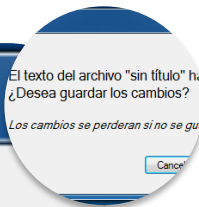
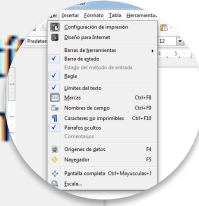
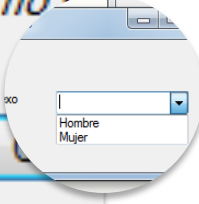


[illegible]

1. Interfaces de usuario



2. Usabilidad



Desarrollo de aplicaciones multiplataforma

TABLA DE CONTENIDOS

1. INTERFACES DE USUARIO	5
1.1. INTRODUCCIÓN	5
1.2. CONCEPTOS DE INTERFAZ	5
1.3. CLASIFICACIÓN	6
1.4. CARACTERÍSTICAS DE UN GUI	9
1.5. TIPOS DE INTERFACES DE USUARIO	9
1.6. DISEÑO DE INTERFACES	10
1.7. CARACTERÍSTICAS HUMANAS DEL DISEÑO DE INTERFAZ	12
1.8. PASOS PARA EL DISEÑO DE INTERFAZ	13
1.9. PRESENTE Y FUTURO.	15
2. USABILIDAD	17
2.1. INTRODUCCIÓN A LA USABILIDAD	17
2.2. PAUTAS DE DISEÑO DE LAS INTERFACES DE USUARIO	27
2.2.1. DISEÑO DE LA ESTRUCTURA DE LAS INTERFACES DE USUARIO.	30
2.2.2. DISEÑO DEL ASPECTO DE LAS INTERFACES DE USUARIO	39
2.2.3. DISEÑO DE LOS ELEMENTOS INTERACTIVOS DE LAS INTERFACES DE USUARIO	46
2.3. OTRAS PAUTAS DE DISEÑO	49
2.3.1. PRESENTACIÓN DE LOS DATOS.	50
2.3.2. SECUENCIA DE CONTROL DE LA APLICACIÓN	51
2.3.3. SEGURIDAD DE LA INFORMACIÓN	52
2.3.4. ESPECÍFICAS PARA APLICACIONES MULTIMEDIA	52
REFERENCIAS	54

1. Interfaces de usuario

1.1. Introducción

Para muchos usuarios de los sistemas de información, la frustración y la ansiedad forman parte de su vida diaria. Luchan por aprender el lenguaje de órdenes y los sistemas de selección que supuestamente ayudan a realizar su trabajo. Evitar estos problemas es el objetivo principal del Diseño de Interfaces de Usuario.

Los Avances de la Ciencia y la Tecnología han puesto al hombre en un plano intermedio entre lo tangible e intangible, computacionalmente hablando, es ahora tan común el convivir con un computador diariamente que cada vez se hace más imperativo la mejor interacción hombre-máquina a través de una adecuada interfaz (Interfaz de Usuario), que le brinde tanto comodidad, como eficiencia

El Diseño de la Interfaz de Usuario es la categoría de diseño de la aplicación que crea un medio adecuado de comunicación entre el hombre y la máquina.

1.2. Conceptos de interfaz

Lewis y Rieman [1993] definen las interfaces hombre-computadora como:

Las interfaces básicas de usuario son aquellas que incluyen cosas como menús, ventanas, teclado, ratón, los beeps y algunos otros sonidos que la computadora hace, en general, todos aquellos canales por los cuales se permite la comunicación entre el hombre y la computadora.

La idea fundamental en el concepto de interfaz es el de mediación, entre hombre y máquina. La interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como el computador. Esto implica, además, que se trata de un sistema de traducción, ya que los dos "hablan" lenguajes diferentes: verbo-icónico en el caso del hombre y binario en el caso del procesador electrónico.

Por lo tanto podemos definir una Interfaz de usuario como el conjunto de elementos (hardware y software) que actúan como frontera entre el dominio del hombre y el de la máquina con la misión de lograr y facilitar la comunicación entre ambos, casi como un traductor entre el lenguaje del usuario y el del ordenador.

De una manera más técnica se define interfaz de usuario, como conjunto de componentes empleados por los usuarios para comunicarse con las computadoras. El usuario dirige el funcionamiento de la máquina mediante instrucciones, denominadas genéricamente entradas. Las entradas se introducen mediante diversos dispositivos, por ejemplo un teclado, y se convierten en señales electrónicas que pueden ser procesadas

por la computadora. Estas señales se transmiten a través de circuitos conocidos como bus, y son coordinadas y controladas por la unidad de proceso central y por un soporte lógico conocido como sistema operativo. Una vez que la UPC ha ejecutado las instrucciones indicadas por el usuario, puede comunicar los resultados mediante señales electrónicas, o salidas, que se transmiten por el bus a uno o más dispositivos de salida, por ejemplo una impresora o un monitor.

Resumiendo entonces podemos decir que, una interfaz de software es la parte de una aplicación que el usuario ve y con la cual interactúa. Está relacionada con la subyacente estructura, la arquitectura, y el código que hace el trabajo del software, pero no se confunde con ellos. La interfaz incluye las pantallas, ventanas, controles, menús, metáforas, la ayuda en línea, la documentación y el entrenamiento. Cualquier cosa que el usuario ve y con lo cual interactúa es parte de la interfaz. Una interfaz inteligente es fácil de aprender y usar. Permite a los usuarios hacer su trabajo o desempeñar una tarea en la manera que hace más sentido para ellos, en vez de tener que ajustarse al software. Una interfaz inteligente se diseña específicamente para la gente que la usará.

Características de una interfaz

- **Naturalidad.**
- **Facilidad de aprendizaje y uso.**
- **Consistencia.**

1.3. Clasificación

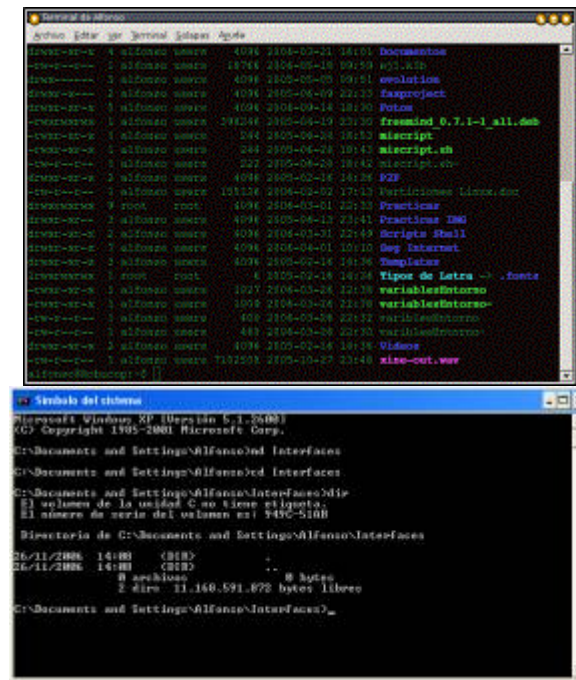
Dentro de las Interfaces de Usuario se distinguen básicamente dos tipos :

- **Interfaz de hardware**, a nivel de los dispositivos utilizados para introducir, procesar y entregar los datos: teclado, ratón y pantalla.
- **Interfaz de software**, destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario observa habitualmente en la pantalla.

De esta clasificación general se puede ir desprendiendo algunas, así por ejemplo según su evolución tenemos: La evolución de las interfaces de usuario corre en paralelo con la de los sistemas operativos; de hecho, la interfaz constituye actualmente uno de los principales elementos de un sistema operativo. A continuación se muestran las distintas interfaces que históricamente han ido apareciendo, ejemplificándolas con las sucesivas versiones de los sistemas operativos más populares.

Interfaces de línea de comandos

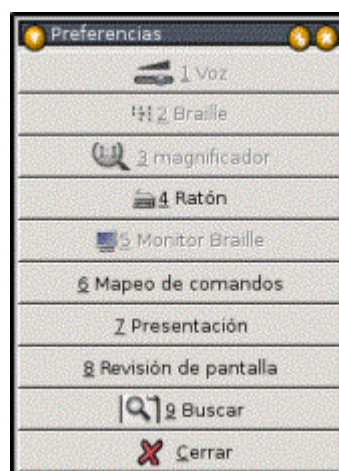
Interfaces de línea de comandos o (*command-line user interfaces*, CUI). Es el interfaz característico del DOS, el sistema operativo de los primeros PC, y es el estilo más antiguo de interacción hombre-máquina. El usuario escribe órdenes utilizando un lenguaje formal con un vocabulario y una sintaxis propia (los mandatos en el caso del DOS). Se usa un teclado, típicamente, y las órdenes están encaminadas a realizar una acción. El usuario no suele recibir mucha información por parte del sistema (ejemplo: indicador del DOS), y debe conocer cómo funciona el ordenador y dónde están los programas (nada está oculto al usuario). El modelo de la interfaz es el del programador, no el del usuario.



En suma, un CUI es adecuado para usuarios expertos, no para novatos. Para aquellos resultan más rápidos, por lo que se puede diseñar un CUI como parte de una interfaz, para que se pueda utilizar una vez que se tenga experiencia.

Interfaces de menús.

Un menú es una lista de opciones que se muestran en la pantalla o en una ventana de la pantalla para que los usuarios elijan la opción que deseen (véase ejemplo). Los menús permiten dos cosas: navegar dentro de un sistema, presentando rutas que llevan de un sitio a otro, y seleccionar elementos de una lista, que representan propiedades o acciones que los usuarios desean realizar sobre algún objeto. Las interfaces de menús aparecen cuando el ordenador se vuelve una herramienta de usuario y no sólo de programadores. Las actuales interfaces gráficas u orientadas a objetos siguen utilizando este tipo de interfaces (los distintos estilos de interfaces no son mutuamente exclusivos).



Existen distintos tipos de menús. Los primeros fueron los menús de pantalla completa, estructurados jerárquicamente:

- Menú de pantalla completa (Menú BIOS)
- Los menús de barra, situados en la parte superior de la pantalla, son profusamente utilizados en las aplicaciones actuales. Contienen una lista de acciones genéricas que dan paso a menús desplegables donde se concretan.
- Menú de barra y menú desplegable. Estos menús pueden llevar a su vez a otros: son los menús en cascada. Pueden cambiar dinámicamente, y deshabilitar opciones que no estén disponibles en un momento dado (marcándolas habitualmente en gris).
- Menús en cascada, Ejemplo la barra de inicio de Windows 95.
- Las paletas o barras de herramientas son menús gráficos con acciones, herramientas y opciones que se pueden colocar en la pantalla. Se utilizan mucho en programas gráficos.
- Los menús contextuales o menús pop-up son los más recientes. Se llaman así porque el contenido del menú depende del contexto de trabajo del usuario. Contienen únicamente las opciones que son aplicables al objeto seleccionado, más algunas de uso frecuente que también son accesibles desde el menú de barra.



Las interfaces de menús, bien estructuradas, son buenas para usuarios noveles o esporádicos. Son fáciles de aprender y de recordar. Pueden existir menús simples y avanzados, para adaptarse al tipo de usuario. Precauciones: no ocupar demasiado espacio de la pantalla, recordar la información acumulada de menús precedentes, no colocar demasiados elementos en el menú, agruparlos de manera lógica (no en orden alfabético, por ejemplo; esto ayuda a recordarlos), permitir la personalización por parte del usuario, usar una terminología adecuada y consistente dentro del programa y con otros programas (*Exit, Quit, Escape, Close, Return, Back*). Las interfaces de menús serán utilizadas normalmente en conjunción con los otros estilos de interfaces.

Interfaces gráficas de usuario

Las interfaces gráficas de usuario o (*graphical user interfaces*, GUI), fueron desarrolladas originalmente por XEROX (sistema Xerox Star, 1981, sin éxito comercial), aunque popularizadas por Apple (Steven Jobs se inspiró en los trabajos de Xerox y creó el Apple Lisa, 1983, sin éxito, y Apple Macintosh, 1984, con éxito debido en gran medida a su campaña publicitaria)

Los tres estilos más comunes de interfaces gráficas hombre-computadora son: Lo que tú ves es lo que puedes conseguir (**WYSIWYG** *What you see is what you get*), Manipulación directa e Interfaces de usuario basados en iconos.

Una GUI es una representación gráfica en la pantalla del ordenador de los programas, datos y objetos, así como de la interacción con ellos. Una GUI proporciona al usuario las herramientas para realizar sus operaciones, más que una lista de las posibles operaciones que el ordenador es capaz de hacer.



1.4. Características de un GUI

- Posee un monitor gráfico de alta resolución.
- Posee un dispositivo apuntador (típicamente un ratón).
- Promueve la consistencia de la interfaz entre programas.
- Los usuarios pueden ver en la pantalla los gráficos y textos tal como se verán impresos.
- Sigue el paradigma de la interacción objeto-acción.
- Permite la transferencia de información entre programas.
- Se puede manipular en la pantalla directamente los objetos y la información.
- Provee elementos de interfaz estándar como menús y diálogos.
- Existe una muestra visual de la información y los objetos (iconos y ventanas).
- Proporciona respuesta visual a las acciones del usuario.
- Existe información visual de las acciones y modos del usuario/sistema (menús, paletas).
- Existen controles gráficos (*widgets*) para la selección e introducción de la información.
- Permite a los usuarios personalizar la interfaz y las interacciones.
- Proporciona flexibilidad en el uso de dispositivos de entrada (teclado/ratón).

Una característica importante es que el GUI permite manipular los objetos e información de la pantalla, no sólo presentarla. Para usar un GUI, los usuarios deben conocer (o aprender) una serie de conceptos: organización del sistema (ficheros, directorios, configuraciones, etc.), diferentes tipos de iconos y efecto de las acciones sobre ellos, elementos básicos de una ventana, uso de los controles de la GUI, uso del ratón.

Las GUI usan el estilo objeto-acción, en contraposición al acción-objeto de las GUI o las interfaces de menú. El usuario selecciona un objeto, y después la acción a realizar sobre dicho objeto. Los objetos son el principal foco de atención del usuario, lo cual resulta más natural y próximo a su modelo mental.

1.5. Tipos de interfaces de Usuario

Interfaces de Escritorio

Su principal característica es el uso de la metáfora del Escritorio podemos decir que básicamente la interfaz de Windows, hasta Windows 8, se centra en la presentación de información al usuario mediante ventanas, en las que se ejecutan aplicaciones o

simplemente se muestra información de modo interactivo. La organización de la información se realiza mediante el almacenamiento de datos en archivos de diferentes tipos (imágenes, documentos, sonidos, etc.) que agrupa en carpetas según el criterio del usuario. Por ello podemos considerar el explorador de Windows, como el primer interfaz de este sistema operativo en lo que a gestión del equipo se refiere.

Actualmente están proliferando en la Red aplicaciones muy específicas con interfaces altamente innovadoras, muy útiles para trabajar en línea o para el propio ordenador personal. Desde los conocidos sistemas con la meteorología local hasta avanzados buscadores, pasando por multitud de diccionarios, relojes o aplicaciones para acceder rápidamente a servicios web.

Interfaces Web

Todas las Interfaces para la Web están destinadas a ser ejecutadas en los navegadores. Suelen tener gran cantidad de componentes gráficos, aunque de tamaño reducido, es decir, imágenes y recursos de tamaño menor para facilitar la transmisión y evitar grandes retardos en la carga de la página.

También nuevas tendencias en lo que a las Interfaces se refiere:

- Filtrado de los datos mediante conjuntos de deslizadores
- Ojo de Pez
- Mapas de Árboles

Algunas páginas se basan en la sencillez y en la facilidad de uso, otras por el contrario son mucho más sofisticadas con lo que pretenden llamar la atención del usuario. Cada vez se hace más indispensable el uso de sistemas de ayuda adecuados para facilitar la interacción de los usuarios, especialmente cuando se intentan técnicas o procesos innovadores.

De las técnicas de creación de páginas Web entre las que podemos destacar el uso de hojas de estilo CSS (*Cascade Style Sheets*) mediante las que es posible cambiar el aspecto de una página fácilmente, además de hacer diferentes configuraciones de presentación según las preferencias de los usuarios.

1.6. Diseño de interfaces

Desde el punto de vista de la ingeniería de software, la Interfaz de usuario juega un papel preponderante en el desarrollo y puesta en marcha de todo sistema. Es la carta de presentación de todo el proyecto y en ocasiones resulta determinante para la aceptación o rechazo del mismo.

Se estima que del 35% al 45% de los gastos destinados a un proyecto están directamente relacionados con el diseño de la Interfaz.

Para tratar el diseño de interfaces, es necesario centrarnos en primer lugar en tres principios fundamentales del diseño de la comunicación:

- **Modelos de Diseño.** Modelos basados en el tipo de usuario.
 - **Usuario Experto,** no necesita unos menús muy intuitivos, ni unos iconos con unas explicaciones muy claras, sino que con la experiencia y el sentido común un usuario experto puede usar una interfaz que se preocupa más de ofrecer mucha funcionalidad que mucha información.

- **Usuario intermedio**, son usuarios que se encuentran entre los expertos y los novatos. Pueden tener un nivel mediano de conocimiento o uso de las TIC.
- **Usuario novato**, un usuario sin mucha experiencia. Necesita una interacción con el software mucho más guiada, con mucha más ayuda, tanto textual como gráfica.
- **Presentación de la Información.** Definición de los objetos en pantalla, colores, audio, gráficos, uso de metáforas, etc.
 - **Distribución de objetos en pantalla.** No se debe sobrecargar la pantalla colocando demasiados objetos, y los que existen deben estar bien distribuidos.
 - **Uso del color.** El color no sólo es decorativo, también transmite información. Es recomendable diseñar en blanco y negro, pasándolo después a color, así como utilizar una paleta de colores reducida o limitada al aspecto corporativo, como la de los sistemas operativos.
 - **Uso del audio.** Se recomienda el uso de información auditiva cuando aporta información, y siempre ha de permitirse desactivar y regular el volumen. Dependiendo del tipo de las aplicaciones, muchas interfaces suelen añadir una música de ambiente.
 - **Animación.** El uso de la animación puede ser muy importante en situaciones en que el usuario tenga que soportar una moderada espera. Un ejemplo de esto podemos verlo durante la copia de ficheros en Windows o con el cambio del reloj de arena durante un proceso. En ocasiones se usa para aportar atractivo a la Interfaz y llamar la atención en determinadas tareas.
 - **Uso de atajos.** También llamados **shortcuts**, son las teclas rápidas que permiten agilizar acciones (Ctrl+C - Copiar, Ctrl+V - Pegar, por ejemplo). Son muy apreciados por los usuarios expertos que se desenvuelven con soltura en una aplicación con un mayor uso del teclado.
 - **Anticipación.** Las aplicaciones deben intentar anticiparse a las necesidades del usuario, y no esperar que éste busque las opciones. Esto resulta especialmente útil cuando en determinadas situaciones se activan los controles que podemos usar en cada momento, mientras que el resto permanecen desactivados, es algo así como decirle al usuario "**Éstas son las opciones que tienes en esta situación**".
 - **Personalización.** No solamente los menús deben ser personalizables, en muchas ocasiones se debe ofrecer al usuario información visual sobre el estado de su trabajo, temas pendientes, alertas y atajos a lugares de uso frecuente, así como al estado inicial de la aplicación. Un buen ejemplo serían las opciones de favoritos e inicio (Home) de los navegadores de Internet, en las que cada usuario tiene las suyas personales.
 - **Relación con el mundo real.** El uso de iconos que representan objetos del mundo real facilita al usuario la identificación de las acciones que desea ejecutar.
 - **Uso de metáforas.** Generan figuras mentales fáciles de recordar. Buenas metáforas serían el escritorio, la papelera de reciclaje o las ventanas.
 - **Características internacionales.** Problemas derivados de dichas características son mucho más usuales de lo que puede parecer a simple vista. Simplemente el formato de fechas o separadores decimales suele

causar más de un quebradero de cabeza. Esos problemas se acentúan cuando la aplicación va a ser utilizada en países con formas de escritura distintas (hebreo, chino, etc.) y alfabetos propios.

- **Atención a la diversidad.** Usuarios con problemas visuales, auditivos o motrices deben tener acceso a la información de nuestras aplicaciones y utilizarlas como cualquier otro usuario. Para ello se debe realizar una adaptación de la Interfaz o utilizar interfaces adaptativas (que veremos en un apartado posterior).
- **Asistencia y ayuda al usuario.** Sistema de documentación, ayuda en línea, contextual. Todo usuario necesita ser guiado de cara a conocer la Interfaz de una aplicación.
 - **Manuales de usuario.** También llamados Manuales de Procedimiento. Son extensos documentos (ahora suelen ser electrónicos con sencillos sistemas de búsqueda) donde se describe con detalle el funcionamiento de la aplicación. Pueden incluir parte de lo que durante el desarrollo fue la Especificación de Requisitos de Software o, incluso, el Análisis Funcional, aplicándose los mismos criterios de redacción, así como la necesidad de tener dichos manuales siempre actualizados.
 - **Ayudas en línea.** Presente en el ordenador, forma parte de la propia aplicación (.hlp y .chm). Presenta las grandes ventajas de permitir búsqueda indexada o libre, índices de contenido de rápido acceso y referencias cruzadas. Su mayor problema es que, en contra de los manuales de usuario, no permite una lectura profunda y continuada, ya que leer en pantalla provoca mucho más cansancio que leer en papel y está limitado el espacio.
 - **Ayuda contextual.** Forman parte de la ayuda en línea, si bien se refieren al elemento sobre el que el usuario está trabajando en ese mismo momento. Dependiendo de la aplicación, pueden invocarse pulsando F1 con el foco sobre el control cuyo uso se quiere consultar, mediante una opción en el menú pop-up surgido al pulsar el botón derecho del ratón, o bien arrastrando el icono de interrogación que se encuentra en la hilera superior derecha de botones de algunas ventanas.
 - **Tutoriales.** Son guías de usuario conceptualmente diseñadas para ser leídas ordenadamente, a modo de paseo por la aplicación. Están dirigidas a usuarios novatos, y muchas veces son incluidos en la propia ayuda en línea del programa.

1.7. Características humanas del diseño de interfaz

Factores Humanos

Al diseñar interfaces de usuario deben tenerse en cuenta las habilidades cognitivas y de percepción de las personas, y adaptar el programa a ellas.

Así, una de las cosas más importantes que una interfaz puede hacer es reducir la dependencia de las personas de su propia memoria, no forzándoles a recordar cosas innecesariamente (por ejemplo, información que apareció en una pantalla anterior) o a repetir operaciones ya realizadas (por ejemplo, introducir un mismo dato repetidas veces).

La persona tiene unas habilidades distintas de la máquina, y ésta debe utilizar las suyas para soslayar las de aquella (como por ejemplo la escasa capacidad de la memoria de corto alcance).

- **Velocidad de Aprendizaje.** Se pretende que la persona aprenda a usar el sistema lo más pronto posible.
- **Velocidad de Respuesta.** El tiempo necesario para realizar una operación en el sistema.
- **Tasa de errores.** Porcentaje de errores que comete el usuario.
- **Retención.** Cuánto recuerda el usuario sobre el uso del sistema en un período de tiempo.
- **Satisfacción.** Se refiere a que el usuario esté a gusto con el sistema.

Además de éstos existen otros a considerar:

Adecuación

- **Características Físicas.** Cada persona tiene diferentes características físicas. Hay algunas personas que no les gustan los teclados mientras que a otras sí. Es por eso que hay teclados ergonómicos. Lo mismo sucede con el mouse.
- **Ambiente.** El lugar donde va a ser usado el sistema. Cada interfaz tiene que adecuarse al lugar.
- **Visibilidad.** Tomar en cuenta la cantidad de iluminación del lugar. ¿Se refleja el brillo en la pantalla?
- **Personalidad.** De acuerdo a la edad, nivel socio-económico, etc.
- **Cultura.** Los japoneses no tienen las mismas pantallas, ventanas, etc. Este factor es importante si el mercado para el sistema es a nivel internacional.

Motivación

- **Sistemas Vitales.** Son de vida o muerte; muchas personas dependen de ellos. Ejemplo: un sistema para reactores nucleares. Este sistema trabaja en tiempo real, y es de suma importancia la seguridad y efectividad del mismo.
- **Sistemas Comerciales e Industriales.** Sirven para aumentar la productividad y vender más.
- **Sistemas de Oficina, Hogar y Juegos.** Factor importante: el mercado a quien está dirigido; tienen que ser muy amigables y satisfacer al cliente.
- **Sistemas de Investigación.** Realizan tareas muy específicas y tratan de imitar el medio en el que se desenvuelve el usuario.

1.8. Pasos para el diseño de un interfaz

Pasos Clásicos

En el proceso de diseño de una interfaz de usuario se pueden distinguir cuatro fases o pasos fundamentales:

1. Reunir y analizar la información del usuario
 2. Diseñar la interfaz de usuario
 3. Construir la interfaz de usuario
 4. Validar la interfaz de usuario
- **Reunir y analizar la información del usuario:** Es decir concretar a través de técnicas de requerimiento, qué tipo de usuarios van a utilizar el programa, qué tareas van a realizar los usuarios y cómo las van a realizar, qué exigen los

usuarios del programa, en qué entorno se desenvuelven los usuarios (físico, social, cultural).

- **Diseñar la interfaz de usuario:** Es importante dedicar tiempo y recursos a esta fase, antes de entrar en la codificación. En esta fase se definen los objetivos de usabilidad del programa, las tareas del usuario, los objetos y acciones de la interfaz, los iconos, vistas y representaciones visuales de los objetos, los menús de los objetos y ventanas. Todos los elementos visuales se pueden hacer primero a mano y luego refinar con las herramientas adecuadas.
- **Construir la interfaz de usuario:** Es interesante realizar un prototipo previo, una primera versión del programa que se realice rápidamente y permita visualizar el producto para poderlo probar antes de codificarlo definitivamente.
- **Validar la interfaz de usuario:** Se deben realizar pruebas de usabilidad del producto, a ser posible con los propios usuarios finales del mismo. Es importante, en suma, realizar un diseño que parta del usuario, y no del sistema. Existen 11 pasos en el proceso de diseño "centrado en las tareas", similar al anterior pero que desglosa algunas actividades implícitas en otras, así:
 1. Entender quien usará el sistema para hacer qué.
 2. Elegir tareas representativas para el diseño.
 3. Plagiar, copiar o imitar.
 4. Esbozar un diseño.
 5. Pensar acerca del diseño.
 6. Crear un prototipo.
 7. Evaluarla con los usuarios.
 8. Repetir.
 9. Construirla.
 10. Rastrearla.
 11. Cambiarla.

Técnicas y pasos avanzadas para el diseño de interfaces de usuario

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.
- El número de elementos visuales que perciben son 7 ± 2 . Cada elemento nuevo que se añade influye más de lo que se piensa en el usuario.

Elementos de diseño de pantalla y su percepción visual

- **Análisis de 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. Un Principio básico es diseñar primero en blanco y negro, y luego añadir el color.
- **Análisis Audio.**
 - Primero es preciso ver cuándo es más apropiado que la información visual.
 - Segundo, determinar el sonido adecuado.

- Tercero, 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).
- **Análisis 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, 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.).
- **Análisis y 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).

Guías de Expertos

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.

1.9. Presente y futuro.

En el mundo del cine, muchas veces nos introduce en lo que respecta a la relación que tendremos con las máquinas, parece que no es difícil intuir cómo pueden ser las interfaces de usuario que utilizarán las próximas generaciones, incluso es posible que nosotros mismos en unos pocos años. Pero según los expertos las tendencias auguran líneas de evolución bien definidas:

- Entornos altamente virtualizados (realidad virtual).
- Máquinas más semejantes al hombre. Que sigan los mismos patrones deductivos y que se comuniquen de una forma humana.
- Los entornos del futuro constarán con escenas, sonidos y campos táctiles sintetizados, reaccionarán además a mandatos y peticiones explícitas, a estados de ánimo y temperamento.
- Representaciones de información que reaccionen a las características sensoriales, preferencias, habilidades y necesidades de cada individuo.

- Que proporcione contextos y materiales para la síntesis cognitiva y emocional de la información.
- Finalmente los expertos en este terreno, reconocen cuatro líneas de investigación: reconocimiento de caracteres, síntesis de habla, reconocimiento de habla y procesamiento de imágenes.

2. USABILIDAD

Las interfaces gráficas de usuario son las responsables de la interacción que habrá entre los hombres y las máquinas. Una parte importante es la confección de estas interfaces y las herramientas que se usarán, como, por ejemplo, sus elementos como los componentes o los controles. Hace falta también tener en cuenta algunos aspectos concretos de esta interacción cuando se hace el diseño de las interfaces.

2.1. Introducción a la usabilidad

Hay muchas interfaces gráficas de usuario en muchos ámbitos de la vida cotidiana, y las personas son usuarios habituales de muchas interfaces a lo largo del día. habrá algunas interfaces de uso obligado, sin opción a escoger si nos gustan o no, o si se podrían usar otros. Por ejemplo, con el cuadro de interacción de una radio de un coche, se tendrá poco margen de maniobra. Si gusta, bien. Si es poco amigable, o difícil de entender, nos tendremos que adaptar, o bien cambiar de aparato de radio al coche. Pasará una cosa parecida con los cajeros automáticos de las entidades bancarias. Si queremos sacar dinero, nos tendremos que adaptar.

En cambio, con otras interfaces sí que hay alternativas. Si queremos buscar una información o una noticia en Internet, accederemos en un portal, y si su interfaz no es amigable o usable o comprensible, quizás intentaremos buscar la misma información por otros medios o en otros portales de Internet. Una página web con una buena usabilidad tendrá mucho más éxito que otra, con contenidos o informaciones similares, pero una usabilidad no tan buena.

“La usabilidad sirve para adaptar el software a los estilos de trabajo reales de los usuarios, en vez de forzar los usuarios a adaptar sus estilos de trabajo al software.”

Justamente esto será el que nos indicará el éxito que tendrá un software concreto una vez esté en funcionamiento. No solamente tendremos que crear software con un funcionamiento correcto (que haga el que los requerimientos indicaban que tenía que hacer) y con una fiabilidad buena (los resultados que nos ofrece el software no pueden ser dudosos), sino que la usabilidad del software tendrá que ser satisfactoria para los usuarios.

Para conseguir esto habrá que tener en cuenta este aspecto a la hora de desarrollar software de todo tipo, estableciendo de este modo en la fase de análisis los requerimientos de usabilidad, unos objetivos que se tendrán que lograr al finalizar el desarrollo de la aplicación. Estos requerimientos de usabilidad guiarán los diseños de las interfaces que interactuarán con los usuarios y, además, los podremos utilizar para evaluar la calidad del software desarrollado.

Como desarrolladores de aplicaciones se trata de un punto a tener mucho en cuenta, puesto que la usabilidad de las aplicaciones que se presentarán a los futuros usuarios será el primero que verán en los prototipos de las aplicaciones o de las interfaces y será

el primero que evaluarán. También hay que indicar qué será el que podrá diferenciar una aplicación de otra, puesto que todas las aplicaciones cumplirán las reglas de negocio.

2.1.1 ¿Qué es la usabilidad?

En primer lugar, habrá que establecer una definición del término usabilidad. Esta palabra deriva del término inglés *usability*, es decir, indica si una cosa es usable o tiene un uso sencillo.

La usabilidad define el grado de facilidad con que un usuario puede usar una herramienta concreta o cualquiera otro objeto.

Pero hay muchas maneras de entender este concepto, puesto que se tienen que tener en cuenta muchos matices para poder definir o entender el significado de facilidad de uso. Además, también habrá que hablar de otros conceptos para definir la usabilidad, términos como efectividad, eficiencia y satisfacción del usuario. Y todos estos conceptos variarán la apreciación en función de si están afectados por los usuarios, sus objetivos o sus expectativas y la situación de uso de la herramienta o aplicación.

Además, habrá que limitar estas definiciones del término usabilidad a un contexto relacionado con la informática y, más concretamente, a las aplicaciones de gestión que desarrollaréis (aunque estas definiciones también servirán para aplicaciones o portales web).

Nada mejor que ver un ejemplo para entender el concepto de usabilidad.

A continuación mostraremos tres ejemplos de interfaces en que se espera una interacción del usuario con una aplicación determinada, como podría ser, por ejemplo, un editor de textos.

En la figura 1 podéis ver una interfaz que se podría considerar con una usabilidad no muy buena.

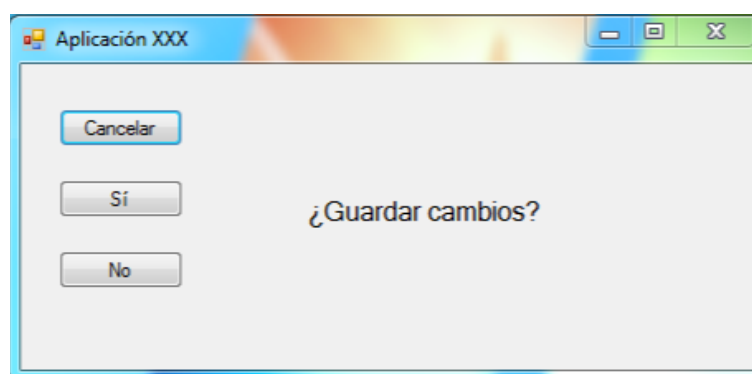


Figura 1. Usabilidad no correcta. Este diálogo no nos muestra las opciones de manera clara e imparcial

Ved cómo hay una pregunta que el usuario tiene que contestar seleccionando la opción que convenga entre las tres posibles. Cómo veis, hay tres botones de respuesta que están a la izquierda, muy alineados, pero con un orden que cuesta un poco de entender. La

pregunta tampoco nos da mucha información. De todas maneras, un usuario con conocimientos mínimos de informática podría interactuar con esta interfaz.

En la figura 2 se puede ver una interfaz con una usabilidad correcta.

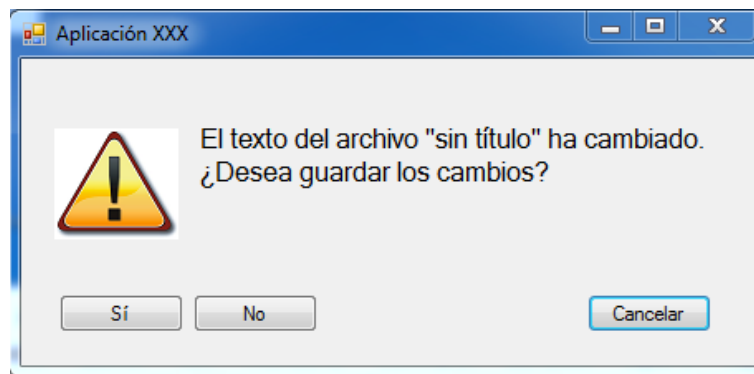


Figura 2. Usabilidad correcta. Este diálogo muestra una usabilidad aceptable.

La figura 2 muestra una imagen que alerta de una situación no prevista, como puede ser que queramos cerrar una aplicación sin haber guardado los cambios. La pregunta que encontramos es bastante más completa y comprensible, y las tres respuestas posibles tienen una ubicación más fácil de identificar, con la primera opción más probable a la derecha.

La propuesta de la figura 3, que muestra un diálogo para la interacción de un usuario con una aplicación. Esta sería de una usabilidad óptima.

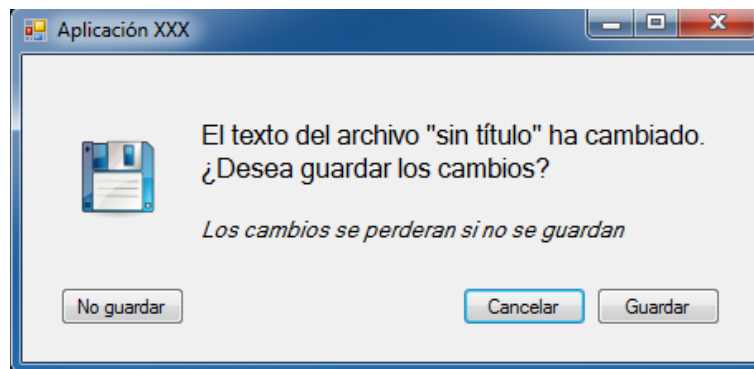


Figura 3. Usabilidad óptima. Este diálogo muestra las opciones de una manera muy clara y sencilla.

En este caso nos encontramos con algunos cambios más. En primer lugar, la imagen que se muestra es más indicativa del tipo de acción que se requerirá, indicando con la imagen del disquete que se trata de una posible operación de carencia de grabación antes de salir de la aplicación.

Además, encontramos que los botones de respuesta para el usuario no están alineados, sino que tienen una distancia irregular entre ellos, buscando un tipo de respuesta por parte del usuario.

También hay que remarcar el hecho que las palabras de los botones de respuesta son verbos, que indican con más claridad las opciones a la hora de seleccionar una.

Normas ISO

Las siglas ISO corresponden a ‘Organización Internacional para la Normalización’. Se trata de una organización no gubernamental fundada el 1947, compuesta por representantes de organismos de normalización de más de cien cincuenta países.

Para establecer una definición más adaptada en nuestro ámbito de estudio, nos fijaremos en las definiciones que podemos encontrar en las normas ISO relacionadas con la usabilidad. Entre otros nos fijaremos en la ISO 9126 y en la ISO 9241. Estas normas ISO nos dan modelos de calidad y calidad en métricas de uso relacionadas con la ingeniería del software y la calidad de los productos desarrollados. La ISO 9241 nos ofrece guías de usabilidad.

A partir de estas normas ISO podemos encontrar definiciones más concretas, como la de la ISO 9241-11, “Guía de la usabilidad” de 1998.

La usabilidad es la parte que trata del grado con que un producto puede ser usado por usuarios específicos para llegar a objetivos especificados con eficacia, eficiencia y satisfacción en un contexto de uso concreto.

Otra definición la podemos encontrar en la norma ISO 9126, que hace referencia a la calidad del producto en la ingeniería del software:

La usabilidad es la capacidad por la que, en el desarrollo del software, el producto final cumpla las funciones de atractivo para el usuario, de capacidad de ser comprendido, aprendido y usado, según las condiciones especificadas de uso.

Como vemos, podemos encontrar muchas definiciones de este término, pero en términos de interacción de las personas con los ordenadores habrá que hablar de la facilidad de uso, la facilidad con que nosotros podremos utilizar una herramienta concreta desarrollada para lograr un objetivo. Pero habrá que añadir otros té como claridad, elegancia con que se ha diseñado la aplicación o lugar web.

Este término también se podrá usar en otros ámbitos como los libros o manuales de usuario (que serán más o menos fáciles de seguir y tendrán más o menos usabilidad), la electrónica de consumo o cualquier objeto mecánico. En todos estos ámbitos se podrá establecer un sistema para medir la utilidad, la facilidad de uso, la facilidad para aprender el uso o la apreciación por parte de un usuario determinado en un contexto determinado.

2.1.2 Dimensiones de la usabilidad: características y atributos

La usabilidad de una interfaz gráfica de usuario determina el grado de desempeño de los objetivos establecidos y permite valorar el resultado. Para conseguir esto es necesario establecer unas características y unos atributos que se puedan valorar. Se conocen como las dimensiones de la usabilidad y son:

- Eficiencia
- Eficacia/efectividad
- Satisfacción
- Atractivo
- Facilitado de aprendizaje
- Facilitado del sistema para ser recordado
- Tolerancia al error

Eficiencia

En términos económicos, la eficiencia se entiende como la relación entre los recursos utilizados y los resultados obtenidos.

Se puede entender la eficiencia en el uso de las aplicaciones informáticas. Es decir, una aplicación que obtiene unos buenos resultados en las medidas de usabilidad aportará más productividad al usuario y permitirá un uso más rápido, más eficiente de la aplicación. Se tratará de poder llevar a cabo más acciones, más interacciones con el software en menos tiempo.

También podríamos entender el término eficiencia en el escenario de desarrollo del software. En el caso de una interfaz de usuario (o de una aplicación informática) los recursos que se habrán utilizado serán las horas destinadas al desarrollo de las aplicaciones o, más concretamente, de las interfaces, junto con el coste de las herramientas de desarrollo empleadas. Los resultados se entenderán como el grado de logros de los objetivos marcados a primeros del desarrollo de las aplicaciones, es decir, en que esta interfaz cumpla los requerimientos.

Eficacia/efectividad

La eficacia o efectividad es la capacidad de conseguir un objetivo planeado o deseado. Es decir, si una aplicación cumple sus objetivos, será eficaz, si no, no lo será.

Cuando hablamos de la usabilidad, hacemos referencia a este término, puesto que será el que nos indicará si las interfaces consiguen todos los objetivos establecidos a comienzos del desarrollo de una aplicación, siendo una interfaz que garantice la efectividad. Además, también hará falta que las funcionalidades establecidas en los menús o iconos de las interfaces hagan el que tienen que hacer.

Satisfacción

La satisfacción se define como un estado de la mente para un ser humano.

¿Puede un usuario de una aplicación informática obtener satisfacción por el uso de esta? Quizás depende de la persona, no lo podemos evaluar con objetividad, pero el que sí que queda claro es que, si conseguimos desarrollar una aplicación con la cual un usuario tiene una actitud positiva cuando la usa (aunque sólo no se sienta incómodo), habremos logrado el objetivo de satisfacción. La satisfacción es el desempeño de los objetivos planteados inicialmente para obtener unos resultados con un cierto grado de gozo para el usuario.

Una vez definidas estas tres características, hay que volver a recordar el concepto de la usabilidad: parte que trata del grado con que un producto puede ser usado por usuarios específicos para llegar a objetivos especificados con eficacia, eficiencia y satisfacción en un contexto concreto de uso.

Quizás ahora esta definición ha quedado más clara. Ahora hay que establecer una serie de guías y normas para llevar a cabo un diseño esmerado a las necesidades de los usuarios, con un grado de la usabilidad más que correcto. Antes, pero, añadiremos algunas características de usabilidad a la definición que hemos hecho más arriba y a sus atributos. Estas características serán:

Atractivo

El significado del término atractivo está muy relacionado con otros ámbitos, pero ahora hay que relacionar esta característica con la usabilidad y las interacciones entre las personas y las máquinas.

Una interfaz es atractiva para un usuario cuando este acepta de buen grado las características y el uso, mostrando una predisposición para utilizarla.

Esta característica queda muy vinculada a otras dimensiones como, por ejemplo, la satisfacción. Una aplicación atractiva ofrecerá al usuario un trabajo agradable y satisfactorio.

Este término también se podría entender como el hecho de producir una actitud positiva en los usuarios.

Facilidad de aprendizaje

Las interfaces tienen que ser muy inteligibles y amigables para permitir el aprendizaje intuitivo por parte del usuario.

Si una interfaz sigue las premisas del entorno en que se ejecutará y usa menús e iconos parecidos, la adaptación y el aprendizaje por parte del usuario serán mucho más rápidos.

Esta característica está asociada al atributo de eficacia. Una aplicación eficaz se tiene que poder aprender y utilizar de manera fácil y rápida. Si una interfaz requiere un uso continuo de la documentación de usuario, su eficiencia no podrá ser nunca buena.

Facilidad del sistema para ser recordado

Una aplicación que se tiene que utilizar de manera esporádica, o de manera regular, pero, por ejemplo, una vez al mes, tiene que cumplir la característica siguiente: se tiene que recordar fácilmente como se usa.

Si las funcionalidades o los iconos son difíciles de interpretar, cada vez que un usuario tenga que interactuar con una interfaz probablemente necesitará utilizar el manual de usuario que indique con claridad cómo se llega a unas determinadas funcionalidades.

Esta situación determinará la predisposición de un usuario a usar esta interfaz y el tiempo que necesitará para utilizarla.

Tolerancia al error

Cuando un usuario interactúa con una aplicación, siempre hay opciones que no se pueden seleccionar en un momento determinado o datos que se tienen que llenar en algún formulario. Hay aplicaciones que no permiten el uso de una opción y no explican por qué, o que permiten que se use, permiten que la aplicación falle.

Otras aplicaciones no permiten seleccionar una opción si no es válida en este momento, o indican al usuario que hay que introducir la letra del DNI, por ejemplo.

Esta característica, como su nombre indica, mide el grado en que la interfaz evita los errores o ayuda a superarlos.

Algunas de estas dimensiones de usabilidad han sido definidas como las 5 E's en el artículo *"Using the 5E's to Understand Users"*. Estas 5 Es están referidas a *effective, efficient, engaging, error tolerant, easy to learn*, es decir, efectividad, eficiencia, atracción/satisfacción, tolerancia al error y facilidad de aprendizaje.

2.1.3 Normas ISO relacionadas con la usabilidad



Hay varias normas ISO relacionadas con el término usabilidad. La ISO (Organización Internacional para la Normalización) se creó con el encargo de promover el desarrollo de las normas internacionales de fabricación, comercio y comunicación, y para todas las industrias derivadas de la electricidad y la electrónica.

Las normas ISO se dividen por códigos que indican diferentes maneras de clasificar las definiciones de estándares, tanto de procedimientos y procesos como requerimientos y atributos de productos y servicios.

En cuanto a la calidad, tenemos las normas 9000, que hacen referencia a los sistemas de gestión de la calidad, en el ámbito de fundamentos y vocabulario, requisitos o directrices.

Si nos fijamos en las investigaciones referentes a los aspectos ergonómicos en la interacción hombre-máquina, se basan en algunos estándares internacionales, como por ejemplo los siguientes:

ISO 9241. Aporta requerimientos y recomendaciones relacionados con las características del software y el hardware, como también del entorno que tiene que mejorar la usabilidad y los principios ergonómicos en el uso de las nuevas tecnologías con terminales visuales. Esta norma está dividida en muchas partes, añadidas a lo largo de los años noventa. Las partes 10 y de la 12 a la 17 son las que están relacionadas con el software, e inciden especialmente en los diseñadores y los responsables de las pruebas de las interfaces de usuarios.

ISO 9126. Esta norma desarrolla el modelo de calidad en el software, proponiendo unos atributos de calidad como la funcionalidad, la fiabilidad, la usabilidad, la eficiencia, la facilidad de mantenimiento y la portabilidad.

ISO 13407. Esta norma explica las actividades requeridas para el diseño de interfaces centradas en el usuario. Estas actividades o requerimientos hacen referencia a todo el ciclo de vida del desarrollo del software, incidiendo en la fase de diseño de interfaces. Es una norma pensada para las cabezas de proyectos informáticos o para los responsables del diseño de interfaces de este. (Esta ISO está siendo revisada en la ISO 9241-210:2010).

Estándar Internacional	Descripción
ISO 9241	Requisitos ergonómicos para trabajo de oficina con terminales de visualización. Parte 1 - Introducción general (1997). Parte 2- Guía sobre los requisitos de la tarea(1992). Parte 10 – Principios de diálogo (1996). Parte 11 – Guía sobre la usabilidad (1998). Parte 17 – Diálogos para rellenar formularios (1998).
ISO 9126	Ingeniería del software. Calidad del producto. Parte 1 – Modelo de calidad. Parte 4 – Calidad en el uso de las métricas.
ISO 6385 (1981)	Principios ergonómicos en el diseño de sistemas de trabajo
ISO 13407 (1999)	Procesos de diseño centrado en el hombre para sistemas interactivos
ISO 10075 (1991)	Principios ergonómicos relacionados con la carga de trabajo mental. Términos generales y definiciones.
ISO/IEC 14598	Tecnología de la información. Evaluación de los productos del software
ISO 11581	Tecnología de la información – Interfaces y símbolos de sistemas de usuario. Símbolos y funciones de iconos.
ISO 11064 (2000)	Diseño ergonómico de los centros de control. Parte 1- Principios para el diseño de los centros de control. Parte 2 – Principios para la ordenación de las salas de control y sus anexos.
ISO/DIS 14915-1 (2002)	Principios ergonómicos del software para interfaces de usuario multimedia. Parte 1. Principios de diseño y estructura. Parte 2 – Navegación y control multimedia. Parte 3 – Selección y combinación de medios.

Tabla 1. Resumen de las normas ISO sobre la orientación de estándares.

2.1.4 Medidas de usabilidad de aplicaciones: tipos de métricas

Llegados a este punto, encontramos que hemos conocido la definición de usabilidad, hemos analizado las dimensiones y hemos visto algunas normas sobre esta. Ahora hay que hacer otro paso adelante y hacernos las preguntas siguientes:

¿Cómo se diseña una interfaz de usuario que cumpla los requisitos de usabilidad? ¿Cómo podemos saber si una interfaz cumple los requisitos de usabilidad que nos hemos planteado?

Precisamente estos son los temas siguientes a tratar. A partir de la segunda cuestión hablaremos de diseño de interfaces, pero ahora hay que hablar de las métricas que podemos usar para medir la usabilidad de las aplicaciones y, en concreto, de sus interfaces.

Volvemos a plantearnos la pregunta: una vez nos encontramos ante una interfaz, ¿cómo podemos evaluar la usabilidad?

Es una pregunta difícil de responder. Hay muchos parámetros que se pueden valorar. Muchos son parámetros subjetivos, difíciles de medir. Quizás para un usuario una

interfaz es intuitiva, es satisfactoria, pero para otro usuario no lo es. Es por eso que tenemos que buscar métricas objetivas.

Podemos encontrar uno de los objetivos de los test de usabilidad según Kit (1995, pág. 96), que nos indica que los test de usabilidad de las aplicaciones se hacen «*para adaptar el software a los estilos de trabajo reales de los usuarios, en vez de forzar los usuarios a adaptar sus estilos de trabajo al software*»

Metodologías de gestión de proyectos

Hay muchas metodologías de gestión de proyectos informáticos y muchos tipos de ciclos de vida de desarrollo del software. Una propuesta de metodología de gestión de proyectos dividiría el desarrollo en estudio de viabilidad, análisis, diseño, desarrollo, finalización y transferencia.

Además, nos tenemos que plantear otras preguntas, como, por ejemplo: ¿en qué parte del desarrollo de aplicaciones informáticas tendremos que llevar a cabo las medidas de usabilidad?, ¿en la fase de fin y transferencia?, ¿durante el desarrollo?, o ¿hay que hacer pruebas en el análisis y durante el diseño? Este punto también es importante. En función del momento en que aplicamos las medidas nos podremos encontrar con unos resultados u otros y, además, estaremos a tiempo de superar los errores que se hayan hecho, o, si lo hemos medido demasiado tarde, quizás no hay posibilidad de volver atrás sin provocar un gran descalabro en la planificación.

Se podrían llevar a cabo algunas evaluaciones cualitativas, pero hay que establecer un sistema cuantitativo de la usabilidad. Para definir cuantitativamente la usabilidad, se han definido algunas métricas que intentan evaluar las aplicaciones desde un punto de vista de las características y atributos de la usabilidad.

Tipo de datos.

Los datos cualitativos son las que nos ofrecen información de calidad, expresada de manera verbal o escrita por los usuarios investigados. Es decir, no buscaremos respuestas controladas como un sí, o un no, o una selección entre diferentes opciones, sino que dejaremos respuestas abiertas porque se escoge la opinión real y subjetiva del usuario.

Los datos cuantitativos son datos que se pueden cuantificar. Estos datos se tratarán posteriormente con métodos matemáticos para llegar a conclusiones. Normalmente las respuestas serán numéricas o booleanas, valoraciones con una respuesta gradual o porcentajes subjetivos, y las preguntas serán mucho más concretas.

Para obtener datos cualitativos hay que conseguir informaciones de usuarios finales. En el apartado siguiente se hablará con detalle de las pruebas de usuario, pero, a modo de ejemplo, podemos entender estos datos cualitativos como el resultado de encuestas, de valoraciones subjetivas, de preguntas con respuesta abierta, opiniones del producto, etc.

En cambio, si hablamos de datos cuantitativos, habrá que analizar el comportamiento del usuario ante la nueva aplicación, web o interfaz, recogiendo informaciones objetivas como pueden ser:

- Tiempo dedicado a cada interfaz antes de escoger una opción
- Tiempo o número a veces de utilización de la ayuda o documentación
- Frecuencia de uso de las opciones
- Número de errores en el uso de las interfaces por usuario o cada cierto tiempo
- Número de acciones completadas en un tiempo determinado
- Comparación de tiempo utilizado en segundas o terceras visitas a una misma interfaz
- Número de sugerencias o quejas del producto

A partir de estas métricas cuantitativas intentaremos evaluar la usabilidad desde el punto de vista de sus dimensiones (efectividad, eficiencia, atractivo, tolerancia al error y facilidad de aprendizaje).

Para evaluar la usabilidad de una interfaz, de una página o portal web o de una aplicación informática, hay muchas técnicas que usan las métricas expuestas más arriba.

Según Rubin [1994], hay cuatro tipos de test. Estos test sirven para evaluar la usabilidad a lo largo de las diferentes fases del ciclo de vida de un proyecto de desarrollo de software.

Estos cuatro tipos de test son:

- **Test exploratorio.** Se llevará a cabo en las fases iniciales del ciclo de vida (requisitos, análisis, diseño). Tendrá como objetivo la evaluación de la eficiencia de los conceptos de diseño inicial y localizar errores iniciales en la definición de las necesidades y asunciones de los usuarios.
- **Test de evaluación de operaciones y aspectos del producto o servicio.** Durante las fases iniciales e intermedias del proyecto (desde la fase de toma de requisitos hasta la de desarrollo). Servirá para evaluar las conclusiones extraídas de los test exploratorios a primeros del desarrollo del software para validar que no se han propagado los errores.
- **Test de validación.** Se lleva a cabo durante las fases finales (pruebas, finalización y transferencia). Servirá para evaluar si el producto final cumple los requisitos predeterminados de usabilidad establecidos al iniciar el proyecto, permitiendo de este modo avanzarse en las posibles deficiencias del producto. Este test certificará la usabilidad del producto.
- **Test de comparación.** Este test implica a todas las fases del proyecto. Irá comprobando el producto con los que ofrece la competencia, y también comprobará las diferentes alternativas de diseño con el objetivo de escoger la más sencilla de usar y de aprender. Este test se podrá aplicar en paralelo con otros tipos de test.

2.1.5 Pruebas de expertos y pruebas con usuarios

Hasta ahora hemos visto el concepto de usabilidad, sus dimensiones, sus características; hemos visto métricas para medirla, pero ahora habrá que analizar las interfaces

mediante pruebas hechas por expertos en la materia y pruebas hechas por usuarios finales de las aplicaciones desarrolladas.

Estas pruebas estarán relacionadas con las características y dimensiones, como también la medida de la usabilidad y los tipos de métricas existentes.

Utilizarán algunas de las métricas descritas, tanto las cualitativas en el caso de las pruebas de expertos como las cuantitativas en las pruebas con usuarios.

En primer lugar, habrá que tener muy claro que el que probamos será el sistema desarrollado (y, sobre todo, las interfaces), no los usuarios o verificadores. Cuanto antes hacemos las pruebas dentro del desarrollo del proyecto, mejor será para corregir errores, pero habrá que saber distinguir entre las pruebas que pueden hacer los expertos durante el desarrollo y las interfaces que se podrán mostrar a los usuarios.

En estos casos, habrá que tener en cuenta que ciertas decisiones tomadas a raíz de los resultados de las pruebas se tendrán que justificar muy bien ante los usuarios, acción que implica cierta burocracia. En otras situaciones, las pruebas ayudarán a elegir entre varias alternativas planteadas, puesto que el usuario nos acabará de dar un golpe de mano en algunas decisiones.

Para llevar a cabo las pruebas habrá que desarrollar previamente un plan de pruebas. Este plan tendrá que prever:

- El **alcance** de las pruebas (qué probaremos y que no, hasta donde llegaremos).
- Los **propósitos** de estas (qué son los objetivos, las razones o las justificaciones para probar en cada caso).
- Las **fechas y los lugares** donde se llevarán a cabo (si iremos a casa del usuario, si haremos que venga él, si necesitaremos una ubicación especial, cuánto rato tendrá que durar, en cuántas sesiones diferentes, etc.).
- Los participantes en las pruebas (cuántos usuarios, si los usuarios serán *stakeholders*¹ o no tendrán ninguna vinculación con el desarrollo del proyecto, cuántos expertos, si estarán vinculados con el proyecto o serán externos).

2.2. Pautas de diseño de las interfaces de usuario

A la hora de desarrollar un proyecto de creación de un software, habrá que tener en cuenta, en todas las fases de su ciclo de vida, la creación de las interfaces que facilitarán la interacción con los usuarios. En cada una de las fases se tendrán que tomar decisiones referentes a las interfaces de usuario, en más o menos mide, que tendrán un efecto decisivo en el éxito del proyecto.

Fases de un proyecto

Recordamos que las fases del ciclo de vida de un proyecto de desarrollo de software son:

- Estudio de viabilidad
- Análisis
- Diseño

¹ Es un término inglés utilizado por primera vez por R. E. Freeman en su obra: “*Strategic Management: A Stakeholder Approach*” (Pitman, 1984), para referirse a «quienes pueden afectar o son afectados por las actividades de una empresa»

- Desarrollo
- Finalización
- Transferencia

Para desarrollar las pautas de diseño de las interfaces de usuario de una aplicación informática, se tendrán que tener en cuenta, a lo largo de todas las fases del proyecto, diferentes elementos como por ejemplo la estructura, el aspecto y los elementos de las interfaces de usuario.

Para establecer las pautas de diseño, también habrá que conocer y entender el tipo de usuario para el cual se desarrolla el software y, en definitiva, las interfaces con las cuales interactuará. Estos podrán ser usuarios expertos, usuarios intermedios, o bien usuarios noveles, sin mucha experiencia.

Un usuario experto. Dentro de los usuarios expertos todavía podemos hacer una segunda clasificación. Hay que diferenciar entre el usuario experto en el uso de las tecnologías de la información y el usuario experto en las reglas de negocio en que ubicamos el software. Sea como fuere, un usuario experto no necesita unos menús muy intuitivos, ni unos iconos con unas explicaciones muy claras, sino que con la experiencia y el sentido común un usuario experto puede usar una interfaz que se preocupa más de ofrecer mucha funcionalidad que mucha información. Por ejemplo, para este tipo de usuario experto podremos ofrecer muchas funcionalidades con muchos menús y submenús o con más necesidad de pasos hasta llegar a la opción deseada. Este usuario tendrá la capacidad de encontrar la opción que necesita y podrá navegar por las opciones sin muchos problemas.

Los usuarios intermedios son usuarios que se encuentran entre los expertos y los noveles. En este caso no hablaremos de conocimiento de negocio. Pueden tener un nivel mediano de conocimiento o uso de las TIC. Esto hará que no necesitamos ofrecerlos tantas facilidades ni información como a los usuarios noveles, pero sí que habrá que facilitarlos el trabajo tanto como sea posible.

Un usuario novel. Un usuario sin mucha experiencia (en el uso de las TIC o en conocimientos sobre las reglas del negocio a que pertenece el software) necesita una interacción con el software mucho más guiada, con mucha más ayuda, tanto textual como gráfica. Un usuario novel necesita ver unos iconos mucho más explícitos, que le faciliten la comprensión de las opciones. También necesita tener una ayuda muy accesible y fácil de entender, como, por ejemplo, mostrar pequeñas frases o informaciones cuando el ratón pasa por encima de un icono, indicando la utilidad (llamadas *tooltip* o indicador de función, como se puede ver en la figura 4).

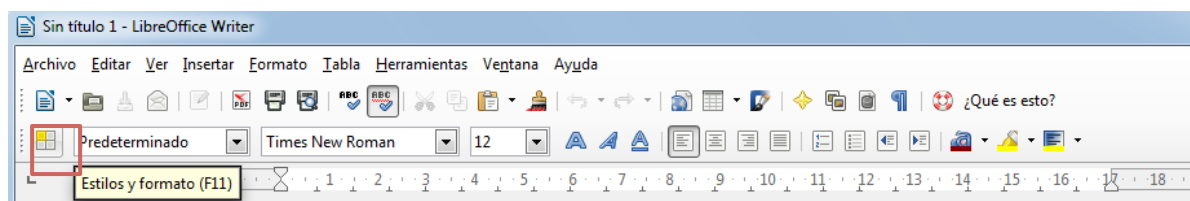


Figura 4. Indicador de función: informa sobre el significado de un icono.

¿Cómo se puede saber cuándo una aplicación se tiene que diseñar pensando en usuarios expertos, usuarios intermedios o usuarios noveles? Esta pregunta no es sencilla de responder.

Una aplicación informática desarrollada desde cero, puede estar inicialmente planteada para usuarios noveles. Si los usuarios usarán la aplicación de manera esporádica, o bien, si habrá una rotación de usuarios muy grande en la utilización de la aplicación, entonces esta se tendrá que diseñar para usuarios noveles.

Si los usuarios siempre serán los mismos, a la cabeza de un cierto tiempo de uso del software, estos mismos usuarios se podrán considerar usuarios expertos y, quizás, reclamarán otro tipo de interfaz. Un ejemplo sería una interfaz de un TPV que usarán diferentes camareros o encargados a lo largo del día en un restaurante.

El mismo pasará con las interfaces web. Tendremos que conseguir un compromiso entre diseño agradable, funcionalidades completas y facilidad de uso.

Por todo esto podemos concluir que habrá que hacer partícipes los usuarios implicados en el proyecto de la decisión del tipo de interfaz que se tendrá que diseñar.

En el diseño de una interfaz habrá que tener en cuenta algunos principios básicos. Por ejemplo, en 1971 Hansen proponía cuatro principios básicos:

- **Conocer el usuario:** sus capacidades, sus necesidades funcionales y su evolución.
- **Minimizar la memorización:** utilizando selección en lugar de entrada de datos, empleando nombres con sentido en lugar de códigos y nombres crípticos, haciendo que los resultados de las operaciones sean predecibles y facilitando el acceso a ayudas y a la documentación.
- **Optimizar las operaciones:** posibilitando la ejecución rápida de operaciones usuales, preservando la consistencia visual, aprovechando las capacidades memorísticas del usuario, y organizando y reorganizando las órdenes de acuerdo con la utilización que el usuario hace.
- **Permitir los errores:** dando al usuario mensajes de error entendedores, diseñando para evitar los errores comunes, permitiendo que las acciones sean reversibles y garantizando la integridad del sistema en caso de errores graves.

La mayor parte de la interacción con un ordenador se hace de manera visual. Esto hace que las decisiones sobre la tipografía, la distribución de la información en la pantalla o de la selección de los colores más apropiados no solamente harán que la interfaz sea más o menos atractiva, sino que la harán más o menos usable y determinarán el éxito. Algunos principios de diseño gráfico a tener en cuenta son:

- El principio de **agrupamiento**. Organizar el espacio visible en bloques separados de controles similares y con un título para cada bloque. Los sistemas de ventanas actuales aplican este principio de manera recursiva. Esto ayudará al usuario a encontrar la información que necesita y a formarse un modelo conceptual del funcionamiento del programa.
- El principio de **visibilidad y utilidad**. Los controles usados frecuentemente tienen que ser visibles y fácilmente accesibles. Análogamente se tendrán que diseñar sistemas para esconder o para comprimir los menos utilizados.

- El principio de **consistencia inteligente**. Utilizar una distribución de la información similar para funciones similares, para habituar los usuarios a encontrar la información en los mismos lugares en situaciones parecidas.
- El principio de **economía del diseño**. Omitir cualquier elemento que no aporte ninguna información a las interfaces gráficas de usuario.
- El principio del **color como suplemento**. Utilizar los colores con medida para enfatizar información, sin ser un elemento exclusivo para comunicar información. El color es más fácil de usar mal que de una manera correcta.
- El principio de **reducción del desorden**. Es un principio que resume el resto de principios. Si sólo son visibles los controles utilizados más frecuentemente, agrupados en grupos con sentido, con un uso minimalista del color y sin elementos superfluos, entonces se tendrá una interfaz bastante atractiva y bastante funcional en la cual el desorden y la arbitrariedad se habrán reducido al mínimo.

A continuación se tratarán diferentes pautas de diseño que tendremos que tener en cuenta tanto en el que afecta a la estructura, al aspecto como a los elementos que componen una interfaz de usuario.

2.2.1. Diseño de la estructura de las interfaces de usuario.

Cuando hablamos de la estructura de las interfaces de usuario nos referimos a todos los elementos que compondrán esta interfaz de usuario y su ubicación dentro de esta. Este conjunto de elementos y su ubicación formarán la estructura de la interfaz.

Los elementos más habituales que podemos encontrar en la estructura de las interfaces, y que nos permitirán la comunicación entre el hombre y la máquina, son los siguientes: menús, ventanas, cuadros de diálogo y atajos de teclado.

Menús

El primer elemento que encontraremos en una interfaz será el menú o los menús, que nos guiarán por todas las posibilidades que nos ofrecerá la aplicación. En función del tipo de aplicación y de la interfaz correspondiente, estos menús serán más o menos necesarios y más o menos profundos.

Por ejemplo, continuando con el caso de una interfaz que se pueda usar a bares o restaurantes, hará falta que nos situamos y nos adaptamos a las necesidades que puedan tener, para entender que, muy probablemente, tendremos interfaces a usar con pantallas táctiles o con terminales portátiles. Si se trabaja con pantallas táctiles no es productivo añadir menús de texto, puesto que su accesibilidad será fuerza complicada. En cambio, las opciones representadas con unos iconos grandes nos permitirán utilizar las interfaces de manera más adecuada.

Los menús son una lista ordenada de opciones que se mostrarán en la interfaz, con posibilidad de contener submenús en cada opción, que facilitarán la interacción con el usuario de una aplicación, ofreciéndole la posibilidad de navegar por la interfaz y de escoger la funcionalidad que quiere utilizar en un momento determinado.

La importancia de los menús en las interfaces radica en el hecho que los usuarios finales de las aplicaciones serán los responsables de su utilización y tendrán que tener la autonomía en su uso, sin la necesidad de una comunicación continua con el programador o diseñador.

A continuación podemos ver algunos ejemplos de diferentes tipos de menús que podremos utilizar en las interfaces de una aplicación informática.

Barra de menús

Es el menú más habitual, tanto a aplicaciones específicas como de uso genérico, como pueden ser de navegación por Internet o de aplicaciones ofimáticas.

Este menú (veáis la figura 5 y figura 6) se suele encontrar en la parte superior de la aplicación. Cada menú contiene otro menú desplegable. Y dentro de estos, en algunas opciones, tenemos otros menús desplegables. Los podríamos denominar los menús en cascada. Las opciones a que podremos acceder serán todas las opciones que nos ofrecerá la aplicación informática, repitiendo algunas de las opciones que nos han ofrecido los iconos que encontraremos en las barras de herramientas.

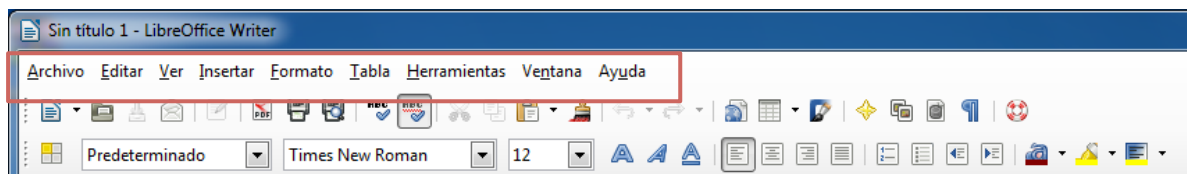


Figura 5. Barra de menús.

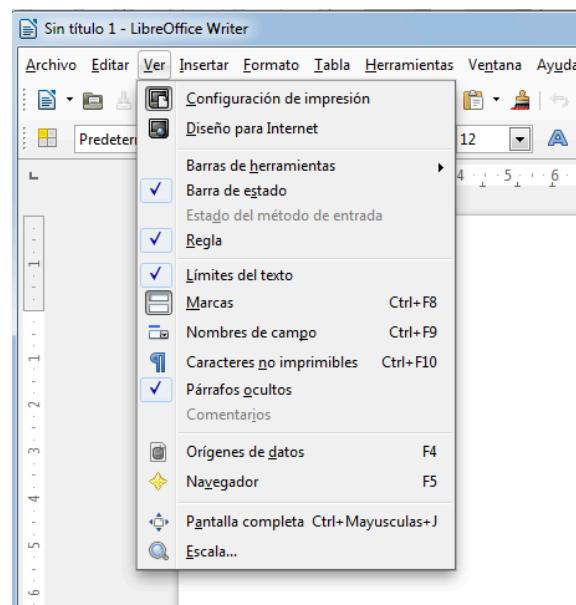


Figura 6. Barra de menús desplegada.

Normalmente, las aplicaciones de uso genérico se pueden personalizar según los gustos o las necesidades de los usuarios, al igual de las paletas o barras de herramientas u otras funcionalidades de las aplicaciones. Esto dará la posibilidad de escoger qué opciones se quieren ver o no en los menús, o que, a veces, las mismas aplicaciones muestren sólo las opciones más utilizadas de manera prioritaria. En otros casos, incluso se podrá seleccionar el idioma de estos menús. Estas posibilidades son bastante más complicadas

de encontrar en las aplicaciones hechas a medida, salvo que el usuario final las haya establecido como requisito.

Barras de herramientas o paletas de herramientas

Este otro tipo de menú (ver la figura 7) es complementario del anterior. En las aplicaciones encontraremos los dos tipos de menús. El 90% de las veces los usuarios usarán los menús de las barras de herramientas, dejando para casos muy concretos la necesidad de acceder a los barra de menús.

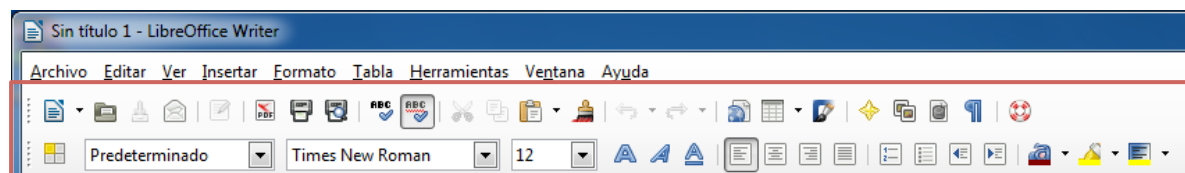


Figura 7. Barra de herramientas.

Cómo hemos comentado, en algunos casos concretos puede llegar incluso a sustituir barra de menús, como en los ejemplos de las aplicaciones desarrolladas para pantallas táctiles o para dispositivos móviles.

Será importante seleccionar correctamente los iconos a usar en las barras de herramientas porque se interpreten correctamente. También hay que indicar con la ayuda de pequeñas explicaciones el significado de los iconos cuando pasamos el ratón por sobre.

Como ejemplo de iconos mostraremos algunas de las más habituales que podremos encontrar en muchas aplicaciones de edición de textos (aplicaciones ofimáticas). Entre ellas podemos encontrar las que hacen referencia en las propiedades de las fuentes o de los párrafo. En la tabla 2 se pueden encontrar algunos ejemplos:

Icono	Descripción/funcionalidad
	Centrar el texto. Pertenece al grupo que hace referencia al párrafo. Esta opción nos permitirá centrar el texto seleccionado en medio de la hoja de edición sin ninguna alineación ni a la izquierda ni a la derecha.
	Justificar. Pertenece al grupo que hace referencia al párrafo. Una vez seleccionado un texto, la justificación lo alineará en los márgenes izquierdo y derecho, agregando espacios adicionales entre las palabras si es necesario. Se crea una homogeneidad en los laterales en la apariencia del texto que se edita.
	Negrita. Pertenece al grupo que hace referencia a la fuente. Aplicar al texto seleccionado la propiedad de texto en negrita, haciendo la fuente algo más gruesa y vistosa sin cambiarla de medida.
	Subrayado. Pertenece al grupo que hace referencia a la fuente. Aplica al texto seleccionado la propiedad de texto subrayado, añadiendo una línea muy fina debajo de los caracteres.
	Color de la fuente. Pertenece al grupo que hace referencia a la fuente. Aplica al texto seleccionado la propiedad de cambio de color. En vez del color por defecto, normalmente el negro, podemos escoger otro color a aplicar en este texto.
	Reducir sangrado. Pertenece al grupo que hace referencia al párrafo. Una vez seleccionado el texto al cual queremos aplicar esta acción, este texto se

desplazará hacia la izquierda, lo cual reducirá los espacios que había entre el comienzo de la línea y el del texto.

Tabla 2. Ejemplos de iconos habituales.

Otras aplicaciones más específicas podrán tener los propios iconos no tan fácilmente identificables. Por esta razón, en este caso o en otras aplicaciones pensadas para usuarios noveles, un recurso como el indicador de función nos podrá ayudar mucho.

En muchas de las aplicaciones, las barras de herramientas se pueden configurar. De este modo el usuario podrá escoger qué funcionalidades puede tener a mano porque son más habituales y qué no le harán tanta falta.

Menús contextuales

El menú contextual es la ventana que se abrirá, en determinados sistemas operativos, cuando hagamos clic con el botón secundario del ratón encima de un espacio concreto de una aplicación. Normalmente son menús que estarán ocultos al usuario y que el usuario activará encima de un objeto determinado. Los menús contextuales se abrirán con una ventana flotante junto al objeto.

Por ejemplo, si hemos empezado a trabajar con un sistema operativo y clicamos con el botón derecho del ratón en el escritorio, veremos una serie de opciones que nos permitirán trabajar con su configuración, ofreciéndonos opciones como la de crear nuevos elementos (carpetas, archivos de ofimática, etc.) o de ir a la configuración otros elementos del sistema.

En caso de usar el menú contextual en uno entorno a sistema operativo, veremos que las opciones que podemos escoger crecen a medida que instalamos programas, de tal manera que será un elemento vivo.

Muchas aplicaciones de uso genérico ofrecen menús contextuales. Por ejemplo, las aplicaciones ofimáticas como la *Libre Office Writer*.

Cómo se puede ver en las figuras 8 y 9 el menú contextual nos ofrecerá unas opciones en función del lugar concreto de la aplicación en que clicamos con el botón derecho. El menú contextual que podremos ver será diferente si se genera en las barras de herramientas que si se genera en el texto que estamos editando. A la vez encontraremos que en un mismo objeto, un menú contextual tendrá unas opciones en un momento determinado y otras en otro momento. Esto será así porque un objeto o un elemento de la interfaz puede pasar por diferentes estados y diferentes contextos a lo largo de la ejecución del software.

De este modo, un menú contextual ofrecerá las funcionalidades que podemos aplicar a un objeto en un momento determinado.

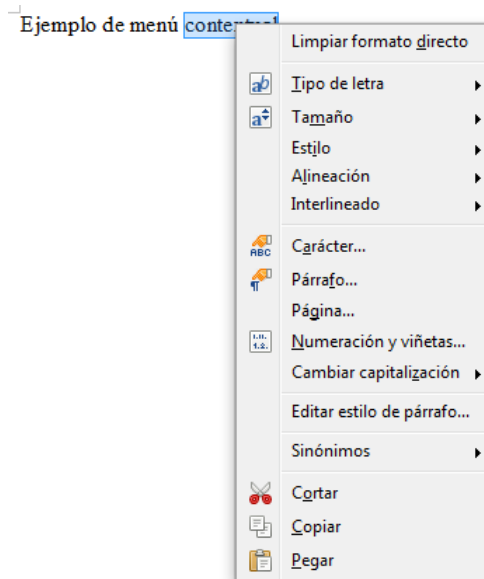


Figura 8. Menú contextual sobre texto

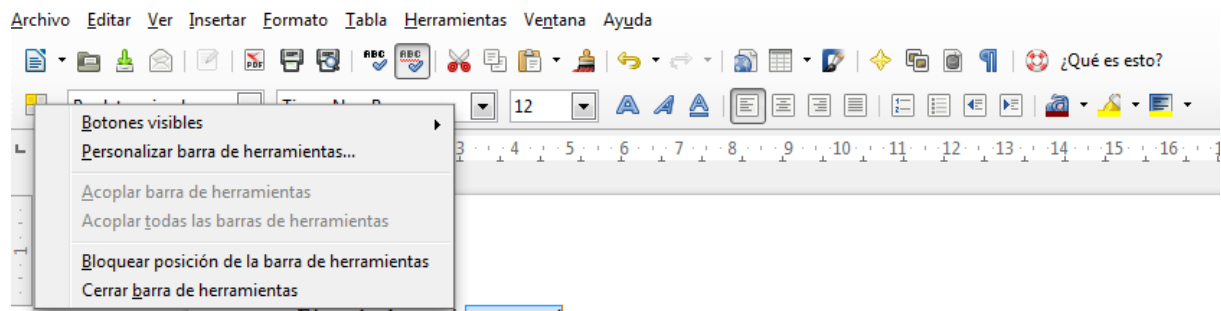


Figura 9. Menú contextual sobre barra de herramientas.

¿Para qué servirá el menú contextual? Nos facilitará el trabajo más habitual con la aplicación que usamos. Cuando abrimos un menú contextual obtendremos una ventana con las funcionalidades relacionadas con la ubicación del puntero (encima de qué objeto se encuentre) en el momento de pedir el menú. En el supuesto de que se muestra en la figura 9, podemos ver como al generar el menú contextual obtenemos un menú con funcionalidades específicas.

Menús desplegables

El *drop-down list* (lista desplegable) es un elemento GUI que se podrá usar en las interfaces, pero también será un tipo de menú. Si lo relacionamos con otros elementos GUI, podemos encontrar algunas similitudes con los cuadros de lista o *listbox* (listas desplegables con todas las opciones visibles) o los *combobox* (listas de opciones desplegables que permiten escribir).

En este apartado nosotros nos fijaremos en los menús desplegables como una opción más de menú para escoger entre una lista de alternativas.

Los menús desplegables ofrecen al usuario la posibilidad de escoger entre un número determinado de opciones. El menú no estará desplegado, sino que hará falta pulsar encima para ver las

opciones que se ofrecen. Cuando el usuario puede escoger una opción, la lista desplegable se cierra, vuelve al estado inicial con la opción seleccionada y aplica la funcionalidad vinculada.

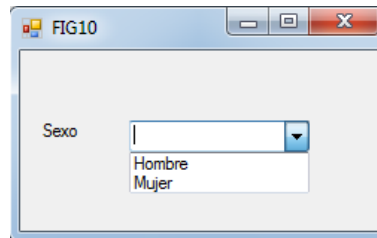


Figura 10. Menú desplegable abierto.

Habría que diferenciar entre los menús desplegables estáticos y los dinámicos. Los estáticos ofrecen un número determinado de opciones que se habrán programado manualmente, que no se modificarán salvo que se modifique el software desarrollado.

Los menús desplegables dinámicos cogerán las opciones de una base de datos o de un fichero. Si se ha modificado la información externa, los menús ofrecerán unas opciones diferentes en diferentes momentos de ejecución.

Menús de pantalla completa

Una opción cada vez más en desuso es la de los menús de pantalla completa. Estos menús eran más habituales con versiones más antiguas de los sistemas operativos, pero todavía se pueden encontrar en algunas aplicaciones o en determinados entornos de trabajo, como las aplicaciones para dispositivos móviles. Para entender un ejemplo de menú de pantalla completa, podemos hacer referencia a los menús que encontramos en las BIOS de las placas madre. Podéis ver un ejemplo en la figura 11.

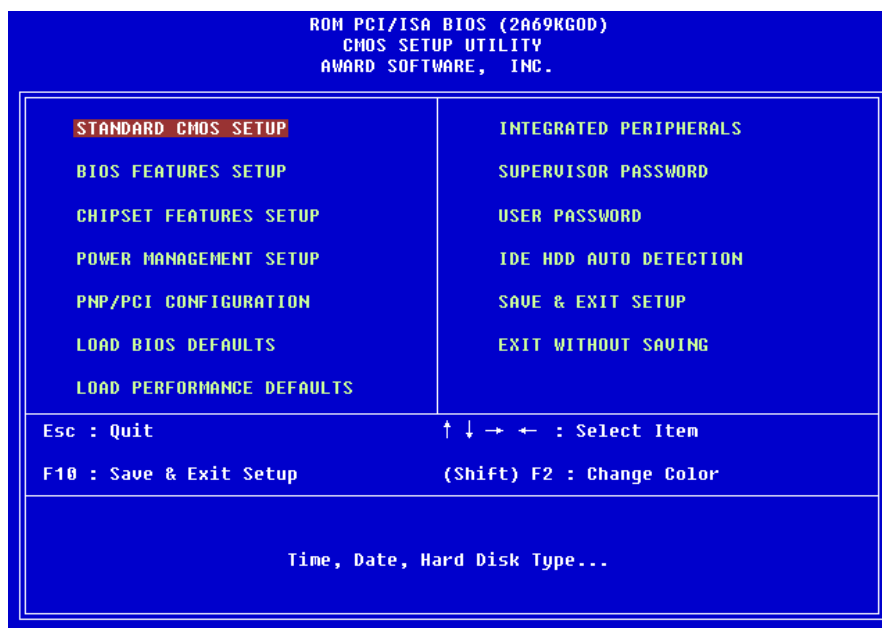


Figura 11. Menú de pantalla completa.

Este tipo de menú monopoliza toda la pantalla y sólo nos permite interactuar con ella, hasta que no seleccionamos la opción de salir. Normalmente, los menús de pantalla completa no permiten la convivencia con otros menús como los contextuales o los de barras de herramientas.

Una vez vistos los diferentes tipos de menús con que podremos trabajar, hay que ver algunas **indicaciones de diseño** de estos a tener en cuenta a la hora de desarrollar nuestras interfaces.

Es importante seguir los estándares de diseño del entorno en que se usará la aplicación (es decir, en función del sistema operativo con que se trabaje, habrá que adaptarse en las pautas de diseño establecidas por el sistema operativo y por las aplicaciones de uso genérico habituales para este entorno).

En términos de usabilidad del menú se tendrán que tener en cuenta algunos aspectos que pueden comportar problemas si no se diseñan correctamente: normalmente los usuarios sólo verán el primer nivel de jerarquía del menú y no, el resto de las opciones. Si tienen que acceder a un segundo o tercer nivel, tendrán que memorizar donde se encuentra, cosa que puede provocar problemas con usuarios de una determinada edad o con problemas psicomotores.

Otras recomendaciones para seguir un estándar en el diseño de menús podrían ser:

- Utilizar una opción de ayuda en los menús de barra o un cuadro de diálogo con los datos de contacto de los desarrolladores del software.
- Se tiene que llegar a un compromiso entre los menús y su número de opciones y de submenús. No se puede hacer una jerarquía tan grande que después nos cueste recordar donde es una opción.
- Los menús no pueden ocupar mucho espacio en la interfaz.
- Los términos se tienen que escoger de manera clara y concisa en las opciones de los menús y en los iconos de las barras de herramientas.
- Los menús se tienen que diseñar de manera visible atendiendo a la mejor usabilidad por parte de los usuarios.
- Se tiene que permitir que los usuarios personalicen, dentro de lo posible, los menús.
- Los menús tienen que ser los adecuados tanto para usuarios expertos como para usuarios noveles.

Ventanas

WYSIWYG

El significado de WYSIWYG es what you see is what you get, es decir, lo que ves es el que recibes.

El sistema más utilizado actualmente en aplicaciones informáticas es el sistema de ventanas. Este sistema sigue las características de los sistemas operativos que permiten un trabajo con más de una ventana (o con más de una aplicación) a la vez siguiendo la

filosofía del *WYSIWYG*. Este sistema permite el acceso de cualquier usuario al ordenador (excepto los usuarios con deficiencias visuales).

WIMP

El significado de WIMP es window, icon, menu, pointer, es decir, ventanas, iconos, menús y punteros.

En términos informáticos, una ventana es un espacio o área que contiene una interfaz de usuario que puede permitir la entrada de datos y muestra la salida del sistema. Las ventanas implementan en concepto *WIMP*.

Las ventanas se pueden clasificar según varios tipos: las ventanas de utilidad, las ventanas de aplicación y las ventanas emergentes o *pop-up*.

Las ventanas de aplicación contienen las barras de menús y las áreas con los documentos con que se trabajará. Son las ventanas principales de las aplicaciones.

Las ventanas de utilidad son ventanas pequeñas que aportan barras de herramientas u otras informaciones a las ventanas de aplicación, ofreciéndolos un complemento.

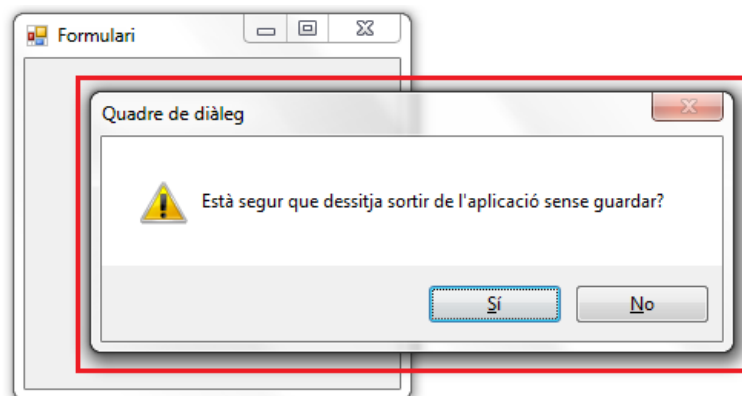


Figura 12. Ventanas emergentes.

Las ventanas emergentes también son ventanas complementarias a las ventanas de aplicación. Llevan a cabo una función de interacción con el usuario ofreciendo alguna información o pidiendo algún dato. Podéis ver un ejemplo en la figura 12.

Algunas de las propiedades más importantes de las ventanas son:

- Las ventanas pueden cambiar de medida, haciéndose más grandes o más pequeñas dentro del área de trabajo.
- Las ventanas se pueden desplazar por la pantalla tanto de manera horizontal como vertical, sólo arrastrándolas por su barra de título.
- Las ventanas pueden estar visibles o no visibles (ocultas), aunque continúen abiertas o, incluso, trabajando. Si una ventana se minimiza, se puede recuperar desde la barra de tareas del sistema operativo.

Cuadros de diálogo

Los cuadros de diálogo o *dialog box* son un tipo de ventana especial que permite ofrecer al usuario una información determinada o recoger de este una respuesta a alguna acción concreta de la aplicación.

Tipo de diálogos

- **Diálogos modales:** mediante ventanas esenciales, de respuesta obligada.
- **Diálogos no modales:** mediante ventanas no esenciales, de respuesta no obligada para los usuarios.

Podemos clasificar los cuadros de diálogo en diferentes tipos: los **esenciales**, los de tipos **alerta** o los que permiten continuar trabajando con un programa. Los cuadros de diálogo **esenciales** son interacciones obligatorias para los usuarios. Los cuadros se abren de manera automática o con alguna acción del usuario, para indicar un error o para prohibir el acceso a una determinada opción o para indicar la introducción incorrecta de algún tipo de dato. Si no se contesta al cuadro, no se puede continuar trabajando.

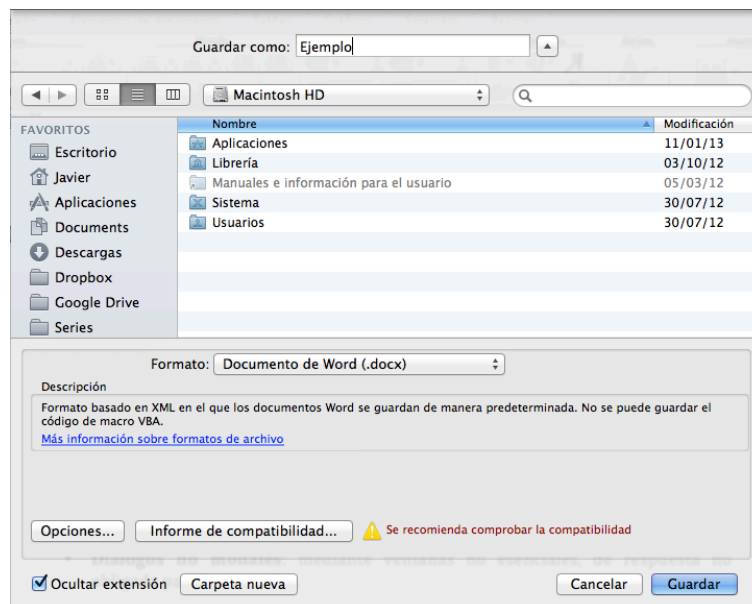


Figura 13. Cuadro de diálogo.

Los cuadros de diálogo no esenciales no interrumpen el funcionamiento de la aplicación, como los anteriores. Se pueden encontrar dentro de la misma ventana de la interfaz o abrirse en una nueva ventana, pero sin la obligatoriedad de responder al mensaje para continuar trabajando. Podéis ver un ejemplo de un cuadro de diálogo se puede observar en la figura 13.

Los cuadros de diálogo de tipo alerta son los que dan una información o un mensaje después de la interacción del usuario, y este tendrá que seleccionar la opción de continuar para poder trabajar. No son esenciales, pero se tipifican como un tipo independiente por su característica informativa de alertas.

Atajos de teclado

Otros elementos importantes en el diseño de las interfaces de usuario son los atajos de teclado que se pueden establecer para acceder a determinadas funcionalidad.

Un atajo de teclado es una combinación de teclas o una combinación de teclas y clic de ratón que nos permite acceder a una funcionalidad de manera directa sin tener que pasar por el menú de barra o las barras de herramientas.

Atajo de teclado

Un ejemplo universal de atajo de teclado es el que nos permite copiar un texto seleccionado (Control+C) y engancharlo en otro lugar de la interfaz (Control+V)

Es importante conocer el entorno a ejecución de las interfaces diseñadas. Un entorno con un sistema operativo Windows ofrecerá una serie de atajos de teclado muy conocidas por usuarios expertos. Por ejemplo, trabajamos con la ventana que trabajamos, con Control+Esc, accederemos al menú Inicio del Windows de manera prioritaria.

También hay atajos de ratón. Por ejemplo, si hacemos clic dos veces en la barra de título de cualquier ventana, esta se maximizará.

Los atajos de teclado (y de ratón) nos permiten trabajar de manera más rápida y más eficiente con el sistema informático y, más concretamente, con la aplicación con la cual estamos trabajando. Es importante dotar una aplicación de atajos para las funcionalidades que se utilicen con más frecuencia, estableciendo estas con los usuarios en las fases de cumplimiento de requisitos.

2.2.2. Diseño del aspecto de las interfaces de usuario

El aspecto de las interfaces es tanto o más importante que la estructura y los otros elementos interactivos. A la hora de diseñar las interfaces de usuario es importante, cómo hemos visto, la estructura, con componentes como los menús, las ventanas, los cuadros de diálogo y los atajos de teclado. Escoger los elementos y donde tienen que ir es una tarea importante.

En el diseño del aspecto de las interfaces de usuario nos tendremos que fijar en temas como los colores, las fuentes que se usarán para los diferentes apartados de la interfaz, los iconos y la distribución de los elementos.

Hay que tener en cuenta que para tratar todos estos aspectos no quedará clara la línea de separación entre el desarrollador de software o de aplicaciones informáticas y el diseñador. Muchas empresas delegan estas funciones a la misma persona, pero, poco a poco, esta tarea se empieza a diferenciar y se contrata una o más personas especializadas en diseño, específicamente para este apartado.

El diseño de la interfaz de usuario es un elemento muy importante (en algunos casos se podrá considerar crítico) en el producto final a entregar. Tenemos que tener en cuenta que algunas decisiones serán muy personales (ya conocemos el dicho “contra gustos no hay nada escrito”), pero se tendrán que seguir unas indicaciones universales.

Colores

Modelo RGB

El modelo codifica cada color en función de la intensidad de los colores primarios que se usan para formarlo. Los colores primarios son el azul, el verde y el rojo. Por ejemplo, el color negro será (0,0,0), es decir, la ausencia de color. El color blanco será todo el contrario, (255,255,255), los tres color al nivel máximo.

Los colores es el primero que veremos cuando nuestra vista se fije en un objeto o, en nuestro caso, en una aplicación informática o una página web. Esta primera visión o interrelación de los usuarios con las interfaces es muy importante, prácticamente crítica. Un usuario no dejará de usar una interfaz por razones de discrepancias con los colores, pero, quizás, lo obligará a hacer cambios, si los puede pedir.

Las paletas de colores nos darán la opción de escoger entre una variedad muy amplia de colores para todos los elementos de la interfaz, tanto para el fondo como para los iconos y las letras. Incluso se podría escoger usar una fotografía de fondo.



Figura 14. Paleta de colores.

A modo de ejemplo, en la figura 14 se muestran las opciones a la hora de escoger colores con el programa Paint. Esta paleta está basada en el modelo **RGB** (del inglés *red, green, blue*), modelo basado en la intensidad que daremos a los tres colores primarios. Esto nos ofrece la posibilidad de escoger entre más de dieciséis millones de colores.

Aspectos del color

- **Tonalidad:** tipo de longitud de ola de la radiación.
- **Brillo:** intensidad subjetiva del color.
- **Saturación:** pureza del color.

Por ejemplo, la luz blanca no tiene tonalidad, pero sí brillo.

Pero podemos ver que, además de poder escoger la intensidad de cada color primario, nos dan la opción de escoger la intensidad de tres aspectos del color, como son la tonalidad o matiz, la luminosidad o brillantez y la saturación.

Hay otros modelos de colores, como el modelo **CMYK** (del inglés *cyan*, *magenta*, *yellow* y *key/black*). Este modelo, que se usará más en la impresión en colores, está basado en la absorción de la luz. Otro modelo, pareciendo al CMYK, es el **RYB** (*red*, *yellow*, *blue*) utilizado de manera más común en las bellas artes.

Si hablamos de colores en el diseño, tendremos que tener en cuenta varios aspectos.

El más habitual encontrarse contraste entre los colores para poder entender mejor la información. Por ejemplo, usaremos texto con letras oscuras sobre un fondo más claro, o a la inversa. También habrá que seguir una **coherencia** en la elección de los colores, intentando relacionar un apartado de la aplicación o un conjunto de funcionalidades con un color y vincular este color con todo el que tenga que ver con las opciones relacionadas con el apartado (por ejemplo, la gestión de los usuarios o la gestión de los proveedores, etc.).

Otro tema importante a la hora de escoger los colores es el sistema operativo para el cual se desarrollará la aplicación. Tenemos que intentar adaptarnos a las características del sistema operativo, siguiendo su tipo de ventanas, iconos, colores, etc.

También hay que tener en cuenta la resolución de la pantalla en que se desarrollará la interfaz. Hasta hace poco, la medida habitual era la de 640×480 píxeles con 256 colores. Hoy en día, prácticamente todos los desarrolladores trabajan con resoluciones de 800×600 o superiores con color real o *true color*.

Hay una serie de normas para la utilización efectiva del color en las interfaces gráficas de usuario. En caso de seguirlas, el diseño cromático de la interfaz será correcto desde un punto de vista cromático. Es muy difícil dar un conjunto de normas cerradas, debido a la gran cantidad de variables que influyen en la representación del color y en su percepción posterior.

Algunas de estas normas son:

- Evitar la visualización simultánea de colores complementarios muy saturados.
- Evitar el color azul puro para el texto o para líneas muy finas: el sistema visual de los seres humanos no está preparado para recibir estímulos con una longitud de onda muy corta. El azul va bien para el fondo, pero no para el texto.
- Los usuarios de más edad necesitan colores con una luminosidad más alta para poderlos distinguir.
- Los colores se perciben diferentes si las condiciones de iluminación ambientales cambian: con la luz del sol, de bombilla incandescente o de fluorescente, los colores se perciben diferentes.
- La magnitud de un cambio que se detecte en un color cambia a lo largo del espectro: los pequeños cambios en rojos y en verdes son más difíciles de detectar que en los amarillos o en los azules y verdes.
- Hay que evitar el uso de una gama de colores basados en una sola tonalidad: los usuarios con problemas de visión los percibirán como iguales.
- La luminosidad no es igual al brillo: dos colores con la misma luminosidad pueden ser percibidos con diferente brillo dependiente de la tonalidad.
- Diferentes tonalidades tienen niveles diferentes de saturación: por ejemplo, el amarillo siempre parece que está menos saturado que el resto de colores.

- No todos los colores son igualmente legibles: estamos acostumbrados a leer en negro sobre blanco; obtendremos la máxima legibilidad con fondos claros y letras oscuras.
- Los colores se interfieren mutuamente: la percepción de un color cambia dependiendo de los colores que lo rodean.
- Hay que tener en cuenta la diferencia entre el color de pantalla y el color impreso: los dispositivos visualizadores y los de impresión no visualizan los colores del mismo modo (el uno utiliza RGB y el otro CMYK). Pensáis que un color que utilizáis en pantalla puede no tener nada que ver con el mismo color impreso.
- Se tienen que agrupar los elementos relacionados sobre un mismo color de fondo.
- Los colores similares expresan significados similares: se pueden mostrar relaciones entre elementos asignándolos tonalidades similares o saturaciones diferentes de la misma tonalidad.
- Los colores luminosos y saturados atraen la atención del usuario.
- Hay que ordenar los colores por su posición en el espectro: si es necesaria una utilización extensiva del color en una interfaz, se tiene que hacer de manera ordenada. La manera más natural es la del espectro, representada por el mnemotécnico inglés ROYGBIV (rojo, naranja, amarillo, verde, azul, añil y violeta).

Finalmente, también se tiene que tener en cuenta si algún usuario puede tener problemas visuales con ciertas combinaciones de colores. Esto es más fácil de saber si el proyecto se desarrolla para un determinado grupo de usuarios; si, en cambio, se desarrolla para usuarios no conocidos o cualquier persona puede ser usuario, entonces lo tendremos que tener en cuenta de una manera más genérica.

Fuentes

Las fuentes que se usarán en una interfaz también es otro elemento importante, tanto por el tipo como por la medida de la fuente.

Las fuentes son el formato que daremos a las letras que compondrán los mensajes de texto para interaccionar con el usuario.

Hay muchos tipos de fuentes y muchas posibilidades de usarlas, pero hay que tener cuidado con el hecho que estas fuentes estén incorporadas al sistema operativo o que no sean compatibles, cosa que puede hacer que no se puedan usar de manera adecuada.

Para poder escoger las fuentes tendremos que tener mucho en cuenta los sistemas operativos o los entornos de ejecución de las aplicaciones desarrolladas. Cómo se ha comentado en el apartado que hace referencia al color, habrá que adecuarse a las características y/o a las fuentes del sistema operativo y procurar estar en sintonía.

Además, como se ha hecho dicho más arriba, habrá que utilizar fuentes llamadas básicas porque sean aceptadas por cualquier entorno a trabajo o sistema operativo. En caso de usar una fuente no básica, habrá que empaquetarla con la aplicación desarrollada e instalarla con el software de instalación.

La **tipografía** es la apariencia que tiene el texto y es el elemento de diseño gráfico más básico de una interfaz. Algunas de sus características son las siguientes: la fuente, el cuerpo, el *serif*, el peso y la inclinación.

- **Fuente.** Es el tipo de letra. Por ejemplo: *Helvética*, *Times*, *Brush Script*, etc.
- **Cuerpo.** El cuerpo es el tamaño de la letra y se mide en puntos o picas (1 pica = 12 puntos).
- **Serif.** El serif es una herencia romana de la escritura con cincel encima piedra y consiste en una pequeña proyección al final de los palos de las letras. Las letras pueden ser con serif o sin.
- **Peso.** El peso hace referencia al grueso de la letra. Esta puede ser en negrita o normal.
- **Inclinación.** Según la inclinación, la letra puede ser vertical o cursiva (denominada también *itálica*).

La legibilidad del texto se ve afectada por diferentes factores:

- **Proporcionalidad.** Es una característica de la mayoría de fuentes que consiste a asignar un espacio horizontal a cada carácter de acuerdo con la anchura de este. Las fuentes que asignan a todos los caracteres el mismo espacio se denominan fuentes de anchura fija (en que un carácter como una y ocupa el mismo espacio que un carácter como una m).
- **Tamaño de la fuente.** Hay que recordar que el texto en una pantalla se lee un 25% más despacio que en papel. Un texto normal tiene como tamaño apropiado 10 puntos. Una de 12 puntos se lee todavía mejor. Usando diferentes tamaños se pueden conseguir jerarquías de texto (títulos, subtítulos, cabeceras, etc.).
- **Mayúsculas/minúsculas.** La mezcla de mayúsculas y minúsculas hace los textos más legibles. Esto es debido al hecho que cuando leemos reconocemos las palabras familiares por la forma y no las leemos carácter a carácter.
- **Espaciado e interlineado.** El espaciado, tanto de carácter como de palabra, depende de la fuente que utilizamos y es recomendable no tocarlo. La modificación del espaciado se puede justificar en situaciones determinadas como en títulos o etiquetas de botones, pero pocas veces en el texto normal. En cuanto al interlineado, hay que tener presente que cuanto más pequeño sea más difícil será la lectura porque costará encontrar el comienzo de las líneas.
- **Longitud de la línea.** Si la longitud de la línea de texto es demasiado larga, se hace difícil de encontrar el comienzo de la línea siguiente, y si es más larga del que podemos alcanzar con el movimiento del ojo nos obligará a mover la cabeza, cosa que nos provocará fatiga. Los textos impresos en un libro tienen que tener una longitud de unos sesenta caracteres, y los de una columna de diario, de unos treinta caracteres.
- **Justificación.** La justificación es la inserción de espacios extra entre las palabras para alinear las líneas tanto a la derecha como la izquierda. Esta técnica, aunque visualmente es más estética, retarda la lectura.
- **Maquetación.** La maquetación es la distribución de los varios párrafos de texto (y otros elementos gráficos) en un espacio determinado. La primera impresión que tenemos al abrir un libro, ejecutar un programa o visitar un web es debida a la maquetación. La maquetación puede hacer que nos quedamos en el web que estamos visitando o que vamos a otro de contenidos similares pero con una presentación visual más atractiva.

- **Márgenes.** Hay que evitar el “síndrome del procesador de textos”, que consiste a escribir de parte a parte de la página sin dejar márgenes ni a la derecha ni a la izquierda. Hay que dejar márgenes generosos de acuerdo con vuestra maquetación y con la longitud de la línea.
- **Distinción tipográfica.** Hay que utilizar los cambios tipográficos de cursiva, el cambio de fuente y negrita sólo cuando aporten realmente información y sin abusar. El uso excesivo de fuentes y de recursos tipográficos se denomina corrientemente *fontitis*.

Iconos

Los iconos que se usan en las interfaces también son importantes en la interacción con los usuarios. Ya hemos hablado de algunos iconos en el apartado de los menús, cuando hacíamos referencia al menú de barras de herramientas. Hablaremos del mismo tipo de iconos con las mismas funcionalidades, pero en este apartado nos fijaremos más en los iconos de acción de las interfaces de usuario.

*Los iconos ofrecen la representación de una unidad de significado.
Si esta unidad de significado se quisiera expresar con texto,
ocuparía mucho más espacio del que ocuparemos usando un icono.*

Estas unidades de significado pueden ser ideas, conceptos, acciones u otras. Por ejemplo, podemos tener el icono que representa con un dibujo de un disquete la acción de grabar un trabajo hecho o un documento.

De este modo podremos representar más acciones con un espacio más limitado, además de ofrecer una usabilidad mejor, más directa, más rápida e intuitiva, siempre que los iconos que se usen sean conocidas.

Cuando un usuario usa de manera repetitiva una aplicación, los iconos y su significado se memorizan fácilmente. En cambio, el uso de una aplicación por parte de muchos usuarios diferentes o de un mismo usuario de manera no muy continuada implicará la necesidad de utilizar iconos muy fácilmente reconocibles para los usuarios, que sean parecidos a las que se usan de manera habitual en otros sistemas operativos u otras aplicaciones muy utilizadas.

Hay interfaces en que el aspecto y el impacto visual es muy importante, prácticamente son unos de los objetivos de la interfaz. Hace falta que la interfaz llame la atención del usuario de manera muy impactante. En estos casos, también es muy adecuado usar iconos. El usuario conocerá el funcionamiento de la interfaz a medida que descubra los iconos y su significado.

A la hora de diseñar interfaces con iconos, también tenemos que tener en cuenta las limitaciones que pueden presentar.

A veces, la misión de los iconos es difícil al intentar representar una acción o idea difícilmente representable. Si hay acciones similares es complicado encontrar iconos que diferencien con exactitud los matices que harán que las acciones sean diferentes. Además, la interpretación por parte de los usuarios será muy personal y puede dar pie a

cometer errores. Es por esta razón que se recomienda no identificar con iconos acciones críticas de una aplicación.

A la hora de diseñar o escoger iconos, tenemos que tener en cuenta que hay de diferentes tipos. Las que se basan en la analogía ofrecen imágenes que intentan parecer el que quieren representar. Las que se basan en las muestras simbolizan elementos que intervienen en el que se quiere referenciar. Las que se basan en los símbolos son representaciones más abstractas. Los iconos arbitrarios reclaman que el usuario aprenda el significado.

Hay que hacer un uso razonable de los iconos, sin utilizar el recurso más del recomendable. La velocidad de reconocimiento de los iconos acaba siendo pareciendo a la del texto, pero nunca es superior. De hecho, se producen más errores con el uso de los iconos que con el uso de los menús de texto. Estos errores pueden influir en el aprendizaje de la aplicación por parte del usuario o, incluso, en su rechazo o su falta de satisfacción.

Distribución de los elementos

Llegados a este punto, se ha hablado de diferentes conceptos importantes en el aspecto de una interfaz de usuario, como pueden ser los colores, las fuentes y los iconos. En el apartado anterior se han visto diferentes elementos de la estructura de las interfaces como los menús, las ventanas, los cuadros de diálogo y los atajos de teclado. Ahora toca hablar de cómo se distribuyen estos elementos por la interfaz.

La distribución de los elementos tiene que permitir un uso de las interfaces y un aprendizaje óptimo de estas a los usuarios. Hay que decidir la manera de presentar los menús de texto, las barras de herramientas, los iconos, los cuadros de diálogo, etc. y su ubicación en la interfaz. Los elementos tienen que presentar una distribución uniforme, teniendo en cuenta la agrupación de funcionalidades relacionadas y las zonas de la pantalla que los usuarios visualizarán más.

Los objetivos de la distribución de los elementos por la interfaz son los siguientes: conseguir que la interfaz sea fácil de usar, fácil de aprender, segura, fiable y efectiva a la hora de llevar a cabo las acciones necesarias para una aplicación. Además tiene que ser consistente.

Al igual que se ha ido comentando a lo largo de este apartado, es interesante adaptar la interfaz desarrollada en el entorno de trabajo de esta, distribuyendo los elementos de manera análoga a otras aplicaciones o interfaces del sistema operativo.

Todo aquello que el diseñador pone en el diseño de una interfaz tiene que estar justificado, tiene que tener una consistencia y tiene que seguir una organización espacial. Las interfaces de un navegador de Internet, por ejemplo, contienen unos ejemplos que están dispuestos de una manera determinada en el espacio que será familiar para los usuarios. Esta distribución obedece a unas normas de diseño de GUI, y ofrece una coherencia externa con otros programas similares que son conocidos.

Algunos autores, como el artista y diseñador de la Bauhaus, Josef Albers, han hablado así del diseño:

Diseñar es planificar y organizar, ordenar, relacionar y controlar. Resumiendo, comprende todo el que sea ir en contra del desorden y la casualidad. Por lo tanto, muestra una necesidad humana y describe el pensamiento y la actuación del ser humano.

Otro concepto que hay que tener en cuenta es el **número mágico de Miller**:

Todos los componentes gráficos de una interfaces de una aplicación informática (menús, áreas de datos, iconos, diálogos, tableros o imágenes) están dispuestos encima de una retícula imaginaria que es respetada a lo largo de la aplicación (los menús no cambian de lugar ni tampoco lo hace el área de visualización de datos, por ejemplo).

El tamaño, distribución y número de divisiones de la retícula es crucial para la usabilidad de la interfaz. Estos parámetros están determinados por el tipo de información que hay que visualizar, por la resolución de la pantalla y por las condiciones de visualización (no es el mismo un cajero automático que un teléfono móvil).

En general, el número de divisiones de la retícula sigue el recurrente número mágico de Miller de 7 ± 2 , que es el número máximo de elementos que una persona puede retener en la memoria de corto plazo.

El mismo esquema se puede aplicar recursivamente para cualquier elemento, tanto si es un tablero de control como una barra de menús, una barra de iconos o un diálogo.

Otro aspecto que hay que tener en cuenta en la organización espacial de nuestra información son las relaciones entre los diferentes elementos de información. Los elementos de información relacionados por el contenido también tienen que estar visualmente relacionados.

2.2.3. Diseño de los elementos interactivos de las interfaces de usuario

Hay otros muchos elementos que se usan en las interfaces de usuario, además de los que hemos comentado hasta ahora. Tanto los menús como las ventanas, los cuadros de diálogo o los iconos son elementos que facilitan la interacción entre el usuario y la interfaz.

Elementos de una GUI

Entre los elementos de una interfaz de usuario podemos encontrar menús, ventanas, texto (tipografía digital), iconos, dispositivos de control y otros elementos de entrada y/o de salida de datos como botones de órdenes, como los radio buttons, las casillas de selección (check box), los scrolls, o las listas de distribución.

Estos elementos permiten la navegación por la aplicación, muestran las alternativas que tiene el usuario y le ofrecen la posibilidad de ejecutar algunas acciones. Son siempre,

pero, informaciones estáticas. Otros elementos que ofrecerán otro tipo de información al usuario, una información más dinámica, por ejemplo, con el resultado de los datos obtenidos a partir de una consulta u ofreciendo la posibilidad de añadir información a la ya almacenada. Entre estos otros elementos encontramos los botones de órdenes, los *radio buttons*, las casillas de selección o las listas desplegables.

Botones de órdenes

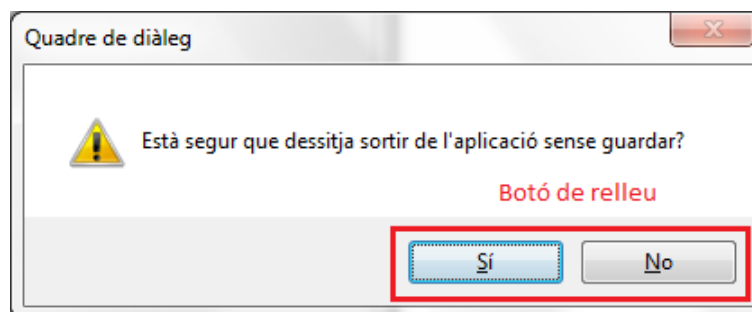
Los botones de órdenes se consideran dispositivos de control. De este modo un botón es un objeto de control de la interfaz de usuario. Estos botones permiten introducir datos de confirmación a la aplicación, pero también permiten seleccionar datos o introducir de nuevas en el sistema.

Los botones nos sirven para construir las interfaces y actúan como interpretaciones visuales de las funcionalidades que representan. Los botones son parte, como los iconos, de la gramática visual de las interfaces de usuario.

Podemos encontrar varios tipos de botones, como los botones de opción (*radio buttons*), los botones de confirmación (*check box*), los botones de acción, las pestañas, las barras de arrastre (*sliders*), barras de desplazamiento o *scroll*, etc.

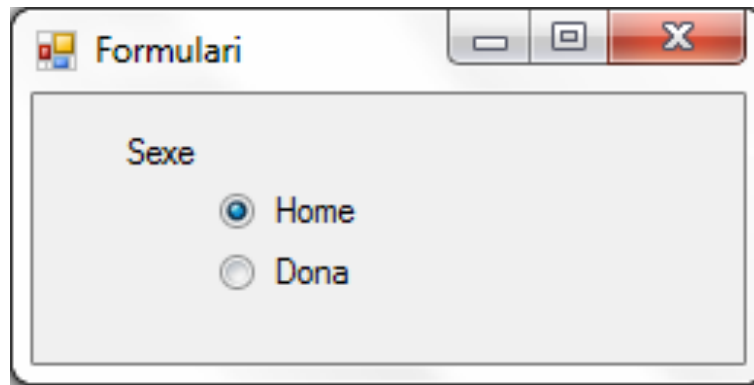
Botones de acción

Son los botones más habituales, los que más se usan en todo tipo de sistemas operativos. Estos botones buscan la simulación de volumen. Un botón de acción contiene un texto o una imagen que identifica una funcionalidad de la aplicación. Cuando un usuario elige un botón de acción y clicla encima, el botón simula un hundimiento y lleva a cabo la acción relacionada con él. Podéis ver un ejemplo de este tipo de botones a la figura 15.



Radio buttons

Los botones en forma de radio o botones de opción ofrecen al usuario una lista de opciones que tienen a la izquierda o la derecha (menos habitual) una pequeña redonda para poder cliclar adentro. Si hemos seleccionado una de las opciones, esta redonda pequeña se llenará con un punto y nos indicará que esta es la opción escogida. Hay que tener en cuenta que, con la opción de los botones de opción, sólo podemos seleccionar una de las opciones ofrecidas.

A screenshot of a Windows application window titled "Formulari". Inside the window, there is a label "Sexe" followed by two radio buttons. The first radio button is selected and is labeled "Home". The second radio button is not selected and is labeled "Dona".

Como ejemplo de utilización tendríamos las encuestas, en que detrás de una pregunta se ofrecen cuatro o cinco respuestas, de las cuales el usuario sólo puede seleccionar una. Al final hay un botón de acción sobre el cual se tiene que clicar para indicar que esta es nuestra selección correcta. En la figura 16 podéis ver un ejemplo donde se hace escoger el género a un formulario.

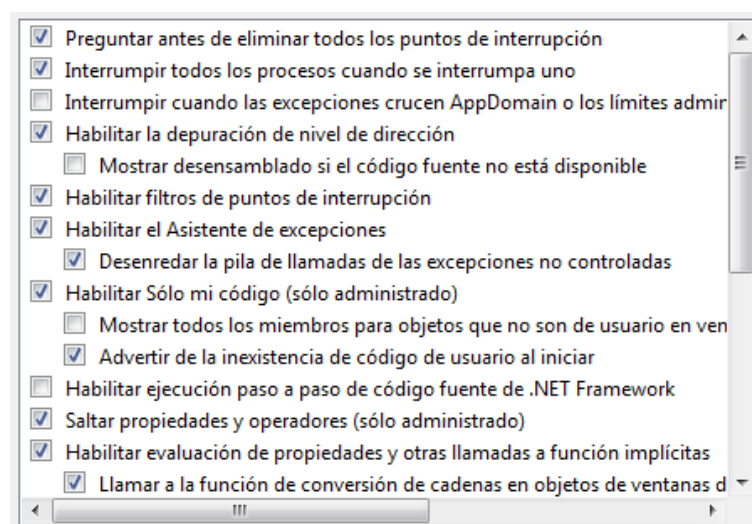
Check box

Estos botones son muy parecidos a los anteriores, pero con la particularidad que, a la izquierda o la derecha del texto indicativo de las opciones, están representados con pequeños recuadros. En caso de seleccionar la opción correspondiente a un recuadro, este se llenará con un símbolo de *check*.

La gran diferencia entre los botones de acción y los botones de *check box* es que los *radio buttons* sólo podemos seleccionar una opción. En cambio, con los *check box* podemos seleccionar tantas opciones como queramos.

Un ejemplo podría ser un formulario de registro en que un usuario está introduciendo sus datos y se preguntado por sus aficiones. tendrá que seleccionar tantas cómo quiera dentro de una lista de quince o veinte opciones.

Podéis ver otro ejemplo de aplicación de los *Check Box* en la figura.17.

A screenshot of a Windows application window showing a list of options with checkboxes. The options are: "Preguntar antes de eliminar todos los puntos de interrupción" (checked), "Interrumpir todos los procesos cuando se interrumpa uno" (checked), "Interrumpir cuando las excepciones crucen AppDomain o los límites admir" (unchecked), "Habilitar la depuración de nivel de dirección" (checked), "Mostrar desensamblado si el código fuente no está disponible" (unchecked), "Habilitar filtros de puntos de interrupción" (checked), "Habilitar el Asistente de excepciones" (checked), "Desenredar la pila de llamadas de las excepciones no controladas" (checked), "Habilitar Sólo mi código (sólo administrado)" (checked), "Mostrar todos los miembros para objetos que no son de usuario en ven" (unchecked), "Advertir de la inexistencia de código de usuario al iniciar" (checked), "Habilitar ejecución paso a paso de código fuente de .NET Framework" (unchecked), "Saltar propiedades y operadores (sólo administrado)" (checked), "Habilitar evaluación de propiedades y otras llamadas a función implícitas" (checked), and "Llamar a la función de conversión de cadenas en objetos de ventanas d" (checked). The window has a scrollbar on the right side.

Listas desplegables

Otros elementos interactivos de las interfaces de usuario son las listas desplegables, conocidas en inglés como *combo box*.

Alternativas

Drop-down list: es un menú desplegable en que podremos escoger una opción dentro de una lista de posibilidades, sin poder escribir.

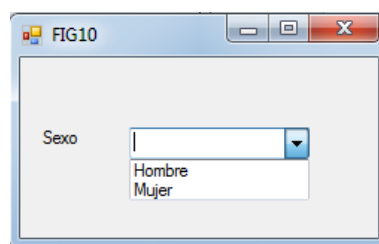
List box: nos ofrece una lista de opciones con todas ellas a la vista, sin ningún desplegable.

Para entender los *combo box* hay que hacer una revisión otros elementos parecidos, pero con matices que los hacen diferentes. Hay los menús desplegables (*drop-down list*), que nos servían tanto para mostrar un conjunto de informaciones como para usar su funcionalidad de selector de opciones.

En cambio, el *combo box* nos permite escribir en un cuadro de texto, a la vez que podemos ver una lista de opciones. Se trata del mismo sistema que usa el buscador Google: cuando empezamos a escribir la palabra o frase que queremos buscar, nos va mostrando propuestas de palabras de busca, basadas en otras búsquedas de otros usuarios o en el historial de búsquedas que nosotros habremos hecho.

Una lista desplegable es una combinación de un cuadro de texto (text box) y un menú desplegable (listbox).

Si quisiéramos ofrecer una lista de opciones sin permitir la opción de agregar un nuevo valor, entonces estaríamos obligados a usar el *list box*, puesto que el *combo box* no tendrá ninguna opción que permita esta configuración. En cambio, sí que podremos añadir a la lista de opciones desplegables nuevos ítems. Con las propiedades del elemento podemos agregar nuevos ítems sin tener que actualizar la interfaz. podéis ver un ejemplo en la figura 18.



2.3. Otras pautas de diseño

El diseño de las interfaces de usuario tiene que tener en cuenta muchos elementos y muchas reglas y detalles para llegar a ofrecer una usabilidad aceptable.

Ahora hay que hablar de los elementos y reglas vinculados con la gestión de los datos, tanto del que tiene que ver con su presentación, como del relacionado con su posible manipulación o destrucción. Es decir, la vinculación de los datos a una base de datos con la interfaz con que trabajará el usuario.

Otros elementos a tener en cuenta son la secuencia de control de la aplicación, la seguridad de la información y otras pautas específicas para el diseño de interfaces con elementos multimedia.

Antes hay que revisar algunas de las reglas no escritas que tenemos que tener en cuenta en el diseño de interfaces de usuario:

- El usuario de la interfaz tiene que tener el control de esta (y, además, sentir que tiene este control).
- Se tienen que usar mensajes y textos descriptivos, fáciles de entender para los usuarios.
- Los mensajes que se darán al usuario tienen que ser positivos y activos sin parecer insultantes o graciosos.
- Las acciones y tareas vinculadas se tienen que poder interrumpir y reiniciar en otro momento.
- Los usuarios tienen que poder utilizar el teclado y el ratón para comunicarse con la interfaz.
- Conocer la cultura del país puede ayudar a evitar malentendidos en los mensajes al usuario.
- La interfaz tiene que poder ser personalizada y manipulada.
- Tiene que facilitar la navegación por la interfaz, encontrar las funcionalidades y la salida o cierre de esta.
- Tiene que permitir diferentes niveles de uso de la interfaz para diferentes niveles de usuarios (con más o menos experiencia).
- Tiene que intentar facilitar al máximo el trabajo a los usuarios, por ejemplo, mostrando listas en las cuales se pueda escoger una opción antes de tener que teclear el nombre de esta opción.
- Tiene que asociar acciones a los diferentes objetos presentes en las interfaces.
- Tiene que hacer la interfaz transparente para los usuarios.

2.3.1. Presentación de los datos.

Los datos que se mostrarán usando las interfaces de una aplicación tienen que seguir unas pautas concretas para no confundir los usuarios. Según qué información se quiere presentar se tiene que tener acceso a un apoyo externo que contenga los datos, o bien se tendrán los datos incorporados en nuestra interfaz.

Fijaremos en caso de que los datos estén ubicados en un apoyo externo a la aplicación, normalmente en una base de datos o en un archivo de texto o binario. Quizás queremos usar datos que se encuentran en otras aplicaciones o son resultados de estas, pero para hacer esto trataremos la información con ficheros intermedios hasta hacerla llegar a nuestro software.

La aplicación desarrollada (y sus interfaces) servirá para comunicar el usuario con estos datos, pero esto se tiene que hacer de manera transparente para el usuario.

A la hora de diseñar la presentación de los datos, tenemos que tener en cuenta algunas cuestiones:

- **¿Qué tipo de información quiere ver el usuario?** ¿Quiere tener acceso a la información de gestión o quiere tener acceso a información estadística o de

resumen para poder tomar decisiones? Para poner un ejemplo, será diferente la manera de presentar los datos a un usuario que tiene que dar de alta nuevos proveedores, tiene que poder modificar sus datos y tiene que poder crear pedidos a estos proveedores, que si el usuario está interesado a tener un resumen anual de las ventas por referencia, por mes y por tienda de los productos que compra a un proveedor.

- **¿Qué uso en querrá hacer?** ¿Querrá sólo consultar, o tendrá que poder modificar, borrar y crear nueva información? También se tratará diferente la información a que se podrá dar acceso para gestionar y para manipular que los datos que serán sólo de consulta.
- **¿Si los datos se modifican de manera continua, el usuario tiene que ver reflejados estos cambios en el mismo momento?**
- **¿La información a mostrar será numérica o en modo texto?**

Todas estas cuestiones nos ayudarán a decidirnos por una manera de presentar los datos o por otra.

Un buen conocimiento de los usuarios también nos ayudará a ofrecer una presentación adecuada de los datos con las cuales tendrá que trabajar.

2.3.2. Secuencia de control de la aplicación

Toda aplicación con una interfaz de usuario ofrece muchos elementos y muchas funcionalidades a los usuarios. Estos elementos nos traerán a poder llevar a cabo muchas operaciones, la mayoría de las cuales trabajarán con datos. Algunas de estas operaciones serán directas y se ejecutarán de manera inmediata. A otras operaciones, podremos llegar desde diferentes ubicaciones de la interfaz. Otras operaciones serán compuestas y comportarán más de un acceso o una modificación a la base de datos.

Tenemos que tener en cuenta que ciertas operaciones se podrán interrumpir por un mal uso de la interfaz o de la aplicación o por razones ajenas al software desarrollado. Otras operaciones tendrán que ser confirmadas por el usuario para validar la ejecución correcta.

La secuencia de control de la aplicación se refiere a las acciones que el usuario llevará a cabo con el software y la lógica de ejecución de este. Es decir, si el usuario pide ejecutar una operación que comporta varias acciones con la base de datos, tendremos que establecer un control de la secuencia de operaciones que se llevarán a cabo para garantizar que la información se ha mantenido consistente en la base de datos.

Un buen diseño de una interfaces (y de una aplicación en general) tiene que incluir unos objetivos que incluyan acciones de control de consistencia, una necesidad mínima de acciones de control y flexibilidad en los controles de secuencia para adaptarse a las diferentes necesidades de los usuarios.

Uno de los puntos más determinantes en la satisfacción y en la aceptación por parte del usuario de una aplicación informática a la hora de utilizar una interfaz es la sensación de control que tendrá en una sesión de uso de la aplicación con varias acciones interactivas.

Si no puede controlar la secuencia de interacción, se sentirá sin el control de la aplicación, insatisfecho, cosa que puede provocar el rechazo a la aplicación entera.

2.3.3. Seguridad de la información

La información con que trabajaremos estará en una base de datos. La introducción, modificación o borrado de los datos se hará por medio de una interfaz. Una vez esta se ha usado y los datos están estables en la base de datos, esta tiene que estar disponible para cuando los usuarios la necesiten y tiene que estar asegurada.

Para conseguir esto hay que establecer un proceso de mantenimiento y una utilización que no comprometa la integridad. Además, se pueden usar otros sistemas para asegurar la disponibilidad de la información, lo cual minimizaría el riesgo de quiebra de los sistemas informáticos. Estos pueden ser sistemas de reserva o de emergencia que hagan copias de seguridad de manera automática y regular.

Otro método para asegurar la integridad de la información consiste a establecer sistemas para evitar el acceso directo en las bases de datos por parte de usuarios no controlados, en caso de trabajar con una información crítica para las organizaciones. Estos sistemas pueden incluir un sistema de permisos para perfiles de usuarios o unas contraseñas. Con este sistema, sólo un número determinado de usuarios tendrá permiso para modificar los datos.

Además, también hay que minimizar los errores humanos que se pueden producir al gestionar esta información los usuarios, manipulando la aplicación o errores al introducir los datos.

Para evitar esto, se puede pedir la confirmación de los datos al introducir información y comprobar que esta cumple las características establecidas por los requisitos, como por ejemplo el tipo de datos que se pueden introducir en un *text box* (sólo números o sólo letras o con un formato determinado).

Otra manera de asegurar la información es hacer revisar al usuario los datos que ha introducido o con los cuales ha tratado, usando un sistema de pregunta-respuesta mediante cuadros de diálogo esenciales o, también, algún cuadro de diálogo de tipo alerta. Si un usuario tiene que introducir datos mediante una interfaz, antes de grabar la información en la base de datos se podrá mostrar un cuadro de diálogo con los datos al usuario para pedir su confirmación.

2.3.4. Específicas para aplicaciones multimedia

En función del tipo de interfaces que se desarrollen o del tipo de aplicación o el entorno para el cual se plantee el uso de la aplicación, nos podemos encontrar con la necesidad de usar algunos elementos multimedia específicos como por ejemplo la animación o el sonido.

Si hablamos de interfaces desarrolladas para dispositivos móviles o para entornos web, podremos usar algunos elementos como el sonido en determinadas acciones o determinados elementos creados con Flash que aporten algunas animaciones, bien por necesidad o bien para dotar las interfaces de un diseño más agradable.

Otros ejemplos de utilización son la utilización de alertas sonoras para determinados errores en la interacción con la interfaz, o la utilización de animaciones en la ayuda al usuario, animaciones mucho más explícitas que ciertas indicaciones en formato texto.

Otro caso es el que nos podemos encontrar en aplicaciones para el control de la gestión de una empresa, en que se busque dotar de estadísticas, gráficas y muchos números los directivos porque puedan tomar decisiones. Estas informaciones, en vez de mostrarlas con unos gráficos normales en dos dimensiones, se pueden desarrollar usando herramientas multimedia o animaciones que muestren de manera más entendedora los datos que se quieren consultar.

Para resumir, podríamos remarcar uno de los objetivos de la usabilidad y el diseño de interfaces: “Una interfaz de usuario estará muy diseñada cuando el programa se comporte exactamente como el usuario piensa que lo tendría que hacer.”

Referencias

1. *Enciclopedia Encarta 99, Interfaz de Usuario*
2. *Enciclopedia del Estudiante, LARPRESS 99, Interfaz Hombre-Máquina.*
3. *Instituto Técnico Superior México, curso de interfaz de Usuario*
<http://www.bayesinf.com/spanish/product/forphone/help/4inteelem/contens.htm>
4. *Universidad Autónoma de Guadalajara, Tutorial "Diseño de una Interfaz Gráfica"*
5. *SIGGRAPH de México*
6. *LINEBACK, Graphical User Interface Timeline*
7. *COMUNICACIÓN HOMBRE MÁQUINA – Universidad de sevilla*
8. *Interfaz de usuario - Monografias.com*
9. *Desenvolupament d'interfícies, Miguel Ángel Carpintero Rodríguez, IOC, 2010.*