

TEMA

46



CEDE

*Diseño
de interfaces
gráficas
de usuario.*

elaborado por
EL EQUIPO DE PROFESORES
DEL CENTRO DOCUMENTACIÓN

1. DISEÑO DE INTERFACES DE USUARIO

1.1. HISTORIA

La tecnología de interfaz de usuario, al igual que el hardware, ha pasado por una serie de generaciones. Estas generaciones contienen o parecen contener a las anteriores, y se pueden clasificar cronológicamente como sigue:

- **Hasta 1945:** no existía ningún paradigma de interfaz de usuario, y se hacía acceso directo de forma manual al hardware.
- **1945-1955:** programación en modo *batch* o por lotes. La primera generación de interfaces no era interactiva, ya que la interacción entre el sistema y el usuario se restringía a un único punto en el tiempo. Todos los mandatos del usuario tenían que ser especificados antes de que el usuario conociera el resultado de cualquiera de ellos. Se recomienda que tales modos *batch* proporcionen alguna opción al usuario para controlar continuamente el progreso del trabajo *batch*, de forma que pueda interrumpir o modificar el trabajo.
- **1955-1965:** lenguajes de mandatos. También denominadas interfaces en línea. Los sistemas de tiempo compartido se inventaron alrededor de 1960 como un medio para permitir a varios usuarios tener acceso simultáneo interactivo a un único servidor. Uno de los problemas principales de estos sistemas es la pequeña cantidad de recursos de ordenador disponibles para soportar la interfaz de cualquier usuario, por lo que, a menudo, se utilizan interfaces en línea. Éstas eran básicamente interfaces de una dimensión, en las que el usuario sólo podía interactuar con el ordenador en una línea que servía como línea de mandato. Estas interfaces se implementaron originalmente en las máquinas teletipos, aunque las últimas versiones utilizan pantallas tipo tern-únal. Debido a que las interfaces en línea no permitían a los usuarios navegar por la pantalla, la interacción se limitaba a diálogos pregunta-respuesta y a la introducción de mandatos con parámetros.
- **1965-1980:** pantallas completas con menús estrictamente jerárquicos. El espacio de diseño de la interfaz es de dos dimensiones. Un uso clásico de la pantalla completa es el de los diálogos para rellenar informes, en los que el usuario tiene un número de campos etiquetados que puede editar en la secuencia que desee. Estos diálogos todavía existen en las interfaces modernas en forma de cajas de diálogo, las cuales son más dinámicas que los informes tradicionales, ya que contienen menús desplegados y ayuda para que el usuario rellene el informe. Además de los menús, muchas de las in-

terfaces de pantalla completa utilizan también las teclas de función como una forma de interacción. Estas teclas aceleran la interacción con la aplicación en cuestión.

- **1980-1997:** ventanas, iconos, menús y ratón. También denominadas interfaces gráficas de usuario. La mayoría de las interfaces actuales de usuario pertenecen a esta categoría, a veces denominada sistemas WIMP (Windows, Icons, Menus and Pointing device). Las interfaces windows añaden casi una tercera dimensión a las dos dimensiones inherentes a cada ventana debido a la posibilidad de superponer ventanas (está claro que superponer ventanas no es verdaderamente una tercera dimensión, ya que no es posible ver el contenido de la ventana que está debajo, por tanto, podríamos decir que tienen "dos dimensiones y media"). El estilo de interacción utilizado en muchas interfaces gráficas de usuario es la manipulación directa, la cual se basa en la representación visual de los objetos del diálogo que tengan interés para el usuario. Esto permite al usuario controlar el diálogo con sólo mover los objetos por la pantalla y manipularlos con el ratón. Sin embargo, las interfaces de manipulación directa pueden resultar más difíciles de utilizar que las tradicionales, debido a que son más dependientes de un control fino sobre el ratón.

- **1997-Futuro:** interfaces no basadas en mandatos. La próxima generación de interfaces ya se está desarrollando. Es probable que la tendencia de las generaciones previas continúe, y que la dimensión de las interfaces de usuario aumente de 2.5 a 3 o más dimensiones. Las formas comunes de añadir una dimensión a las interfaces de usuario consisten en añadir tiempo (en forma de animación), sonido o voz, así como una verdadera tercera dimensión espacial en forma de sistemas de realidad virtual. Las dos predicciones más fáciles con respecto a la siguiente generación de interfaces de usuario son que incluirán una dimensionalidad más alta con más tipos de medios y que serán altamente portables y personales, a la vez que se utiliza tecnología de comunicaciones para conseguir una gran conectividad. Otro estilo de diálogo para las interfaces de usuario de la próxima generación puede ser el de las interfaces de usuario no basadas en mandatos. Todos los estilos de interfaz hasta ahora han tenido en común, al menos, el concepto de mandato, y se basaban en el principio de un diálogo explícito entre el usuario y el ordenador en el que el usuario ordenaba al ordenador que hiciese ciertas acciones específicas. Por el contrario, muchos esfuerzos actuales de investigación apuntan a sistemas que permitan al usuario centrarse en el dominio en lugar de tener que controlar al ordenador explícitamente. En estos sistemas futuros, el ordenador se encargará de tomar la responsabilidad de la interacción, basando sus acciones en sus observaciones del usuario, utilizando tecnologías como el seguimiento del ojo, reconocimiento de gestos y análisis semi-inteligente de las acciones del usuario.

1.2. CICLO DE VIDA PARA EL DISEÑO DE UN INTERFACE

Las distintas fases de que consta este ciclo son:

Recopilar información

1. Organizar el equipo de diseño.
2. Procurar el acuerdo entre dirección y clientes.
3. Dirigir las entrevistas con los usuarios.
4. Presentar cuestionarios escritos a los clientes.
5. Realizar el análisis detallado de las tareas, en especial, el de las más frecuentes.
6. Hablar con desarrolladores de sistemas parecidos e intercambiar información.
7. Estimar los costes de desarrollo, formación, uso y mantenimiento.
8. Preparar un calendario con hitos y revisiones observables.
9. Diseñar la estrategia de pruebas en laboratorio.

Definir los requisitos y la semántica (Pseudocódigo)

1. Definir los requisitos de nivel medio (lo que vamos a usar para hacer la aplicación) y los de nivel alto (lo que el usuario quiere).
2. Considerar las posibles alternativas en el secuenciamiento de las tareas.
3. Organizar las tareas en unidades.
4. Crear las acciones de las tareas.
5. Crear las acciones informáticas.
6. Especificar la seguridad y privacidad de la información.
7. Crear los documentos guía.
8. Obtener acuerdo entre dirección y clientes en cuanto a objetivos, requisitos y semántica.

Especificar la sintaxis y las funciones

1. Comparar formatos de visualización alternativos.
2. Crear la sintaxis de las acciones (pseudocódigo).
3. Diseñar la información de retroalimentación para cada acción (que se ha hecho, que se está haciendo y que se va a hacer).
4. Desarrollar los mensajes de error y su tratamiento, por ej. ofreciendo alternativas como reintentar, cancelar (deshacer hasta el error inmediato) o anular (deshacer todo hasta el principio).
5. Especificar los tiempos de respuesta del sistema ratón, teclado, etc...

6. Planificar la asistencia a los usuarios con las ayudas y los tutoriales.
7. Escribir los manuales de usuario y de referencia.
8. Revisar, evaluar y repasar las especificaciones del diseño.
9. Realizar pruebas piloto con papel y lápiz del prototipo.

Especificar dispositivos físicos

1. De impresión (impresoras, plotters...).
2. Colores, tamaño y resolución de la pantalla.
3. Teclado.
4. Gráficos, audio y otros dispositivos periféricos.
5. Requisitos para las líneas de comunicación.
6. Tener en cuenta el entorno de trabajo (luz, ruido, espacio en la mesa).

Desarrollo del software

1. Decidir y emplear el sistema de gestión de interfaces de usuario (Visual Basic, C++, ...).
2. Producir un diseño modular de arriba abajo (top/down).
3. Enfatizar la posibilidad de modificación y mantenimiento por ej. código con comentarios).
4. Asegurar su fiabilidad y seguridad.
5. Proporcionar una documentación adecuada.
6. Producir versiones del producto final.
7. Dirigir las pruebas del software tanto de las versiones como del producto final.

Integración del sistema y distribución a los usuarios

1. Dirigir las pruebas de aceptación y sincronización del sistema.
2. Realizar pruebas de campo de los manuales escritos (que el programa haga lo que el manual, la ayuda on-line y los tutoriales dicen).
3. Suministrar entrenamiento a los usuarios.
4. Supervisar la distribución.

Información a los usuarios

1. Proporcionar consultas telefónicas.
2. Desarrollar un buzón de sugerencias.
3. Dirigir las entrevistas con los usuarios.

4. Hacer disponibles las noticias y los boletines del interfaz.
5. Organizar reuniones de grupo.
6. Responder a los usuarios sobre las sugerencias que hagan.
7. Seguir la pista a los errores más frecuentes de los usuarios.

Plan de evolución

1. Diseñar para mantener y reparar de forma fácil.
2. Si formamos parte de un grupo de trabajo, es conveniente reservar tiempo para la realización de mejoras.
3. Medir el rendimiento del usuario.
4. Mejorar el tratamiento de los errores.
5. Planificar las revisiones de una manera regular.

1.3. PRINCIPIOS DE UN BUEN DISEÑO DE UNA INTERFACE

1. El estado del sistema y las acciones alternativas deben ser visibles en todo momento (lo que la máquina hace y lo que se puede hacer para pararlo).
2. El diseño del sistema de imágenes de interfaz debe tener detrás un buen modelo conceptual (es decir, un buen análisis).
3. El interfaz debe incluir medidas que muestren las relaciones entre etapas (cuando el usuario realiza acciones distintas, que vea como pasa de unas a otras).
4. El usuario siempre debe recibir retroalimentación.

1.4. PERSPECTIVAS DE DISEÑO DE UNA INTERFACE

Existen tres puntos de vista para el diseño del modelo de una interface. Dicho modelo tiene como objetivo explicar el comportamiento del sistema y ayudar a la toma de decisiones en el mismo.

- **Punto de Vista del Diseñador:** en este punto el diseñador mezcla los requisitos de presentación y operación del usuario y las posibilidades técnicas/tecnológicas a su alcance. Este debe adecuarse en todo momento al entorno de desarrollo de la aplicación y presentar soluciones homogéneas con el resto del desarrollo.

El modelo del diseñador describe los objetos que utiliza el usuario, su presentación al mismo y las técnicas de interacción para su manipulación.

Por lo tanto el modelo del diseñador consta de tres partes:

- **Presentación:** es lo primero que capta la atención del usuario , pero no es lo mas importante ya que pasa a segundo plano cuando la aplicación realiza alguna operación solicitada por el usuario.
- **Técnicas de interacción:** esta parte está muy relacionada con la plataforma, entorno, lenguajes, sistemas,... en los que se lleva a cabo la aplicación.
- **Relación entre los objetos:** en este punto el diseñador integra los dos puntos anteriores para conformar objetos funcionales de interacción hombre-máquina.
- **Punto de Vista del Programador:** En este punto de vista el programador ha recibido una serie de especificaciones formales del diseñador, que ha su vez las recibió del usuario, y es el programador el que lleva a la práctica cada uno de los objetos de interacción.
- **Punto de Vista del Usuario:** Aquí es el usuario el que tiene una visión de las interfaces que necesita para realizar las funciones encomendadas a la aplicación. Esta visión se puede obtener en una primera aproximación después de analizar el trabajo que realiza diariamente el usuario, y que deberá de poderse realizar en la aplicación.

El diseño de interfaces variará mucho en función de los usuarios a los que vayan orientados los diseños. A continuación se describen tres tipos de usuarios y las características de diseño que se deberían tener en cuenta:

- **Novicios:** No tienen conocimientos sintácticos ni semánticos sobre el sistema. Para este tipo se tendrá en cuenta:
 - Restringir vocabulario a un número pequeño de términos familiares.
 - Mantener un número pequeño de posibilidades a realizar.
 - Conducir a ejecutar tareas sencillas para dar confianza y reducir ansiedad.
 - Mucha retroalimentación, con muchos mensajes de error claros y sencillos.
 - Buen manual de usuario (claro y sencillo de manejar).
 - Tutoriales en línea.
- **Intermitentes:** Usan la aplicación de vez en cuando, por tanto, tendrán conocimiento semántico pero no sintáctico. Para este tipo de usuario se tendrá en cuenta:
 - Construir una estructura simple y consistente de comandos, de menús y de terminología.

- Usar el reconocimiento para realizar tareas (iconos con mnemónicos).
- Sistema muy bien protegido contra daños (pues este usuario explora mucho).
- Ayudas en línea que le recuerden la sintaxis.
- Manuales muy bien organizados (respecto del índice y del glosario).

– **Expertos:** Dominan la semántica y la sintaxis del sistema, requieren hacer rápidamente su trabajo. Para este tipo de usuario se tendrá en cuenta:

- Dar tiempos de respuesta cortos (optimizando algoritmos).
- Poca retroalimentación (pues les distrae) Lenguaje de macros muy potente.

2. CRITERIOS DE DISEÑO

2.1. ERGONOMÍA DEL DISEÑO DE UNA INTERFACE

El viaje al mundo del diseño de pantallas comienza con una discusión con el usuario, la parte más importante de cualquier sistema informatizado. Como hemos comentado, es de aceptación general que el sistema sería más fácil de utilizar si lo hubiera diseñado el usuario, por tanto, el diseño de la interfaz debe estimular al usuario haciéndolo cómplice del sistema.

El ser humano es un organismo complejo con una variedad de atributos que tienen una influencia importante en el diseño de pantallas. Entre estos atributos, destacamos la percepción, la memoria, el aprendizaje, la capacidad y diferencias individuales. Por tanto, debe cultivarse la satisfacción personal y, consecuentemente, mejorar la aceptación del sistema, permitiendo que las personas mejoren sus conocimientos mientras utilizan el sistema.

Para realizar un buen trabajo, la interfaz con el ordenador debe ser, entre otras cosas, lo más fácil, amigable y agradable posible, y se debe usar un diálogo que se acerque al lenguaje natural en vez de la jerga informática. Entre las consideraciones a tener cuenta a la hora de diseñar interfaces se encuentran las siguientes:

- Características deseadas: simplicidad, claridad y fácil de comprender. Será necesario tener claridad visual, de forma que los elementos estén agrupados de forma comprensible y con significado en vez de al azar y de forma confusa.
- Saber dónde situar la información en la pantalla. Será necesario indicar un punto de partida obvio en la esquina superior izquierda de la pantalla, reservar áreas específicas de pantalla para diferentes tipos de información (como, por ejemplo, mandatos, mensajes de error, títulos y campos de datos, de forma que esta consistencia se mantenga

en todas las pantallas) y proporcionar una composición que guste visualmente (es decir, que esté balanceada, sea simétrica, sea predecible, secuencial, simple, con agrupamientos, etc.).

- Saber qué información situar en la pantalla. Para ello, hay que poner sólo la información que es esencial para la toma de una decisión o para la ejecución de una acción (¡No inundar al usuario con información!) y poner todos los datos relacionados con una tarea en una única pantalla (así el usuario no tiene que recordar datos de una pantalla a otra).
- Saber cómo situar la información en la pantalla. Así, en cuanto a las fuentes de letras, se recomienda utilizar minúsculas para el texto con la letra inicial de la frase en mayúsculas; para las etiquetas, encabezamientos o subtítulos utilizar mayúsculas. En cuanto a las palabras, se recomienda no usar jerga, sino utilizar palabras cortas, familiares, etc. También es necesario saber como alinear y/o resaltar el texto y las palabras, dónde situar las ilustraciones, los campos de datos, etc.
- La interfaz de entrada debe recoger todos los datos necesarios, sin introducir errores, para el sistema. De esta forma, la interfaz contiene una protección contra errores de entrada. Así mismo, también debe recoger los datos minimizando el número de teclas pulsadas por el usuario. Las entradas deben estar bien estructuradas y ser fáciles de comprender y utilizar. Se deben usar nombres precisos y permitir abreviaturas cuando se necesiten introducciones rápidas de datos. Se deben evitar las entradas repetitivas. Igualmente, el diseño de la salida asegura que se extraen todos los datos suministrados por el sistema y que esas salidas están estructuradas de forma que sean fáciles de leer.
- El color añade una nueva dimensión a la facilidad de uso de la pantalla, ya que atrae la atención del usuario. Si se utiliza de forma adecuada, puede resaltar la organización lógica de una pantalla, facilitar la separación de componentes y acentuar las diferencias. Por el contrario, si se usa inadecuadamente, puede distraer y fatigar la visión debilitando la facilidad de uso del sistema. Por ejemplo, en las pantallas gráficas se recomienda no utilizar más de seis colores a la vez, evitar colores extremos (rojo y azul, amarillo y púrpura), evitar colores que no tengan contraste (blanco y amarillo, rojos, azules).

Para finalizar, diremos que el diseño de interfaces de usuario es un proceso ordenado que empieza en los requisitos y finaliza con la implementación.

2.2. CREACIÓN DE PROTORIPOS PRELIMINARES Y DIÁLOGOS

El propósito de la interfaz es muy simple: recoger de los usuarios la información del sistema y ponerla a disposición de otros usuarios. La información recogida se llama **entrada del sistema** y se almacena en la base de datos para ponerla a disposición de los demás usuarios. La información suministrada se llama **salida del sistema**. Es decir, el diseño de interfaces cubre tanto las entradas como las salidas.

Las entradas y salidas pueden ser **interactivas** o **no interactivas**. En una interfaz interactiva, el usuario se comunica directamente con el ordenador y la salida se obtiene muy poco tiempo después de la entrada. En el caso de entradas o salidas no interactivas, las transacciones se reúnen en formularios en el punto de recepción y después se introducen en el ordenador al mismo tiempo. Estas transacciones se procesan y un tiempo después se producen las salidas, las cuales se pasan al usuario. Así, el tiempo transcurrido desde la introducción de los datos hasta que se obtiene una respuesta puede ser considerable.

El diseño de interfaz interactivo provoca un diálogo hombre-máquina que permite un intercambio rápido de información entre los ordenadores y sus usuarios humanos, mientras que la interfaz no interactiva utiliza un soporte de papel para contener la información en el que las entradas normalmente se realizan en formularios especialmente diseñados o ficheros de entrada y las salidas se producen en un listado impreso o en una inserción en Base de Datos o fichero procesado.

Será necesario definir los formatos individuales de las pantallas utilizadas. En el caso de utilizar un paquete estándar, habrá que evaluar la posibilidad de adoptar el tipo de formato del producto. Entre los aspectos a considerar en los formatos de las interfaces se destacan: encabezamiento, cuerpo principal, pie, teclas de función, teclas de ayuda y líneas de visualización de los mensajes de ayuda.

También hay que describir, de forma detallada, los diálogos entre pantallas y su encajamiento. Para ello, es útil representarlas jerárquicamente, de forma que en los niveles superiores se representen los menús, y en los niveles inferiores las pantallas de diálogos, representativas de funciones o procesos concretos del sistema. En esta representación jerárquica nos interesa identificar los menús o diálogos concretos en función de los grupos de usuarios que los utilicen.

Será también necesario determinar los diálogos que se consideran críticos para un funcionamiento correcto del nuevo sistema. Los diálogos críticos se determinan en función de factores como: tipo de proyecto, grado de cambio con respecto al sistema actual, complejidad

de los trabajos del sistema. Para ello, es útil tener en cuenta los siguientes criterios: frecuencia de uso de estos diálogos, acceso a gran número de entidades de datos del sistema, gran número de elementos de datos asociados con el diálogo, cambio en el modo habitual de trabajo en el sistema actual, diálogos comunes a diferentes grupos de usuarios, diálogos que contienen opciones complejas de navegación, etc.

Por último, habrá que realizar un prototipo dinámico que permita la simulación (introducción y validación de datos por pantalla) para los diálogos considerados como críticos. Como recomendaciones para realizar este prototipo se tendrá en cuenta: la determinación del punto de entrada a cada pantalla y sus posibilidades de navegación a otras, la especificación de los datos asociados a cada pantalla (longitud, reglas de validación, mensajes de ayuda, etc.), la evaluación de la consistencia del diseño, asegurando que toda la información necesaria para el usuario está contemplada en las pantallas, y la confirmación con el usuario de la validez de los diseños de pantalla realizados.

3. DOCUMENTACIÓN

Dentro de este apartado se puede distinguir diferentes tipos de documentación:

- Tutoriales que identifican con una visión rápida lo que hace la aplicación.
- Manuales de Referencia que explican con claridad lo que hace la aplicación.
- Guía de usuario experto, con indicaciones puntuales sobre la aplicación.

Para todos ellos se debe tener en cuenta:

- Evitar atribuir al ordenador características humanas.
- Evitar ciertos verbos o tiempos verbales asociados a los sistemas informáticos.

NO	SÍ
Conoce	Procesa
Piensa	Informa
Comprende	Calcula
Memoriza	Almacena
Da	Recupera

- Evitar ciertos verbos o tiempos verbales asociados a los usuarios.

NO	SÍ
Preguntar	Usar
Decir	Dirigir
Comunicar con	Operar

- Se debe poner al usuario como sujeto de la acción.

Esta documentación ha de ser distinta de la documentación escrita tanto cuantitativamente como cualitativamente y se basa fundamentalmente en los distintos tipos de ayuda siguientes:

En línea:

Por ej. el Help del DOS o en man del UNIX son una lista ordenada de tareas. Su principal defecto, reside en que para usuarios inexpertos es de difícil búsqueda en ella ya que deben conocer previamente la orden concreta que realiza la acción concreta.

Sensible al contexto:

Por ej. F1 en Windows, ofrece ayuda al usuario sobre la acción que en ese momento este esté realizando.

Sensible al puntero de ratón o cursor:

Por ej. los mnemónicos al pasar por encima de los iconos y las líneas de texto que aparecen en la barra de estado al poner el foco en una determinada opción de menú.

Hipertexto:

Por ej. el texto verde y subrayado señalado por una mano que nos lleva tras hacer clip al tema concreto.

4. HERRAMIENTAS PARA LA CONSTRUCCIÓN DE INTERFACES

Debe diseñarse una interfaz ergonómica que no ocasione cansancio visual, auditivo o mental al usuario. Para lograrlo tendremos en cuenta las características de las siguientes herramientas.

Presentación de información

No se deben colocar demasiados objetos en la pantalla, y los que existen deben estar bien distribuidos. Cada elemento visual influye en el usuario no sólo por sí mismo, sino también por su combinación con el resto de elementos presentes en la pantalla.

Cada elemento (por ejemplo, el fondo, una línea, un recuadro, el espacio fuera del recuadro), nuevo que se añade influye más de lo que se piensa en el usuario.

Color

Es probablemente el elemento de la interfaz que con más frecuencia es mal utilizado. El color comunica información, no es sólo decorativo (ejemplo: reforzar mensajes de error). Deben utilizarse combinaciones adecuadas (por ejemplo, las paletas proporcionadas por los sistemas operativos). El color debe atraer la atención, pero no cansar después de un rato de trabajo. Es especialmente importante seguir las líneas de diseño existentes. Principio básico: diseñar primero en blanco y negro, y luego añadir el color.

Audio

- Es preciso ver cuándo es más apropiado que la información visual.
- Determinar el sonido adecuado.
- Permitir la personalización (volumen y desactivación).

Como en el caso de los colores existen guías de uso. En lugares de trabajo abiertos, puede ser poco efectivo; además, puede ser embarazoso para algunas personas. El sonido debe usarse para informar, no cuando no añade nada nuevo (por ejemplo, un mensaje de aviso de correo o de bienvenida, respectivamente, al iniciar una sesión de trabajo).

Animación

Se define como un cambio en el tiempo de la apariencia visual de un elemento gráfico. Ejemplos de su uso: progreso de acciones (copia de ficheros en Windows 95, instalación de programas), estado de procesos (iconos de impresora), acciones posibles (cambios en el cursor al desplazar el ratón). La animación puede ayudar a subrayar iconos importantes, mostrar el estado de un objeto particular o explicar su comportamiento.

Diseño internacional

Debe hacerse un uso adecuado de la terminología. Hay mucho trabajo en este campo. Debe tenerse cuidado con las diferencias culturales (gestos, terminología, dibujos, formatos de teléfonos o calendarios, etc.).

Elección de controles

Muchas veces existe la duda de qué controles utilizar. En realidad no existe una única forma correcta. Un aspecto a considerar es la escalabilidad (menú de 10/1000 elementos; ejemplo: programas del menú inicio de Windows 95).

Usar un menú de barra o de paleta, permitir arrastrar objetos o no (problema: no existe indicación visual de que se pueda arrastrar el objeto: ¿qué objetos se pueden arrastrar? ¿a dónde se pueden arrastrar? ¿qué ocurrirá cuando lleguen allí? ¿se podrá deshacer la acción?).

Existen diversas guías de diseño sacadas de expertos y comités, que complementan a las reglas de oro estudiadas anteriormente. Por citar algunas de ellas:

- Demasiada simetría puede hacer las pantallas difíciles de leer.
- Si se ponen objetos sin alinear, hacerlo drásticamente.
- Asimetría = activo, simetría = sereno.
- Elementos de tamaño y color similares se perciben como pertenecientes a un grupo.
- Asumir errores en la entrada del usuario.
- Diseñar para el usuario, no para demostrar los propios conocimientos tecnológicos.
- Unos gráficos espectaculares no salvarán a una mala interfaz.

Para ver ejemplos sacados del mundo real de buenos y malos diseños (si bien no del campo de la computación) puede visitarse la página WEB Bad Human Factors Designs de Michael Darnell en <http://www.baddesigns.com>.

RESUMEN

Las fases del diseño de una interface gráfica son:

- Recopilar información.
- Definir los requisitos y la semántica (Pseudocódigo).
- Especificar la sintaxis y las funciones.
- Especificar dispositivos físicos.
- Desarrollo del software.
- Integración del sistema y distribución a los usuarios.
- Información a los usuarios.
- Plan de evolución.

Entre las consideraciones a tener cuenta a la hora de diseñar interfaces se encuentran las siguientes:

- Características deseadas: simplicidad, claridad y fácil de comprender. Será necesario tener claridad visual, de forma que los elementos estén agrupados de forma comprensible y con significado en vez de al azar y de forma confusa.
- Saber dónde situar la información en la pantalla. Será necesario indicar un punto de partida obvio en la esquina superior izquierda de la pantalla, reservar áreas específicas de pantalla para diferentes tipos de información (como, por ejemplo, mandatos, mensajes de error, títulos y campos de datos, de forma que esta consistencia se mantenga en todas las pantallas) y proporcionar una composición que guste visualmente (es decir, que esté balanceada, sea simétrica, sea predecible, secuencial, simple, con agrupamientos, etc.).
- Saber qué información situar en la pantalla. Para ello, hay que poner sólo la información que es esencial para la toma de una decisión o para la ejecución de una acción (¡No inundar al usuario con información!) y poner todos los datos relacionados con una tarea en una única pantalla (así el usuario no tiene que recordar datos de una pantalla a otra).

- Saber cómo situar la información en la pantalla. Así, en cuanto a las fuentes de letras, se recomienda utilizar minúsculas para el texto con la letra inicial de la frase en mayúsculas; para las etiquetas, encabezamientos o subtítulos utilizar mayúsculas. En cuanto a las palabras, se recomienda no usar jerga, sino utilizar palabras cortas, familiares, etc. También es necesario saber como alinear y/o resaltar el texto y las palabras, dónde situar las ilustraciones, los campos de datos, etc.

- La interfaz de entrada debe recoger todos los datos necesarios, sin introducir errores, para el sistema. De esta forma, la interfaz contiene una protección contra errores de entrada. Así mismo, también debe recoger los datos minimizando el número de teclas pulsadas por el usuario. Las entradas deben estar bien estructuradas y ser fáciles de comprender y utilizar. Se deben usar nombres precisos y permitir abreviaturas cuando se necesiten introducciones rápidas de datos. Se deben evitar las entradas repetitivas. Igualmente, el diseño de la salida asegura que se extraen todos los datos suministrados por el sistema y que esas salidas están estructuradas de forma que sean fáciles de leer.

- El color añade una nueva dimensión a la facilidad de uso de la pantalla, ya que atrae la atención del usuario. Si se utiliza de forma adecuada, puede resaltar la organización lógica de una pantalla, facilitar la separación de componentes y acentuar las diferencias. Por el contrario, si se usa inadecuadamente, puede distraer y fatigar la visión debilitando la facilidad de uso del sistema. Por ejemplo, en las pantallas gráficas se recomienda no utilizar más de seis colores a la vez, evitar colores extremos (rojo y azul, amarillo y púrpura), evitar colores que no tengan contraste (blanco y amarillo, rojos, azules).

Para finalizar, diremos que el diseño de interfaces de usuario es un proceso ordenado que empieza en los requisitos y finaliza con la implementación.

EDITA Y DISTRIBUYE:

CENTRO DOCUMENTACIÓN DE ESTUDIOS Y OPOSICIONES
C/ CARTAGENA, 129 - (91) 564 42 94 - 28002 MADRID - <http://www.cede.es>