

## DESARROLLO DE INTERFACES

UD3. USABILIDAD

**2CFGS**

**DAM**

# USABILIDAD



**Pau Gallego Garcia**

CFGS DISEÑO APLICACIONES MULTIPLATAFORMA

Módulo: 0488 – Desarrollo de Interfaces

UD3. Usabilidad

# USABILIDAD

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

## 1. INTRODUCCIÓN A LA USABILIDAD

Existen 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 elegir si nos gustan o no, o si se pudieran utilizar otras. Por ejemplo, con el cuadro de interacción de una radio de un coche, se tendrá poco margen de maniobra. Si gusta, bueno. Si es poco amigable, o difícil de entender, deberemos adaptarnos, o bien cambiar de aparato de radio en el coche.

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

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

Para conseguirlo habrá que considerar este aspecto al desarrollar software de todo tipo, estableciendo así en la fase de análisis los requerimientos de usabilidad, objetivos que deberán alcanzarse para finalizar el desarrollo de la aplicación. Estos requerimientos de usabilidad guiarán los diseños de las interfaces que van a interactuar 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 muy en cuenta, puesto que la usabilidad de las aplicaciones que se presentarán a los futuros usuarios será lo primero que verán en los prototipos de las aplicaciones o de las interfaces y será lo primero que evaluarán. También es necesario indicar qué será lo que podrá diferenciar una aplicación de otra, ya que todas las aplicaciones cumplirán las reglas de negocio.

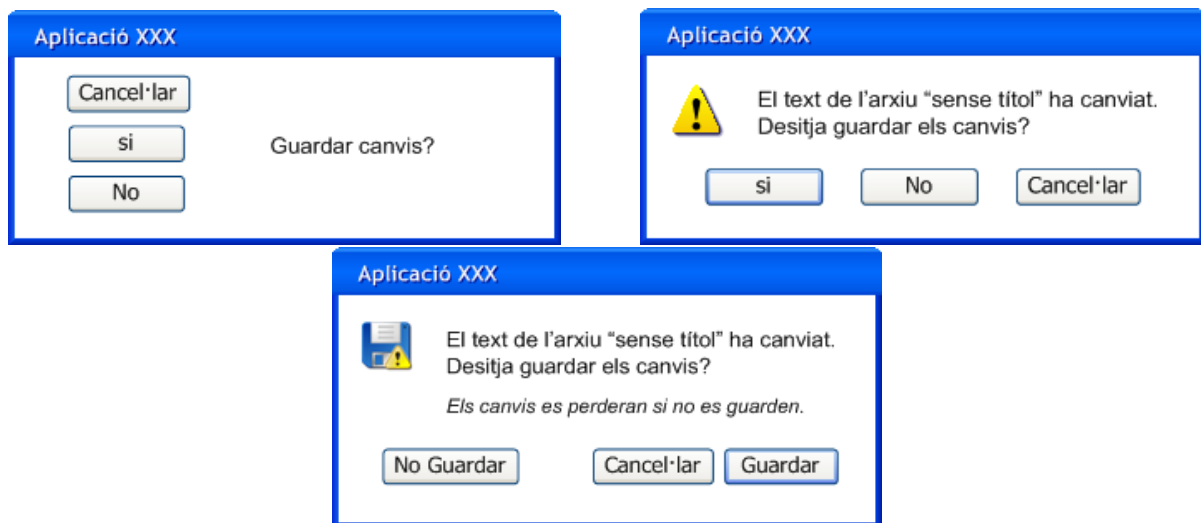
### 1.1 ¿Qué es la usabilidad?

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

Pero hay muchas formas de entender este concepto, ya que deben tenerse en cuenta muchos matices para poder definir o entender el significado de facilidad de uso. Además, también será necesario hablar de otros conceptos para definir la usabilidad, términos como efectividad, eficiencia y satisfacción del usuario. Estos conceptos variarán la apreciación según si los afectan los usuarios, sus objetivos o expectativas y el 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 vas a desarrollar (aunque estas definiciones también servirán para aplicaciones o portales web).

A continuación, veremos tres ejemplos de interfaces en los que se espera una interacción del usuario con una aplicación determinada, cómo podría ser, por ejemplo, un editor de textos. Observaremos tres ejemplos distintos, donde algunos de ellos no son correctos.



En este último caso encontramos algunos cambios más significativos. En primero lugar, la imagen que se muestra es más indicativa del tipo de acción que se requerirá, contiene un símbolo de alerta, pero la imagen del disquete indica que se trata de una posible operación de carencia de grabación antes de salir de la aplicación. Además, encontramos que los botones de respuesta del 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 que las palabras de los botones de respuesta son verbos, que indican con claridad las opciones al seleccionar una.

Para establecer una definición más adaptada a nuestro ámbito de estudio, nos fijaremos en las definiciones que podemos encontrar en las normas ISO relacionadas con la usabilidad.

## NORMAS ISO

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” en 1998.

La usabilidad es la parte que trata del grado con el que un producto puede ser usado por usuarios específicos para alcanzar 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 para que, en el desarrollo del software, el producto final desempeñe 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 la que nosotros podremos utilizar una herramienta concreta desarrollada para conseguir un objetivo. Pero habrá que añadir otros términos como claridad, elegancia con la que se ha diseñado la aplicación o sitio web.

Este término también podrá utilizarse 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 su utilidad, facilidad de uso, facilidad para aprender su uso o apreciación por parte de un usuario determinado en un contexto determinado.

## 1.2 DIMENSIONES DE LA USABILIDAD: CARACTERÍSTICAS Y ATRIBUTOS

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

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

### EFICIENCIA

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

Puede entenderse la eficiencia en el uso de las aplicaciones informáticas. Es decir, una aplicación que obtiene unos buenos resultados en las medidas de usabilidad aportará mayor

productividad al usuario y permitirá un uso más rápido, más eficiente de la aplicación. Se tratará de poder realizar 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 utilizados serán las horas para desarrollar las aplicaciones o 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 al principio del desarrollo de las aplicaciones, es decir, en que esta interfaz cumpla los requerimientos.

## EFICACIA/EFFECTIVIDAD

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, ya que será el que nos indique si las interfaces alcanzan todos los objetivos establecidos al comienzo del desarrollo de una aplicación, siendo una interfaz que garantice la efectividad. Además, también será necesario que las funcionalidades establecidas en los menús o iconos de las interfaces hagan lo que deben hacer.

## SATISFACCIÓN

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

¿Puede un usuario de una aplicación informática obtener satisfacción por el uso de ésta? Puede entenderse la eficiencia en el uso de las aplicaciones informáticas. Es decir, una aplicación que obtiene unos buenos resultados en las medidas de usabilidad aportará mayor productividad al usuario y permitirá un uso más rápido, más eficiente de la aplicación. Se tratará de poder realizar más acciones, más interacciones con el software en menos tiempo. Quizás depende de la persona, no podemos evaluarlo con objetividad, pero lo que sí queda claro es que, si conseguimos desarrollar una aplicación con la que un usuario tiene una actitud positiva cuando la utiliza (aunque sólo no se sienta en ella) incómodo), habremos alcanzado el objetivo de satisfacción. La satisfacción es el cumplimiento de los objetivos planteados inicialmente para obtener unos resultados con cierto grado de gozo para el usuario.

Una vez definidas estas tres características, hay que recordar el concepto de usabilidad: parte que trata del grado con el que usuarios específicos usan un producto 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 es necesario establecer una serie de guías y normas para llevar a cabo un diseño esmerado en las necesidades de los usuarios, con un grado de la usabilidad más que correcto. Sin embargo, antes añadiremos algunas características de usabilidad a la definición que hemos hecho más arriba ya sus atributos. Este término podría entenderse como producir una actitud positiva en los usuarios.

## ATRACTIVO

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

Una interfaz es atractiva para un usuario cuando éste acepta con agrado sus características y su uso, mostrando una predisposición para utilizarla.

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

### **FACILIDAD DE APRENDIZAJE**

Las interfaces deben ser suficientemente inteligibles y amigables para permitir el aprendizaje intuitivo por parte del usuario.

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

Esta característica está asociada con el atributo de eficacia. Una aplicación eficaz debe poder aprenderse y utilizarse de manera fácil y rápida. Si una interfaz requiere un uso continuo de la documentación de usuario, su eficiencia nunca podrá ser buena.

### **FACILIDAD DEL SISTEMA PARA SER RECORDADO**

Una aplicación que debe utilizarse de forma esporádica, o de forma regular, pero, por ejemplo, una vez al mes, debe cumplir la siguiente característica: debe recordarse fácilmente cómo se utiliza.

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 utilizar esa 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 deben rellenarse en algún formulario. Hay aplicaciones que no permiten el uso de una opción y no explican por qué, o que permiten que se utilice, permiten que la aplicación falle.

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

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

## **1.3 NORMAS ISO RELACIONADAS CON LA USABILIDAD**

Existen 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, para todas las industrias salvo la electricidad y la electrónica.

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

Por lo que respecta 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 los siguientes:

- ISO 9241. Aporta requerimientos y recomendaciones relacionados con las características del software y el hardware, así como del entorno que debe 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 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, eficiencia, facilidad de mantenimiento y portabilidad.
- ISO 13407. Esta norma explica las actividades requeridas para el diseño 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 los jefes de proyectos informáticos o para los responsables del diseño de interfaces de éste.

#### 1.4 MEDIDAS DE USABILIDAD DE APLICACIONES: Tipos de métricas

¿Cómo se diseña una interfaz de usuario que cumpla los requisitos de usabilidad? ¿Cómo saber si una interfaz cumple los requisitos de usabilidad que nos hemos planteado? Volvemos a plantearnos la pregunta: una vez nos encontramos ante una interfaz, ¿cómo podemos evaluar su usabilidad?

Es una pregunta difícil de responder. Hay muchos parámetros que pueden valorarse. Muchos son parámetros subjetivos, difíciles de medir. Quizás para un usuario una interfaz es intuitiva, es satisfactoria, pero para otro usuario no lo es. Por eso debemos buscar métricas objetivas.

Además, debemos plantearnos otras preguntas, como: ¿en qué parte del desarrollo de aplicaciones informáticas deberemos realizar las medidas de usabilidad?, ¿en la fase de finalización y transferencia?, ¿en el desarrollo?, ¿o hay que hacer pruebas en el análisis y durante el diseño? Este punto es también importante. Según el momento en que apliquemos las medidas tendremos resultados u otros y, además, seremos a tiempo de superar los errores que se hayan hecho, o, si lo hemos medido demasiado tarde, quizás no se puede volver atrás sin provocar un gran desbarajuste en la planificación.

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

Para obtener **datos cualitativos es necesario 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, será necesario 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 elegir una opción.**
- **Tiempo o número de 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, existen muchas técnicas que utilizan las métricas expuestas más arriba.

**Según Rubin (1994), existen 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. Existen cuatro tipos de test:

- **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 al principio 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 adelantarse a las posibles deficiencias del producto.
- **Test de comparación.** Este test implica en **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 utilizar y aprender.** Este test podrá aplicarse en paralelo con otros tipos de test.

## **1.5 PRUEBAS DE EXPERTOS Y CON USUARIOS**

Estas pruebas estarán **relacionadas con sus características y dimensiones, así como la medida de la usabilidad y los tipos de métricas existente.**



En primer lugar, habrá que tener muy claro que lo 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 realizar 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 tendrán que justificarse 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 echar una mano en algunas decisiones.

Para llevar a cabo las pruebas será necesario desarrollar previamente un plan de pruebas. Este plan deberá prever:

- El alcance de las pruebas (qué probaremos y que no, hasta dónde llegaremos).
- Los propósitos de éstas (cuáles son los objetivos, las razones o las justificaciones).
- 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 tiempo tendrá que durar, en cuántas sesiones diferentes, etc.).
- Los participantes en las pruebas (cuántos usuarios, si los usuarios serán stake holders o no tendrán ninguna vinculación con el desarrollo del proyecto, cuántos expertos, si estarán vinculados con el proyecto o serán externos).

## 2. PAUTAS DE DISEÑO DE LAS INTERFACES DE USUARIO

A la hora de desarrollar un proyecto de creación de un software, será necesario 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 mayor o menor medida, que tendrán un efecto decisivo en el éxito del proyecto.

*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 la estructura, el aspecto y los elementos de las interfaces de usuario.*

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

→ **Un usuario experto.** Dentro de los usuarios expertos todavía podemos realizar 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 las 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 utilizar 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 mayor 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 demasiados problemas.

- **Los usuarios intermedios** son usuarios que se encuentran entre los expertos y los novatos. En este caso, no hablaremos de conocimiento de negocio. Pueden tener un nivel medio de conocimiento o uso de las TIC. Esto hará que no necesitemos ofrecerles tantas facilidades ni información como a los usuarios noveles, pero sí que habrá que facilitarles 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 juego del negocio al 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 su utilidad (llamadas tooltip o indicador de función)

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

- *Conocer al usuario.*
- *Minimizar la memorización.*
- *Optimizar las operaciones.*
- *Permitir los errores.*

La mayor parte de la interacción con un ordenador se realiza de forma 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 sólo hará que la interfaz sea más o menos atractiva, sino que la harán más o menos usable y determinarán su 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 blog. Los sistemas de ventanas actuales aplican este principio de forma recursiva.
- **El principio de visibilidad y utilidad.** Los controles usados frecuentemente deben ser visibles y fácilmente accesibles.
- **El principio de consistencia inteligente.** Utilizar una distribución de la información similar para funciones similares, para habituar a los usuarios a encontrar la información en los mismos sitios en situaciones similares.
- **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 principio de reducción del desorden.** Es un principio que resume el resto de los principios vistos anteriormente. 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 lo suficientemente

atractiva y lo suficientemente funcional en la que el desorden y la arbitrariedad se habrán reducido al mínimo.

## 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 ésta. Este conjunto de elementos y su ubicación formarán la estructura de la interfaz.

### MENÚS

El primer elemento que encontraremos en una interfaz será el menú o 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.

*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 escoger la funcionalidad que desea utilizar en un momento determinado.*

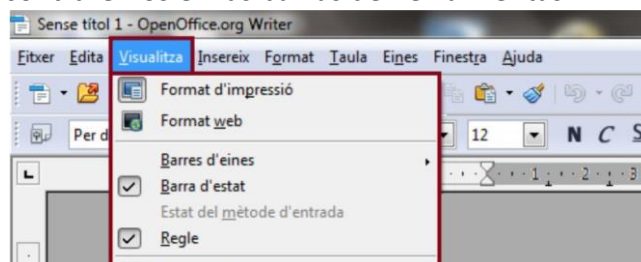
La importancia de los menús en las interfaces radica en que los usuarios finales de las aplicaciones serán los responsables de su utilización y tendrán que tener la autonomía en su uso, sin la necesidad de una comunicación continua con el programador o diseñador.

A continuación, podemos ver algunos ejemplos de distintos tipos de menús que podremos utilizar en las interfaces:

#### MENÚ DE BARRA

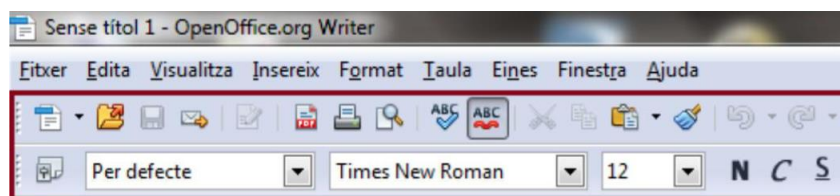
Es el menú más habitual, tanto en aplicaciones específicas como de uso genérico, como pueden ser de navegación por Internet o aplicaciones ofimáticas.

Este menú suele encontrarse en la parte superior de la aplicación. Cada menú contiene otro menú desplegable. Y dentro de éstos, en algunas opciones, tenemos otros menús desplegables. Los podríamos llamar los menús en cascada. Las opciones a las 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.



#### BARRA DE HERRAMIENTAS

Este otro tipo de menú es complementario al anterior. En las aplicaciones encontraremos ambos tipos de menús. El 90% de las veces los usuarios utilizarán los menús de las barras de herramientas, dejando para casos muy concretos la necesidad de acceder a los menús de barra.



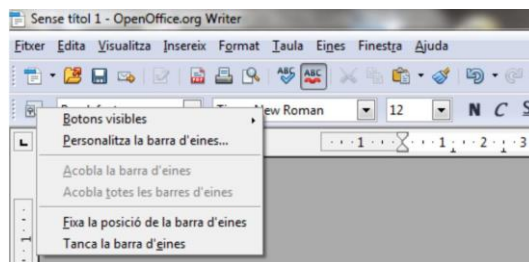
Será importante seleccionar correctamente los iconos a utilizar en las barras de herramientas para que se interpreten correctamente. También es necesario indicar con la ayuda de pequeñas explicaciones el significado de los iconos cuando pasamos el ratón por encima.

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

### **MENÚS CONTEXTUALES**

El menú contextual es la ventana que se abrirá, en determinados sistemas operativos, cuando hacemos clic con el botón secundario del ratón (botón derecho o botón izquierdo en función de cómo esté configurado) sobre un espacio concreto de una aplicación. Normalmente son menús que permanecerán ocultos al usuario y que el usuario activará sobre un objeto determinado. Los menús contextuales se abrirán con una ventana flotante junto al objeto.

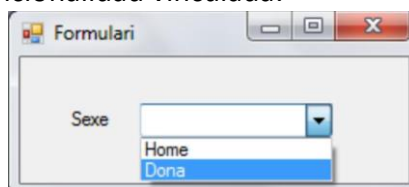
En caso de utilizar el menú contextual en un entorno de sistema operativo, veremos que las opciones que podemos escoger crecen a medida que instalamos programas, de modo que será un elemento vivo.



¿Para qué servirá el menú contextual? Nos facilitará el trabajo más habitual con la aplicación que utilizamos. Al abrir un menú contextual obtendremos una ventana con las funcionalidades relacionadas con la ubicación del puntero (sobre qué objeto se encuentre) en el momento de pedir el menú.

### **MENÚS DESPLEGABLES**

Los menús desplegables ofrecen al usuario la posibilidad de elegir entre un determinado número de opciones. El menú no estará desplegado, sino que habrá que pulsar sobre él para ver las opciones que se ofrecen. Cuando el usuario puede elegir una opción, la lista desplegable se cierra vuelve al estado inicial con la opción seleccionada y aplica la funcionalidad vinculada.



Habrà 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 a menos que se modifique el software desarrollado. Los menús desplegables dinámicos tomarán las opciones de una base de

datos o de un archivo. Si se ha modificado la información externa, los menús ofrecerán unas opciones diferentes en distintos momentos de ejecución.

## **VENTANAS**

El sistema más utilizado en la actualidad 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) al mismo tiempo siguiendo la filosofía del WYSIWYG**. Este sistema permite el acceso de cualquier usuario al ordenador (excepto usuarios con deficiencias visuales).

*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.

Algunas de las propiedades más importantes de las ventanas son:

- **Las ventanas pueden cambiar de tamaño**, haciéndose mayores o menores dentro del área de trabajo.
- **Las ventanas se pueden desplazar por la pantalla tanto de forma 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.

## **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 éste una respuesta** a alguna acción concreta de la aplicación.

Podemos clasificar los cuadros de diálogo en distintos tipos: los **esenciales**, los de **tipo alerta** o los que **permiten seguir trabajando con un programa**. Los cuadros de diálogo esenciales son interacciones obligatorias para los usuarios. Los cuadros se abren de forma 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 en el cuadro, no se puede seguir trabajando.

## **ATAJOS DE TECLADO**

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 forma directa sin tener que pasar por el menú de barra o las barras de herramientas.

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

## **2.2 DISEÑO DEL ASPECTO DE LAS INTERFACES DE USUARIO**

En el diseño del aspecto de las interfaces de usuario deberemos fijarnos en temas como los **colores, las fuentes que se utilizarán para los diferentes apartados de la interfaz, los iconos y la distribución de los elementos**.

El diseño de la interfaz de usuario es un elemento muy importante (en algunos casos podrá considerarse crítico) en el producto final a entregar. Debemos tener en cuenta que algunas

decisiones serán muy personales (ya conocemos el dicho “contra gustos no hay nada escrito”), pero habrá que seguir unas indicaciones universales.

## COLORES

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

Si hablamos de colores en el diseño, deberemos tener en cuenta varios aspectos. Lo habitual es encontrar contraste entre los colores para poder entender mejor la información. 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 ese color con todo lo que tenga que ver con las opciones relacionadas con el apartado.

Existen una serie de normas para la utilización efectiva del color en las interfaces gráficas de usuario. De seguirlas, el diseño cromático de la interfaz será correcto desde un punto de vista cromático.

- Evitar la visualización simultánea de colores complementarios muy saturados.
- Evitar el color azul puro para el texto o 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 mayor edad necesitan colores con mayor luminosidad para poder distinguirlos.
- Los colores se perciben diferentes si las condiciones de iluminación ambiental cambian.
- Evitar el uso de una gama de colores basados en una sola tonalidad.
- La luminosidad no es igual al brillo.
- Diferentes tonalidades tienen distintos niveles de saturación.
- No todos los colores son igualmente legibles.
- Los colores se interfieren mutuamente.
- Hay que tener en cuenta la diferencia entre el color de pantalla y el color impreso.
- Agrupar los elementos relacionados sobre un mismo color de fondo.
- Los colores similares expresan significados similares.
- Los colores luminosos y saturados atraen la atención del usuario.
- Ordenar los colores por su posición en el espectro.

## FUENTES

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

Hay muchos tipos de fuentes y muchas posibilidades de utilizarlas, pero hay que tener cuidado con que estas fuentes estén incorporadas al sistema operativo o que no sean compatibles, lo que puede hacer que no se puedan utilizar de forma adecuada.

Además, como se ha dicho anteriormente, habrá que utilizar fuentes llamadas básicas para que sean aceptadas por cualquier entorno de trabajo o sistema operativo. En caso de utilizar



una fuente no básica, deberá empaquetarse 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.

La legibilidad del texto se ve afectada por distintos factores:

- **Proporcionalidad.** Es una característica de la mayoría de las fuentes que consiste en asignar un espacio horizontal a cada carácter de acuerdo a su anchura.
- **Tamaño de la fuente.** Cabe 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 aún mejor.
- **Mayúsculas/minúsculas.** La mezcla de mayúsculas y minúsculas hace los textos más legibles.
- **Espaciado e interlineado.** El espaciado, tanto de carácter como de palabra, depende de la fuente que utilicemos y es recomendable no tocarlo. 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 encontrar el comienzo de la siguiente línea, y si es más larga de lo que podemos abarcar con el movimiento del ojo nos obligará a mover la cabeza, cosa que nos provocará fatiga.
- **Justificación.** La justificación es la inserción de espacios extra entre las palabras para alinear las líneas tanto a la derecha como a la izquierda.
- **Maquetación.** La maquetación es la distribución de los distintos párrafos de texto (y de otros elementos gráficos) en un espacio determinado.
- **Márgenes.** Hay que evitar el “síndrome del procesador de textos”, que consiste en escribir de lado a lado de la página sin dejar márgenes ni a la derecha ni a la izquierda.
- **Distinción tipográfica.** Es necesario utilizar los cambios tipográficos de cursiva, el cambio de fuente y negrita sólo cuando aporten realmente información y sin abusar de ellos.

## ICONOS

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

De esta forma 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 utilicen sean conocidos.

Es necesario hacer un uso razonable de los iconos, sin utilizar el recurso más de lo recomendable. La velocidad de reconocimiento de los iconos acaba siendo similar a la del texto, pero nunca es superior. De hecho, se producen más errores utilizando 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 falta de satisfacción.

## DISTRIBUCIONES DE LOS ELEMENTOS

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 realizar las acciones necesarias para una aplicación**. Además, debe ser consistente.

Al igual que se ha ido comentando a lo largo de este apartado, es interesante adaptar la interfaz desarrollada al entorno de trabajo de la misma, distribuyendo los elementos de forma análoga a otras aplicaciones o interfaces del sistema operativo. Todo lo que el diseñador pone en el diseño de una interfaz debe estar justificado, debe tener una consistencia y debe seguir una organización espacial.

Otro aspecto a tener en cuenta en la organización espacial de nuestra información es las relaciones entre los distintos elementos de información. Los elementos de información relacionados por el contenido también deben estar visualmente relacionados.

## 2.3 DISEÑO DE LOS ELEMENTOS INTERACTIVOS DE LAS INTERFACES DE USUARIO

Hay otros muchos elementos que se utilizan 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.

### BOTONES DE ÓRDENES

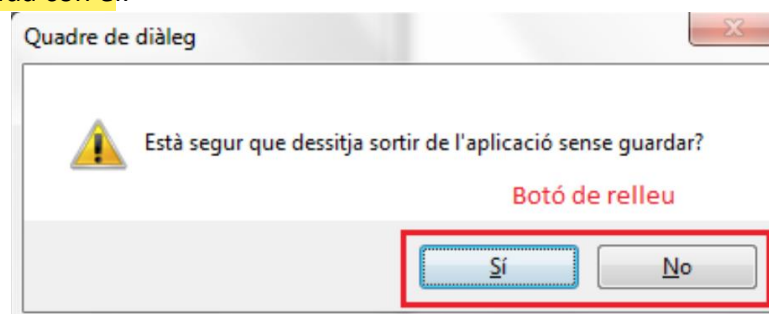
Los botones de órdenes se **consideran dispositivos de control**. De esta forma un botón es un objeto de control de la interfaz de usuario. Estos botones permiten **introducir datos de confirmación en la aplicación**, pero también permiten seleccionar datos o introducir datos en el sistema.

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:

### BOTONES DE RELIEVE

Son los **botones más habituales, los que más se utilizan en todo tipo de sistemas operativos**. Su nombre le debe a su forma, que busca la simulación de volumen. Un botón de relieve contiene un texto o una imagen que identifica una funcionalidad de la aplicación. Cuando un usuario elige un botón de relieve y pulsa sobre él, **el botón simula un hundimiento y realiza la acción relacionada con él**.

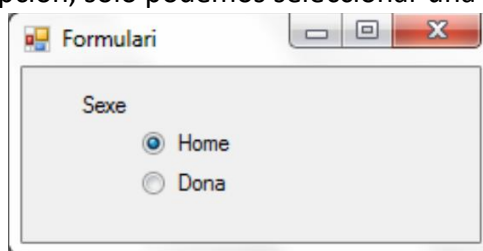


### BOTONES DE RADIO

Los botones en forma de radio o botones de opción ofrecen al usuario una lista de opciones que tienen a la izquierda o derecha (menos habitual) una pequeña redonda para poder clicar dentro. Si hemos seleccionado una de las opciones, esta redonda pequeña se llenará con un



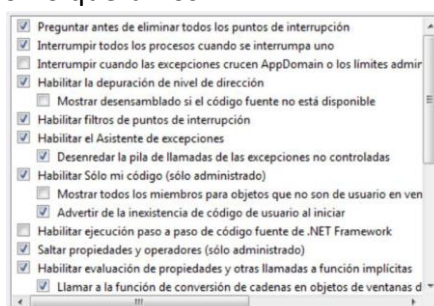
punto y nos indicará que ésta es la opción escogida. Hay que tener en cuenta que, con la opción de los botones de opción, sólo podemos seleccionar una de las opciones ofrecidas.



### CHECKBOX

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

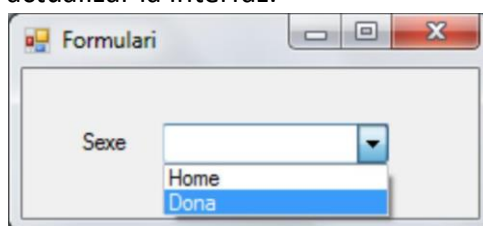
La gran diferencia entre los botones de relieve y los botones de check box es que en los radio buttons sólo podemos seleccionar una opción. En cambio, con los check boxes podemos seleccionar tantas opciones como queramos.



### LISTAS DESPLEGABLES

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

Si quisiéramos ofrecer una lista de opciones sin permitir la opción de agregar un nuevo valor, entonces estaríamos obligados a utilizar el list box, ya que el combo box no tendrá ninguna opción que permita esta configuración. Sin embargo, 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.



## 3. OTRAS PAUTAS DE DISEÑO

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 de lo relacionado con su posible manipulación o destrucción. Es decir, la vinculación de los datos a una base de datos con la interfaz con la que trabajará el usuario.

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

### 3.1 PRESENTACIÓN DE LOS DATOS

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

La aplicación desarrollada (y sus interfaces) servirá para comunicar al usuario con estos datos, pero esto debe hacerse de manera transparente para el usuario.

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

Todas estas cuestiones nos ayudarán a decidimos por una forma de presentar los datos o por otra.

### **3.2 SECUENCIA DE CONTROL DE LA APLICACIÓN**

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 del mismo. Es decir, si el usuario pide ejecutar una operación que comporta diversas acciones con la base de datos, deberemos 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 interfaz (y de una aplicación en general) debe 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 distintas 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 diversas acciones interactivas. Si no puede controlar la secuencia de interacción, se sentirá sin el control de la aplicación, insatisfecho, lo que puede provocar el rechazo a la aplicación entera.

### **3.3 ASEGURAMIENTO DE LA INFORMACIÓN**

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

Para ello es necesario establecer un proceso de mantenimiento y una utilización que no comprometa su integridad. Además, se pueden utilizar otros sistemas para asegurar la disponibilidad de la información, lo que minimizaría el riesgo de quiebra de los sistemas informáticos. Estos pueden ser sistemas de reserva o de emergencia que realicen copias de seguridad de forma automática y regular.

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

### 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 que se plantee el uso de la aplicación, podemos encontrarnos con la necesidad de utilizar algunos elementos multimedia específicos como la animación o el sonido.

Si hablamos de interfaces desarrolladas para dispositivos móviles o para entornos web, podremos utilizar 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 a 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.

Para resumir, podríamos remarcar uno de los objetivos de la usabilidad y el diseño de interfaces: “Una interfaz de usuario estará bien diseñada cuando el programa se comporte exactamente cómo el usuario piensa que debería hacerlo.”