseño ya tomadas.

También, adelantarse a las siguientes etapas puede ser adecuado. Por ejemplo, investigar sobre características del hardware, nivel de servicios de las herramientas de software disponibles, es muy útil para determinar la factibilidad y los costos del diseño especificado.

En definitiva, la tarea del diseñador es un proceso de revisión iterativa refinando el paso actual como los anteriores, y además, puede incorporar conocimientos de pasos posteriores, que influirán en sus decisiones actuales.

iseño, las fuentes de información consideradas pueden ser modelos realizados en la etapa de análisis como también versiones previas de diseño desarrollados y evaluados anteriormente:

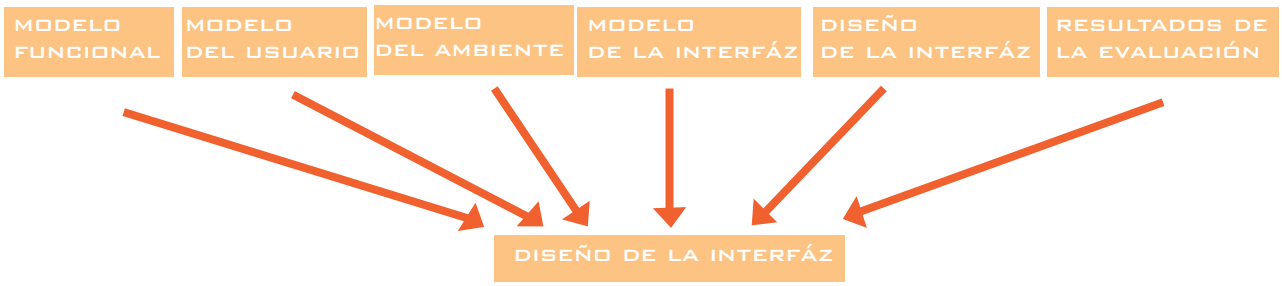


Figura 2.4: Fuentes de información de la Etapa de Diseño.

En este figura se visualizan distintas vertientes de información, que debe considerar el diseñador de la interfaz en cada versión prototípica de su diseño.

Esto implica, como consecuencia, una constante comunicación entre los diferentes roles involucrados. Hay una interacción cíclica entre el diseñador del sistema con el diseñador de la interfaz, éste a su vez con el programador de la interfaz, además se puede incluir al rol del evaluador que conjuntamente con el usuario es fuente de modificaciones.

En todo este contexto y arquitectura de trabajo, el diseñador tendrá básicamente que diseñar en forma gradual y decidir en forma metódica, sobre múltiples cuestiones de la interfaz, a saber:

Decisiones de diseño	Explicación
Determinar el estilo de interacción	■Es necesario elegir qué estilo de interacción se proveerá en el sistema. Que puede ser un diálogo secuencial, con manipulación directa, concurrente o basado en comandos. ■También puede ser combinaciones de tipos diferentes de diálogo.
Decidir sobre el tipo de interfaz a proveer	❖Analizar si el sistema contará con una interfaz textual, visual, icónica o con características híbridas. Si brindará aspectos de adaptación, evolución, y/o de inteligencia. ■También, dependiendo del tipo de aplicación, si es una interfaz para un sitio de Internet, o para aplicaciones móviles o para sistemas telefónicos.
Decidir cuestiones de visualización	■Debe diseñar los aspectos de las pantallas y representación de la información, de los estados, de las entradas y salidas. ■Debe decidir sobre los paradigmas y objetos de interacción. ■También, se debe analizar sobre el diseño de las ayudas.
Analizar aspectos de hardware	■Se tiene que decidir sobre los elementos de hardware que se necesitarán, destinados a la interacción entre el hombre y la máquina. ■Si no son tradicionales, por ejemplo, una alfombra que detecte la presencia de un usuario, se debe diagramar todo el bosquejo y especificar los requisitos necesarios para que un ingeniero la pueda crear. ■Lo mismo, si se pretende utilizar por ejemplo, algún sistema biométrico de voz en aplicaciones móviles.
Especificar cuestiones de comportamiento	■Es responsabilidad del desarrollador de la interfaz, no sólo especificar cuestiones de visualización, sino detallar la forma en que la interfaz se va a comportar frente a los usuarios. ■Se debe planificar el comportamiento de los objetos, el feedback, el control de la interacción, el sistema de asistencia, el manejo de errores. ■Se debe analizar la correlación de las diferentes pantallas del sistema, modos de activación y cierre, o el mapa de navegación de páginas en caso de ser un sitio Web. ■Se debe planificar la adaptación que se desee proveer o cuestiones no tradicionales como aspectos inteligentes, de inferencia, de evolución. ■Se debe plantear también, la relación de la interfaz con la aplicación, definiendo formas de integración e invocación de funciones.
Diseñar las funcionalidades y servicios a nivel de interfaz	■Todos los aspectos dinámicos y comportamiento de la interfaz deberá ser programado, por lo tanto requerirá un previo proceso de diseño. ■En el caso de las interfaces con signos de adaptación, se debe analizar la lógica de la adaptación y su aplicación, como también su mecanismo de recolección de la información del usuario y de sus sesiones.
Diseñar el almacenamiento externo en bases de datos o en archivos	■Mucha información de la interfaz debe ser registrada para poder ser usada en las próximas sesiones. Por ejemplo, información de algún estado que el usuario quiere posteriormente reanudar, acciones que fueron suspendidas. ■También se puede almacenar perfiles de los usuarios y sus características. Preferencias, hábitos, comportamiento del usuario.

Tabla 2.4: Decisiones de Diseño.

Por lo que se concluye de la tabla anterior, la complejidad del desarrollo de la interfaz reside fundamentalmente en la heterogeneidad y multiplicidad de las cuestiones que deben decidirse en el diseño, además de la arquitectura prototípica en la que se trabaja y de la coordinación de los diferentes roles que participan en ella.

Recordar

❒El proceso de refinamiento iterativo de la interfaz, hace que el diseñador requiera de representaciones de diseño flexibles.
❒Los conceptos de la aplicación no están expuestos a grandes cambios pero, la expresión o visualización de los mismos es muy probable que sí vaya adquiriendo diferentes alternativas.

Características del Proceso de Diseño

La etapa de diseño es un proceso complejo, donde pueden visualizarse tres etapas importantes, que son:

■La divergencia: está determinada por el espacio de búsqueda de todas las posibles soluciones. La parte esencial para esta etapa, es la colección de información necesaria para el proceso de diseño y la investigación de posibles alternativas. Se recolecta información de los modelos de usuarios analizados, del ambiente, de la aplicación, se investiga en el mercado propuestas similares, los estándares existentes, entre otras cuestiones. La etapa es puramente exploratoria y mucha de la información que se maneja aquí, es incierta.

■Etapa de Transformación: aquí la información, los datos, objetivos propuestos en la etapa anterior, son utilizados para disminuir la complejidad del problema. En esta etapa, se sintetiza la información, se descartan datos irrelevantes, se refina y se postula un nuevo delineamiento del diseño. Es importante, investigar sobre normativas de diseño o recomendaciones publicadas, que sean apropiadas para el caso y poder aplicarlas en el bosquejo.

■Etapa de Convergencia: la misma apunta a determinar una solución de diseño bien definido, a partir del estado inicial inestable.

Estas etapas, pueden observarse en el siguiente gráfico:

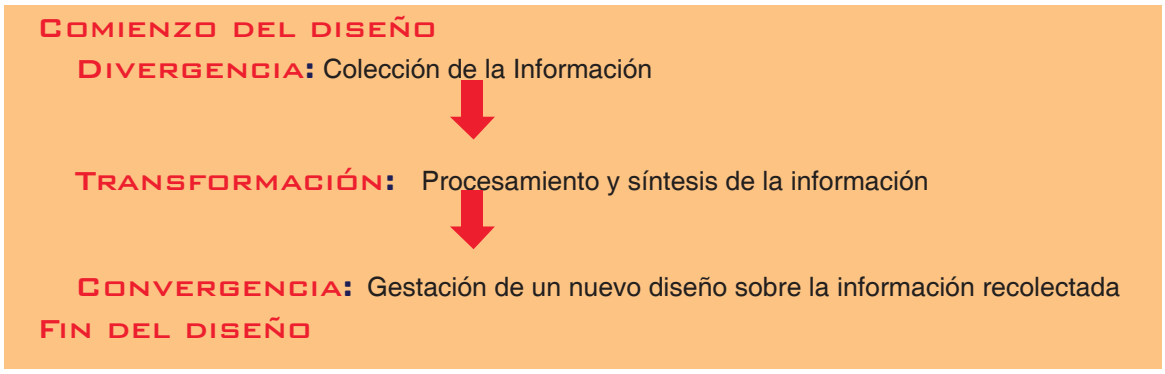


Figura 2.5: Etapas involucradas en el Diseño de la Interfaz.

La transformación de los problemas más complejos a más sencillos se logra, recolectando información útil sobre el espacio de diseño. La recolección de ejemplos, estereotipos, estándares permite reestructurar el problema en nuevas formas, disminuyendo así la complejidad del problema inicial y su nivel de incertidumbre.

Obviamente, el proceso de diseño no es tan simple y directo, por ejemplo, durante la etapa de convergencia puede suceder que ciertos aspectos queden sin resolver, por lo que será necesario reformular las etapas previas y comenzar a recaudar mayor información.

Importante

■Las etapas del proceso de diseño, denominadas divergencia, transformación y convergencia, se desarrollan cada vez que se rediseñe la interfaz, o sea que se efectúan en cada iteración del ciclo de vida prototípico.

VI - La interfaz y la Prototipación

Los métodos convencionales para el desarrollo de sistemas interactivos tienen una naturaleza analítica, exacta y presentan un proceso secuencial, donde las etapas de desarrollo de suceden una tras otra, una vez que estén completas y correctas.

Esta metodología tradicional, puede ser adecuada para el desarrollo de la parte de la aplicación del sistema, pero presenta una gran rigidez cuando se la considera para desarrollar la interfaz del usuario, porque deben incorporarse los factores humanos y es necesario realizar testeos con los usuarios, desde el principio.

El diseño de la interfaz constituye un proceso permanente e iterativo de corrección con la participación de los usuarios, y esta esencia iterativa impone la necesidad de cambios en el tradicional y lineal ciclo de vida del software.

Es necesario una propuesta metodológica que permita trabajar en forma experimental, debido a que no existe una teoría suficientemente firme que prevea y asegure, sobre una interfaz que aún no esté construida, cuáles serán concretamente los posibles efectos que ella puede ocasionar sobre los distintos usuarios, o cómo será el comportamiento de los mismos frente a las futuras interacciones con el sistema. Más aún, cuando sobre dicha teoría, se deban sustentar la fundamentación del diseño y del resto de las etapas de desarrollo.

Dentro de la comunidad de HCI, y como forma de sustentar el diseño centrado en el usuario, se introduce a los prototipos como uno de los conceptos esenciales para el gerenciamiento de las interfaces del usuario.

Se establece la necesidad de desarrollar prototipos sobre la componente de interacción, para ser evaluados, reconstruidos y refinados hasta que la interfaz final haya sido diseñada. El usuario se ve como la parte integral de todo este proceso.

La Prototipación es un proceso que se caracteriza por una secuencia de prototipos que son construidos hasta que se obtenga un buen entendimiento de los requerimientos del software. Es utilizada cuando la forma que debe tomar la interacción hombre máquina no está inicialmente definida [Pressman, 1991]. A través de ella, los ingenieros en software pueden adquirir una idea clara del producto a desarrollar.

Cuando los especialistas en HCI desarrollan los prototipos, se convierten en contribuyentes claves para la conformación de la fase de requerimientos de la interfaz como también, posiblemente de la aplicación.

El proceso prototípico puede involucrar la gestación de varias versiones del sistema, donde cada una de ellas constituye un ambiente para la corrección, evaluación y testeo, cuyos resultados servirán y se utilizarán para que la versión siguiente sea más detallada, completa y correcta que la anterior, teniendo en cuenta la perspectiva del usuario.

Dentro de este proceso, cada vez que se genera una versión prototípica de la interfaz, se debe redefinir el prototipo utilizado, modificarlo y volverlo a evaluar. Por lo tanto, el prototipo, a medida que se va corrigiendo y perfeccionando, debe cumplimentar las siguientes etapas que se ilustran a continuación:

Figura 2.6: Etapas de un Prototipo.

Si uno analiza las etapas de desarrollo de los prototipos, puede encontrar una correspondencia con las fases de la ingeniería de la interfaz. La etapa de Definición del prototipo, tiene una estrecha relación con el proceso de especificación de requerimientos. La etapa de Construcción, abarca las etapas de diseño y de representación del bosquejo de la interfaz. En la etapa de Ejercitación, se incluyen las etapas de implementación y, posiblemente, las de conversión e instalación. Por último, la etapa de Evaluación, presenta ciertas analogías con las etapas de testeo de usabilidad y entrenamiento de la interfaz del usuario, sólo que en estos casos estamos trabajando con versiones incompletas del sistema.

Características de la Prototipación

En su concepción más amplia, la Prototipación puede ser considerada una metodología de desarrollo de software [Reilly, J.P., 1996], mientras que, para otros autores es considerada un método o técnica de representación de diseño de la interfaz. Este es el caso de la Ingeniería de Usabilidad [Nielsen, J.,1994], donde se la incluye dentro de la Fase de Diseño.

En ambos casos, la misma permite desarrollar una visión prematura del sistema, donde se puntualiza en un principio, la visualización, la interacción y el comportamiento global del mismo, abstrayéndose de las características computacionales y proveyendo un marco para que los futuros usuarios del sistema puedan observar el comportamiento del mismo e interactuar con él antes de que sea finalmente desarrollado.

El proceso de testeo y de verificación en el desarrollo del software es muy importante, porque permite analizar si los requerimientos de especificación iniciales son satisfechos por el sistema.

Como dicho proceso, clásicamente se encuentra como una de las etapas ya finales del ciclo de vida tradicional del software, los problemas o errores encontrados en esos términos son muy difíciles y costosos de solucionar, pues requieren modificaciones en todas las etapas anteriores, que están concluidas casi en su totalidad.

En cambio, con la Prototipación, hay una concurrencia entre las etapas del desarrollo del sistema, la interacción del usuario con una versión prematura e incompleta de la interfaz, permite evaluar anticipadamente los requerimientos iniciales de la misma. De esta manera, las modificaciones o cambios requeridos, podrán ser incorporados en las próximas versiones del diseño.

La Prototipación puede ser conceptuada como una metodología, pues determina un marco en donde todas las etapas de desarrollo de la interfaz, se suscitan. A través de ella, se representa el diseño, se evalúa, se rediseña, se implementa parcialmente, y hasta, se descubren los requerimientos del usuario.

Este último aspecto, es una de las características más importantes de la Prototipación como proceso metodológico y es que, a través de ella, se va construyendo el modelo del sistema a desarrollar. Cuando se plantea el problema en la etapa de requerimientos, no requiere la realización de un modelo completo. Tiene, por filosofía, que éste nunca alcanzará la completitud hasta que no sea evaluado y analizado, rediseñado iterativamente, con la participación directa y activa del usuario y del resto de los roles.

Entonces, es una metodología que puede partir desde un modelo inicial incompleto hasta posiblemente, ausente. Los requerimientos, que constituyen la base para la correcta construcción de los sistemas, son desarrollados y completados mediante la propia participación y experiencia que los usuarios van adquiriendo con el prototipo.

El modelo es evolutivo, dinámico, se va completando en las sucesiones versiones que se generan del sistema, y además en las etapas concurrentes de análisis, diseño e implementación, que están muy unidas y donde la realimentación es una constante.

Entonces, el mismo modelo es utilizado para diseñar, implementar y testear el sistema pero, a su vez, los resultados de estas etapas permitirán modificarlo para completarlo aún más. Así sucesivamente, hasta alcanzar la completitud del mismo.

Ventajas de la Prototipación

Una de las ventajas atribuidas a esta metodología, es una mejora importante en la comunicación entre los usuarios del sistema y el equipo que desarrolla el software.

Se evitan muchos problemas de comunicación, desentendimientos, malas interpretaciones que surgen entre ellos debido fundamentalmente a que cuentan con profesiones, formaciones o conocimientos esencialmente distintos.

Al usuario le cuesta mucho desde su perspectiva, formular sus ideas y comunicar las especificaciones completas del sistema, y muchas veces, esas descripciones son traducidas a términos computacionales por los analistas o diseñadores, en forma incorrecta.

Además, los sistemas desarrollados mediante la Prototipación tienden a garantizar la calidad de uso y satisfacción de los usuarios, en el sentido que los mismos usuarios fueron los que participaron en las distintas evaluaciones en donde se analizaron, entre otras cuestiones, si los requerimientos iniciales fueron considerados.

Otra de las ventajas importantes, es que reduce el costo total del desarrollo del sistema. Esto está muy relacionado con la capacidad que ofrece de adelantar y superponer el proceso de evaluación y testeo, con el resto de las etapas. Así, los problemas o errores encontrados, pueden ser detectados a tiempo y solucionados, mientras el sistema está desarrollándose.

Tipos de Prototipos

Los prototipos se pueden clasificar de diferentes maneras, de acuerdo a su tipo de evolución, medios utilizados, técnicas empleadas, grado de fidelidad.

De acuerdo a su tipo de evolución, los prototipos pueden ser revolucionarios o evolutivos:

Clasificación según Tipo de Evolución	Explicación
Prototipo Revolucionario	■El prototipo es diseñado, construido, evaluado y es utilizado hasta que se alcanza la completitud del modelo del sistema, una vez que esto sucede se comienza desde cero con la implementación del producto final. ■Es un prototipo que se descarta, una vez que sea utilizado como herramienta representativa del diseño de la interfaz del usuario.
Prototipo Evolutivo	■El prototipo evoluciona y se va optimizando mediante las modificaciones iterativas hasta alcanzar la implementación completa del sistema de software final. ■En las versiones finales del prototipo se deberán incluir los algoritmos funcionales de la componente de aplicación.

Tabla 2.5: Clasificación del Prototipo según grado de Evolución.

Si tenemos en cuenta las técnicas empleadas en el proceso de construcción de prototipos, se pueden distinguir los siguientes tipos:

Clasificación según Técnicas empleadas	Explicación
Prototipos MockUps	■Son prototipos muy sencillos que únicamente describen interacciones particulares con el sistema. Permiten implementar un escenario, una situación particular de una interacción o un diálogo con el sistema. ■Consiste de una serie de diseños de pantallas presentadas al usuario en papel o por computadora, a través de las cuáles son analizadas y evaluadas heurísticamente. ■Es la técnica de prototipación más simple, menos costosa y es utilizada comúnmente en etapas iniciales.
Escenarios	■Pueden ser impresos en papel o en video, describen la situación actual de un proceso de trabajo. ■Son utilizados en las entrevistas realizadas a los usuarios.
Glosarios	■Son documentaciones que especifican la terminología del lenguaje profesional del dominio de la aplicación. ■Ofrecen referencias a objetos relevantes usados en el dominio de aplicación.
Manuales del usuario tempranos	■Proveen documentación y detalles específicos de las operaciones provistas y reflejan los diferentes modos de uso.
Herramientas de software	■Basado en una técnica de representación formal, define una versión ejecutable del sistema. ■Hay prototipos para interfaces representadas por diagramas de transición y estado, hay prototipos basados en gramáticas BNF ó en mecanismos orientados a eventos.

Tabla 2.6: Clasificación del Prototipo según Técnicas utilizadas

Cualquiera de estas técnicas permiten de diferente manera, la realización de evaluación de usabilidad de la interfaz, se presta a un juego de comparaciones y testeos basados en un conjunto de criterios u objetivos ya preestablecidos.

Además, los prototipos pueden realizarse utilizando medios como el papel o mediante un graficador, procesador de textos, software presentacionales como PowerPoint, software de instrucciones asistida como Adobe Director, sitios para diseño y testeo de bocetos o wireframes como Cacoo, Pencil, Balsamik. Existen prototipos programados mediante lenguajes de programación como Java, Delphi, Visual Basic, Toolbook, entre otros, donde se permite además representar el control de diálogo.

Finalmente, los prototipos se pueden clasificar por su grado de fidelidad. La fidelidad depende de ciertos factores como ser, del medio utilizado para la representación prototípica, de las características representadas en la interfaz, del grado de completitud funcional que presenta, como ser, si incluye validaciones, ayudas, mensajes de error, funciones sintácticas, entre otros.

Con respecto a la fidelidad de los prototipos, los mismos pueden ser:

Clasificación según Fidelidad	Explicación
Prototipo de Baja Fidelidad	■Son una sucesión de pantallas estáticas en papel o por computadora. ■La sucesión de pantallas y el flujo, efecto de acciones del usuario deben ser contadas textualmente o representadas a mano. ■Son más fáciles de realizar, más rápidos pero menos eficientes.
Prototipo de Alta Fidelidad	■Son considerados especificaciones ejecutables del diseño. ■Describen el comportamiento y los aspectos dinámicos de la interfaz. ■Representan la interacción, el control del diálogo y el estilo del diálogo. ■En ellos se ve reflejado las dependencias en vivo, el asincronismo, el flujo de control de aberturas y cierres de contextos, el feedback, que incluya validaciones, ayudas y mensajes de error. ■Son más difíciles de realizar, pero más expresivos, más verídicos y mejores en la comunicación y presentación.

Tabla 2.7: Clasificación del Prototipo según Fidelidad.

Importante

■En un proceso de ingeniería de la interfaz del usuario, se pueden utilizar prototipos de diferente fidelidad.

■Los prototipos de baja fidelidad, pueden ser utilizados al principio para presentar ante el usuario diferentes propuestas de diseño de la interfaz. Cuando el usuario elige una, se comienza entonces con el ciclo de vida prototípico completo, desarrollando las sucesivas versiones en alta fidelidad.

VII - Etapa de Implementación

En esta sección, se va a describir cómo es la estructura del software necesaria para generar una interfaz del usuario y cómo está compuesta. Además, se van a analizar, las herramientas de desarrollo que surgieron para construir la componente de interacción de un sistema.

Una vez que esté desarrollado el diseño de la interfaz, que haya sido analizado, testeado, aceptado por los usuarios del sistema, se procede a la etapa de implementación. La programación de una interfaz del usuario alcanza usualmente, grandes proporciones de código, ocupando más del 50% de la codificación total del sistema.

La complejidad no reside sólo en la extensión del código, sino en cómo se estructura el software necesario, para generar una interfaz del usuario y cómo está compuesta.

Una manera de estructurar el software de la interfaz, es definirla como una jerarquía de máquinas abstractas, cada una proveyendo servicios determinados a las capas superiores y requiriendo servicios de las capas inferiores.

La modularización del software en capas de abstracción, permite tener independencia entre las capas, respecto a las implementaciones de cada una de ellas, tener un mecanismo de protección y ocultamiento de la información que cada una maneja.

Así, se define una taxonomía del software de la interfaz del usuario en donde se divide al sistema interactivo, en los siguientes niveles:

Figura 2.7: Software de la Interfaz del usuario dividido en diferentes capas de abstracción

Las capas de abstracción del software, entonces son:

Capas del software	Explicación
Manejadores de dispositivos	■Manejan los requerimientos específicos de entrada y salida, realiza el control del dispositivo físico. ■Por ejemplo, el manejador de dispositivos envía instrucciones a nivel de pixeles para mostrar el cursor en una determinada posición.
Sistema de ventanas	■Controla los recursos reales involucrados en la interacción, maneja los recursos de dispositivos físicos de entrada y salida. ■Provee las abstracciones utilizadas para la interacción. ■Por ejemplo, el sistema de múltiples ventanas.
Objetos de interacción	■Capa que maneja y controla los objetos de interacción con el cuál el usuario interactúa. ■Es una entidad que el usuario puede percibir y manipular con dispositivos de interacción físicos tales mouse, teclado. ■Un objeto de interacción incluye una componente de presentación (output) y una componente de interacción (input).
El controlador del diálogo	■Es una porción del software de la interfaz que controla el medio utilizado y la secuenciación de las interacciones del usuario. ■Determina la lógica de los eventos. ■Por ejemplo, determina cuándo un objeto de interacción, está habilitado para el usuario. ■También, controla el estilo de la interacción, por ejemplo, controla multidiálogos abiertos en una caja de diálogos.
Corazón funcional	■Pertenece a la aplicación. Permanece totalmente ignorante de la forma en que sus estructuras de datos y funciones son expuestas al usuario, vía los objetos de interacción. ■Debe ignorar tanto el medio usado para la entrada y salida, como la forma específica en la cuál la información es trasmitida. ■Por ejemplo, no debería saber si en un sistema de ventanas se utiliza el audio o gráficos en la salida, o qué lenguaje es usado para presentar el texto, ó si las entradas fueron ingresadas por teclado o con reconocimiento de voz.

Tabla 2.8: Capas del Software.

Cada capa, funciona como una máquina abstracta, que provee ciertos servicios. El usuario de estos servicios se denomina cliente, y el proveedor de los mismos se denomina servidor. Cada capa intermedia, debe controlar una porción del total de la interfaz, y debe proveer un mapeo entre las capas inferiores y superiores a ella.

Entonces, proporcionan al diseñador del diálogo, una manera de especificar la interfaz en un alto nivel de abstracción, automatizando todo el proceso de traducción al código resultante y facilitando todo el desarrollo del ciclo de vida prototípico de la interfaz.

Herramientas para el Desarrollo de la Interfaz

Como se dijo anteriormente, crear una buena interfaz del usuario para un sistema es una tarea difícil. El software para soportar estas interfaces, usualmente es muy extenso, complejo y difícil de refinar.

Además, como la ingeniería de una interfaz del usuario, generalmente consiste en una repetición del ciclo de diseño, prototipación y evaluación, hasta que los requerimientos del usuario sean satisfechos, esta iteración encarece mucho el proceso de desarrollo.

En pos de reducir el alto costo de diseño e implementación de la misma, surgieron softwares dedicados al gerenciamiento de las interfaces.

Una herramienta de software para el desarrollo de una interfaz del usuario puede ir desde una simple rutina, a un ambiente completo para el análisis, representación, diseño, implementación y evaluación de la interfaz.

Surgen, así los denominados “UIT, User interface Toolkits”, que son librerías de rutinas que pueden ser invocadas para implementar ciertos aspectos de la interfaz, de bajo nivel.

Particularmente, estas librerías proveen la implementación de algunos objetos de interacción como calendarios interactivos, gráficos estadísticos, con el manejo de sus dispositivos físicos asociados. Los toolkits, solamente cubren la etapa de la implementación de alguna parte o componente de la interfaz del sistema.

Por otro lado, se encuentran los “UIMS, sistemas para el gerenciamiento de las interfaces del usuario”. Estos son completos ambientes interactivos, generalmente basados en un conjunto de herramientas para proveer e integrar la mayoría de las etapas del desarrollo de la interfaz.

En particular, los UIMS pueden soportar la mayoría de las fases del ciclo de vida de la interfaz del usuario, como su diseño, prototipación, ejecución, evaluación, modificación y mantenimiento.

Estos sistemas, administran los detalles de la pantalla, sus entradas y salidas asociadas y la interacción con el resto del programa. Proveen técnicas de interacción, permiten manipular la secuencia de eventos, el control del diálogo, más todo el manejo de dispositivos de hardware.

Además, permiten que todo este proceso de desarrollo lo pueda realizar los propios desarrolladores del diálogo, quienes pueden no ser puramente programadores.

Importante

El trabajo de los UIMS, tienen por objetivo simplificar el desarrollo de las interfaces del usuario gráficas. Lo mismo ocurrió en la década del ´70 con el surgimiento de administradores de base de datos, DBMS que contribuyen en la gestión de toda información almacenada en bases de datos.

La siguiente figura, describe las componentes de un UIMS:

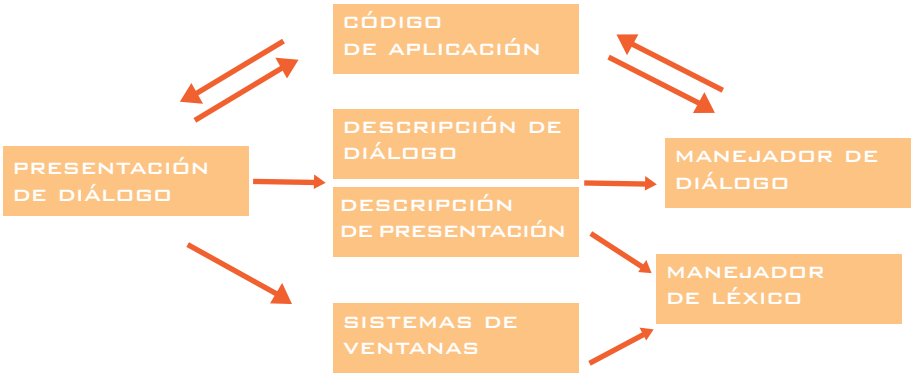


Figura 2.8: Componentes de un UIMS.

El UIMS separa el código, de la lógica de la aplicación de la de la interfaz del usuario, introduciendo un controlador de diálogo entre ambos. Es posicionado al tope del paquete gráfico de las ventanas, pero cabe aclarar que hay implementaciones de UIMS, que proveen su propio sistema de ventanas.

El manejador léxico, procesa las entradas del usuario convirtiendo las entradas a eventos lógicos y los pasa al controlador del diálogo. Este controlador es la descripción del diálogo, es la interpretación semántica de las secuencias de eventos lógicos, que recibe desde el manejador léxico. El manejador del diálogo luego produce comandos para invocar las funciones de la aplicación correspondientes.

En el proceso de la salida, el módulo de presentación de datos acepta las notificaciones, establece la descripción perceptiva de la interpretación y realiza llamados a funciones del sistema de ventanas o del paquete gráfico.

La interfaz semántica, entre el manejador del diálogo y la aplicación, permite independencia del diálogo que facilita el desarrollo y la consistencia.

Desde el punto de vista de la usabilidad, la arquitectura UIMS, permite aplicaciones interactivas, con control externo fluyendo desde y hacia la interfaz.

Los constructores de interfaces gráficas basados en UIMS, son la primer generación de herramientas que focalizan el diseño centrado en el usuario. Permiten especificaciones o modificaciones del diálogo, rápidas y visuales. La capacidad de programación visual, es especialmente adecuada para soportar diseño iterativo. Soportan el desarrollo y refinamiento de interfaces, con editores visuales y componentes reusables.

VIII - La importancia de la Evaluación de la Interfaz

La evaluación, juega un papel importante en la ingeniería de interfaz, en todas las etapas del ciclo de vida. La evaluación puede servir para varios propósitos, dependiendo en la madurez del sistema.

Durante la investigación y desarrollo, la evaluación centrada al usuario, es muy importante para descubrir y validar conceptos iniciales. El equipo de desarrollo necesita feedback creíble para testear la adecuación del diseño y saber cómo mejorarlo. Los desarrolladores además, necesitan comparar alternativas de diseño.

Durante la evolución del diseño, técnicas de evaluación de usabilidad pueden aplicarse para estimar requerimientos de entrenamiento. Además, se requieren datos confiables para determinar cuando el diseño iterativo ha alcanzado los objetivos de usabilidad.

Es importante, para efectuar los diferentes procesos de evaluación, la identificación precisa de los usuarios potenciales del producto y el poder establecer una buena comunicación con ellos.

Por lo general, se trabajan con muestras de más de 20 personas, que deben ser lo más representativas posibles a la comunidad de usuarios a la que está dirigido el sistema. La contratación de usuarios para cumplimentar el rol de evaluadores, debe ser proporcional y compatible, a los perfiles detectados en la modelización de usuarios.

Hay dos tipos principales de métodos de evaluación, que han emergido en el área de HCI, que difieren en el tipo de evaluadores que intervendrán en ella: Los Métodos de Indagación y los Métodos de Inspección. Ambos pueden ser utilizados en un mismo proyecto y en diferentes etapas de desarrollo.

Métodos de Indagación

En este tipo de evaluación, son los potenciales usuarios del sistema quienes llevarán a cabo el proceso evaluativo. Ellos, cumplirán el rol de evaluador de la interfaz.

Los métodos de indagación, son los más utilizadas en el ciclo de vida prototípico, tanto para conformar la etapa de requerimientos, como en los tests de mercado.

La técnica de evaluación de indagación más importante en el área de HCI, es el “Test de Usabilidad” o “Test Empírico del Usuario”.

El test empírico del usuario, involucra la recolección de datos sobre la interacción del mismo con el bosquejo de diseño, durante el desarrollo de escenarios de tareas.

Los estudios empíricos y experimentos en laboratorios, dan información precisa sobre cuestiones específicas. Pueden ser utilizados para comparar la efectividad de alternativas de diseño, identificar y diagnosticar problemas de usabilidad, o para validar que el diseño ha satisfecho sus objetivos.

Para generar feedback, los usuarios generalmente prueban prototipos y participan en entrevistas sobre el contexto de su ambiente de trabajo.

Una definición de trabajo de usabilidad para testeos empíricos, puede ser determinado en términos de eficiencia y éxito de la performance de las tareas que el usuario debe realizar en el sistema.

Generalmente, estas mediciones reflejan el grado de usabilidad y productividad registrando, variables tales

como tiempo empleado en tareas, errores, patrones de comportamiento durante las sesiones de testeo. Los datos subjetivos de los usuarios también son informativos para chequear satisfacción y diagnosticar problemas de usabilidad.

El proceso de entrenamiento, es un aspecto crítico de usabilidad. Esto se debe a que los usuarios, ya en sus primeras experiencias en la interacción con el sistema, intentan y se esfuerzan en aprender a operarlo.

Se presentan problemas de memorización, evidenciado por usuarios que repetidamente intentan recordar los comandos y procedimientos, agravándose la situación en aquellos sistemas que no han sido diseñados cuidadosamente. Dificultades en el entrenamiento puede también reflejar un número de otros problemas generales, como la desorientación.

Los reportes verbales pueden ser también registrados mientras el usuario está efectivamente evaluando el sistema. La ventaja de estas técnicas de "pensar en alto" es que los usuarios permiten a los diseñadores conocer lo que ellos piensan, sus problemas, sus temores respecto al uso del sistema.

Los reportes verbales se utilizan en combinación con los datos recogidos en la observación del comportamiento, así se obtiene un alcance mayor de información. Esta técnica es muy útil para evaluar aspectos de diseño de alto nivel, por ejemplo su mapeo con el modelo conceptual diseñado.

Las técnicas de recolección de datos, para tests empíricos de usuarios han sido descriptos por Schneiderman [Schneiderman, 1988]. Múltiples cámaras de video son generalmente utilizadas para registrar la performance de los usuarios frente a la realización de tareas y para obtener comentarios de los usuarios. Pasar videos de usuarios experimentando problemas de interacción son pruebas suficientes y poderosas para convencer a los programadores de la importancia de optimizaciones de diseño.

Métodos de Inspección

En los métodos de inspección, se contrata a evaluadores externos que sean expertos en HCI, para cumplimentar el proceso de testeo de la interfaz. Por los costos que esto conlleva, por lo general, se los organizan, una vez que se tienen versiones prototípicas bastantes avanzadas y completas.

Estos métodos, constituyen revisiones de diseño de software y son comunes en proyectos de desarrollo, como también, para establecer estándares en interfaces del usuario.

El tipo más popular de técnicas de inspección es el llamado "Evaluación heurística", en donde participa un pequeño equipo de expertos que independientemente examinan el sistema.

Más que chequear el sistema sobre estándares inflexibles, ellos buscan las partes de la interfaz donde se han desobedecido algún principio de diseño. La evaluación heurística ha mostrado ser la mejor técnica para la detección de problemas de interacción.

Los expertos concluyen con un informe donde se puntualizan todos los sectores de la interfaz que presentan fallas a nivel de usabilidad, se detalla la normativa que se infringe y pueden incluir posibles soluciones o alternativas de depuración.

Importante

- Las evaluaciones pueden ser de campo, de laboratorio o de mercado.
  - Las evaluaciones de campo, significa que se probará el sistema frente a usuarios en el lugar de trabajo de los mismos.
  - En las de laboratorio, los usuarios deben acudir a la oficina donde los desarrolladores de software están trabajando.
  - En la de mercado, la evaluación se efectúa una vez que el sistema está siendo usado por los usuarios, estando en modo productivo.

Ejercitación

A - Cite al menos 5 características de la interfaz de usuario que justifiquen la siguiente afirmación:

> La metodología de Prototipación es adecuada para el diseño e implementación de la interfaz del usuario.

_____

_____

_____

_____

_____

B - Cite al menos 3 tareas, que debe realizar un desarrollador de software previo al diseño de la interfaz.

_____

_____

_____

_____

_____

C. Dadas las siguientes afirmaciones, responder Verdadero o Falso.

- Los testeos empíricos del usuario se llevan a cabo a través de expertos que evalúan la interfaz según estándares de diseño. V F
- El feedback generado por los usuarios durante las evaluaciones heurísticas refleja el grado de usabilidad y productividad del sistema. V F
- El lenguaje de programación Java provee su propio paquete para el diseño de componentes de la interfaz. V F
- En la etapa de diseño se verifica lo que el usuario conoce del dominio de la aplicación y del uso de la computadora. V F
- Las evaluaciones de campo permiten evaluar la interfaz del sistema mientras los usan usuarios reales en entornos controlados. V F

Referencias

❒Baecker, R. M., Grudin, J., Buxton, W. A. S., and Greenberg, S. (1995). Readings in Human-computer Interaction: Toward the Year 2000. Morgan Kaufmann Publishers: San Francisco, CA.

❒Bailey, R. W. (1989). Human Performance Engineering: Using Human Factors/Ergonomics to Achieve Computer System Usability. Englewood Cliffs: Prentice Hall.

❒Barry W. Boehm (1988). A Spiral Model of Software Development and Enhancement. IEEE Computer.

❒Bass, Len, Coutaz, Joëlle (1991). Developing Software for the User Interface. Reading, MA, Addison-Wesley Publishing.

❒Caroll, J. M. (1995). Scenario-based Design: Envisioning Work and Technology in System Development. Wiley: New York, NY.

❒Coutaz, J. (1991). Architectural Design for User Interfaces. Proceedings of ESEC'91.

❒David N. Card and Robert L. Glass (1990). Measuring Software Design Quality. Prentice Hall, Engewood Cliffs, New Jersey.

❒Gould, J. D. (1995). How to Design Usable System. Morgan Kaufmann Publishers.

❒Hix,D., and Hartson, H. R. (1993). Developing User Interfaces: Ensuring Usability through Product and Process. John Wiley and Sons.

❒Isensee, S., and Rudd, J. (1996). The Art of Rapid Prototyping. International Thomson Computer Press, ISBN 1-85032-215-5.

❒John, Bonnie E., Bass, Len (2001). Usability and software architecture. In Behaviour and Information Technology, p. 329-338.

❒Laurel, Brenda (1990). The Art of Human-Computer Interface Design. Addison-Wesley Publishing Co.

❒Nielsen, Jakob (1994). Usability Engineering.Nielsen, Jakob; Mack, Robert (1994). Usability Inspection Methods.

❒Norman, Donald (1991). Cognitive Engineering in User Centered System Design.

❒Pfleeger, Shari (1991). Software Engineering: The Production of Quality Software. Macmillan.

❒Preece, Jenny (1994). Human-computer interaction. Addison-Wesley.

❒Pressman, Roger (1996). Software Engineering: A Practitioner's Approach. McGraw-Hill.

❒Reilly, J. P. (1996). Rapid Prototyping: Moving to Business-centric Development. Thomson Computer Press: Boston, MA.

❒Rudd, Jim, Stern, Ken, Scott I (1996). Low vs.High Fidelity Prototyping Debate. Interactions 96´.

❒Ruoff, C., and Horowitz, E. (1995). A System for Specifying and Rapidly Prototyping User Interfaces.

❒Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, MA.

Capitulo III

Normas de Diseño de Interfaces Visuales

03

Normas de Diseño de Interfaces Visuales

El diseñador de la interfaz del usuario puede hacer uso de importantes recursos visuales y multimediales, tales como imágenes, videos, sonido, fotos. Pero, será indispensable analizar y aplicar normas de diseño específicas para lograr un mejor aprovechamiento del espacio, forma, color y contenido.

Temas a Tratar

I -Introducción
II -Características de las Interfaces Visuales
III - Descripción de las Técnicas de Interacción Visual
IV - Las Técnicas de Interacción Visual y sus Aspectos de Diseño
- Múltiples Ventanas
- Menús
- Manipulación Directa
- Gestos
- Feedback Visual
- Dependencias
- Animaciones
- Exploradores de Objetos
- Cajas de Diálogo

Ejercitación
Caso de estudio
Referencias

I - Introducción

Desde cierto punto de vista, se requiere un entrenamiento mayor y más específico para leer un libro que para contemplar y apreciar una pintura, aunque la forma de construcción de la imagen puede ser tanto o más compleja que la confección de un texto escrito.

La eficiencia comunicacional como su impacto visual y el grado de información transmitido en un instante de tiempo, se debe medir desde la percepción del usuario y no desde la complejidad de las herramientas utilizadas.

En el área de Informática, a mediados de la década de los 80', se produjeron importantes cambios que revolucionaron la forma de representar y expresar el diálogo hombre-computadora, sustituyendo al texto, utilizado hasta ese momento como único recurso de interacción.

Debido a los grandes avances tecnológicos, la aparición de pantallas con mayor definición, dispositivos de interacción gráficos como el ratón -inventado en 1968 por Douglas Engelbart- o el lápiz óptico, comenzaron a desarrollarse para los sistemas de software, un estilo de interfaz más poderoso que el textual.

De esta manera, surgieron las denominadas "Interfaces Gráficas" o "Interfaces Visuales", denotadas comúnmente con las siglas "GUI" (de Graphical User Interface) o "VUI" (de Visual User Interface) respectivamente.

Este tipo de interfaces incrementaron exponencialmente el poder representativo de la información como también los posibles medios de expresión por parte de los usuarios. Esto último mediante múltiples mecanismos de interacción como el movimiento, la voz, el señalamiento de objetos, el arrastre, buscan facilitar las analogías con el mundo real. Ejemplos de esto son los carteles de advertencia o de error como el cartel de pare de la circulación vehicular.

Esto trajo aparejado consecuentemente una acentuada evolución en el campo de los sistemas operativos para el soporte de este tipo de interfaces, como también de los lenguajes de programación que se adecuaron para poder desarrollarlas minimizando los costos y la complejidad que se les atribuyen.

Desde el punto de vista metodológico también hubo cambios importantes, surgió el concepto de Independencia de Diálogo que, como se explicó en el Capítulo 1-Sección 5, estipula una nueva forma de encarar y tratar los sistemas interactivos. Es una estrategia que realiza una separación desde los tiempos iniciales del desarrollo, entre la interfaz del usuario o componente de diálogo y la aplicación o componente de cómputos, dando lugar a nuevas arquitecturas del software.

Esto provocó por parte de los desarrolladores una mayor autonomía para tratar temas específicos de la interfaz, una necesidad de especialización en el tema de factores humanos y de investigación sobre principios de diseño acordes a este tipo de representaciones visuales.

II - Características de las Interfaces Visuales

Es muy difícil expresar una definición concreta de lo que una Interfaz del Usuario Visual significa. Los límites entre una interfaz del usuario visual y la textual, aunque parezcan interfaces muy contrapuestas, pueden ser difusos y llevar a realizar ciertos cuestionamientos como ser: ¿qué es lo que debe ser visual, los datos o el diálogo?, ¿es suficiente con poner dibujitos en los resultados del sistema?, ¿es suficiente con programar en un lenguaje visual, para que la interfaz del usuario sea visual?

Fundamentalmente en el proceso de transición, en el cuál la interfaz del usuario textual deja de ser la interfaz tradicional y se impone la interfaz gráfica como tal, surgieron muchísimas interfaces del usuario híbridas, o sea que en parte pueden considerarse textual y en parte visual. Hoy en día, también puede ocurrir esto, donde ambos tipos de interfaces siguen vigentes y hasta pueden convivir en un mismo producto. Este es el caso de algunos celulares con interfaz textual que no admiten manipulación directa pero que presentan algunos íconos.

Entonces, específicamente ¿qué características debe cumplir una interfaz del usuario para considerarse visual?. Hay autores que son más estrictos y la definen con más restricciones y condicionamientos. Otros, en cambio, son más generales en su definición.

Aquí, se va a abordar una de las concepciones más amplias que se le atribuyen a las Interfaces Visuales, en donde se estipula que "una interfaz del usuario visual es aquella que se encuentra presente en sistemas de computación que soportan algunas de las siguientes características: visualización, interacción visual, o programación visual".

Estas características que se mencionan en la definición, hacen referencia a qué componente o parte de la interfaz debe estar dotada de cualidades visuales, si la componente de presentación debe ser visual –visualización-, o el diálogo que provee la misma debe serlo –interacción visual- o, si el usuario tiene la posibilidad de programar algoritmos, consultas, estrategias de búsquedas, que la interfaz le permita hacerlo en forma visual–programación visual-.

Para aclarar un poco más estos conceptos, se va a explicar cada uno de ellos a continuación.

Visualización:

Se entiende por visualización al uso de representaciones visuales para ilustrar los datos y entidades que intervienen en los programas, como también las entradas y salidas del sistema. Estas representaciones visuales pueden ser gráficos como diagrama de tortas o gráficos estadísticos, imágenes como la imagen de un usuario o avatar en un sistema de mensajería instantánea, ó secuencias de animación como por ejemplo una presentación o demostración automática de una funcionalidad de un sistema donde se ilustra una secuencia de acciones.

En la visualización, son los objetos los que tienen asociado una representación visual, pero no es obligatorio que la interacción hombre-máquina o el modo de dialogar con el sistema también sean visuales.

El dominio de aplicación de estos sistemas puede verse incluido dentro de procesamiento de imágenes, robótica, manejo de base de datos de imágenes, entre otros.

Por ejemplo, en la figura siguiente se muestra la pantalla de un Sistema de Escrutinio de Votos, donde la interacción con el mismo es secuencial, basada en menús funcionales, pero la representación de los resultados electorales se realiza visualmente mediante diagramas de tortas.

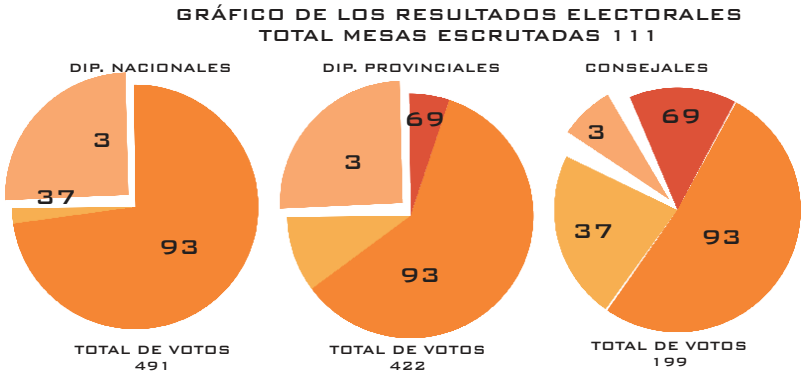


Figura 3.1: Representación visual sobre los resultados de un escrutinio de votos.

Interacción Visual:

Una interfaz del usuario no sólo se compone de la representación de los datos de entrada, resultados y de los estados del sistema.

En una interfaz del usuario se encuentra también la comunicación, la conversación entre el usuario y el sistema, cuestiones de comportamiento que deben ser expresados y representados de alguna manera.

Una interfaz provee interacción visual cuando el usuario para comunicarse con el sistema y llevar a cabo sus intenciones, utiliza diálogo asincrónico, basado en manipulación directa y en eventos. El usuario puede expresarse seleccionando, señalando, arrastrando, moviendo objetos presentes en la pantalla. Un ejemplo sería un aplicativo de geometría donde el usuario puede mover, arrastrar, dimensionar figuras geométricas.

Por otra parte, la interfaz también debe utilizar mecanismos visuales para expresarse ante el usuario, esto es en el caso de dar indicaciones, aclaraciones, mensajes de error u otro tipo de diálogo que vaya dirigido desde la máquina al usuario. Entonces, una interfaz visual debe utilizar gráficos, colores, movimientos, animaciones, sonido para transmitirle información al usuario del sistema.

El dominio de aplicación donde se encuentra interacción visual está constituido por las interfaces basadas en manipulación directa, interfaces gestuales, interfaces icónicas, video juegos, entre otros.

Por ejemplo, en la figura siguiente se muestra la pantalla de un Sistema para la Confección de Carteles, en donde en el momento de la impresión, se muestra una animación sobre la progresión de este proceso:

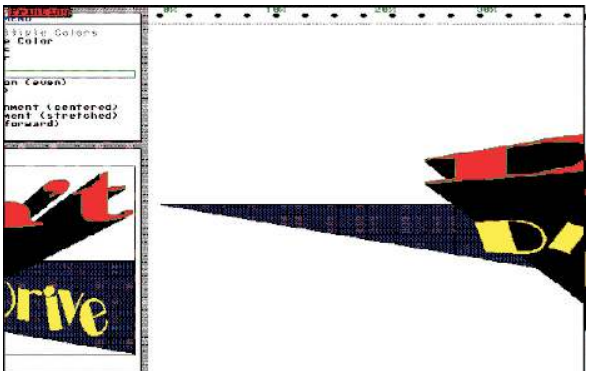


Figura 3.2: Interacción visual en el proceso de impresión de un Sistema de Confección de Carteles

Programación Visual:

Se entiende por programación visual al uso de expresiones visuales en donde intervienen íconos ó gestos durante el proceso de la programación.

Aquí, tanto el lenguaje como sus constructores de programación deben tener una representación visual en vez de textual. Las reglas y sus sentencias deben ser representados gráficamente para especificar los algoritmos en un lenguaje visual.

Los datos, parámetros u objetos que puedan intervenir en los algoritmos no necesariamente deben ser visuales, aunque por lo general para mantener coherencia con el lenguaje provisto, son representados gráficamente, siempre y cuando esto sea factible.

El dominio de aplicación para estos lenguajes de programación visual incluye diseño visual de interfaces del usuario, programación visual de consultas a bases de datos, manejo de formularios y diseño asistido por computación.

Como ejemplo, se puede ilustrar al lenguaje de programación procedural IconAuthor, que presenta tanto sus constructores como sus sentencias en forma visual mediante el uso de íconos representativos. El programador, para confeccionar sus algoritmos, tiene que realizar una serie de selecciones, arrastres de íconos, elegir algunas opciones, completar algunos datos en forma visual, desentendiéndose de cuestiones sintácticas.

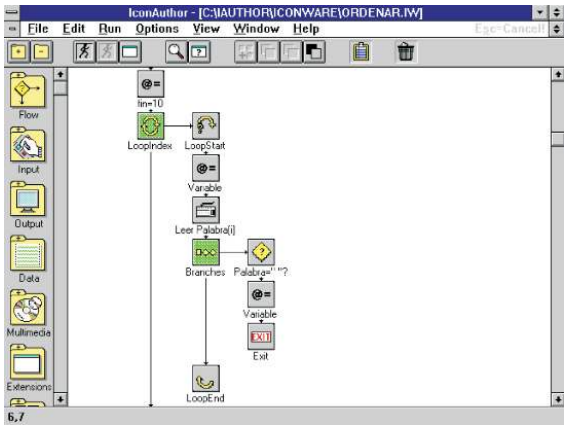


Figura 3.3: Programación visual en el lenguaje IconAuthor.

Importante

■Sobre el concepto de Interfaz del Usuario Visual hay algunos autores que indican la necesidad de cumplir con todas las características de visualización, interacción y programación visual para definirla. Otros, en cambio, estipulan que con solo proveer alguna de ellas ya es condición suficiente.

■Lo esencial es que se trate de diseñar sistemas que minimicen los costos de interpretación y ejecución por parte de los usuarios.

III - Descripción de las Técnicas de Interacción Visual

Una vez que el concepto general de interfaz visual sea entendido claramente, se profundizará en las técnicas de interacción visual que conforman o pueden conformar una interfaz del usuario.

Se entiende por técnica de interacción visual, al conjunto de formas que pueden ser empleadas o combinadas por el diseñador de la interfaz del usuario para especificar un aspecto, un comportamiento visual o un estilo de interacción que adoptará la interfaz frente a los usuarios del sistema interactivo.

Entonces, las técnicas de interacción visual son los recursos o los mecanismos que el diseñador emplea para confeccionar una interfaz del usuario visual, afectando fundamentalmente el proceso y representación del diálogo más que a nivel de apariencia de la interfaz.

Entre las técnicas visuales más comúnmente utilizadas, se pueden mencionar a:

A- Múltiples ventanas:a:

Esta técnica ha sido una de las técnicas visuales más populares de los últimos años.

La revolución realmente ocurrió cuando el concepto de manipular ventanas resultó apropiado en el mundo de las estaciones de trabajo.

Es una técnica que permitió implementar la Metáfora de escritorio, creada por Xerox Park en 1970, en donde se trató de construir una ilusión al usuario, de visualizar al sistema como una oficina, con un montón de papeles o actividades en desarrollo, con la simulación de los objetos y servicios propios de las mismas.

A nivel de factores humanos, esta metáfora permite que el usuario perciba un ambiente que le resulta familiar, mediante la visualización de elementos, el acceso a funciones fácilmente y la localización de contextos lógicos, pudiendo apreciar a la computadora como una herramienta de fácil utilización y entendimiento. Un ambiente en donde por ejemplo, el usuario asocia el concepto de guardar un objeto en un fichero naturalmente.

Ahora, ¿qué se entiende realmente por ventana?. Específicamente, una ventana es definida como “un área rectangular, que contiene una aplicación de software o un archivo de información. Permite la definición de un contexto independiente, donde se agrupan subtareas o datos, del mismo nivel de abstracción”.

Por su definición, se puede decir que con la utilización de varias ventanas en una interfaz del usuario, se permite multiplexar y compartir el uso de la pantalla, que muchas veces la misma resulta ser muy limitada cuando se debe expresar gran cantidad de información.

Además, ofrece un poder significativo para la organización de la información y el control de presentación, con sus títulos, bordes y colores de fondo que denotan agrupamientos particulares.

Entre sus principales ventajas se encuentran la optimización del uso del estado real de la pantalla, la facilidad de movimiento de un contexto a otro para trabajar simultáneamente en diferentes tareas como por ejemplo tener una planilla de cálculo separada del correo electrónico y, la posibilidad que el programa de aplicación pueda entablar distintos hilos de diálogo o de ejecución con los usuarios, como por ejemplo permitir el llenado de más de un formulario a la vez.

A continuación, se muestra una pantalla con múltiples ventanas sobre un sistema de Taller Mecánico desarrollado por alumnos de la cátedra, en donde se tiene información detallada de cada empleado que atendió la reparación del automóvil seleccionado.

Figura 3.4: Ejemplo múltiples ventanas

B- Menús:

Una interfaz orientada a menús, notoriamente reemplazó en la década del 80´al lenguaje de comandos, puesto que le ofrecía al usuario la posibilidad de elegir entre una serie de opciones, la función que necesitaba ejecutar, más que escribir el comando textualmente.

Desde el punto de vista de los factores humanos, los menús disminuyen el esfuerzo mental y el entrenamiento exigido al usuario de definir secuencias complejas de comandos, le evitan el empleo de una sintaxis restringida, y de memorizar cuestiones técnicas y del lenguaje.

Entonces, se puede decir que “un menú es una lista limitada de opciones que pueden ser seleccionadas por el usuario y que están disponibles dentro de un marco visual común”.

Existen diferentes estilos de menús, se encuentran los menús horizontales -comúnmente denominados menu bar-, que tienen sus opciones organizadas en una línea plana. Los menús verticales que por el contrario, presentan sus opciones, una encima de la otra. Los menús pull-down combinan los dos anteriores, puesto que se componen de un menú horizontal del cuál se despliega para cada una de sus opciones en un menú vertical.

Los menús jerárquicos son aquellos donde el conjunto de opciones están distribuidas en varios niveles de menús anidados, formando una estructura en cascada o en multinivel. El usuario dentro de un determinado menú tiene dispuestas opciones destinadas a la apertura de otros.

También, se encuentran los menús pop-up que se caracterizan por no están presentes en la pantalla, son invocados cuando se necesitan, mediante una combinación de teclas o mediante el botón derecho del mouse. Por el contrario, se puede citar los menús de pantalla completa o full screen, que disponen su conjunto de opciones de una forma que ocupa casi la totalidad de la pantalla.

Además, los menús presentan una serie de características como ser su capacidad de deslizar sus opciones -menús deslizables o con scrolling-, de permitir seleccionar más de una opción –menús de selección múltiple,

de presentar dos versiones, una con un contenido mínimo de opciones más utilizadas y otra en forma completa, de permitir el uso de atajos o combinación de teclas para seleccionar alguno de sus ítems.

A los menús se les puede atribuir funciones sintácticas importantes como ser la configuración de su aspecto, los colores, el fondo, el tipo y tamaño de letras utilizados, como también ofrecer diferentes criterios de ordenación de las opciones, reorganización general de los menús, el agregado o quite de opciones dentro de un menú, entre otros.

Entre las ventajas principales de esta técnica visual, se encuentran la facilidad de acceso a las funciones, simplicidad de manejo, reducción del espacio de la pantalla destinado a la muestra de funciones, pero se les atribuye por el contrario, que tienden a ser ineficientes, que la interacción con muchos menús anidados resulta ser engorrosa, fundamentalmente para los usuarios experimentados del sistema.

La pantalla siguiente muestra como ejemplo un sitio educativo sobre el Desarrollo de Huertas, desarrollado por alumnos de la cátedra, en donde se visualiza un menú pull-down.

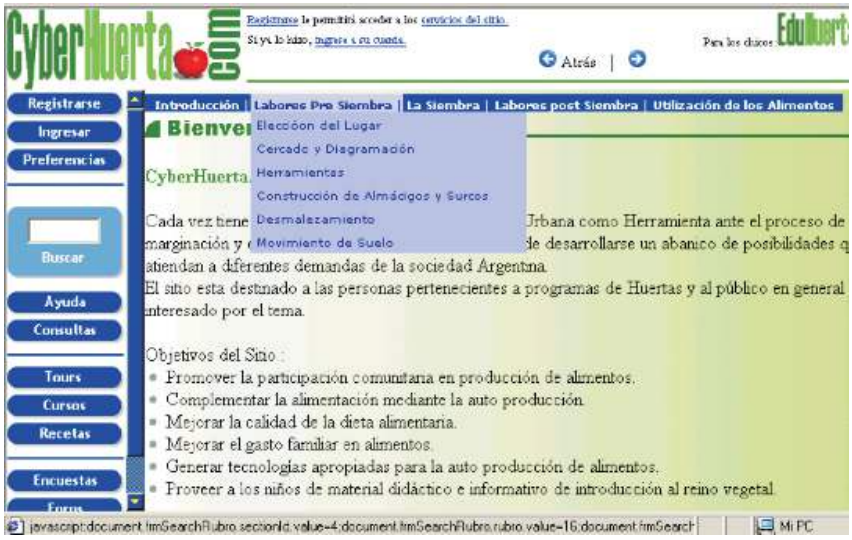


Figura 3.5: Ejemplo de Menús.

C- Manipulación directa:

En el Capítulo 1-Sección 3, se explicaron los diferentes tipos de diálogo humano-computadora y se presentaron dos modelos de interacción: la metáfora de Modelo del Mundo y la Conversacional.

Entonces, "la manipulación directa es una técnica de interacción que responde a la metáfora de Modelo del mundo, en donde el usuario tiene a disposición objetos que pueden ser accionados de múltiples maneras, como ser mediante el arrastre, selección, múltiple selección, cliqueo, entre otros".

A diferencia del mundo conversacional, donde se tiene una descripción lingüística de los objetos y de las acciones a realizar sobre los mismos, con la manipulación directa, el usuario directamente demuestra sus intenciones en vez de escribirlas.

En el Modelo del Mundo, electrónicamente se hace mímica de los objetos del mundo real y de sus formas de uso.

Desde el punto de vista de los factores humanos, la Manipulación Directa responde a pensamientos psicológicos tales como los expresados por los filósofos Jean Piaget [Piaget, 1947] y Jerome Bruner [Bruner, 1984] que indican que nuestro entendimiento del mundo está fundamentalmente relacionado con una estimulación visual y experiencia táctica sobre los objetos, donde uno combina habilidades "activas" -manipulación de objetos-, "icónicas" -reconocimiento visual para tender a la comparación y contraste- y, "simbólicas" -entender grandes secuencias de razonamiento visual.

También, varios investigadores han analizado los factores cognitivos subyacentes de este estilo de interfaz y encontraron dos aspectos positivos muy importantes desde la percepción del usuario: la noción de distancia y la de unión.

Factores relacionados con Manipulación Directa	Explicación
Noción de Unión	■La noción de unión es el sentido de manipular e interactuar con los objetos directamente sobre la pantalla, más que realizar una conversación sobre ellos. ■Permite a los usuarios sentir la sensación de estar involucrado directamente con el mundo de los objetos, sin intermediarios.

Noción de Distancia	■La noción de distancia se refiere al esfuerzo mental del usuario necesario para trasladar sus pensamientos e intenciones a los requerimientos físicos del sistema que está usando. ■En una interfaz del usuario, esta distancia está reflejada por la relación entre la tarea que el usuario tiene en mente y la forma en que esta tarea debe ser realizada vía la interfaz. ■Un uso efectivo de manipulación directa significa elegir una apropiada representación gráfica para los objetos y acciones del problema, así la interfaz le brinda la posibilidad al usuario de reducir su esfuerzo cognitivo para desarrollar sus tareas.

Tabla 3.1: Factores relacionados con la Manipulación Directa

En la siguiente pantalla, se muestra un sistema sobre una Sala de Pediatría desarrolllado por alumnos de la cátedra, en donde el usuario puede arrastrar la imagen del bebé a alguna de las camas disponibles de la sala, mediante la técnica de manipulación directa.

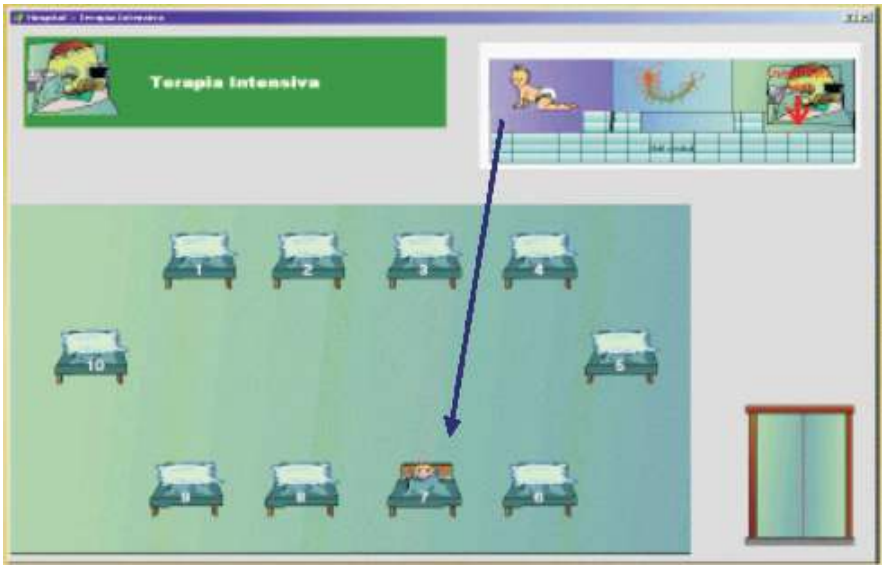


Figura 3.6 Ejemplo de Manipulación Directa.

D- Feedback visual:

No todo el diálogo se establece del usuario hacia la máquina. Frente a las acciones del usuario, la interfaz debe reaccionar actuando, explicando, respondiendo, asistiendo o cualquier otro tipo de manifestación, que va dirigida hacia el usuario. Se requiere de una técnica que soporte planeamiento de tareas, detección de errores y recuperación de errores, por lo tanto se requiere que la interfaz tenga indicadores explícitos del estado del sistema [D.Norman, 1986].

Entonces, se denomina al feedback visual, a una reacción de la interfaz frente a un estímulo. Es el proceso de reflejar sobre la pantalla el resultado de alguna operación realizada por el usuario, realizándose esto de una manera gráfica o mediante algún recurso visual.

En el contexto de HCI, el feedback es una expresión de salida producida por el sistema luego de procesar la entrada de un usuario. El usuario luego interpreta el feedback, evalúa la situación y actúa en base a ella.

El feedback es esencial para que el usuario no se sienta a merced de la aplicación. Cuando una acción conlleva mucho proceso interno es necesario que el usuario esté informado de ello, así puede disponer de su tiempo y anticipar claramente sus acciones.

Existen dos tipos de feedback: el feedback superficial y el semántico.

Tipos de Feedback	Explicación
Feedback Superficial	■Un feedback superficial es aquella respuesta del sistema que suministra información propia de la entrada y salida, y de los elementos que conforman las pantallas. ■La misma se constituye a través de información que se maneja a nivel de interfaz del usuario. ■Por ejemplo, indicar que un casillero en blanco debe ser completado.

Feedback Semántico	■Un feedback semántico es aquella respuesta en donde se debe extraer información de la aplicación para lograr una respuesta más apropiada, significativa, más completa y relacionada al dominio de la aplicación. ■Por ejemplo, mostrar información sobre el estado y progresión de una determinada transacción.

Tabla 3.2: Tipos de Feedback.

Un feedback informativo e inmediato permite hacer del sistema una sucesión de estados explícitos, permite responder mediante confirmaciones, validaciones, detecciones de errores, rechazos, demoras, aclaraciones.

Hay que tener en cuenta que los indicadores de error o mensajes de error son un tipo específico de feedback.

A continuación, se muestra un ejemplo de feedback informativo, que los alumnos aplicaron sobre un sitio de Turismo.

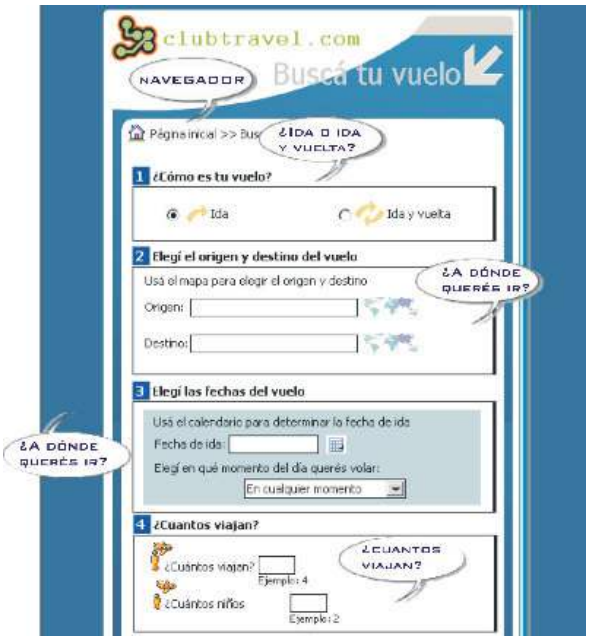


Figura 3.7: Ejemplo de Feedback.

E- Gestos:

En el entorno de HCI, gestos es una marca, estela o camino que traza el usuario sobre la pantalla, para expresar alguna información o representar alguna acción que quiera realizar.

Una interfaz que admite gestos como forma de interacción por parte del usuario, implica que el mismo puede realizar una manipulación cuyo trazo o camino que genere en ese movimiento, es analizado e interpretado por la interfaz, punto por punto.

Entonces, se puede decir que una sesión de entrada gestual consiste de una secuencia o un camino de puntos logrado por determinadas acciones que el usuario desempeña mediante dispositivos físicos como el teclado o el mouse. Es similar a una sentencia de comandos, ya que se compone de operaciones y parámetros.

El sistema luego, trata de realizar el análisis sintáctico de los gestos y de los objetos que intervienen en la sesión de entrada y los transforma en llamados a procedimientos semánticos.

En los siguientes ejemplos, se muestra el uso de gestos para especificar las operaciones de borrado (A) e inserción de textos (B), que el sistema debe interpretar.

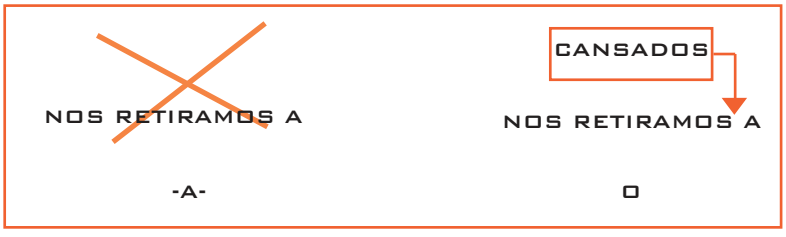


Figura 3.8: Ejemplos de Interacción Gestual.

La interfaz que admite este poderoso recurso, está basada en una tecnología de reconocimiento de imágenes, en donde el sistema es capaz de entender la configuración significativa de trazos, puntos, que constituyen la información del camino.

Se puede observar que hoy en día las interfaces del usuario adoptaron la interacción gestual, ya que es mas natural donde los humanos están acostumbrados a interactuar mediante gestos y usar herramientas "tocandolas". La tecnología ha avanzado lo suficiente como para que el poder de cálculo disponible alcance para que un ordenador pueda interpretar los gestos. Existen cascos, anteojos, muñequeras, u otros dispositivos o comandos para hacer efectivos los gestos, hasta inclusive las cámaras web son muy útiles para detectar e interpretar movimientos de la cara, manos o cualquier otra parte del cuerpo.

F- Dependencias:

Muchas veces, un mismo concepto semántico es representado sobre el espacio de la pantalla de varias maneras, para enfatizar su grado de importancia.

Por ejemplo, puede suceder que la variable temperatura de una caldera sea visualizada mediante diferentes mecanismos, a través de un casillero textual, de la imagen de un termómetro y de un rectángulo que se va coloreando de acuerdo al calor de la misma.

En estos casos, hay que mantener siempre una coherencia entre los valores que expresan estas diferentes representaciones, puesto que hay una dependencia significativa entre ellas.

Su definición formal es la siguiente, "una dependencia especifica una relación causal que debe ser siempre mantenida". Sucede cuando al manipular un objeto de la pantalla puede significar consecuencias en la aplicación que, a su vez, causen efectos directos o indirectos sobre otros objetos de la misma pantalla.

Esta herramienta es muy poderosa y permite que el sistema, ante la manipulación de un objeto determinado, ajuste todas las partes, como así también, los objetos que dependen de él.

Por ejemplo, en la siguiente pantalla, se muestra el sistema de Ruleta realizado por alumnos de la cátedra, en donde el jugador "Seba" puede observar su estado de situación o balance, a través de diferentes objetos dependientes, como ser el cuadro de "Balance Gral", la carita amarilla triste, el cuadro "Seba" con su saldo negativo.

Figura 3.9: Ejemplo de Dependencias

G- Animación:

La animación es una técnica de interacción que visualiza el proceso interno de un sistema dinámico. Es una parte integral en la interfaz visual donde los cambios en el sistema están provocados por acciones autónomas del propio objeto animado que complementa las operaciones del usuario.

Mediante la animación, la interfaz es capaz de simular el comportamiento complejo de problemas dinámicos. Presenta como ventaja, la posibilidad de representar una gran cantidad de información en un período reducido de tiempo y de garantizar la comprensión de determinadas características que no son obvias con los gráficos únicamente. La capacidad expresiva de una animación es muy significativa y puede ser muy útil para el feedback de la interfaz en el caso de representar visualmente el estado y el progreso de un proceso en curso, como por ejemplo, animar el proceso de desfragmentación de archivos del disco.

A continuación, se muestra una pantalla con una animación para representar el estado de una impresión de un cartel.

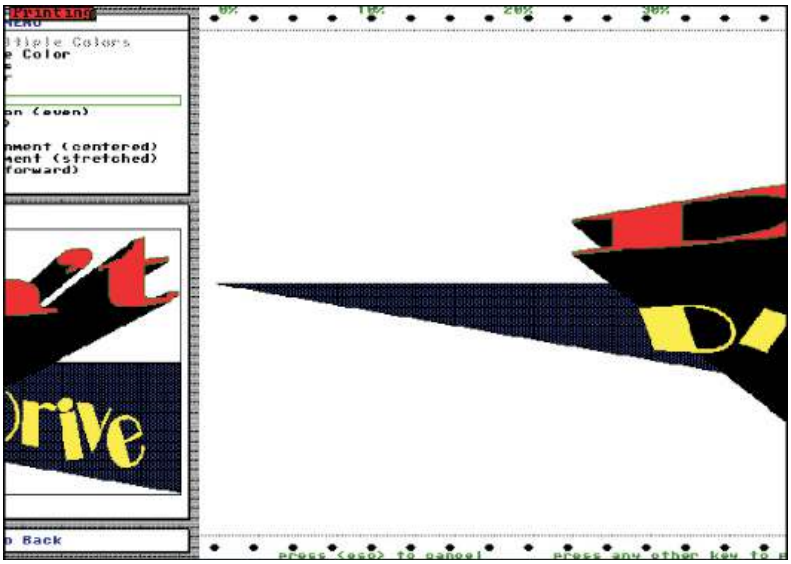


Figura 3.10: Ejemplo de Animación.

H- Explorador de Objetos:

La organización de la información en la pantalla es un problema muy complicado y más aún cuando se trata de datos masivos. La técnica visual de explorador de objetos o grillas interactivas permite mostrar un gran caudal de datos en una manera ordenada y categorizada, donde el usuario puede manejar y consultar la información en forma agrupada dentro de un contexto particular.

La misma se define como “una técnica de interacción que permite visualizar una lista de objetos compuesto por varios atributos”. Se visualiza como una tabla donde cada fila es la especificación de cada objeto y en las columnas se muestran los valores de sus atributos y características. Además, se caracterizan por su condición de interactividad, o sea el usuario puede manipular, seleccionar, editar, desplazar filas, entre otras cuestiones.

Además el explorador puede proveer al usuario funciones sintácticas que optimicen su manejo, como ser la ordenación de sus filas mediante diferentes criterios, el filtrado de filas, el agregado y quite de columnas, la configuración visual del mismo, entre otras.

A continuación, se muestran ejemplos de grillas pertenecientes a un sistema de Cursos a Distancia desarrollado por alumnos de la cátedra. El primero de ellos, es utilizado para la carga de participantes a un curso, por lo que es editable. Mientras que la segunda grilla, muestra resultados de búsquedas realizadas, que en este caso, es hallar información sobre los cursos tomados por un alumno, cuyo apellido es “García”. Este explorador presenta como características que no es editable pero es de selección múltiple.

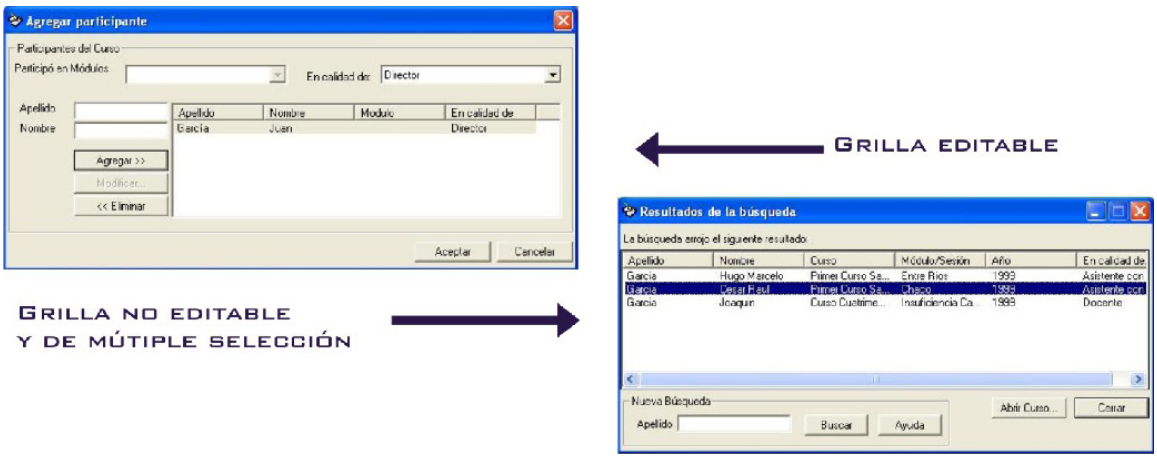


Figura 3.11: Ejemplo de Grillas

I- Cajas de Diálogo:

La apariencia de una caja de diálogos es muy parecida a la de un formulario de oficina, donde hay casilleros a completar, ítems a seleccionar, entre otros elementos.

En el área de HCI, una caja de diálogos constituye un contexto rectangular, donde el usuario puede especificar opciones y acciones usando botones y controles tales como listas, casilleros de entrada, listas deslizables, botones para tildar, entre otros.

El usuario, a través de una caja de diálogos tiene agrupados ciertos controles y mecanismos visuales que le permiten dialogar con el sistema en forma bidireccional y mediante manipulación directa.

Tiene presentes campos de entrada para ingresar texto, puede seleccionar, arrastrar, puede ver mensajes a través de una región de salida y puede contar con una asistencia específica por cada control y objeto de interacción presente.

Habitualmente, los distintos campos son evaluados independientemente pero la confirmación o cancelación se realiza a nivel de caja de diálogo en general.

En la figura siguiente, se presenta una caja de diálogo en donde el usuario debe interactuar con diferentes objetos de interacción.

Figura 3.12: Ejemplo de una Caja de diálogo.

Una vez analizadas las técnicas de interacción visual más importantes, se puede concluir que la combinación y utilización de estos recursos visuales constituyen la esencia de una interfaz visual.

Las bases conceptuales de estos paradigmas como el concepto general de las interfaces gráficas, fueron desarrolladas por Xerox Parc que, desde fines de la década del 70', pregonaba sobre ideas tales como múltiples ventanas, manipulación directa, uso de íconos como carpetas, documentos y el uso del mouse.

De esta manera, surgieron estilos de interfaces gráficas estándares como Interfaces WYSIWYG que son las iniciales en inglés de What You See Is What You Get, o sea todo lo que ves es lo que puedes tomar, e Interfaces WIMP, iniciales de Windows, Icons, Menus And Pointing Device, o sea ventanas, íconos, menús y dispositivo de señalamiento.

La primera hace referencia explícita a la técnica de manipulación directa, donde todo lo que el usuario ve, lo puede "tocar" o manipular, mientras que en las interfaces WIMP hace referencia a las técnicas de interacción más importantes como ventanas, menús y manipulación directa.

Una cuestión que hay que tener en cuenta respecto a las técnicas de interacción visual, es que la mayoría de estos recursos vienen ya desarrollados e implementados, encapsulados mediante componentes en librerías de objetos de interacción, incluidos en los lenguajes de programación visuales o incorporados dentro de los sistemas operativos, como es el caso de los sistemas de ventanas.

En estos casos, ya viene estipulados sus atributos particulares, su funcionamiento y un comportamiento predefinido, sólo resta que el desarrollador de la interfaz los incluya en su diseño, los integre y los aplique adecuadamente en pos de mejorar y simplificar la interacción con el usuario.

En las próximas secciones se van a analizar recomendaciones de diseño importantes referente a cada técnica de interacción visual, de modo que permita la gestación de interfaces visuales de calidad.

Importante

■Tener en cuenta que el mal empleo de las técnicas de interacción puede hacer que decaiga considerablemente el poder representativo y expresivo de las mismas, por lo tanto las normas que acompañanan cada técnica son esenciales y deben ser aplicadas adecuadamente.

IV- Técnicas de interacción Visual: Aspectos de Diseño

Las normas de diseño para Interfaces Visuales, son específicas a cada paradigma de interacción visual utilizado en la misma. Fundamentalmente, se basan en analizar cada uno de los recursos visuales explicados y en definir ciertas pautas de diseño para aprovechar al máximo, los beneficios a nivel de factores humanos atribuidos a cada uno.

Por ello, se van a listar a continuación, los aspectos de diseño para cada paradigma de interacción visual.

Múltiples Ventanas

Esta técnica visual como se explicó en la sección anterior, ofrece dos ventajas muy importantes, permite optimizar el uso real de la pantalla y segundo, permite que el usuario pueda llevar a cabo múltiples diálogos con el sistema teniendo en cuenta contextos particulares fácilmente reconocibles.

Respecto al diseño de las mismas, vale aclarar que mucho de sus aspectos y comportamientos van a estar delineados y definidos por el sistema operativo subyacente al software que se esté desarrollando.

En la mayoría de los casos, es el sistema operativo quien se encarga del sistema de ventanas gestionando cuestiones de control, apariencia y funcionamiento de las mismas.

De todas maneras, muchos elementos de las ventanas pueden ser configurados y adaptados en manos del desarrollador de la interfaz, por lo que es muy útil analizar una serie de consideraciones de diseño para que se pueda aprovechar al máximo este importante recurso visual. A continuación se mostrará la tabla de recomendaciones respectivas a este punto:

Recomendaciones sobre múltiples ventanas	Explicación
Analizar las funcionalidades sintácticas	■Hay funciones sintácticas de las ventanas pre-establecidas, tales como maximizar, minimizar, redimensionar, mover y cerrar. ■Hay que analizar si estas funciones son adecuadas para el contexto o parte de la interfaz que se esté diseñando, sino habría que deshabilitarlas o quitarlas directamente. ■En el caso que las mismas estén disponibles, hay que garantizar su utilización correcta, analizando cómo queda el contenido de las ventanas cuando el usuario las manipula con estas funciones.
Considerar el modo de apertura de la ventana	■Todas las ventanas del sistema interactivo en cuestión, deben ser activadas de igual forma. Vía combinación de teclas, mediante un menú funcional, una barra de botones. ■Se debe proveer un modo de acceso homogéneo a las mismas. ■Hay que analizar si la ventana se abre en modo exclusivo o no, dependiendo de lo que es beneficioso para el usuario y para el tipo de diálogo que se quiera establecer. ■En el caso del modo exclusivo, todos los eventos serán atendidos sólo por una ventana, desactivando todas las demás, hasta que el usuario culmine el diálogo con la misma. ❖Por ejemplo, si se trata de una ventana de ayuda o asistencia, el usuario necesitará tenerla abierta en forma permanente mientras está sesionando con otras ventanas, por lo que el modo de apertura exclusivo no sería adecuado para este caso.
Definir el ámbito de la ventana	■En el caso que la ventana se pueda mover, hay que establecer sobre qué área de la pantalla se permite dicha movilidad. ■Puede ser que la ventana pueda moverse únicamente dentro del espacio de la ventana en donde se produjo el llamado, o dentro del radio del ambiente del sistema, o en caso contrario, o tener un grado de independencia y estar en cualquier parte de la pantalla.
Analizar el cierre de las ventanas	■Hay que pensar el significado que se le va a dar a la función de cerrar una ventana. ■A veces puede suceder que cerrar una de ellas implique cerrar otras ventanas que se abrieron anteriormente. Hay que analizar en qué casos esto es conveniente. ■Por ejemplo, puede ser que la ventana que se cierra sea la última de una cadena de ventanas que constituyen una transacción por lo que es lógico pensar en que se cierren todas las ventanas intermedias también.

Diseñar el contenido de las ventanas	■El usuario en su proceso de aperturas y cierres de ventanas, se encuentra inmerso en una estructura de navegación, que debe estar lógica y concientemente diseñada. ■La navegación entre ventanas debe estar sustentada mediante una estructura balanceada y equilibrada. El anidamiento excesivo de ventanas en alguna parte del sistema, dificulta la interacción y el acceso a dicho sector, como también induce a la pérdida de contexto. ■El usuario debe percibir el sentido en que fueron organizadas las ventanas, debe poder entender el mapa de navegación subyacente y bajo qué criterios se estructuró la navegación de ese modo.
Analizar la estructura de navegación de las ventanas	■Todas las ventanas del sistema deben estar cuidadosamente diseñadas, estableciendo claramente las zonas de títulos, de funciones, de mensajes, área de contenido, el marco. ■Puede existir categorización de ventanas con sus diseños particulares, como ser ventanas de formularios, ventanas de mensajes, ventanas de resultados, ventanas de funciones.

Tabla 3.3: Recomendaciones de diseño para Múltiples Ventanas.

Importante

■El diseñador de la interfaz del usuario, no debe verse influenciado por cuestiones preestablecidas sobre el sistema de ventanas, impuestas por el sistema operativo que utilice ni por el lenguaje de programación. Se debe tener claro conocimiento sobre los costos y el grado de factibilidad de poder llevar a cabo su diseño propuesto.

Menús

Los menús presentan la posibilidad al usuario de poder elegir entre una serie de opciones en vez de escribirlas textualmente, evitándole definitivamente cometer innumerables errores sintácticos y de tipeo.

Pero, a pesar que se les atribuye la cualidad de ser simples de usar, en el caso de la combinación de un menú bar con menús jerárquicos comúnmente utilizados en los sistemas interactivos actuales, pueden complicar el acceso a las opciones si no fueron pensados a través de una lógica de diseño apropiada.

Por lo tanto, existen para los menús, una serie de recomendaciones a tener en cuenta en el momento de diseñarlos e incluirlos en la interfaz del software.

Recomendaciones	Explicación
Analizar el contenido del menú	■Un menú debe estar compuesto de opciones claras, significativas. ■Las mismas no deben llevar a interpretaciones erróneas ni que genere ambigüedad ante el usuario. ■Al usuario no le debe insumir costo alguno, el entendimiento de las opciones desplegadas en un menú. ■Pueden estar acompañadas por hints, o mensajes explicativos.
Analizar el orden de aparición de las opciones	■El usuario, al interactuar con el menú, debe entender la forma en que el diseñador organizó la lista de las opciones, debe encontrarle un motivo o criterio de ordenación de las mismas. ■Las opciones pueden desplegarse ordenadas alfabéticamente.
Analizar el acceso a las opciones	■El usuario siempre debería poder acceder a las opciones de un menú aunque no cuente en ese momento con el mouse. ■Se deben proveer mecanismos vía el teclado para la activación del menú como de alguna de sus opciones.
Analizar el nivel de abstracción	■Un menú debe contener una serie de opciones que tengan el mismo nivel de abstracción, el mismo nivel de detalle. ■No pueden existir opciones más generales que otras, tampoco debe existir intersección entre opciones ni que la definición de una de ellas incluya a otra. ■Por ejemplo, que en un menú estén las opciones Consultar y debajo Consultar Saldos.

Determinar una organización adecuada de las opciones	■En el caso de un menú bar, donde de cada opción se despliegue un menú jerárquico, hay que analizar la forma en que fueron organizados. ■Se debe establecer una organización que simplifique el acceso a las tareas, que sea entendible y natural para el usuario. ■Debe existir un sentido en el agrupamiento de las opciones en distintos menús, debe haber una estructuración semántica, significativa. ■Puede ser que las opciones horizontales del menú bar, sean objetos mientras que las verticales sean funciones asociadas a los mismos. O al revés, que horizontalmente se desplieguen los procesos y verticalmente los objetos a los que se los aplican. ■Los menús jerárquicos deben proveer una misma cantidad de niveles de profundidad. No deben estar desbalanceados o desnivelados ni tampoco proveer anidamientos engorrosos. ■Estudios revelan que no es conveniente superar más de siete opciones por menú y proveer más de tres niveles de anidamiento. ■Además, hay que definir un criterio para los diferentes niveles de menús, que haya una organización por nivel de detalle, por nivel de importancia o por otro tipo de lógica.
Analizar funciones sintácticas asociadas	■A los usuarios se les puede proveer funciones para configurar los menús del sistema, modificando su aspecto, tamaño, tipo y colores de letras de las opciones, como su feedback asociado para la opción seleccionada, la opción que denota un menú, la opción por donde el mouse está pasando, entre otros. ■También, puede proveer funciones de adaptación donde el usuario pueda agregar o quitar opciones, adelantarlas y cambiarlas de lugar, modificar la jerarquía, cambiar el orden de aparición de las mismas, entre otras cuestiones.

Tabla 3.4: Recomendaciones de diseño para Menús.

Manipulación Directa

La manipulación directa determina un lenguaje de interacción muy particular y poderoso, que tiene atribuido ventajas muy importantes desde el punto de factores humanos.

Debido a la representación gráfica de los objetos y a su disponibilidad en el ámbito de la pantalla, brinda una estimulación visual que motiva a que el usuario los accione o los manipule en pos de expresar sus intenciones.

El usuario en una interfaz con manipulación directa, demuestra lo que pretende hacer con el sistema mediante acciones representativas, en vez de escribir expresiones o comandos relativos a ello. Pero, de todas maneras, si la manipulación directa está mal aplicada, por ejemplo el usuario no entiende qué significan esos objetos gráficos, ni sabe qué pueden hacer con ellos, muchas de sus ventajas se vuelven en contra.

Entonces, hay cuestiones de diseño referidas a la manipulación directa que deben ponerse en práctica, para garantizar sus múltiples beneficios.

Recomendaciones	Explicación
Analizar la representación gráfica	■Hay que estudiar si la representación gráfica de los objetos permite el reconocimiento de los mismos. ■El usuario debe entender qué información representan los objetos presentes en la pantalla.
Analizar los costos de la manipulación	■Se refiere a si la interacción mediante la manipulación directa disminuye claramente el esfuerzo cognitivo y manual del usuario para llevar a cabo sus intenciones. ■Por ejemplo, si requiere mucha puntería, se necesita tener pulso o se debe arrastrar sobre lugares muy pequeños, la manipulación se vuelve costosa de realizar, demasiado rígida y puede terminar en desuso.
Determinar si la manipulación es explícita	■Hay que comprobar que la manipulación que se le permita al usuario sea obvia y esté explícitamente representada. ■Esto significa que hay que denotar con colores, formas u otros recursos visuales, los elementos que se pueden mover, los lugares admisibles de la manipulación, los movimientos permitidos, entre otras cuestiones. ■Puede que la misma esté acompañada por algún mensaje que incite a la realización de movimientos, selecciones y arrastres.

Proveer distintos medios para la manipulación	■Analizar y brindar diferentes mecanismos para realizar la manipulación. ■Que la manipulación sea posible tanto mediante el ratón como mediante el teclado, u otro dispositivo de interacción.
Combinar manipulación directa con feedback	■Se debe reflejar el estado previo y posterior a una manipulación, como también los movimientos que son admisibles, los objetos que se pueden manipular y las zonas permitidas para llevar a cabo la manipulación. ■Se debe representar visualmente las manipulaciones no válidas, por ejemplo trasladar un objeto sobre otro no permitido, que éste se inhiba o visualice una señal de rechazo.
Que sea obvia	■La interfaz debe expresar de alguna manera que la manipulación existe y debe incentivar al usuario a realizarlas. ■La manipulación directa debe estar acompañada de algún mensaje que incite a la realización de dicho movimiento.
Alcance de la manipulación	■El usuario debe tener claro mediante la interfaz, los límites de la manipulación. Si realmente, puede manipular objetos entre distintas ventanas, entre distintas áreas de la misma o no.

Tabla 3.5: Recomendaciones de diseño para Manipulación Directa.

Feedback Visual

Al feedback se lo puede describir como las respuestas que da el sistema frente a actividades que realiza el usuario. Es la expresión del estado resultante del sistema luego de acciones efectuadas por el usuario.

El feedback puede estar presente antes de la realización de una acción, en forma de feedback preventivo, durante ella, o al finalizarse, manifestándose a través de los mensajes de error o de cierre.

El feedback debe mostrar las variables físicas del sistema, puede surgir cuando ciertas acciones o datos fueron olvidados, cuando se necesita informar sobre la evolución de un proceso cuyo tiempo de respuesta es demasiado largo, o cuando se cometieron errores.

Antes de analizar cómo diseñar de la mejor manera el feedback del sistema, se deben considerar dos cuestiones: la inmediatez y la granularidad del feedback.

Aspectos del Feedback	Explicación
Inmediatez	■Hace referencia a la rapidez en que surge el feedback respecto a la acción correspondiente efectuada por el usuario. ■Determina el lapso de tiempo entre la acción y la reacción.
Granularidad	■Se refiere al tipo de acción a la que está asociado el feedback. ■Si el feedback surge ante acciones muy elementales, entonces el feedback tiene una granularidad muy fina, si surge ante transacciones, se dice que es de grano grueso.

Tabla 3.6: Aspectos relacionados al Feedback.

Entonces, respecto a los aspectos de diseño, se debe considerar las siguientes recomendaciones:

Recomendaciones	Explicación
Analizar la información expresada	■El feedback debe ser informativo. ■Debe expresar claramente el estado del sistema, el estado del proceso o del diálogo que se esté llevando a cabo con el usuario. ■Debe dar datos precisos, correctos, adecuados al contexto, a la sesión y a la actividad que esté realizando el usuario.
Determinar categoría o clases de feedback	■Es importante que el usuario cuente con diferentes clases de feedback en su interacción con el sistema. ■Debe existir al menos un feedback para aclaraciones, para confirmaciones, para los procesos -tanto en su iniciación, duración como en su finalización-, para los estados del sistema, para los errores.

Diseñar la visualización adecuada	■Se debe representar al feedback de la manera más entendible y significativa para el usuario. ■La representación visual puede ser compleja como en el caso de visualizar la progresión de un proceso en marcha, en donde se recurre a animaciones. ■La forma en que se visualiza el feedback debe ser homogéneo ante circunstancias similares, o ante cada categoría de feedback.
Analizar la inmediatez	■La aparición del feedback debe ser inmediato a la acción que lo produce, debe dar una respuesta rápida ante la misma. ■Se debe evitar respuestas lentas donde el usuario ya comienza a concentrarse en otros objetivos o entra en otros contextos.
Analizar la granularidad y aparición del feedback	■Hay que determinar a qué tipo de operaciones o acciones se le va a asociar el feedback y cuándo es necesario que aparezca. ■Hay que identificar las actividades a confirmar, validar, rechazar, corregir, aclarar o que determinan un procesamiento significativo. ■Si se asocia al feedback a operaciones muy elementales –grano fino-, el diálogo con el sistema termina siendo muy interrumpido y controlado. ■Hay casos donde es necesario proveer feedback de grano grueso, a nivel de transacciones o actividades más generales.
Diseñar el feedback ante una situación de error	■Como regla general, los errores deben ser reportados al usuario tan rápido como se haya concretado, efectivamente producido y detectado, para facilitar la posibilidad de reparación del mismo. ■Hay que analizar cuándo realmente es necesario que surja el mensaje de error para evitar tanto una interfaz rígida y controladora, como por el contrario una interfaz descuidada. ■Por ejemplo, ante una entrada de datos incompleta por parte del usuario, la interfaz debería esperar. Puede que el usuario deje momentáneamente los casilleros por la mitad y luego los culmine. ■Hay que determinar la mejor forma de representar visualmente el error cometido y el estado erróneo resultante. Se puede incluir una animación de lo ocurrido, con atajos donde el usuario pueda “saltar” a un estado previo seguro.
Diseñar un feedback preventivo	■En una interfaz visual, los conceptos, funciones, objetos son representados visualmente ante el usuario, por lo que hay una relación entre los elementos de la aplicación con sus representaciones dentro de la interfaz. ■A veces por determinadas circunstancias que pueden surgir dentro de la interfaz como desde la componente de cómputos, el acceso a estos elementos no está permitido, por lo que este estado debe reflejarse en las representaciones de los mismos. ■La interfaz del usuario debe visualmente manifestar la habilitación o inhabilitación de los elementos en las pantallas. ■Debe mostrar claramente lo que está activo de lo inactivo, lo permitido de lo restringido, para evitar que el usuario realice actividades infructuosas. ■Hay que analizar la forma visual que tomará el feedback preventivo. Se puede indicar la acción no permitida de alguna manera, con algún color o ícono, puede verse inhabilitada, por lo que no va a reaccionar ante el clic del mouse, o directamente puede no ser mostrada en forma momentánea.
Proveer feedback para las demoras	■Es fundamental que se le informe al usuario qué está sucediendo más que nada cuando los tiempos de respuestas de algún proceso en curso, son largos. ■Hay que tener en cuenta que todo sistema tiene un determinado ritmo de trabajo y de respuesta que el usuario se va acostumbrando. Esto hace que el usuario construya cierta expectativa sobre los tiempos de demora del sistema. ■Cuando el sistema tarda más que de costumbre, la interfaz debe darle al usuario un feedback indicando que el sistema no está muerto y está procesando.

Analizar el uso de feedback para reflejar el modo corriente	■El feedback también puede utilizarse para reflejar el modo en que está funcionando la interfaz del sistema. ■Puede ocurrir que, en un determinado momento, la interfaz o alguna de sus componentes esté funcionando en modo de edición o no, si permite múltiple selección o no, si admite modificación o no, si permite arrastre o no, entre otros. ■Por ejemplo, hay editores que tratan la entrada de textos como comandos si están en modo "input". ■En general, los modos no se pueden evitar pero un feedback explícito debería mostrar el actual modo.
Analizar el uso de las diferentes representaciones del cursor como feedback	■Esto se refiere a que al cursor del mouse o de cualquier otro dispositivo de apuntamiento, se le puede asociar diferentes imágenes dentro del espacio de la pantalla. ■Esto puede aprovecharse para manifestar determinados estados o transmitir cierta información. ■Por ejemplo, el cursor "mano" denota la posibilidad de cliquear esa opción, mientras que el cursor en forma de "anteojos" puede expresar el estado de consulta en vez de edición.

Tabla 3.7: Recomendaciones de Diseño del Feedback.

Importante

■El feedback es un recurso muy importante que debe ser aprovechado al máximo, puesto que de él depende la confiabilidad del sistema y su capacidad de transparencia.
■Un mal diseño del feedback puede llevar a malas interpretaciones del estado del sistema y permitir además, que el usuario continúe interactuando en un estado inseguro y erróneo.

Gestos

Al igual que en el caso de manipulación directa, el gesto debe estar diseñado de tal manera que simplifique la interacción del usuario y no la complique aún más.

Por lo tanto, debe ser diseñado con cautela respectando las siguientes consideraciones:

Recomendaciones	Explicación
Analizar los gestos admitidos por el sistema	■Hay que estudiar cuáles serán los trazos que el usuario podrá hacer y que serán admitidos e interpretados por el sistema. ■Estos trazos deben ser significativos y representativos para el usuario y para la acción o concepto que él pretende expresar a través de los mismos. ■Si para el usuario es natural realizar determinado gesto, por ejemplo el de borrar parte de un dibujo con una goma, y el sistema se lo permite, sentirá un alto grado de satisfacción y lo incentivará a seguir utilizando este método visual.
Determinar la simplicidad de realización del trazo	■Esto significa evaluar de alguna manera, si la entrada de gestos simplifica la interacción del usuario con el sistema. ■Si se requiere del usuario un muy buen pulso, o realizar un grafismo o marca demasiado delineada, que implique varios intentos de su parte, esto determina una cierta complejidad que debe ser evitada.
Establecer los dispositivos de interacción a utilizar	■Los gestos por lo general son realizados por el usuario a través de dispositivos de interacción como mouse, lápiz ópticos u otros comandos. Pero también, deben estar disponibles a través de otras alternativas como el teclado o el dedo en el caso de pantallas digitales.
Analizar la dirección de los gestos	■Por lo general, los gestos es una interacción que va del usuario al sistema, pero por cuestiones de homogeneidad puede que la interfaz también se exprese en forma gestual. ■Hay que analizar si es necesario que la interfaz también utilice este recurso, determinando un lenguaje gestual bidireccional.

Analizar el alcance de los gestos	■La interfaz debe expresar claramente dónde y cuándo se admite la realización de gestos. ■El usuario debe tener en claro, si un gesto puede sobrepasar una ventana, si puede ir de un objeto de interacción a otro, si puede marcar ciertos elementos de la pantalla.
Establecer las funciones sintácticas asociadas	■Los gestos deben estar acompañados de funciones sintácticas que permitan su control y manejo. ■Se puede proveer funciones donde se permita borrar los trazos realizados, desmarcar, permitir que las marcas sean imprimibles.

Tabla 3.8: Recomendaciones de diseño para Gestos

Dependencias

Una dependencia es una relación entre dos o más objetos de la pantalla que debe estar visualmente establecida y mantenida coherentemente a lo largo del tiempo.

Es muy importante que el usuario sepa de la existencia de estos objetos dependientes puesto que al modificar uno de ellos, se modificarán los restantes también. Se deben considerar fundamentalmente las siguiente cuestiones:

Recomendaciones	Explicación
Visualización implícita de la dependencia	■Es necesario contar con elementos o recursos visuales que distingan los objetos relacionados. ■Se puede utilizar colores comunes, fondos específicos, bordes, lugares o mensajes que denoten la dependencia entre los objetos.
Duración de la dependencia	■Hay que analizar hasta cuándo dura la relación de dependencia entre los objetos, si durante una sesión, en todas ellas, mientras tenga sentido, mientras sea necesario.
Alcance de la dependencia	■Hay que analizar la distancia permitida entre los objetos que están involucrados en una dependencia. ■Los mismos pueden pertenecer a diferentes ventanas, diferentes páginas o estar en distintos marcos, generando costos adicionales para la interfaz que debe manejar y controlar la dependencia entre ellos.
Configuración de la dependencia	■Se refiere a si se permite activar o desactivar esos vínculos entre los objetos que están determinados por la dependencia.

Tabla 3.9: Recomendaciones de diseño para Dependencias

Animación

Las animaciones establecen un mecanismo visual que es muy poderoso y expresivo para denotar situaciones dinámicas en espacios limitados. Pero sus principales ventajas se logran específicamente cuando el usuario tiene el foco de atención en la animación, o sea mientras él la está observando.

En el caso que el usuario esté desarrollando una determinada tarea y estén activadas ciertas animaciones no relacionadas con la misma, puede provocar distracción y molestias al usuario. Es necesario seguir una serie de recomendaciones de diseño para lograr un buen uso de las mismas:

Recomendaciones	Explicación
Analizar la legibilidad y claridad de la animación	■Hay que evaluar si la animación está bien diseñada y expresa claramente el proceso dinámico que represente. ■Hay que analizar si los cuadros son suficientes para representar el proceso animado y si la velocidad es la adecuada, o hace confusa la interpretación de la animación.
Aplicación de la animación	■Hay que evaluar en qué casos es importante recurrir a una animación para expresar un concepto. Puede ser apta para mostrar conceptos dinámicos, para las demoras, ayudas que indiquen cómo se hace tal actividad, mensajes de error que repitan las acciones efectuadas y marquen dónde y cómo se efectuó el error.

Repetición de la animación	■En algunos casos puede que sea necesario que la animación se repita más de una vez, para poder reflejar correctamente el mensaje que representa.
Arranque de la animación	■Por lo general, la animación no debe estar siempre activada llamando la atención en forma permanente. El usuario debería decidir cuándo activar la animación y cuándo culminarla. ■Los mecanismos tradicionales para activar una animación es al cliquearla, al tocarla o mediante un clic explícito de un botón de arranque. ■Se podría proveer mecanismos automatizados para la activación que detecten el foco de atención del usuario.
Funciones sintácticas de la animación	■Analizar las funciones necesarias para que el usuario pueda controlar y manejar la animación. ■Se pueden proveer funciones que agreguen o quiten cuadros, que aumente o disminuya la velocidad de la misma, que se configure las veces de repetición, que se pueda programar sus arranques y duraciones, entre otros.

Tabla 3.10: Recomendaciones de diseño para Animaciones

Exploradores de Objetos

Los exploradores o grillas interactivas son componentes de interacción bastantes complejos que permiten presentar gran cantidad de información en manera concentrada y organizada. Los mismos pueden tener ciertas capacidades a nivel de cada celda, filas, columnas y browser en general. Se deben considerar ciertas recomendaciones de diseño para que estas capacidades y servicios puedan ser aprovechados por el usuario.

Recomendaciones	Explicación
Analizar si el contenido es adecuado	■Hay que analizar si la información está bien expresada, si efectivamente visualiza los múltiples datos con una correcta descripción de sus atributos. ■Hay que verificar la correcta representación de los tipos de datos asociados a las celdas, que tengan el formato adecuado. ■Hay que evaluar la necesidad de permitir celdas con ciertas capacidades o servicios incluidos, como el acceso a una calculadora desde una celda numérica, o el acceso a un calendario desde una celda fecha.
Determinar el estado del explorador	■El usuario debe tener en claro cuál es el estado actual de la grilla con la que está interactuando.Si está en modo edición, o sólo puede ser consultado, si se permite múltiple selección o simple selección, si puede eliminarse filas, si se pueden renombrar las columnas, etc.
Establecer funciones sintácticas asociadas	■El explorador debe proveer funciones a nivel de celdas, de filas, de columnas y a nivel de grilla en general. ■Se debe permitir la ordenación por determinados criterios, mover objetos de lugar, permitir copiar y pegar celdas. ■También debe permitir el agregar, mover y quitar filas o columnas, filtrar, proveer funciones de búsquedas y accesos directos. ■A nivel de grilla debe tener funciones de asistencia, aceptación, cancelación, impresión de los datos, entre otros.
Configuración visual del explorador	■El usuario debe poder configurar al explorador de acuerdo a su estilo y preferencias. ■Se le debe permitir cambiar tipos de letras, tamaños, colores, fondos, hacer un zoom del contenido, determinar la cantidad de filas a desplegar, etc.
Configuración por defecto	■Debe contar con configuraciones predeterminadas a la que pueda recurrir el usuario si quiere deshacer sus configuraciones personales.

Tabla 3.11: Recomendaciones de diseño para Exploradores de Objetos.

Cajas de Diálogo

Las cajas de diálogo es un recurso visual compuesto por varios objetos de interacción.

Dentro de ellas, se pueden encontrar casilleros de texto, listas seleccionables, listas deslizables, botones, imágenes, animaciones, que a su vez tienen sus propias cuestiones de diseño.

En el momento de diseñar una caja de diálogos es recomendable tener en cuenta las siguientes cuestiones:

Recomendaciones	Explicación
Analizar el contenido de la caja de diálogos	■Es necesario analizar el uso adecuado de los controles y objetos de interacción que componen la caja de diálogos. ■Hay que evaluar si son correctos para cada concepto semántico que representen.
Distribución de los objetos de interacción	■Las componentes de la caja de diálogo deben estar bien diagramadas y dispuestas. ■Deben tener el tamaño y ocupación dentro del área acorde a la longitud de la información a representar. ■No debe existir superposición o poco espacio entre las mismas, que provoque una visualización engorrosa del contenido.
Analizar la ordenación de los elementos	■La forma de ordenación y disposición de los elementos de una caja de diálogos debe ser significativa. ■Puede agruparse en forma alfabética, por orden de importancia, por nivel de prioridad, por grado de uso. ■El usuario debe percibir una lógica de sectorización o un criterio de ordenación.
Funciones sintácticas asociadas	■La caja de diálogos debe proveer servicios a nivel de componente como a nivel de caja en general. ■A nivel de componente, se puede proveer valores por defecto, ayudas contextuales, ejemplos, acotaciones configurables. ■A nivel de caja de diálogo, se debe permitir una ayuda general, salida, cancelación, estado por defecto, impresión, suspensión, guardado, entre otros.
Analizar el feedback asociado	■Hay que determinar el tipo de feedback que se va a proveer y el alcance del mismo. ■Con respecto al tipo de feedback, se refiere a si se provee validaciones, confirmaciones, mensajes de error, aclaraciones. ■Mientras que, el alcance significa si es a nivel de control o componente o a nivel de la caja en general.
Determinar aspectos de visualización	■La caja de diálogos debe estar visualmente definida, con una correcta elección de los colores, fondo, formato, tipografía, botones, títulos.

Tabla 3.12: Recomendaciones de diseño para Cajas de Diálogo.

Caso de Estudio 3 - Normas de Diseño de Interfaces Visuales

En esta sección se presenta un caso de estudio, donde trabajaron alumnos de la cátedra Diseño centrado en el Usuario - ciclo lectivo 2005. Esta materia pertenece al plan de estudios de la carrera Licenciatura en Informática de la Universidad Nacional de La Plata.

El caso descripto ilustra como debería analizarse la interfaz de sistema según los Principios de Diseño para Interfaces Visuales.

Objetivo

El objetivo de este trabajo fue que los alumnos tuvieran que aplicar los principios de diseño para Interfaces Visuales en un proceso de evaluación de la interfaz de usuario de un sistema de gestión de una Clínica Médica. La evaluación se realizó en pequeños grupos y luego, entre todos los integrantes del curso, se debatieron y propusieron diferentes alternativas de solución.

Características del Caso de estudio

El sistema facilita la gestión administrativa de una clínica médica: altas, bajas, consultas y modificaciones de pacientes, profesionales, turnos, guardias y consultorios. Este sistema fue desarrollado por un grupo de alumnos en forma personal y fue expuesto en la cátedra para su evaluación. Siempre es mejor la experiencia si se evalúan sistemas que otros alumnos hayan desarrollado o utilizan habitualmente. La pantalla inicial del sistema es la siguiente:

Figura 3.13

Desarrollo del Caso de estudio

A continuación se presentará la evaluación realizada por un grupo de estudiantes, identificando el cumplimiento o no de cada uno de los principios de diseño para Interfaces Visuales. Describe cada recomendación y muestra las partes del sistema que presentan inconvenientes.

Ventanas

Respecto a las funcionalidades sintácticas, todas las ventanas del sistema proveen minimización, maximización, redimensión, movimiento, incluso aquellas donde no es necesario.

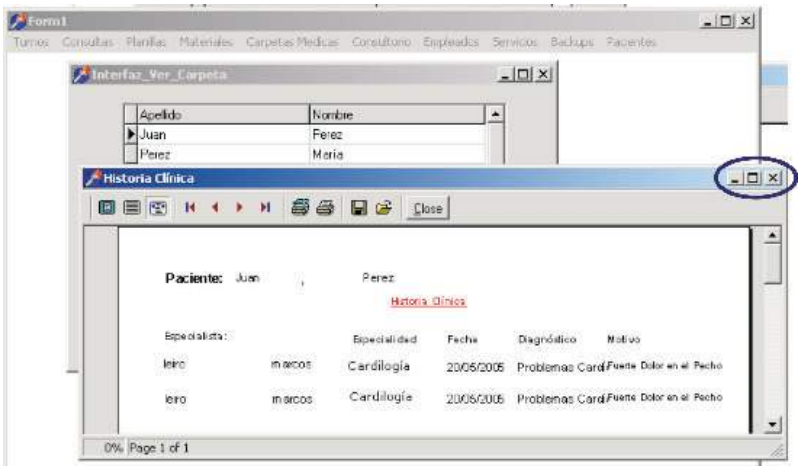


Figura 3.14

El cambio de tamaño de la ventana no va acompañado de una re distribución de los elementos que la componen.

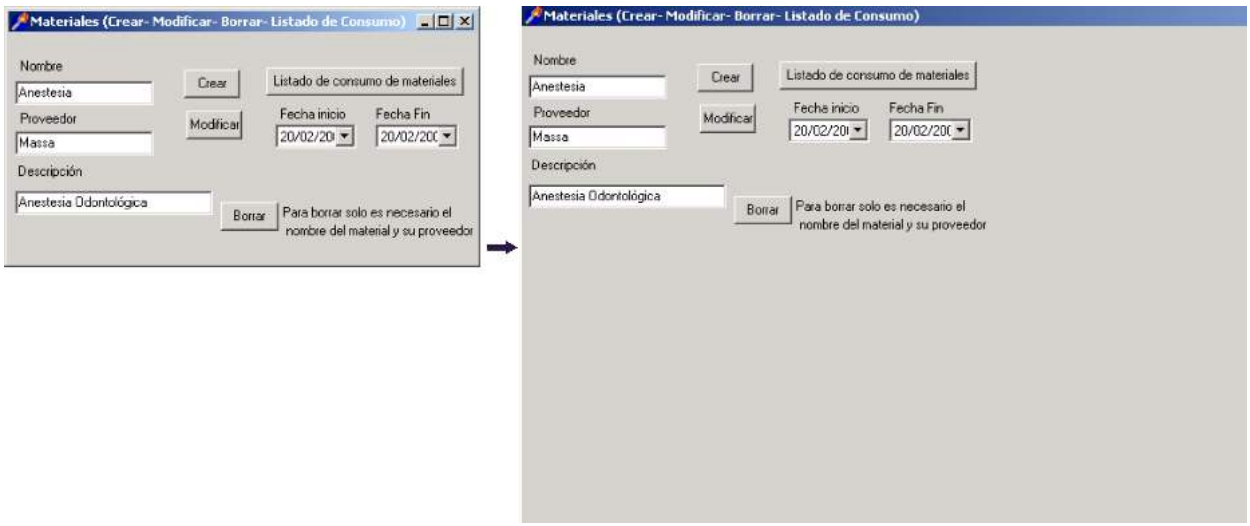


Figura 3.15

La mayoría de las ventanas son accedidas mediante la barra de menús de la aplicación - figura 3.16- y en algunos casos a través de botones - figura 3.17

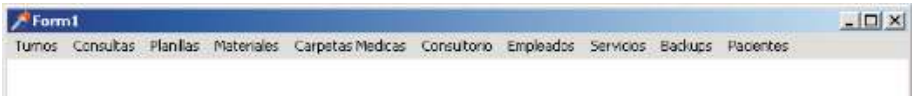


Figura 3.16

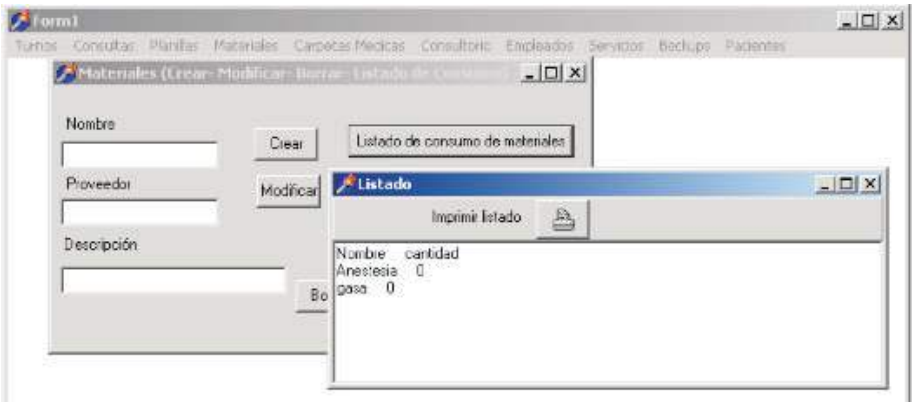


Figura 3.17

Respecto al modo de activación de las ventanas, todas se abren en modo exclusivo y no posee una lista de ventanas activas.

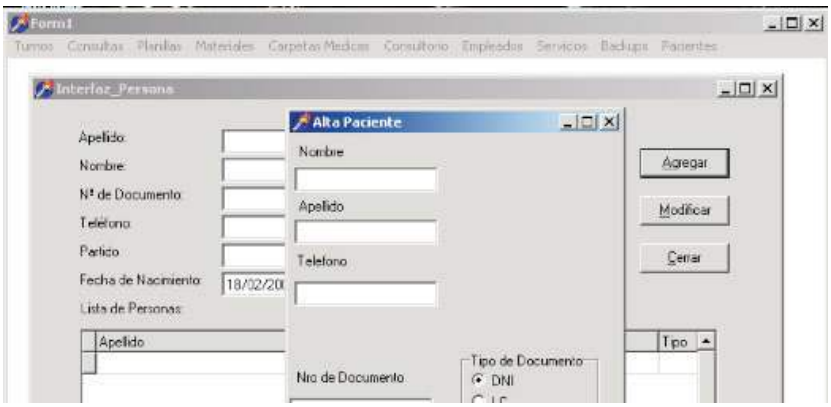


Figura 3.18

En cuanto a el ámbito de la ventana, se abren dentro de la ventana madre, pero pueden desplazarse fuera de ésta.

El contenido de las ventanas, en general no es homogéneo y no están bien distribuidos. Por ejemplo, supongamos que se desea dar de alta un consultorio. Seleccionamos la opción del menú principal Consultorio -> Crear . La ventana que se despliega es la siguiente:

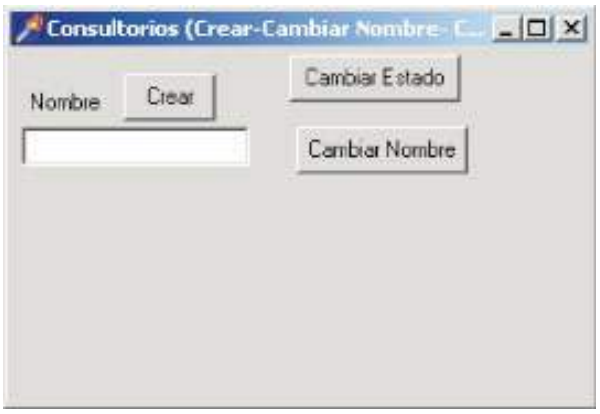


Figura 3.19

Al presionar los distintos botones, aparecen más componente dentro de la misma ventana que no están bien distribuidos y dificultan la interacción. Por ejemplo si presionamos sobre el botón “Cambiar Estado”, aparecen más componentes:

Figura 3.20

Luego si se quiere cambiar el nombre del consultorio se agregan a la ventana estos componentes.

Figura 3.21

La ventana posee componentes que no se encuentran alineados y no se sabe a qué acción (Creación, Cambio de estado o de Nombre) pertenecen. Hay un problema de diseño en cuanto a las operaciones disponibles y la forma en que pueden llevarse a cabo en la ventana. Un ejemplo de formato no homogéneo se ilustra a continuación. Las ventanas son muy diferentes entre sí. Las componentes no están posicionados siempre en el mismo lugar.

Figura 3.22

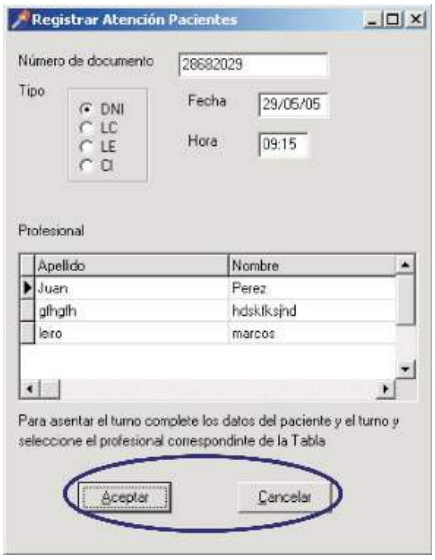


Figura 3.23

No todos los listados poseen un título, y aquellos que lo tienen poseen distinto formato de fuente y posición dentro de la ventana, e incluso se titulan diferente dependiendo del contexto.

Figura 3.24

Figura 3.25

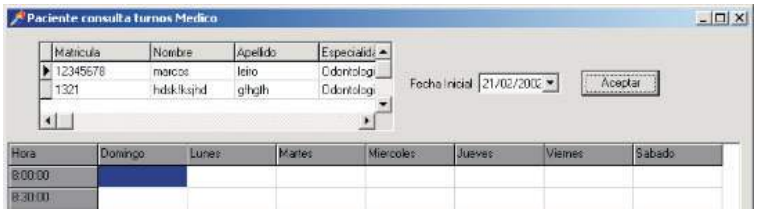


Figura 3.26

Este criterio presenta algunos inconvenientes. El orden en que aparecen las opciones no es el más natural ya que por ejemplo el alta de cualquier entidad debería ir antes de la modificación y la baja.

Figura 3.27

Los nombres de las acciones no son entendibles, no tienen mensajes explicativos asociados y tampoco el mismo nivel de abstracción. Por ejemplo en el menú Pacientes de la imagen superior, la opción Cancelar significa cancelar el turno de un paciente. Semánticamente debería estar en el menú Turnos.

El menú Turnos tampoco es muy sencillo de entender, porque se mezcla la disponibilidad del médico con el alta baja y modificación del turno para un paciente. Además se agrega la posibilidad de registrar una guardia (que es sin turno) y de registrar la atención de un paciente, que tampoco tiene que ver con el turno.

Figura 3.28

Los menús no tienen un nivel superior de anidamiento de 2 niveles

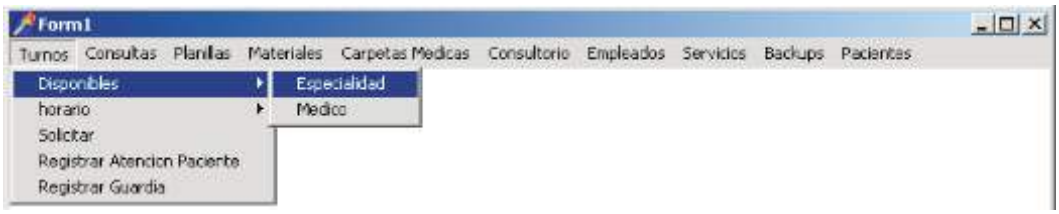


Figura 3.29

Respecto a la categorización, en los puntos superiores hemos visto ejemplos concretos donde este aspecto no está considerado. Por ejemplo cancelar un turno a través de la opción Pacientes -> Cancelar, o registrar una atención del servicio de Guardia a través del menú Turnos (ver figura 3.29).

Las opciones del menú se acceden utilizando el mouse, desde una única opción dentro del Menú. Sin embargo, encontramos algunas excepciones: dar de alta un turno o listar los turnos, donde la misma ventana se encarga de ambos aspectos.

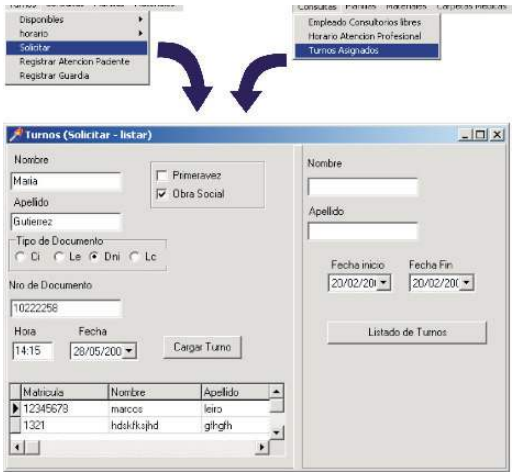


Figura 3.30

Ídem las opciones Materiales, Consultorio, y Empleados donde una misma ventana se encarga de todas las funciones.

No brinda mecanismos para modificar la jerarquía o la categorización de los menús, es decir, no brinda ningún mecanismo de reconfiguración de las opciones.

Feedback

El sistema no posee un feedback homogéneo, ni a nivel de ventana ni de acción.

Por ejemplo las operaciones de alta de un material, empleados, servicios informan a través de una ventana el éxito o no de la operación.

Figura 3.31

Sin embargo la operación de dar de alta un informe en una carpeta médica no brinda ningún mensaje que informe el éxito de la operación.

En la ventana Interfaz_Servicios, que permite agregar, modificar y eliminar servicios, informa solo acerca del éxito de la operación agregar y no de las restantes.

Los mensajes de notificación de error o altas o bajas generalmente tienen el mismo formato. Por ejemplo el mensaje de confirmación de cambio de estado de un consultorio es el siguiente:

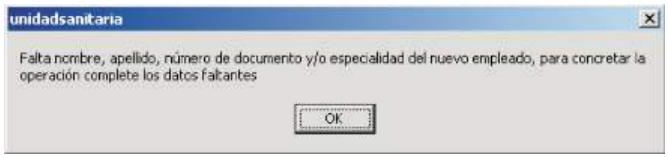


Figura 3.32

En este punto tampoco es homogéneo. No siempre es correcto y adecuado al contexto. Por ejemplo aquí el sistema notifica que el paciente existe, pero no es así. En realidad se está escribiendo como DNI letras y no un número y ese es el motivo del error.

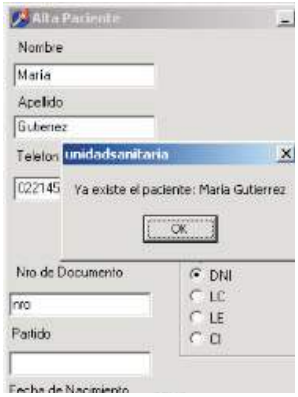


Figura 3.33

Otro ejemplo claro: en esta ventana se informa que hay un error pero no dónde.

Figura 3.34

En cuanto a la aparición de feedback, en todos los casos aparece al finalizar la operación y no luego de completar ciertos campos. Las respuestas son rápidas. Respecto a la necesidad del feedback, como mencionamos previamente, el sistema realiza las validaciones al confirmar los datos ingresados. Sin embargo la notificación de los errores es secuencial, no indica de una sola vez todos los errores cometidos.

Por ejemplo, la operación de registrar un turno. La ventana, como vemos a continuación, solicita varios datos. Muchos de ellos son obligatorios y no solo no están resaltados, sino tampoco se validan e informan a la vez.

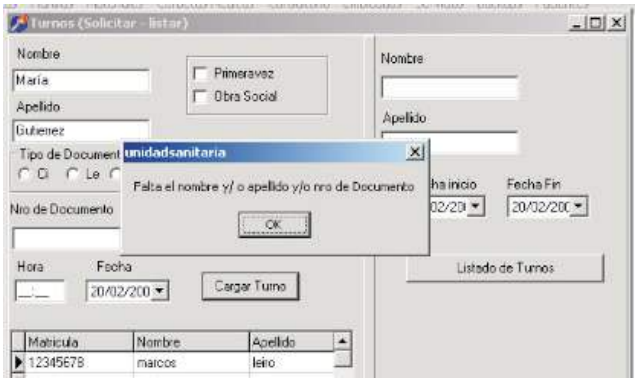


Figura 3.35

Luego de completar los datos solicitados, aún faltan datos obligatorios como el horario. La confirmación de los datos de la ventana vuelve a emitir un mensaje de error, pero ahora respecto al horario.

Figura 3.36

Este punto está contemplado en algunos puntos, por ejemplo la selección de las fechas se realizan a través de la selección de la misma en un calendario, o el tipo de documento siempre se realiza a través de la selección entre los disponibles.

Figura 3.37

Sin embargo existen casos donde, si bien el feedback preventivo está contemplado, su implementación no es clara. Por ejemplo, la ventana que permite manipular los consultorios oculta/visualiza componentes de acuerdo a la operación que se desea realizar (ver figuras 3.19, 3.20 y 3.21).

No se informa al usuario el grado de avance de ninguna de las operaciones. Por ejemplo, la ventana que permite realizar un Backup no brinda ninguna información acerca del grado de avance de la operación. El sistema queda como muerto. Tampoco existe el feedback que refleje el modo corriente o mapee movimientos de dispositivos de apuntamiento.

Manipulación Directa

No existen botones íconos en el sistema. Los botones son fáciles de apuntar y seleccionar. El sistema está diseñado para interactuar a traves del mouse, exceptuando la entrada de datos o cuadros de texto que se realiza por teclado. La tecla Tab no se encuentra programada, no tiene una secuencia lógica dentro de una ventana que facilite la entrada de datos.

No brinda mecanismos de arrastrar y soltar para ninguna de sus componentes.

Gestos

No hay manejo de gestos en el sistema.

Dependencias

En algunos casos, las dependencias se encuentran correctamente programadas y en otro no. Por ejemplo, la operación para cambiar el nombre a un servicio es un ejemplo donde este criterio no se respeta. En primer lugar seleccionamos el nombre del servicio a modificar de la lista de disponibles y escribimos el nuevo nombre en el casillero inferior.

Figura 3.38

Sin embargo, al desplegar la lista vemos el nombre modificado.

Figura 3.47

Animaciones

El sistema no provee animaciones

Exploradores de Objetos

Numerosas ventanas muestran todos los datos en grillas, por ejemplo la figura 3.17, la 3.30, la 3.35. Ninguna de ellas posee funciones sintácticas asociadas que permitan ordenación por diferentes criterios, mover objetos de lugar, copiar y pegar.

La configuración del explorador es estática en todos los casos. El número de columnas, el tipo de letras, la cantidad de filas, etc. es fijo y no se ofrecen operaciones para cambiarlo.

Es editable a nivel de celdas, pero no es posible registrar ningún cambio. Puede modificarse el largo de la columna, no el ancho de la fila. No soporta múltiple selección.

Conclusiones

A modo de conclusión de la evaluación de este sistema, los alumnos abordaron a la siguiente tabla de clasificación:

Principio	Calificación
Ventanas	☑ Bien
Menú	☑ Regular
Feedback	☑ Regular
Manipulación directa	☑ Bien
Gestos	---------
Dependencias	☑ Bien
Animaciones	--------
Explorador de Objetos	☑ Regular

Ejercitación

A. Indicar si las siguientes expresiones son verdaderas o falsas. Tachar lo que no corresponda:

- La manipulación directa es un tipo de diálogo secuencial. V F
- Para proveer múltiples hilos de diálogo entre el usuario y el sistema, es necesario que la interfaz tenga múltiples ventanas. V F
- Los mensajes de error son un tipo especial de feedback. V F
- Los menús permitieron aislar a los usuarios de cuestiones sintácticas. V F
- El usuario es quien debe controlar y mantener las dependencias entre los objetos relacionados. V F
- Las animaciones siempre optimizan la interacción humano-computadora. V F
- Los browsers permiten una organización centralizada de la información. V F
- El usuario al interactuar con una caja de diálogo debe percibir un criterio de ordenación en la disposición de los objetos que la componen. V F

B - La interfaz del usuario es el software que soporta el diálogo bidireccional entre el usuario y el sistema. Indicar con una cruz, qué dirección del diálogo asocia cada uno de los paradigmas de interacción: si va del usuario al sistema, del sistema al usuario, en ambas direcciones o en ninguna.

Paradigmas de interacción visual				
Menús				
Manipulación directa				
Feedback				
Dependencias				
Animaciones				
Explorador de objetos				
Cajas de diálogo				

Referencias

- Foley, Jim; Dam, Van (1991). Fundamentals of Interactive Computer Graphics.
- Caroll J.M. Mack R.L. y Kellog, W.A. (1997). Interface Metaphors and User Interface Design. Human-Computer Interaction INTERACT '87. North-Holland, Amsterdam.
- Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, MA.
- Hutchins, E.L, Hollan, J.D., and Norman, D. A. (1986) Direct Manipulation Interfaces, in D. A. Norman and S. W. Draper (eds), User-Centered System Design, Lawrence Erlbaum, 87124.
- Shi Kuo Chang (1985) Image Information Systems. Proceedings of the IEEE on Visual Communications Systems, Vol. 73.
- Shi Kuo Chang (1986). Image Database Systems. Academic Press.
- Shi Kuo Chang (1987). Visual Languages: A Tutorial and Survey. IEEE Software Magazine, Vol. 4.
- Shi Kuo Chang (1990).Principles of Visual Programming Systems, (ed.), Prentice-Hall.
- Shi Kuo Chang (1990).Visual Languages and Visual Programming. Plenum Publishing Corporation.
- Shi Kuo Chang (2001). Software Engineering and Knowledge Engineering. Columbia University Press.
- Kevin Mullet y Darrell Sano. (1995). Designing Visual Interfaces.Sun Microsystems Inc.

Capitulo IV

Cualidades de una interfáz icónica

04

Cualidades de una Interfaz Icónica

Desde la percepción humana, se requiere un entrenamiento mayor y más específico para leer un libro que para contemplar y apreciar una pintura, aunque la forma de construcción de la imagen puede ser tanto o más compleja que la confección de un texto escrito.
Si la imagen posee un enorme poder de expresión y comunicación, el ícono lo supera aún más.

Temas a Tratar

I -Introducción
II -Características de una Interfaz Icónica
- El Diseño Icónico y los 7 Pasos de Norman
- El Diseño Icónico y la Correspondencia
- El Diseño icónico y la Consistencia

III -Ventajas de una Interfaz Icónica
IV -Complejidad del Diseño Icónico
V -Especificación Formal de una Interfaz Icónica
Ejercitación
Caso de estudio
Referencias

I - Introducción

En el capítulo anterior, se pudo observar las características de las Interfaces Visuales y cuáles son sus beneficios desde el punto de vista de los factores humanos. Entre las ventajas más interesantes, se pueden apreciar:

- El usuario dispone de una representación permanente de los objetos y de las funciones de interés.
- El usuario se abstrae de emplear una sintaxis compleja para expresar sus intenciones a cambio de utilizar recursos visuales como selección, arrastre, clics de objetos.
- El usuario puede realizar múltiples operaciones simultáneas, cada una reflejando una ejecución y efecto inmediatos.

Estas cualidades aún pueden ser más elocuentes si se exige que las representaciones visuales que se utilicen para los objetos, como la manipulación de los mismos, estén regidas mediante un diseño cuidadoso, bien estudiado y que sea comprensible dentro de la realidad del usuario.

Este es el caso de las interfaces icónicas, que están encuadradas dentro de las interfaces visuales. Son una clase especial de las mismas, en donde se le exige una visualización significativa y un comportamiento fiel, respecto al objeto representado.

Veamos la diferencia entre una interfaz visual y una icónica, mediante un ejemplo. Supongamos que en una parte de la pantalla, se necesita mostrar la temperatura de un ambiente. En el caso de las interfaces visuales, se podría utilizar un rectángulo con diferentes tonalidades de naranja para indicar el grado de calor y de violeta, para el frío. Esta sería una representación visual válida, puesto que se utilizan recursos visuales como figuras geométricas y el uso del color.

Pero, en el caso de un diseño icónico, esto no alcanzaría, no sería una representación correcta. Para el concepto de temperatura, sería necesario utilizar una visualización más familiar, más representativa. Por consiguiente, se debería en primera medida, utilizar colores que universalmente simbolizan el calor y el frío, que serían la gama del rojo y del azul, respectivamente. Y segundo, se debería reemplazar la figura del rectángulo, por una imagen más significativa como la de un termómetro, directamente.

En un diseño icónico, se trabaja con un nivel muy elevado de representatividad, utilizando recursos visuales más elaborados como íconos, símbolos, metáforas, signos, demostraciones o simulaciones extraídas de la realidad del usuario.

A través de estos recursos se deben representar icónica o metafóricamente todas las componentes del sistema y sus funcionalidades, sus estados, datos de entrada, de salida, sus entidades.

Además, el diseño icónico puede aplicarse tanto en el lenguaje de interacción provisto por el sistema, en la representación de la información, como también dentro de los constructores u operadores de un lenguaje de programación.

Se puede aplicar este diseño sobre todas las técnicas de interacción siempre y cuando sean modificadas para mantener y reflejar la coherencia y representatividad icónica obligatoria.

Los teléfonos celulares actuales, la televisión digital y los distintos dispositivos que podemos encontrar en la actualidad que disponen de pantallas reducidas han potenciado el uso de íconos como metáfora para el acceso a distintas aplicaciones. Además la potencia de cálculo y las pantallas táctiles han facilitado el desarrollo de interfaces gestuales que se complementan con los íconos para enriquecer la experiencia de los usuarios. Por ejemplo, para eliminar una foto en un teléfono celular con cámara alcanza visualizar la foto en miniatura en el album de fotos del celular, “pincharla” con el dedo y arrastrarla hacia la papelera simbolizada claramente con un ícono.

II - Características de una Interfaz Icónica

El diseño icónico de una interfaz del usuario se distingue del diseño visual por la calidad y la semántica expresada a través de los recursos visuales que se emplean en él.

La filosofía icónica parte de proveer una imagen del sistema que concuerde fielmente con la representación mental que el usuario tenga sobre el problema. Apunta fundamentalmente a que el usuario perciba que su mundo real con el representado son compatibles.

Esto significa que se debe trabajar profundamente sobre modelos de usuarios para poder detectar cómo es su mundo, cómo él ve las entidades que maneja, cómo interactúa, de qué manera él trabaja con ellas.

De esta manera, el diseñador podrá ser capaz de construirle así, una simulación o una proyección “casi real” de ese mundo, sobre el espacio de la pantalla.

Como su nombre lo indica, en un diseño icónico, el instrumento visual más importante que se utiliza, tanto para visualizar la información como para expresar el diálogo, es el ícono.

Pero, ¿qué se entiende por el concepto de ícono?. En este contexto, un ícono es mucho más que una pequeña imagen dentro de un botón de una barra de herramientas.

La definición más adecuada de ícono fue especificada por Grittins [86] y Rogers [89], que lo describen

como una imagen, una figura o un símbolo que representa un concepto subyacente.

También, fue definido por Shi Kuo Chang [90] como un par (parte física, parte lógica), en donde la parte física sería la imagen del mismo y la parte lógica, su significado.

A través de esta última definición, se puede inferir que lo que distingue al ícono de una imagen tradicional, es que tiene una semántica o parte lógica asociada que debe estar coherentemente solidificada a su parte física, a lo largo de toda la existencia del mismo.

Ahora bien, otra duda que uno se puede cuestionar es ¿dónde se aplica realmente el diseño icónico? ¿solamente en la representación?. La respuesta correcta a estos interrogantes, es no.

Análogamente al diseño visual, el diseño icónico puede estar presente en cualquier parte de la interfaz. Puede ser aplicado en:

La Visualización Icónica:

En este caso, se utiliza al ícono como un medio de representación, para modelar metafóricamente tanto el dominio de la aplicación como los conceptos que se manejan a nivel de interfaz.

Se puede visualizar icónicamente todas las componentes y entidades del sistema, sus funcionalidades, datos de entrada, de salida, sus estados posibles.

Por ejemplo, se puede mostrar en la siguiente pantalla, la representación icónica de una sala de hospital, con el estado de libre u ocupado de sus camas.

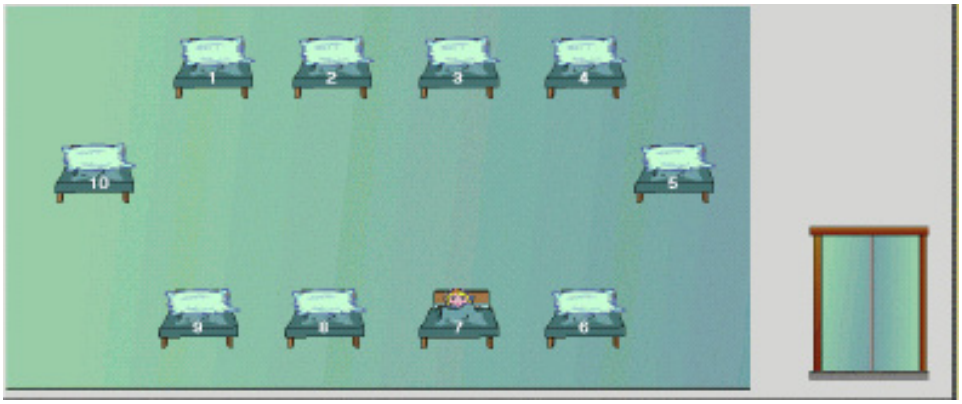


Figura 4.1: Ejemplo de visualización icónica

O también, es el caso del diseño icónico de un sistema groupware sobre conferencias, donde se visualiza claramente el estado de las butacas, si están libres u ocupadas, y el de los participantes, si solicitan o no permiso para hablar.

Figura 4.2: Otro ejemplo de visualización icónica

La Interacción Icónica:

El diseño icónico aplicado en la interacción entre el hombre y la máquina, da lugar a un diálogo asincrónico. Este diálogo puede estar acompañado por la selección de íconos, por una manipulación directa que debe ser significativa, por un feedback semántico, por gestos metafóricos, por una animación representativa, por menús icónicos, dependencias, mediante el cuál los usuarios llevan a cabo sus tareas en una forma natural y simple.

A través de la visualización e interacción icónica, el usuario interactúa con un ambiente casi real, donde se muestra reflejado y proyectado todo su mundo sobre el espacio de la pantalla.

Un ejemplo de una interacción icónica se puede demostrar en la siguiente pantalla, donde para guardar un escrito, el usuario debe trasladarlo desde el escritorio hacia el archivo de papeles, mediante el arrastre.

Figura 4.3: Ejemplo de interacción icónica

La Programación Icónica:

En este caso, se debe representar icónicamente cada constructor, sentencia u operador del lenguaje de programación. Esto trae como consecuencia, que la acción de programar se convierta en un proceso de ensamblado y armado de piezas icónicas que conforman las partes del programa.

Por ejemplo, como trabajo de grado en la Facultad de Informática de la UNLP, se construyó un lenguaje icónico para consultar bases de datos relacionados, denominado CLAIV, donde se representaron tanto las tablas de datos como los operadores del Álgebra relacional, mediante íconos representativos.

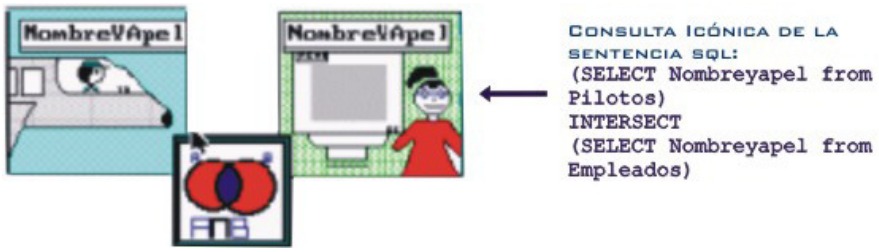


Figura 4.4: Ejemplo de programación icónica

En la mayoría de estos casos, la interfaz del usuario se convierte en un sistema icónico, a partir del cuál se provee un lenguaje especial conformado por la disposición, manipulación, arrastres, movimientos, acciones, gestos, animaciones de íconos, que conviven dentro de dicho sistema.

El lenguaje utilizado en este tipo de interfaces, llamado lenguaje icónico, consiste en un conjunto de sentencias icónicas construidas a partir de una sintaxis y semántica ya definida.

Estas sentencias se definen como un arreglo u ordenamiento espacial de íconos (miembros del sistema) puesto que surgen de una manipulación de los mismos sobre el espacio de la pantalla.

Como en cualquier lenguaje donde se realizan análisis sintáctico y semántico de las sentencias, aquí también se debe analizar e interpretar las sentencias icónicas en pos de alcanzar las acciones semánticas.

El análisis sintáctico de una sentencia icónica, denominado “compilación espacial”, es el análisis del arreglo espacial de íconos que determina la estructura sintáctica interna. Y, el análisis semántico consta de una interpretación de la sentencia icónica para determinar su significado. El mismo es denominado “interpretación espacial”.

Todo esto constituye un ambiente icónico que puede ser utilizado como una capa superior de otros sistemas tradicionales permitiendo la traducción de sentencias icónicas, movimientos y manipulación de íconos sobre la pantalla a comandos propios del lenguaje subyacente.

Estas interfaces a parte de proveer las ventajas atribuidas a los interfaces visuales respecto de las mejoras

en la interacción con el usuario, actúan como filtros donde muchos errores pueden ser detectados en ese nivel superior, permitiendo la generación de sentencias internas correctas.

Antes de comenzar con las ventajas de un diseño icónico, es importante analizar la relación entre el mismo con respecto a los 7 Pasos de Norman.

El Diseño icónico y los 7 Pasos de Norman

El concepto de los 7 Pasos de Norman, introducido por Donald Norman en 1988, hace referencia a las etapas que suceden dentro de la mente humana cuando un usuario interactúa con un sistema de cómputos en pos de resolver un problema.

Estos pasos se pueden describir mediante el siguiente gráfico:

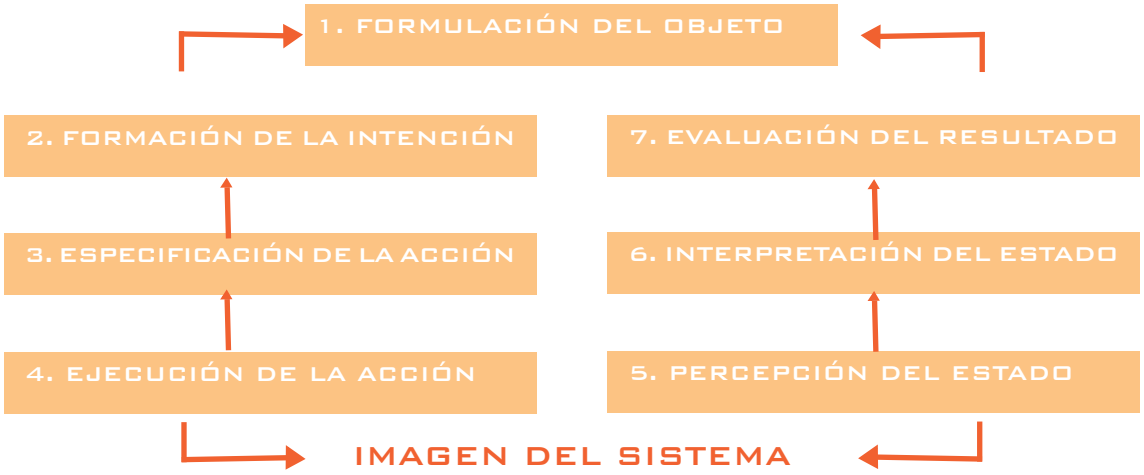


Figura 4.5: Los 7 pasos de Norman

Aquí, se puede observar que el usuario, frente a una actividad a desarrollar en la computadora, plantea un objetivo principal (paso 1), que para llevarlo a cabo debe formular una intención (paso 2). La misma se transformará en una acción que deberá ser ejecutada, afectando la imagen del sistema (pasos 3 y 4).

El estado del sistema frente a esta acción del usuario, se modifica. Dicho cambio debe ser detectado e interpretado por el usuario (pasos 5 y 6). Y finalmente, debe evaluar si el resultado de su accionar, satisface su objetivo inicial, o sea si realmente "hizo lo que quería" (paso 7).

Pero, ¿de qué manera se puede relacionar el diseño icónico con este proceso?. Uno puede partir sobre la definición del diseño icónico y analizar que, si en el mismo se insta a utilizar representaciones visuales y mecanismos de interacción que concuerden con la realidad del usuario, la imagen del sistema, su manipulación y la manifestación del feedback, debería ser conocido y entendido por él.

Entonces, mediante una interacción fácilmente manejable y de una representación física fácilmente perceptible, el diseño icónico intenta por definición, reducir el esfuerzo mental que el usuario debe llevar a cabo tanto en la ejecución de sus acciones como en la interpretación de las consecuencias de las mismas.

El Diseño Icónico y la Correspondencia

Hay que tener en cuenta que, el esfuerzo mental que el usuario debe emplear, cuando traslada su intención en requerimientos propios del sistema, se denomina "distancia de Ejecución". Y, el que el usuario debe realizar cuando analiza el estado del sistema en términos de su representación mental, se denomina "distancia de evaluación", (en inglés, Gap of Execution y Gap of Evaluation, respectivamente).

Unos de los principios básicos de diseño de interfaces, que permite minimizar los gaps de ejecución y evaluación, es el concepto de Correspondencia Directa, especificado por Hutchins y D.Norman, en 1986.

Este principio determina el grado de representatividad de los objetos que intervienen en el proceso interactivo, estipulando que debe existir concordancia entre los objetos computacionales y no computacionales.

Pero, ¿qué se entiende por objetos computacionales?. Los objetos computacionales son los conceptos, entidades, datos, funciones extraídos del problema real, que serán registrados, representados y manejados dentro del sistema. Por lo tanto, los elementos, funciones, conceptos extraídos del mundo real, son los objetos no computacionales, que serán representados y automatizados dentro del sistema, por los objetos computacionales.

Entre los objetos computacionales, tenemos:

■Los objetos semánticos: son las entidades, datos o funciones del dominio, representadas dentro del corazón funcional. Ellos viven dentro de la aplicación. Por ejemplo, el concepto de “alumno”, representado como un registro de datos.

■Los objetos sintácticos: son conceptos, datos o funciones que la interfaz necesita para procesar alguna actividad interna a ella. No es propia al dominio semántico. Por ejemplo, el concepto de “número de visitas de una página”.

■Los objetos de interacción: son los elementos que conviven dentro de la interfaz del usuario y que son reflejados al usuario. Determinan la representación de los objetos semánticos y sintácticos dentro del espacio de la pantalla. Por ejemplo, la imagen de una “ficha” de datos para representar el concepto de “alumno”.

Entonces, lo que indica este principio, es que los objetos reales deben ser fielmente modelados por los objetos semánticos y que éstos, más los sintácticos deben ser coherentemente reflejados ante el usuario, mediante los objetos de interacción.

Los diseñadores deben lograr que el usuario vea en el espacio de la pantalla, una simulación de su mundo real o, de la representación mental que él posea de ese mundo, y el diseño icónico presenta las cualidades esenciales para lograrlo.

Se puede clarificar a la correspondencia directa mediante el siguiente gráfico ilustrativo:

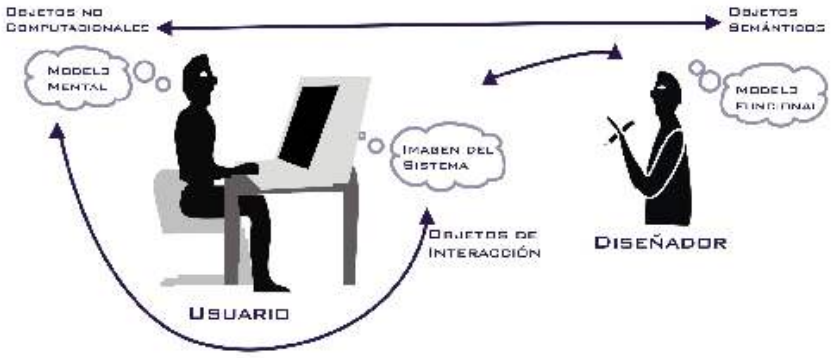


Figura 4.6: Correspondencia directa

El diseñador debe definir una interfaz del usuario que compatibilice los modelos del usuario, con el funcional y con la imagen del sistema, permitiendo que las expectativas del usuario, sobre una buena representación propuesta por el sistema, sea satisfecha.

El diseño icónico justamente parte de trabajar íntimamente con la representación mental del usuario, cómo él ve y maneja los conceptos y funciones de su problema real, y tiene por objetivo, aplicar ese conocimiento del usuario en pos de proveer la representación más significativa dentro de la interfaz.

Por lo tanto, dentro de la filosofía icónica se incluye y se aplica el concepto de correspondencia directa analizada.

El Diseño Icónico y la Consistencia

La consistencia es una de los objetivos más importantes que la interfaz del usuario debe respetar.

La misma estipula que la interfaz del usuario debe seguir ciertos patrones de conducta, de interacción y de visualización homogéneos, coherentes en todas las partes del sistema, donde se reduzca la posibilidad de ambigüedades y falsas interpretaciones.

Cuando se analiza el diseño icónico y se profundiza sobre la definición más compleja de ícono, especificada por Shi Kuo Chang, se describe al ícono como una dupla (parte lógica, parte física) donde el autor, además denota una relación unívoca entre la imagen y el significado. Esto significa que ambas partes deben tener una correspondencia uno a uno, o sea una imagen está asociada a un único concepto específico y viceversa.

En base a esta descripción, no sería admisible en un diseño icónico, tener dos íconos con igual imagen pero diferente significado, o por el contrario, representar un mismo concepto mediante dos imágenes distintas.

De esta manera, se puede entrever que las interfaces icónicas tienden a proveer un mayor grado de consistencia que cualquier otro tipo de interfaz, debido a que está incorporada desde su concepción.

III Ventajas de una Interfaz Icónica

El diseño icónico, se caracteriza por promover el uso de imágenes y otros recursos visuales con un alto nivel semántico que finalmente, derivan en el concepto general de ícono.

Es lógico pensar, que mediante estas representaciones visuales significativas y mecanismos de interacción cuidadosamente diseñados, el usuario dialoga con una interfaz en donde el simbolismo, el alto

grado de representatividad y familiaridad, la comparación inmediata, el contraste entre imágenes fácilmente identificables, son aspectos muy positivos que vienen incorporados y ya cultivados en ella.

Debido a esto, y sumado a la consistencia que se impone entre la imagen y el significado de los mismos, se puede afirmar que, desde el punto de vista de factores humanos, este tipo de interfaces corren con más ventajas que las interfaces visuales tradicionales.

Entre las ventajas más importantes, se tienen:

Aspectos positivos del Diseño Icónico	Explicación
Reconocimiento Inmediato	■Los íconos, gracias a su fuerte conexión con su significado, reduce la distancia semántica entre los objetos modelados y los objetos reales, permitiendo un fácil reconocimiento de los mismos. ■Permite una identificación instantánea de las entidades o funcionalidades modeladas a través de sus imágenes, captando en forma automática, el concepto asociado.
Simplicidad de Ejecución	■En estas interfaces se reduce el esfuerzo mental del usuario para llevar a cabo sus acciones. ■Mediante un diseño icónico de los gestos, manipulación directa, movimientos de íconos se logra proyectar las acciones reales del usuario en la computadora. ■Así, el usuario puede trasladar los pensamientos o tareas en mente a los requerimientos físicos del sistema, a muy bajo costo.
Reducción del uso de la memoria del usuario	■El ser humano es mejor reconociendo que memorizando. Y, en este tipo de interfaces, se aprovecha al máximo el impacto visual de comunicación de los íconos. ■Los usuarios trabajan directamente con objetos familiares en vez de tener que memorizar el significado de ciertas expresiones textuales o de visualizaciones insignificantes.
Alto nivel de Expresividad	■Es muy conocido el dicho que “una imagen vale más que mil palabras”, y esto se magnifica aun más cuando la imagen es icónica, o sea tiene un significado subyacente. ■El ícono expresa su significado subyacente con el nivel más alto y único de detalle y de abstracción, que es su propia imagen.
Mayor poder de Síntesis	■Los íconos condensan espacio e información porque expresan ideas, significados a través de sus imágenes significativas. ■En una representación textual, existe el problema de analizar el caudal de texto a utilizar. Si se emplean pocas palabras puede llevar a ambigüedad, si se emplean muchas, puede provocar cansancio y falta de interés. ■En representaciones visuales no icónicas, se debe recurrir a otros medios para poder explicar qué se quiso expresar en ellas. ■En cambio, los íconos con sus imágenes más elaboradas y detalladas, optimizan el uso del espacio de la pantalla.
Transmite más información que la original	■El caudal de información que se obtiene de una expresión icónica es aún mayor a la explícitamente visualizada. ■Permite a los usuarios extraer más información por observación directa, comparación y contraste. ■El texto es incremental, mientras mayor información se tenga que expresar, mayor cantidad de palabras se debe agregar.
Entendimiento Universal	■El diseño icónico provee un medio de comunicación entendido universalmente. Trascienden las fronteras, los hábitos y costumbres. ■Se evitan los problemas de traducción y de adecuaciones para cada región e idioma.
Mayor Confiabilidad	■Cuando el usuario trabaja con representaciones difíciles de distinguir y de comprender, se encuentra en una situación de total inseguridad e insatisfacción y lo lleva a cometer errores de interpretación. ■En las interfaces icónicas, el usuario actúa y maneja representaciones fieles a su realidad, que deben ser consistentes en todo el sistema.

Mayor Flexibilidad	■El usuario es quién controla la interacción con el sistema, mediante acciones asincrónicas de los íconos representados. ■Para llevar a cabo una tarea, los usuarios tienen la libertad de seleccionar, manipular, probar, combinar los íconos que crea apropiados en forma directa, sin referencias implícitas y sin tener que respetar una sintaxis compleja.
Mejora aspectos de Transparencia	■La transparencia es una condición de la interfaz del usuario, la cuál debe reflejar claramente sobre la pantalla, los estados del sistema, los procesos que están en curso, con sus etapas de inicio, ejecución y finalización. ■Como son características complejas que atañen al usuario considerablemente, se requiere la utilización de animaciones, símbolos e íconos que sean fielmente representativos, como se estipula en un diseño icónico.
Reducción de la tasa de errores	■Al minimizar la entrada de datos, los errores de tipeo y de sintaxis desaparecen. ■En un marco icónico, se trabaja con una validación y corrección a nivel espacial o visual, donde el usuario no percibe el concepto de error como en las interfaces tradicionales. ■Aquí, indicar que un ícono fue movido a un lugar incorrecto o que una combinación de íconos no está permitida, se refleja mediante la inhabilitación de dichos íconos o a través de una animación que deshaga el movimiento realizado.

Tabla 4.1: Ventajas del Diseño Icónico

IV Complejidad del Diseño Icónico

Intentar reflejar el mundo real del usuario o su representación mental del problema sobre el espacio de la pantalla, no es una tarea sencilla.

El alto grado alto de percepción y de expresión que pueden proveer las interfaces icónicas, implica un importante trabajo de diseño para obtener el mejor aprovechamiento de esas cualidades.

Debido a las restricciones y condicionamientos impuestos dentro de la definición de diseño icónico sobre las representaciones visuales a utilizar y la manipulación a proveer, es lógico pensar sobre las dificultades que un diseñador puede atravesar en el momento de encarar el diseño de una interfaz icónica.

Fundamentalmente, el costo mayor reside en la etapa de diseño. Mientras más tiempo, cautela y dedicación se destina a esta etapa, mayor rendimiento a nivel de factores humanos, se puede obtener de la misma.

Con respecto al desarrollo, se puede aclarar que hay una falta de herramientas específicas para la construcción de interfaces icónicas, por lo que cuestiones de representatividad, fidelidad, construcción de simulaciones, mantenimiento de consistencia, deben estar totalmente a cargo del desarrollador de la misma.

A continuación, se listarán las principales complejidades a nivel de diseño que se le atribuyen a las interfaces icónicas:

Complejidades del Diseño Icónico	Explicación
Dificultad en la Visualización de ciertas entidades	■Hay entidades y conceptos en donde se hace difícil encontrar la mejor imagen que represente a los mismos. ■En estos casos, la parte lógica ya está determinada, pero se complica hallar la mejor parte física asociada, puesto que no tienen una imagen inherente, ni siquiera extraída desde el mundo real del usuario. ■Las situaciones posibles donde esto puede ocurrir, pueden ser: -Cuando las entidades o los conceptos a representar tienen un alto grado de abstracción o generalización. Por ejemplo, el concepto de globalización. -Cuando el concepto a expresar es complejo, y requeriría una construcción de imagen más elaborada como una animación o simulación.

	-Cuando el concepto a expresar está compuesto por una secuencia de otras definiciones. Por ejemplo, representar icónicamente el concepto de reenviar en un correo electrónico, que implica la suma de los procesos de recibir y de enviar.
Costos en proveer consistencia teniendo en cuenta el contexto	■Es costoso independizar la relación imagen-significado del contexto en el cual el ícono se encuentra. ■Por ejemplo, uno puede asociar al concepto de borrar con la imagen de una goma. De esta manera, en un sistema de alumnos, este ícono debería estar presente tanto en un contexto de administración de alumnos como en un contexto de carga de notas. Pero, es muy distinto borrar un alumno que una nota del mismo. ■En ciertos casos es preferible comprometer el grado de consistencia, mostrando diferentes imágenes en determinados contextos.
Costos en la Integración con otros sistemas	■En el caso que el sistema icónico deba interactuar con otros sistemas de software, sean o no icónicos, se complica la uniformidad en la representatividad de los conceptos que debe prevalecerse en la filosofía icónica, arriesgando el nivel de consistencia.
Procesos extras de validación e interpretación espacial	■En este tipo de interfaces, es necesario proveer y controlar los procesos de validación e interpretación de los movimientos y manipulaciones de los íconos, como de otros recursos icónicos.Por lo tanto, se dificulta la forma de manejar y controlar el diálogo.
Incremento de los niveles de conversión	■Por lo general, la interfaz icónica provee un lenguaje de comunicación intermedio entre el usuario y el resto del sistema. ■Mientras más se acerca al mundo del usuario, más se aleja de las especificaciones internas, por lo que se complica el proceso de traducción de las acciones del usuario, a procedimientos internos del sistema, aumentando los niveles de conversión.

Tabla 4.2: Complejidades del Diseño Icónico

Importante

■A pesar de las dificultades atribuidas a las interfaces icónicas, es importante aclarar que en HCI, se debe apuntar a lograr la mayor eficiencia comunicacional. Esta eficiencia debe ser medida desde la percepción del usuario y no desde la complejidad de su desarrollo.

V - Especificación Formal de una Interfaz Icónica

En esta sección, se analizará a la interfaz icónica desde un punto de vista formal. Esta formalización permitirá conocer aún más la filosofía icónica y puede ser utilizada como un medio de especificación y descripción de la interfaz, y ser desarrollada en tiempos previos a su diseño. Formalmente, se define al ícono como un objeto con una representación dual compuesta de una parte lógica y una parte física. La parte lógica describe el significado del objeto mientras que su parte física denota su imagen. Por lo tanto, a un ícono se lo puede describir como el par ordenado (xl,xf), donde xl referencia la parte lógica y xf la física.

Cuando en una interfaz se utiliza como medio de expresión y representación al ícono, la misma se convierte en un “sistema icónico”. Este sistema inicialmente contará con un conjunto de íconos estructurados denominados “íconos primitivos”. Estos íconos presentes en la interfaz, pueden ser manipulados o combinados de alguna manera, que traiga como consecuencia la generación de nuevos íconos que intervendrán en el sistema en forma transitoria o permanente.

Por lo tanto, el cardinal del conjunto de íconos presentes en una interfaz puede ser incrementado mediante la creación de nuevos íconos, que surgen de alguna composición o combinación de los íconos miembros de dicho conjunto. Estos íconos generados de otros ya existentes se denominan “íconos complejos” pues expresan un concepto visual más elaborado.

Las combinaciones y relaciones entre íconos con el propósito de generar otros, están regidas mediante determinadas reglas. En dichas reglas intervienen ciertos operadores que, por ser aplicados o por operar sobre los íconos, son llamados “operadores icónicos”.

En definitiva, a un sistema icónico formalmente, se lo puede definir como un sistema compuesto de un conjunto de íconos primitivos o conjunto generador, más un conjunto de operadores icónicos.

A un sistema icónico descrito de esta manera, se lo puede describir o especificar formalmente, a través de una gramática. Pero, ¿cómo estará compuesta esta gramática?. Principalmente, al sistema icónico se lo puede describir como una quíntupla G formada por las siguientes componentes:

- Un conjunto de objetos lógicos o conceptos, denominado VL;
- Un conjunto de objetos físicos o imágenes, denominado VF;
- Un conjunto finito y no vacío de nombres de íconos, denominado S;
- Un elemento de S inicial, que denota el nombre del ícono axioma, o sea el ícono principal a describir, denominado x0;
- Un conjunto de reglas icónicas, denominado R. Las mismas permiten la correspondencia entre las imágenes y los conceptos. Por lo tanto, es una función que va desde S a LxVF, donde L es el conjunto de partes de VL U S.

Entonces, se puede decir que un sistema icónico es una quíntupla G=(VL, VF, S, x0, R).

Ahora bien, ¿qué se puede decir de los íconos primitivos y de los complejos, cuando se analiza el sistema icónico de esta manera?.

Lo que se puede observar que, los íconos primitivos se caracterizan porque su parte lógica xl está incluido totalmente en el conjunto VL y su intersección con S es vacío. Tienen la forma ({rótulo},imagen), donde los rótulos pueden denotar nombres de objetos, procedimientos u operadores de acuerdo a lo que el ícono represente. En cambio, los íconos complejos se caracterizan porque la intersección de xl con S ya no es vacío, esto significa que el ícono complejo apunta a otros íconos y define relaciones icónicas. El formato de este tipo de íconos es ({OP,y1,y2,..,yn},imagen), donde OP es un operador icónico que opera sobre los subíconos y1,y2,..,yn para crear un nuevo ícono.

Para los íconos complejos, se necesita profundizar un poco más sobre cómo se crean, cómo se aplican determinados operadores a sus subíconos. Para ese fin, se va a explicar los principales operadores icónicos, que dependerán del tipo de manipulación que se desee proveer.

Los operadores icónicos operan sobre los íconos y modifican tanto la parte lógica (su significado), como la parte física (la imagen) del ícono resultante, en forma simultánea.

El concepto de dualidad del ícono, es esencial y la aplicación de los operadores la sigue preservando. Una característica importante de los íconos es que su parte lógica y física son mutuamente dependientes, por lo tanto la modificación de una de ellas afectará la otra y viceversa.

Entre los operadores icónicos, se encuentran:

Operadores Icónicos	Explicación
Operadores MAT y DMA	■El operador MAT, se denomina de materialización y devuelve la imagen de un ícono. ■El operador DMA, se denomina de desmaterialización y devuelve la parte lógica de un ícono.
Operador SUP	■Se aplica a dos subíconos argumento y da como resultado, un ícono cuya parte física es la superposición de las imágenes de los íconos argumentos y la parte lógica es una mezcla conceptual de los significados de los mismos.
Operador CON	■Es un operador n-ario y determina una disposición de los íconos argumentos en donde los mismos se deben tocar o rozar. ■Su parte lógica es una combinación conceptual a definir.
Operador VER y HOR	■El operador VER es un operador binario que da como resultado un ícono cuya parte física será las imágenes de sus argumentos combinadas verticalmente. ■El operador HOR, en cambio da como resultado un ícono cuya parte física será las imágenes de sus argumentos combinadas horizontalmente. ■En ambos casos, una mezcla conceptual de los significados de sus argumentos determinará la parte lógica del ícono resultante.
Operador CXT	■Este operador realiza una interpretación contextual. ■El nuevo significado dependerá de ambos íconos argumentos mientras que la parte física sólo coincidirá con la del primero. ■La interpretación conceptual puede introducir, obviamente, una gran cantidad de atributos adicionales al ícono resultante. ■Este operador es muy útil en sentencias icónicas cuya interpretación varía con el tiempo.

Otros operadores	■De acuerdo a la manipulación que se le permite a los íconos, se debe especificar el tipo de operador. Pueden existir operadores para la selección, arrastre, cliqueo, doble cliqueo, cliqueo contextual, entre otros.

Tabla 4.3: Operadores icónicos

Ejemplo de la Especificación formal de un Diseño Icónico

Para poder entender aun más la especificación formal de un sistema icónico, vamos a trabajar con un ejemplo concreto.

A continuación, se va a describir parcialmente, al sistema CLAIV. Este sistema, realizado en un marco investigativo, provee un lenguaje icónico para consultar bases de datos relacionales.

Los usuarios, sin la necesidad de tener conocimientos previos del Álgebra relacional, podrán a través de CLAIV, realizar cualquier tipo de consulta. El mismo para efectuarla deberá seleccionar, mover íconos que representan las tablas y los operadores relacionales, y la misma interfaz, realizará las validaciones y las conversiones necesarias para trasladar dicha consulta a una expresión SQL correcta.

En la Figura 4.7, se ilustra una pantalla de CLAIV, donde el usuario realizó una consulta para averiguar los nombres y apellidos de los pilotos que son empleados de una determinada aerolíneas.

Vale aclarar que las partes físicas de los íconos eran realizadas por los mismos usuarios en un proceso de configuración, donde además se le proveía de una librería gráfica. En cambio, las partes lógicas, eran en estos casos, los atributos de las tablas iniciales.

El usuario, en su proceso de realización de la consulta, tiene que llevar al área central, las tablas y los operadores necesarios. En el caso de las proyecciones y condiciones a los atributos, el usuario debió utilizar los mecanismos de doble cliqueo y cliqueo contextual respectivamente.

Una vez que el usuario confeccionaba la consulta, la misma podía ser ejecutada o guardada. En este caso, como la consultas de tablas generan nuevas tablas, esto implicaba la creación de un ícono complejo, cuya imagen se debía definir y su significado era la especificación de dicha consulta.

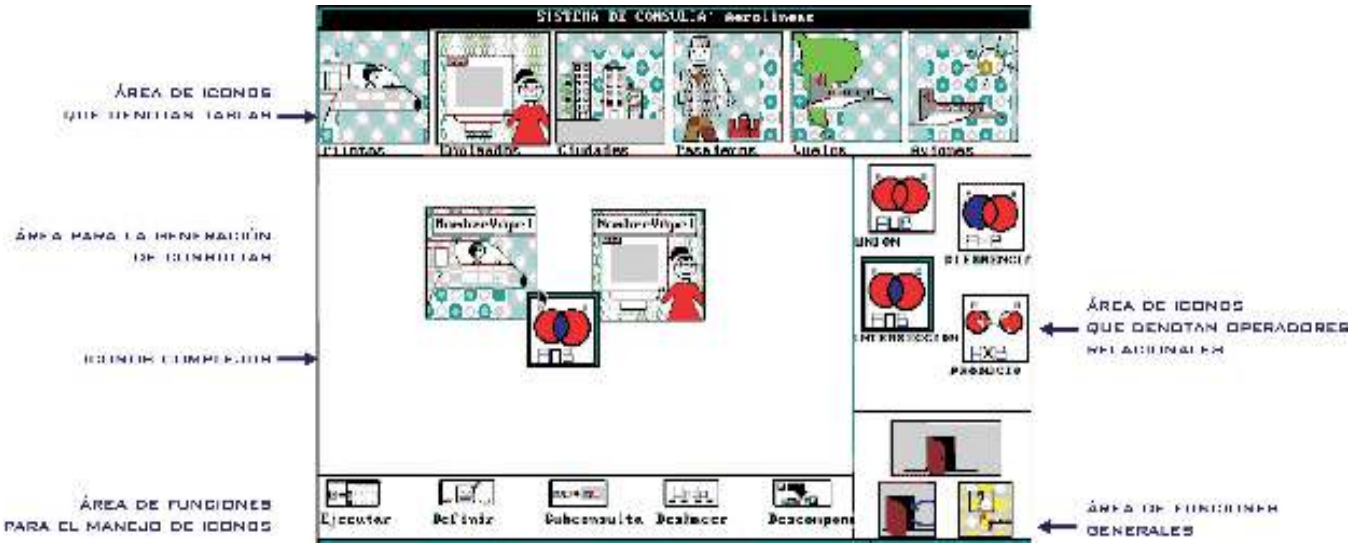


Figura 4.7: Ejemplo de una consulta icónica realizada en CLAIV

Veamos, entonces, a grandes rasgos cómo sería la especificación formal de CLAIV, definida a través de la gramática G= (VL, VF, S, x0, R).

■VL, es el conjunto de partes lógicas, o sea el esquema de las tablas relacionales, además de los íconos operadores.

■VF, es el conjunto de partes físicas, o sea las imágenes de los íconos elaboradas por el mismo usuario del sistema.

❖S, es el conjunto de nombres de íconos, que en este caso serían los nombres de cada tabla provista más los nombres posibles a las consultas específicas que se pueden hacer como ser, <consulta-unión>, <consulta-intersección>, <consulta-proyección>, etc.

■X0, es el axioma, que en este caso sería una consulta general en CLAIV

■R, sería el siguiente conjunto de reglas:

<consulta-CLAIV>::= <tabla> / <consulta>;
<tabla>::= <Pilotos> / <Empleados> / <Vuelos> / <Ciudades> ....;
<consulta>::= <consulta-INTERSECCION> / <consulta-UNION> / <consulta-PROYECCION> /
<consulta-WHERE> .....;
<consulta -INTERSECCION>::= CON(<consulta-CLAIV>, <icono-INTERSECCION>, <consulta-CLAIV>);
<consulta -UNION>::= CON(<consulta-CLAIV>, <icono-UNION>, <consulta-CLAIV>);
...
<tabla>::= <icono-PILOTOS> / <icono-EMPLEADOS> /<icono-VUELOS>.....;

<icono-PILOTOS>::= (, “esquema de PILOTOS”);

<icono-EMPLEADOS>::= (, “esquema de EMPLEADOS”);

...

<icono-INTERSECCION>::= (, “INTERSECT”);

<icono-UNION>::= (, “UNION”);

Aquí, se puede visualizar ciertas partes de la interfaz icónica, cómo se van describiendo las diferentes componentes de la misma, y las manipulaciones que se pueden hacer entre los íconos participantes a través de los operadores icónicos necesarios.

Caso de Estudio 4 - Interfaces Icónicas

El caso de estudio consiste en el análisis de la interfaz icónica un de prototipo de sistema para la gestión comercial de una galería de arte a través de la Web.

El prototipo fue desarrollado por el estudiantes Julián Grigera durante la cursada del año 2005.

Objetivo

El objetivo de este caso de estudio es analizar una interfaz icónica a partir del estudio de sus componentes de visualización e interacción abordadas en este capítulo. En particular, los íconos utilizados para representar el dominio de la aplicación, los conceptos que se manejan a nivel de interfaz así como también el diálogo asincrónico que da lugar a este tipo de interfaz.

Características del Caso de estudio

El objetivo de este sitio web es la venta de cuadros por Internet. Cuenta con dos secciones: la galería y el taller. En la galería el usuario podrá visualizar las obras expuestas, comprar láminas o comprar un cuadro. En la sección del taller podrá enmarcar una lámina entre las disponibles en la galería o incorporar una nueva. En este contexto también podrá efectivizar una compra.

El sistema provee un mecanismo de registración para la gestión interna de la galería y el taller.

En la figura 4.8 se presenta la página principal.

Figura 4.8: Página principal del sistema icónico para la Galería de Arte

Desarrollo del Caso de Estudio

¿Cómo evidencia el diseño icónico?

El diseño icónico se evidencia desde la página inicial, donde se presentan las dos alternativas a seguir dentro del sistema: la galería de arte y el taller. La visualización icónica de estos dos conceptos es la siguiente:

Figura 4.9: Iconos para representar las secciones principales del sistema

Cuando el usuario mueve el mouse sobre las imágenes de galería y taller se obtiene como feedback la iluminación de cada opción. Este mecanismo de feedback, que contribuye a la transparencia de la interfaz, es utilizado a lo largo de todo el sistema para destacar la opción a seleccionar. En la siguiente figura se muestra como se ilumina la galería, si es que el mouse se encuentra sobre ella, o sobre el taller.

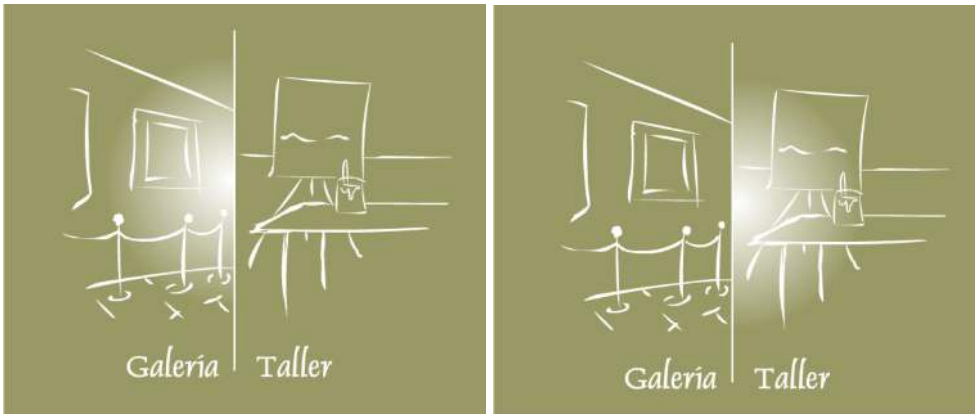


Figura 4.10: Ejemplo de transparencia de la interfaz a través de la iluminación de la sección seleccionada.

Ingresando a la galería, la página que se despliega es la siguiente:

Figura 4.11: Página de la sección Galería

En el centro de la pantalla aparecen las láminas disponibles, en tamaño miniatura. En la parte superior izquierda se mantuvo el icono de la galería. De esta forma el usuario no tiene dudas acerca de la sección del sistema donde se encuentra, mejorando la confiabilidad en el mismo. La barra de herramientas de la derecha presenta las operaciones que el usuario puede realizar en este contexto. Los íconos que representan la funcionalidad disponible se presentan a continuación:

Enviar una imagen al taller para seleccionar un marco adecuado.

Incorporar una nueva imagen a la galería.

Comprar la lámina.

Salir de la galería.

La ventana del taller es la siguiente:

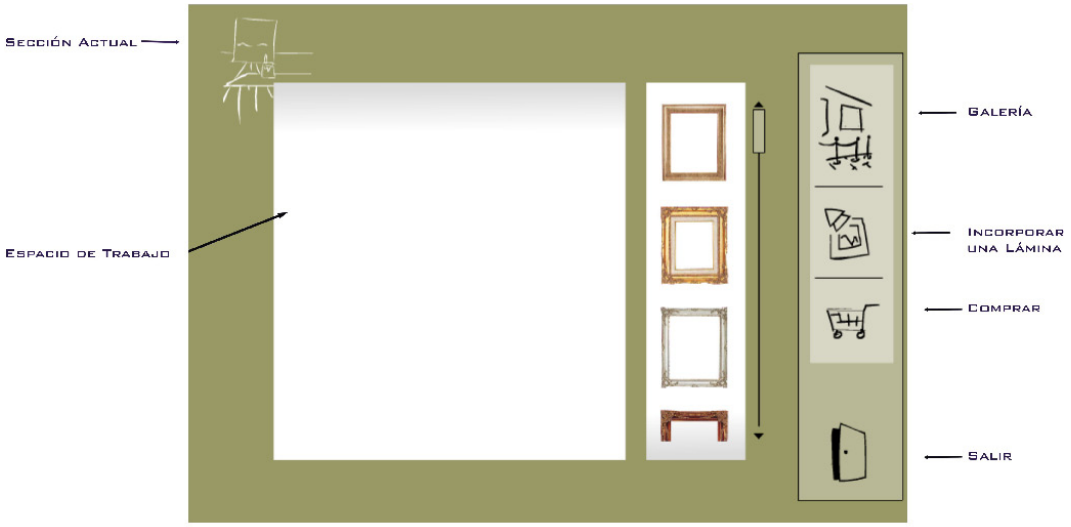


Figura 4.12: Página de la sección Taller

La distribución de los objetos de interacción en la pantalla y el diseño de los íconos utilizados es coherente con la visualización de la galería, garantizando la consistencia en todo el sistema. En la parte central se encuentra el espacio de trabajo del usuario, donde podrá componer sus propios cuadros y realizar compras. Una nueva barra lateral se visualiza para presentar los distintos marcos disponibles.

Figura 4.13: Marcos a seleccionar a través de drag and drop

Como mencionamos previamente, desde la galería y el taller el usuario puede realizar compras. La ventana de compras, que se accede a través del carrito, es la siguiente:

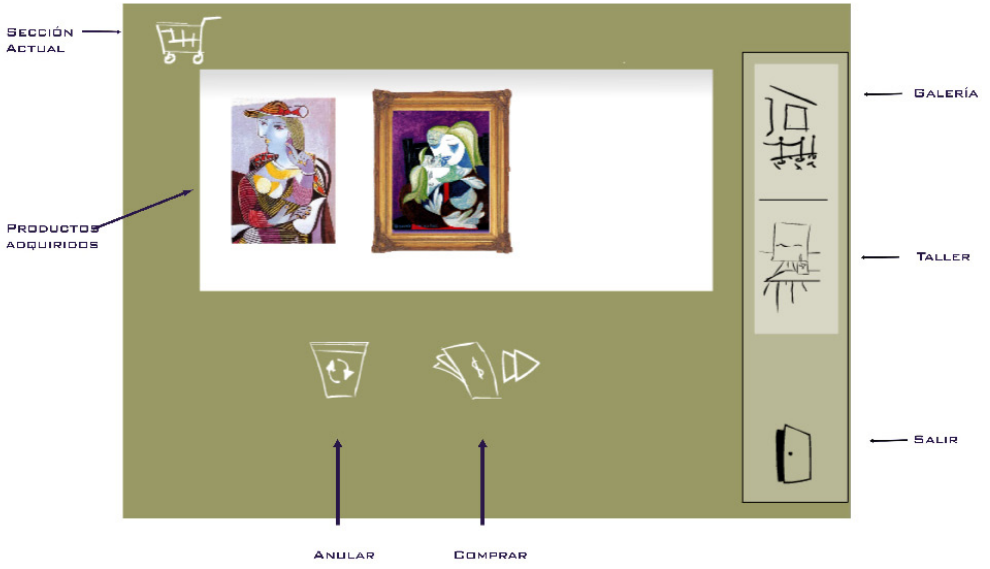


Figura 4.14: Página para realizar compras

Los íconos de la galería, el taller y la salida del sistema se mantienen en esta sección, en forma consistente. En este contexto, el centro de la pantalla visualiza los productos adquiridos por el usuario y aparecen nuevos íconos para anular la compra o efectivizarla.

¿Que mecanismos de interacción posee?

El mecanismo de interacción en todas las páginas es el sugerido para este tipo de interfaz: selección de íconos, feedback semántico y menus íconicos. De esta manera los usuarios utilizan el sistema en forma natural y simple. Para realizar una operación, el usuario siempre tendrá disponible la manipulación directa de los objetos de la interfaz por medio de la técnica de drag and drop. Por ejemplo, en el contexto de la galería, las láminas se seleccionan y se arrastran hacia el taller para elegir un marco. O se arrastran hacia el carrito para registrar una compra.

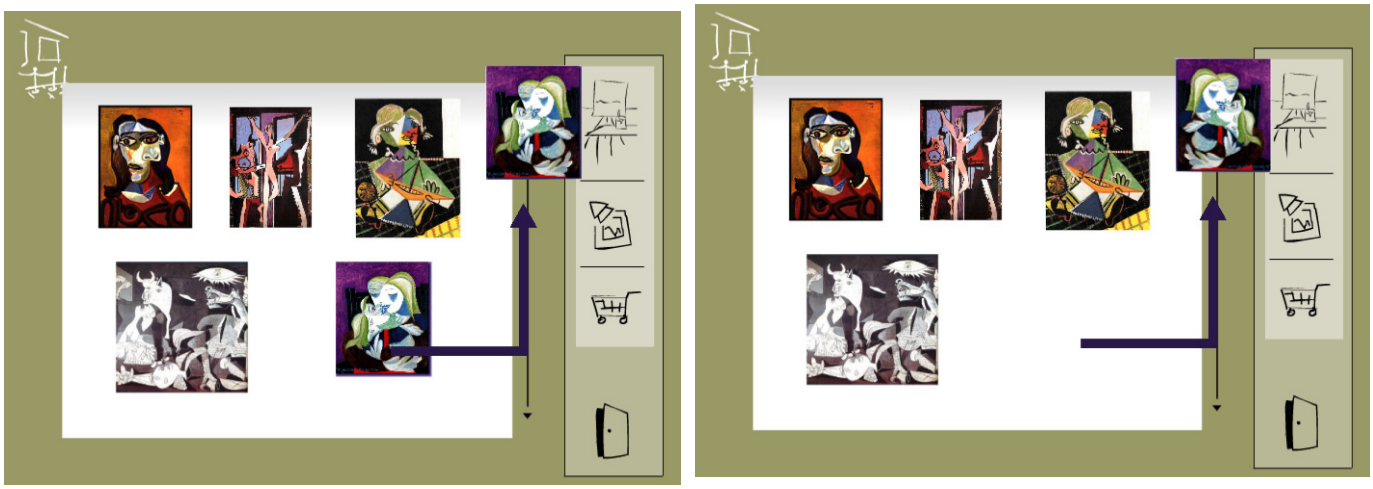


Figura 4.15: Ejemplo de manipulación a través de drag and drop para seleccionar una lámina.

En el contexto del taller el usuario elige el marco de preferencia y lo arrastra hacia la lámina elegida. Mediante esta interacción, se muestra el diseño final del cuadro, con la lámina y el marco seleccionados por el usuario.

Figura 4.16: Ejemplo de drag and drop para elegir un marco. A la izquierda se puede observar la lámina y a la derecha el cuadro terminado.

Otro mecanismo de interacción es el clic sobre el objeto, por ejemplo para cargar una nueva imagen o salir del sistema.

Ejercitación

El sistema sobre el que se realizará la práctica, fue diseñado por alumnos de la cátedra Diseño centrado en el Usuario, a partir de la siguiente especificación:

> *Se desea diseñar icónicamente un sistema para el servicio de una Clínica Veterinaria.*
> *La clínica cuenta con dos áreas: venta de productos y atención de animales. Esta última involucra dos salas: animales en observación y terapia intensiva. Cada sala cuenta con 5 camillas. Asociado a cada camilla tenemos la ficha del animal, donde figuran sus datos, el veterinario a cargo y el tratamiento a suministrar. El tratamiento especifica un registro de medicamentos, dieta, actividades y horarios.*
> *Diseñe un prototipo del sistema teniendo en cuenta, como mínimo la siguiente funcionalidad:*
> *Vista de las salas, camillas y estados de las mismas.*
> *Altas, Bajas de un animal con su correspondiente ficha.*
> *Reservas, ocupación y desocupación de las camillas.*
> *Manejo gestual de la ficha, donde deberá proveer funciones para altas, bajas, modificaciones y consultas de medicamentos (inyecciones, pastillas, jarabe, suero, pomadas, etc) con sus dosis, dieta, actividades (masajes, caminar, etc) y estudios pendientes (análisis, placas, etc.)*
> *Debe quedar*
> *registrado el nombre de la persona que brindo el tratamiento, fecha y horario.*

Considerando las figuras que se presentan a continuación, responder las siguientes preguntas:

1. ¿Cómo evidencia el diseño icónico? Identificar, marcando con un círculo, los elementos icónicos de la interfaz de la pantalla inicial del sistema para una Veterinaria.

2. ¿Qué mecanismo de interacción le parece adecuado? Proponer al menos tres actividades que podrá realizar el usuario final del sistema e indicar como las llevaría a cabo.

3. De acuerdo a sus respuestas dadas en el punto anterior, ¿qué características debería tener el usuario final para obtener el mayor beneficio posible del sistema?

4. De acuerdo a las recomendaciones de diseño vistas en este capítulo, ¿qué problemas encuentra? ¿cómo los solucionaría?

Figura 4.17

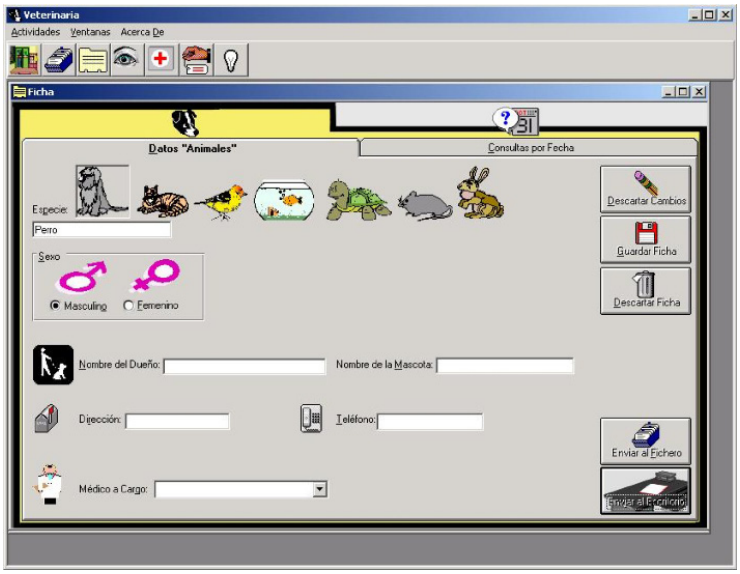


Figura 4.18

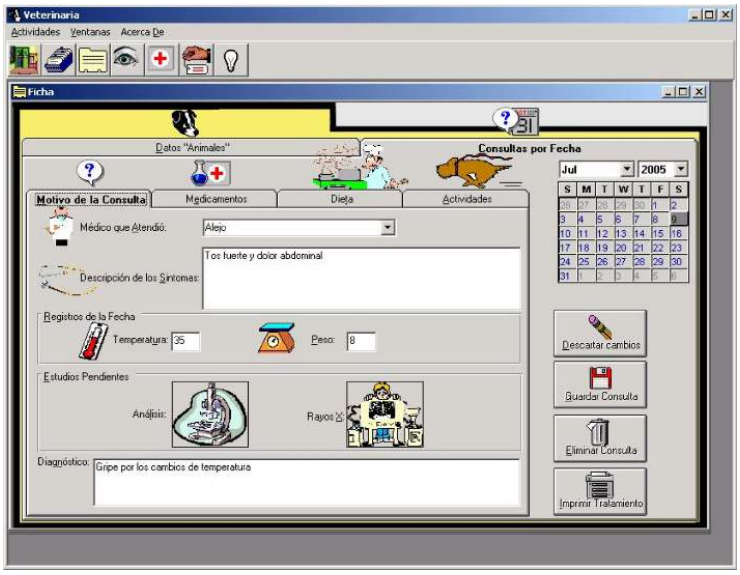


Figura 4.19

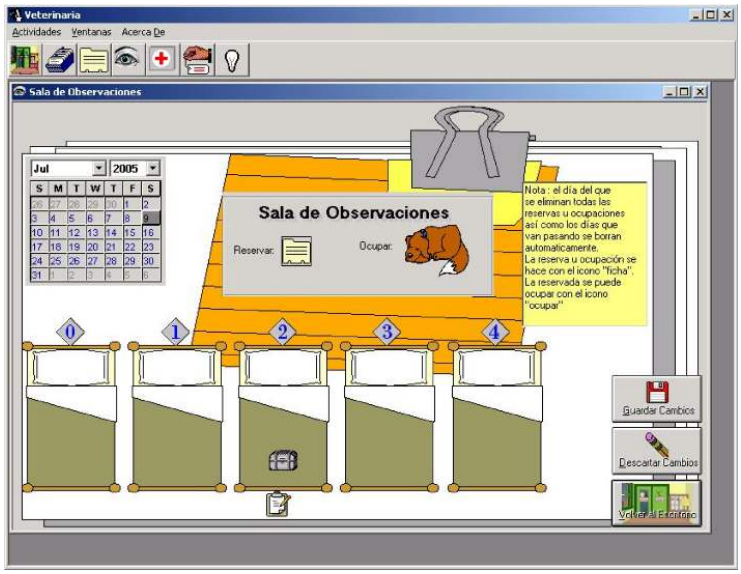


FIGURA 4.20

Referencias

- Jean Piaget (1988). Piaget en el aula. Cuadernos de Psicología Nro. 163.
- Jean Piaget (1977). La psicología de la inteligencia. De la Pedagogía. Editorial Paidós.
- Jerome S. Bruner (1984). Acción, pensamiento y lenguaje. Editorial Alianza.
- Foley, Jim; Dam, Van (1991). Fundamentals of Interactive Computer Graphics.
- Caroll J.M. Mack R.L. y Kellog, W.A. (1997). Interface Metaphors and User Interface Design. Human-Computer Interaction INTERACT ?87. North-Holland, Amsterdam.
- Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, MA.
- Hutchins, E.L, Hollan, J.D., and Norman, D. A. (1986) Direct Manipulation Interfaces, in D. A. Norman and S. W. Draper (eds), User-Centered System Design, Lawrence Erlbaum, 87124.
- Shi Kuo Chang (1985) Image Information Systems. Proceedings of the IEEE on Visual Communications Systems, Vol. 73.
- Shi Kuo Chang (1987). Visual Languages: A Tutorial and Survey. IEEE Software Magazine, Vol. 4.
- Shi Kuo Chang (1990).Principles of Visual Programming Systems, (ed.), Prentice-Hall.
- Shi Kuo Chang (1990).Visual Languages and Visual Programming. Plenum Publishing Corporation.
- Shi Kuo Chang (2001). Software Engineering and Knowledge Engineering. Columbia University Press.
- Kevin Mullet y Darrell Sano. (1995). Designing Visual Interfaces.Sun Microsystems Inc.

Capitulo V

La Interfaz Web y sus Principios de Diseño

05

La Interfaz Web y sus Principios de Diseño

La calidad de un sitio Web no está determinada por la complejidad, la cantidad de recursos y elementos de avanzada que se incluyan en ella, sino por los principios que se hayan respetado en su diseño. Aplicar las correctas normas de diseño es lo único que garantizará la usabilidad del sitio y el sobrevivir en un mundo de tanta competitividad.

Temas a Tratar

I- Introducción
II- Características Específicas de un Sitio Web
III- Particularidades en el Desarrollo Web
- Actividades previas al Diseño del Sitio
- La Ingeniería de Usabilidad

IV- Principios de Diseño del Sitio Web
- Principios de Diseño a Nivel de Sitio
  - La estructura del sitio Web
  - La navegación del sitio Web
  - Los hipervínculos
- Principios de Diseño a Nivel de Página
  - Diseño general de las páginas Web
  - La página inicial
  - Página con formularios
- Principios de Diseño a Nivel de Contenido
  - Diseño general del contenido
  - Lenguaje empleado

Casos de Estudio
Ejercitación
Referencias

I - Introducción

Hoy en día, desarrollar un sistema de software implica además desarrollar un conjunto de productos adicionales, tales como un manual del usuario interactivo, un instalador que sea amigable, soporte en línea, descargas automáticas de actualizaciones, servicios de FAQs o preguntas frecuentes, correo electrónico, entre otros. Herramientas que en forma integral y desde diferentes líneas, intentan simplificar tareas, asistir, guiar y comunicarse con los usuarios finales del sistema.

Sumado a esto y debido al crecimiento exponencial de usuarios de Internet, también es usual incluir el desarrollo de un sitio Web para complementar con más información y servicios, y para tener finalmente, presencia en Internet.

Independientemente si el sitio es un medio de comunicación complementario o no, deberá hacer el mejor aprovechamiento posible de las ventajas de comunicación que ofrece Internet, que comprende una multitudinaria comunidad de usuarios con características, costumbres, formaciones y culturas diferentes.

Sin embargo, el desarrollador Web deberá saber que su producto no será el único en Internet, sino que va a formar parte de un sin fin de páginas con servicios y contenidos similares, que se va a enfrentar a una gran competitividad, tan compleja que puede poner en riesgo la futura vida útil de su sitio.

Estos sitios pueden variar en su magnitud, su complejidad, sus servicios o en su aspecto. El usuario puede navegar desde páginas estáticas, hasta tener la posibilidad de llenar formularios, realizar transacciones a bases de datos, como también acceder a aplicaciones colaborativas como mensajería, foros, encuentros virtuales, cursos a distancia.

No importa si ofrece páginas simples o complejas, con objetos de interacción tradicionales o más complicados, pero lo que sí importa es el grado de permanencia en Internet y su poder de aprehensión, o sea su capacidad de poder capturar la atención del usuario de tal manera, que se convierta en un lector asiduo del sitio.

Pero ¿cómo se lograría esto?. El adornar las páginas con colores, animaciones e imágenes atrayentes, lamentablemente no es la clave para garantizarlo. No es una cuestión de centrar la atención exclusívamente en los efectos visuales. Lo que sí puede hacer la diferencia, es un “adecuado diseño”, que sea acorde al usuario conectado, al contexto, al contenido del sitio y a su aplicación subyacente.

Partiendo de sitios con servicios e información similares, cuestiones de diseño son determinantes para que el usuario pueda distinguir uno de otro, percibir su calidad trayendo como consecuencia que el usuario lo elija, se sienta cómodo y satisfecho con él. Sitios, cuyo diseño debe efectuarse bajo normas y recomendaciones específicas, que ayuden a lograr una integración coherente entre una estructura simple, un buen contenido, una representación visual efectiva, una interacción personalizada y una navegación intuitiva.

El diseño es lo que garantizará la usabilidad del sitio o calidad de uso, que es ni más ni menos, lograr la mejor productividad de todo el sistema, basándose en un alto grado de satisfacción por parte de los lectores del mismo.

II -Características específicas de un Sitio

El surgimiento de Internet, como tecnología de comunicación entre usuarios, abrió las puertas a nuevas posibilidades de interacción y cooperación. Aplicaciones de groupware y trabajo cooperativo como encuentros virtuales, comunicación cara a cara, educación a distancia, modificarán radicalmente los patrones de comunicación y afectarán completamente nuestra vida cotidiana.

Desde el punto de vista del ser humano, el usuario se adecuará a estas nuevas pautas de interacción, utilizando las aplicaciones para obtener comunicación en línea con otras personas ubicadas remotamente, en pos de compartir recursos, colaborar en actividades grupales y transmitir ideas.

Desde la perspectiva de la Ingeniería de Software, esto significa reformular todos los mecanismos y metodologías involucradas en el desarrollo de la interfaz del usuario. Partiendo desde técnicas de análisis, aspectos de diseño, recomendaciones, mecanismos de prototipación, implementación y testeos de usabilidad, que deben ser adecuados a estas nuevas potencialidades.

Las políticas de diseño que se deben emplear en la interfaz de un sitio Web, no coinciden con la de interfaces gráficas, ni multimediales, ni tampoco con los principios considerados al desarrollar un boletín o suplemento informativo desde un procesador de textos o herramienta para publicaciones.

Entonces, como las interfaces del usuario presentes en las páginas Web tienen características y particularidades propias, la misma debe ser construida basada en delineamientos particulares y mediante patrones de diseño que deberán ser reconsiderados específicamente para este caso.

A continuación, se analizarán cuáles son las cuestiones específicas que caracterizan a las interfaces de

sitios Web, que las distinguen del resto de las interfaces y que es fundamental que el diseñador las tenga en cuenta al momento de encarar la construcción de un sitio.

Respecto al Perfil del Usuario:

En primera medida, hay que tener en cuenta que los usuarios de Internet presentan características muy heterogéneas, pueden ser de cualquier nacionalidad, con diferencias socio-culturales muy grandes, pueden contar con distinto grado de formación y pueden acceder mediante distintos recursos tecnológicos.

Para esto, se va a analizar posteriormente cómo acotar el espacio de usuarios hacia una comunidad de lectores más específica, trabajando directamente con uno o más perfiles de usuarios a la que va estar dirigido el sitio, y realizando estudios y modelos descriptivos de dichos lectores destinatarios, apropiados para lograr una correcta interacción con ellos.

Lo que sí se puede determinar en términos generales, es cuáles son los aspectos comunes que cualquier usuario de Internet posee y cuál es su comportamiento habitual frente a un sitio.

Principalmente, el usuario pasa de leer toda una información ya establecida y pautada como en el caso de diarios, revistas y libros, a seleccionar información y a construirla basándose en movimientos, acciones, desplazamientos y cliqueos. Por ello, la información está expresada en forma temporal y se genera como parte de la interacción y movimientos del usuario.

El “look & feel”, o sea la “apariencia y percepción” de una interfaz para la Web difiere de las interfaces tradicionales, debido a que en este caso el “feel” domina completamente la experiencia del usuario. El usuario pasa de “leer” a “hacer”, generando emocionalmente un impacto más fuerte.

El usuario de Internet se comporta más impaciente en línea, y requiere motivación para el movimiento, exploración y búsqueda de la información.

Tiempos medios de lectura de carga, cantidad de cliqueos para acceder a la información requerida, tiempos de navegación, entre otros, son valores que en el diseño de los sitios Web toman mayor preponderancia. Por ejemplo, Google Analytics (http://www.google.com/analytics) que es un servicio gratuito de estadísticas de sitios Web, permite analizar un sitio desde determinadas métricas con el objeto de obtener una versión más optimizada del mismo. Para realizar el monitoreo, Google Analytics añade un JavaScript a cada una de las páginas que desea analizar y carga la información al servidor Google almacenándola en la cuenta del usuario.

Existen métricas como “time on page” y “time on site”, “tiempo en página” y “tiempo en sitio” para analizar el perfil del usuario, por ejemplo su lugar de origen o desde donde se consulta para obtener informacion mas representativa y útil. El “tiempo en página” permiten tener noción sobre promedios de duración que un lector se encuentra en una página antes de saltar a la siguiente. El “tiempo en sitio” en cambio, da una idea sobre el tiempo de visita o la permanencia del usuario dentro del sitio.

Respecto a Cuestiones de Diseño:

En este caso, hay que analizar específicamente las cuestiones de multimedia y de navegación para la Web, que no coinciden con el desarrollo de hipermedia.

Con respecto a la multimedia, ésta debe ser restringida. La necesidad de ser “modernos” en la Web, -en inglés referenciado con el término “to be cool”-, que implica tomar ventaja de las aspectos tecnológicos más recientes y novedosos, se ha creído por mucho tiempo ser el causa más importante para lograr la permanencia en Internet.

Sin embargo, numerosos estudios de usabilidad han comprobado que muchas veces utilizar nuevas tecnologías, tales como mapas de imágenes, animaciones, elementos Shockwave u otros efectos visuales atrayentes, complican considerablemente la interacción con el sitio, producen distracciones, molestias, demoras, y hasta son considerados difíciles de utilizar.

Además, como se verá en las próximas secciones, las reglas del buen diseño restringen las cuestiones tecnológicas ya que, por ejemplo, se recomienda minimizar el tiempo de carga y ancho de banda de tal forma que el acceder a la próxima página del sitio, no consuma más tiempo que el de dar vuelta la página de un diario o revista.

También, se recomienda tener consideraciones especiales en el contenido e ilustración. Se aconseja texto estructurado y resaltando sólo lo más importante, imágenes chicas, poca carga visual, diseño estético pero minimalista.

Por tal motivo, los diseñadores han sido forzados a utilizar nuevas tecnologías pero únicamente en forma alternativa o complementaria a un diseño más moderado y simple.

Con respecto a la navegación, su problema inherente de pérdida de contexto, se agrava aún más en un entorno de la Web, donde existen varias líneas de navegación. En un sitio, pueden estar disponibles la navegación por contenido, por estructura, navegación atrás y adelante, navegación adaptada y otras más, que se describirán posteriormente en este capítulo.

Debido a esto, hay que considerar que el modelo de navegación de la Web difiere del de las aplicaciones que no son para Internet. Esta diferencia causa confusión a los usuarios y los diseñadores necesitan trabajar específicamente con normas de diseño especiales para la navegación, en pos de ayudar a los usuarios a utilizar la aplicación Web satisfactoriamente.

Además, hay que tener en cuenta que, dentro del entorno de la Web, la navegación tiene una fuerte relación con la forma en que la misma es percibida por los usuarios. La navegación se ve afectada por la apariencia de los hipervínculos, por su ubicación, por la visualización de su entorno, por la información que acompaña la estructura de la navegación y por otros recursos, haciendo aún más complicado su diseño y tratamiento.

Respecto a los Objetivos:

Al desarrollar un sitio Web, el diseñador de la interfaz debe proponerse ciertos objetivos que son específicos para este entorno, además de los que se plantean a nivel de interfaz general.

Es importante pretender que la interfaz Web sea simple, amigable, flexible, transparente como se planteó en el Capítulo I, pero además hay que tener en cuenta ciertos propósitos especiales.

En el mundo tan competitivo de la Web, donde se plantea el concepto de "user engagement", que implica "captar al usuario" mediante un diseño eficiente de la presentación y contenido del sitio, el objetivo de usabilidad será crucial para la supervivencia de los sitios.

Las técnicas de Ingeniería de Usabilidad, permiten la creación de sitios Web más fáciles de usar, ayudando a los desarrolladores a focalizar su atención en el usuario más que en la tecnología.

Analizar las necesidades de los usuarios, sus objetivos, sus pretensiones respecto a las actividades que se desarrollarán dentro del sitio, se considera más importante que el adoptar los últimos aspectos de avanzada.

Desde ya que, si la última tecnología va acorde al perfil de los usuarios y ayuda a concretar sus intenciones frente al sitio, no habría problemas en ser incorporada, pero utilizarla por una cuestión de "estar a la moda", es actualmente, inaceptable.

Jacob Nielsen, precursor del concepto de Usabilidad, especifica como complemento a la usabilidad, otros objetivos importantes para el diseño Web que deberían ser considerados, como ser:

Objetivos propios al Diseño Web	Explicación
Legibilidad	■Hay que brindar información fácilmente comprensible, que sea accesible desde diferentes circunstancias.
Expresividad	■Debe existir autenticidad y veracidad en la información que se publica. Además, la escritura y la expresión debe ser correcta.
Localización	■El usuario debe encontrar fácilmente los objetivos del sitio, la funcionalidad, los servicios o la información, con pocos cliqueos. ■Considerar la Ley de Fitts [Paul Fitts, 1964] sobre el tiempo para alcanzar un objetivo con el mouse, que depende de la distancia y el tamaño del mismo.
Navegabilidad	■Se deben habilitar diferentes mecanismos de exploración y se deben flexibilizar e incentivar los movimientos a través del sitio.
Visibilidad	■Hay que diseñar el sitio en forma atractiva pero simple, con patrones de visualización previamente estudiados y testeados.
Interacción	■Se debe flexibilizar el diálogo entre el usuario y el sitio, haciéndolo adaptable según el perfil del lector. ■Hay que evitar molestar y distraer al usuario con elementos emergentes como pop-ups.
Progresividad	■El sitio debe estar estructurado de forma escalable, que permita su propio crecimiento y evolución. ■Hay que trabajar en la actualización de contenidos e informar sobre la antigüedad de la información que se publica.
Características de Internet	■Diseñar específicamente para Internet, con sus características de universalidad, seguridad, accesibilidad. ■Considerar cuestiones de indexación, con criterios para ser multireferenciados entre diferentes sitios. ■Hay que tener en cuenta estándares de intercambio, como ser en noticias, anuncios, foros, debates, carteleras. ■Considerar como punto de partida, la estandarización del diseño de las aplicaciones específicas de Internet, como ser de gobierno electrónico, comercio electrónico, educación a distancia, etc.

Presencia en Internet	■Si se escribe para la Web, es porque se quiere ser encontrado. Jacob Nielsen [Nielsen, 2006] dice que si el usuario no te encuentra primero en los resultados de un buscador Web, pensará que directamente no estás en la Web.

Tabla 5.1: Objetivos del Diseño Web

Importante

■En la construcción de la interfaz de un sitio Web, no son aplicables las cuestiones de diseño del desarrollo de las interfaces gráficas, de boletines informativos, ni de un documento hipermedial.
■La interfaz Web presenta sus características propias, por lo que debe ser tratada específicamente en todas sus etapas de desarrollo.

III - Particularidades en el Desarrollo Web

Es importante que los desarrolladores comprendan que un sitio Web es un software que presenta particularidades específicas, que cuenta con sus propios objetivos, con un caudal de usuarios muy importante y heterogéneo, y que presenta inherentemente ciertos problemas de diseño.

Estas características particulares, que deben ser consideradas a lo largo de todo el proceso de desarrollo de la interfaz del sitio, hacen que el mismo se vea afectado y traiga en consecuencia algunas modificaciones puntuales respecto del desarrollo tradicional de una interfaz del usuario.

Uno de los puntos distintivos del desarrollo Web, es sugerir a los diseñadores a ahondar aún más, en la etapa de requerimientos de la interfaz del sitio, complementando con ciertas actividades específicas que serán de gran utilidad para posteriormente encarar un correcto diseño del mismo.

Además de esto, el desarrollador Web, deberá tener en cuenta que existe una propuesta metodológica que apunta a la construcción de sitios “usables”, que él puede incorporar y aplicar como proceso de desarrollo. Este esquema metodológico es denominado Ingeniería de Usabilidad, fue creado por Jacob Nielsen y será introducido posteriormente.

Otro aspecto importante y característico que el diseñador no podrá saltear, es la investigación, el estudio y selección de las normas específicas para el diseño de una correcta interfaz para la Web.

Se tiene destinado para ello, la última sección de este capítulo, donde se listarán y explicarán los principios de diseño más importantes que deben ser considerados por los desarrolladores Web.

Actividades Previas al Diseño de Sitios Web

Para poder encarar eficientemente el desarrollo del sitio con todas sus dificultades y particularidades, es sumamente necesario tratar esta complejidad antes de comenzar con el proceso de diseño.

Para ello, se sugiere encarar una serie de actividades complementarias a las de la etapa de requerimientos o pre-diseño, de las interfaces del usuario tradicionales, en pos de comprender aún más la problemática inherente a la Web y poder paliarla adecuadamente.

Determinar la Comunidad Destinataria de Lectores:

En el sitio de la BBCMundo.com indica que actualmente Internet cuenta con más de miles de millones de usuarios de todo el mundo. Los mismos podrían ser los futuros lectores del sitio que se está desarrollando.

La masividad de usuarios, el alto grado de heterogeneidad y diversidad entre los mismos, no debe desalentar al diseñador y provocar pasar por alto, a la modelización de usuarios. Se pone el valor en el diseño centrado en el usuario pero trabajando principalmente en acotarlo a una comunidad más específica.

Hay que tener en cuenta que la mayoría de los sitios Web, son construidos para una determinada audiencia, a la que se le destina y se le dedica los servicios, la información y las funcionalidades provistas. Pueden ser alumnos, médicos, clientes, consumidores, amas de casa, adultos mayores, hinchas de un club de fútbol.

Además, hay que considerar que, cualquier medición de éxito de un sitio, se plantea en base a dicha audiencia, es decir número de visitas, permanencia en el sitio del visitante, número de revisitas, ventas, ranking, opiniones, quejas. Por lo tanto, es lógico pensar en profundizar la atención hacia esos usuarios destinatarios, y utilizar este análisis, en pos de ofrecer el mejor diseño y el más adecuado, para los mismos.

Contar con un entendimiento profundo sobre quiénes serán los usuarios destinos del sitio, cuáles son sus perfiles, qué es lo que ellos quieren hacer en el sitio y cómo ellos tienden a comportarse, no es necesariamente un proceso complicado, y más aún, en función de las ventajas que se pueden alcanzar al encararlo en forma sistemática y consistente. Este proceso fundamentalmente, consiste de los siguientes puntos:

Especificación de la Audiencia del sitio	Explicación
Formular y reconocer la audiencia de usuarios	■Debido a la masividad de personas que utilizan Internet hay que analizar y definir la audiencia a la que está dirigido el sitio, definiendo perfiles de usuario.
Analizar el comportamiento esperado de la audiencia	■Hay que determinar qué es lo que se pretende que los usuarios realicen dentro del sitio y en qué tiempos. Por ejemplo, si se quiere principalmente que los usuarios entren al sitio a investigar o bajar información, o para comunicarse con otros, o para informarse. ■Además, analizar si se pretende capturar usuarios temporádicos o permanentes.
Mantener contacto previo con la audiencia	■Es necesario saber que la audiencia debe ser detectada en pos de mantener un contacto directo con ella. Por ello, se deben planificar correctamente citas, encuentros sistemáticos, entrevistarlos, analizar su entorno y perspectiva. ■Si no se puede establecer una comunicación con los usuarios finales del sitio, se puede trabajar con muestras de usuarios, que presenten las mismas características que los patrones considerados en la audiencia.
Recolectar información del usuario	■Se debe verificar las hipótesis planteadas sobre los usuarios. ■Se debe comprender lo que ellos necesitan de la información, formas de pensamiento sobre organización y agrupamiento de la información, expectativas del sitio, entre otras cuestiones. ■Hay que estudiar principalmente cómo son, cuáles son sus niveles de conocimiento sobre la materia, su grado de experiencia con la Web y tipos de sitios similares. ■Al trabajar con usuarios, uno puede reunir escenarios reales y aprender qué es lo que hace que el sitio Web funcione bien o no.
Participación de los usuarios involucrados	■Permitir que los usuarios ayuden en la construcción del sitio. ■Hacerlos participar en decisiones, teniendo en cuenta sus opiniones, críticas y sugerencias. ■Explicitar la política de manejo de la información del usuario.

Tabla 5.2: Pasos en la especificación de usuarios destinatarios del sitio Web

Planificar el sitio web:

El reconocimiento y modelización de los lectores a los que se les destina el sitio, es fundamental para desarrollar un producto a la medida y necesidad de los mismos.

Pero además, hay que considerar de suma importancia, al proceso de planificación del sitio, que esencialmente consiste en que el desarrollador tenga en claro cuál será el sentido del sitio y sus objetivos principales. Permite determinar actividades de usabilidad, que son determinantes para el desarrollo de un sitio exitoso.

El desarrollador Web, previo al diseño del sitio, debería pensar principalmente en los siguientes cuestionamientos:

Planteos en la Planificación del Sitio	Explicación
Plantear los objetivos del sitio	■Es fundamental que los objetivos propios del sitio estén claramente especificados. ■La finalidad del sitio puede ser puramente comercial, educativa, financiera, informativa, de contacto de usuarios, para recolectar información, entre otros. ■Los diferentes roles que intervienen en el desarrollo del sitio, diseñadores, programadores, especialistas en usabilidad, especialistas en contenido, el líder del proyecto, los usuarios destinatarios, deben tener en claro “para qué” y “por qué”, el sitio se va a desarrollar.
Analizar concordancia entre los objetivos del sitio y de los usuarios a los que está dirigido	■Hay que plantear los objetivos principales del sitio, teniendo consideración sobre las expectativas y finalidades de los usuarios. ■Ellos accederán al sitio por algún motivo, por alguna necesidad que el sitio debe satisfacer en la forma más directa posible.

Analizar posibles contextos del usuario	■A causa de la heterogeneidad de posibles lectores del sitio, será necesario considerar diferentes escenarios de posible uso. ■Hay que considerar que los usuarios deberán poder acceder al sitio desde diferentes lugares, distintos medios y mediante diferentes recursos tecnológicos. ■El sitio deberá adecuarse y funcionar correctamente a estos diferentes contextos.
Realizar análisis competitivo	■Se debe analizar sitios similares que estén en el mercado, desde diferentes aspectos, su diseño, su contenido, cómo ofrece los servicios que presta. ■Actualizar la estrategia del sitio para evolucionar junto a los competidores. ■Discutir sobre formas y aspectos que optimicen los valores tradicionales, para ser incorporados en el futuro sitio Web.
Plantear medidas de éxito y fracaso	■Hay que fijar mediciones que permitan analizar cuándo se considera que el sitio es exitoso y cuándo no. Por ejemplo, puede que se determine que el éxito dependerá de la cantidad de registrados o asociados al sitio, más que del número de visitas del mismo.
Planificar actividades que promulguen la usabilidad	■El proceso de planificar el sitio para y con los usuarios, constituye de por sí, una actividad de usabilidad. ■Además de organizar futuros testeos de usabilidad, también hay que planear otras formas de involucrar a los usuarios dentro del proceso de desarrollo del sitio. ■Hay que determinar formas para obtener información de los usuarios previo a la etapa de diseño, como también recolectar, averiguar, probar formas de comunicación y de transmisión de contenido, adecuado para ellos. ■También, hay que planear cómo se van a desarrollar y suscitar los prototipos del sitio que se organizarán para los usuarios. ■Tener en cuenta que la Prototipación, será esencial para llevar a cabo estas actividades de usabilidad.

Tabla 5.3: Planteos en la Planificación del sitio Web.

Pensar en la Prototipación:

Es fundamental que los diseñadores Web, se preparen y se organicen para aplicar en forma efectiva, la Prototipación durante el proceso de diseño del sitio.

La Prototipación es fundamental para trabajar con todo lo que se ha planificado del sitio, constituye un medio poderoso de representación y de comunicación de las distintas alternativas de diseño y, establece un marco adecuado para la puesta en marcha de ideas y toma de decisiones.

Además, a través de los prototipos, que son versiones parciales del software que se está desarrollando, se puede obtener de antemano la comprobación y corroboración de cuestiones de usabilidad.

Ellos son de gran utilidad para la organización y realización de evaluaciones, donde se puede obtener parámetros sobre la performance de los usuarios y su grado de satisfacción, previamente a invertir el tiempo necesario, esfuerzos y los costos, para la construcción final del sitio.

Partiendo de las definiciones que se detallaron en el Capítulo II, y teniendo en cuenta el contexto de la Web, un prototipo será entonces una versión incompleta del sitio, donde reflejará parcialmente ciertos contenidos y servicios.

Por lo general, se inicia con un prototipo de la página principal y alguna de sus páginas secundarias, para que el usuario vaya observando y percibiendo el diseño general del sitio. Luego, se van confeccionando versiones prototípicas más avanzadas y completas, donde la estructura de la navegación y cuestiones de interacción pueden ser probados frente a los usuarios.

Importante

■La Prototipación es una metodología que permite trabajar con cuestiones de interfaz en forma separada del aplicativo subyacente, respetando la estrategia de Independencia de Diálogo.
■El separar el “look & feel” de un sitio respecto de sus aplicativos funcionales, desde las etapas iniciales del desarrollo durante su diseño y en el testeo, permitirá un esquema de trabajo más organizado, eficiente, que va a incidir favorablemente en la calidad del producto final.

Ingeniería de Usabilidad

La usabilidad o calidad de uso, es uno de los objetivos más importantes de la interfaz del usuario. La misma especifica el grado de efectividad que tiene la interacción que se provee entre los usuarios y el sistema.

Es una conjunción de proveer simplicidad en la interfaz, que sea fácil de usar y aprender, brindar eficiencia y productividad en la utilización e interacción y lograr satisfacción subjetiva por parte de los usuarios.

Jacob Nielsen [Nielsen, 1994] es el promotor de este concepto y él introdujo a la Ingeniería de Usabilidad como una metodología de desarrollo adecuada a la construcción de sitios Web.

Esta metodología implica encarar el proceso de desarrollo de un sitio donde se tiene en cuenta cuestiones de usabilidad desde el principio y a lo largo del mismo.

Mediante este esquema metodológico, se ayuda a los desarrolladores a focalizar su atención en los usuarios más que en la tecnología. Por ello, provee una estructura clara para la participación efectiva de los usuarios, mejorando la calidad y el trabajo de reingeniería.

Está constituido por un conjunto de procesos, tales como:

- Conocimiento del usuario
- Definición de objetivos de usabilidad
- Análisis heurísticos, Análisis competitivo
- Diseño iterativo, Diseño paralelo, Diseño participativo
- Diseño coordinado de la interfaz del usuario global
- Prototipación
- Testeo empírico
- Recolección de feedback o información de retorno sobre el uso real del sitio

Hay procesos que ya lo hemos explicado en el capítulo de Desarrollo de la Interfaz del usuario, pero hay otros que requieren cierta explicación, como los que se van a detallar a continuación:

Procesos de la Ingeniería de Usabilidad	Explicación
Definición de objetivos de usabilidad	■Implica la definición de metas para poder alcanzar la usabilidad. ■Se incluye como mínimo los objetivos generales de la interfaz.
Análisis heurísticos	■Para alcanzar un grado de usabilidad en un sitio se requiere realizar determinado análisis heurístico como ser: -Seguir recomendaciones publicadas -Investigar heurísticas especificas al tipo de interfaz -Utilizar técnicas como testeo empírico.
Análisis competitivo	■Es un proceso en el cuál se presentan casos presentes en el mercado, considerando aspectos del diseño y se analizan sus características, ventajas y desventajas.
Diseño iterativo, participativo y paralelo	■Es un proceso que se adecua a la “cultura de la Web”, donde los sitios se encuentran en constante cambio. ■La percepción de un usuario de un sistema cambia con el tiempo. El uso del sistema va generando nuevos requerimientos que surgen en función del mismo uso y limitaciones funcionales del sistema antes no detectadas. ■Se organizan tests de usabilidad para evaluar el sitio frente a usuario, técnicas de focus group para analizar pros y contras. ■Se analizan y se prueban varias propuestas de diseño. ■Es un proceso de reingeniería donde el usuario tiene una participación activa.
Testeo empírico	■Implica el desarrollo de tests evaluativos para medir la adecuación del diseño. ■Permite fundamentar futuros desarrollos prototípicos. ■Se emplea durante y al final del desarrollo iterativo.
Recolección de feedback de los usuarios	■Hace referencia a seguir obteniendo la participación del usuario una vez que el producto haya sido instalado y puesto en marcha. ■Es crucial para permitir el rediseño del mismo. ■Entre las prácticas comunes se encuentran opciones de correo, registro interno de información de usuarios, sugerencias, preguntas frecuentes que realizan, entre otros.

Tabla 5.4: Explicación de ciertos Procesos de la Ingeniería de Usabilidad

En el ciclo de vida de la Ingeniería de Usabilidad intervienen ciertos procesos que anteriormente fueron explicados y que denotan tres estadios dentro del desarrollo, la fase de pre-diseño, de diseño propiamente dicho y de post-diseño.

En la siguiente ilustración, se muestran los estadios y los procesos que lo conforman:

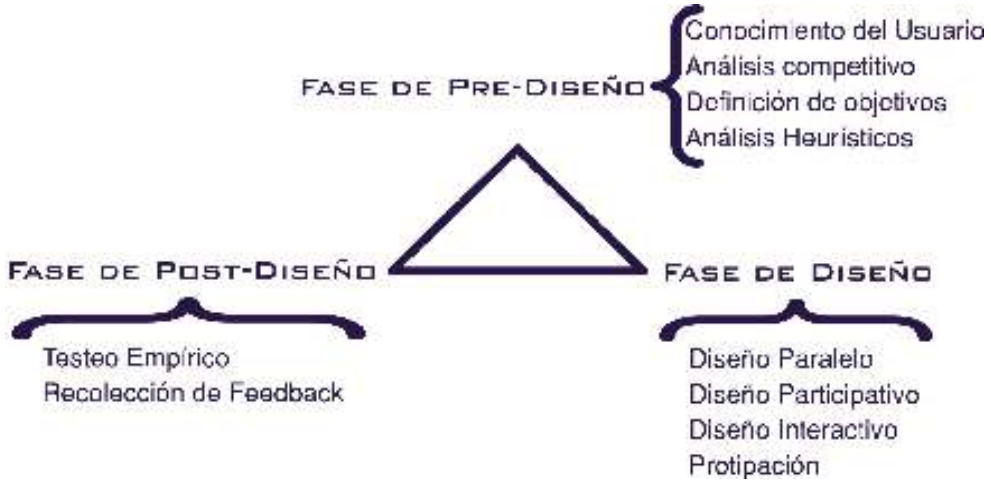


Figura 5.1: Estadíos de la Ingeniería de Usabilidad

Estos estadios tienen una correspondencia, con las etapas tradicionales del desarrollo del software, en donde se tienen las etapas de requerimientos, diseño, implementación y evaluación principalmente.

La relación entre estas etapas y las fases de la Ingeniería de Usabilidad, se visualizan en el siguiente gráfico:

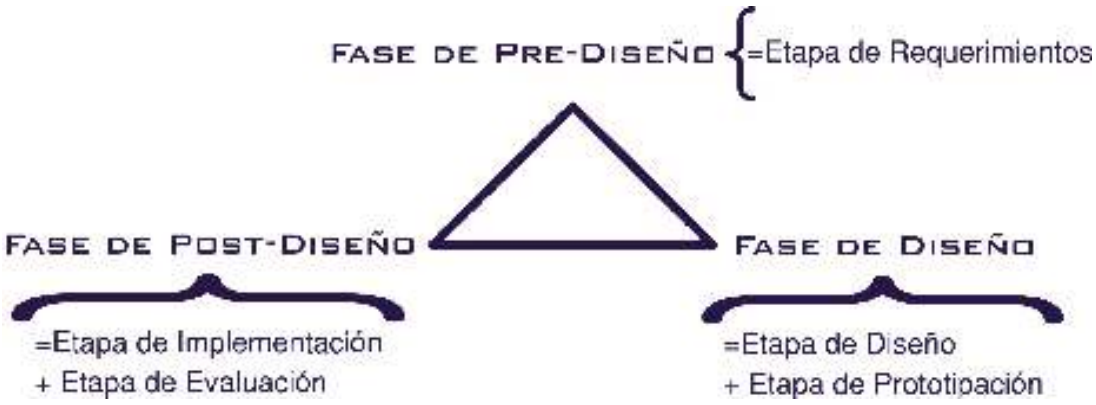


Figura 5.2: Relación con las etapas tradiciones de desarrollo de software.

IV - Principios de Diseño de un Sitio Web

Todos las actividades, análisis y planificaciones que los desarrolladores Web pueden llevar a cabo en la etapa inicial de requerimientos, son determinantes para encarar correctamente el diseño del mismo.

Otra tarea preliminar que incidirá enormemente en la calidad del producto final, es la investigación y el recopilado de los principios y recomendaciones de diseño, que existen actualmente para la Web.

Hay innumerables normas y sugerencias sobre el buen diseño, que se aconsejan cumplir dentro del desarrollo del sitio Web, mayormente publicadas en la World Wide Web Consortium, http://www.w3c.es. Este consorcio promueve y desarrolla los estandares de la Web, como HTML 4 y 5, este ultimo de reciente publicación. HTML5 mejora en forma significativa cuestiones de accesibilidad y semántica del contenido de la Web. Para mas información sobre la Web semántica se puede acceder a la sección de web semántica de la W3C. Las cuestiones de accesibilidad se abordan en el siguiente capitulo. Otro ejemplo de definición de estandares de la W3C es la recomendación sobre el uso de hojas de estilo en cascada, o CSS por sus siglas en inglés Cascading Style Sheets. Las hojas de estilo permiten separar la visualización del contenido de un sitio Web y asegurar la misma continuidad visual en todo el sitio. Facilitan diferentes formas de presentación, por ejemplo al ser accedido desde diferentes dispositivos o para ajustarse a las preferencias de los usuarios, sin necesidad de modificar los cientos de páginas que conforman el sitio Web. Se recomienda usar una única hoja de estilo para todas las páginas del sitio, en un archivo separado. Por cuestiones de accesibilidad, es importante tener en cuenta que el sitio debe poder visualizarse correctamente aún sin hojas de estilos, no usar atributos que ignoren las preferencias de los usuarios, no usar más de dos tipos de letras diferentes.

Otra recomendación es el considerar el desarrollo de políticas P3P, Platform for Privacy Preferences. El mismo es un proyecto de la W3C que fomenta la publicación de las políticas de privacidad de los sitios Web en un formato estándar, que sea entendido por los clientes y los sitios Web. Los clientes Web deberían brindar una herramienta amigable para configurar sus opciones de privacidad y luego, al navegar por los diferentes sitios se bloquearían las cookies en forma puntual, y no en forma general como es el comportamiento hasta ahora. Los sitios Web deben definir sus políticas de privacidad en forma abierta y los clientes Web deberían mostrárselas al usuario en forma amigable para que las configure según sus preferencias. Existe un debate sobre si su uso es adecuado o no, sobre las leyes que lo respaldan y la dificultad de configuración, e incluso algunos navegadores muy usados recomiendan que sean deshabilitadas.

Las normas de accesibilidad propuestas por la WAI -Web Accesibility Initiative- perteneciente a la W3C, también constituyen una guía de recomendaciones para el diseño Web. Constituyen un estándar que agrupan estrategias, guías y recursos para permitir que la Web sea accesible para todos, más allá de las capacidades psico-fisicas de los usuarios, el equipamiento disponible, velocidad de conexión y entorno desde donde lleva a cabo la interacción. Es recomendable tener en cuenta estas guías desde el inicio del desarrollo del sitio Web. En el capítulo VI se analizarán estas normas con mayor profundidad.

La W3C también hace referencia a las consideraciones sobre Web móvil, ¿el sitio Web soportará acceso a través de dispositivos inalámbricos del tipo teléfono celular o PALMs? Las Mobile Web Best Practices (MWBP) constituyen una guía para hacer sitios web usables desde dispositivos móviles. Las interfaces móviles ameritan un capítulo aparte, que no han sido incluidas en este libro. Si desea obtener más información consultar en http://www.linti.unlp.edu.ar/tiki-page9419.html?pageName=pubRec-2006.

Es importante también considerar otras cuestiones técnicas específicas a la Web, como es la seguridad en las aplicaciones Web. Hoy en día es fundamental implementar aplicaciones seguras, a fin de garantizarle al usuario que sus transacciones se realizan de forma confiable y evitar ataques clásicos, como virus y gusanos, robo de identidad y datos personales a través de programas espías o spyware; o direcciones de correo electrónico que solicitan que confirmen determinados datos personales almacenados en un dominio, por ejemplo una entidad bancaria, y en realidad la información reenviada es destinada a personas ajenas (phishing).

Si bien los usuarios deben conocer estas cuestiones y actuar responsablemente durante su navegación, es importante brindar desde el sitio Web que se está diseñando un conjunto básico de aspectos de seguridad. Por ejemplo, para recoger información sensible en línea, se recomienda implementar algún mecanismo de seguridad adecuado, por ejemplo certificados SSL (Secure Socket Layer), particularmente si el sitio Web contiene formularios o actividades transaccionales. Utilizar firma digital y diseñar mensajes de advertencia claros para los usuarios al acceder a sitios no seguros. Minimizar la complejidad de las contraseñas, tanto como sea posible. Es muy común encontrar un papel adherido a los monitores con las credenciales para acceder al sistema. Solicitar como identificación de usuario la dirección de correo electrónico, así se garantiza la unicidad y rápida memorización. Si el sitio funciona como puerta de entrada para otros sistemas que requiere autenticación, intentar implementar single-sign on, es decir, el usuario ingresa sus credenciales una única vez y estas “le abren la puerta” a las otras aplicaciones asociadas. Por ejemplo, el sitio Web para la gestión de alumnos que brinda operaciones transaccionales, como escribirse a una mesa de examen, podría interconectase con el sistema de bibliotecas y el sistema de la plataforma virtual y el alumno podrá navegar entre los tres sistemas sin necesidad de tener que ingresar las credenciales al cambiar de contexto.

La seguridad es un tema complejo, pero que es importante considerar determinados aspectos, aún cuando no tengan que ver estrictamente con cuestiones de usabilidad.

Las políticas de uso aceptable y de confidencialidad de la información también deberían ser consideradas, cuidadosamente elaboradas y publicadas en forma visible y comprensible. Las políticas de uso aceptable, abreviado AUP por Acceptable Use Policy, se denominan también términos del servicio o términos de uso. Constituyen un conjunto de normas que indica como debe usarse un servicio en línea, por ejemplo que ante determinadas acciones no adecuadas del usuario puede suspendérsele el servicio. En general son escritos por los creadores de sitios web, empresas, universidades, etc. principalmente para reducir las potenciales acciones legales que pueden ser tomadas por un usuario. Las políticas referidas a la confidencialidad de los datos, determinan qué información que el usuario brinda al sitio, por ejemplo edad, teléfono, dirección de correo electrónico, gustos personales u ocupación, pueden ser compartidas a terceras partes. Existen organizaciones independientes, como TRUSTe y Better Business Bureau, entre otras, que certifican que las políticas de confidencialidad son cumplidas, responden a estándares internacionales, y además resuelve disputas de privacidad de la información de particulares.

El diseñador puede, luego de un proceso de análisis general, acotar las reglas que va a considerar, armándose su propio repertorio de normas de acuerdo al objetivo del sitio que se va a encarar, al tipo de páginas y componentes que contendrá y, de acuerdo al tipo de contenido que se pretende proveer.

En esta sección, se van a listar los principios de diseño más importantes que por lo general, se recomienda aplicar y cumplir dentro de la interfaz del usuario del sitio Web. Para mayor claridad, se presentarán estas normas en forma organizada y clasificada, de acuerdo a las siguientes categorías:

- **Normas a nivel de sitio:** donde aquí, se analizarán aquellas normas específicas tanto a la estructura del sitio, a la navegación como a los hipervínculos.
- **Normas a nivel de página:** aquí se van a tratar las reglas de diseño para las páginas que conformarán el sitio, agrupando las normas para el diseño general de las mismas, específicamente en la página inicial y en la de formularios.
- **Normas a nivel de contenido:** se explicarán las normas que deben ser aplicadas a nivel de la información que se va a transmitir, teniendo en cuenta características del lenguaje, formas de diálogo, de organización del texto y demás cuestiones.

Principios de Diseño a Nivel de Sitio

Como se dijo anteriormente, las normas que se refieren al sitio en general van a ser agrupadas a su vez, en normas de diseño a nivel de estructura del sitio, a nivel de navegación y a nivel de hipervínculos.

La Estructura del sitio Web

Uno de los puntos más importantes a considerar dentro del diseño del sitio en general, es la estructura del mismo.

En un ambiente tan visual como la Web, el buen diseño de la interfaz tiene mucho que ver con el sentido y el grado de expresividad de dicha visualización, no sólo aplicada en la representatividad de la información, sino a la estructura organizativa del sitio.

Construir y mantener una relación coherente y estrecha entre la arquitectura y la interfaz, es decir, entre la lógica de la estructura y el significado visual, es fundamental para lograr la mejor adecuación e interacción del usuario respecto al sitio.

Muchas veces los usuarios se pierden dentro de un sitio Web, simplemente por el hecho que el mismo carece de un delineamiento organizado o estructurado, o sea que cuestiones estructurales no fueron encaradas desde el principio, en forma deliberada.

En ciertos casos, los diseñadores comienzan a construir sistemática e irreflexivamente numerosas páginas de distintas clases y tamaños, enganchadas entre sí mediante una cantidad de hipervínculos, sin analizar una lógica general de organización.

Esto se manifiesta posteriormente dentro de la interfaz, provocando innumerables problemas de navegación e interacción, muy difíciles de subsanar una vez que el sitio ya ha sido desarrollado.

Como es sabido, la forma de organización de la información inherente a la Web es la organización hipermedial, o sea que las páginas y sus hipervínculos son representados y tratados internamente como nodos y arcos de un hipergrafo.

No sólo se proveen hipervínculos para relacionar contenido, sino para acceder a las funciones principales, a servicios secundarios, adicionales, a información no tan relevante o de relleno, para ir atrás y adelante, arriba o abajo, para enlazar páginas por acceso directo o acortamientos, para acceder a otros sitios, o por otras cuestiones.

Por ello, para poder paliar esta situación, el desarrollador debe primeramente definir y trabajar con una estructura básica, en donde se consideren únicamente las páginas más relevantes del sitio y los hipervínculos esenciales para poder acceder a dicha información.

Esta estructura, más simple y reducida que el grafo anterior, es la que debe ser visualizada y transmitida al usuario, para que el mismo pueda percibir al sitio a través de una perspectiva más sencilla y pueda a partir de allí, entender la lógica y el significado de esta estructuración inicial.

Una vez definido el esqueleto estructural del sitio, el diseñador podrá agregar la restante información adicional y todos las hiper-conexiones que crea necesarias, por lo que esta estructura básica debe estar definida de tal forma, que soporte las futuras extensiones y amplificaciones.

Entonces, el diseñador deberá trabajar arduamente en el diseño general del sitio y en la definición de una estructura elemental del mismo, como se explican en las siguientes normas:

Principios de Diseño a Nivel de Estructura	Explicación
Organizar el contenido del sitio, previo a la carga del mismo	■Se debe analizar la cantidad de información a publicar en el sitio, y estudiar formas de organización hipermedial.
Definir la estructura elemental del sitio	■Se debe fijar una estructura simple de organización del sitio, que sea lógica y significativa, constituida por las páginas e hipervínculos más relevantes del sitio. ■Hay que construir una infraestructura sólida para soportar el nivel de actividad y dinamismo de los usuarios. ■Se puede optar como estructura principal del sitio, alguna de los siguientes esquemas:
Hacer que la estructura sea usable	■Para hacer que la estructura sea usable, hay que considerar conjuntamente, los siguientes aspectos:
Evaluar si la estructura es extensible	■Hay que comprobar que la estructura analizada permita y soporte escalabilidad, crecimiento y posibles cambios en el sitio.

Secuencial	■las páginas principales se vinculan en forma lineal.
Jerárquica	■las páginas se vinculan en forma de árbol.
En Estrella	■las páginas se vinculan a través de una página central.
de búsqueda	■hay una búsqueda central para el hallazgo y acceso directo de las páginas más importantes.
Grafo reducido	■las páginas se encuentran hiperconectadas mediante una profundidad y anchura moderados.

Visualización	■la estructura debe tener una visualización aparente y acorde a ella.
Prioridad	■dentro de la estructura debe posicionarse primero lo más importante.
Consistencia	■se debe hacer el “look&feel” de la interfaz acorde a la estructura.
Flexibilidad	■la estructura debe brindar distintas alternativas y permitir diferentes caminos para alcanzar la información que el usuario requiere.
Inmediatez	■la estructura debe solventar cuestiones de acceso directo y acortamientos, permitiendo que el usuario realice sus intenciones en pocos pasos.

Tabla 5.5: Normas de diseño a nivel de estructura.

La Navegación del sitio Web

La navegación se refiere a la forma en que los usuarios construyen su exploración a través del sitio, mediante la utilización de los hipervínculos, que constituyen el medio para trasladarse y recorrer los diferentes puntos del mismo.

El usuario podrá navegar dentro de una misma página o entre varias páginas que pueden pertenecer a un mismo sitio o no.

Si uno tiene en cuenta la organización hipermedial de la Web, entonces, se puede decir que la navegación implicaría la determinación y recorrido de un camino dentro del grafo subyacente, conformado por aquellos nodos y arcos que respectivamente, son las páginas visitadas y los hipervínculos activados.

En la siguiente tabla, se resumirán las recomendaciones de diseño más importantes para el proceso de navegación:

Principios de Diseño a Nivel de Navegación	Explicación
Definir diferentes tipos de navegación	■En un sitio, se pueden proveer diferentes tipos de navegación que deben ser estar bien definidos y tratados coherentemente:
Navegación por estructura	■determinada por los hipervínculos que permite recorrer las páginas más importantes del sitio. Es denominada “cross-referencing”
Navegación de la sesión	■permitir al usuario navegar hacia delante y atrás en el sitio, sin acceder al historial del navegador. ■Se la denomina “cross-navigation”
Navegación contextual	■mantener la noción el contexto. Noción de la ubicación del usuario, con información de los pasos anteriores, actual y posteriores.
Navegación adaptada	■adaptar la navegación al usuario, adelantando hipervínculos, según estudios que se realizan durante las sesiones. ■se debe trabajar con tácticas que identifiquen la dirección IP del usuario en pos de customizar su interfaz, utilizando técnicas como IP-Sensing.
Navegación interactiva	■se puede ayudar en la navegación del usuario en forma interactiva, mediante preguntas sobre las intenciones del usuario y acceso directo.
Establecer la mejor visualización de la navegación o “feedback navegacional”	■Se debe especificar la mejor manera de visualizar los diferentes tipos de navegación que se proveen. ■Cada tipo de navegación debe ser visualizados con un color, tipo de letra, fondo, ubicación, que sea específico para él y que lo identifique a lo largo de todo el sitio. ■Se puede trabajar con distintas barras de navegación o cualquier otra forma visual, pero cada tipo de navegación debe estar claramente distinguido, agrupado, localizado y visualizado.
Tratar los problemas inherentes a la navegación	■A la navegación en general se le atribuyen ciertos problemas que deben ser considerados y tratados a nivel de diseño. ■Entre los problemas más importantes, se tiene:
Pérdida de contexto	■se debe informar al usuario sobre dónde se encuentra, de dónde viene y hacia dónde puede ir. ■se los debe ayudar en una transacción, indicando en qué paso está, qué es lo que hizo y qué le resta hacer ■no interrumpir la navegación generando ventanas nuevas.
Dependencia con el navegador	■se debe evitar que el usuario acceda a las funciones del navegador. ■debe existir unión entre datos y control dentro del mismo sitio.
Demora en la navegación	■hay que paliar el problema de las demoras, mediante información de contexto, mensajes informativos, ayudas.
Pérdida del usuario	■el usuario debe identificar los hipervínculos externos, o sea los que acceden a otros sitios. ■hay que informarlo y advertirlo de esto.
Analizar cuestiones específicas	■Permitir accesos directos, complementarlo con un sistema de búsqueda. ■Considerar la “regla de 3 clicks”, para alcanzar un objetivo. ■Proveer un mapa de navegación accesible en todo momento. ■Proveer un tour dentro del sitio, que lleve al usuario a los puntos más sobresalientes del mismo.

Tabla 5.6: Normas de diseño a nivel de navegación.

Los Hipervínculos

Un hipervínculo es una palabra, frase, imagen o símbolo, que permite a los usuarios saltar de un lado a otra ubicación del mismo sitio o de algún sitio externo.

Como en el resto de los casos, existen normas específicas a tener en cuenta en el diseño de los hipervínculos, que se van a resumir a continuación:

Principios de Diseño a Nivel de Hipervínculos	Explicación
Destinar una ubicación independiente para los hipervínculos	■Los hipervínculos deben estar ubicados en forma independiente. ■Si son hipervínculos asociados a la información, no deben estar entremezclados al texto, para evitar distracción e interrupción y facilitar la lectura local. Pueden ubicarse al costado o al final. ■Lo mismo sucede con los hipervínculos asociados a un tipo de navegación.
Distinguir semánticamente los hipervínculos	■Cada hipervínculo debe estar referenciado por un término o imagen de acuerdo al significado y al destino del mismo. ■No debe existir ambigüedad, intersección o inclusión entre hipervínculos. ■Evitar el uso de la frase “Cliquee aquí” o “más info”
Categorizar los hipervínculos	■Los hipervínculos deben ser claramente distinguibles y clasificados según estén asociados al contenido, a los servicios, a la funcionalidad, a los distintos tipos de navegación. ■Además, hay que diferenciar hipervínculos internos de los externos que permiten al usuario abandonar el sitio.
Proveer información complementaria	■Se debe brindar información sobre el destino del hipervínculo -descripción, tamaño- para que el lector pueda predecir hacia donde se dirige. ■Se debe proveer información de los hipervínculos en la barra de estados ■Se puede utilizar texto en los hipervínculos o gráficos acompañados por texto para clarificar el sentido de los mismos.
Considerar aspectos de visualización■	■Hay que analizar el aspecto visual del hipervínculo, su color, disposición, tamaño, longitud que debe ser acorde al sentido del mismo. ■Se debe controlar el tamaño del hipervínculo. Si es textual, no debe superar más de 15 caracteres. ■Se deben utilizar colores significativos para los diferentes estados de los hipervínculos. El usuario debe percibir claramente el estado del hipervínculo, si fue visitado, seleccionado, si está habilitado o no. ■Se puede recurrir a estructuras visuales como tablas, menús o índices de hipervínculos.
Proveer Consistencia y homogeneidad	■Se debe mantener consistencia entre el nombre del hipervínculo y del título de la página destino. ■Aspectos de visualización de los hipervínculos como lugar, imagen, tipografía utilizada para los mismos deben mantenerse homogéneos en todas las partes del sitio.
Proveer Correctitud	■Se deben proveer hipervínculos correctos. Si son externos deben saltar a páginas de sitios que realmente existan en la Web. ■Hay que mantener actualizada las referencias URL que puedan dirigir ciertos hipervínculos.

Tabla 5.7: Normas de diseño a nivel de hipervínculos.

Principios de Diseño a Nivel de Página

En esta sección, se van a listar las recomendaciones de diseño a nivel de página, separando las normas según el diseño general, la página principal y para aquellas que provean formularios.

En el caso que el diseñador desee proveer otro tipo de páginas como ser, de transacciones, de contacto, publicitarias, entonces va a tener que reunir las normas específicas de acuerdo a ello y respetarlas en su diseño del sitio.

Mediante estas normas se recomienda al diseñador comenzar con definiciones de moldes o "templates" de páginas, teniendo en cuenta cada tipo de página que el sitio proveerá.

Esto constituye un buen esquema de trabajo grupal, puesto que provee una guía de diseño de cómo deben construirse las páginas del sitio que el equipo de trabajo deberá respetar, y fundamentalmente para garantizar consistencia a lo largo de todo su desarrollo.

Diseño general de las Páginas

Todas las páginas del sitio deben construirse a partir de la definición de un modelo, estipulado y diseñado bajo determinados lineamientos.

El mismo consistirá en la definición de pautas relacionadas con la diagramación general, sectorización, organización del contenido, aspectos de visualización como el logo, especificación del tipo de letra general, su tamaño, fondos y colores a utilizar.

Este modelo conformará el diseño general de páginas que deberá estar definido, a parte de la definición de moldes más específicos para diferentes tipos de páginas.

Con respecto al diseño general de las páginas, se deben respetar las siguientes recomendaciones:

Principios de Diseño General de Páginas	Explicación
Clasificar el contenido	■Toda página de un sitio debería tener como contenido cierto tipo de información, clasificada de la siguiente manera: **Contenido primario**: ■conformado por el título, subtítulos, objetivos, descripción de la página. **Contenido informativo**: ■conformado por la misma información que se desea transmitir en la página. **Contenido funcional**: ■con los servicios que se brinda y la funcionalidad. **Contenido secundario**: ■conformado por otras funciones e información adicional. **Contenido contextual**: ■formado por propagandas, artículos, sugerencias, relacionadas con la información publicada. **Otros**: ■servicios adicionales como ayudas, salidas, impresión, guardado, de comunicación como e-mail, FAQs, contactos, entre otros.
Sectorizar de acuerdo al contenido	■Las páginas deben ser sectorizadas claramente de acuerdo al tipo de contenido analizado en el punto anterior. ■Deberá tener un sector para títulos, un sector funcional, otro para el contenido informativo, otro para servicios adicionales, etc.
Diagramación general	■Se debe establecer cuestiones generales de cómo serán las páginas del sitio, definiendo tipo de contenido y sectores que va a tener, la especificación de un área común. ■Se deben tomar decisiones sobre formas de organizar el contenido informativo, con párrafos extensos o no, si se van a preferir brindar páginas extensas o sintéticas pero aumentando el número de hipervínculos. ■Se deben especificar las cuestiones de visualización, tipografía, fondos, tamaños, y gama general de colores a utilizar. ■Recordar la recomendación sobre el uso de hoja de estilos mencionada en el sección de principios de diseño de un sitio.

Especificar un área común	■Cualquier página del sitio, debe contener un sector o contar con un marco común que identifique al sitio. ■En este marco común, puede tener un logo, un menú con ciertas opciones e información que permanecerá en todas las páginas.
Proveer homogeneidad	■Se debe proveer un diseño homogéneo en todo el sitio. ■El sitio debe ser consistente en dos aspectos: **Visual**: ■Visual significa utilizar para los mismos tipos de páginas, el mismo tipo de letra, colores, fondos y demás recursos visuales. **Conceptual**: ■Emplear terminología, conceptos, definiciones consistentemente a lo largo de todo el sitio.
Aprovechar los recursos visuales para denotar significado	■Los recursos visuales pueden transmitir información y denotar ciertos significados. ■Por ejemplo, la semántica del color rojo para denotar peligro, o el amarillo para precaución. ■Por lo tanto, hay que considerar las siguientes cuestiones: -Utilizar la tipografía adecuada acorde al contenido del sitio. -Usar colores para denotar importancia y capturar atención. -Usar la noción de tamaño para transmitir importancia y permitir distinto grado de reconocimiento y memorización. -Emplear la localización de objetos para transmitir importancia o secuencia de acciones. Lectura de izquierda a derecha y de arriba abajo. -Usar la noción de agrupamientos y criterios de ordenación. -Utilizar metáforas, explicar conceptos mediante asociaciones, analogías, ejemplos, conceptos familiares al usuario.
Evitar sobrecarga visual	■El diseño debe ser estético y minimalista, combinar el buen gusto con utilización de mínimos recursos por parte del cliente. ■Usar imágenes pequeñas inferiores a 40 Kb y multimedia solamente como medio alternativo e incluyendo el soporte necesario. ■Utilizar escasas animaciones y movimientos, sólo para denotar conceptos muy importantes pues atraen mucho la atención. Además, deben contener funcionalidad sintáctica para interrumpirlos, reasumirlos, configurarlos.
Considerar aspectos de diseño universal	■Tener pensamiento global, distintos formatos de archivos, adecuación a distintas conexiones, tipos de monitor, tipos de navegadores y cuestiones de internacionalización. ■Respecto a este último punto, en la W3C existe un grupo de trabajo, el W3C Internationalization (I18n), que intenta asegurar un conjunto de formatos y protocolos que puedan utilizarse en todos los idiomas y culturas, garantizado el acceso universal.
Probar el sitio desde diferentes situaciones	■Hay que analizar el sitio desde distintos aspectos técnicos, probando velocidad de carga, volumen de información, desde diferentes recursos y medios. ■El tiempo de navegación de una página a otra, debería aproximarse al de la vuelta de página de un libro. ■Tener en cuenta la compatibilidad entre diferentes navegadores dado que ciertas etiquetas HTML o funciones de JavaScript no se visualizan de la misma manera en la familia Mozilla y en el Internet Explorer de Microsoft. ■Recordar las normas WAI recomendadas en el sección de principios de diseño de un sitio para brindar acceso universal.
Permitir signos de adaptación del sitio	■El usuario debe adaptar, personalizar, modificar ciertas características de visualización, interacción y utilización del sitio. ■Debe mantenerse esta configuración a lo largo de la sesión y en las próximas sesiones.

Tabla 5.8: Normas de diseño a nivel de diseño general de páginas.

La Página Inicial

La página inicial es sumamente importante, no sólo por ser la primera impresión que el usuario tiene del sitio entero, sino por ser la puerta de entrada, la que lo introduce al mismo y se lo presenta.

Habitualmente, el usuario tiene como expectativas de la página inicial, el obtener en forma directa un panorama general de lo que el sitio ofrece, y de lo que se puede hacer en él.

Los principios correspondientes a nivel de página inicial, son:

Principios de Diseño a Nivel de Página Inicial	Explicación
Proveer información general del sitio	■Se debe claramente especificar lo que se ofrece en el sitio. ■Hay que proveer una página inicial clara y sintética que puntualice el objetivo del sitio y provea un índice de los temas abordados en él.
Definir el aspecto y diagramación inicial	■El aspecto y la organización que se percibe en la página inicial, da una primera perspectiva de lo que se puede encontrar en todo el sitio. ■La página inicial debe estar diagramada y visualizada acorde al diseño general del sitio, aunque posea ciertos rasgos distintivos, para sobresalirla del resto.
Visualizar el marco común a todas las páginas	■Desde la página inicial debe estar presente el marco común a todo el sitio, aunque en ésta tenga un aspecto distinguible. ■Debe visualizarse el logo, el nombre del sitio, funciones principales, menús, que estarán siempre presentes a lo largo de todo el sitio.
Dar información sobre la registración	■Si el sitio requiere que el usuario se registre en forma obligatoria, antes de navegar, entonces debe dar información de ello en la página inicial. ■Se debe sugerir al usuario cómo debe proceder, informándole sobre los beneficios de registrarse, y donde se observe el estado de inhabilitación de los servicios e hipervínculos previo a ello.
Dar información sobre el estado del sitio	■Se debe informar permanentemente sobre el estado del sitio y de los servicios. ■Se debe indicar si hay páginas en construcción o no y cuándo van a completarse. ■Se debe señalar fecha de últimos cambios o de actualizaciones.
Definir áreas prioritarias	■Hay que promover áreas prioritarias que se desea que el lector visite, recomendaciones, páginas más importantes, más visitadas, entre otras.
Permitir el acceso a la página inicial	■Se debe ir fácilmente a la página inicial desde cualquier punto del sitio.
Proveer servicios adicionales	■Proveer ayuda general del sitio, un sistema de búsqueda para el acceso directo, un mapa del sitio, un tour recomendado por el autor, un asistente interactivo, ayuda en línea, entre otros.

Tabla 5.9: Normas de diseño a nivel de página inicial.

Página con Formularios

Un formulario es una componente visual que permite reunir un conjunto de diferentes objetos de interacción, en pos de permitir la entrada de datos por parte del usuario.

El mismo puede estar compuesto de casilleros de entrada de textos, títulos, listas de selección, botones, casilleros de chequeo, entre otros.

Existen normas específicas a páginas que posean formularios, como las siguientes:

Principios de Diseño de Formularios	Explicación
Proveer un diseño flexible de formulario	■El usuario puede comenzar a completar el formulario desde cualquier punto del mismo, suspenderlo, reasumirlo, ir hacia el principio o hacia el final del mismo, sin inconvenientes. ■Permitir un orden arbitrario o flexible de llenado de datos.
Proveer funciones sintácticas	■Brindar funciones específicas de formularios, como ser soporte para suspender el llenado, reanudarlo, guardarlo, imprimirlo, obtener ayuda.
Brindar medios alternativos de activación de los casilleros	■Permitir la activación de los casilleros a través del teclado, con la tecla <Tab> por ejemplo. ■O mediante acceso directo a los mismos con teclas de función o atajos.
Distinción visual	■Distinguir entre entradas obligatorias de las que no lo son, entradas importantes de las que no lo son.
Proveer validación	■Proveer validaciones léxicas, sintácticas asociadas a cada casillero de entrada y a todo el formulario.
Proveer feedback adecuado	■Proveer mensajes aclaratorios, ayudas, ejemplos, rango de posible valores de entrada para el ingreso de datos ■Analizar la frecuencia e inmediatez de aparición
Diseñar mensajes de error eficientes	■Brindar mensajes claros, no ambiguos, que den alternativas a seguir, que sugieran formas de solucionar el conflicto y salir de él. ■Analizar que el tiempo entre la aparición del mensaje respecto del momento en que se cometió el error sean apropiados.
Considerar el diseño de las componentes del formulario	■Tener en cuenta aspectos de diseño a nivel de cada componente del formulario y de la disposición y visibilidad de las mismas. ■En el caso de los botones radiales, no se puede utilizar más de 5 opciones. ■En el caso de los casilleros de entrada de texto, se recomienda proveer el tamaño adecuado, no truncar, no utilizar scrolling, mostrar estados posibles como ser de edición o no.
Considerar cuestiones de seguridad	■La información que se intercambia entre el cliente y el servidor debe contar con algún mecanismo de encriptación. Bajo ningún concepto debe viajar un texto plano.

Tabla 5.10: Normas de diseño a nivel de página de formularios.

Principios de Diseño a nivel de Contenido

Escribir para la Web, es muy distinto que el realizar publicaciones en papel. Por ejemplo, está comprobado que la lectura horizontal y la presencia de los monitores decrecen la velocidad de lectura en un 25 %.

Por lo tanto, es necesario considerar ciertos aspectos de diseño a nivel de contenido, que permita mejorar el grado de comunicación y de percepción.

Se van a presentar recomendaciones de diseño a nivel general como también del lenguaje empleado.

Diseño general del Contenido

Principios de Diseño general de contenido	Explicación
Organizar y visualizar el contenido acorde a su nivel de importancia	■Se debe capturar la atención al usuario sobre la información importante, por lo tanto la ubicación y visualización debe estar diseñados para resaltar lo relevante y distinguirlo de lo que no lo es.
Proveer el contenido apropiado	■Se debe brindar información y servicios acorde al objetivo. ■El contenido de la página debe ser coherente con el título y con el hipervínculo que permite su acceso. ■El contenido de una página debe tener el mismo nivel de abstracción.

Publicar contenido comprensible	■La gramática a utilizar, la forma de expresión debe ser simple, que no complique la interpretación por parte del usuario. ■El usuario no debe acudir a la navegación para entender qué se quiso decir en un determinado punto, o cuál es su significado.
Aprovechar diferentes recursos visuales para expresar el contenido	■Aprovechar el uso de fragmentaciones del texto como frases, párrafos, tabulaciones, listados, tablas. ■Utilizar objetos de interacción como menús, ventanas, ficheros interactivos o tab-set, capas o layers para sintetizar espacio.
Proveer una visualización estándar adicional	**■Usar tipo de alineación estándar a izquierda, letra de tamaño habitual 12 y de tipo sans serif aunque sea como configuración alternativa.**
Relacionar contenido con la navegación	**■Cada nivel de navegación debe significar mayor nivel de detalle.** **■El contenido debe ser progresivo acorde a la navegación.**
Considerar el diseño particular de los objetos de interacción	**■Hay que considerar que cada objeto de interacción cuenta con su propio repertorio de reglas de diseño, como ser:**
Respecto a los menús	■no se puede mostrar más de 10 opciones, ni con más de 3 niveles de jerarquía ■se debe diferenciar selección múltiple o simple ■visualizar los ítems seleccionados, visitados, habilitado
Respecto al Área de texto	■visualizar el estado de edición o lectura. ■proveer funciones sintácticas
Respecto a las Tablas y listas	■visualizar el estado que puede ser seleccionable/no, editable/no. ■proveer funciones sintácticas de ordenamiento, filtrado, proyección, selección
En el caso del fichero interactivo o Tab set	■se debe visualizar claramente cuál es el tab activo. ■se debe ordenar el fichero en forma acorde con la información mostrada. ■diferenciar funciones a nivel de ficha y de fichero
Respecto a los Layers	■analizar feedback sobre su existencia. ■analizar funciones provistas y dependencias con la hoja contendedora.

Tabla **5.11: Normas de diseño a nivel general de contenido.**

Lenguaje Empleado

Principios de Diseño a Nivel del Lenguaje empleado	Explicación
Proveer un lenguaje acorde al objetivo y a la información a transmitir	■Emplear un lenguaje apropiado al objetivo de la página, utilizar una terminología acorde al tipo de información que se publica.
Utilizar un lenguaje acorde al usuario	■Emplear un lenguaje acorde al perfil o perfiles de usuarios al que está dirigido. ■No utilizar palabras técnicas ni extranjeras
Utilizar expresiones correctas	■No deben mostrarse errores de ortografía ni gramaticales. ■No deben emplearse expresiones confusas, complejas ni que estén compuestas por varias frases y muchos conectores lógicos. ■No abreviar ni truncar expresiones.
Manejar aspectos de visualización	■El lenguaje debe manifestarse a través de colores, tipografía, tamaños en forma coherente. ■Debe ser acorde a la información que se expresa y al perfil de usuarios a quién está dirigido.

Tabla 5.12: Normas de diseño a nivel de Lenguaje empleado.

Caso de Estudio 5 - La interfaz web y sus principios de diseño

El objetivo de este caso de estudio es presentar una práctica donde los estudiantes analizan un sitio Web que sea de su interés, según los nuevos conceptos adquiridos al momento. Evaluar un sitio Web que usan habitualmente según las pautas de un buen diseño Web promueve la reflexión al visitarlo desde otro punto de vista, y evita la demora que conlleva el análsis de un sitio que no se conoce desde su objetivo hasta las formas de interacción.

Objetivo

La práctica consistió en la evaluación de un sitio Web de elección de cada estudiante teniendo en cuenta las consideraciones de diseño trabajadas en las clases teóricas. Uno de los informes destacados fue el realizado por los alumnos Alan Vido (alanvido@yahoo.com) y Facundo Manual Quiroga durante la cursada 2010, sobre el portal de la Biblioteca de la Facultad de Informática. La aplicación permite llevar adelante la gestión completa de libros y usuarios, y obtener los correspondientes reportes.

Es importante destacar que el sitio fue re programado durante el año 2011 y en las etapas avanzadas de la implementación se realizaron testeos de usabilidad que mejoraron notoriamente la interacción con el usuario final. El sitio actual puede consultarse en http://catalogo.info.unlp.edu.ar/

A continuación se presenta un resumen del informe. La entrega completa puede consultarse en el repositorio de trabajos de alumnos de la Facultad, http://dpsace.linti.unlp.edu.ar

Características del Caso de estudio

Se analizó el sitio institucional de la biblioteca de la Facultad de Informática de la UNLP, http://biblioteca.info.unlp.edu.ar/, así como el catálogo online de dicha institución, http://catalogo.info.unlp.edu.ar/. La evaluación se realizó considerando los principios de diseño a nivel sitio de ambos sitios por separado, y luego de los principios a nivel de página y de contenido de los 2 sitios juntos.

Los dos sitios evaluados se utilizan con distintos fines en la misma institución y sin embargo los estilos son diferentes, como se puede obervar en la figura 5.1 y 5.2 Para el caso de estudio se presentan los resultados de la evaluación del catálogo, por tener mas funcionalidades y componentes de interacción aunque se incluyen algunas referencias al sitio de la biblioteca.

Desarrollo del Caso de Estudio

Se analizó el sitio institucional de la biblioteca de la Facultad de Informática de la UNLP, http://biblioteca.info.unlp.edu.ar/, así como el catálogo online de dicha institución, http://catalogo.info.unlp.edu.ar/. La evaluación se realizó considerando los principios de diseño a nivel sitio de ambos sitios por separado, y luego de los principios a nivel de página y de contenido de los 2 sitios juntos.

Los dos sitios evaluados se utilizan con distintos fines en la misma institución y sin embargo los estilos son diferentes, como se puede obervar en la figura 5.1 y 5.2 Para el caso de estudio se presentan los resultados de la evaluación del catálogo, por tener mas funcionalidades y componentes de interacción aunque se incluyen algunas referencias al sitio de la biblioteca.

Figura 5.1 Página inicial de acceso al cátalogo de libros. Es posible consultar e interactuar con el sitio de esa fecha a través de webback machine, en http://goo.gl/tsNum

Figura 5.2 Página de acceso a la información institucional de la Biblioteca. Es posible consultar e interactuar con el sitio de esa fecha a través de webback machine, en http://goo.gl/wXzHr

En la figura 5.3 se presenta el mapa del sitio para la gestión del catálogo de libros. Como se puede observar, la aplicación contiene distintas secciones con contenido claramente identificado.

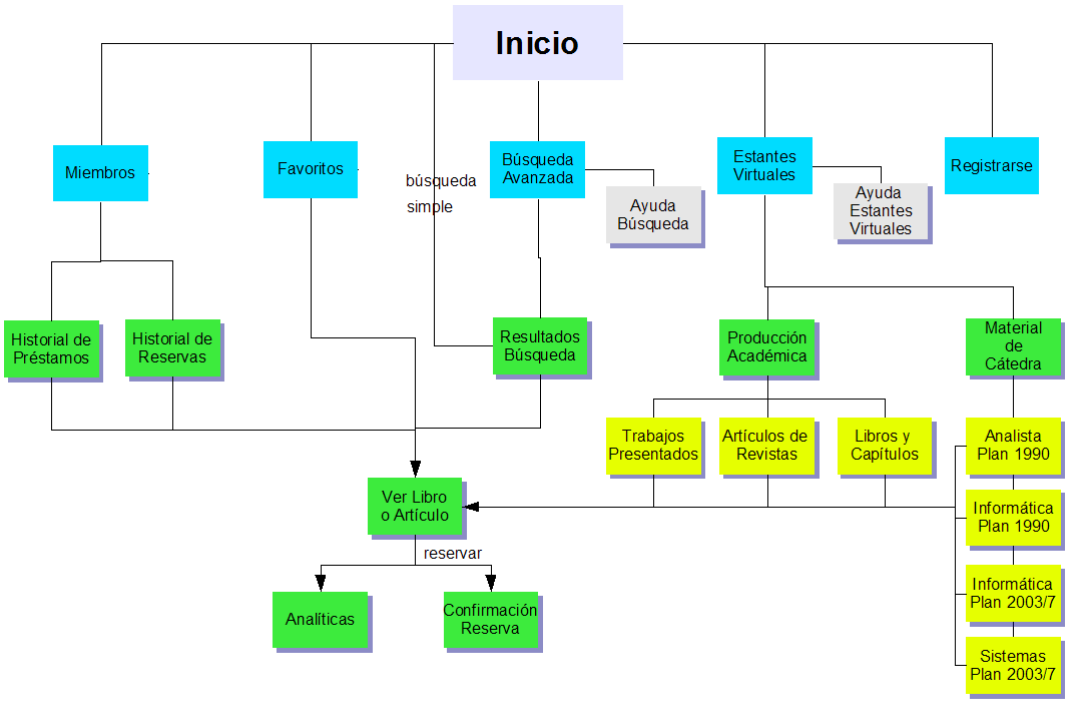


Figura 5.3: Estructura del catálogos de libros de la Facultad de Informática

Principios de diseño a nivel sitio De los catálogos virtuales

La organización del sitio es clara y concreta. Presenta una estructura de búsqueda, ya que es el servicio más importante que provee, aunque también es posible navegar la información por temas o tipo mediante los estantes virtuales, organizados jerárquicamente. Esta estructura brinda flexibilidad en las búsquedas dado que puede encontrar información a través de la búsqueda común, la avanzada, los estantes virtuales, los favoritos, y el historial de préstamos. La barra de búsqueda que se encuentra la parte superior todo el tiempo colabora con la inmediatez. Desde la página de un libro podemos acceder directamente a una lista de todos los libros del autor o de los temas del libro, como así también en ciertos casos a la bibliografía de la materia que lo utiliza.

Sin embargo, presenta ciertos problemas con determinadas recomendaciones referidas a que la estructura sea usable. La visualización no es adecuada dado que no hay un orden ni tampoco se comunica al usuario que el principal medio de interacción es la búsqueda. Desde la página inicial, se presenta como primera opción la búsqueda que es el elemento de interacción principal con el sitio, junto con él inicio de sesión del usuario. Luego, si bien la barra de búsqueda está siempre visible, toma un lugar secundario en la interfaz.

Respecto a definir diferentes tipos de navegación, la principal es la navegación por estructura, a través del proceso de búsqueda se encuentran enlaces iniciales que pueden ser seguidos descubriendo nueva información. La navegación contextual aplica, pero el sitio es muy simple y no hay procesos que lleven varios pasos.

No se observan pérdidas de contexto, dependencia del navegador o demora en la navegación que conlleve la pérdida del usuario y problemas inherentes a la navegación. Se cumple la regla de los 3 clics. Sin embargo, presenta problemas de feedback navegacional dado que no hay ninguna distinción visual importante entre la navegación por búsqueda, mediante los estantes virtuales, los favoritos y a través del historial de préstamos como se puede observar en las figuras 5.4,5.5,5.6, 5.7

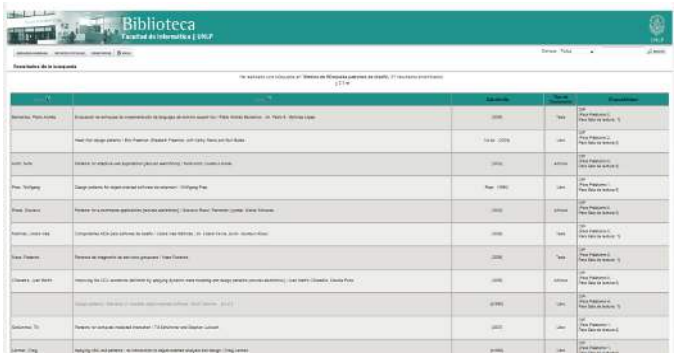


Figura 5.4: Resultados de búsqueda

Figura 5.5: Estantes Virtuales

Figura 5.6: Favoritos

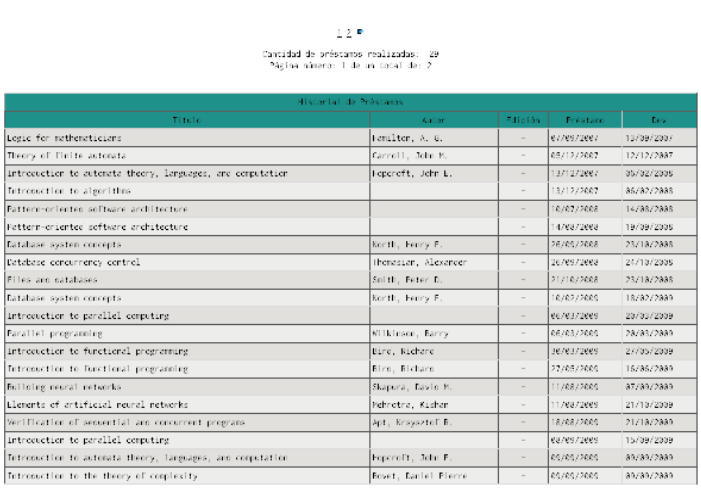


Figura 5.7: Historial de préstamos

Respecto a la localización de navegación, si bien la posición de los elementos de navegación es consistente en todo el sitio, no hay una clara razón por la cual se utilizan 2 menús en vez de uno solo, siendo tan pocos sus elementos. No hay mapa de navegación, y tampoco sería posible tener uno ya que la estructura del sitio no es clara y el mismo no es tan extenso.

A nivel de Hipervinculos se encuentan varios problemas que cumplen a medias estas recomendaciones. Por ejemplo en la lista de resultados que se ilustran en la figura 5.4,5.5,5.6 y 5.7, los nombres de los libros y de los autores sirven también como enlaces, lo cual no es apropiado en ocasiones ya que pueden ser muy largos.

No hay distinción sintáctica entre enlaces internos y externos, aunque se infiere fácilmente su destino por el contexto, como se puede observar en la figura 5.8. Tampoco hay diferencia de color entre los enlaces no visitados y el texto plano, pero sí entre visitados, no visitados y seleccionados.

Figura 5.8: No existe una distinción entre hipervínculos internos y externos.

Tampoco provee información complementaria suficiente. En la página de resultados (figura 5.4), para cada libro se muestra el autor, si tiene ejemplares disponibles y la edición. Podrían agregarse iconos para mostrar si el libro tiene varios autores, para distinguir entre libros, trabajos de tesis, etc., y para reconocer más fácilmente la cantidad de ejemplares disponibles.

Principios de diseño a nivel Página

Respecto a la clasificación del contenido, todas las paginas tienen contenido primario como título y subtitulo, pero las paginas “ayudas e instructivos” y “noticias” no tienen objetivos, descripción ni contenido y no se informa de este problema. Idem para contenido informativo y funcional. El contenido contextual es provisto por la barra lateral derecha y es fijo. La información adicional está provista en todas las páginas, permite impresión, envió por mail y cambiar el tamaño de la fuente. En catálogos se permite la impresión y exportación. No están disponibles en el formulario de contacto, mapa del sitio y tampoco en buscar. Las FAQ están vacías.

La sectorización se respeta en la mayoría de las paginas, excepto en catálogos y mapa del sitio que tienen un aspecto totalmente distinto.

Respecto a la diagramación general, a grandes rasgos la diagramación es respetada por todas las páginas, pero “Catálogos” no la respeta en ninguno de sus aspectos. La forma de organizar el contenido es dispar principalmente en “institucional” en donde no se respetan convenciones tipográficas ni de aspecto en texto presentado así como tampoco en los hipervínculos.

Respecto a la homogeneidad, no se provee una homogeneidad visual a lo largo del sitio, las principales falencias están en “Catálogos” e “Institucional” en donde no se respeta ninguna convención de aspecto. Los recursos visuales se utilizan de manera poco eficaz y en algunas situaciones de manera confusa como es el caso de la sección “institucional” en donde no se respetan los colores y tamaños de fuente que representan a cada entidad en la página. El sitio hace escaso uso de imágenes y/o animaciones y las presentes no son “pesadas” en general, no es el caso de galería de fotos (5.3Mb)

Con respecto al idioma, no se presenta la posibilidad de cambiarlo o cambiar los formatos en los que se presenta la información, con respecto a la adecuación a distintas conexiones el sitio en general es bastante “liviano” y puede ser accedido fácilmente, Excepto por la galería de fotos. Al no hacer uso intensivo de imágenes y animaciones el sitio se presenta bien sin imágenes, los textos alternativos son poco explicativos. El sitio en general no provee características de adaptación. Los tiempos de carga de las páginas rondan los 2 segundos, excluyendo la principal 10 segundos (489Kb) y la galería de fotos 100 segundos (5.3Mb). el servidor que lo aloja aborta conexiones y produce que las páginas se vean incompletas.

Principios de diseño a nivel de Contenido

A nivel de página inicial, el sitio presenta problemas relacionados con proveer información general de un sitio. Como se puede observar en la figura 5.1, la página inicial especifíca vagamente lo que se puede hacer en el sitio, el usuario debe descubrirlo navegando. No provee un índice de los temas/funciones que se pueden encontrar en el sitio.

También en la definición del aspecto y diagramación inicial. La página inicial respeta en general la diagramación del sitio, pero esto difiere mucho al buscar en la sección catálogos en la cual el aspecto es totalmente distinto. El marco, menús y demás elementos comunes que se visualizan en la página principal, se visualizan en todas las secciones menos cuando se está buscando en los catálogos, lo que da la impresión de haber salido del sitio.

Respecto a información sobre el estado del sitio, la página inicial no informa sobre la fecha de última actualización ni sobre el estado del sitio, paginas en construcción, etc. Así como tampoco define areas prioritarias.

Respecto a proveer servicios adicionales, el mapa del sitio provisto es poco intuitivo (figura 5.9), la sección de ayudas y preguntas frecuentes está vacía. No se provee un tour ni asistentes interactivos

Figura 5.9 (Mapa del sitio): Es poco intuitivo

Los formularios identifican en color rojos los campos obligatorios, con una clara distinción visual. Sin embargo presentan varios problemas de usabilidad ya que permiten un orden de llenado arbitrario, pero si no se completa un campo requerido y se guarda, muestra un mensaje de error y se debe volver a completar todo el formulario.. El sistema, en general, no provee validación de entradas ni ayudas. El único formulario que las provee es el de contacto. Esto atenta contra la flexibilidad.

El sitio no provee funciones sintácticas específicas de los formularios, no permite guardar y reanudar, suspender y si se comete un error hay que volver a completar todos los datos. El único formulario que provee funciones sintácticas es el de contacto. Las únicas ayudas como puede ser imprimir o cambiar el tipo de fuente son provistas por el marco.

Se permite la navegación por los campos pero luego de pasar por todos los otros elementos de la página, al obtener el foco estos elementos no producen feedback, así, hasta que no se está posicionado sobre un campo del formulario, el usuario no sabe dónde se encuentra el foco. Es decir, no brinda medios alternativos de activación de los casilleros.

El sitio no provee ningún tipo de validación en los campos del formulario, lo que vuelve engorroso su uso. El único formulario que provee funciones sintácticas es el de contacto.

El sitio solo provee feedback cuando se envía en formulario que contiene errores, los mensajes aclaratorios son vagos, no se provee feedback mientras se completa el formulario. El único formulario que provee feedback es el de contacto.

Los mensajes de error son vagos y aparecen cuando el usuario ya envío el formulario, y lo obligan a tener que re completar todo el mismo. En el caso del formulario de contacto los mensaje de error aparecen en color rojo y de manera intimidatoria (Figura 5.10)

Figura 5.10 (Formulario-contacto): Los mensajes de error aparecen en rojo, son intimidatorios y vagos a la vez

Principios de diseño general de contenido y del lenguaje empleado

Este principio no presentó mayores dificultades dado que, en general, la organización del contenido es correcta. El contenido que se provee respeta las normas y es coherente con su título y/o hipervínculo. Teniendo el sitio tan poca información, se respeta el nivel de abstracción del contenido. Hay páginas que no proveen contenido, como ayudas y noticias. El sitio respeta la relación entre especificidad de contenido y pasos de navegación. Igual que en los puntos anteriores los excesivos menús dificultan su uso.

La gramática es correcta, la forma de expresión es simple. Se aprovechan recursos simples como párrafos, tabulaciones y tablas. No se utilizan recursos como tab-sets o layers para desplegar información y sintetizar espacio. El formato de la visualización es estándar, respeta el tipo de la fuente y los tamaños aproximados.

Respecto a los objetos de interacción, los menús están bien diseñados, respetan las normas de profundidad y contenido, lo que dificulta la navegación es que hay 3 barras de menús dos de las cuales tienen prácticamente las mismas opciones. Ninguna de estas barras provee feedback. Solo la central para el elemento seleccionado. Respecto a las tablas y vistas, no se visualiza el estado de las tablas y las escasa funciones sintácticas provistas carecen de funcionamiento.

En general, la terminología y el lenguaje empleado a lo largo del sitio son apropiados y consistentes y no se detectan errores de ortografía y/o gramaticales.

Conclusiones

Al finalizar esta evaluación queda en evidencia que no seguir las recomendaciones básicas para el diseño de un sitio web conlleva directamente a reducir su usabilidad, provocando una mala experiencia en los usuarios, esto produce lo abandonen pronto o incluso que directamente dejen de usarlo.

Como corolario se puede extraer que antes de poner un sitio web en producción, se debe evaluarlo, testearlo y corregirlo hasta que cumpla con las normas que se pretende cumplir y su funcionamiento cumpla los objetivos especificados.

En lo personal este trabajo nos ha mostrado muchos errores que se cometen con frecuencia al desarrollar un sitio web y como estos reducen su usabilidad, posibilitándonos mejorar a la hora de enfrentar nuevos proyectos.

Ejercitación

A. Citar al menos cinco diferencias entre diseñar la interfaz de un sitio Web y la interfaz WYSIWYG (What You See Is What You Get).

_____

_____

_____

_____

_____

B. Responder verdadero o falso.

- El tipo de información que se publica en un sitio Web no afecta al diseño de la interfaz del mismo. V F
- Al diseñar un sitio Web hay que pensar en un perfil de usuario general V F
- Es recomendable enriquecer el sitio Web con presentaciones Flash muy coloridas y animadas. V F
- Realizar testeos empíricos del sitio Web forma parte de la Ingeniería de Usabilidad V F
- Es fundamental facilitar el rediseño de los sitios Web V F
- Una vez publicado el sitio no es necesario tomar muestras de la interacción del usuario con el mismo. V F

C. Unir con flechas la actividad de la izquierda con las etapas previas al diseño del sitio Web de la derecha.

Actividades	Etapas
Analizar sitios similares	Determinar la Comunidad Destino
Comprobación y corroboración de cuestiones de usabilidad.	
Determinar las actividades que llevarán a cabo los usuarios	
Recolección de feedback de usuarios	Planificar el Sitio
Considerar que los usuarios pueden acceder desde diferentes lugares, distintos medios y mediante diferentes recursos tecnológicos.	
Estudiar el grado de experiencia del usuario con la Web.	
Establecer mediciones de cuando se considera un sitio exitoso y cuando no.	Prototipación
Atender las sugerencias de los usuarios	

Referencias

❒Tognazinni, B. First Principles.
❒Nielsen, Jakob (1994). Usability Engineering.
❒Nielsen, Jakob. (2000). Designing Web Usability: The Practice of Simplicity.
❒Nielsen, Jakob; Mack, Robert (1994). Usability Inspection Methods.
❒Nielsen, Jakob; Loranger, Hoa (2006). New Riders Press.
❒Niederst, J.; Gustafson, Aaron (2007). Learning Web Design: A Beginner's Guide to (X)HTML, StyleSheets, and Web Graphics.
❒Lopuck, Lisa (2006). Web Design For Dummies.
❒Niederst, J.; Gustafson, Aaron (2006). Web Design in a Nutshell: A Desktop Quick Reference.
❒Smith, Jim (2007). How to Start a Home-Based Web Design Business.
❒Jul, Susanne; Furnas, George. Navigation in Electronic World.
❒Devi, Michael; Conrad, Frederik. Usability Testing of World Wide Web Sites.
❒Bachiochi, D.;Berstene, M. Usability Studies and Designing Navigational Aids for the Web.
❒http://www.w3c.es
❒http://www.useit.com
❒http://www.jnd.org
❒http://www.cs.umd.edu/~ben/
❒http://www.asktog.com/basics/firstPrinciples.html
❒http://usableweb.com/
❒http://msdn.microsoft.com/en-us/library/ms537341(VS.85).aspx
❒https://bugzilla.mozilla.org/show_bug.cgi?id=366611
❒http://www.linti.unlp.edu.ar/tiki-page9419.html?pageName=pubRec-2006
❒http://www.useit.com/alertbox/20041025.html
❒http://www.useit.com/alertbox/20001126.html
❒https://www.atccapitalmarkets.com/securitisation-services/security-trustee
❒http://www.w3c.es/divulgacion/guiasbreves/internacionalizacion
❒http://www.w3.org/International/
❒TRUSTe http://www.truste.org/
❒Better Business Bureau http://www.bbb.org/
❒http://www.roirevolution.com/blog/2008/05/time_on_page_and_time_on_site_how_confident_are_yo.html
❒http://www.google.com/analytics

06

Accesibilidad en la Web

Realizar un sitio que pueda funcionar bajo "condiciones óptimas" únicamente, implica excluir millones de personas que no podrán accederlo, y que tienen el derecho de ser usuarios del mismo.
Internet debe ser para "todos", donde la palabra "todos" no significa sólo usuarios con las mejores herramientas, los últimos navegadores, las máquinas más rápidas, y las mejores condiciones físicas y mentales.

Temas a Tratar

I- Introducción
II- Accesibilidad en la Interfaz del Usuario
- Consideración de distintas discapacidades
- Análisis de herramientas para discapacitados

III- Accesibilidad en la Web: su problemática
- Falta de Diseño Universal en la Web
- Falta de Cumplimiento de las Normas de Accesibilidad
- Problemáticas con los Recursos de Adaptación

IV- Legislación y Normas Internacionales sobre Accesibilidad
V- Normas de Accesibilidad en la Web
- Principios generales de Accesibilidad
- Recomendaciones de Accesibilidad de la W3C-WAI
  - -WCAG 1.0
  - -WCAG 2.0

VI- Comprobación de la Accesibilidad Web
Casos de Estudio
Ejercitación
Referencias

I - Introducción

Internet, considerada como la red mundial de información sin fronteras, que revoluciona el área de las comunicaciones y que permite un acercamiento cada vez mas estrecho entre las diferentes culturas y razas del mundo, es considerada para muchos otras personas, como una nueva barrera a afrontar.

Personas que, por su condición socio-económica, por discapacidades que posean u otras limitaciones propias del ser humano, no cumplen con las condiciones y requisitos necesarios que demandan la mayoría de los sitios Web, quedando excluidas de este importante avance tecnológico.

Por ejemplo, analicemos el caso de las personas con discapacidad visual. Según datos estadísticos proporcionados por la OMS –Organización Mundial de la Salud-, se estima que hay acerca de 85 millones de personas ciegas y débiles visuales en el mundo.

Ahora bien, estas personas, que les resultaría sumamente útil y provechoso poder comunicarse, hacer trámites, estudiar, investigar a través de Internet, ¿realmente, pueden hacerlo?

Los desarrolladores Web, ¿consideran a las personas con ciertas discapacidades visuales, físicas o mentales, como posibles usuarios de su sitio?. Si Internet debe ser para "todos", ¿ellos efectivamente trabajan en un diseño universal?

La respuesta lamentablemente es, no. La mayoría de los sitios presentan un diseño únicamente visual, con elementos llamativos, uso de animaciones y presentaciones en Flash, que en el caso de los disminuidos visuales y ciegos, les puede resultar imposible de acceder aunque estén dotados de herramientas auxiliares de adaptación.

Debido a esto, surge el concepto de "Accesibilidad en la Web", que significa que todo sitio debe ser construido y diseñado de manera tal, que pueda ser accedido por cualquier persona, independientemente de su condición física, mental, o recursos tecnológicos que posea.

Esto implica contemplar los requisitos de todos los posibles usuarios desde las primeras fases de los diseños de productos, de manera que las personas más excluidas, generalmente, aquellas de bajos recursos, gente de mayor edad, o minusválidos, se conviertan en una parte importante de los posibles clientes.

Organizaciones internacionales como la WAI, (Web Accesibility Initiative), perteneciente a la W3C (World Wide Web Consortium), se ocupan específicamente de establecer y publicar normas de accesibilidad, para garantizar el diseño universal del sitio y el uso de elementos estándares.

Además, recomiendan sobre la utilización de herramientas de diseño y promueven sistemas de validación, para que los desarrolladores Web puedan evaluar el nivel de accesibilidad de su sitio. En algunos casos, se les detecta errores de diseño y se les presenta alternativas de solución.

En este capitulo, específicamente se va a tratar el tema de la accesibilidad en la Web teniendo en cuenta las diferentes discapacidades que un usuario puede padecer. Para ello, en primer instancia, se va a analizar lo que significa una interfaz del usuario accesible y cuáles son sus alcances. Luego, se va a profundizar sobre los problemas que presenta Internet respecto a las cuestiones de accesibilidad y por último, se van a listar las normas más importantes que hacen al diseño accesible de un sitio.

II - Accesibilidad en la Interfaz del Usuario

El concepto de accesibilidad en Informática puede ser muy amplio y tener muchas aserciones, porque puede aplicarse tanto al soporte físico o sea al hardware de una computadora, como al soporte lógico o software.

Entonces, elementos muy heterogéneos entre sí pueden ser accesibles, dispositivos de hardware como un teclado, una impresora, un ratón, como también diferentes tipos de software que pueden ir desde plataformas o sistemas operativos, herramientas de autor, administradores de contenido o CMS hasta aplicaciones de gran envergadura o páginas Web.

Aunque en todos los casos, la accesibilidad hace referencia a proveer un diseño universal en los productos que se estén desarrollando, para que pueda ser entendido y utilizado por cualquier clase de persona, puede existir sutiles diferencias en sus definiciones y ser tratadas mediante normas de accesibilidad específicas, de acuerdo al tipo de elemento que se esté tratando.

Entonces, si profundizamos sobre el área de interacción hombre-máquina y aplicamos la definición de accesibilidad específicamente a la componente de la interfaz del usuario, se puede abordar a la siguiente concepción:

"Una interfaz del usuario es accesible si para su construcción, diseño e implementación, se han tenido en cuenta los requisitos necesarios, para que pueda ser utilizada por la mayor cantidad de personas posibles. Esto es, independientemente de cuestiones de discapacidad, barreras del lenguaje, recursos tecnológicos, o cualquier otro obstáculo posible".

Como en este capítulo, se va a abordar específicamente la problemática del discapacitado y su interacción con la Web, entonces, situados en este contexto particular, se puede ahondar aún más en la definición, y establecer que "Una interfaz accesible para la Web, es aquella que permite una interacción y representación del contenido del sitio, diseñados de tal manera que pueden ser adquiridas, percibidas y operadas por cualquier persona que padezcan algún tipo de discapacidad, aunque sea mediante herramientas especiales".

Ahora bien, una vez que individualizamos y entendemos el concepto al que debemos respetar, pueden surgir innumerables inquietudes y cuestionamientos sobre ¿cómo alcanzar esta definición?, ¿cómo lograr realizar una interfaz con estas características?, ¿cuál es realmente el nuevo escenario que hay que considerar, donde se contemple al usuario discapacitado con sus problemáticas?, ¿cuáles serán las normas o principios de diseño específicos que hay que analizar y aplicar en el futuro producto?, entre otros.

A lo largo de todo este capítulo, se va a intentar responder todos estos interrogantes, pero en principio, lo que se debe hacer desde el equipo de desarrollo del sitio, es adquirir conciencia y asumir que dentro de la comunidad de usuarios destino, se deberán incluir a personas que pueden no poseer todas las capacidades psíquicas, físicas o funcionales, en condiciones óptimas, o que no posean los recursos tecnológicos ideales para acceder al sitio. O sea, deben dejar de considerar al usuario potencial de su producto, únicamente como aquel con facultades y circunstancias ideales.

Para ello, en la etapa de modelización, se debe incorporar un proceso de análisis de distintos perfiles de usuarios discapacitados, en donde se estudie su problemática específica, y donde también se indague sobre los recursos y herramientas de adaptación que ellos pueden utilizar en su ordenador para poder interactuar con las aplicaciones.

Por lo tanto, se pueden distinguir dos actividades fundamentales que se adicionan, desde las etapas iniciales de la interfaz accesible, que son:

- Consideración e investigación de las distintas discapacidades, y
- Conocimiento y análisis de las herramientas de adaptación que ellos usualmente pueden emplear en pos de comunicarse con el ordenador.

Además, como en todo desarrollo de un sitio Web, en donde se debe probar el sitio frente a distintos recursos, tipos de monitor, diferentes navegadores, comprobando su buen funcionamiento, ahora hay que extender y profundizar más este proceso de evaluación, considerando e incluyendo las distintas soluciones informáticas que existen de adaptación, tanto de hardware como de software.

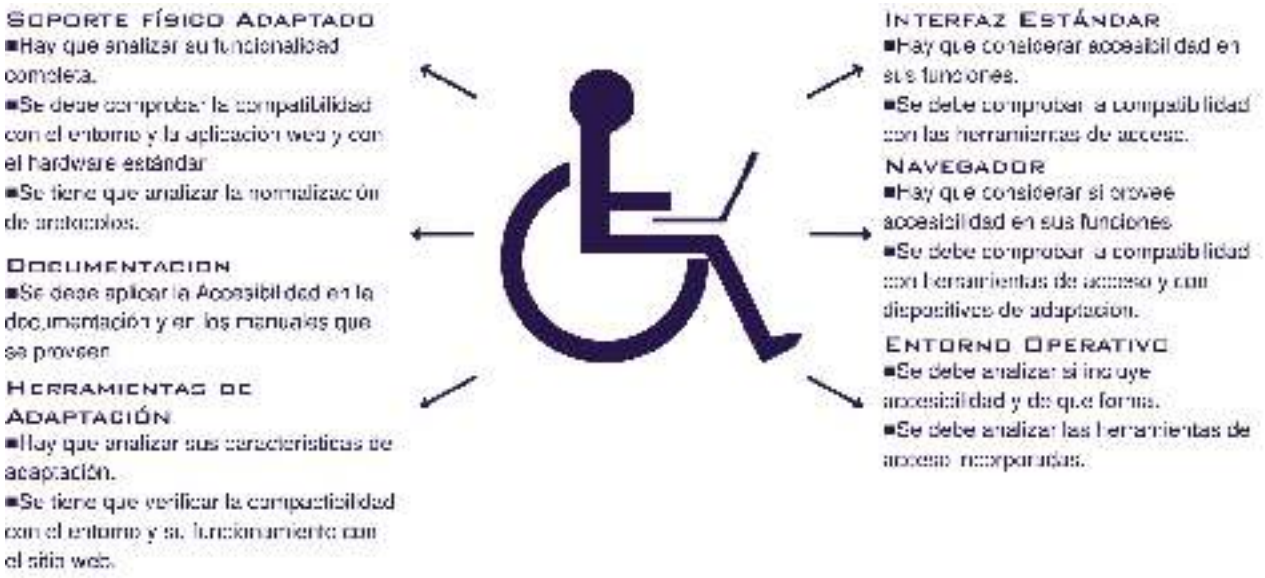


Figura 6.1: Soluciones informáticas de adaptación.

Entonces, el desarrollador Web, deberá afrontarse a una mayor complejidad y diversificación tanto en el soporte físico como lógico de una computadora para uso del discapacitado, teniendo en cuenta este panorama de acción y todas sus consideraciones.

Consideración de las Distintas Discapacidades

Como se dijo anteriormente, una interfaz Web accesible es aquella que debe presentar su contenido y proveer un comportamiento tal, para que pueda ser percibida, entendida y utilizada adecuadamente, por personas con diferentes condiciones y capacidades. La misma debe reconoce diferentes clases de interlocutores, y debe amoldarse a sus necesidades y formas de interacción.

La personas con discapacidad, podrán entonces acceder al sitio accesible pudiendo captar mediante

diferentes modos, la información que allí se publique y haciendo uso de la funcionalidad provista.

Hay que comprender que ese sitio además de convertirse en su herramienta de trabajo, constituirá un elemento fundamental en el proceso de su integración social.

Las discapacidades que un usuario puede padecer son muy variadas y cada una determinan problemáticas de acceso y de interacción muy específicas.

En la siguiente tabla, se detallarán algunas de las diferentes discapacidades que un ser humano puede adolecer y que un sitio Web deberá contemplar afectando su diseño, y se considerarán cuáles son sus realidades o complejidades respecto a lo que al uso de Internet se refiere:

Tipos de Discapacidades Humanas	Problemática respecto al Uso de Internet
Discapacidad Visual	■Están incluidos los usuarios minusválidos visuales, aquellos que posean cierta afección visual como daltonismo y los ciegos. Por el diseño visual de la Web, son los más afectados en el uso de Internet y los más excluidos. ■Muchas personas ciegas utilizan herramientas que facilitan una descripción hablada o convertidas a Braille de las páginas, pero no todos los sitios permiten que estas herramientas funcionen correctamente, ni tampoco proveen ellos mismos los medios auditivos necesarios. ■Las personas con problemas de visión, que no son ciegas, necesitan aumentar el tamaño, el contraste o las características generales de visibilidad en función de sus necesidades visuales, pero muy pocos sitios proveen estas funcionalidades de adaptación, ni tampoco están preparados para que el discapacitado pueda hacerlo desde mecanismos propios.
Discapacidad Auditiva	■Están incluidas las personas sordas, sordomudas y con dificultades auditivas. ■Estas personas son las que menos problemas tienen al acceder a Internet, a menos que se utilicen mecanismos de interacción exclusivamente auditivos. ■Las personas con dificultades auditivas que no alcanzan la sordera tienen problemas con los cambios y determinados rangos de frecuencia y para localizar y distinguir determinados sonidos. ■Ellos como los sordos, necesitan que los videos, audio clips, mensajes auditivos que ofrezca el sitio, sean subtitulados o provean una versión alternativa en texto. ■Además de tener problemas para detectar informaciones auditivas, los usuarios sordos pueden no ser capaces de hablar correctamente. Para ellos y en el caso directo de los sordomudos, el uso de reconocimiento de voz, deberá ser analizado especialmente.
Discapacidad Motriz	■Las dificultades de las personas con problemas físicos suelen ser derivados de su falta de coordinación, su debilidad, la dificultad para alcanzar las cosas o la imposibilidad de mover alguna o algunas extremidades. ■Estos usuarios pueden tener problemas con el uso del mouse, rapidez requerida para realizar una determinada acción, presión simultánea de teclas. ■Se puede presentar casos donde la única manera que tienen para operar una PC, es mediante un único dedo, mouse de pie, licornios.
Discapacidad Cognitiva	■Aquí se incluyen los discapacitados mentales, psicológicos como gente depresiva, como también con problemas de adicción como drogadictos. ■Ellos presentan diferentes capacidades de razonamiento, memorización, comprensión y distintos grados de rapidez mental. ■Les puede costar la realización de tareas complejas, anidadas, transacciones de varios pasos, como también la interpretación de mucho caudal de información en un instante determinado de tiempo ■Aspectos visuales como animaciones, objetos parpadeantes pueden afectarles, produciéndoles ciertas perturbaciones no deseables.

Tabla 6.1: Tipos de discapacidades y sus problemáticas.

Análisis de las herramientas para discapacitados

Los usuarios discapacitados necesitan contar con herramientas tanto de hardware como de software que, de acuerdo al tipo de afección que posean, le permitan suplir las facultades que tienen condicionadas y los ayuden a llevar a cabo sus intenciones frente al sitio Web.

Estos elementos auxiliares de hardware o software que se encuentran en el mercado para asistir al discapacitado, se los conocen comúnmente bajo los siguientes nombres: herramientas informáticas de adaptación, adaptaciones informáticas, o ayudas técnicas.

Estas herramientas presentan características y formas de uso muy diferentes entre sí. Determinan una serie de complejidades propias del uso de herramientas adicionales. En el caso del software de adaptación, el mismo debe instalarse en el ordenador y funcionar correctamente, mientras que en el caso del soporte físico de adaptación, además de instalarlo y configurarlo, también hay que comprobar la correcta conexión al mismo.

Respecto al software de adaptación, el mismo se puede clasificar en las siguientes categorías:

- Los Revisores de Pantallas: trabajan directamente sobre la salida del monitor. Se componen de:
  - Los Ampliadores de Imágenes
  - Los Lectores de Pantalla.
- Los Revisores de Documentos: trabajan directamente sobre los archivos de datos. Abarcan a:
  - Los Lectores de Documentos,
  - Los Reconocedores de Caracteres (OCR), y
  - Los Navegadores de Internet Especializados
- Los Tomadores de Notas
- Los Software de Presentación
- Software especiales de sonido

En la siguiente tabla, se explicarán los sistemas de software de adaptación más importantes a considerar en el desarrollo de un sitio accesible:

Herramientas de software de adaptación	Explicación
Ampliador de imágenes	■Es una herramienta de tipo revisor de pantalla, ideal para discapacitados visuales. ■Es un software que permite capturar la salida al monitor y dar diferentes grados de ampliación y zooming. ■Se tiene al ZoomText de la compañía estadounidense Ai Squared, Supernova de Dolphin Computer Access, que incluye capacidades adicionales de lectura de documentos.
Lector de pantalla	■Es muy utilizado por usuarios ciegos, puesto que es un software que reconoce y describe oralmente la presencia de distintas componentes de interacción, como ser texto, cuadros de edición, botones, menús, lista de opciones, entre otros. ■Este programa sirve de interfaz entre la tarjeta de vídeo, los sistemas de síntesis de voz y/o los terminales de lectura Braille, y el usuario de la computadora. ■Los lectores de pantalla actuales más populares son el Hal de Dolphin, el JAWS del grupo Freedom Scientific, el Orca y NVDA de software libre. También, se encuentra el pequeño lector de pantalla Narrator de Microsoft incorporado en Windows XP.
Lector de documentos	■Es un programa que tiene como finalidad la transmisión por voz del contenido de archivos de texto, y que suelen ir acompañados de otras herramientas como deletreo de palabras. ■Lo integran aquellos programas llamados de texto hablado o Text-to-Speech, como puede ser el TextAssist o el Monologe de First Byte. También está el TextAloud de NextUp Technologies, que permite el guardado de archivos de sonido.
Reconocedor de caracteres	■Tiene la capacidad de digitalizar páginas de texto. Esta funcionalidad denominada OCR (Optical Character Recognition), es lo que utilizan los ciegos para acceder a información escrita en papel, mediante el ordenador.

	■Ejemplos son OmniPage de Caere, TextBridge de Xerox, y el popular Open Book de la empresa norteamericana Arkenstone.
Navegador especializado de Internet	■Estos programas suelen incluir voz y ampliación de imagen, facilitan en gran medida la lectura correcta del contenido Web, acceso a hipervínculos y demás componentes de una página. ■Un ejemplo es el navegador pwWebSpeak de la sociedad norteamericana The Productivity Works. ■IBM, realizó el producto Home Page Reader, que se lo incorpora a un navegador convencional, para proveer la adaptación.
Tomador de notas	■Se tratan de pequeños ordenadores portátiles diseñados especialmente para discapacitados visuales, para asistirlos en la escritura de distintos tipos de notas.
Software de presentación	■Por ejemplo, se tiene el software de presentación del teclado en pantalla, donde es totalmente configurable para adaptarse a las necesidades del usuario. ■Estos pueden operarse desde el ratón o mediante un pulsador.
Software especiales de sonido	■Para las personas sordas, que necesitan tener información visual o textual de sonidos que utiliza el ordenador. ■Por ejemplo, la función “ShowSounds” que proveen algunos sistemas operativos.

Tabla 6.2: Herramientas Informáticas de Adaptación.

Entre los productos de hardware, se pueden considerar:

Hardware de Adaptación	Explicación
Sintetizador de voz	■Es un dispositivo para equipar la PC, el cuál permite convertir texto en lenguaje hablado. ■Ofrece diferentes idiomas y distintas formas de interlocución.
Magnificadores de imágenes para TV y monitores de PC	■Son adaptadores destinados para personas que tengan disminución visual, pues se encargan de agrandar los caracteres, imágenes y demás contenidos de una pantalla. ■Son aparatos que toman la salida del televisor o monitor y las transforma, generalmente aumentando la imagen transmitida. ■Los primeros equipos de ampliación que aparecieron, como el Vista de Telesensory Systems, se componían de una tarjeta de interfaz que se insertaba dentro del ordenador, a la cual se conectaba internamente la tarjeta de vídeo y externamente el monitor, y que hacía de puente lupa entre una y otro.
Equipos de multifunciones	■Trae incorporados varias funcionalidades de adaptación en forma integrada. Como mínimo incluye sintetizador de voz y amplificador de pantalla.
Sistema de lectura	■Es un aparato que se comporta como una fotocopiadora parlante.
Graficadores de relieve	■Son dispositivos que permiten preparar y generar gráficos táctiles.
Reconocedores ópticos Braille	■Tienen la capacidad de detectar y leer código Braille. ■Pueden realizar conversiones de código Braille a formato digital, o a lenguaje hablado, etc.
Dispositivos de entrada especiales	■Como ser teclado Braille, emuladores de teclado y mouse. ■Teclados ergonómicos para una sola mano. ■Como licornios, mouse operados con el pie, para el mentón.
Otros insumos especiales para no videntes	■Son elementos especialmente diseñados para no videntes, como por ejemplo termómetros parlantes, lentes sonares, los cuales emiten un sonido particular al detectar un obstáculo.
Dispositivos para el Seguimiento de ojos	■Muy utilizados por personas con dificultades motrices, que no pueden usar el mouse. ■También se encuentran para ellos, los punteros alternativos licornios, punteros de manos, entre otros.

Tabla 5.3: Productos de Hardware de Adaptación.

III-Accesibilidad en Internet: su problemática

En la sociedad de la información en que nos hallamos inmersos, todas las personas deberían tener la posibilidad de acceder a la información que nos ofrece Internet, sin que los usuarios con discapacidad sean una excepción.

Para ellos, este medio resulta muy interesante, ya que permite el acceso a diferentes recursos como catálogos o bases de datos, tanto desde su propio domicilio como desde bibliotecas o centros de documentación.

Pero lamentablemente, la situación actual de Internet hace muy notoria la brecha digital entre los usuarios que pueden acceder a la información digital y los que no, haciendo aún mayor las dificultades que las personas con algún impedimento tienen que sortear, para poder acceder a esta herramienta tan poderosa.

Los problemas más graves que presenta Internet respecto a lo que Accesibilidad se refiere y que se van a tratar en esta sección, son:

- Falta de diseño universal en la Web,
- Incumplimiento de las normas de Accesibilidad, y
- Problemas con las Herramientas de Adaptación.

Falta de Diseño Universal en la Web

Cuando el discapacitado interactúa con sitios Web, se encuentra con un diseño general que es predominantemente visual, y que en la mayoría de las discapacidades que un ser humano pueda padecer, perjudica y dificulta aún más la interacción con el mismo.

Desde su concepción, le es inherente a la Web, la utilización de elementos llamativos, con páginas que incluyen animaciones, videos, elementos de control, marcos, tablas, sonido, texto parpadeante, imágenes de fondo, y otras componentes que pueden poner en riesgo la accesibilidad del sitio.

Esto hace referencia, a que por ejemplo, las animaciones y los elementos titilantes no son adecuados para algunas discapacidades mentales, o utilizar ciertas combinaciones de colores y contrastes no son aptos para personas daltónicas o con disminución visual.

Lo que se sugiere no es abolir los recursos visuales sino que estos no sean los únicos medios de comunicación provistos y que al menos vengan acompañados por texto para su descripción y explicación.

Se menciona al texto como una alternativa estándar, puesto que muchas herramientas informáticas de adaptación proveen funcionalidad para convertir de texto a otros medios de comunicación como sonido o sistema Braille.

Como se verá más adelante, las normas de accesibilidad son recomendaciones que hacen mucho hincapié en la utilización de medios alternativos de comunicación y de representación de la información, como también, en el uso del texto como una alternativa fundamental que debe estar siempre vigente.

Hasta el momento, la tendencia general es que el desarrollador Web apunte más a proveer un mayor impacto visual y poner en la interfaz la mejor artillería de avanzada que el lograr mejorar la audiencia, ampliar la comunidad de usuarios destinataria y poder agradar a todos.

Pero esta conducta está cambiando o deberá revertirse en poco tiempo. No se puede diseñar para la Web con restricciones sino bajo un pensamiento universal que unifique a todos.

Además, hay que tener bien en claro que uno no se puede apartar de la filosofía de la Web, que es justamente compartir, publicar para todos, estar conectados globalmente, más allá del tiempo y del espacio.

Y, lo que se debe incluir en esa filosofía es que debe ser más allá de las condiciones físicas o mentales que tengamos.

Que “Internet sea para Todos” es una afirmación que depende fundamentalmente del desarrollador Web.

Incumplimiento de las Normas de Accesibilidad

Otro problema que existe es que, aunque las páginas presenten una preponderancia del diseño visual, si al menos cumplieran con ciertas normas de accesibilidad, se podría asegurar que el discapacitado pueda accederla e interactuar con ella utilizando ciertas herramientas de adaptación.

Por lo general las páginas no están normalizadas, no respetan las distintas pautas de estandarización y accesibilidad establecidas.

Profesionales y expertos de todo el mundo conformaron consorcios y organizaciones internacionales, la mayoría en Estados Unidos y Canadá, las cuáles se ocupan de establecer y publicar normas de accesibilidad, en pos de preparar y orientar a los desarrolladores Web en una cultura universal.

La más importante es la organización WAI, de la W3C Consortium, que orienta y estructura el desarrollo global de la WWW centrándose en el desarrollo tecnológico, presenta recomendaciones y pautas para la creación de páginas accesibles.

También trabajan en este campo, EASI (Equal Access to Software and Information), WeABLE, DOIT, NCAM (National Center for Accesible Media), ATRC (Adaptive Technology Resource Centre de la Universidad de Toronto), Include (Finlandia dependiente de la Unión Europea), Microsoft Enable, Starling Access Services (Canadá), entre otros.

A pesar que hubo avances e iniciativas en esta área desde la década del 90', la existencia de estas normas no es ninguna garantía de que los desarrolladores de hardware y software la respeten. Existe una falta de interés, de conocimiento y de aplicación de normas de accesibilidad por parte de los diseñadores Web.

Entre los diversos factores que llevan a esta situación de incumplimiento, se puede mencionar:

Causas posibles de incumplimiento de normas de accesibilidad	Explicación
Falta de legislación	■Aún faltan muchos paises que cuenten con alguna legislación vigente la cuál se ocupe o haga referencia al tema de la accesibilidad en plataformas y sistemas informáticos. ■Es sumamente necesario y más aún, en Internet, que se cuente con normativas legales, para exigir a los desarrolladores a proveer alternativas de diseño de su sitio, que sean accesibles. ■En la sección 4 de este Capítulo se explicará con más detalle cuestiones de legislación.
Desinterés	■Los diseñadores y programadores Web, en su afán de lograr sitios cada vez más espectaculares, pasan por alto la problemática que presentan muchas discapacidades humanas y sus respectivas normas de accesibilidad, incurriendo en un acto de discriminación.
Problemas de Difusión	■Los organismos encargados de establecer las distintas normas y pautas de accesibilidad, carecen en cierto sentido, de la capacidad de dar a conocer el valioso trabajo que están realizando. ■Los estándares permanecen generalmente ocultos a la mayoría de los programadores.
Prejuicios Infundados	■La mayoría de los diseñadores Web, suponen que una página adaptada a los diferentes tipos de discapacidades que pueda presentar un usuario puede quitar estética a la misma y ocasionar su futuro fracaso. ■Pero tales preconceptos son totalmente infundados puesto que un sitio accesible permite el acceso a una mayor cantidad de usuarios.

Tabla 6.4: Causas posibles al Incumplimiento de Normas de Accesibilidad.

Debido a estas y otras posibles causas, resulta complejo incorporar de manera natural y universal los estándares de accesibilidad, por ello la mayoría de los sitios Web y páginas se ven posicionados en un lugar de difícil acceso por parte de los discapacitados.

Problemáticas con herramientas de adaptación

Otro punto que el diseñador Web por lo general omite en su proceso de desarrollo, es el análisis de la problemática del discapacitado y del estudio de los diferentes recursos que el minusválido puede utilizar en pos de adaptar su PC para poder acceder a Internet.

El desarrollador debe tener en cuenta que pueden existir miles de personas que utilizan herramientas de adaptación, por lo que él debería en primer instancia conocer esta situación y luego, evaluar el comportamiento de su sitio frente a las mismas.

Estas personas discapacitadas, además del tipo de afección o limitación que tienen que padecer, deben contar con las herramientas de adaptación adecuadas y lidiar con los problemas que éstas pueden acarrear, al integrarlas y utilizarlas desde cada sitio, simplemente por el mero hecho de querer acceder a Internet.

Con datos recopilados directamente de los mismos usuarios afectados, como en forma indirecta, incursionando entre las listas de discusión y preguntas frecuentes, se puede sintetizar la problemática que ellos afrontan respecto al uso de adaptaciones informáticas, de la siguiente manera:

Problemas con las Herramientas de Adaptación	Explicación
Problemas de Incompatibilidad	■El usuario discapacitado a veces debe sufrir complicaciones de incompatibilidad entre las distintas herramientas que él instala en su ordenador. ■También, la herramienta de adaptación puede presentar incompatibilidad con ciertas funciones del sistema operativo o con el navegador subyacente, generando conflictos con los programas ya instalados.
Generación de Versiones más avanzadas de un sitio	■A medida que la tecnología avanza, proporcionalmente se desarrollan versiones de sitios más complejos, que incluyen componentes FLASH, estilos, applets, layers. ■Muchas veces, las herramientas que posee el discapacitado queda obsoleta, teniendo que adquirir y aprender otras herramientas o nuevas versiones de la misma.
Manejo Heterogéneo de los periféricos	■El usuario discapacitado en su interacción con sus diferentes herramientas de adaptación instaladas, más el sitio que está accediendo, debe recordar qué teclas usar, si se le permite o no usar el teclado para acceder a los hipervínculos, qué dispositivo emplear en un determinado momento. ■Esto se debe a que interactúa con distintos tipos de softwares desarrollados por diferentes personas o empresas.
Falta de ayudas o tutoriales accesibles	■Paradójicamente, hay herramientas de adaptación que tienen escasa ayuda, que es totalmente insuficiente y que además no son accesibles. ■Lo mismo ocurre con los sitios Web que por lo general no incluyen un buen sistema de ayuda, y que si lo hacen, tampoco presentan características accesibles.
Problemas de Portabilidad	■Un discapacitado puede contar con las mejores herramientas auxiliares en su hogar, las cuáles pueden mejorar su calidad de vida. Pero, ¿qué pasa cuando esta persona intenta salir al mundo exterior?. Los lugares de concurrencia masiva no están preparados en este sentido.

Tabla 6.5: Problemas con las Herramientas de Adaptación.

Estos problemas que presentan muchas de las herramientas auxiliares pueden llegar a ser complicados hasta para usuarios no discapacitados. La gravedad es aún mayor, al tratarse de herramientas informáticas destinadas a personas minusválidas, estos problemas se transforman en obstáculos que son difíciles de enfrentar.

Esta es una realidad que está en manos del desarrollador Web en revertirla aunque sea incluyendo ya en el sitio algunos servicios de adaptación para ellos. Por ejemplo, si el sitio acompaña cada texto con mensajes hablados, si para cada imagen hay una explicación textual y oral de la misma, si la letra se puede ampliar, si los colores se pueden configurar, si los videos están subtitulados, entonces personas ciegas, sordas o con otras problemáticas podrían acceder sin complicaciones y sin necesidad de instalar recursos extras.

Lo que se pretende en definitiva, es que así como el diseñador está a la espera de nuevas tecnologías para incorporarla en sus futuros sitios, que también esté a la vanguardia de lo que va surgiendo en concepto de Accesibilidad, desarrollando así sitios novedosos pero que además estén aptos para discapacitados.

Importante

■Hay que tener en cuenta que un sitio Web puede manifestar su capacidad de accesibilidad de dos maneras:

-El sitio es diseñado, desarrollado considerando recomendaciones de accesibilidad y es probado teniendo en cuenta posibles herramientas de adaptación, permitiendo el buen funcionamiento e integración de las mismas.

-El sitio se desarrolla desde cero, con medios alternativos de comunicación, con funcionalidades de adaptación y configuración necesarias, para que los manejen las personas con casi todo tipo de discapacidad, sin necesidad de utilizar ningún tipo de accesorio.

IV- Legislación sobre Accesibilidad

En cuesiones de legislación y normas internacionales sobre Accesibilidad, existen varios países que ya impartieron leyes o reglamentos que colocan a la Accesibilidad en la Web como una obligación legal. Ejemplos de esto se encuentra la Sección 508 de Estados Unidos, realizada en 1986 pero modificada en los años 92' y 98', y la Ley 34/2002 de dicho país sobre servicios de la sociedad de la información y de comercio electrónico (LSSICE), que establece que las Administraciones Públicas tomen las medidas necesarias para que la información disponible en sus respectivas páginas de Internet, sean accesibles.

En Japón, se encuentra el programa e-Japan Priority Policy Program, confeccionado en el 2001, en Italia se encuentra la legislación "Stanca Act" y la BITV perteneciente a Alemania.

España se caracteriza por ser el propulsor en esta temática pues brindó la primera norma existente en todo el mundo, haciendo referencia a la creación accesible de páginas Web. La primer norma fue la española UNE 139802:1998 EX, denominada "Informática para la salud. Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad de las plataformas informáticas. Soporte lógico". Esta norma fue revisada y ampliada, dividiéndose en dos y dando lugar a las normas UNE 139802:2003 y UNE 139803:2004.

Actualmente, España cuenta con AENOR que es un organismo para la Normalización y Certificación y que actúa como el garante de la disponibilidad del documento y su estabilidad, asegurando un proceso formal de cambio.

Por otra parte, cabe aclarar que aunque no se trata de una norma en sí, la Unión Europea ha acordado tomar como norma de facto las Directrices de Accesibilidad que produce la iniciativa WAI de la W3C. Estas se explicarán en la sección Guías de Accesibilidad de la W3C-WAI del presente capítulo.

Muchas de estas normativas pertenecientes a diferentes Naciones, pueden tener concordancia. Por ejemplo, las normas de AENOR garantizan plena compatibilidad con la WCAG 1.0 de la WAI, e incluso contiene un anexo en el que se presenta la equivalencia entre los puntos de las normas españolas y los de dichas directrices.

Existe una Convención sobre los derechos de las personas con discapacidad, aprobada por las Naciones Unidas el 13 de diciembre de 2006. Fue abierta a la firma a partir del 30 de marzo de 2007, fecha en que fue firmada por Argentina y debe ser ratificada por el Congreso Nacional.

La Convención recoge la accesibilidad como uno de sus principios generales. Indica que los Estados que forman parte de la misma, se obligan a asegurar y promover el pleno ejercicio de todos los derechos humanos y las libertades fundamentales de las personas con discapacidad, sin discriminación alguna por motivos de discapacidad. A tal fin, en cuanto a la accesibilidad y participación de las personas con discapacidad en la Sociedad de la Información y del Conocimiento, se obligan específicamente a:

- Emprender o promover la investigación y el desarrollo, y promover la disponibilidad y el uso de nuevas tecnologías, incluidas las tecnologías de la información y las comunicaciones, ayudas para la movilidad, dispositivos técnicos y tecnologías de apoyo adecuadas para las personas con discapacidad, dando prioridad a las de precio asequible;
- Proporcionar información que sea accesible para las personas con discapacidad sobre ayudas a la movilidad, dispositivos técnicos y tecnologías de apoyo, incluidas nuevas tecnologías, así como otras formas de asistencia y servicios e instalaciones de apoyo;

Por otra parte, el artículo 9 está completamente dedicado a la accesibilidad en general, lo que incluye la accesibilidad en el medio físico y la accesibilidad en los sistemas de información y comunicación.

En Argentina, a fines del 2010 se aprobó por unanimidad en la Cámara de Senadores de la Nación la Ley n°26.653 sobre accesibilidad en los contenidos de páginas web, que obliga a las entidades gubernamentales o dependientes de ella a proveer sitios accesibles a todos los usuarios.

Desde el Estado Nacional argentino, a través del Decreto 378/2005 se aprobaron los lineamientos estratégicos para el Plan Nacional de Gobierno Electrónico y los Planes Sectoriales, en donde se ha impulsado notablemente el uso de las TICs -Tecnologías Informáticas y de Comunicación- aplicado a mejorar la gestión y los servicios de la administración pública nacional, garantizar la transparencia, la participación y facilitar el acceso a la información pública.

En base a esto, la ONTI, Oficina Nacional de Tecnologías de la Información de Argentina, entidad que brinda el marco de apoyo, asesoramiento y delineamientos generales, generó un documento oficial, titulado "Sitios y Portales de Internet para la Administración Pública Nacional" en donde da una serie de recomendaciones a administradores y diseñadores de sitios web públicos, tratando específicamente el tema de usabilidad y accesibilidad Web.

La ONTI además es la entidad que debe asesorar, controlar y monitorear la aplicación de la Ley n°26.653 mencionada anterioremnte.

V- Normas de Accesibilidad en la Web

En la actualidad, hay una gran variedad de documentos técnicos que abordan la problemática del acceso a la Informática por parte de personas con discapacidad y ofrecen diferentes soluciones para la accesibilidad a plataformas informáticas. Pero este tipo de documentos suelen ser internos a los propios entornos que los generan, suelen tener dificultades de difusión, sufren cambios con bastante frecuencia, crecen en demasiada e inmanejable documentación o resultan difíciles de conseguir.

En cambio, lo que se intenta realizar desde los organismos internacionales que tratan el tema de Accesibilidad, es la estipulación de normas que permiten un mayor grado de estandarización y centralización.

Una norma es un documento público al que pueden consultar todas las personas interesadas en el acceso a la última información. El organismo normalizador pertinente, es el garante de la disponibilidad del documento y su estabilidad asegurando un proceso formal de cambio.

Además, poniendo las miras un poco más lejos, si algún día se consiguiera promulgar una ley que garantizara la accesibilidad a la informática a todos los ciudadanos argentinos, resultaría conveniente tener una normativa ya desarrollada, sobre la que se pudiera apoyar la ley.

Por lo tanto, el objetivo de la definición de normas es doble, conseguir un documento formalmente estable y preparar el camino a una posible legislación futura.

Importante

- Es importante aclarar que los repertorios de normas o guías de recomendaciones de accesibilidad deben considerarse en los diferentes estadíos de gestación del software, en el diseño, implementación y evaluación.

Principios generales de Accesibilidad

Debido a la cantidad considerable de normas de accesibilidad que se publican para la Web, aquí se va a realizar una selección de las más importantes y van a estar organizadas mediante el siguiente esquema:

- Principios de Accesibilidad para el Software: aquí, se analizarán las normas que está dirigidas a las aplicaciones en general. Hay que tener en cuenta que el navegador como el mismo sitio, si es que tiene una aplicación subyacente, estaría encuadradas dentro de esta categoría.
- Principios de Accesibilidad específicas para el Sitio Web: aquí se van a tratar las reglas de diseño tanto a nivel de sitio como a nivel página.
- Principios de Accesibilidad específicos a HTML: aquí se van a mencionar las normas de accesibilidad de acuerdo a las diferentes componentes que pueden conformar una página HTML. Se va a considerar específicamente, la accesibilidad en Listas,Tablas, Imágenes y gráficos de datos, Objetos Multimedia y otras Componentes.

Hay que tener en cuenta, que una norma puede estar diseñada para beneficiar más a un tipo de discapacidad que a otra. Por lo tanto, cuando se mencionen cada una, se va a especificar para qué grupo de discapacitados hace referencia. Se va a utilizar una convención o nomenclatura para clasificar los distintos impedimentos que un ser humano pueda sufrir. La tabla 6.6 muestra esta nomenclatura.

Las normas de accesibilidad tienen como principal objetivo incentivar al diseñador Web, en una cultura universal de diseño. La mayoría de ellas, apuntan principalmente, a diseñar el sitio con determinados principios que garanticen que el mismo, pueda ser utilizado correctamente por las herramientas de adaptación que usan las diferentes personas discapacitadas.

Nomenclatura sobre Tipos de Discapacidades	Explicación
LF	Incluye a personas con limitaciones motrices.
LP	Especifica al grupo de personas con limitaciones psíquicas.
LA	Hace referencia a personas con limitaciones auditivas de leves a moderadas.
LS	Incluye a personas con limitaciones auditivas severas.
LV	Incluye a personas con limitaciones visuales.
LC	Incluye a personas que sufren ceguera.
LSC	Hace referencia a personas sordo-ciegas.

Tabla 6.6: Nomenclatura para Referenciar las distintas Discapacidades.

Estos principios de accesibilidad se pueden aplicar sobre cualquier software en general. En el contexto de la Web, servirían para aplicarlas dentro del navegador como también para todo sitio que tenga alguna aplicación subyacente.

Dentro de estas normas, se encuentran reglas que hacen referencia a la accesibilidad de ciertas componentes de interacción generales como ser botones de acción, ventanas, menús, feedback, entre otros. No tiene alcance sobre cuestiones más específicas como la codificación en HTML.

Los principios que se pueden aplicar a cualquier aplicativo para que aseguren un nivel de accesibilidad, son:

Normas de Accesibilidad Generales	Tipo de Discapacidad que hace referencia
■La visualización de la información en la pantalla, como la interpretación de la misma, no debe estar sujeta a requisitos temporales. No se puede fijar un tiempo determinado para la lectura de la información. ■Si estos requisitos temporales existieran, debe ofrecerse la posibilidad de configurar el tiempo asociado.	■LF, LP, LA, LS
■No imponer la reacción del usuario en un tiempo máximo predeterminado. Tales mecanismos afectan el acceso a personas con limitaciones motoras en los miembros superiores.	■LF, LP, LA, LS
■La información en pantalla no debe ser transmitida a través de los colores o la forma de sus elementos.	■LV
■Cuando exista una forma de visualización de textos, como ser a través cuadro de textos, se debe poder recorrer con el cursor su contenido.	■LV, LC, LSC
■Se debe evitar el uso de gráficos que tengan dibujado el texto como parte de ellos. ■El texto debe ser independiente del gráfico de fondo, así podrá ser accedido por las herramientas de adaptación.	■LC, LSC
■Todos los botones deben tener asociada una etiqueta de texto y se debe facilitar una opción para que sólo se vea la etiqueta.	■LV, LC, LP, LSC
■Todas las funciones que se pueden realizar en la aplicación a través del ratón deben también ser accesibles por teclado.	■LF
■Todos los menús deben ser accesibles desde el teclado, incluyendo su activación y el movimiento por sus opciones	■LF
■Los menús de la aplicación deben ser circulares, es decir, al alcanzar el último elemento se debe pasar al primero y viceversa. ■Este sistema de movimiento circular debe aplicarse también a los diferentes sectores de la ventana, recorrido de opciones, menús, barras de herramientas o de hipervínculos.	■LF, LV, LC, LSC
■Las barras de herramientas, paletas, cajas de diálogo o cualquier otro elemento de la interfaz, deben ser accesibles por teclado y tener etiquetas textuales, aunque sea opcionales.	■LF
■El manejo de las ventanas deben posibilitar el ajuste de su tamaño y localización en pantalla también accesibles por teclado.	■LF
■Si se debe realizar acciones simultáneas, como mantener apretada una tecla mientras se pulsa otra, se debe proporcionar un método alternativo más simple, para lograr lo mismo.	■LF

■Se deben emplear medios alternativos para expresar el feedback, aunque sean opcionales. Por ejemplo, cualquier aviso o alarma sonora debe proporcionarse en forma visual y textual.	■LA, LS
■El feedback multimedial debe producirse inmediatamente después de que ocurra el evento que la genera, para evitar confusiones.	■LP, LV, LC, LSC
■Se debe ofrecer funciones que permitan manejar los distintos medios alternativos de comunicación, como también realizar conversiones entre ellos. Por ejemplo, permitir la posibilidad de enviar a salida de audio cualquier información textual.	■LF, LP, LA, LS, LV, LC, LSC
■Las características de accesibilidad de los navegadores deberán cumplir los mismos requisitos que cualquier otro programa de usuario	■LF, LP, LA, LS, LV, LC, LSC
■Las aplicaciones provistas dentro de los sitios Web, como ser programas CGI, Java, applets, scripts, entre otros, deberán cumplir los mismos requisitos de accesibilidad que cualquier otro programa de usuario.	■LF, LP, LA, LS, LV, LC, LSC

Tabla 6.7: Principios de Accesibilidad para el Software.

Principios de Accesibilidad Específicas para sitio Web

El sitio Web, al ser una clase especial de software, debe cumplir con las normas generales de accesibilidad, mencionadas en la sección anterior. Pero, además cuenta con su propio repertorio de normativas.

Se van a presentar las normas de accesibilidad más importantes a nivel de sitio y a nivel de página, que se encuentran en las tablas 6.8 y 6.9 respectivamente.

Normas de Accesibilidad a Nivel de Sitio	Tipo de Discapacidad Destinataria
■Establecer un diseño general del sitio simple, prolijo, que siga una línea de definición y lógica de organización.	LF, LP, LA, LS, LV, LC, LSC
■Usar una estructura de navegación clara, consistente permanente en todo el sitio.	■LF, LP, LA, LS, LV, LC, LSC
■Ofrecer una barra de navegación, que pueda ser expresada alternativamente como texto.	■LF, LP, LA, LS, LV
■Especificar el idioma en el que esta escrito el sitio.	■LF, LP, LA, LS, LV, LC, LSC
■Proveer una descripción del sitio y cómo se navega por él, mediante un mapa del mismo. ■Proveer versiones del mapa mediante distintos medios de comunicación.	■LF, LP, LA, LS, LV, LC, LSC
■Asegurar que todos los elementos y componentes del sitio, puedan ser operados desde el teclado estándar como también de dispositivos adaptados.	■LF, LP, LA, LS, LV, LC, LSC
■Cada hipervínculo debe tener una identificación única, que pueda ser interpretada mediante diferentes medios de comunicación.	■LF, LP, LA, LS, LV, LC, LSC
■Usar listas verticales de enlaces y que se encuentren fuera de los párrafos.	■LF, LP, LA, LS, LV, LC, LSC
■Que cada enlace tenga una explicación acotada y significativa, pues esta información es la que podrá captar la herramienta de adaptación que el discapacitado tenga.	■LF, LP, LV, LC, LSC
■Colocar espacios o caracteres separando cada hipervínculo.	■LV, LC, LP
■Los enlaces deben estar agrupados por una lógica de organización que sea clara y coherente.	■LF, LP, LA, LS, LV, LC, LSC

Tabla 6.8: Principios de Accesibilidad a Nivel de Sitio.

Normas de Accesibilidad a Nivel de Página	Tipo de Discapacidad Destinataria
■Las páginas deben cumplir las normas de accesibilidad o ser construidas de tal manera que puedan ser normalizadas fácilmente por servicios que cumplen tal fin.	■LF, LP, LA, LS, LV, LC, LSC
■Evitar fondos en mosaicos llamativos ya que el texto puede oscurecerse, ni utilizar colores oscuros, ni brillantes. ■Ofrecer un alto grado de contraste entre el texto y el fondo.	■LP, LV
■Evitar texto parpadeante o con algún otro efecto visual molesto. Evitar efectos de marquesinas.Dificulta la lectura y hasta puede provocar ataques de epilepsia.	■LV, LC, LP
■Ofrecer información adicional al comienzo de los párrafos, listas encabezados, etc.	■LF, LP, LV, LC, LSC
■Usar los encabezados (H1, H2, H3, etc.) para expresar la estructura de la página. No usarlos solo para definir tipografías en diferentes tamaños o estilos.	■LF, LP, LA, LS, LV, LC, LSC
■Se debe evitar el uso de textos verticales, emoticones, fechas abreviadas.	■LV,LC,LP
■Proveer títulos y aclaraciones a cualquier agrupamiento de información, como ser capítulos, secciones, como también acrónimos y abreviaturas.	■LV,LC,LP

Tabla 6.9: Normas de Accesibilidad a Nivel de Página.

Principios de Accesibilidad específicos a HTML

En esta sección, se van a listar las normas de accesibilidad más importantes que se aplican a cada componente, que pueda conformar una página HTML. Por lo tanto, se tratará la accesibilidad específicamente en las listas de opciones, tablas, imágenes, otros objetos multimedia, y demás elementos. Cada una de ellas se explicarán en tablas independientes que van desde la tabla 6.10 hasta la tabla 6.15.

Normas de Accesibilidad para Listas	Tipo de Discapacidad Destinataria
■Incluir un título descriptivo de la lista.	■LF, LP, LA, LS, LV, LC, LSCLC, LP
■Rotular numéricamente los elementos de la lista para facilitar su ubicación, también en el caso de las listas anidadas. ■Por ejemplo: rotular el primer nivel con números y los subitems con letras, o rotular en forma jerárquica: 1, 1.1, 1.2, 1.2.1	■LP, LV
■Incluir un salto de línea entre los elementos de la lista, para diferenciar bien cada ítem.	■LF, LP, LA, LS, LV, LC, LSC
■La lista debe poder ser manipulada por el teclado, mouse o cualquier otro dispositivo de señalamiento.	■LF, LP, LA, LS, LV, LC, LSC
■Finalizar todas las sentencias, encabezados, ítems de lista con algún símbolo de puntuación apropiado, para ayudar a interpretarlas.	■LF, LP, LV, LC, LSC

Tabla 6.10: Normas de Accesibilidad para Listas.

Normas de Accesibilidad para Tablas	Tipo de Discapacidad Destinataria
■Utilizar las tablas sólo para diagramar contenido, no para sectorizar la página.	■LV, LC
■Utilizar encabezamiento de fila y columna en forma consistente. No usarlos para alterar tipografía o alineación.	■LF, LP, LA, LS, LV, LC, LSC

■Si una tabla tiene dos o mas filas de títulos, usar marcas jerárquicas para identificarlas.	■LF, LP, LA, LS, LV, LC, LSC
■Asegurarse de que la información tenga sentido si se la lee línea por línea a través de la tabla. ■Esto es debido a que algunos Lectores de Pantalla o Texto “pronuncian” la información línea por línea.	■LP, LV, LC, LSC
■Acompañar la tabla con un hipervínculo a una página que contenga una versión lineal y textual de la información.	■LP, LV, LC, LSC

TABLA 6.11: Normas de Accesibilidad para Tablas.

Normas de Accesibilidad para Imágenes	Tipo de Discapacidad Destinataria
■Los gráficos internos (formato GIF) o gráficos que necesiten visualizadores externos (JPEG), deben incluir un texto alternativo que describa su contenido. ■Los caracteres alfanuméricos contenidos dentro de las imágenes y los gráficos, no deben reemplazar el texto alternativo.	■LV, LC, LP
■Las imágenes utilizadas como viñetas deben llevar como texto alternativo un asterisco (*). Esta es una convención difundida internacionalmente.	■LF, LP, LA, LS, LV, LC, LSC
■Dar nombre a los archivos que tengan relación con su contenido.	■LF, LP, LA, LS, LV, LC, LSC
■Si el contenido de una imagen o el gráfico es muy importante, se debe hacer un resumen del mismo en otra página y enlazarla utilizando un hipervínculo.	■LP, LV, LC, LSC
■Si se utilizan mapas sensibles hay que poner una lista de todos los enlaces a los que se puede acceder a través de él, o se puede dar una página alternativa en modo texto.	■LP, LV, LC, LSC

Tabla 6.12: Normas de Accesibilidad para Imágenes y Gráficos de Datos.

Normas de Accesibilidad Objetos Multimedia	Tipo de Discapacidad
■Las piezas de audio que vayan en una página deberán tener un enlace a una página con descripción textual del mismo.	■LV, LC, LP
■Incluir un texto alternativo que provea una descripción general de la animación utilizada. ■Esta norma se aplica a los siguientes formatos: Gif animados, Videos, Animaciones mediante scripts o applets.	■LF, LP, LA, LS, LV, LC, LSC
■Cada vez que se incluyan elementos sonoros, indicar su aparición y función con algún elemento visual. ■Dar relato en modo texto en forma sincronizada.	■LA, LS
■Incluir como opción para el usuario, el subtitulado o una transcripción de los elementos de sonido y voz del vídeo.	■LA, LS

Tabla 6.13: Normas de Accesibilidad para Objetos Multimedia.

Normas de Accesibilidad para Mapas de Imágenes	Tipo de Discapacidad
■Toda área activa de la imagen debe poseer un texto alternativo que describa el hipervínculo al que enlaza.	■LV, LC, LP
■Hacer que los hipervínculos sea accesibles para personas con discapacidad visual mediante el desplazamiento con la tecla Tab o flechas.	■LF, LP, LA, LS, LV, LC, LSC
■Usar solo mapas de tipo cliente.	■LA, LS

■Si no se ha logrado una versión accesible del mapa de imagen como opción alternativa o adicional, se puede: - Crear en algún lugar de la página una lista en formato texto de todos los hipervínculos provistos mediante el mapa de imágenes. - Agregar un texto alternativo de toda la imagen que sostiene el mapa indicando, por ejemplo, "mapa de imagen, ver abajo lista de vínculos".	■LA, LS

TABLA 6.14: Normas de Accesibilidad para Mapas de Imágenes.

Normas de Accesibilidad para otras Componentes	Tipo de Discapacidad
■Si se utilizan hojas de estilo, organizar los documentos de modo que puedan ser leídos de forma comprensible, aun si el navegador no soporta esta característica. ■Permitir definir estilo por lector.	■LF, LP, LA, LS, LV, LC, LSC
■Los formularios deberían estar acompañados de una información de audio.	■LP, LV, LC, LSC
■En los formularios se debe poner la etiqueta a la izquierda y alineada horizontalmente con la primera línea de campo de entrada o visualización de datos.	■LC
■Con respecto a los marcos, se recomienda minimizar o evitar el uso de los mismos.	■LV, LC, LP
■Es conveniente colocar para cada marco el atributo "title" o "name". Además, debe referenciar a un archivo html.	■LF, LP, LA, LS, LV, LC, LSC

Tabla 6.15: Normas de Accesibilidad para otras Componentes

Importante

■En estas secciones, se detallaron principios de diseño más importantes que inducen a garantizar la accesibilidad tanto en un software general como específico a un sitio Web.
■Se pretende que el sitio siga funcionando bien y que la información esté disponible, aunque sea operado por distintos dispositivos de interacción o mediante herramientas de adaptación.
■Vale destacar la gran importancia que se concede a la utilización de medios alternativos según el caso de discapacidad en cuestión.

Recomendaciones de Accesibilidad de la W3C-WAI

En esta sección, se van a explicar una serie de pautas de accesibilidad publicadas por la Iniciativa de Accesibilidad en la Web -WAI-, área perteneciente al Consorcio Internacional de la W3C.

Estas pautas conforman una Guía de Accesibilidad para Contenido Web, la misma es comúnmente referenciada por WCAG, que son las iniciales en inglés de Web Content Accessibility Guidelines. Actualmente, se encuentra la versión WCAG 1.0, que estuvo vigente desde el año 1999 y una nueva versión WCAG 2.0 que fue lanzada en abril de 2008.

El objetivo de la elaboración de estas Guías de Accesibilidad, es apuntar a una especificación de diseño universal explicando cómo hacer accesible el contenido de los sitios Web y promover su utilización en pos de lograr mayor funcionalidad y universalidad. Conforma un documento estable y puede ser utilizado como material de referencia por parte de los desarrolladores de contenidos de la Web, creadores de páginas y diseñadores de sitios, como también para los desarrolladores de herramientas de creación.

El respetar estas pautas hará la Web más accesible para todos los usuarios incluyendo las personas que sufren alguna discapacidad, como para aquellas que utilizan diferentes aplicaciones de usuario, que acceden desde distintos navegadores y desde diferentes dispositivos, mediante acceso móvil como celular o PALMS, desde el auto. También facilitará el acceso a aquellos usuarios que se conectan a Internet con recursos limitados, como monitores blanco y negro, conexiones telefónicas lentas, entorno de manos libres o alguna otra condición no óptima, que generalmente no son consideradas en el momento de diseñar una página Web.

Cabe aclarar que estas pautas no desalientan a los desarrolladores en la utilización de imágenes, vídeo,

ni de otros recursos multimediales, por el contrario explican cómo hacer los contenidos multimedia más accesibles a una amplia audiencia.

El documento WCAG 1.0, es parte de una serie de pautas de accesibilidad publicadas por la WAI, que incluye su nueva versión WCAG 2.0, como también las Pautas de Accesibilidad de las Aplicaciones de Usuario "WAI-User Agent" y las Pautas de Accesibilidad para Herramientas de Creación "WAI-Autools".

En las secciones siguientes, se van a explicar en rasgos generales, las características de la Guía de Accesibilidad WCAG 1.0 y su nueva versión, la WCAG 2.0.

WCAG 1.0

Descripción:

La Guía de Accesibilidad para Contenido Web WCAG 1.0, es un documento conformado por 14 pautas o normas de accesibilidad recomendadas por la WAI.

Las pautas tratan los aspectos de accesibilidad y proporcionan soluciones de diseño. Intentan que el desarrollador Web considere otras situaciones y otros contextos que el usuario puede estar inmerso en el momento de acceder al sitio Web y que no sean justamente circunstancias ideales.

Estas pautas se fundamentan en base a dos aspectos generales: asegurar una transformación correcta, que se tratan principalmente en las pautas 1 a 11, y hacer el contenido comprensible y navegable, recogido por las pautas 12 a 14.

El asegurar una transformación correcta significa construir el sitio de tal manera que permita la viabilidad de su utilización bajo diferentes circunstancias. Apunta principalmente a:

- que el desarrollador separe el contenido de la estructura y de la presentación, utilice las etiquetas en su concepción original, trabaje con independencia de dispositivos. Esto último se refiere a que las páginas deben poder ser usadas por personas que no dispongan de ratón, con pantallas pequeñas, de baja resolución, en blanco y negro, etc.
- impulsar el resurgimiento del texto, puesto que pueden ser interpretados por la mayoría de los mecanismos de navegación y de los diferentes recursos de hard y software. El texto puede ser presentado al usuario como una voz sintetizada, en braille y en texto visible.
- considerar la importancia de medios alternativos, o sea crear sitios que funcionen mediante diferentes canales de comunicación, así el lector que puede tener distintas capacidades sensoriales, pueda acceder a la información.

Respecto a hacer comprensible y navegable el contenido, se refiere a promover un lenguaje claro, proporcionar mecanismos de navegación simples, proveer información de orientación y contexto, para evitar que el usuario se desoriente o se retrase en el acceso a la información.

Contenido de la Guía WCAG 1.0:

Como se menciona anteriormente la recomendación WCAG 1.0, está organizada por 14 normas o pautas de accesibilidad que constituyen los principios generales para alcanzar el diseño accesible de un sitio Web. A continuación se van a presentar cada una de ellas con una breve descripción:

1. Proveer medios alternativos equivalentes: esta pauta recomienda el uso de texto para todo elemento no textual, proporcionar descripciones auditivas a la presentación específicamente visual, subtitular vídeos o películas presentes en la página Web, etc.
2. No depender del color, forma o imagen para expresar contenido: aquí la guía sugiere que toda información transmitida a través del color, también esté disponible sin él, que los colores utilizados entre el fondo y el primer plano tengan buen contraste o también, que el contenido se perciba correctamente en monitores blanco y negro, etc.
3. Usar hojas de estilo y marcado apropiadamente: la W3C sugiere utilizar hojas de estilo para controlar la presentación de los documentos Web como también crear documentos válidos a nivel de gramáticas formales publicadas. Se recomienda en esta pauta utilizar unidades relativas en los valores de los atributos o propiedades, para que se vean correctamente desde cualquier resolución de monitor.
4. Aclarar los lenguajes utilizados: aquí se recomienda aclarar el idioma principal del documento como también identificar claramente los cambios de idioma que existan dentro del contenido Web.
5. Proveer tablas accesibles: en este punto, la guía WCAG 1.0 sugiere utilizar las tablas sólo para lo que originalmente fueron concebidas, es decir para representar en forma tabular la información. No se

admite utilizarlas para diagramar o sectorizar una página.

6. Asegurar que las páginas que utilizan nuevas tecnologías se transformen correctamente: esto significa que si el sitio trabaja con hojas de estilo, scripts, applets u otros objetos, también pueda ser leído correctamente en caso que el navegador no acepte dichas tecnologías.
7. Asegurar control del usuario en contenido que depende de cambios temporales: esta pauta apunta principalmente a evitar destellos, parpadeos, movimientos continuos en pantalla que el usuario no pueda controlar.
8. Asegurar accesibilidad en interfaces del usuario incrustadas: aquí se refiere a que el desarrollador Web debería trasladar todas estas recomendaciones de accesibilidad a elementos de programación como applets, scripts que estén incorporados en la página.
9. Diseñar para independencia de dispositivos: esta norma promueve el acceso independiente del dispositivo, o sea que el sitio pueda ser usado con otros dispositivos que no sea sólo el ratón y la pantalla. Además se debe asegurar que el sitio pueda ser operado correctamente a través del teclado.
10. Usar soluciones generales o perpetuas de accesibilidad: esta pauta tiene por objetivo que el sitio Web pueda ser operado correctamente tanto desde diferentes herramientas de adaptación como desde los navegadores más antiguos.
11. Usar las guías y tecnologías recomendadas por la W3C: la W3C a través de un proceso consensuado lista las tecnologías recomendadas. Por ejemplo, sugiere utilizar documentos en formato HTML, o TXT en vez de DOC o PDF, para evitar exigir al lector tener instalado productos de software específicos.
12. Proveer información de contexto y de orientación: aquí se recomienda la utilización correcta de títulos, agrupamientos, etiquetas, de información de contexto.
13. Proveer claros mecanismos de navegación: esta pauta sugiere proporcionar mecanismos de navegación claros y coherentes, información de orientación, barras de navegación, mapa del sitio, entre otros, para incrementar la probabilidad de que una persona encuentre fácilmente lo que está buscando dentro del sitio.
14. Asegurar que los documentos sean simples y claros: esto apunta a que la estructura de las páginas sea coherente, que los gráficos sean reconocibles e incluyan una descripción y que el lenguaje sea fácilmente comprensible.

Cada una de estas pautas presenta puntos de verificación que son principios de diseño más específicos que describen la misma. Los puntos de verificación de una norma, tienen asignada una prioridad que puede ser entre 1 y 3.

El significado de las prioridades es el siguiente:

- La prioridad 1: significa que ese punto de verificación “tiene que ser” cumplido por el desarrollador de contenidos.
- La prioridad 2: se refiere a que el punto de verificación “debe ser” cumplido.
- La prioridad 3: presenta el menor nivel de exigencia, indicando que el punto de verificación “puede ser” cumplido por el sitio Web.

Cada punto de verificación pretende ser lo suficientemente específico, como para que cualquiera que revise una página o sitio pueda comprobar que dicho punto haya sido satisfecho.

Es importante aclarar que la WAI incluye un documento adicional, titulado “Técnicas para las Pautas de Accesibilidad al Contenido en la Web 1.0” que explica cómo aplicar los puntos de verificación de cada norma. El mismo trata cada punto de verificación con más detalle y proporciona ejemplos usando el Lenguaje de Marcado de Hipertexto (HTML), las Hojas de Estilo en Cascada (CSS), el Lenguaje de Integración Multimedia Sincronizada (SMIL) y el Lenguaje de Marcado Matemático (MathML). Este documento se encuentra disponible en el sitio http://www.discapnet.es/web_accesible/wcag10/WAI-WEBCONTENT-19990505_es.html#ref-TECHNIQUES y ha sido diseñado para seguir los cambios en la tecnología y es de esperar que sea renovado con frecuencia.

Niveles de Conformidad o Adecuación:

Una vez que el desarrollador Web consideró, analizó y aplicó las normas de accesibilidad en su sitio Web, puede incorporar una identificación que indique que su sitio es accesible.

La WAI presenta una serie de niveles de conformidad o adecuación de accesibilidad, de acuerdo a las prioridades de los puntos de verificación que se hayan cumplido. Esto es:

- Adecuación de nivel A (A): se satisfacen todos los puntos de verificación de prioridad 1.
- Adecuación de nivel Doble A (AA): se satisfacen todos los puntos de verificación de prioridad 1 y 2.

■Adecuación de nivel Triple A (AAA): se satisfacen todos los puntos de verificación de prioridad 1, 2 y 3.

WCAG 2.0

Descripción:

La guía de accesibilidad WCAG 2.0, intenta sustituir, ampliar y mejorar a la especificación 1.0, versión vigente desde 1999. La misma consta de una serie de documentos y recursos que comprende, entre otros, las pautas, listas de comprobación, documentos para la comprensión de WCAG, técnicas para la correcta aplicación en diversos casos de uso.

En esta guía, la W3C asegura que se alcanzó un mayor consenso que la primera vez que se realizaron las WCAG 1.0, teniendo mucho más en cuenta las aportaciones de las personas con discapacidad.

La WCAG 2.0 es más robusta, fácil de evaluar, y soporta más tipos de información que su versión anterior. Otra importante diferencia frente a ella, es la neutralidad tecnológica.

La WCAG 1.0 se basa fundamentalmente en escenarios con HTML, mientras que en esta nueva versión, se intenta dar respuesta a un mayor número de situaciones y aplicaciones, independientemente de la tecnología empleada.

Contenido de la Guía WCAG 2.0:

La WAI sostiene en su alegato que existen hoy en día muchas personas y organizaciones que utilizan las guías WCAG como referente normativo, incluyendo a diseñadores y desarrolladores Web, alumnos, docentes, encargados de formular políticas, entre otros.

Con el objeto de solventar las diferentes necesidades de cada perfil, se provee una organización de la guía en capas de recomendación, las cuales incluyen principios generales, pautas, criterios de comprobación y un conjunto de técnicas obligatorias y de asesoramiento.

También incluye documentación sobre errores comunes, acompañado de ejemplificaciones, hipervínculos a recursos y líneas de código.

Las capas mencionadas son:

Niveles en Capas de la WCAG 2.0	Explicación
Principios	La WCAG 2.0 se funda en 4 principios de diseño, que son: ■Perceptibilidad, en el sentido de promover contenido que sea visible, aparente. ■Operabilidad, o sea asegurar que los elementos de la interfaz permitan una correcta interacción. ■Comprensión que hace referencia a proveer contenido y controles que sean claros, simples y semánticamente ricos. ■Robustez: en pos de asegurar que el contenido Web sea lo suficientemente consistente como para permitir su uso con las tecnologías actuales, anteriores y las venideras.
Las Pautas	■Debajo de los principios, se encuentran las pautas. Estas son en total 12 normas que proveen los objetivos básicos con que el desarrollador debe trabajar para garantizar el contenido Web accesible. ■Estas normas proveen un marco teórico que no se puede testear, por ende, constituyen el cimiento para poder entender los criterios e implementar mejor las técnicas propuestas.
Criterios de comprobación	■Cada pauta tiene asociado un conjunto de criterios de éxito o de comprobación. ■Son 60 criterios en total que se deben cumplir y que tienen como características que sí son comprobables o testeables. ■Están ordenados según su nivel de cumplimiento asociado. -El nivel 1 o “A” significa que el criterio ha sido cumplido con los requisitos mínimos, - El nivel 2 o “AA” y - El nivel mayor de cumplimiento es el nivel 3 o “AAA”, ■Cada criterio cuenta además con información sobre ejemplos, errores comunes y un conjunto de técnicas para resolverlos. Las técnicas conforman la cuarta capa de recomendación.

Técnicas obligatorias y de asesoramiento	■Para cada norma y criterio de comprobación, la WCAG 2.0 presenta una serie de técnicas ■Las cuáles pueden ser: - las técnicas obligatorias que son suficientes como para alcanzar el criterio de éxito y -las técnicas de asesoramiento que van más allá de lo que se requiere objetivamente, las mismas no son testeables y su utilidad depende del contexto y circunstancia en que se las emplee.

Tabla 6.16: Capas de la Guía de Accesibilidad WCAG 2.0

Partiendo de 4 principios básicos, la W3C organiza la Guía WCAG 2.0 como una jerarquía de elementos de distintos niveles de abstracción.

Los criterios de comprobación que pertenecen a las pautas y éstas a su vez a alguno de los 4 principios, están compuestos de técnicas y son elementos básicos para comprobar la accesibilidad y poder determinar el nivel de cumplimiento de la misma.

Por la neutralidad de la tecnología atribuida a la WCAG 2.0, de alguna manera con la cuarta capa, la de las técnicas, se alcanza la especificidad y ejemplificación necesarias para cada tecnología particular. Hay técnicas generales, pero también hay técnicas para HTML y XHTML, CSS, técnicas para Scripting del lado del cliente, del lado del servidor, entre otros.

Niveles de Conformidad o Adecuación:

Existen tres niveles de conformidad:

■WCAG 2.0 Nivel A: cuando se cumplen todos los criterios de éxito de nivel 1 (A) de todas las directrices o se proporciona una versión alternativa conforme al nivel A.

■WCAG 2.0 Nivel AA: cuando se cumplen todos los criterios de éxito de nivel 1 (A) y de nivel 2 (AA) de todas las directrices, o se proporciona una versión alternativa conforme al nivel AA.

■WCAG 2.0 Nivel AAA: cuando se cumplen todos los criterios de éxito de nivel 1 (A), de nivel 2 (AA) y de nivel 3 (AAA) de todas las directrices, o se proporciona una versión alternativa conforme al nivel AAA.

La W3C brinda un documento comparativo entre las dos versiones de la WCAG, denominado “Comparación de los puntos de verificación de la WCAG 1.0 y la WCAG 2.0”, el mismo se encuentra en la dirección URL:
http://www.w3.org/TR/2006/WD-WCAG20-20060427/appendixD.html

VI- Comprobación de la Accesibilidad

Los consorcios internacionales como la WAI, recomiendan además de las Guías de Accesibilidad, la utilización de herramientas automáticas para evaluar la accesibilidad de los sitios Web. Las mismas son referenciadas con el nombre de validadores o revisores de Accesibilidad.

Estos revisores son aplicativos que toman un sitio Web, lo analizan respecto a determinadas normas con las cuáles trabajan y dan un reporte sobre los errores encontrados y principios que se han infringido.

Existen un gran número de estas herramientas las cuáles presentan diferentes características que deben ser consideradas en un proceso de selección.

En primera medida, hay que considerar con qué normativas trabaja la herramienta. Si considera las normas WCAG 1.0, las WCAG 2.0, las de Sección 508 o sigue los estándares de Japón, de Italia, u otros. Hay que analizar los tipos de tecnologías Web que soporta, si simplemente HTML y XHTML, o aquellas más sofisticadas como SMIL, SVG, o MathML.

Es fundamental tener conocimiento si es una herramienta de software libre o es un producto propietario. También, verificar qué plataforma soporta y cómo se puede integrar dentro del ambiente de desarrollo utilizado, si mediante plug-ins en el navegador o como utilitarios dentro del editor o sistema de administración de contenido Web.

También, hay que analizar el tipo de soporte que brinda la herramienta respecto al chequeo de errores, si genera reportes o no, si muestra a través de feedback visual íconos o remarca las etiquetas erróneas sobre la misma página, si permite o no el mecanismo de evaluación paso a paso.

En el caso de herramientas más avanzadas, éstas permiten realizar transformaciones sobre el diseño del sitio, por ejemplo convertirlo totalmente a modo textual.

La mayoría de los revisores clasifican los errores de accesibilidad en dos tipos:

- Errores automáticos: fueron detectados por la herramienta con seguridad. Tiene como alternativa de solución modificar el código del sitio. Por ejemplo, utilizar valores absolutos.
- Advertencias: la herramienta no puede garantizar su existencia. Es el ser humano quién debe comprobar si el error existe realmente y solucionarlo posteriormente. Por ejemplo, constrastes de color de fondo y de texto poco distinguibles.

La W3C ofrece un listado de herramientas que permanentemente actualiza, en las que se destacan:

- Herramientas de revisión automática de la accesibilidad: TAW, Cynthia, Bobby, Wave
- Herramientas de revisión de sintaxis: como el Validator del W3C, W3C CSS Validation Service, RDF Validation Service, XML schema validator
- Herramientas para la revisión Manual de la accesibilidad: HERA, HERA-XP, EDIPO
- Herramientas reparadoras: A-Prompt, LIFT, AccVerify, AccRepair y AccMonitor
- Navegadores alternativos y emuladores: como ser Amaya, Braillesurf, IBM HomePage Reader. Lynx y el emulador Lynxme, Mozilla y Firefox, Opera, Safari, WebTV Emulator
- Barras de revisión de la accesibilidad para navegadores: Barra para Internet Explorer, para Mozilla, para Firefox, etc.

En la figura siguiente se muestra la pantalla del revisor de accesibilidad TAW, validando el sitio Web de la Matanza, una importante localidad de la Pcia.de Buenos Aires de la República Argentina.

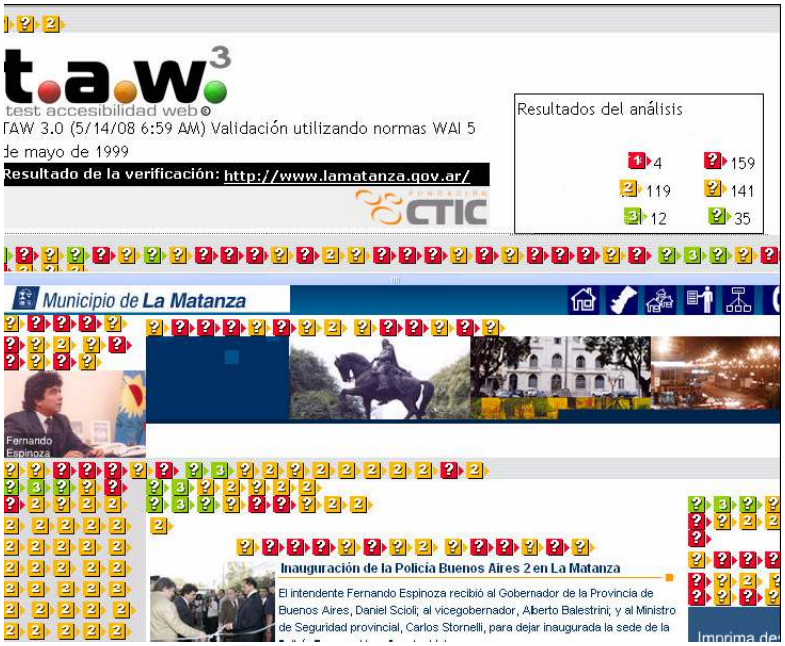


Figura 6.2: Imagen del Revisor de Accesibilidad TAW

En la imagen se perciben los íconos indicadores de error, con una numeración y color que determinan la prioridad del error. La ubicación de los mismos es significativa pues indican el lugar donde el error es cometido. Al cliquear en alguno de estos íconos, se muestra el código HTML que provocó el error y la recomendación de accesibilidad para solucionarlo.

En el cuadro superior derecho, se muestran totales de cantidad de errores producidos en la página clasificados por tipo de prioridad y sin son errores manuales o automáticos.

Vale aclarar que este proceso de validación automática no debe ser el único mecanismo de comprobación de la accesibilidad de un sitio. Es un proceso complementario que debe acompañar evaluaciones manuales.

Teniendo en cuenta un marco metodológico para la comprobación de la accesibilidad, todo desarrollador Web debería considerar básicamente los siguientes pasos, que son previos a la puesta en producción y publicación del sitio Web:

- Revisar el sitio con herramientas de evaluación de accesibilidad automáticas como las mencionadas anteriormente. Tener en cuenta que a veces detecta advertencias que deben ser comprobadas en forma manual.

- Solucionar los problemas detectados por el revisor hasta alcanzar un nivel de conformidad de accesibilidad. Tener en cuenta que implementar la accesibilidad puede implicar modificación de código, alteración del aspecto visual del sitio como también la construcción de páginas alternativas accesibles.
- Validar el sitio con diferentes resoluciones de monitores
- Probar el sitio desde diferentes navegadores y tipos de conexión
- Utilizar el sitio totalmente con el teclado
- Utilizar el sitio solamente con mouse, verificando el llenado de formularios con teclados virtuales
- Comprobar la operabilidad del sitio con alguna herramienta de adaptación, como el JAWS, Hal, Orca u otro lector de pantalla, con algún magnificador de pantalla, lector de documentos, entre otros

Lograr accesibilidad en un sitio es un proceso costoso, que todo desarrollador debe enfrentar para garantizar el acceso a la información a todos, sin discriminaciones.

Caso de Estudio 6

A continuación se presentará una síntesis de un estudio de accesibilidad realizado durante el mes de mayo de 2008, sobre los sitios Web de ministerios y municipios de la Provincia de Buenos Aires. El resultado de este informe se presentó en las Jornadas Argentinas de Computación, JAIIO 2008, y resultó ganador del Premio Nacional de Gobierno Electrónico, en el marco del Simposio de Informática en el Estado.

Objetivo

El objetivo de este caso de estudio es presentar una síntesis del artículo Evaluación de Accesibilidad de Sitios Web Oficiales: Ministerios y Municipios de la Provincia de Buenos Aires. Al través del sitio Web Wayback Machine http://www.archive.org/web/web.php pueden consultarse los sitios Web analizados, con la información y el formato del período durante el cual se realizó la evaluación.

Características del Caso de estudio

Hoy en día, se ha impulsado notablemente el uso de las Tecnologías de la Información y la Comunicación aplicado a mejorar la gestión y los servicios de la administración pública nacional, garantizar la transparencia y la participación y facilitar el acceso a la información pública. Esto se vio reflejado desde el Estado Nacional, a través del Decreto 378/2005, donde se aprobaron los lineamientos estratégicos para el Plan Nacional de Gobierno Electrónico y los Planes Sectoriales.

A partir de entonces, importantes aplicaciones oficiales se desarrollaron y actualizaron, contando con un marco de apoyo, asesoramiento y delineamientos generales, provisto por la ONTI, la Oficina Nacional de Tecnologías de Información.

La Provincia de Buenos Aires cuenta con una comunidad de 13.760.969 de habitantes y 134 municipios bonaerenses. Provee más de 100 sitios oficiales pertenecientes a dichos municipios y más de 30 sitios oficiales de ministerios, secretarías y de otros organismos. Datos que evidencian la presencia de una gran cantidad de usuarios Web.

El artículo describe un proceso de evaluación de accesibilidad de los sitios oficiales de la provincia de Buenos Aires, teniendo en cuenta una muestra de 9 sitios ministeriales y 22 sitios municipales, entre los que incluye partidos con un alto nivel de población como La Matanza. La monografía intenta brindar un aporte significativo, que motive la reflexión y autocrítica el pos de lograr el espíritu expresado desde el gobierno nacional: que la información esté al alcance de todos.

Desarrollo del Caso de Estudio

En esta sección describirá el proceso de evaluación de accesibilidad de sitios oficiales pertenecientes a la Provincia de Buenos Aires.

Las normas internacionales de accesibilidad aplicadas en el proceso fueron las otorgadas por la WAI, denominada “Guía para la accesibilidad y la autoría de páginas”. Como mencionamos en este capítulo, la misma es referenciada comúnmente con las siglas WCAG y su versión vigente es la WCAG 1.0.

Los consorcios internacionales como la WAI, recomiendan además la utilización de herramientas

Validador de accesibilidad utilizado: TAW On Click

automáticas para evaluar la accesibilidad de los sitios Web. Existen un gran número de estas herramientas las cuáles presentan diferentes características. Luego de un proceso de análisis, se seleccionó para este trabajo, la herramienta de validación de accesibilidad TAW On Click, que es simple de utilizar, viene en lenguaje español y es de libre licencia, además puede instalarse como una extensión al navegador Mozilla Firefox. En la figura siguiente se muestra el sitio del municipio de La Matanza, luego de evaluar su accesibilidad mediante la herramienta TAW.

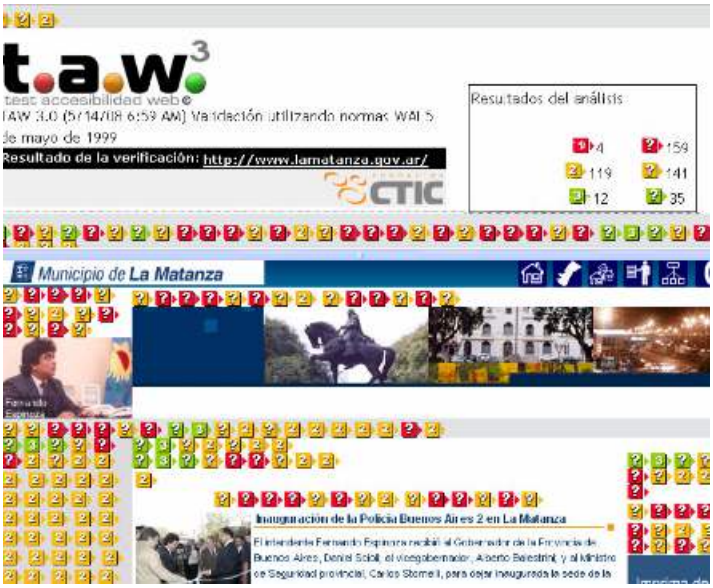


Figura 6.3: Pantalla del sitio de la Matanza evaluado por el validador de accesibilidad TAW.

Como se explicó en el Capítulo 6, en la parte superior derecha, presenta un cuadro con la cantidad de errores encontrados, de tipo automático o manual y según su prioridad.

Los errores automáticos son aquellos que la herramienta detecta dentro del código HTML y asegura su presencia. Los errores manuales son advertencias en donde la herramienta sugiere que el propio evaluador realice el testeo pertinente para verificar si el error existe o no.

En la parte inferior central, la herramienta muestra el sitio Web analizado y con colores puntualiza la ubicación de los errores detectados por ella. Es importante destacar que cuando la herramienta encuentra 0 (cero) errores de tipo automático y prioridad 1, se puede argumentar que el sitio cumple con un nivel de accesibilidad básico.

Sitios Evaluados

La evaluación parte de una muestra de 31 sitios oficiales, de los cuales 9 pertenecen a ministerios y 22 a municipios, que incluyen partidos de más de un millón de personas como es La Matanza.

En las siguientes tablas, se muestran los sitios analizados y se incluye información sobre su condición de accesible o no, luego de haberlos expuesto al validador de accesibilidad TAW.

Sitios Ministeriales de la Provincia de Buenos Aires:

Descripción	¿Es Accesible?
Ministerio de Infraestructura	No es accessible
Ministerio de Justicia	No es accessible
Ministerio de Salud	No es accessible
Gobierno de la Pcia de Buenos Aires	Nivel A
Ministerio de Seguridad	No es accessible
Ministerio de Asuntos Agrarios	No es accessible
Ministerio de Economía	No es accessible
Ministerio de Jefatura de Gabinete y Gobierno	No es accessible
Ministerio de Desarrollo Social	No es accessible

Tabla 6.17: Sitios Ministeriales de la Provincia de Buenos Aires

Sitios Municipales de la Provincia de Buenos Aires:

Descripción	¿Es Accesible?
Almirante Brown	No es accessible
Avellaneda	No es accessible
Azul	No es accessible

Bahía Blanca	No es accessible
Bragado	No es accessible
Brandsen	No es accessible
Campana	No es accessible
Carlos Casares	No es accessible
Chascomús	Nivel A
Escobar	No es accessible
Esteban Echeverría	Nivel A
Florencio Varela	No es accessible
General Pueyrredón	No es accessible
General San Martín	No es accessible
La Matanza	No es accessible
La Plata	No es accessible
Lanas	No es accessible
Moreno	No es accessible
Morón	No es accessible
Pilar	No es accessible
Quilmes	No es accessible

Tabla 6.18: Sitios Municipales de la Provincia de Buenos Aires

Registro de los resultados

Se diseñó y confeccionó una base de datos, que permitió almacenar los resultados arrojados por la herramienta TAW, de cada sitio Web evaluado. Se trabajó con dos tablas, una de "Sitios Analizados", donde se registraron la cantidad de errores detectados en los sitios, ya sean manuales o automáticos y por prioridad 1, 2 o 3 (en las próximas secciones se mencionarán como P1, P2 y P3 respectivamente). La otra tabla es la de "DetalleAccesibilidad" donde para cada sitio se contabiliza la cantidad de ocurrencias de error cometidas en cada norma. Luego, se realizaron consultas que se trasladaron a planillas de cálculo para realizar gráficos y funciones estadísticas.

Análisis de los Resultados de la Evaluación de Accesibilidad

En este apartado, se van a mostrar algunos de los resultados obtenidos a partir de la evaluación realizada.

Resultado sobre Accesibilidad respecto del total de sitios evaluados	
Accesibles	No Accesible
3	28

Tabla 6.19: Resultado sobre Accesibilidad respecto al Total de Sitios evaluados

La tabla permite visualizar el alto índice de sitios inaccesibles dentro de la provincia de Buenos Aires, demostrando una problemática que existe y excede más allá de los ámbitos de la provincia y de las entidades gubernamentales.

❑Análisis comparativo de errores automáticos y manuales por prioridad, teniendo en cuenta sitios municipales y ministeriales

Errores Automáticos entre Municipios y Ministerios			
Tipo de Sitio	P1	P2	P3
Ministerios	433	3667	325
Municipios	425	2315	361

Tabla 6.20: Errores Automáticos por tipo de Sitio evaluado

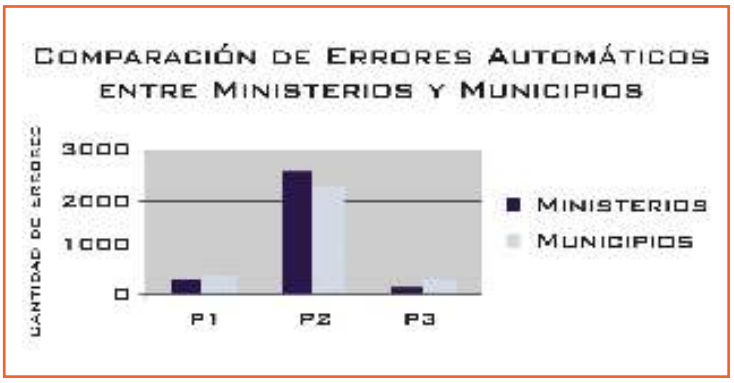


Figura 6.4: Gráfico comparativo de errores automáticos

Respecto a este gráfico, vale aclarar que los errores automáticos requieren para su solución indefectiblemente la modificación del código HTML en que fue desarrollado el sitio Web. Se puede percibir a simple vista que, para lograr un nivel básico de accesibilidad (Nivel A), donde requiere que los errores automáticos de prioridad 1 sean resueltos, no implicaría grandes esfuerzos. Pero esta posibilidad se complica, al querer tratar los errores de prioridad 2, que son muy elevados tanto en los sitios municipales como ministeriales.

Comparación Manuales entre Municipios y Ministerios			
Tipo de Sitio	P1	P2	P3
Ministerios	1651	2628	748
Municipios	2156	3064	862

Tabla 6.21: Cuadro comparativo de errores manuales

Figura 6.5: Gráfico comparativo de errores manuales

Con respecto a los errores manuales, son considerados por la herramienta como advertencias, y no son tenidos en cuenta en el momento de designar una clasificación de accesibilidad. De todas maneras, estos errores deben solucionarse. Los mismos requieren de un gran esfuerzo por parte de los evaluadores, puesto que deben realizar una revisión manual de cada error y corregirlas en el código. Se detecta un alto índice detectado de errores manuales de prioridad 1 y 2.

Análisis sobre errores automáticos y manuales por Municipio

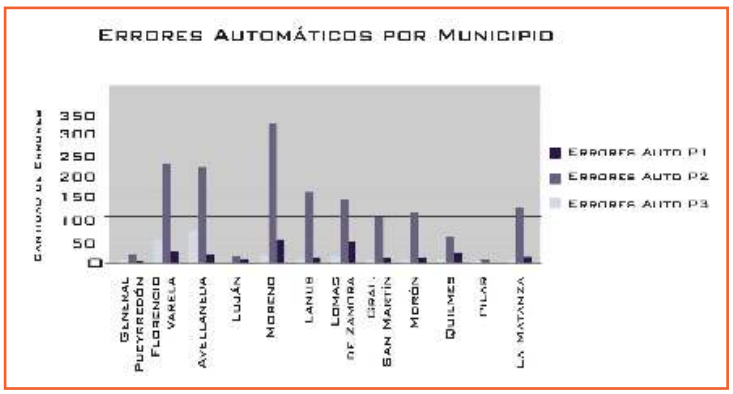


Figura 6.6: Gráfico de errores automáticos por municipio

Este gráfico muestra la tasa de errores automáticos por municipio, esto nos permite observar el alto nivel de problemas de prioridad 2, por lo que los costos para obtener un intermedio de accesibilidad (Nivel AA) son muy significativos.

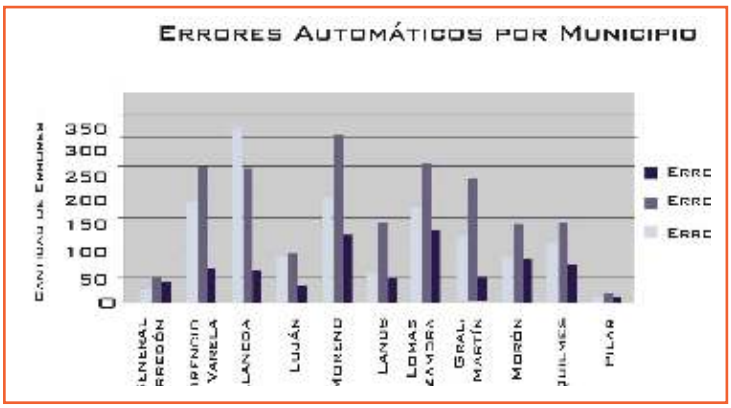


Figura 6.7: Gráfico de errores automáticos por municipio

Aquí, podemos visualizar sitios con un alto nivel de error de prioridad 1, 2 y 3, como ser el de Florencio Varela, Avellaneda, Moreno, Lomas de Zamora y Gral. San Martín. El trabajo para revertir esta situación implica retocar y modificar la mayor parte del código de los mismos.

❑Análisis de los errores de accesibilidad más cometidos por Prioridad

Según los resultados arrojados en la evaluación, los errores más cometidos teniendo en cuenta el total de sitios analizados, son:

De prioridad 1

■No se proporcionó un texto equivalente para todo elemento no textual: este error de tipo automático que implica la necesidad que todo elemento multimedial esté acompañado por texto, fue cometido con una ocurrencia de 160 veces.

■No es seguro que las páginas sigan siendo utilizables sin scripts, applets u otros objetos programados: este error de comprobación manual fue cometido una 195 veces. Implica que los sitios deben poder ser accedidos aunque el navegador utilizado no acepte scripts, applets u otros objetos de programación.

■El documento no puede ser leído sin hoja de estilo: Este error fue cometido 158 veces e implica que el sitio debe ser visto con y sin la tecnología de estilos CSS.

De Prioridad 2

■Se usa características desaconsejadas por las tecnologías W3C: Este error automático fue hallado 1073 veces. Significa que los sitios deben por ejemplo proveer información adicional en formato HTML o texto, no en .PDF o Word.

■No usar valores absolutos: este error es muy cometido. Fue detectado 459 veces y significa que se deben trabajar con coordenadas relativas, así funciona en diferentes resoluciones de monitor.

■No usa hojas de estilo: La W3C aconseja el uso de hojas de estilo pero además el sitio debe funcionar bien en situaciones donde el navegador no las reconozca. Se detectó la ausencia de hojas de estilo 322 veces.

■Poco contraste entre el fondo y la letra: Este de comprobación manual fue hallado 287 veces.

De Prioridad 3

■No se proporciona resúmenes de las tablas: este error automático detecta falta de información adicional a tablas utilizadas en el sitio. Detectó 205 ocurrencias.

■No se proporcionó texto lineal alternativo: Este error fue detectado 207 veces y apunta a proveer alternativas textuales para presentar la información.

Conclusiones

Este trabajo de investigación y análisis que muestra con datos concretos la ausencia notable de sitios accesibles, nos hace pensar en la falta de conciencia por parte de responsables en el desarrollo Web, de la necesidad de hacer accesibles sus sitios. Se destaca de los resultados que la mayoría ni siquiera cumplen con principios de prioridad 1, necesarios para garantizar un mínimo grado de accesibilidad.

Como trabajo futuro, se continuarán analizando los municipios y organismos públicos, se desarrollarán propuestas de difusión, formación y asesoramiento desde la UNLP, que desde el año 1995, viene acumulando una vasta experiencia en temas de accesibilidad Web.

Adicionar a las aplicaciones de e-gobierno presentes en Internet, la cualidad de accesibilidad permitirá garantizar que la administración e información pública esté efectivamente al alcance de todos, acortando significativamente la brecha digital existente. Esto estaría en consonancia con iniciativas para crear sitios accesibles que promueve la Unión Europea.

Ejercitación

A. Responder verdadero o falso.

- La accesibilidad apunta únicamente a las diferentes discapacidades de un usuario. V F
- La WAI, pertenece a la W3C y se ocupa específicamente de establecer y publicar normas de accesibilidad, para garantizar el diseño universal del sitio y el uso de elementos estándares. V F
- Las ayudas técnicas hacen referencia a las herramientas de adaptación que utilizan los usuarios con distintas discapacidades V F
- El diseño universal intenta abolir el uso de multimedia en el diseño de sitios Web V F
- La guía de accesibilidad WCAG 2.0 tiene la misma organización que la WCAG 1.0 V F

B. Indicar los problemas que pueden surgir cuando un usuario ciego quiere instalar un producto que lo asista en su navegación por la Web.

C. Indicar las comprobaciones necesarias desde el punto de vista de accesibilidad, que un desarrollador Web debe realizar antes de publicar su sitio Web.

Referencias

- Abascal,J. The use of guidelines to automatically verify Web accessibility.
- Arrue, I Fajardo (2004). Universal Access in the Information Society. Springer
- Paciello, M. (2000). Web Accessibility for People with Disabilities.
- Lazar J. A. Improving web accessibility: a study of webmaster perceptions.
- Sponaugle, Dudley (2004). Computers in Human Behavior. KD Greenidge.
- Y.Hassan Montero (2006) Factores del diseño web orientado a la satisfacción y no-frustración de uso.
- Revista española de Documentación Científica.
- Rovira, C. Libre acceso en Europa: análisis y valoración de la accesibilidad.
- Marcos, M.; Codina L (2007). El Profesional de la Informacion.
- Casals, Alicia (1997). Ajuts Tecnològics per a disminuïts físics.
- Egea García, Carlos y Sarabia Sánchez, Alicia (2001). Diseño Accesible de Páginas Web.
- Vanderheiden y Katherine R. Vanderheiden (1999). Accesible Design of Consumer Products.
- Pérez Alonso, Beatriz (1997). Internet sin barreras.
- Gregg C. Vanderheiden (1998). Aplication Software Design Guideline.
- Novática: Informática y discapacidades. Monografía; Número 136. 1998.
- Sitios y Portales de Internet para la Administración Pública Nacional. Documento oficial de la ONTI, Oficina Nacional de Tecnologías de la Información de Argentina. http://www.sgp.gov.ar/contenidos/onti/etap/sitio_etap/docs_y_varios/GuiasTecnicas/Pautas_para_Sitios_y_Portales_en_Internet_para_la_APN.pdf
- Plan nacional de Gobierno Electronico. Decreto Nº 378/2005: http://www.argentina.gov.ar/argentina/portal/documentos/decreto378.pdf
- http://www.magnifiers.org/
- http://www.dolphinusa.com/support/
- http://www.fbyte.com/
- http://www.microsoft.com/enable/guides/vision.htm
- http://www.svpal.org/assistive/
- http://www.sidar.org
- http://www.aenor.es
- http://www.legislaw.com.ar
- http://www.access-board.gov/sec508/guide/act.htm
- http://www.congreso.es/public_oficiales/L7/CONG/BOCG/A/A_068-13.PDF
- http://www.kantei.go.jp/foreign/it/network/priority-all/index.html
- http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505
- http://www.w3.org/WAI/intro/wcag10docs.php
- http://www.w3.org/WAI/intro/wcag20.php
- http://qweos.net
- http://www.w3.org/WAI/eval/Overview.html
- http://www.w3.org/WAI/ER/existingtools.html
- http://www.tawdis.net/taw3/cms/es
- http://www.cynthiasays.com/
- http://www.watchfire.com/products/desktop/accessibilitytesting/default.aspx
- http://www.firefoxflicks.magnify.net/
- Fundación de Ciegos Manuel Caragol: www.funcaragol.org
- Organización Nacional de Ciegos de España: www.once.es
- Decreto Nº 378/2005 Plan Nacional de Gobierno Electrónico. http://www.argentina.gov.ar.
- ONTI. http://www.sgp.gov.ar/contenidos/onti/onti.html
- Buenos Aires. La Provincia. http://www.gba.gov.ar/
- Validadores de Accesibilidad http://www.w3.org/WAI/ER/existingtools.html
- AW On Click. http://www.tawdis.net/taw3/cms/es

Conclusiones Generales

La ciencia y la tecnología avanzan y con su proceso, modifican la vida de cada ser humano. Sus horizontes, sus metas, sus propósitos se ven afectados. Muchas personas, con distintas formaciones, edades, experiencias, formas de ser, interactúan con algún sistema de software en su trabajo, en su hogar o para estudiar. Cabe aquí preguntarse cómo proveer la mejor interfaz del usuario que permita que el ser humano pueda aprovechar al máximo las herramientas tecnológicas que se le presentan, sin aumentarle las complicaciones.

Nuestras aspiraciones no se limitan a la formación del lector respecto a las conceptualizaciones más importantes relacionadas con la temática de HCI, sino más bien, promover un ser conciente de su rol de desarrollador de sistemas de software, las cuales deben estar al alcance, al acceso y al servicio de seres humanos.

El eje trasversal de este libro y que circula desde diferentes perspectivas en los distintos capítulos, es la misma interfaz del usuario. Es la componente de software que soporta el diálogo, la conversación que fluye entre un usuario y el sistema de software en ambas direcciones. Se encarga de dialogar con el usuario, de pedir datos de entrada y de mostrar los resultados o las salidas del programa. La interfaz del usuario constituye un puente, un nexo indispensable y único entre el usuario y la componente funcional del sistema. Es un concepto amplio y complejo, puede tomar diferentes dimensiones y se ve afectado por múltiples factores que se intentó analizar y explorar a lo largo de todo este material.

Se aborda toda la problemática de la interacción entre el usuario y el ordenador. Se enseñan características y diferentes estilos de diálogo, como también normas de diseño específicas para desarrollar interfaces del usuario que sean simples de utilizar y fácil de aprender por parte de los seres humanos.

Fundamentalmente, se promueve un proceso de ingeniería de la interfaz que permita la participación del usuario no sólo al final del desarrollo de la misma, cuando él comienza a probar y utilizar el sistema, sino en todas las etapas de gestación.

La interfaz se construye por y para los/las usuarios/as, teniendo en cuenta desde aspectos de visualización, servicios, estilo de interacción, formas de diálogo, comportamiento que debe ser acorde a sus necesidades.

El usuario se lo analiza como el agente principal para garantizar la calidad de uso del sistema. Es imposible analizar grados de satisfacción, eficiencia, cuestiones de uso y de gustos, sin la intervención directa y continua del actor usuario.

En los últimos capítulos del libro, se cubrió un panorama de diferentes tipos de interfaces del usuario, prestando atención a cuestiones específicas de diseño para que el lector pueda comprender la magnitud de problemáticas, costos y recomendaciones a tener en cuenta en el momento de encararlas.

El libro incluye ejemplos, casos de estudio, ejercitación, permitiendo una lectura dinámica y participativa del lector. Se seleccionaron trabajos realizados por alumnos avanzados de la cátedra de Diseño Centrado en el Usuario de la Fac.de Informática de la Universidad Nacional de La Plata, Argentina. A través de los cuáles los alumnos jugaron con el rol de diseñador y evaluador, pero también con el rol de usuario, experimentando problemáticas que sufre cualquier persona que se encuentra frente a un producto de software que quiere utilizar.

Otro punto a aclarar es que en este libro, se ha abarcado los conceptos básicos sobre diseño centrado en el usuario y presentado un conjunto de recomendaciones para el diseño de interfaces tradicionales. Sin embargo, esto es sólo el comienzo. La disciplina HCI continúa creciendo y, junto con el desarrollo de nuevo hardware con nuevas prestaciones y mayor potencialidad de cálculo, se plantean nuevos desafíos que años atrás se imaginaban sólo como parte de la ciencia ficción.

Respecto de las nuevas tendencias en las interfaces se pueden mencionar algunas más novedosas que otras, que no han sido incluidas en esta edición por cuestiones de espacio, pero que permitirá al lector tener una visión del estado del arte de la disciplina, son:

Pen-centric Computing: El objetivo de pen-centric computing es tomar ventaja de la habilidad que tenemos las personas para expresar, con un lápiz, las ideas que rondan en nuestra mente. No es sólo reemplazar el mouse por el lápiz, como es el objetivo de pen computing, sino que es hacer con él las operaciones que se realizan con un lápiz tradicional sobre una hoja de papel, por ejemplo tachar una palabra de un texto implica eliminarla.

Surface Computing: Representa una nueva forma de interacción con el contenido digital. Los canales de entrada tradicionales, como mouse y teclado, desaparecen para dejar paso a la manipulación directa de los objetos de información por medio de gestos naturales con las manos sobre una “mesa digital”. Esta interacción es posible llevarla a cabo en solitario o con otras personas en simultáneo. La pantalla táctil que constituye la mesa digital también reconoce objetos físicos que se ubican sobre ella. Así, es posible ubicar dos teléfonos celulares sobre la mesa digital e intercambiar contactos, fotos y demás información que reside

en cada uno de los dispositivos. Las posibilidades se amplían a límites impensados.

Interfaces Hápticas: La tecnología háptica permite implementar interfaces de usuario donde el sentido del tacto es el protagonista. Utiliza distintos recursos físicos, como aplicación de fuerza, vibraciones y/o movimientos, como mecanismos de entrada y/o de salida. Esta estimulación mecánica permite crear objetos virtuales y manipular distintos dispositivos en forma remota.

Interfaces Sociales - Supple interfaces: La Web 2.0 incluye aplicaciones que permiten construir redes sociales a través de la Web, donde es el usuario quien publica el contenido, se relaciona con otros usuarios a quienes conoce físicamente o no, comparten intereses, responden a diferentes motivaciones, organiza sus relaciones y actividades, estableciendo una forma de interacción con el sitio que responde a su forma de ser en la vida real.

Interfaces móviles: Los dispositivos móviles, particularmente los teléfonos celulares, smartphones y Palms, se han vuelto muy populares en todo el mundo. A través de ellos es posible no sólo comunicarse vía telefónica con otras personas, sino también consultar la hora, registrar alarmas o citas programadas, enviar y recibir mensajes, consultar el correo electrónico, navegar por la Web, acceder al servicio de GPS (Global Positioning System), escuchar música y realizar otras actividades sumamente útiles.

Esperamos que el libro permita familiarizar a los lectores con un tema actual y de gran dinamismo donde, en mayor o menor medida es posible participar activamente en un mundo donde los sistemas de Internet permiten comunicarse con más de mil de millones de personas y más de tres mil millones que usan telefonía celular, y sin embargo decisiones erróneas pueden inhibir el uso por parte de ciertos sectores de la sociedad.

05

Apéndices

Temas a Tratar

Glosario de términos utilizados

Ejercitación Resuelta

- Capítulo I
- Capítulo II
- Capítulo III
- Capítulo IV
- Capítulo V
- Capítulo VI

Glosario de términos utilizados

7 pasos de Norman	Concepto introducido por Donald Norman. Hace referencia a la mente humana durante la interacción con un sistema de cómputos para resolver un problema.
Accesibilidad	Concepto que hace referencia a proveer un diseño universal en los productos que se estén desarrollando, para que pueda ser entendido y utilizado por cualquier clase de persona. Ver Interfaz accesible.
Animación	Técnica de interacción que visualiza el proceso interno de un sistema dinámico. Es una parte integral en la interfaz visual donde los cambios en el sistema están provocados por acciones autónomas del propio objeto animado que complementa las operaciones del usuario
Browser	Técnica de interacción que permite visualizar una lista de objetos compuesto por varios atributos. Se visualiza como una tabla donde cada fila es la especificación de cada objeto y en las columnas se muestran los valores de sus atributos y características.
Caja de diálogo	Contexto rectangular, donde el usuario puede especificar opciones y acciones usando botones y controles tales como listas, casilleros de entrada, listas deslizables, check buttons o radio buttons.
Conocimiento semántico	El conocimiento dependiente del dominio, que es inherente a la aplicación específica.
Conocimiento sintáctico	Convenciones lingüísticas y modales que el operador debe conocer para poder comunicarse con el sistema.
Consistencia	La interfaz debe seguir ciertos patrones de conducta, de interacción y de visualización homogéneos y coherentes en todas las partes del sistema.
Controlador del diálogo	Porción del software de la interfaz que controla el medio utilizado y la secuenciación de las interacciones del usuario. Determina la lógica de eventos.
Dependencias	Técnica de interacción visual. Especifica una relación causal que debe ser siempre mantenida. Sucede cuando al manipular un objeto de la pantalla puede significar consecuencias en la aplicación que, a su vez, causen efectos directos o indirectos sobre otros objetos de la misma pantalla.
Diálogo secuencial	Forma en que los humanos interactúan con una computadora. La interfaz se basa en representaciones implícitas de los objetos. Se pasa de una etapa de diálogo a la siguiente de una forma preestablecida por el sistema. Este tipo de diálogo incluye interacción pregunta-respuesta (request-response), lenguajes de comandos, navegación a través de menús y entradas de datos.
Diálogo sincrónico	Forma en que los humanos interactúan con una computadora. El usuario muestra qué hacer mediante el desplazamiento y manipulación de representaciones visuales de los objetos. El estilo de interacción característico es la manipulación directa y el mouse. La secuenciación de cada camino de interacción es independiente de los demás.
Diálogo basado en eventos	Ver Diálogo asincrónico.
Diálogo Concurrente	Diálogo multi-thread, en el cuál más de un hilo de diálogo puede ser desarrollado paralelamente. Además de haber muchas alternativas de diálogo, estas están abiertas simultáneamente.
Diálogo no lineal	Multiplicidad de caminos de diálogo, de hilos de control simultáneos, de acciones alternativas en un determinado momento.
Diseño centrado en el usuario	Metodología para el desarrollo de un sistema de computadora donde el usuario participa activamente en todas las etapas de principio a fin.
Distancia de ejecución	Esfuerzo mental que hace el usuario cuando traslada su intención en requerimientos propios del sistema.
Distancia de evaluación	Esfuerzo mental que realiza el usuario cuando analiza el estado del sistema.
Estilo de interacción	Mecanismo implementado por la interfaz para comunicarse con el usuario: secuancial, manipulación directa, concurrente o basada en comandos.
FAQs	Preguntas frecuentes. Servicio adicional recomendado para utilizar en los sitios Web.

Interfaz de usuario	Medio por el cual se lleva a cabo la interacción entre el usuario y la computadora. Componente de software y hardware que soporta el diálogo bidireccional entre el usuario y el sistema.
Look and feel	Apariencia y percepción de la interfaz de usuario.
Manipulación directa	Técnica de interacción que responde a la metáfora de Mundo Modal, en donde el usuario tiene a disposición objetos que pueden ser accionados de múltiples maneras, como ser mediante el arrastre, selección, múltiple selección, cliqueo, entre otros.
Manual de usuario	Proveen documentación y detalles específicos de las operaciones provistas por el sistema y reflejan los diferentes modos de uso.
Menús	Lista limitada de opciones que pueden ser seleccionadas por el usuario y que están disponibles dentro de un marco visual común
Metáfora del desktop	Permite que el usuario perciba a través de la computadora, un ambiente que le resulta familiar, mediante la visualización de elementos, el acceso a funciones fácilmente y la localización de contextos lógicos, pudiendo apreciar a la computadora como una herramienta de fácil utilización y entendimiento.
Métodos de indagación	Método de evaluación llevado a cabo por los potenciales usuarios del sistema.
Métodos de inspección	Método de evaluación llevado a cabo por expertos en HCI.
Modelo de usuario	Descripción de las características y propiedades individuales de la interacción entre el usuario y la máquina, permitiendo modelar sus hábitos y reacciones.
Modelo en espiral	Ciclo de vida de una interfaz, definido por Boehm. Cada giro de la espiral está formado por el diseño, desarrollo y evaluación del prototipo.
Modelo en estrella	Ciclo de vida de una interfaz, definido por Hix y Hartson. La evaluación es el proceso central, a partir del cual se refinan las siguientes etapas de desarrollo.
Modelo SSOA	Modelo Sintáctico-Semántico del conocimiento del usuario, definido por Schneiderman.
Objetos computacionales	Conceptos, entidades, datos, funciones extraídos del problema real, que serán registrados, representados y manejados dentro del sistema.
Objetos de interacción	Elementos que conviven dentro de la interfaz del usuario y son reflejados al usuario. Incluye una componente de presentación y otra de interacción.
Objetos no computacionales	Revisar en interfaces icónicas
Paradigma de interacción visual	Recursos o mecanismos que el diseñador emplea para confeccionar una interfaz de usuario visual.
Perfil del usuario	Preferencias, hábitos y comportamiento de la persona que interactuará con el sistema.
Prototipación	Metodología de desarrollo o método o técnica de representación de diseño. Permite desarrollar una visión prematura del sistema, donde se puntualiza en un principio la visualización, la interacción y el comportamiento global del sistema. Permite acatar cuestiones de interfaz , en forma separada del aplicativo subyacente.
UI	Interfaz del usuario
UIMS	Sistemas para el gerenciamiento de la interfaz de usuario gráficas que focalizan en el diseño centrado en el usuario
UIT	User Interface Toolkit, librerías de rutinas que pueden ser invocadas para implementar ciertos aspectos de la interfaz.
User tailorability	Capacidad del usuario de adaptar y personalizar aplicaciones según sus gustos y necesidades.
Usabilidad	Conjunción de todas las cualidades exigidas a la componente de diálogo: simplicidad, confiabilidad, flexibilidad, transparencia y ergonomía.
Ventana	Area rectangular que contiene una aplicación de software o archivo de información. Permite la definición de un contexto independiente, donde se agrupan subtareas o dato del mismo nivel de abstracción.
W3C	World Wide Web Consortium. Foro internacional en el que participan unidades académicas y empresas de todo el mundo y tiene por objetivo desarrollar tecnologías interoperativas para guiar a la red a su máximo potencial. Define estándares presentes en este libro, como PGP, guías de accesibilidad, entre otros.

Feedback visual	Reacción de la interfaz frente a un estímulo. Es el proceso de reflejar sobre la pantalla el resultado de alguna operación realizada por el usuario, realizándose esto de una manera gráfica o mediante algún recurso visual
Formulario	Componente visual de una interfaz, que permite reunir un conjunto de diferentes objetos de interacción en pos de permitir la entrada de datos por parte del usuario
Funciones sintácticas	Facilidades de configuración y personalización que poseen ciertas técnicas de interacción visual. Por ejemplo la posibilidad de configurar el ancho de las columnas, el orden y la tipografía de un browser.
Gestos	Marca, estela o camino que traza el usuario para expresar alguna información o representar la interacción del mismo
Independencia de diálogo	Estrategia que permite que las decisiones de diseño que afectan únicamente a la componente de diálogo se realicen en forma independiente de aquellas que afectan a la estructura del sistema de aplicación y al corazón funcional.
Interfaz Accesible	Interfaz que respeta normas de diseño de universalidad para que pueda ser accedida por cualquier usuario independientemente de sus capacidades físicas, mentales.
Interfaz para groupware	Interfaz social o grupal, que facilita el trabajo en grupo, permitiendo la coordinación, comunicación y proyección entre todos los integrantes a través de la computadora.
Interfaz basada en agentes	Interfaz capaz de comenzar el diálogo y finalizarlo. Se comporta como un interlocutor humano.
Interfaz con inferencia	Interfaz que posee la capacidad de captar secuencias de acciones que el usuario repite con frecuencia.
Interfaz con signos de adaptación	Interfaz sensible a los perfiles individuales de los usuarios y sus estilos de interacción.
Interfaz conversacional	Interfaz que se monta sobre los sistemas telefónicos.
Interfaz evolutiva	Interfaz que tienen la propiedad de cambiar y evolucionar con el tiempo, junto con el grado de perfeccionamiento que el usuario particular va adquiriendo con el sistema.
Interfaz icónica	Interfaz visual específica, que utiliza como medio de representación visual exclusivamente al ícono. Se puede especificar tanto visualización icónica como programación icónica.
Interfaz inteligente	Interfaz con capacidad de razonamiento, de adquisición y aplicación del conocimiento y de comunicación de ideas.
Interfaz móvil	Interfaz de aplicaciones donde el usuario utiliza dispositivos inalámbricos como celulares y PALMs para acceder a la funcionalidad del sistema.
Interfaz para groupware	Interfaz social o grupal, que facilita el trabajo en grupo, permitiendo la coordinación, comunicación y proyección entre todos los integrantes a través de la computadora.
Interfaz para la Web	Interfaz presente en los sitios Web.
Interfaz textual	Interfaz que utiliza el texto como único medio de representación y de interacción. Como ejemplos se pueden citar las interfaces basadas en comandos y las basadas en menús.
GUI	Interfaz de usuario gráfica, que soporta alguna de las siguientes características: visualización, interacción o programación visual.
HCI	Interacción hombre-computadora. Es el intercambio observable de información, datos y acciones, entre un humano y la computadora, y viceversa. Conforma una disciplina dentro de las Ciencias de Computación.
heurística	Método de evaluación realizada por expertos en HCI, a partir de un conjunto de pautas bien definidas.
Hipervínculo	Palabra, frase, imagen o símbolo que permite a los usuarios saltar de un lado a otra ubicación del mismo sitio o un sitio externo.
Icono	Imagen, figura o símbolo que representa un concepto subyacente. Par (parte física, parte lógica) donde la parte física sería la imagen y la parte lógica su significado. Relación unívoca entre imagen y significado.
Ingeniería de usabilidad	Metodología de desarrollo adecuada para sitios Web, promovida por Jacob Nielsen. Consiste en tener en cuenta las cuestiones de usabilidad durante todo el proceso de desarrollo.

WIMP	Windows, icons, menus and pointing device. Estilo de interfaz que hace referencia a las técnicas de ventanas, íconos, menús y manipulación directa
WYSIWYG	What You See Is What You Get Estilo de interfaces gráficas estándares, donde todo lo que el usuario ve, lo puede “tocar” o manipular.

Ejercitación Resuelta capitulo I

A. Indicar si las siguientes expresiones son verdaderas o falsas.

■La disciplina de HCI trata solamente cuestiones de software.

V	F
	x

Falso. Los alcances de esta nueva disciplina son muy difusos y pueden influir en ella factores muy heterogéneos, desde aspectos tecnológicos vinculados al hardware a aspectos sociales y psicológicos del usuario.

■La construcción de una interfaz del usuario es responsabilidad exclusivamente del profesional informático.

V	F
	x

Falso. Es el desarrollador del diálogo, especialista en factores humanos, que se interesa y se encarga de todas las etapas del desarrollo del software de la interfaz del usuario, abarcando el diseño, la implementación y evaluación de la forma, contenido, estilo y secuenciación del diálogo.

■En una interfaz del usuario, puede existir un único tipo de diálogo.

V	F
	x

Falso. Pueden convivir distintos tipos de diálogos.

■La interfaz icónica es una interfaz visual.

V	F
x	

Verdadero. Es una clase específica de las interfaces visuales que utilizan como medio de representación visual exclusivamente al ícono. Se puede especificar tanto visualización icónica como programación icónica.

■El rol del desarrollador del diálogo, incluye la organización, planteo y realización de los modelos de usuarios, del contexto y de aspectos tecnológicos, que conforman la etapa de requerimientos de la interfaz.

V	F
x	

■La independencia entre la componente de interfaz y la de aplicación, sólo se debe aplicar en la etapa de programación de ambas.

V	F
	x

Falso. Se debe aplicar en toda la etapa de desarrollo del sistema de software.

■Los principios de Nielsen constituyen normas de diseño que pueden aplicarse en un proceso evaluativo de la interfaz.

V	F
x	

B. Indicar específicamente a qué componente de Green se está refiriendo en cada caso:

Particularidades de la UI	Componente de Green
Definición de las dependencias entre objetos de interacción.	*Control de Diálogo*
Diseño de las tarjetas magnéticas que permitirán el ingreso al sistema.	*Componente de Presentación*
Definición de las máscaras y formatos de entrada.	*Componente de Presentación*
Definición de las condiciones que llevan a la aparición de un mensaje de error.	*Control de Diálogo*
Diseño de las tablas de bases de datos donde se registraran los perfiles de usuario.	*Modelo Interfaz- Aplicación*

Ejercitación Resuelta capitulo II

A. Cite al menos 5 características de la interfaz de usuario que justifiquen la siguiente afirmación:

La metodología de Prototipación es adecuada para el diseño e implementación de la interfaz del usuario.

- *Los requerimientos iniciales de una interfaz de usuario son en general incompletos. A partir de ellos, y de los requerimientos funcionales se realiza un diseño inicial, que luego se irá refinando hasta llegar a la versión completa y definitiva. Cada bosquejo podrá ser evaluado por el usuario y así obtener más información de él.*
- *Durante la etapa de diseño de una interfaz de usuario deben decidirse numerosas cuestiones, como tipo de interfaz a proveer, cuestiones de visualización y comportamiento, entre otras. Involucran diferentes actores, como expertos en factores humanos, diseñadores visuales, comunicadores sociales, etc. Es por esto las interfaces son proclives a sufrir numerosos cambios hasta obtener la versión definitiva.*
- *La expresión y visualización de los conceptos de una aplicación están expuestas a cambios, que surgirán a medida que el usuario interactúe con el sistema.*
- *La comunicación entre los futuros usuarios y el equipo de desarrollo de software es fundamental, y la prototipación aporta una mejora importante en este sentido al contar con "versiones prematuras" del software.*

Permite solapar las etapas de evaluación y testeo, de manera que errores y problemas encontrados puedan ser solucionados antes de colocar el sistema en producción.

B. Cite al menos 3 tareas, que debe realizar un desarrollador de software previo al diseño de la interfaz.

- *Análisis funcional: determinar el tipo de aplicación a desarrollar y las actividades que el usuario realizará con el sistema durante la vida útil del software.*
- *Modelado de usuarios: es fundamental que el diseñador conozca al usuario e investigue sus características individuales a fin de realizar modelos cuanti y cualitativos del mismo. Conocer al usuario es la única manera de proveer una interfaz que se adecue a sus necesidades.*
- *Investigación del ambiente: contexto donde interactuará con el sistema. Entre otras cuestiones debemos dar respuesta a los siguientes interrogantes: ¿hay mucho ruido? ¿cuál es el rol del usuario dentro de la organización? ¿cuáles son las características técnicas de las máquinas donde se utilizará el sistema?*

C.Dadas las siguientes afirmaciones, responder Verdadero o Falso.

- Los testeos empíricos del usuario se llevan a cabo a través de expertos que evalúan la interfaz según estándares de diseño.
  Falso. Los testeos empíricos los llevan a cabo los usuarios potenciales.

V	F
	X

- El feedback generado por los usuarios durante las evaluaciones heurísticas refleja el grado de usabilidad y productividad del sistema.
  Falso. Las evaluaciones heurísticas las llevan a cabo expertos en HCI.

V	F
	X

- El lenguaje de programación Java provee su propio paquete para el diseño de componentes de la interfaz.
  Verdadero. La mayoría de los lenguajes visuales proveen este tipo herramientas.

V	F
X	

- En la etapa de diseño se verifica lo que el usuario conoce del dominio de la aplicación y del uso de la computadora.
  Falso. Esta verificación se realiza en la etapa de pre diseño.

V	F
	X

- Las evaluaciones de campo permiten evaluar la interfaz del sistema mientras los usan usuarios reales en entornos controlados.
  Falso. Las evaluaciones de campo son en entornos reales.

V	F
	X

Ejercitación Resuelta capitulo III

A. Indicar si las siguientes expresiones son verdaderas o falsas. Tachar lo que no corresponda:

■La manipulación directa es un tipo de diálogo secuencial.

V	F
	x

Falso. Es un paradigma de interacción visual asincrónico y libre, donde el usuario expresa sus intenciones por medio de la manipulación de los objetos de interacción.

■Para proveer múltiples hilos de diálogo entre el usuario y el sistema, es necesario que la interfaz tenga múltiples ventanas.

V	F
	x

Falso. A través del diálogo multi-thread pueden concurrir más de un hilo de diálogo en forma paralela. Por ej. un reloj que esté presente en la interfaz mientras el usuario dibuja.

■Los mensajes de error son un tipo especial de feedback.

V	F
x	

■Los menús permitieron aislar a los usuarios de cuestiones sintácticas.

V	F
x	

Verdadero. Los menús disminuyen el esfuerzo mental y el entrenamiento exigido al usuario de definir secuencias complejas de comandos, le evitan el empleo de una sintaxis restringida, y de memorizar cuestiones técnicas y del lenguaje.

■El usuario es quien debe controlar y mantener las dependencias entre los objetos relacionados.

V	F
	x

Falso. Es el sistema quien debe controlar y mantener las dependencias entre los objetos relacionados.

■Las animaciones siempre optimizan la interacción humano-computadora.

V	F
	x

Falso. Las animaciones es un recurso que es necesario diseñarlo adecuadamente a fin de no entorpecer el diálogo.

■Los browsers permiten una organización centralizada de la información.

V	F
x	

Verdadero. La técnica visual de browsers permite mostrar un gran caudal de datos, en una manera ordenada y categorizada, donde el usuario puede manejar y consultar la información en forma agrupada dentro de un contexto particular.

■El usuario al interactuar con una caja de diálogo debe percibir un criterio de ordenación en la disposición de los objetos que la componen.

V	F
x	

Verdadero. La forma de ordenación y disposición de los elementos de una caja de diálogos debe ser significativa.

B - La interfaz del usuario es el software que soporta el diálogo bidireccional entre el usuario y el sistema.

Indicar con una cruz, qué dirección del diálogo asocia cada uno de los paradigmas de interacción: si va del usuario al sistema, del sistema al usuario, en ambas direcciones o en ninguna.

Paradigmas de interacción visual	
Múltiples ventanas	x
Menús	x
Manipulación directa	x
Feedback	x
Dependencias	x
Animaciones	x
Browsers	x
Cajas de diálogo	x

Ejercitación Resuelta capitulo IV

1. ¿Cómo evidencia el diseño icónico? Identificar, marcando con un círculo, los elementos icónicos de la interfaz de la pantalla inicial del sistema para una Veterinaria.

FIGURA 4.17

2. ¿Qué mecanismo de interacción le parece adecuado? Proponer al menos tres actividades que podrá realizar el usuario final del sistema e indicar como las llevaría a cabo

El mecanismo de interacción adecuado es el sugerido para este tipo de interfaz: arrastrar y soltar.

- *En la pantalla inicial se visualiza claramente el escritorio, donde la zona en blanco invita a arrastrar hacia allí las fichas en blanco (a la derecha) para dar de alta un nuevo paciente.*
- *En la ficha de un paciente, solapa de datos de medicación, arrastrar el frasco de pastillas a la zona de prescripciones habilitará un nuevo renglón para completar una medicación con remedios en presentación de pastillas.*
- *Arrastrar el catálogo de productos al escritorio permite acceder a la base de datos de productos a la venta.*

3. De acuerdo a sus respuestas dadas en el punto anterior, ¿qué características debería tener el usuario final para obtener el mayor beneficio posible del sistema?

El usuario debe tener conocimientos en informática y del dominio del problema, así como también todas sus aptitudes psicomotrices en condiciones óptimas.

4. De acuerdo a las recomendaciones de diseño vistas en este capítulo, ¿qué problemas encuentra? ¿cómo los solucionaría?

Los principales problemas se encuentran en:

- *Reconocimiento inmediato: ciertos íconos no son muy representativos, por ejemplo los íconos para representar la sala de terapia intensiva y la de observaciones. Se podrían solucionar diseñando animaciones o simulaciones.*
- *Entendimiento universal: los íconos mencionados previamente tampoco respetan esta recomendación.*
- *Transparencia: en la ficha de un paciente, solapa de datos de medicación no es claro que tiene que arrastrar los remedios de la derecha al panel de la izquierda para que se habilite un nuevo renglón de carga de datos.*

Ejercitación Resuelta Capitulo V

A. Citar al menos cinco diferencias entre diseñar la interfaz de un sitio Web y la interfaz WYSIWYG (What You See Is What You Get).

- *Comunidad destino: considerar que los usuarios pueden acceder desde diferentes lugares, distintos medios y mediante diferentes recursos tecnológicos.*
- *Objetos de interacción.*
- *Regla de 3 clicks: no más de tres clicks para alcanzar el objetivo.*
- *Alto grado de actualización y posibles rediseños.*
- *Tener en cuenta cuestiones sobre la red, como ancho de banda y disponibilidad.*

B. Responde verdadero o falso.

- El tipo de información que se publica en un sitio Web no afecta al diseño de la interfaz del mismo.

V	F
	x

Falso. La información a mostrar debe ser tenida en cuenta a fin de clasificar el contenido y realizar una sectorización adecuada.

- Al diseñar un sitio Web hay que pensar en un perfil de usuario general

V	F
	x

Falso. Es necesario modelar a los lectores destinatarios, trabajando con perfiles de usuarios.

- Es recomendable enriquecer el sitio Web con presentaciones Flash muy coloridas y animadas.

V	F
	x

Falso. Numerosos estudios han demostrado que utilizar este tipo de recursos complican la interacción y hasta son considerados difíciles de interactuar.

- Realizar testeos empíricos del sitio Web forma parte de la Ingeniería de Usabilidad

V	F
x	

Verdadero. Permiten medir la adecuación del diseño. Se emplean durante y al final del desarrollo iterativo del sitio Web.

- Es fundamental facilitar el rediseño de los sitios Web

V	F
x	

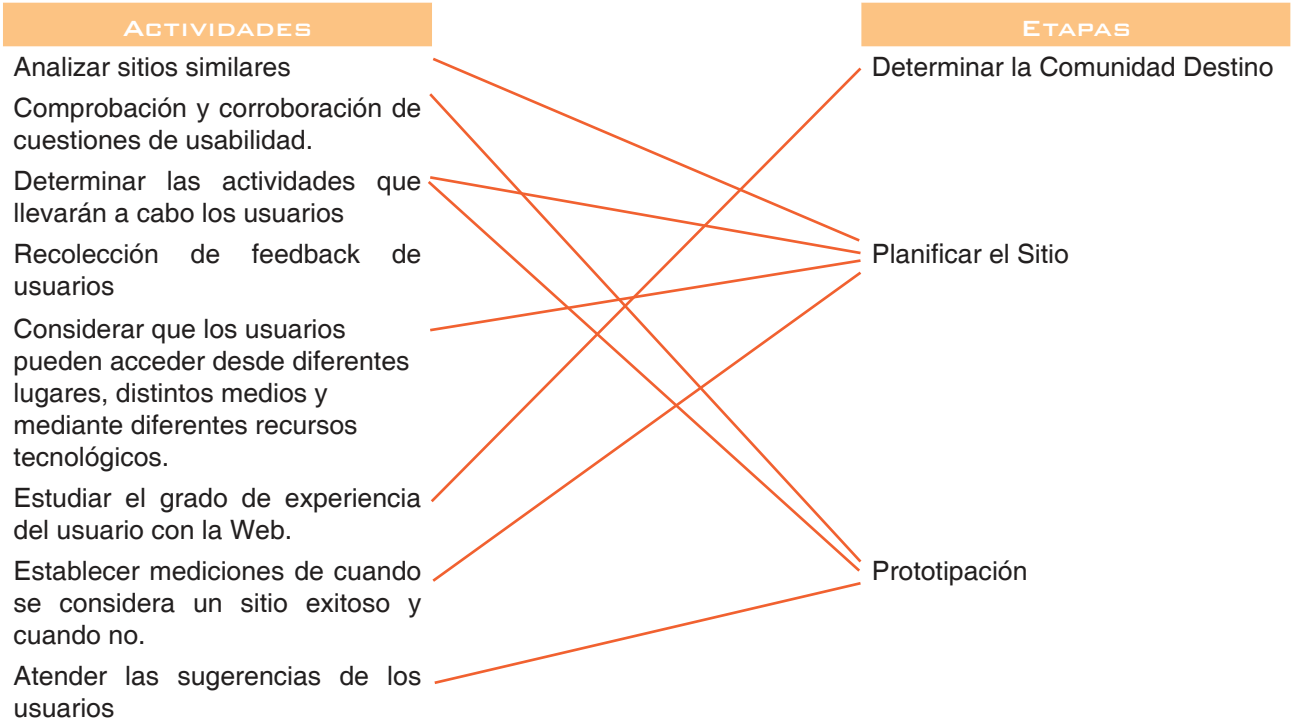
Verdadero. Los sitios se encuentran en constante cambio.

- Una vez publicado el sitio no es necesario tomar muestras de la interacción del usuario con el mismo.

V	F
	x

Falso. Es necesario recolectar el feedback de los usuarios para determinar si cumple con los medidas de éxito propuestas, y si es necesario realizar ajustes o no.

C. Unir con flechas la actividad de la izquierda con las etapas previas al diseño del sitio Web de la derecha.



Ejercitación Resuelta Capitulo VI

A. Responde verdadero o falso.

■La accesibilidad apunta únicamente a las diferentes discapacidades de un usuario.

V	F
	x

Falso. La Accesibilidad en la Web significa que todo sitio debe ser construido y diseñado de manera tal, que pueda ser accedido por cualquier persona, independientemente de su condición física, mental, recursos tecnológicos que posea como versiones de navegador, tipo de monitor, formas de conexión que utiliza para acceder a la Web. y desde diferentes dispositivos, mediante acceso móvil como celular o PALMS, desde el auto, etc.

■La WAI, pertenece a la W3C y se ocupa específicamente de establecer y publicar normas de accesibilidad, para garantizar el diseño universal del sitio y el uso de elementos estándares.

V	F
x	

Verdadero

■Las ayudas técnicas hacen referencia a las herramientas de adaptación que utilizan los usuarios con distintas discapacidades.

V	F
x	

Verdadero

■El diseño universal intenta abolir el uso de multimedia en el diseño de sitios Web

V	F
	x

Falso. Las pautas sobre diseño universal no desalientan a los desarrolladores en la utilización de imágenes, vídeo, ni de otros recursos multimediales, por el contrario explican cómo hacer los contenidos multimedia más accesibles a una amplia audiencia. Se apoyan en el uso alternativo de texto para que las ayudas técnicas o adaptaciones informáticas puedan tener información adicional de esos recursos multimediales.

■La guía de accesibilidad WCAG 2.0 tiene la misma organización que la WCAG 1.0

V	F
	x

Falso. La Guía de Accesibilidad WCAG 2.0 provee una organización en capas de recomendación, las cuales incluyen 4 principios generales, 12 pautas, criterios de comprobación o de éxito dentro de estas pautas, y un conjunto de técnicas obligatorias y de asesoramiento. Mientras que la Guía WCAG 1.0 se organiza en 14 pautas que incluye puntos de verificación.

B. Indique los problemas que pueden surgir cuando un usuario ciego quiere instalar un producto que lo asista en su navegación por la Web.

■*Problemas de Incompatibilidad: El usuario discapacitado a veces debe sufrir complicaciones de incompatibilidad entre las distintas herramientas que él instala en su ordenador. Hay ciertas funciones del sistema operativo o del navegador subyacente, que pueden generar conflictos con los programas ya instalados.*

■*Generación de Versiones más avanzadas de un sitio: A medida que la tecnología avanza, proporcionalmente se desarrollan versiones de sitios más complejos, que incluyen componentes FLASH, estilos, applets, layers. Muchas veces, las herramientas que posee el discapacitado queda obsoleta, teniendo que adquirir y aprender otras herramientas o nuevas versiones de la misma.*

■*Manejo Heterogéneo de los Periféricos: El usuario discapacitado en su interacción con sus diferentes herramientas de adaptación instaladas, más el sitio que está accediendo, debe recordar qué teclas usar, si se le permite o no usar el teclado para acceder a los hipervínculos, qué dispositivo emplear en un determinado momento.*

■*Falta de ayudas o tutoriales accesibles: hay herramientas de adaptación que tienen escasa ayuda, que es totalmente insuficiente y que además no son accesibles.*

■*Problemas de Portabilidad: Los lugares de concurrencia masiva no están preparados para ser utilizados por personas con discapacidad. No suelen tener instalado adaptaciones informáticas.*

C. Indique las comprobaciones necesarias desde el punto de vista de accesibilidad, que un desarrollador Web debe realizar antes de publicar su sitio Web.

■*Revisar el sitio con herramientas de evaluación de accesibilidad automáticas como el TAW. Tener en cuenta que a veces detecta advertencias que deben ser comprobadas en forma manual.*

- *Solucionar los problemas detectados por el revisor hasta alcanzar un nivel de conformidad de accesibilidad. Tener en cuenta que implementar la accesibilidad puede implicar modificación de código, alteración del aspecto visual del sitio como también la construcción de páginas alternativas accesibles.*
- *Validar el sitio con diferentes resoluciones de monitores*
- *Probar el sitio desde diferentes navegadores y tipos de conexión*
- *Utilizar el sitio totalmente con el teclado*
- *Comprobar la operabilidad del sitio con alguna herramienta de adaptación, como el JAWS, magnificador de pantalla, lector de documentos, entre otros.*

Acerca de los autores

Lic. Francisco Javier Díaz es Licenciado en Matemática Aplicada de la Universidad Nacional de La Plata (UNLP) así como también posee el título de Calculista Científico de dicha Universidad. Es Investigador Adjunto sin director de la Comisión de Investigaciones Científicas de la Provincia de Bs. As. (CIC-BA) desde 1990.

Fue el primer Decano electo de la Facultad de Informática de la Universidad Nacional de La Plata en 1999 y actualmente mantiene el cargo en dicha Facultad, habiendo sido reelecto en el año 2004.

Actualmente es el Director General, Científico y Tecnológico del Centro de Procesamiento de la Información (CeSPI) de la UNLP. Este centro se encarga de la administración de la red académica, administrativa y de investigación de la UNLP y tiene además la responsabilidad de administrar la conexión de esta red con el Backbone (apoyo) RIU nacional y con el enlace Internacional a Internet. Al mismo tiempo el CeSPI maneja los servicios administrativos de la Universidad, por ejemplo el sistema de alumnos y la liquidación de sueldos.

Es Profesor Titular Ordinario de la Facultad de Informática de la UNLP. El Lic. Díaz es asimismo Director del Laboratorio de Investigación de Nuevas Tecnologías en Informática (LINTI) de la Facultad de Informática de la UNLP, en el que trabajan 50 personas dedicadas a la investigación y es la base de 10 cátedras pertenecientes a la carrera de Licenciatura en Informática y también es el sostén de estudios prácticos en el Master de Redes de Datos, del que el Lic. Díaz es también director.

http://www.linti.unlp.edu.ar

Lic. Ivana Harari es Licenciada en Informática de la Universidad Nacional de La Plata (UNLP/1992) y Especialista en Docencia Universitaria (UNLP/2012). Se encuentra desarrollando la tesis final de la Maestría en Redes de Datos (UNLP/2002). Trabaja en el de tema de investigación denominado “Interfaces Hombre Computadora, Accesibilidad y Usabilidad”. Como integrante del Laboratorio de Investigación en Nuevas Tecnologías Informáticas de la UNLP, ha participado de numerosos proyectos de evaluación de sistemas desde el punto de vista de comunicación hombre–máquina, para distintas organizaciones públicas y privadas. Ha dirigido tesinas de Licenciatura en Informática y de Maestría. Ha dictado cursos sobre la temática en distintas universidades nacionales, como la Universidad Nacional de La Patagonia Austral y la Universidad Nacional del Litoral. Ha publicado artículos en prestigiosos eventos científicos nacionales e internacionales. Actualmente se desempeña como Profesor Adjunto de la cátedra Diseño Centrado en el Usuario y de Interfaces adaptadas para dispositivos móviles de la Facultad de Informática de la Universidad Nacional de La Plata.

http://www.linti.unlp.edu.ar

Lic. Ana Paola Amadeo es Licenciada en Informática de la Universidad Nacional de La Plata (UNLP/2001) y Profesor Autorizado de la UNLP (2007). Actualmente se encuentra realizando el Doctorado en Ciencias Informáticas en la UNLP. Como integrante del Laboratorio de Investigación en Nuevas Tecnologías Informática - LINTI, ha participado de proyectos de evaluación de sistemas desde el punto de vista de comunicación hombre – máquina, para distintas organizaciones públicas y privadas. Ha dictado cursos sobre la temática en distintas universidades nacionales, como la Universidad Nacional de La Patagonia Austral y la Universidad Nacional del Litoral. Ha publicado artículos en prestigiosos eventos científicos y revistas, nacionales e internacionales. Actualmente se desempeña como Jefe de Trabajos Prácticos de la cátedra Diseño Centrado en el Usuario de la Facultad de Informática de la Universidad Nacional de La Plata y Profesor Adjunto de Seminario de Lenguajes y Proyecto de Desarrollo en la misma unidad académica.

http://www.linti.unlp.edu.ar