

Usabilidad

PID_00167607



Universitat Oberta
de Catalunya

www.uoc.edu

Índice

1. Principios generales de usabilidad	5
1.1. Visibilidad del estado del sistema	5
1.2. Consistencia	6
1.3. Control del usuario	8
1.4. Prevención de errores	8
1.5. Estructura visible	11
1.6. Interfaz explorable	12
1.7. La Ley de Fitts	13
1.8. Modalidad	14
1.9. Metáforas	15
1.10. Uso del color	15
1.11. Mensajes de error	16
1.12. Tiempos de respuesta	18
2. Usabilidad y legibilidad de contenidos	20
2.1. Pautas para la legibilidad	22
3. Evaluación de la usabilidad	24
3.1. Paseo cognitivo	24
3.2. Análisis de tareas	26
3.3. Grupos focales	26
3.4. GOMS	27
3.5. Inspección de usabilidad	27
3.6. Test con usuarios	28
4. Evaluación heurística	29
4.1. Desarrollo	29
4.2. Recomendaciones	30
4.3. Duración	31
4.4. Heurísticas	31
5. Paseo cognitivo	33
5.1. Preparación	33
5.2. Desarrollo	34
5.3. Duración	34
5.4. Recomendaciones	34
6. Realización de un test con usuarios	36
6.1. Preparación	36
6.2. Desarrollo	38
Bibliografía	41

1. Principios generales de usabilidad

El éxito de una aplicación interactiva es determinado por diversos factores (calidad artística, interés intrínseco de los contenidos, etc.), aunque existe una condición ineludible: el usuario debe poder alcanzar fácil y cómodamente sus objetivos.

El concepto de **usabilidad** se refiere a la facilidad de uso de una aplicación interactiva o, como define el estándar ISO 9241:

"Usabilidad es la medida en que un producto puede ser utilizado por determinados usuarios para conseguir unos objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso definido."

La ISO 9126-1 propone una definición similar: "Usabilidad es la capacidad de un software para ser comprendido, aprendido y utilizado y para resultar atractivo al usuario, cuando se utiliza en condiciones determinadas."

En ambos casos, es importante resaltar que la usabilidad se refiere a unas condiciones de uso específicas: no todas las aplicaciones interactivas pueden ser diseñadas siguiendo exactamente los mismos criterios de usabilidad, ya que la naturaleza misma de la aplicación, el tipo de usuario al que se dirige, o el contexto de utilización son factores determinantes. En este sentido, la usabilidad se compone tanto de **atributos objetivos** (tiempo empleado por el usuario para conseguir un objetivo, número de errores cometidos,...), como de **atributos subjetivos** (satisfacción de uso).

Algunos autores han establecido unos principios fundamentales o **heurísticas de usabilidad** para el diseño de interfaces de entorno gráfico, aplicables tanto a aplicaciones interactivas como a sitios web. Parten de la premisa de que el usuario no debe verse obligado a aprender rutinas complejas, sino que ha de poder navegar de la manera más intuitiva, rápida y efectiva posible.

1.1. Visibilidad del estado del sistema

La **autonomía del usuario** se basa en la percepción por parte de éste de que ostenta el control de la aplicación; para ello, debe disponer de información continua y clara sobre el estado del sistema, sin que éste ejecute acciones automáticas o aleatorias.

Los **mecanismos de información sobre estado** son necesarios para que el usuario pueda adaptar sus acciones a los cambios producidos en el sistema. En caso de que estos mecanismos fallen, se estará forzando a sobrecargar la memoria a corto plazo con datos temporales, de manera que se limitará la capacidad del usuario para planear nuevas acciones.

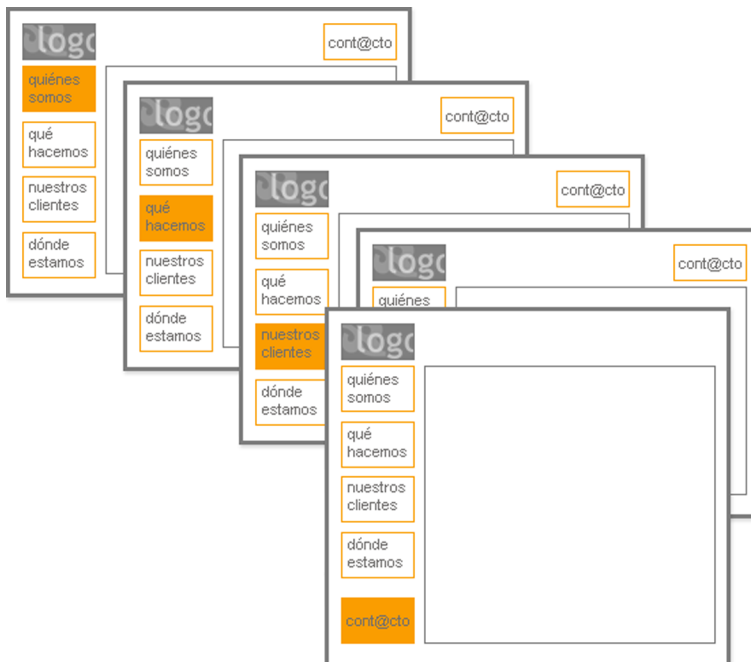
La información de estado debe ser constantemente actualizada y su representación visual ha de orientarse a facilitar la comprensión.

Por otra parte, el usuario no se siente cómodo cuando no puede percibir dónde están los **límites del sistema**. Es tan inadecuado "asfixiar" al usuario restringiendo excesivamente su libertad de acción, como abandonarlo en un contexto demasiado amplio, que él pueda interpretar como inmanejable. Internet es un buen ejemplo de esta situación: la incapacidad para predecir los límites del entorno puede producir desorientación, ansiedad e inseguridad en el usuario.

1.2. Consistencia

El respeto de la consistencia tanto en la forma como en la función es un factor clave en la usabilidad de una interfaz; el comportamiento de los elementos de una aplicación debe ser **constante** y **predecible**.

La consistencia supone la fijación de unas **constantes representativas** a lo largo de una aplicación, de manera que el mismo tipo de información llegue al usuario siempre de idéntica forma.



La consistencia también se refiere a la **experiencia del usuario**. Se facilita el uso de una aplicación cuando se respetan los conceptos de diseño que se han convertido en estándares en el entorno cultural del usuario.

Como señala Shneiderman, la consistencia debe respetarse en:

- 1) **Acciones:** para tareas similares, el usuario debe poder ejecutar la misma secuencia de acciones.
- 2) **Terminología:** los conceptos utilizados en menús, contenido, ayuda, etc., deben mantenerse a lo largo de toda la aplicación.
- 3) **Elementos gráficos:** la retícula, la gama de colores, la aplicación de tipografía y otros elementos gráficos deben mantenerse constantes en todo el sistema.

Según Tognazzini, la importancia de mantener la consistencia varía según el **tipo de aplicación**. Este autor expone una lista en la que los primeros elementos exigen mayor vigilancia de la consistencia que los últimos:

- Interpretación del **comportamiento del usuario**. Por ejemplo, los atajos de teclado deben mantener su significado.
- **Estructuras invisibles**. Se refiere a acciones asociadas de manera no visible a elementos de la interfaz. Por ejemplo, si el usuario puede redimensionar la ventana de la aplicación arrastrando su contorno, este control debe mantenerse siempre (no sería coherente que unas veces pudiera hacerlo y otras veces no, sin que exista un elemento gráfico que lo indique).
- **Pequeñas estructuras visibles**. Se refiere a los iconos, barras de desplazamiento, etc., que deben mantener su apariencia. Generalmente, la posición de estos elementos en pantalla también debe respetarse.
- **Elementos de diseño**, es decir, el estilo visual que define la aplicación.
- **Un conjunto (o *suite*) de productos**. Se refiere a un conjunto de aplicaciones que forman parte del mismo sistema (Microsoft Office, por ejemplo, es una *suite* compuesta por Excel, Word, PowerPoint y otras utilidades). La consistencia debe mantenerse entre todas las aplicaciones que forman parte de la *suite*, aunque con las variaciones adecuadas para favorecer la flexibilidad de funciones.
- **Instalación**. Deben preverse las variaciones que pueda sufrir la aplicación al distribuirse. Por ejemplo, si se utilizan fuentes que dependan del sistema del usuario, no deberían existir variaciones importantes según las condiciones de instalación.

- **Plataforma.** Se refiere al mantenimiento de la consistencia en plataformas distintas, como Apple/Windows.

1.3. Control del usuario

El sistema debe **adaptarse siempre al usuario**. No puede obligarse al usuario a investigar cómo puede realizar sus tareas, como consecuencia de un diseño complejo o poco intuitivo.

El usuario debe disponer de toda la información que necesite, y de las herramientas necesarias para ejecutar cada acción prevista.

Las acciones del usuario no pueden dar lugar a **consecuencias imprevistas**. Por ejemplo, no pueden asociarse acciones al movimiento del cursor en pantalla, porque el usuario ha de tener la capacidad de desplazar el cursor con toda libertad.

En casos especiales, como en el diseño de juegos interactivos, en los que se motiva al usuario a tener una actitud explorativa, este principio incrementa su complejidad al obligar al sistema a mantener la previsión de respuestas para todas las acciones posibles, más aleatorias y experimentales que en una aplicación normal.

1.4. Prevención de errores

El diseñador debe utilizar una metodología de prevención de errores que disminuya tanto como sea posible la posibilidad de acciones equivocadas por parte del usuario. Algunos de los puntos que cabe tener en cuenta son:

- Los **campos numéricos** no deben aceptar la introducción de caracteres alfabéticos.
- En **campos de introducción de texto**, deben aparecer por defecto los valores más probables (si son previsibles), o valores de ayuda.

Campos de fecha de nacimiento

Mi perfil

Bienvenido a tu cuenta personal de Nike. Aquí guardamos toda tu información de forma segura. Continúa y actualiza tu información.

Información Personal	Inscripción	Direcciones	Historial/Seguimiento de Pedidos
Información Personal *Nombre <input type="text"/> *Apellido <input type="text"/> *Sexo <input type="radio"/> Masculino <input type="radio"/> Femenino *Fecha de nacimiento <input type="text" value="DD"/> <input type="text" value="MM"/> <input type="text" value="YYYY"/> <small>Menor de 13 años. Leer información</small> Nombre Nike (Opcional) <input type="text"/> <small>(Nombre de pantalla)</small> *Correo electrónico <input type="text"/> <small>(Este será tu nombre de usuario)</small> Número de móvil (Opcional) <input type="text"/>		CONTRASEÑA *Contraseña <input type="password"/> *Vuelve a introducir la nueva contraseña <input type="password"/> VERIFICACIÓN DE TU CUENTA Por favor, selecciona una pregunta de momento olvidas tu contraseña. *Elige una pregunta <input type="text" value="Escoge una pregunta"/> *Respuesta de seguridad <input type="text"/> <input type="checkbox"/> Información general de Nike y Mantenimiento <input type="checkbox"/> Mantenedme informado de nuevas ofertas. <input type="button" value="CANCELAR"/> <input type="button" value="ENVIAR"/>	
DIRECCIÓN *País <input type="text" value="Escoge un país"/> <input type="button" value="▼"/> Dirección 1 <input type="text"/> Dirección 2 (Optional) <input type="text"/> Ciudad <input type="text"/> Código Postal <input type="text"/>			

Los campos de fecha de nacimiento contienen valores de ayuda, que orientan al usuario sobre el formato de los datos.
 Fuente: <http://store.nike.com/emeastore/?#,es,ES,;stage,frontpage---//////0>

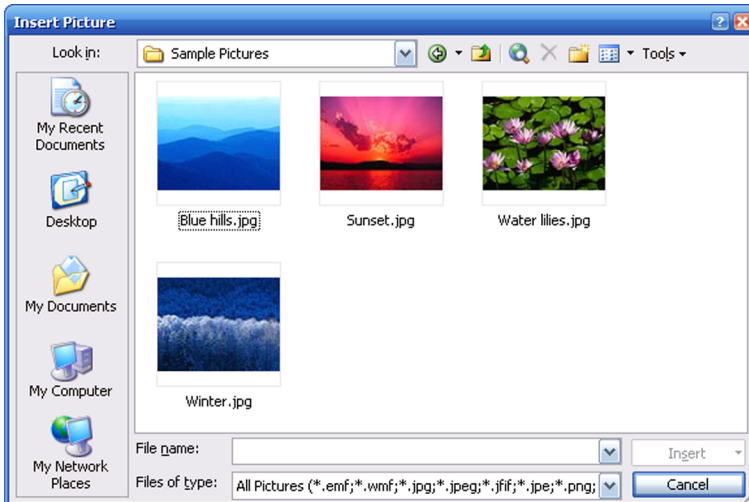
- En **opciones de selección de varios ítems** es recomendable utilizar listas o menús desplegables para que el usuario no tenga que introducir caracteres mediante el teclado.

DIRECCIÓN	
*País	<input type="text" value="Escoge un país"/> <input type="button" value="▼"/>
Dirección 1	AFGANISTÁN
Dirección 2 (Optional)	ALBANIA
Ciudad	ALEMANIA
Código Postal	ANDORRA
	ANGOLA
	ANGUILA

Fuente: <http://store.nike.com/emeastore/?#,es,ES,;stage,frontpage---//////0>

- Para opciones de **gestión de ficheros**, es recomendable mostrar una lista de los ficheros seleccionables, de manera que el usuario no tenga que teclear su nombre.

Opción de insertar imagen, de Word



- En aplicaciones complejas, es recomendable proporcionar al usuario **ayuda sensible al contexto**.

Menú contextual de Photoshop



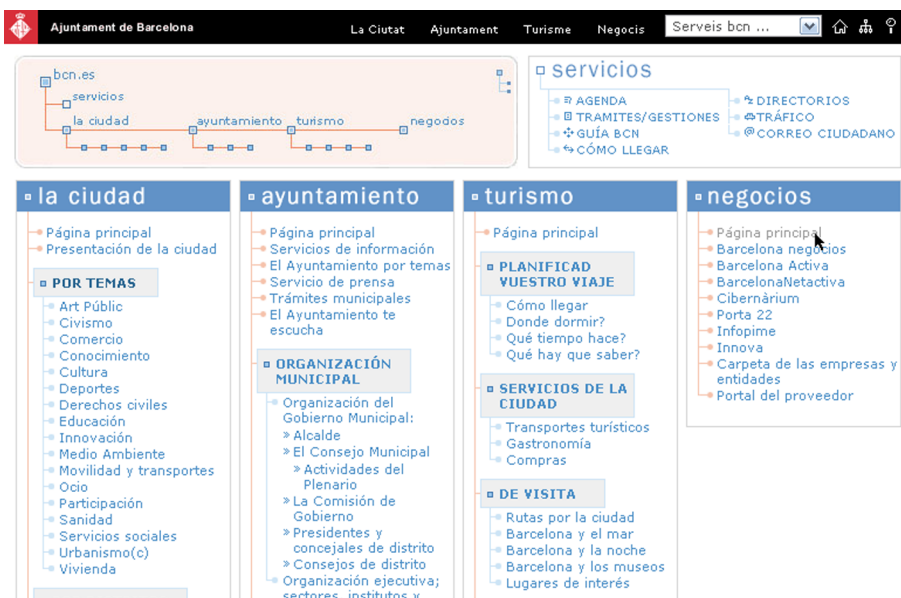
Los usuarios nunca deberían perder el trabajo efectuado por culpa de errores propios o del sistema. Por ejemplo, no se debe permitir la salida de una aplicación en la que se puedan trabajar procesos sin preguntar antes si se quieren **guardar los cambios**.

Por otra parte, para aplicaciones con contenido extenso, es aconsejable prever un **historial** para que el usuario pueda reconocer los apartados por los que ha pasado y acceder a ellos con facilidad. Si el usuario sale de una aplicación de este tipo, debe permitírsele retornar con facilidad al último punto visitado.



1.5. Estructura visible

No se puede exigir al usuario que construya por sí mismo un mapa de la aplicación. Por lo tanto, deben preverse **índices o mapas** que representen la estructura del sistema, y que permitan acceder a los diferentes apartados.



Fuente: <http://www.bcn.es/castella/mapaweb/emapaweb.htm>

El contenido de la aplicación debe organizarse de manera que la estructura sea lo más plana posible, evitando la existencia de excesivos **niveles de opciones**.

El establecimiento de una **estructura visible** es especialmente importante en sitios web, ya que el usuario se encuentra navegando en un entorno ilimitado (Internet). Es aconsejable provocar en el usuario la percepción de que se mantiene en el mismo sitio, y que los contenidos aparecen en una estructura constante.



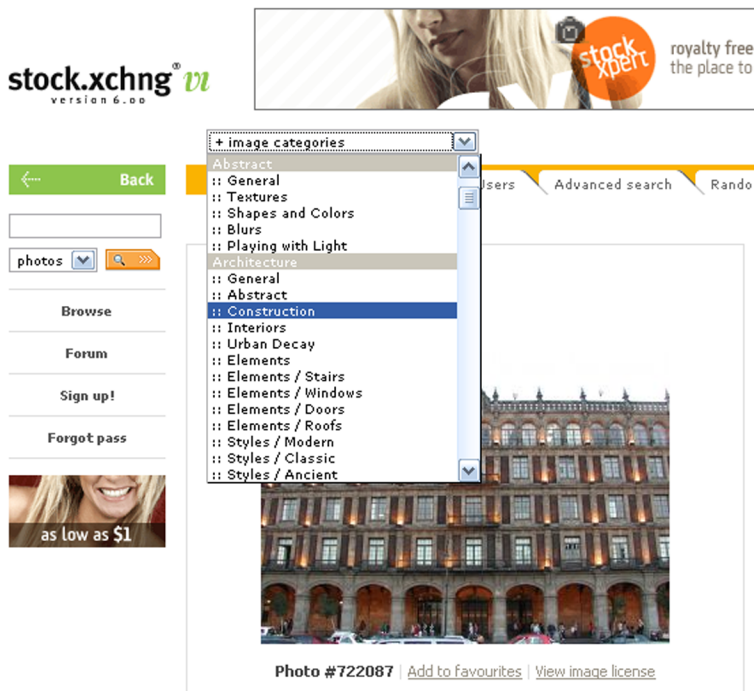
1.6. Interfaz explorable

La estructura de una aplicación debe estar diseñada de manera que el usuario sepa qué rutas existen, y cómo llegar hasta cualquier punto del sistema. No obstante, esto no implica que se fuerce al usuario a seguir un camino definido, impidiéndole cualquier otra acción. El usuario debe sentir que tiene **libertad** para navegar por el sistema, y que no va a caer en **caminos sin salida**.

La capacidad de explorar se ve favorecida por la **reversibilidad de las acciones**: si el usuario sabe que puede anular cualquier acción que haya ejecutado, tendrá más libertad para experimentar nuevas rutas. En caso contrario, la amenaza de cometer errores puede cohibir sus acciones.

Junto con la capacidad para investigar el entorno, debe existir una **ruta rápida** para acceder a los contenidos, de manera que tanto el usuario que requiere rapidez como el que quiere explorar el entorno puedan sentirse satisfechos.

Menú desplegable para el acceso rápido a los contenidos



Fuente: <http://www.sxc.hu/photo/722087>

Si la aplicación está dirigida a **usuarios inexpertos**, las rutas deben estar mucho más dirigidas, ya sea limitando el número de posibilidades o mediante una ayuda exhaustiva.

La **consistencia visual** juega un papel fundamental en este aspecto: aunque el usuario se mueva con toda libertad por la aplicación, debe sentir que siempre está en el mismo entorno. Esto es especialmente importante en el caso de sitios web, en los que el usuario debe poder identificar siempre el entorno particular en que se encuentra.

1.7. La Ley de Fitts

Según la ley de Fitts, "el tiempo requerido para conseguir un objetivo es proporcional a la distancia y el tamaño del objetivo". En este sentido, las opciones más importantes deben tener **mayor tamaño** o ser **más visibles** que las secundarias.

También la **localización** es importante. Los cuatro lados de una ventana son las zonas más fácilmente accesibles para el usuario. Obsérvese que en los entornos Mac y Windows, las opciones principales suelen colocarse en barras situadas en los límites de pantalla.



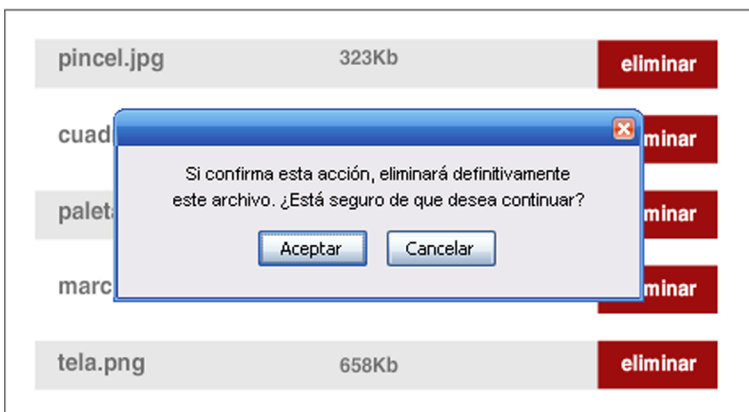
1.8. Modalidad

Los modos sirven para **contextualizar** temporalmente las acciones del usuario.

Un comportamiento modal mal diseñado implica restricciones graves a la libertad de acción del usuario. Los primeros interfaces de usuario eran altamente restrictivos: debía seleccionarse siempre el modo de trabajo (por ejemplo, "modo Copiar") y después ejecutar la acción ya contextualizada (por ejemplo, seleccionar el fragmento del texto que hay que copiar). Actualmente, la modalidad en entornos gráficos suele favorecer el aprendizaje y la facilidad de uso de una aplicación.

Las ventanas de diálogo modales suspenden temporalmente todas las acciones, y obligan al usuario a responder a la cuestión expuesta en la ventana; deberían aparecer solamente en procesos breves.

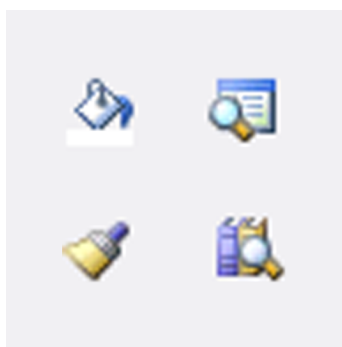
Lista de archivos:



1.9. Metáforas

Cuando el usuario se encuentra trabajando con un sistema complejo, construye un modelo mental del mismo para imaginar cómo está organizado. Este modelo le permite anticipar el comportamiento del sistema sin tener que memorizar reglas abstractas. El diseño de la interfaz debe trabajar previendo cuál es el modelo mental adecuado.

Las interfaces gráficas utilizan las metáforas visuales para hacer inteligibles las funciones del sistema mediante elementos que remiten al **mundo real**. Las funciones son representadas a través de objetos que resultan familiares al usuario y que ostentan un comportamiento similar en su entorno habitual.



Las metáforas en las que se basa una interfaz deben estar bien seleccionadas, en el sentido de que el usuario pueda interpretar correctamente el modelo conceptual de la aplicación. Asimismo, su diseño debe ser **consistente** (mismas funciones, mismas metáforas). El uso de metáforas incorrectas o inconsistentes no solamente resulta inútil, sino que entorpece la navegación.

El diseñador debe tener en cuenta que la **perspectiva del usuario** puede ser diferente de la suya, de manera que lo que resulta intuitivo para el diseñador puede no serlo para el usuario. El **modelado del usuario** y la **evaluación** de la interfaz por parte de usuarios inexpertos son fundamentales, ya que ayuda a descubrir problemas de este tipo.

1.10. Uso del color

El color puede utilizarse con varios propósitos: atraer la mirada del usuario hacia un punto concreto en pantalla, distinguir elementos en gráficos complejos, organizar la información, enfatizar los mensajes de alerta o sugerir un tono emocional.

Ved también

El modelado del usuario se estudia en el apartado 6 del módulo "Diseño centrado en el usuario".

Ved también

La evaluación de la usabilidad se estudia en el apartado 3 de este mismo módulo.

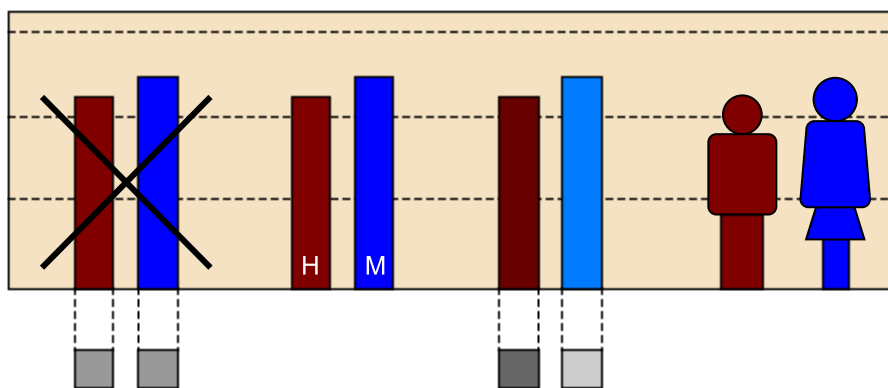


No obstante, se calcula que el 10% de la población sufre alguna alteración en la percepción del color. Los problemas más extendidos se asocian a la discriminación de los tonos de rojo y verde, aunque también pueden existir dificultades para reconocer el azul.

Por ello, cuando el color se utiliza para **representar datos**, debe acompañarse de algún recurso que garantice que los usuarios con problemas de percepción visual pueden distinguir adecuadamente las categorías. Estos recursos pueden consistir en etiquetas textuales, en la diferenciación de niveles de gris o en la variación del contorno.

Ved también

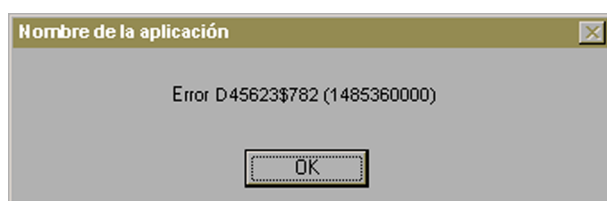
La representación de datos se estudia en el apartado 11 del módulo "Diseño".



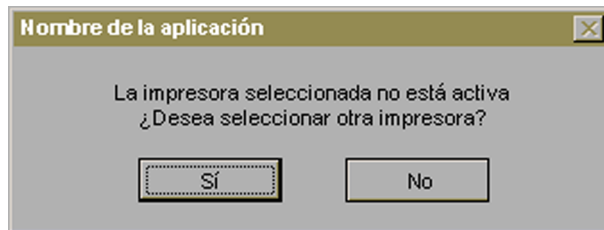
1.11. Mensajes de error

La **prevención efectiva** reduce el número de ocasiones en que es necesario mostrar un mensaje de error. Aun así, puede ser necesario incluir alguno de estos mensajes, que deben tener en cuenta los siguientes principios:

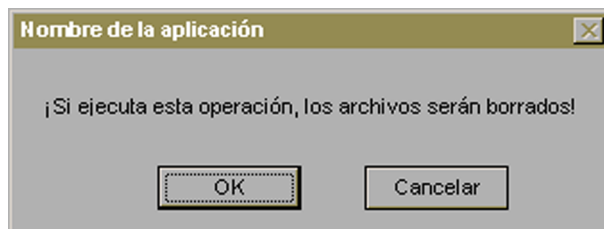
- El mensaje debe describir el problema en **términos sencillos y ser positivo y explicativo**. El usuario no tiene por qué conocer la terminología informática, o las causas técnicas por las que el sistema no puede llevar a cabo una acción. Un mensaje demasiado genérico ("Error de sintaxis" en lugar de "Falta cerrar el paréntesis") o excesivamente técnico no tiene ninguna utilidad, y es desfavorable.



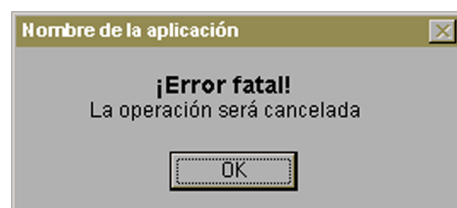
- Generalmente basta con indicar al usuario cuáles son sus **opciones** para resolver el error. Una manera adecuada de presentar un mensaje de error consiste en formular una pregunta clara y breve, a la que el usuario pueda responder con varias opciones.



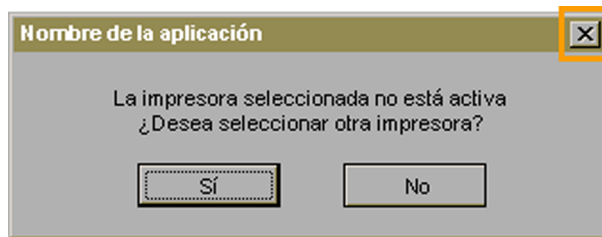
- Debe evitarse la utilización de **signos de exclamación** en el mensaje, o su construcción en **letras mayúsculas**. Ambos recursos son visualmente similares al grito verbal, de manera que causan alarma e incomodidad en el usuario.



- La palabra "**error**" debe evitarse. No contiene información, y conduce a que el usuario se sienta culpable e inseguro.



- Las **señales auditivas** deben utilizarse con moderación para no causar alarma excesiva o saturación. Es aconsejable aplicarlas solamente a situaciones críticas. En todo caso, el usuario ha de poder controlar su volumen o desactivarlas.
- Todos los mensajes de error deben incluir una opción clara que permita **cerrar la ventana**.



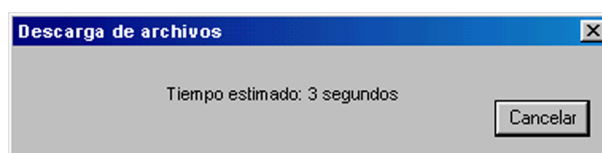
1.12. Tiempos de respuesta

La respuesta del sistema a las acciones del usuario debe ser **inmediata** y **clara**, y se presenta en forma de **señales visuales o auditivas** que indiquen que la acción del usuario ha sido detectada, y que se ha iniciado el proceso solicitado. Cualquier retraso en la respuesta puede conducir a engaño al usuario en dos sentidos: o bien el sistema no ha detectado la acción, o bien la relación acción-efecto no se había interpretado correctamente.

Las tareas que lleve a cabo el sistema no deben impedir que el usuario continúe con sus acciones habituales.

Para conseguir que las tareas del ordenador no supongan que el usuario experimente la sensación de que el ordenador está bloqueado, pueden utilizarse los siguientes recursos:

- Todas las opciones de menú, iconos, etc. deben ofrecer una **respuesta visual inmediata**.
- El usuario ha de poder **cancelar** cualquier proceso.
- Es aconsejable que aparezca un **indicador gráfico de espera** (por ejemplo, un reloj) para cualquier acción que pueda durar más de $\frac{1}{2}$ segundo. Este indicador debe estar animado para que no parezca que el sistema está bloqueado.
- Debe aparecer un mensaje que indique la **duración** para cualquier proceso que dure más de 2 segundos.



- El **indicador de progreso** puede mostrar el estado del proceso mediante una barra animada u otro recurso similar. Puede acompañarse de un indicador numérico de porcentaje.
- Los **mensajes textuales** que indican qué tarea se está llevando a cabo en cada momento ofrecen confianza y motivación al usuario.
- Cuando el proceso supere los **10 segundos**, es aconsejable que su finalización e indique mediante un sonido y un aviso visual, de manera que el usuario sepa que ya dispone de los resultados.
- Debe bloquearse la posibilidad de que el sistema inicie un proceso repetidamente después de varios clics sucesivos en un mismo botón. Esto puede ocurrir con frecuencia cuando se trata de usuarios inexpertos o en entornos de ejecución lenta.
- Las **respuestas auditivas** deben utilizarse moderadamente para no producir saturación, especialmente si se prevé que la aplicación se instalará en un entorno en el que pueda ejecutarse simultáneamente en varias estaciones de trabajo. Las señales auditivas son especialmente útiles en caso de avisos que prevengan de la posibilidad de error grave como consecuencia de una acción del usuario. El usuario debe poder desactivar estas señales, o controlar su volumen.

2. Usabilidad y legibilidad de contenidos

El texto que muestra el contenido de una aplicación debe aparecer con un **contraste** adecuado para la lectura. La mejor combinación es la de texto negro sobre un fondo blanco o amarillo pálido.

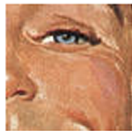
Bookatainment: An Interview with Jim Heimann

Written by **Steven Heller**

Published on February 13, 2007.

Filed in **Voice: Journal of Design** in **Off the cuff**.

↓ Comments (0)



Who wouldn't want a job where all one does, day after day, is track down rare and exotic design ephemera? OK, maybe it's not for everybody—some people would rather ski, surf or golf—but it's a

fine living for Jim Heimann. As the executive editor for Taschen America, the Los Angeles publishing arm of Benedikt Taschen's eclectic international imprint, Heimann is a cultural archaeologist who digs for and roots out the detritus that anonymous designers have left

Fuente: <http://voice.aiga.org/content.cfm?ContentAlias=%5Fgetfullarticle&aid=2535158>

El **tamaño de la letra** debe ser el adecuado para monitores estándar. La elección de un tamaño apropiado es fundamental en el caso de caracteres numéricos; el texto suele tener un grado de redundancia que permite al usuario leer sin detenerse a examinar cada uno de los caracteres, lo cual no ocurre con las secuencias numéricas.

El tamaño de letra de pantalla no debería ser inferior a los 9 puntos, y es aconsejable aumentar el interlineado 3 o 4 puntos por encima del tamaño de letra. En pruebas de lectura en pantalla se ha determinado que la letra que resulta más cómoda para los usuarios es la Arial a 12 puntos, seguida por la Times New Roman a 12 puntos.

- ➔ Texto a 8 puntos: presenta dificultades de lectura en pantalla.
- ➔ Texto a 9 puntos: supone el límite inferior de legibilidad en pantalla
- ➔ Texto a 11 puntos: es perfectamente legible. El interlineado es de 15 puntos, para favorecer la legibilidad.
- ➔ Arial a 12 puntos sin suavizar
- ➔ Times New Roman a 12 puntos sin suavizar
- ➔ Arial a 12 puntos suavizada

Las **líneas de texto** no deben ser excesivamente cortas (provocan problemas de composición), pero es aconsejable no superar los 40/50 caracteres.

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque id turpis. Pellentesque scelerisque nibh ut</p>	<p>Donec bibendum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla sed</p>	<p>Imperdiet et nonummy vel, libero. Quisque pharetra lacus vel massa. Etiam aliquam. Vestibulum ante ipsum.</p>
---	--	--

Línea demasiado corta

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque id turpis. Pellentesque scelerisque nibh ut augue. Etiam metus risus, elementum quis, cursus a, eleifend vitae, lacus. Mauris laculis tincidunt nulla. Donec nec ante. Nam in est ac nulla fermentum

Línea correcta

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque id turpis. Pellentesque scelerisque nibh ut augue. Etiam metus risus, elementum quis, cursus a, eleifend vitae, lacus. Mauris laculis tincidunt nulla. Donec nec ante. Nam in est ac nulla fermentum tincidunt. Cras convallis tortor eget tortor. Duis pharetra felis sit amet augue. Suspendisse potenti. Suspendisse at arcu. Donec bibendum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae;

Línea demasiado larga

Los **márgenes** alrededor del texto deben ser lo suficientemente amplios como para permitir una diferenciación visual clara entre el bloque de texto y los demás elementos de la interfaz.

Para aplicaciones de gran difusión, es fundamental tener en cuenta las condiciones ópticas de las personas de edades superiores a 45 años, que pueden precisar caracteres de mayor tamaño.

2.1. Pautas para la legibilidad

Además de las heurísticas generales, el diseño de los contenidos de un sitio web debe observar unos principios de usabilidad particulares (Nielsen, 2000), como se detalla a continuación:

Ved también

Las heurísticas de la usabilidad se estudian en el apartado 1 de este mismo módulo.

- **Brevedad:** La lectura en pantalla es más lenta e incómoda que la lectura en papel. Las páginas deben ser breves, y los contenidos sucintos y concretos, aunque no exentos de estilo.
- **Lectura en diagonal:** Los usuarios tienden a no leer por entero el texto en pantalla, sino que rastrean visualmente la página. Se recomienda:
 - Estructurar los contenidos en dos (o tres) niveles de titular (encabezado y subencabezados), con encabezados significativos.
 - Utilizar listas con viñetas para enumerar elementos, en lugar de incluirlos en bloques de texto uniformes.
 - Incluir negritas para destacar las palabras clave; también puede utilizarse texto coloreado, aunque en este caso el color elegido debe ser distinto al de los vínculos.
- **Lenguaje estructurado:** Las páginas deben organizarse en pirámide: lo más importante debe encontrarse al principio, de manera que el usuario no se vea forzado a leer toda la página para encontrar la conclusión. Puesto que muchos lectores sólo leen la primera frase de cada párrafo, es importante aplicar la regla de "una idea, un párrafo". Las frases deben ser sencillas.
- **Fragmentación:** Aunque el texto deba ser conciso, el contenido puede mantener su profundidad, dividiendo la información en nodos interconectados. Pueden incluirse páginas secundarias con contenido extenso y detallado.

Es recomendable no fragmentar excesivamente el texto, ya que supone que el usuario debe navegar entre múltiples páginas.
- **Títulos de página:** Los títulos de página suelen utilizarse como referencia principal, y son los que quedan almacenados si la página se incluye en la lista de favoritos. Deben ser explicativos y breves. Cada página debería presentar un título distinto, para ser válido como referencia.
- **Tratamiento gráfico:** Los factores que favorecen la legibilidad son:
 - Contraste entre el texto y el fondo. La legibilidad máxima se obtiene de texto negro sobre fondo blanco.
 - La tipografía debe presentar un tamaño relativamente grande (a partir de 11 puntos), para que sea legible incluso para personas con dificultades de visión.
 - A tamaño pequeño, se recomienda utilizar una tipografía de palo seco.
 - Las animaciones dificultan mucho la lectura.

- Para bloques de texto extensos, es recomendable utilizar la alineación a la izquierda.
- Las líneas de texto no deben superar los 40/50 caracteres.

3. Evaluación de la usabilidad

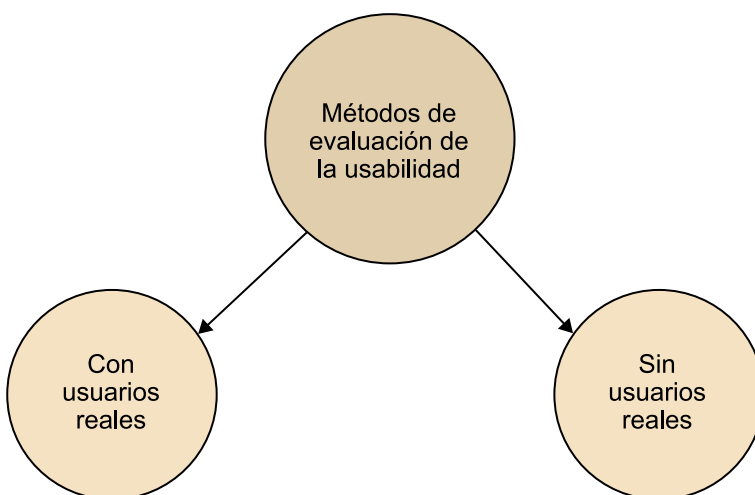
La evaluación de la usabilidad es la fase más importante del proceso de diseño centrado en el usuario. Puede llevarse a cabo en varias etapas, tanto durante el proceso de diseño y desarrollo como después del mismo.

Ved también

Consultad el apartado 4 del módulo "Diseño centrado en el usuario".

Hay distintos **métodos de evaluación de la usabilidad**, y pueden clasificarse en dos grandes grupos:

- 1) Los que recogen datos de los **usuarios reales**.
- 2) Los que pueden llevarse a cabo **sin los usuarios reales**.



La elección de un método u otro depende básicamente de tres factores: el presupuesto reservado a la evaluación, la adecuación al tipo de proyecto y las limitaciones de calendario.

Los principales métodos son los que se describen a continuación, aunque los más utilizados son el paseo cognitivo, el test con usuarios, y la evaluación heurística.

3.1. Paseo cognitivo

El paseo cognitivo (*cognitive walkthrough*) puede ser realizado **en cualquier fase del diseño** utilizando un prototipo, un borrador o el producto final.

Ved también

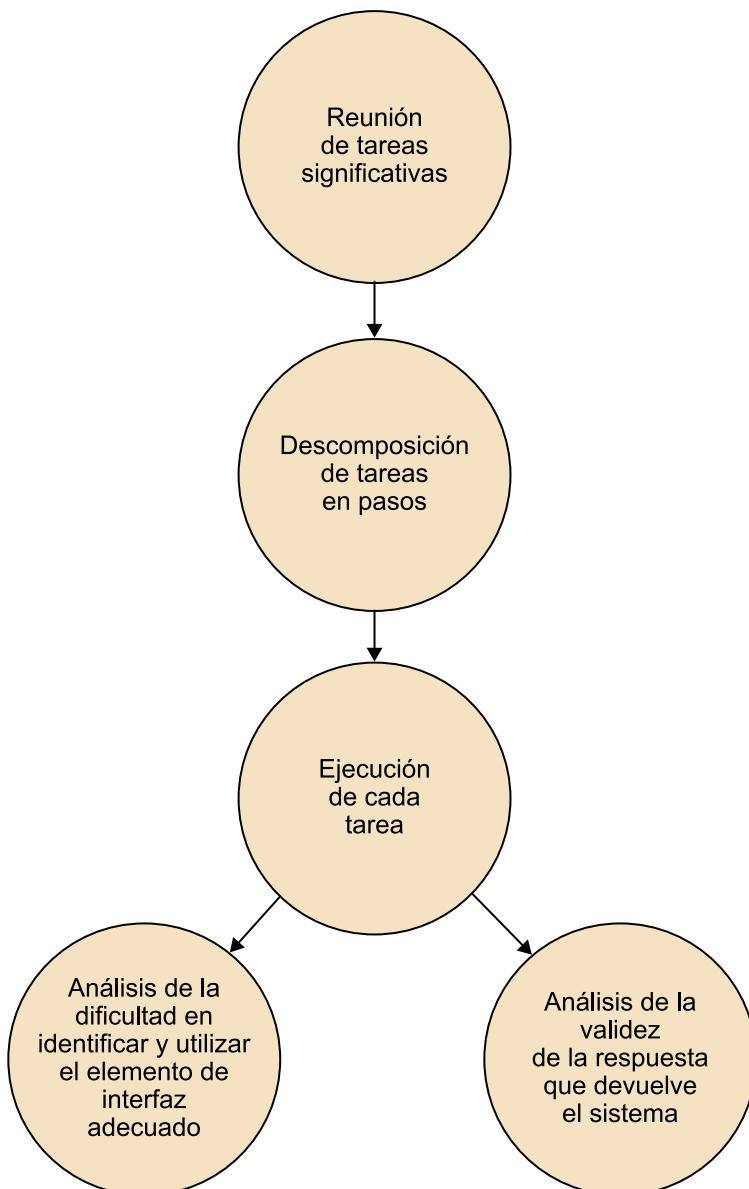
Consultad los apartados 4, 5 y 6 de este mismo módulo.

Teniendo en cuenta los objetivos de los usuarios, los evaluadores reúnen un grupo de **tareas significativas** y las descomponen en los **pasos** necesarios para realizarlas. A continuación se ejecuta cada tarea, analizando si puede resultar difícil para el usuario identificar y utilizar el elemento de la interfaz más adecuado para su objetivo, y si la respuesta que devuelve el sistema es suficientemente clara.

El paseo cognitivo tiene en cuenta los factores que intervienen en el **proceso mental** de toma de decisiones por parte del usuario, como la memoria de trabajo y la habilidad para razonar.

Este método es muy adecuado para comprobar la usabilidad de un sistema orientado a **usuarios con poca o ninguna experiencia**, que operan de modo exploratorio para aprender a utilizar la aplicación.

Etapas de un paseo cognitivo



3.2. Análisis de tareas

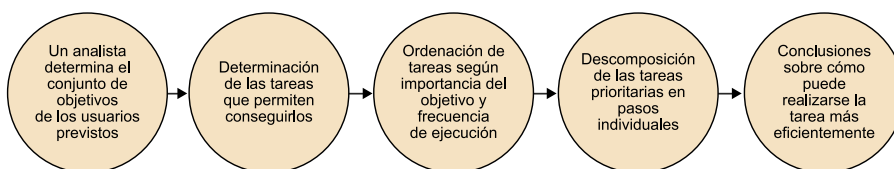
Este método evalúa cómo consiguen las personas sus **objetivos** mediante el software.

Mediante la observación y entrevistas con los usuarios, un analista determina el conjunto de objetivos de los usuarios previstos. A continuación se definen las **tareas** que permiten conseguirlos, y se ordenan de acuerdo con la importancia del objetivo y la frecuencia de ejecución de la tarea.

Las tareas prioritarias se descomponen en pasos individuales. El nivel de descomposición puede variar, dependiendo del sistema evaluado. A continuación, el análisis sugiere cómo puede realizarse la tarea más eficientemente, o propone nuevas tareas que puedan alcanzar más efectivamente los objetivos.

El análisis se realiza siempre desde la **perspectiva del usuario final**.

Etapas del análisis de tareas



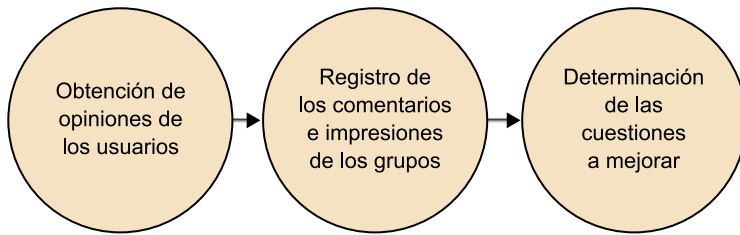
3.3. Grupos focales

Los grupos focales (*focus groups*) no es un método de evaluación de la usabilidad, sino que permite recoger requerimientos para el diseño de una aplicación. No obstante, puede utilizarse ocasionalmente cuando existe un prototipo para obtener **opiniones de los usuarios** y comprobar las reacciones iniciales a un diseño; también resulta muy eficiente para detectar en qué medida el diseño difiere de las expectativas de los usuarios.

Es un método de **bajo coste**; aunque la conducción de un grupo focal puede ser complicada, permite sacar a la luz cuestiones excepcionales, que en un análisis de tareas no serían descubiertas. Para contrastar resultados, se recomienda evaluar al menos dos grupos por proyecto.

El conductor del *focus group* redacta los comentarios e impresiones de los grupos, y sugiere las cuestiones que hay que mejorar.

Método de focus group

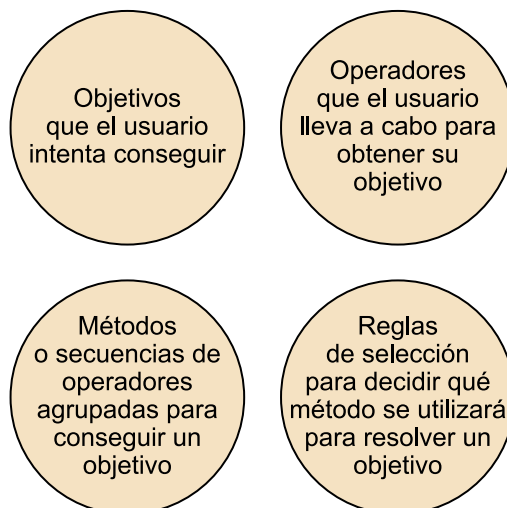


3.4. GOMS

Se trata de un conjunto de técnicas definidas por Card, Moran y Newell en 1983 para modelar y describir el proceso humano de ejecución de tareas. El acrónimo GOMS se refiere a:

- **Objetivos** (*goals*) que el usuario intenta conseguir, especificados jerárquicamente;
- **Operadores**, o microoperaciones que el usuario lleva a cabo para obtener su objetivo;
- **Métodos**, o secuencias de operadores agrupadas para conseguir un objetivo determinado, y
- **Reglas de selección**, que se utilizan para decidir qué método se utilizará para resolver un objetivo cuando existen varias posibilidades.

Método GOMS



3.5. Inspección de usabilidad

Las inspecciones son siempre llevadas a cabo por **expertos en usabilidad**, a partir de una serie de premisas o guías que derivan de estudios en interacción hombre-ordenador, ergonomía, diseño gráfico, diseño de información y psicología cognitiva.

Los expertos se centran especialmente en las áreas que puedan resultar problemáticas para los usuarios.

El método de inspección más utilizado en la evaluación de sistemas interactivos, es el de **evaluación heurística**.

3.6. Test con usuarios

Éste es el método más utilizado cuando se quieren conocer los problemas de usabilidad con los que puede encontrarse el usuario final. Se basa en la observación de los usuarios mientras ejecutan unas tareas representativas. La repetición del test con varios usuarios permite descubrir qué aspectos del diseño necesitan mejorarse.

Ved también

La evaluación heurística se estudia en el apartado 4 de este mismo módulo.

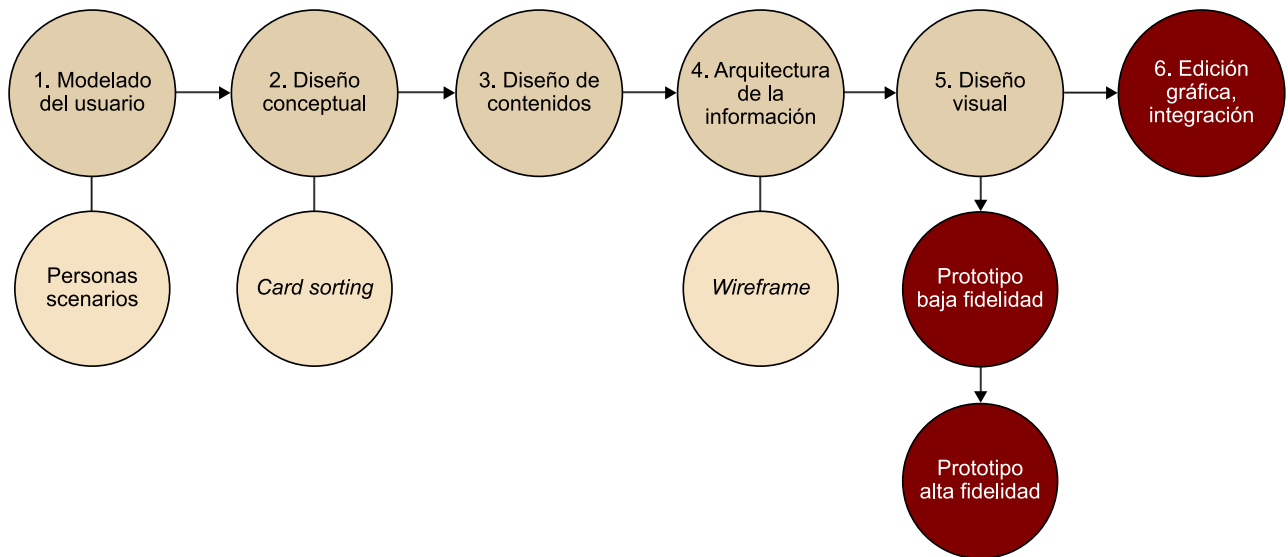
4. Evaluación heurística

La evaluación heurística es un **método de inspección**, es decir, se realiza por un grupo de expertos en usabilidad, que examinan la interfaz y determinan su grado de cumplimiento de los **principios de usabilidad** (o heurísticas).

Ved también

Los principios de la usabilidad se estudian en el apartado 1 de este mismo módulo.

Evaluación heurística en las fases del proceso de diseño



4.1. Desarrollo

Durante la sesión, el evaluador recorre la interfaz varias veces, examina los elementos de interacción, y los compara con los principios de usabilidad (heurísticas). El evaluador puede añadir principios de usabilidad adicionales, u otras cuestiones que sean relevantes para la interfaz evaluada.

Cada evaluador decide por sí mismo cómo quiere evaluar la interfaz, aunque se recomienda que recorra toda la aplicación al menos dos veces (la primera, para obtener una visión de conjunto, y la segunda, para entrar en detalle).

Puesto que los evaluadores no están utilizando la aplicación para realizar una tarea real, la evaluación heurística puede llevarse a cabo en las primeras fases de desarrollo, incluso en el diseño sobre papel.

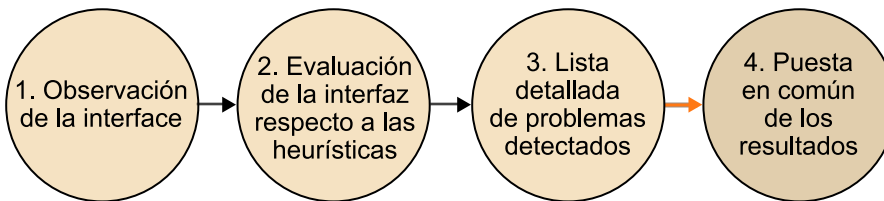
El resultado de la evaluación heurística debe ser una lista detallada de cada uno de los problemas de usabilidad hallados en la interfaz, con referencias a los principios que no se han respetado en cada caso, y las opiniones del evaluador.

Después de la evaluación, puede llevarse a cabo una puesta en común de los resultados, en la que participen los evaluadores, los observadores y miembros del equipo de diseño. La reunión puede tener carácter de *brainstorming*, y resulta muy útil para proponer soluciones a los problemas detectados, así como para comentar los aspectos positivos del diseño.

Ved también

El brainstorming se estudia en el apartado 2 del módulo "Diseño".

Fases de la evaluación heurística



4.2. Recomendaciones

Os recomendamos que tengáis en cuenta lo siguiente en la realización de una evaluación heurística:

- 1) La evaluación heurística debería hacerse por más de un experto, ya que uno sólo difícilmente podría encontrar todos los problemas de usabilidad existentes en la interfaz. Según Jakob Nielsen, el número ideal de evaluadores es de tres a cinco; superar este grupo no suele añadir información.
- 2) Cada evaluador examina la interfaz individualmente. Sólo cuando todos los evaluadores hayan comentado su inspección, podrán comunicarse entre sí para comentar los resultados. Es importante mantener el aislamiento inicial para que los resultados de cada evaluador sean independientes y no sesgados.
- 3) Para registrar los resultados de la evaluación, puede pedirse a cada evaluador que escriba un informe, o bien pueden grabarse los comentarios del evaluador a medida que examina individualmente la interfaz.
- 4) Puede haber un observador que ayude a los inspectores a utilizar la interfaz en caso de problemas; este observador puede encargarse de anotar los comentarios de los evaluadores, de manera que éstos no tengan que redactar informes. En todo caso, el observador se limitará a registrar los comentarios, sin realizar interpretaciones personales.
- 5) El observador puede responder a las preguntas de los evaluadores acerca del contexto de utilización de la interfaz; también puede ayudarlos en caso de problemas graves, aunque en este caso no debe tomar nunca la iniciativa, sino esperar a que el evaluador solicite la ayuda.

4.3. Duración

Una sesión de evaluación heurística suele durar una o dos horas por evaluador. Si la interfaz es muy compleja, puede dividirse en varias sesiones, cada una de ellas dedicada a una parte de la interfaz.

4.4. Heurísticas

Los diez principios generales de usabilidad son los siguientes:

- 1) Visibilidad del estado del sistema. El sistema debe mantener siempre informado al usuario de lo que está ocurriendo, y proporcionarle respuesta en un tiempo razonable.
- 2) Consistencia entre el sistema y el mundo real. El sistema debe utilizar el lenguaje del usuario, con expresiones que le resulten familiares. La información debe aparecer en un orden lógico.
- 3) Control del usuario. El usuario debe tener la capacidad de abandonar en cualquier momento una situación indeseada o accidental. Asimismo, podrá deshacer o repetir una acción.
- 4) Consistencia y estándares. El lenguaje utilizado debe ser coherente con las convenciones del sistema operativo.
- 5) Prevención de errores. Es importante prevenir la existencia de errores; si, a pesar de todo, deben aparecer mensajes de error, éstos han de contener opciones de confirmación antes de ejecutar las acciones de corrección.
- 6) Es mejor reconocer que recordar. Para que el usuario no se vea obligado a memorizar continuamente detalles de la navegación, los objetos, acciones y opciones deben estar a la vista. El usuario no tiene que recordar información de una parte de una ventana de diálogo a la siguiente. Las instrucciones de uso o la ayuda del sistema deben estar a la vista, o ser fácilmente accesibles.
- 7) Flexibilidad y eficiencia de uso. El sistema debe estar preparado para satisfacer tanto a los usuarios novatos, como a los experimentados. Para éstos, resulta muy recomendable incorporar atajos de teclado, que permiten acelerar el proceso de interacción. Los usuarios han de poder configurar sus propios atajos de teclado para acciones frecuentes.
- 8) Diseño práctico y sencillo. Las pantallas o páginas no deben contener información innecesaria o irrelevante, ya que distrae al usuario y entorpece la navegación. Si aun así es necesario incluir información auxiliar, ésta puede colocarse en páginas distintas, accesibles a través de enlaces.

Ved también

Para una descripción completa de estas heurísticas y otros principios generales de usabilidad, véase el apartado 1 en este mismo módulo.

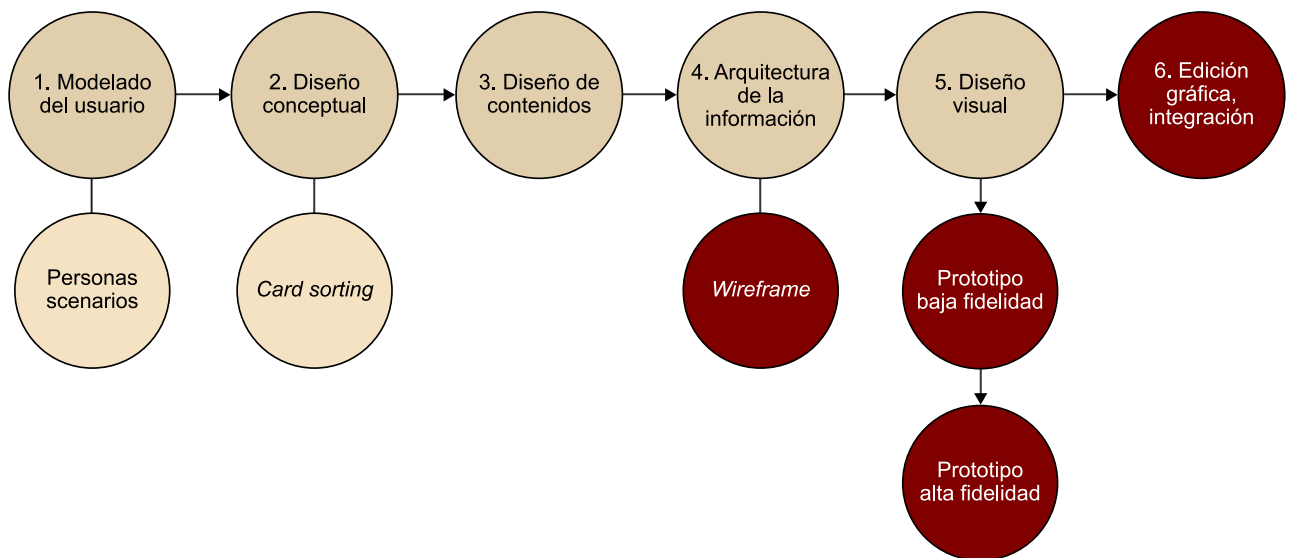
9) El usuario debe disponer de ayuda para reconocer, diagnosticar y deshacer errores. Los mensajes de error deben presentarse con un lenguaje sencillo, indicando el problema de manera precisa, y sugerir las posibles soluciones.

10) Ayuda y documentación. Aunque es mucho mejor que el usuario pueda navegar sin ayuda, la complejidad de un sistema puede recomendar incluir documentación de ayuda. Esta documentación debe ser fácil de encontrar, centrarse en la tarea del usuario, enumerar claramente los pasos que deben llevarse a cabo, y no ser extensa.

5. Paseo cognitivo

El paseo cognitivo (*cognitive walkthrough*) es un método de bajo coste que permite validar el diseño desde sus **primeras fases de desarrollo**, y evaluar los aspectos generales de la navegación. Puede llevarse a cabo sobre los primeros esbozos en papel.

Paseo cognitivo en las fases del proceso de diseño



5.1. Preparación

Basándose en los *scenarios* definidos en el proceso de diseño, se describe un conjunto de **tareas representativas**. Es importante que en ellas intervenga el mayor número de elementos de la interfaz, especialmente aquellos que se sospecha que pueden resultar problemáticos.

Las tareas deben describirse en un lenguaje sencillo y directo, sin hacer referencia a ningún aspecto de la interacción entre pantallas.

Los participantes deben tener características similares a las de los **usuarios finales**, aunque pueden incluirse algunos con diferente perfil para obtener distintas perspectivas.

Es importante realizar una **sesión piloto** para detectar problemas en la descripción de las tareas o de las pantallas, o la falta de pasos o de datos.

Ved también

Los *scenarios* se estudian en el apartado 6 del módulo "Diseño centrado en el usuario".

5.2. Desarrollo

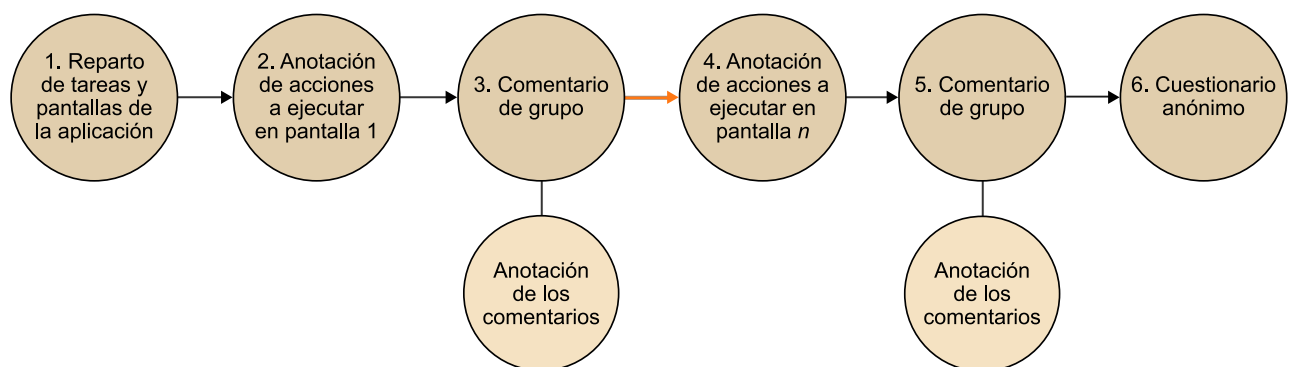
Se reparte a cada participante una carpeta o bloc que contiene las tareas que se deben completar y las pantallas de la aplicación (una por hoja). Se le pide que anote en cada hoja qué acciones cree que debería realizar en la pantalla para ejecutar la tarea descrita.

Después de que los participantes hayan revisado individualmente la primera pantalla, se comenta en grupo y se detallan las cuestiones que pueden presentar problemas. El moderador anota o registra los comentarios.

A continuación el moderador explica qué acción realizaría un usuario hipotético para pasar a la siguiente pantalla, que se comenta de la misma manera. El paseo continúa hasta que se han revisado todas las pantallas.

Al final de la sesión, es recomendable distribuir un cuestionario anónimo para confirmar que los participantes encajan en el perfil demográfico requerido, y para obtener comentarios subjetivos.

Fases del paseo cognitivo



5.3. Duración

Una sesión de paseo cognitivo puede durar aproximadamente dos horas.

5.4. Recomendaciones

Os recomendamos que tengáis en cuenta lo siguiente en la realización de un paseo cognitivo:

- 1) Los participantes deben tener claro que se está evaluando la interfaz, no sus capacidades.
- 2) El moderador debe reunir todos los comentarios realizados y escribir sus observaciones personales tan pronto como sea posible después del paseo.

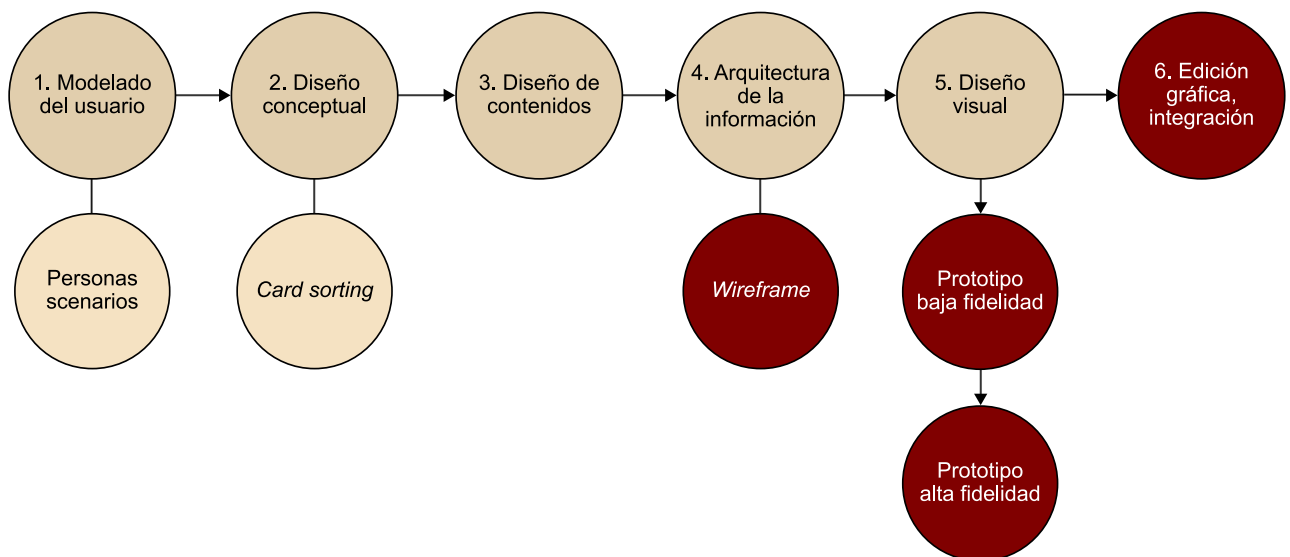
- 3) El comentario en grupo de cada pantalla debe comenzar sólo cuando todos los participantes han terminado de escribir sus anotaciones personales.
- 4) Es recomendable que otros miembros del equipo de desarrollo estén presentes en la sesión.
- 5) El moderador debe evitar que los participantes cambien de opinión basándose en los comentarios de otros participantes.
- 6) Las pantallas deben tener un aspecto representativo, pero sencillo, para que los participantes no se entretengan en aspectos gráficos.

6. Realización de un test con usuarios

La realización de un test con usuarios es un método que se basa en la **observación y análisis de las acciones de un grupo de usuarios**, anotando los problemas que se presenten para poder resolverlos posteriormente. Es útil para garantizar que el sistema puede llevar a cabo las tareas previstas de manera eficiente y satisfactoria.

Este método puede llevarse a cabo en las **primeras etapas** de diseño de la aplicación, incluso sobre los prototipos. En todo caso, siempre debería realizarse **después de una evaluación heurística**, que puede detectar los primeros problemas de usabilidad.

Test de usuario en las fases del proceso de diseño



6.1. Preparación

La evaluación debe llevarse a cabo en un local o laboratorio en el que exista un ordenador para realizar la prueba, y en el que no haya interferencias externas o ruidos imprevistos.

Debe contarse con al menos **cinco participantes**; pueden ser amigos, compañeros de trabajo, etc., aunque es preferible que tengan un perfil similar al del **usuario focal** de la aplicación. Cada participante realiza la prueba por separado, acompañado por un **observador**.

Ved también

El usuario focal se estudia en el apartado 6 del módulo "Diseño centrado en el usuario".

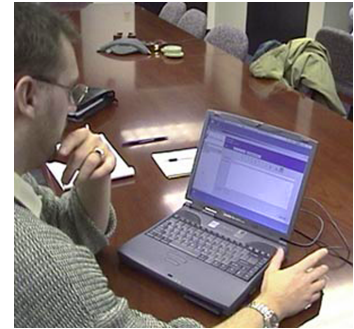
El observador debe disponer de un bloc para anotar las observaciones. Asimismo, pueden asistir como observadores otros miembros del equipo de desarrollo, aunque es recomendable que permanezcan en una habitación distinta.

El evaluador debe estar atento no sólo a lo que el usuario dice, sino también a sus expresiones y gestos. Para ello resulta muy útil contar con una cámara que grabe la sesión, previo consentimiento del usuario.

El observador dirige la prueba mediante un **guión** que especifica qué **tareas** debe llevar a cabo el participante; deben seleccionarse tareas derivadas de los *scenarios* definidos en la etapa de diseño, y priorizar las que sean susceptibles de ocasionar problemas de usabilidad. No es imprescindible seguir estrictamente el guión establecido, ya que puede utilizarse solamente como medio de orientación.

Antes de empezar, es importante establecer un **ambiente cómodo** para el participante, que seguramente estará inquieto. Para ello:

- Se explicarán claramente los objetivos de la prueba. El participante debe tener muy claro que no se están evaluando sus capacidades, sino la aplicación; si existe algún problema durante la prueba, no será culpa suya sino del diseño.
- En ningún caso deben explicarse las características de la aplicación, ya que uno de los objetivos del test es comprobar si el diseño es eficaz y fácil de comprender.
- Se indica al usuario que su nombre no figurará en el registro del test, es decir, que es anónimo.
- El local en el que se realiza la prueba ha de ser tranquilo, sin ruidos que puedan distraer al participante.
- No deben comentarse aspectos personales del participante; todos los comentarios deberían referirse a la aplicación.
- Se debe motivar al usuario para que exprese en voz alta cualquier pensamiento, opinión, problema, etc.
- El observador no ayuda al usuario a solucionar problemas de uso de la interfaz; su función es simplemente la de observador silencioso.



6.2. Desarrollo

El evaluador ejecuta la aplicación. La primera información que habrá que obtener es el grado de entendimiento, de manera que en una primera fase se pide al usuario que, sin hacer todavía nada, observe lo que ve y explique cuál cree que es el contenido y el objetivo de la aplicación, y que exprese sus opiniones personales, aunque las opiniones acerca de cuestiones estéticas son poco relevantes.

A continuación, se analiza la facilidad de uso de la aplicación. Para ello, se le comenta al usuario cuántas tareas va a tener que realizar, y se le describe la primera (por ejemplo, "debería obtener información sobre el curso de diseño html, ¿cómo lo haría?"); se le pide que piense siempre en voz alta.

Cuando el usuario finaliza una tarea se le describe la siguiente, hasta realizarlas todas. Si el usuario no consigue finalizar alguna de ellas, se le debe dejar claro que el problema está en el diseño, no en sus acciones, y se pasa a la siguiente.

Es aconsejable anotar el tiempo que el usuario invierte en la realización de cada tarea.

Al final del test, puede solicitarse al usuario que rellene un formulario con datos estadísticos (el cuestionario es anónimo, no incluye el nombre del participante) del tipo siguiente:

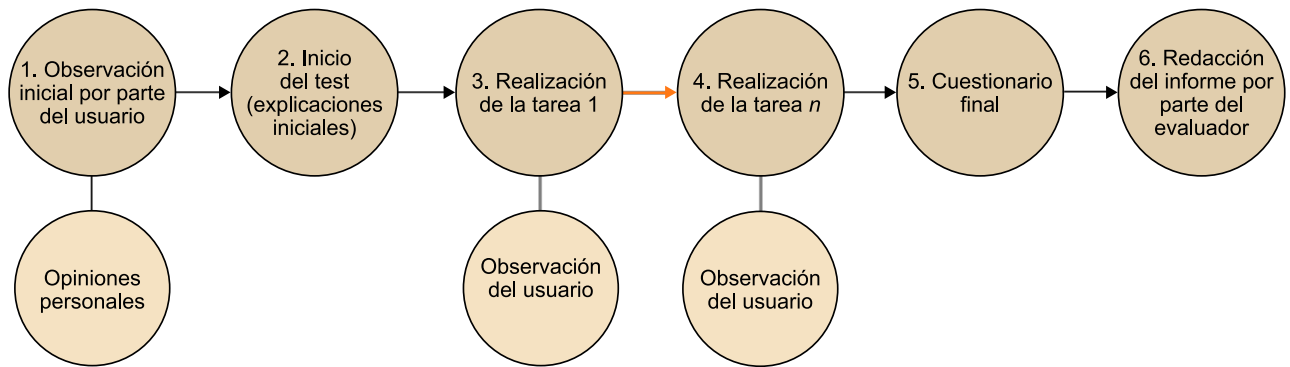
- Edad
- Profesión
- Nivel académico
- Frecuencia de uso de Internet

Además, el cuestionario puede contener preguntas concretas acerca del diseño, valoradas en una escala de 1 (completamente de acuerdo) a 5 (completamente en desacuerdo), como:

- El producto es fácil de utilizar
- Siempre sé que estoy dentro de la aplicación
- Es fácil perderse
- Es difícil aprender a utilizarla
- No tengo suficiente formación para utilizarla
- La ayuda es útil

Finalmente, el evaluador debe redactar un informe con todo lo que se haya anotado durante las pruebas. Este informe debería indicar qué problemas de usabilidad se han encontrado.

Fases del test de usuario



Bibliografía

Bibliografía del apartado 1

Hassan; Martín; Iazza (2004). *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*. Disponible en línea en: <<http://www.hipertext.net/web/pag206.htm>>

Hobart, J. (1995). *Principles of good GUI Design*. Disponible en línea en: <http://axp16.iee.org.mx/Monitor/v01n03/ar_ihc2.htm>

Information Services and Technology. *Usability Guidelines*. Disponible en línea en: <<http://web.mit.edu/ist/usability/usability-guidelines.html>>

Krug, S. (2006). *No me hagas pensar*, Madrid: Pearson Educación.

Lynch, P. J. (1994). *Visual design for the user interface*, Yale Center for Advanced Instructional Media. Disponible en línea en: <<http://cal.bemidjstate.edu/webtraining/yalemanual/papers/gui1.html>>

Nielsen, J., Loranger, N. (2006). *Usabilidad. Prioridad en el diseño web*, Madrid: Anaya Multimedia.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Reading: Addison-Wesley Publishers.

Skaalid, B. (1999). *Human-Computer Interface Design*. Disponible en línea en: <http://www.usask.ca/education/coursework/skaalid/theory/interface.htm>

Tognazzini, B. (1998). *First Principles*. Disponible en línea en: <<http://www.asktog.com/basics/firstPrinciples.html>>

UsabilityNet (2003). Disponible en línea en: <<http://www.usabilitynet.org/home.htm>>

Bibliografía del apartado 2

Nielsen, Jakob (2000). *Usabilidad. Diseño de sitios web*, Madrid: Ed. Pearson Educación.

Bibliografía del apartado 3

Hom, J. (2003). *The Usability Methods Toolbox*. Disponible en línea en: <<http://jthom.best.vwh.net/usability/>>

Krug, S. (2006). *No me hagas pensar*, Madrid: Pearson Educación.

Nielsen, J.; Loranger, N. (2006). *Usabilidad. Prioridad en el diseño web*, Madrid: Anaya Multimedia.

Usability First (2004). *Usability Methods*. Disponible en línea en: <<http://www.usabilityfirst.com/methods/index.txt>>

Bibliografía del apartado 6

Montero, Y. H.; Martín Fernández, F. J. (2003). *Método de test con usuarios*. Disponible en línea en: <http://www.nosolousabilidad.com/articulos/test_usuarios.htm>

