



CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

Índice Tema 8

Introducción.

1. Concepto.

- 1.1. Ideas de interfaz.
- 1.2. Ilusión del usuario.

2. Interfaz de usuario. Interacción hombre-máquina.

- 2.1. Modelos de diseño de interfaz.
- 2.2. Tipos de interfaces.
- 2.3. Ventajas e inconvenientes.
- 2.4. Funcionamiento modo gráfico y modo texto.
- 2.5. Metáfora del escritorio.

3. Principios del diseño de interfaces.

- 3.1. Principios fundamentales.
- 3.2. Características humanas del diseño de interfaz.
- 3.3. Características de un interfaz.

4. Presentación de la información.

- 4.1. Dispositivos utilizados con interfaces.
- 4.2. Dispositivos para discapacitados.

5. Asistencia y ayuda al usuario.

- 5.1. Sistemas de ayuda.

6. Evaluación de interfaces.

- 6.1. Ciclo de evaluación del diseño de la interfaz.

()

()

()

()



CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

TEMA 8

Diseño del interfaz de usuario. Interacción hombre-máquina. Concepto. Principios de diseño de interfaces. Presentación de la información. Asistencia y ayuda al usuario. Evaluación de interfaces.

INTRODUCCIÓN.

Las organizaciones han dedicado un gran esfuerzo a la labor de análisis de los procesos y tareas de producción, con objeto de racionalizar el trabajo, y conseguir así un aumento de la rentabilidad. Esto ha ocasionado una creciente necesidad de manipulación de la información.

El trabajo de oficina ha cobrado una gran importancia y se plantea la necesidad de la mecanización de la misma. La aparición de los ordenadores personales ha contribuido a facilitar la automatización de diversas tareas de oficina como el proceso de textos, hojas de cálculo, bases de datos, autoedición, etc.

No obstante, la implantación del ordenador personal parece estar restringida al ámbito de las secretarías y los administrativos, cuando lo ideal, para lograr el máximo beneficio a las grandes inversiones necesarias para automatizar una oficina, es considerar también a directivos, profesionales y técnicos a los que el ahorro de tiempo y facilidad de acceso a la información que necesitan para su trabajo, les permita dedicarse a actividades más creativas o productivas para la organización.

El proceso de información es una de las funciones básicas en toda organización y comprende todo tipo de tratamiento y almacenamiento. Para comunicar y diseminar la información se utilizan todos los mecanismos que la tecnología de comunicaciones pone a nuestro alcance: teléfono, correo, correo electrónico, redes de comunicaciones de todo tipo, medios visuales, etc.

Como conclusión podemos extraer que los sistemas de proceso de información de las organizaciones son cada vez más complejos.

En la implantación de los sistemas de información, juega un papel decisivo la consideración del aspecto humano de los sistemas informáticos. En este sentido, la convivencia entre todos los elementos que se manejan en un entorno de trabajo es necesaria para hacer receptivos y útiles los sistemas informáticos.

1. CONCEPTO.

Interfaz de usuario (IU): conjunto de elementos que actúan como frontera, entre el dominio del hombre y de la máquina con la misión de lograr y facilitar la comunicación entre ambos. Referido a los sistemas de información, esto incluye el software, el hardware y la documentación (manuales, ayudas, referencias, tutoriales...).

1.1. IDEAS DE INTERFAZ.

De un modo genérico, una interfaz son dispositivos de adaptación a dos entidades distintas. Una interfaz tiene como misión conciliar las características funcionales de dos entidades que han de entenderse. En un ordenador, una interfaz es tanto un grupo de especificaciones, como una entidad lógica o física. La interfaz es también un conjunto de convenios que permiten el intercambio de información entre dos entidades que pueden ser físicas o no. Una interfaz hombre/máquina es un conjunto de dispositivos que la máquina en acción debe presentar al ser humano para proporcionarle la mejor adaptación posible ante sus mecanismos de entrada y salida. La interfaz de usuario actúa como intermediario entre el sistema informático y el usuario. Dado el aumento de la complejidad de los ordenadores, es deseable que sea la interfaz de usuario, el que se acerque a éste, es decir, debe ser lo más coloquial posible. Ello supone que ha de ser natural, fácil de aprender, fácil de usar y consistente.

Actualmente las tecnologías de interfaces de usuarios ofrecen sistemas gráficos basados en menús, ventanas, iconos, comandos y uso de ratón. Los sistemas avanzados incorporan la posibilidad de reconocimiento y síntesis de voz. Ejemplo de interfaz coloquial son Macintosh, Microsoft Windows...

Otra tecnología más reciente son los llamados sistemas multimedia, en los que la comunicación con el usuario se basa en información de texto, voz, datos, gráficos e imágenes.

La interfaz del usuario es la forma en la que las funciones del programa se ponen a disposición del usuario.

La interfaz de usuario tiene un impacto decisivo en la actividad de un sistema informático, pues es quien interviene en el grado de facilidad de aprendizaje y manejo de todas sus herramientas.

Cada vez se elaboran interfaces más potentes y complejas internamente, debido al aumento sistemático de la capacidad y posibilidad de los sistemas de información (procesadores más rápidos, más memoria disponible, etc.) aumentando por consiguiente la complejidad tecnológica. Mediante interfaces sencillas de prestación y manejo se oculta la complejidad del sistema.

1.2. ILUSIÓN DEL USUARIO.

La gran mayoría de los usuarios habituales de los ordenadores personales suelen estar lejos de comprender o de estar interesados en comprender la máquina física. Para ellos ha sido preciso crear la ilusión de que el ordenador es muy sencillo. Siempre que ha sido posible, se han instrumentado mecanismos de diálogo, operación o ejecución, con el fin de hacerles transparentes (en el sentido de «no visibles») todos los niveles de máquinas lógicas. A esto se le llama «ilusión de usuario».

Así se comprende la aparente paradoja de que los ordenadores, siendo cada vez más complejos, tienden a ser cada vez más simples. Es una ilusión producida porque al usuario sólo se le permite ver una máquina virtual hecha a medida de la complejidad que aquél es capaz de interpretar.

Cuando el usuario no es informático, la ilusión se consigue por un lenguaje coloquial, que toma la forma de un programa empaquetado, con sus comandos de uso o menús, actualmente combinados con iconos en pantalla, un ratón de control, ventanas imaginarias, teclas sensibles al tacto, palabras o frases definidas en un vocabulario restringido, un lenguaje de consulta con sintaxis cercana al lenguaje natural, etc.

En la informática personal, se ha diseñado una nueva situación y se han preparado unos instrumentos para un nuevo usuario y un nuevo mercado, en el que lo importante es la posibilidad abierta de empaquetar aplicaciones por un costo razonablemente bajo y modular, controlables por el usuario desde un lenguaje coloquial y ejecutadas con una eficacia para él desconocida.

El concepto de interfaz de usuario es el que nos permite materializar esas ideas.

2. INTERFAZ DE USUARIO. INTERACCIÓN HOMBRE-MÁQUINA.

Si consideramos las interacciones entre dos personas que trabajan en equipo, observamos que esa relación/comunicación a veces no es sencilla, incluso puede ser muy complejo transmitir/recibir información precisa correctamente. Si consideramos ahora el caso en que una de las partes no es una persona, sino una máquina; que esa máquina no puede estar preparada para responder a los cuasi-infinitos estímulos y por diferentes medios que una persona puede transmitir, sino sólo para y por algunos, y por medios limitados y que además quien programa la máquina quizá ni siquiera conoce a la persona con la que va a comunicarse, nos damos cuenta de la complejidad e importancia que puede llegar a tener un elemento que condicionará directamente la mediación hombre-máquina. El sistema que facilita la interacción y la comunicación entre dos sistemas de distinta naturaleza, el ser humano y el ordenador, es lo que llamamos interfaz de usuario.

La interfaz de un sistema de información ha de ser, genéricamente, natural, fácil de aprender, fácil de usar y consistente.

Algunas de las causas por las que un diseño de interfaz de usuario apropiado y de calidad es importante en el contexto de diseño del sistema SW:

- Los usuarios frecuentemente juzgan un sistema por su interfaz, y aunque esto pueda parecer solamente una percepción errónea del usuario, el diseño de la interfaz es fundamental para el éxito.
- Si la interfaz de usuario está bien diseñado, el usuario encontrará la respuesta esperada a su actuación, evitando los efectos sorpresa.
- Un buen programa con una interfaz pobre, transmite mala imagen; mientras que una buena interfaz puede realzar un programa mediocre; pero no debe primar la estética sobre la utilidad.
- Un diseño de interfaz pobre puede provocar que un usuario cometa errores catastróficos. No se debe encerrar al usuario pero sí controlar la casuística de acciones y eventos.
- El diseño de una interfaz de usuario pobre y poco amigable es la razón por la que muchos sistemas quedan aparcados, o fracasa su implantación.
- El diseño de interfaz de usuario tiene tanto que ver con el estudio de las personas como con los aspectos de la tecnología.

Un aspecto clave del éxito de la implantación de la tecnología en entornos de trabajo es la convivencialidad. Consiste en la coexistencia del aspecto humano con el técnico mediante una combinación de técnicas que casi podríamos calificar de arte. Así, habrá que compaginar adecuadamente: eficacia, simplicidad y coherencia, dando facilidades de personalización pero sin crear relaciones maestro-esclavo.

El objetivo de la convivencialidad es integrar las máquinas con los individuos, y a éstos en las organizaciones, y para ello, en los sistemas de información, es clave la interfaz.

2.1. MODELOS DE DISEÑO DE INTERFAZ.

Desde el punto de vista de la ingeniería del software podemos pensar en varios modelos para el diseño de interfaces:

- Modelo de diseño. Un modelo de diseño del sistema completo incorpora representaciones de datos, arquitectónicas, de interfaces y procedimentales del software.
- Modelo de usuario. Hace hincapié en los perfiles de los distintos usuarios.
- Percepción del sistema. Producida por los creadores del sistema.
- Imagen del sistema. Imagen del sistema que llega a la cabeza del usuario.

Los cuatro modelos (de diseño, de usuario, percepción del sistema e imagen del sistema) permiten al diseñador de interfaz satisfacer el elemento clave del diseño de la interfaz de usuario: «Conocer al usuario, conocer las tareas».

2.2. TIPOS DE INTERFACES.

A lo largo de la historia de los sistemas de información se han utilizado distintos tipos de interfaces de usuario que han ido evolucionando tanto en función de las necesidades de los trabajos que se requieren de las máquinas, como de la necesidad de optimizar y facilitar el trabajo de las personas. A menudo los distintos tipos de interfaces no son excluyentes, es decir, puede disponerse de más de un tipo para un mismo sistema.

A) Interfaces de líneas de comandos: CUI (Command-line User Interface).

Se interactúa con ellas mediante sentencias dadas desde la línea de comandos, son las más antiguas de la interacción hombre-máquina, no son muy amigables ya que son difíciles de aprender, pero son muy rápidas para usuarios expertos.

El usuario escribe órdenes encaminadas a realizar una acción, utilizando un lenguaje formal con un vocabulario y una sintaxis relativamente estrictos, no recibe mucha información por parte del sistema, y debe conocer cómo funciona el ordenador y dónde están los programas (casi nada está oculto). Los comandos no tienen por qué estar formados por una serie de palabras clave. En ocasiones están formados por combinaciones de teclas. En el teclado de un ordenador personal existen hasta 12 «teclas de función» y 2 teclas muy particulares la «Ctrl» (control) y la «Alt» (alternativa). Mediante combinaciones de estas dos últimas con las de función o incluso con las propias teclas alfabéticas muchos programas realizan el acceso a sus funciones con una interfaz orientada a comandos. Cuando no se recuerda la secuencia correcta de comandos es difícil que el usuario llegue a la opción deseada sin la ayuda de otra persona o del manual del producto. El modelo de esta interfaz es el del programador, no del usuario.

Las ventajas de este tipo de interfaz las encontramos en su flexibilidad, potencia y control por parte de usuarios experimentados, ya que suelen poder programarse a base de «scripts». No es apropiado para principiantes, pero puede ser parte de la interfaz para que se pueda utilizar una vez que se tenga experiencia.

No obstante cuenta con serios inconvenientes:

- Carga de memoria al usuario, ya que se necesitan memorizar los comandos, y la ayuda no es fácil.
- Los nombres de los comandos no siempre son adecuados a las funciones, o puede dar lugar a que el significado de ciertas órdenes sea mal comprendido o interpretado.
- La sintaxis suele ser muy estricta, esto puede llevar a cometer errores muy graves.

B) Interfaces basadas en menús.

Un menú consiste en una serie de opciones que se presentan en pantalla y ante las cuales el usuario debe realizar la selección de una de ellas. El aspecto de los menús puede ser bastante variado y en especial la forma de acceder a las distintas opciones que en ellos se presentan, pero su filosofía siempre es la misma.

Lo que identifica claramente a un programa basado en menús frente a uno que lo haga en comandos es que todas las opciones que en un momento dado se pueden utilizar están presentes en pantalla o pueden aparecer de una forma simple. Seleccionando una de las opciones se puede llegar finalmente a la función deseada, o bien aparecer un submenú de items relacionados con la opción elegida. Debido al elevado número de funciones que presentan la mayoría de los productos de cualquier tipo, lo normal es que sea necesario navegar por distintos menús hasta llegar a activar la función deseada.

Las interfaces de menús aparecen cuando el ordenador se convierte en una herramienta del usuario y no sólo de los programadores. Permiten «navegar» dentro de un sistema presentando rutas que llevan de unas opciones a otras. Seleccionan elementos de una vista que representan propiedades o acciones que los usuarios desean realizar sobre algún objeto.

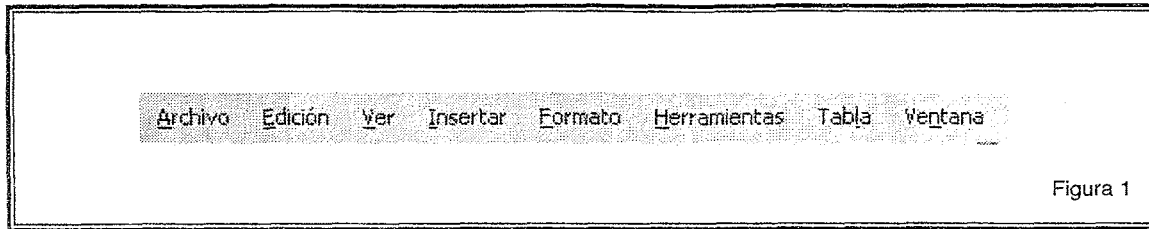
A su vez podemos distinguir las interfaces de menú en:

- Menús de pantalla completa.

Son los primeros dentro de los de este tipo, las opciones ocupan toda la pantalla y suelen estar todas a la vista, aunque seleccionando alguna de ellas nos puede llevar a otra pantalla completa de menús (no es habitual, ya que se pierde rápidamente la noción de la situación).

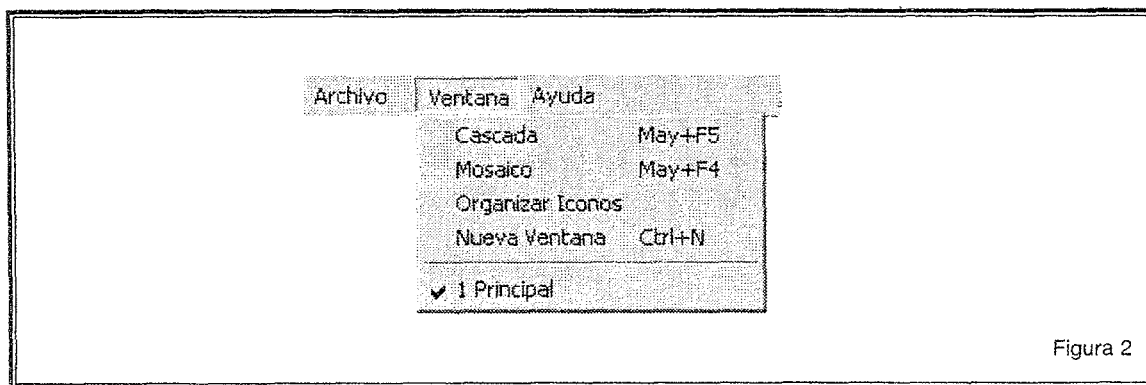
- Menús de barra.

Suelen estar en la parte superior o inferior de la pantalla dejando el resto para mostrar información u otros usos de la aplicación. Contienen una lista de acciones genéricas que dan paso a acciones directas, o bien abren menús desplegables donde se concretan.



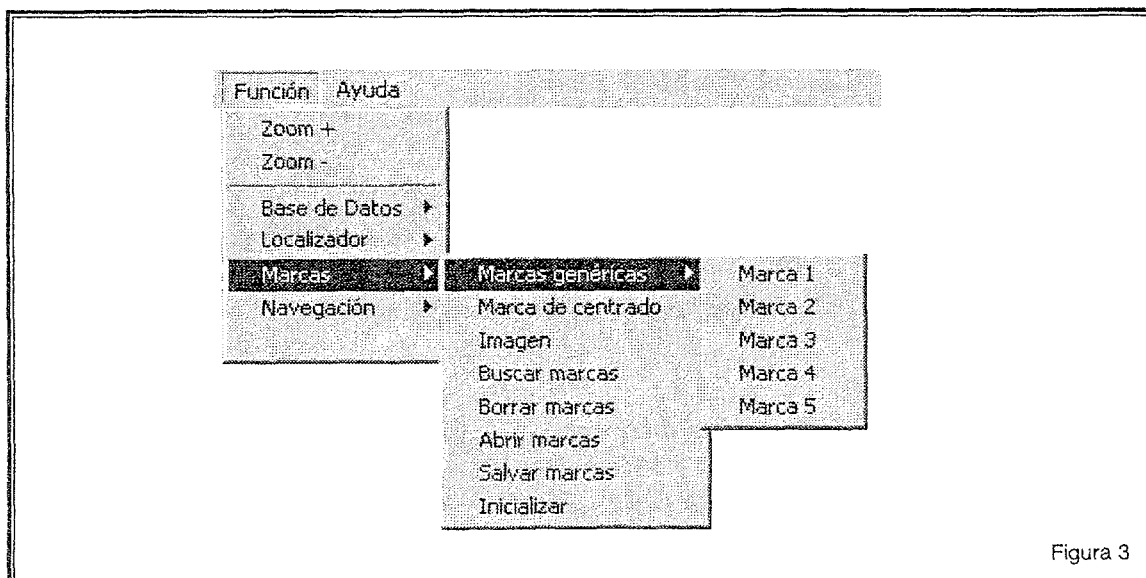
- Menús desplegables.

Así llamados porque simulan la acción de desplegarse o replegarse en base a un título común o acción genérica del desplegable que puede ser una acción perteneciente a un menú de barra.



- Menús en cascada.

Los menús desplegables pueden llevar a su vez a otros, en cascada. Es habitual que éstos, además de poder desencadenar acciones, sirvan para habilitar o deshabilitar opciones, cambiar características...



- Menús gráficos: paletas o barras de herramientas.

Son paletas con acciones, herramientas y opciones que se pueden colocar en pantalla, son habituales en aplicaciones gráficas, de trazado, retoque fotográfico, diseño...

- Menús contextuales o pop-up.

Las opciones presentadas son las posibles respecto del contexto en que estamos, son fáciles de aprender pero lentos para usuarios expertos.

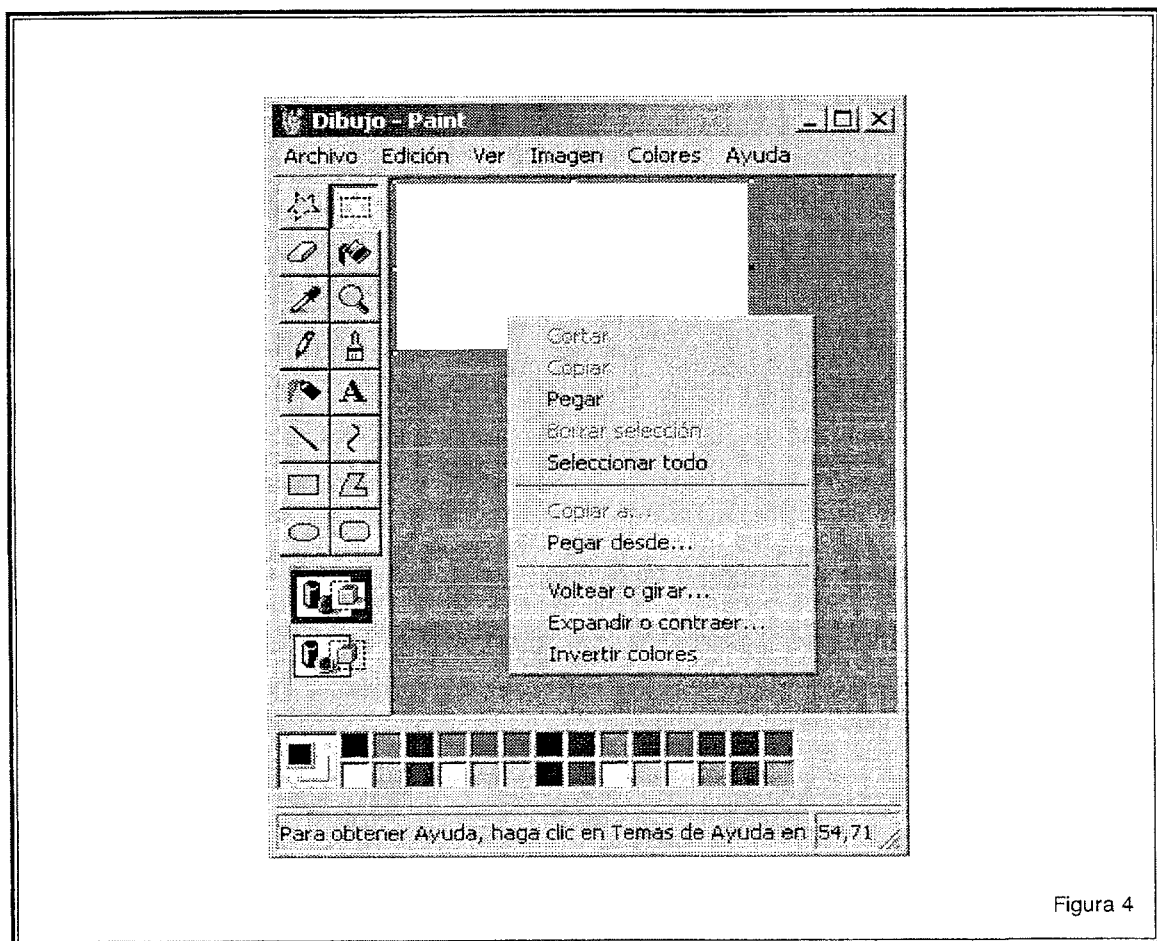


Figura 4

Las interfaces de menú son buenas para usuarios noveles o esporádicos (son fáciles de aprender y recordar). Normalmente se usan en conjunción con otros tipos de interfaces.

Se recomienda seguir las siguientes indicaciones para obtener menús coherentes e intuitivos:

- No ocupar demasiado espacio en pantalla.
- Recordar la información acumulada de menús precedentes.

Los menús permiten una personalización directa, esto es, como el usuario tiene la idea de que están a la vista todas sus opciones, los menús pueden mostrar más o menos opciones dependiendo del tipo de usuario al que se presentan, sus características, su rol en ese momento, etc.

C) Interfaces basadas en gráficos: GUI (Graphical User Interface).

Están orientadas a la aplicación, poseen un dispositivo apuntador (típicamente un ratón) y consideran la superficie de la pantalla como un todo continuo en el que se sitúa cualquier elemento en cualquier posición ya que es accesible con el apuntador allí donde esté. Se enumeran a continuación algunas características principales:

- Promueve la consistencia de la interfaz entre programas.
- Sigue el paradigma de la interacción elemento-acción, mezclado con el estilo acción-objeto de los CUI o las interfaces de menú.
- Se puede manipular la información, no sólo visualizarla.
- Provee elementos de interfaz estándar como menús y diálogos.
- Existe una muestra visual de la información y los objetos (iconos y ventanas). Usuario activa funciones presentadas con iconos.
- 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).
- La interacción con el usuario y el sistema se realiza por medio de eventos.
- Los usuarios pueden ver en la pantalla los gráficos y textos tal como se verán impresos (WYSIWYG).
- Además de los conceptos de copiar, cortar, pegar... aparecen otros como arrastrar, soltar...

D) Interfaces orientadas a objeto: OUIs (Object Oriented User Interfaces).

Su aspecto es similar al de las GUIs. La diferencia estriba en el modelo subyacente: las GUIs son interfaces orientadas a la aplicación, mientras que las OUIs están orientadas al objeto. El estilo de interacción de las OUIs es el de objeto-acción y se esconde la organización del sistema al usuario.

Actualmente existe una mezcla de productos orientados a la aplicación y al objeto, aunque se está produciendo una migración a estos últimos. Las aplicaciones están dejando paso a conjuntos de objetos.

2.3. VENTAJAS E INCONVENIENTES.

Cada tipo de interfaz tiene ventajas e inconvenientes.

Las ventajas de una interfaz basada en gráficos o menús son evidentes de cara al usuario que se acerca por primera vez a una herramienta dotada de tal interfaz, por ser intuitivo y por la continua sensación de sentirse guiado por el propio programa hacia la función buscada.

Sus inconvenientes aparecen a medida que el usuario va ganando destreza, ya que las operaciones tienden a hacerse tediosas debido a que el elevado número de opciones y funciones que proporcionan los programas hacen que sea necesario emprender un auténtico safari a través de los menús y submenús. En líneas generales, una interfaz basada en comandos es más difícil de aprender al principio, pero una vez dominada, su utilización diaria es más ágil.

Este compromiso entre facilidad de aprendizaje y agilidad de uso es un tópico en la discusión sobre el software. Algunos fabricantes ofrecen la posibilidad de trabajar alternativamente en cualquiera de los modos, o bien en sus productos aparecen mezclas de estos tipos de interfaz.

2.4. FUNCIONAMIENTO MODO GRÁFICO Y MODO TEXTO.

Hemos hablado sobre la forma en la que el usuario dialoga con el programa para utilizar las funciones que éste le ofrece. Pero en paralelo con esta discusión hay que tener en cuenta otro aspecto igualmente importante de la interfaz, relacionado con la discusión sobre pantallas, y que consiste en la forma en la que el usuario ve su programa. Esta distinción, es decir que los programas pueden funcionar modo gráfico o en modo texto.

Los programas que trabajan en modo gráfico son capaces de proporcionar interfaces mucho más sofisticadas que los que sólo lo hacen en modo texto. Estas interfaces suelen contener iconos, y están pensadas para soportar la filosofía «WYSIWYG» iniciales de «What You See is What You Get» (lo que ves es lo que obtienes). Uno de los grandes inconvenientes de algunos programas que trabajan en modo texto es que no representan fielmente lo que posteriormente se va a obtener sobre el papel. Así, un procesador de textos que trabaje en este modo no podrá mostrar por pantalla letras en cursiva, ni caracteres de gran tamaño, mientras que tanto unos como otros se pueden obtener por impresora. En un programa dotado con interfaz WYSIWYG constantemente se muestra por pantalla lo que finalmente se obtendrá impreso. Este modo de funcionamiento es imprescindible en programas de dibujo o de autoedición, y cada vez más frecuente en los procesadores de texto.

El modo texto hace referencia a que sólo se pueden utilizar caracteres incluidos en un determinado repertorio. Este tipo de interfaces están prácticamente obsoletas y se mantienen por compatibilidad con sistemas heredados.

Actualmente la práctica totalidad de las interfaces son en modo gráfico.

2.5. METÁFORA DEL ESCRITORIO.

Dentro del objetivo general, de hacer más sencilla la relación máquina/usuario y hacer más amigables los productos que ofrece un ordenador, existe la actitud de huir de la abstracción intelectual para dar una concreción figurativa a la representación material del conocimiento.

Esta actitud persigue el uso de la metáfora, representando en la pantalla los nodos y enlaces no por nodos y vectores de grafos en espacios geométricos, sino intentando representar en ella simbólicamente la temática de la información. Y ello no sólo fundamentalmente a efectos de ilustración sino, sobre todo, tratando de conseguir que las funcionalidades ofrecidas al usuario simbolizadas por iconos o ubicación en el seno del símbolo ordenador, tengan relación o recuerden la función correspondiente del objeto o tema al que remite la metáfora. Un ejemplo que contempla la actitud descrita, común en la mayor parte de los GUIs actuales, lo constituye la metáfora del escritorio, base de la interfaz de la mayoría de las interfaces gráficas, según la cual la pantalla es la mesa de trabajo. En un escritorio se podría tener un informe abierto, una carpeta con documentos, una calculadora, el correo, etc., pudiendo el trabajador simultanear tareas.

El trabajador deja de realizar algunas tareas para comenzar o continuar otras. Si cualquier interfaz es capaz de provocar en el usuario la ilusión de poder trabajar de ese modo, facilitaría la aceptación y su adaptación a él. Mediante el uso de técnicas sofisticadas se intenta conseguir en una pantalla de ordenador toda la parafernalia en que pudiera consistir el trabajo de una persona en su escritorio, usando iconos que pueden representar ficheros de direcciones, papeleras, calculadoras, correo, etc.

3. PRINCIPIOS DEL DISEÑO DE INTERFACES.

Una interfaz inteligente es más fácil de aprender y de usar si se diseña específicamente para el usuario que la explotará. Las cuestiones importantes en el diseño son ¿Quién es el usuario?, ¿Cómo aprende a interactuar con el software?, ¿Qué es lo que espera del sistema?, ¿Cómo interpreta la información proporcionada por el sistema?, etc. Los programas son utilizados por usuarios con distintos niveles de conocimiento, no existe una interfaz válida para todos los usuarios y tareas. Entonces hay que dar cierta libertad al usuario para que decida el modo de interacción más adecuado.

Uno de los pioneros en el diseño de interacción moderna, Larry Constantine, identificó en 1994 varios principios a aplicar en el desarrollo de una interfaz:

1. Estructura: organiza con significado.
2. Simplicidad: haz fáciles las tareas comunes.
3. Visibilidad: muestra toda aquella información necesaria para una tarea.
4. Retroalimentación: mantén informados a los usuarios.
5. Tolerancia: permite cancelar, deshacer, volver.
6. Reutilización: reduce la necesidad de los usuarios de recordar.

Jakob Nielsen es uno de los personajes más reconocidos del diseño de interfaz web, basándose en su experiencia profesional y en el trabajo de antecesores acuñó una serie de principios reconocidos a aplicar en el diseño de interfaces, denominadas heurísticas. Nielsen ha promocionado tan bien estos principios que prácticamente se ha apropiado de ellos:

1. Visibilidad del estado del sistema.
2. Encaje entre el sistema y el mundo real.

3. Libertad y control por parte del usuario.
4. Consistencia y estándares.
5. Prevención de errores.
6. Reconocimiento antes que recuerdo.
7. Flexibilidad y eficiencia en el uso.
8. Diseño estético y minimalista.
9. Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores.
10. Ayuda y documentación.

3.1. PRINCIPIOS FUNDAMENTALES.

Con el objetivo de proporcionar al usuario una estructura conceptual clara y consistente, ajustando la presentación a las capacidades del usuario para hacer el máximo con la menor cantidad de elementos, puede decirse genéricamente que lo que hay que hacer es: organizar, economizar y comunicar.

Para ello, se consideran fundamentales las siguientes reglas para el diseño de interfaces.

Dar control al usuario: el diseñador debe dar al usuario la posibilidad de hacer trabajo en vez de suponer ése lo que éste desea hacer. La interfaz debe ser lo suficientemente flexible para adaptarse a las exigencias de los distintos usuarios, debe usar términos y conceptos extraídos de la experiencia del tipo de usuario, permitir a los usuarios el uso de teclado o ratón, permitir que interrumpan su tarea reanudándola más tarde, utilizar mensajes descriptivos, respuestas significativas (visuales y auditivas, para garantizar la comunicación en ambos sentidos).

Es muy útil que se puedan deshacer acciones, poder hacer una cómoda navegación por el producto y salida del mismo (distintos niveles de uso del producto para usuarios con distintos niveles de experiencia), así como personalizar la interfaz.

Tener el control también puede resultar peligroso, puede resultar conveniente pedir verificación de cualquier acción destructiva importante, pero en caso de que sea así el trabajo habitual, poder desactivar estas confirmaciones.

Reducir la carga de memoria del usuario significa: aliviar la carga de corto alcance (deshacer, rehacer, copiar, pegar...); basarse en el reconocimiento antes que en el recuerdo (elegir de una lista en vez de teclear); proporcionar indicaciones visuales de dónde está el usuario, qué está haciendo y qué puede hacer a continuación; proporcionar atajos de teclado (teclas rápidas, abreviaturas de menús...); asociar acciones a objetos (menús de tipo contextual); utilizar metáforas del mundo real; presentar sólo la información necesaria (opciones simples/avanzadas, asistentes o wizards); hacer clara la presentación visual; utilizar verbos de acción sencillos para nombrar las órdenes y AYUDA.

Consistencia, de todo tipo.

Interna. Utilizar las mismas convenciones y reglas aplicadas a todos los elementos de la interfaz del sistema de información, tanto para la presentación, para el comportamiento (el mismo objeto se comporta igual en todas partes) o para el resultado de interacciones (el usuario espera los mismos resultados cuando interactúa de la misma forma).

Externa. Con el mundo real, en relación a las otras plataformas existentes, a otros productos y con las convenciones actuales (innovación).

En la realización de tareas. Proporcionar al usuario indicaciones sobre el proceso que está realizando (wizards o asistentes).

Fomentar la libre exploración de la interfaz autoprotegiéndose de los errores que puedan hacerlo fallar.

Anticipación.

Las aplicaciones deberían intentar anticiparse a las necesidades del usuario y no esperar a que el usuario tenga que buscar la información, recopilarla o invocar las herramientas que va a utilizar.

Autonomía.

El ordenador, la IU y el entorno de trabajo deben estar a disposición del usuario. Se debe dar al usuario el ambiente flexible para que pueda aprender rápidamente a usar la aplicación. Sin embargo, está comprobado que el entorno de trabajo debe tener ciertas cotas, es decir, ser explorable pero no azaroso.

Protección del Trabajo.

Se debe poder asegurar que el usuario nunca pierda su trabajo, ya sea por error de su parte, problemas de transmisión de datos, de energía, o alguna otra razón inevitable.

A modo de resumen: minimizar el número de acciones de entrada de datos; permitir personalización; interacción sintonizada al modo preferido del usuario; dejar al usuario controlar el flujo interactivo; eliminar entradas innecesarias y buscar la eficiencia en el diálogo, el movimiento y el pensamiento.

3.2. 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 a las de la máquina, y ésta debe utilizar las suyas para completar las de aquélla.

- 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 ratón.
- Ambiente. El lugar donde va a ser usado el sistema. Cada interfaz tiene que adecuarse al lugar.
- Visibilidad. Tener en cuenta la cantidad de iluminación del lugar.
- Personalidad. De acuerdo a la edad, nivel socio-cultural, 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.

Accesibilidad.

La accesibilidad es el grado en que un producto puede ser usado por una persona con algún tipo de discapacidad (física, cognitiva o tecnológica) respecto a cómo lo usaría una persona sin discapacidad. Debe diferenciarse claramente este concepto del de usabilidad, que se define como «grado en que un producto puede ser usado por los usuarios especificados para obtener los resultados especificados con efectividad, eficiencia, y satisfacción en un contexto de uso especificado».

Conseguir una interfaz accesible se torna crítico cuando hablamos de aplicaciones con arquitectura web, ya que es el escenario donde menos datos tenemos del usuario potencial, pero el desarrollo de este tipo de aplicaciones es una tendencia con mucha fuerza en el ámbito de las tecnologías de la información y las comunicaciones (TIC).

Una aplicación web tiene la característica de que su interfaz no suele diferenciarse de un sitio web normal desde el punto de vista del usuario. Entonces podemos hacer un paralelismo entre las características de accesibilidad que debe tener la interfaz de una aplicación web y la accesibilidad de un sitio web tradicional.

La accesibilidad web significa acceso a la web por cualquiera sin importar la discapacidad, e involucra: a las aplicaciones (y sitios) web, para que las personas con discapacidad puedan percibir, comprender, navegar, e interactuar; los Navegadores Web y reproductores de medios, para que puedan ser usadas eficazmente por personas con discapacidad, y que trabajen bien con tecnologías de ayuda que algunas personas discapacitadas usan para acceder a la Web; y herramientas de creación de la Web, y tecnologías Web en desarrollo que soporten la producción de contenido y sitios Web accesibles, y que puedan ser usadas eficazmente por personas con discapacidad.

El World Wide Web Consortium (W3C) desarrolla tecnologías interoperables (especificaciones, directivas, software y herramientas) para impulsar la Web a su máximo potencial. Es un consorcio internacional, neutral, con más de 400 miembros que promueve la evolución e interoperabilidad de la Web y tiene un fuerte interés en su universalidad. Opera desde tres sedes: MIT para Norteamérica, ERCIM para Europa, Universidad de Keio para Asia. Tiene cuatro «dominios»: Arquitectura, Interacción, Tecnología y Sociedad e Iniciativa de Accesibilidad Web (Web Accessibility Initiative, WAI).

Con la intención de estandarizar y unificar criterios en el ámbito de la accesibilidad, el W3C impulsa la Iniciativa de Accesibilidad Web (Web Accessibility Initiative, WAI).

Puesto que la accesibilidad Web es un problema de muchos niveles, WAI tiene cinco niveles de trabajo.

- Asegurar que las tecnologías Web soportan la accesibilidad.
- Desarrollar directrices para la accesibilidad.
- Mejorar herramientas para evaluar y corregir la accesibilidad Web.
- Desarrollar materiales para educación y expansión.
- Coordinarse con la investigación y el desarrollo.

Recomendación del W3C de las líneas directivas para conseguir accesibilidad.

1. Percibible: hacer el contenido percible por cualquier usuario.

Núcleo de los puntos de control para la directiva 1.

- | | |
|---------------|--|
| 1.1. [NÚCLEO] | Todo contenido no textual que puede ser expresado con palabras tiene un equivalente textual de la función o información no textual que quiere comunicar. |
| 1.2. [NÚCLEO] | Proporcionar medios equivalentes para presentaciones dependientes del tiempo. |
| 1.3. [NÚCLEO] | Información y estructura son separables de la presentación. |
| 1.4. [NÚCLEO] | Todos los caracteres y palabras del contenido deben ser descifrados sin ambigüedades. |

Extensiones de los puntos de control para la directiva 1.

- 1.5. [EXTENSIÓN] La estructura se ha hecho perceptible a más gente a través de presentación(es), de la colocación, y de etiquetas.
 - 1.6. [EXTENSIÓN] Los contenidos del entorno principal son más fácilmente diferenciables que los del ambiente secundario tanto por las deficiencias visuales como auditivas.
2. OPERABLE. Asegurar que el contenido de los elementos de la interfaz es operable por cualquier usuario.

Núcleo de los puntos de control para la directiva 2.

- 2.1. [NÚCLEO] Toda la funcionalidad es operable como mínimo a través de un teclado o una interfaz de teclado.
- 2.2. [NÚCLEO] Los usuarios pueden controlar en cualquier momento los límites en sus lecturas, interacciones, o respuestas excepto cuando este control no es posible debido a la naturaleza de eventos de tiempo real o competiciones.
- 2.3. [NÚCLEO] Los usuarios pueden evitar el parpadeo de la pantalla.

Extensiones de los puntos de control para la directiva 2.

- 2.4. [EXTENSIÓN] Se han agregado la estructura y los mecanismos alternativos de navegación para facilitar la orientación y el movimiento por el contenido.
 - 2.5. [EXTENSIÓN] Se proporcionan métodos para reducir al mínimo los errores y para obtener recuperaciones rápidas y eficientes.
3. COMPRENSIBLE. Hacer el contenido y los controles comprensibles para el mayor número de usuarios posible.

Núcleo de los puntos de control para la directiva 3.

- 3.1. [NÚCLEO] El idioma del contenido puede ser establecido por programa.

Extensiones de los puntos de control para la directiva 3.

- 3.2. [EXTENSIÓN] La definición de abreviaturas y acrónimos puede ser determinada sin ambigüedad.
- 3.3. [EXTENSIÓN] El contenido no se escribirá de forma más compleja de lo necesario, y/o se aportará un suplemento con formas más simples del contenido.
- 3.4. [EXTENSIÓN] La disposición y el comportamiento del contenido es consistente o predecible pero no idéntico.

4. ROBUSTO. Usar tecnologías Web que maximicen la posibilidad de que el contenido funcione con las tecnologías de accesibilidad y agentes de usuario actuales y futuros.

Núcleo de los puntos de control para la directiva 4.

- 4.1. [NÚCLEO] Las tecnologías se usan de acuerdo con las especificaciones.

Extensiones de los puntos de control para la directiva 4.

- 4.2. [EXTENSIÓN] Las tecnologías que necesitan los contenidos se usan ampliamente y están disponibles.

- 4.3. [EXTENSIÓN] Las tecnologías usadas para la presentación y la interfaz de usuario soportarán accesibilidad, o se proporcionarán versiones del contenido alternativas que permitan la accesibilidad.

3.3. CARACTERÍSTICAS DE UNA INTERFAZ.

Podríamos resumir las características de una buena interfaz diciendo simplemente que tiene que ser USABLE, pero conseguir esto no es fácil, para conseguirlo se pueden seguir heurísticas¹ que ayudan a conseguir interfaces usables.

Una interfaz es natural, cuando provoca al usuario sentimientos de estar como en casa. Todo trabajador tiene una forma de actuar y organizarse dentro de un entorno que domina, al cual está acostumbrado y del que es muy difícil que salga de forma voluntaria.

Entonces la manera de conseguir que un trabajador de la información acepte un cambio hacia una automatización, es crearle un sistema lo más parecido a su entorno de trabajo y si es posible con mejoras y ventajas que le permitan realizar trabajos que antes no podía o suponían un esfuerzo lo suficientemente grande como para inducirle a intentar evitarlos.

Una interfaz debe ser fácil de aprender. Debe proporcionar al usuario un sistema de ayuda lo suficientemente potente como para permitir manejarlo a una persona inexperta. Este sistema de ayuda debe tener como mínimo una serie de comandos «HELP» que permitan de una forma general o en un momento determinado consultar al usuario cuáles son las opciones que puede elegir para continuar su trabajo y cómo hacerlo.

El sistema de ayuda puede llegar a ser un obstáculo para el usuario una vez que domine el producto, por lo que esta ayuda no debe ser totalmente automática sino que debe permitir que sea invocada en el momento que se necesite. Una estrategia común para solucionar el problema consiste en la introducción al comienzo del trabajo de un perfil de usuario. Según el grado de perfil, la interfaz ejecutará unas acciones u otras.

Una interfaz consistente debe mantener una uniformidad a lo largo de toda su exposición. Esto es un problema que aparece comúnmente en paquetes integrados en los que cada componente de éste actúa en un mundo distinto e incluso con distinta filosofía, lo que obliga al usuario a cambiar constantemente la mentalidad de trabajo según vaya pasando de un componente a otro.

¹ Heurística: técnica de la indagación y del descubrimiento. En algunas ciencias, manera de buscar la solución de un problema mediante métodos no rigurosos, como por tanteo, reglas empíricas, etc.

Diálogo simple y natural: las interfaces de usuario deben ser simplificadas tanto como sea posible, debido a que cada elemento adicional de información en una pantalla es una cosa más que el usuario debe aprender, una cosa más que posiblemente comprenderá mal y una cosa más entre las que tiene que buscar cuando esté tratando de encontrar lo que necesita. Las interfaces deben igualar la tarea del usuario en una forma tan natural como sea posible.

Lo ideal es presentar sólo la información que el usuario necesita y nada más, en el lugar y tiempo que se está necesitando. Tanto los objetos de información como las operaciones deben ser utilizados en una secuencia parecida a la forma más efectiva y productiva en que el usuario hará las cosas.

La organización de los elementos en la pantalla debe usar las mejores reglas de la percepción humana. Estas reglas dicen que las cosas son vistas como un grupo o una unidad si ellas están cercanas, o parecen semejantes respecto a forma, color, tamaño o tipografía. Estos principios de estructura gráfica deberían ser usados para ayudar al usuario a comprender la estructura de la interfaz.

Los principios de diseño gráfico pueden también ayudar al usuario a priorizar su atención hacia la pantalla haciendo que en ésta se encuentren los elementos para un diálogo importante.

Con respecto al color en el diseño de pantallas, las tres pautas principales son:

- No sobrecargarlo.
- Estar seguro de que la interfaz puede ser usada sin los colores. Probar cómo se ve la interfaz en una pantalla monocromática.
- Tratar de usar el color sólo para categorizar, diferenciar y resaltar, no para dar información, especialmente información cuantitativa.

La regla «menos es más». Añadir información y campos de datos a una interfaz puede distraer la atención del usuario de la información primaria.

Generalmente es posible identificar la información que es verdaderamente importante para el usuario y cuál lo habilitará para ejecutar casi todas sus tareas.

La regla de «menos es más» no sólo se aplica al contenido de la información sino también a la selección de características y mecanismos de interacción para un programa.

Hablar el lenguaje de los usuarios. Como parte de un diseño centrado en el usuario, la terminología en interfaces de usuario debe estar basada en el lenguaje de los usuarios y no en términos orientados al sistema.

Los diálogos deben estar redactados en el lenguaje nativo del usuario y no en un lenguaje extranjero.

Se debe tener cuidado de no usar palabras que no tengan un significado estándar.

Hablar el lenguaje del usuario no siempre implica limitar el vocabulario de la interfaz a un conjunto pequeño de palabras usadas comúnmente.

Mapeos y metáforas. Una forma de alcanzar la meta de tener un diálogo orientado al usuario es tener mapeos entre el desplegado de información de la computadora y el modelo conceptual de información del usuario.

Las metáforas como interfaces de usuario son una buena manera de lograr el mapeo entre el sistema de información y algún sistema de referencia conocido por los usuarios en el mundo real. Aunque las metáforas representan problemas potenciales con respecto a la internacionalización, debido a que no todas las metáforas tienen igual significado en todas las culturas.

La consistencia es uno de los principios de usabilidad básico y de los más importantes.

La misma información debe ser presentada en el mismo lugar en todas las pantallas y cajas de diálogo y debe ser formateada en la misma forma para facilitar su reconocimiento.

La consistencia no es sólo una cuestión del diseño de pantalla, también incluye consideraciones de la tarea y estructura de funcionalidad del sistema.

Retroalimentación. El sistema debe informar continuamente al usuario acerca de lo que está haciendo y cómo está interpretando las entradas de información. La retroalimentación no debe esperar hasta que una situación de error haya ocurrido: debe proveer retroalimentación positiva y parcial cuando la información está disponible.

La retroalimentación del sistema no debe ser expresada en términos abstractos y generales.

Diferentes tipos de retroalimentación pueden necesitar diferentes grados de persistencia en la interfaz.

La retroalimentación llega a ser especialmente importante en caso de que el sistema tenga tiempos de respuesta largos para ciertas operaciones.

Salidas claramente marcadas. A los usuarios no les gusta sentirse atrapados por la computadora. Para incrementar el sentimiento de que el usuario tiene el control del diálogo, el sistema debe ofrecerle una forma fácil de salir de muchas situaciones.

Atajos. Aunque es posible operar una interfaz con el conocimiento de unas cuantas reglas generales, es posible para los usuarios expertos ejecutar operaciones frecuentes especialmente rápido, usando atajos de diálogos.

Los usuarios deben ser capaces de reutilizar su historia de interacción.

Buenos mensajes de error. Un principio básico para el diseño de interfaz de usuario es reconocer que el usuario cometerá errores no importa lo que se haga para mejorar la interfaz y, por lo tanto, debe ser fácil recuperarse de esos errores.

Las situaciones de error son críticas para la usabilidad por dos razones:

1. Por definición, representan situaciones donde el usuario tiene problemas y no podrá usar el sistema para lograr su meta.
2. Presentan oportunidades para lograr que el usuario conozca mejor el sistema.

Los mensajes de error deben seguir cuatro reglas básicamente:

- Deben ser presentados en un lenguaje claro y evitar códigos oscuros.
- Deben ser precisos, no vagos ni generales.
- Deben ayudar al usuario en forma constructiva a resolver el problema.
- Deben ser gentiles y no intimidar al usuario o poner la culpa explícitamente en él.

Además de tener buenos mensajes de error, el sistema también debe proveer buena recuperación de errores.

La forma más común de implementar mensajes multinivel es tener sólo dos niveles y suplir el mensaje inicial con botones que nos lleven a una mayor información. También es posible enlazar una palabra a un lugar en la ayuda en línea.

Prevenir errores. Mejor que tener buenos mensajes de error, es recomendable evitar situaciones de error.

Los modos son una fuente frecuente de errores y frustraciones del usuario y deben ser evitados tanto como sea posible.

Ayuda y documentación. La existencia de ayuda o documentación no reduce los requerimientos de usabilidad de la interfaz, al contrario. Los usuarios usan los manuales y sistemas de ayuda para:

- Búsqueda de información relevante.
- Comprender la información.

De acuerdo al uso que el usuario da a la ayuda puede necesitar tres niveles de información:

- Tarjetas de referencia corta o ayudas de trabajo.
- Tutoriales y/o manuales introductorios.
- Manuales de referencia para usuarios expertos.

4. PRESENTACIÓN DE LA INFORMACIÓN.

La información puede presentarse de muchas maneras distintas: básicamente con texto, dibujos y sonido, etc. La combinación de las distintas formas de presentación es sobre lo que más control se tiene para optimizar la interacción (la colocación, el tamaño, el contexto, el color, la resolución). Pero una presentación llamativa no siempre es lo más conveniente, hay que tener en cuenta que hablamos de una herramienta de trabajo que puede que tenga que usarse continuamente durante horas (evitar el «efecto Las Vegas»).

- Recomendaciones:

- Mostrar sólo la información relevante (en cada contexto).
- No abrumar al usuario con datos.
- Utilizar etiquetas consistentes, abreviaciones estándar y colores predecibles.
- Producir mensajes de error significativos.
- Usar mayúsculas y minúsculas, tabulaciones y agrupamiento del texto para ayudar a entenderlo.
- Usar ventanas para compartimentar.
- Usar representaciones analógicas para representar información que es más fácilmente asimilable de esa manera.
- Estudiar la geografía disponible en la pantalla y usarla eficientemente.

Así como para la comunicación oral entre las personas se necesita un lenguaje, y éste va más allá del lenguaje hablado, utilizando gestos, posturas... (lenguaje corporal), en la comunicación hombre máquina deben optimizarse todos los elementos del lenguaje principal, el lenguaje visible, que se concreta en todas las técnicas gráficas para comunicar.

Elementos principales del lenguaje visible.

- Disposición o Layout. Se refiere a los formatos, proporciones, mallas, organización...
- Tipografía. Tipos de letras y estilos. Se recomienda no usar más de tres tipos de letra, para distintos tipos de información.
- Color y textura. El color es probablemente el elemento de la interfaz que con más frecuencia es mal utilizado.
 - El color, textura, luminancia... no son decorativos, son una potente herramienta de comunicación. Evitar preferencias personales.
 - La memoria para la información en color es más eficaz que para blanco y negro.
 - Una buena práctica es diseñar en blanco y negro y luego añadir color.
 - El color debe utilizarse para agrupar ítems relacionados.
 - Una vez establecida una codificación de color, ésta debe ser utilizada en toda la interfaz (incluida la documentación).
 - Utilizar como máximo 5 ± 2 colores.
 - Distinguir colores centrales y periféricos: azul para áreas grandes sin detalles o texto; rojo, verde, para captar la atención no sólo en la periferia, también en el centro; negro, azul, blanco para la periferia.

- Imágenes. Desde fotografías reales hasta lo abstracto (iconos, símbolos, caricaturas...).
- Animación. Es el cambio en el tiempo de la apariencia visual, por ejemplo: progreso de acciones (copia de ficheros, etc.), estado de medios, acciones posibles (cambios en el cursor al desplazar el ratón...). Pueden ayudar a reforzar iconos importantes, mostrar el estado de un objeto, explicar su comportamiento y ayudar a que el usuario espere.
- Múltiples vistas. Para estructuras y procesos complejos.
- Identidad visual.
- Diseño internacional. Tener en cuenta las diferentes culturas (terminología, símbolos, alfabetos...).

Aunque el lenguaje audible puede y debe usarse en las interfaces, utilizarlo intensamente es, en general, más pesado que el visible tanto para el usuario como para la máquina, además tiene otros inconvenientes cuando se está en salas compartidas (contaminación acústica). Es una buena práctica utilizarlo solamente para informar, no cuando no ocurra nada. Pueden seguirse estas pautas para su uso:

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

4.1. DISPOSITIVOS UTILIZADOS CON INTERFACES.

Los grandes avances en la electrónica combinados con el aumento de capacidad de proceso de los ordenadores personales que permiten manejar y gestionar cada vez más dispositivos da como resultado gran cantidad de dispositivos que cada vez están más extendidos.

Si bien el mercado doméstico puede considerarse cada vez más como laboratorio de ensayo de diferentes dispositivos, sobre todo los que están directamente relacionados con los juegos y el ocio, la gran mayoría de sistemas de información están basados en los tradicionales sistemas de E/S comprensibles para las personas, es decir, pantalla, teclado y algún tipo de dispositivo señalador (ratón, trackball...). Los sistemas de sonido también están presentes en casi todos los puestos de trabajo, y aunque en rigor deben considerarse parte de la interfaz, tanto las aplicaciones heredadas como las que se desarrollan en la actualidad hacen un uso relativo del audio en los entornos de producción.

Así pueden mencionarse como elementos de interfaz masivamente extendidos, joysticks, gamepads, volantes, desktops (combinaciones de teclado y ratón), cámaras, todo tipo de sistemas de conducción de coches, motocicletas, aeronaves, etc.

HMD.

Pantalla acoplada a la cabeza (acrónimo de Head Mounted Display) que muestra una imagen de computadora ante cada uno de sus ojos. La HMD cuenta también con un sistema de seguimiento de posición para controlar la posición de la cabeza del usuario y la dirección en la que esté mirando. Uti-

lizando esta información, el ordenador recalcula imágenes y genera una imagen ligeramente distinta para cada ojo, que debe coincidir con la dirección hacia la que esté mirando el usuario. También se usa la información de posición para transmitir sonidos a través de los auriculares, de forma que parezca que su origen tiene una posición definida en el espacio.

Centrándonos en la interfaz gráfica más extendida, los llamados sistemas de ventanas, si consideramos sus componentes, se pueden abstraer ciertas características comunes a todos ellos; un componente está formado por:

- **Propiedades.** Que representan un valor o estado asociado al componente (son atributos y algo más...).
- **Métodos.** Cada método proporciona un comportamiento de un componente, ya que obligan al componente a realizar una acción.
- **Eventos.** Un evento es una señal, externa o interna a la aplicación, que produce la ejecución de un bloque de código que se escribe como un método (ejemplos: pulsar el ratón, llamar a un método, modificar el valor de una propiedad...).

Los componentes no se encuentran aislados, sino agrupados dentro de contenedores.

Los contenedores también son componentes y se utilizan para recoger y gestionar otros componentes. Un ejemplo de contenedores son las ventanas.

Una ventana es una parte de la superficie total de una pantalla que adquiere cierta independencia para la interacción con el usuario. Habitualmente, el empleo de ventanas permite mantener varias funciones de un programa activas, aunque sólo en una de ellas se pueda trabajar. Por ejemplo, existen procesadores de texto que permiten tener abiertos varios documentos a la vez, lo cual se puede conseguir mediante el empleo de ventanas, una por cada documento abierto.

La edición sólo se podrá hacer en una de ellas, pero siempre se tendrá a la vista el resto de los ficheros. En otras ocasiones, las ventanas se utilizan para presentar la información de un sistema de ayuda o dar al usuario determinados mensajes.

Normalmente, una interfaz basada en ventanas lleva asociada la utilización de un señalador (casi siempre un ratón), o al menos así se recomienda. Se trata de un dispositivo que es preciso desplazar sobre la superficie de la mesa y que está dotado al menos de un botón que se activa mediante la presión con el dedo. Un programa que permita el empleo de ratón responderá a su desplazamiento moviendo a lo largo de la pantalla un puntero que lo representa. Por ejemplo, la forma normal de activar un icono por medio de un ratón consiste en desplazar la imagen del ratón sobre el mismo y pulsar.

Un ratón es una forma alternativa al teclado para la interacción con programas, habitualmente considerada más simple e intuitiva. Piénsese que tanto un teclado como un ratón son elementos realmente extraños para alguien que se acerca por primera vez a la informática, pero en el caso segundo el funcionamiento es mucho más simple ya que sólo hay dos operaciones básicas con él, que son el desplazamiento y la pulsación. Si a esto se le une la presencia en pantalla de iconos, que suelen ser auto-explicativos, se puede entender que con estos elementos sea más sencillo el manejo de aplicaciones.

Un icono es un pequeño dibujo que representa una función del programa. Ejemplo, para ver el contenido de un disco, podría existir un icono con forma de disco que, al ser activado, mostrara el contenido del mismo.

4.2. DISPOSITIVOS PARA DISCAPACITADOS.

Actualmente existen lo que se conocen como células braille, que es un conjunto de elementos eléctricamente móviles los cuales dispuestos en línea constituyen una línea braille. Esta línea usada como terminal del ordenador es capaz de reproducir en braille, mediante software y la interfaz adecuada, una línea de texto convencional. El usuario lee esa línea pasando el dedo sobre ella como si se tratara de una línea impresa. Una vez leída, un nuevo conjunto de caracteres ocupa el lugar de los anteriores, y de esta manera se prosigue hasta completar un texto dado. El uso de mecanismos bidimensionales usando líneas braille y dispositivos hápticos ² se muestra también como una alternativa viable a través de una mejora del confort.

Hay que tener en cuenta que la salida táctil no es la única manera para transmitir a la persona ciega la información de textos codificados en memoria del computador. Los sintetizadores de voz permiten a la información escrita hacerla inteligible a través del oído.

El sintetizador de voz puede leer una pantalla en modo texto, pero no es suficiente para manejar programas y formatos complicados de pantalla. Para conseguirlo es necesario utilizar además un software lector de pantalla (screen reader). Este software permite recorrer la pantalla y así acceder a cualquier software basado en modo texto.

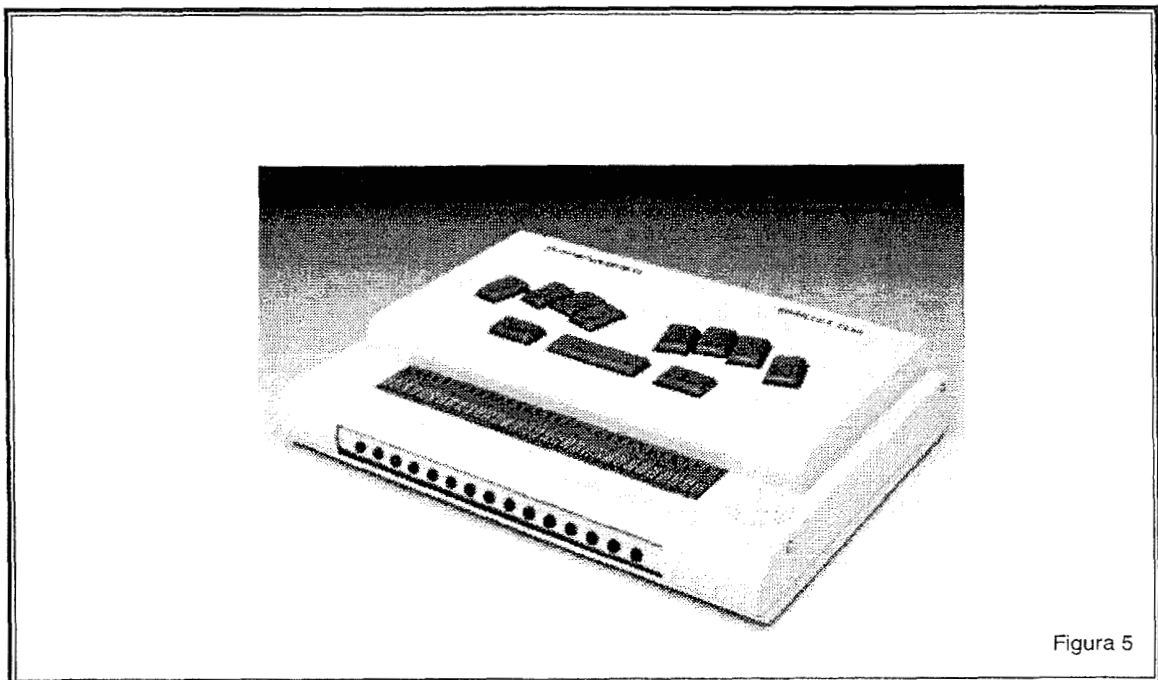


Figura 5

² Un dispositivo háptico es aquel que es capaz, al realizar el operario una acción sobre él, de realimentarle un valor de resistencia a esa acción (realimentación cinestésica).

Inteligencia cinestésico-corporal: este tipo de inteligencia se relaciona con la posibilidad que tiene el individuo para controlar sus movimientos y manejar objetos.

5. ASISTENCIA Y AYUDA AL USUARIO.

Cuando un usuario necesita ayuda, la interfaz tiene que estar preparada para que la información que el sistema pueda proporcionar le llegue cuanto antes, no dificulte en la medida de lo posible la tarea que estaba realizando, y una vez que el usuario se dé por enterado dejarlo en el mismo punto donde se encontraba antes de solicitar la ayuda. Para ello, no sólo interviene la interfaz, el sistema tiene que estar diseñado para atender estas peticiones. El subsistema de ayuda, si se pretende que sea eficiente, no es trivial, ya que se necesita llevar un seguimiento del punto en que se encuentra el usuario cuando solicita ayuda.

Asistencia y ayuda al usuario se articula en torno a:

- Etiquetas, mensajes, respuestas del sistema.
 - Todos los mensajes de aviso o error:
 - En el argot del usuario, pero sin fomentar un argot muy cerrado.
 - Información constructiva.
 - Indicar posibles consecuencias.
 - Señal audible o visible (distinto color, un símbolo).
 - No debe culpar al usuario.
- Sistema de ayuda integrado en el paquete SW:
 - Ayuda de interfaz gráfica: consta de barras de botones, índice y búsqueda por temas, ventanas emergentes y cuadros de diálogo.
 - Ayudas expertas: permiten al usuario realizar una tarea guiándolo paso a paso a través de preguntas que concluyen con la realización de la tarea.
 - Asistente de ideas o gráfico: el programa va introduciendo sugerencias prácticas sobre la forma en la cual tratar el objeto o realizar una acción.
 - Ayudas sensibles al contexto: dependiendo del lugar de la aplicación se proporciona un tipo de ayuda relativa a la operación que se quiere realizar.
 - Tutoriales.
- Documentación y manuales de todo tipo: de usuario, de administración, instalación... Toda esta documentación si es exhaustiva puede llegar a ser enorme, incluso ocupar más que la propia aplicación.

5.1. SISTEMAS DE AYUDA.

Independientemente del modelo de interfaz, la gran mayoría de los programas incorpora lo que se denomina un sistema de ayuda. Se trata de evitar que el usuario tenga que acceder a los manuales para resolver una duda puntual que le surja durante la operación del programa.

Los mejores sistemas de ayuda son los que se denominan sensibles al contexto. Cuando se está en un determinado punto del programa y no se sabe cómo actuar, una ayuda sensible al contexto (invocada mediante un evento específico) es capaz de determinar la circunstancia que determina la petición de ayuda y proporcionar un auxilio muy concreto sobre la materia que interesa. Otros sistemas de ayuda no son tan flexibles, y simplemente dan acceso a una serie de paneles en los que se explica de una forma muy general e insuficiente en la mayoría de los casos la operativa general del programa.

6. EVALUACIÓN DE INTERFACES.

6.1. CICLO DE EVALUACIÓN DEL DISEÑO DE LA INTERFAZ.

Podemos considerar el ciclo de evaluación del diseño de la interfaz como un proceso iterativo basado en la construcción de prototipos.

Una vez creado el prototipo, hay que determinar si satisface las necesidades del usuario, es decir, si es de calidad. La evaluación puede ir desde un método informal, en el que el usuario proporciona sus percepciones, hasta un estudio diseñado formalmente que use un método estadístico para la evaluación de cuestionarios rellenos por usuarios finales, por ejemplo:

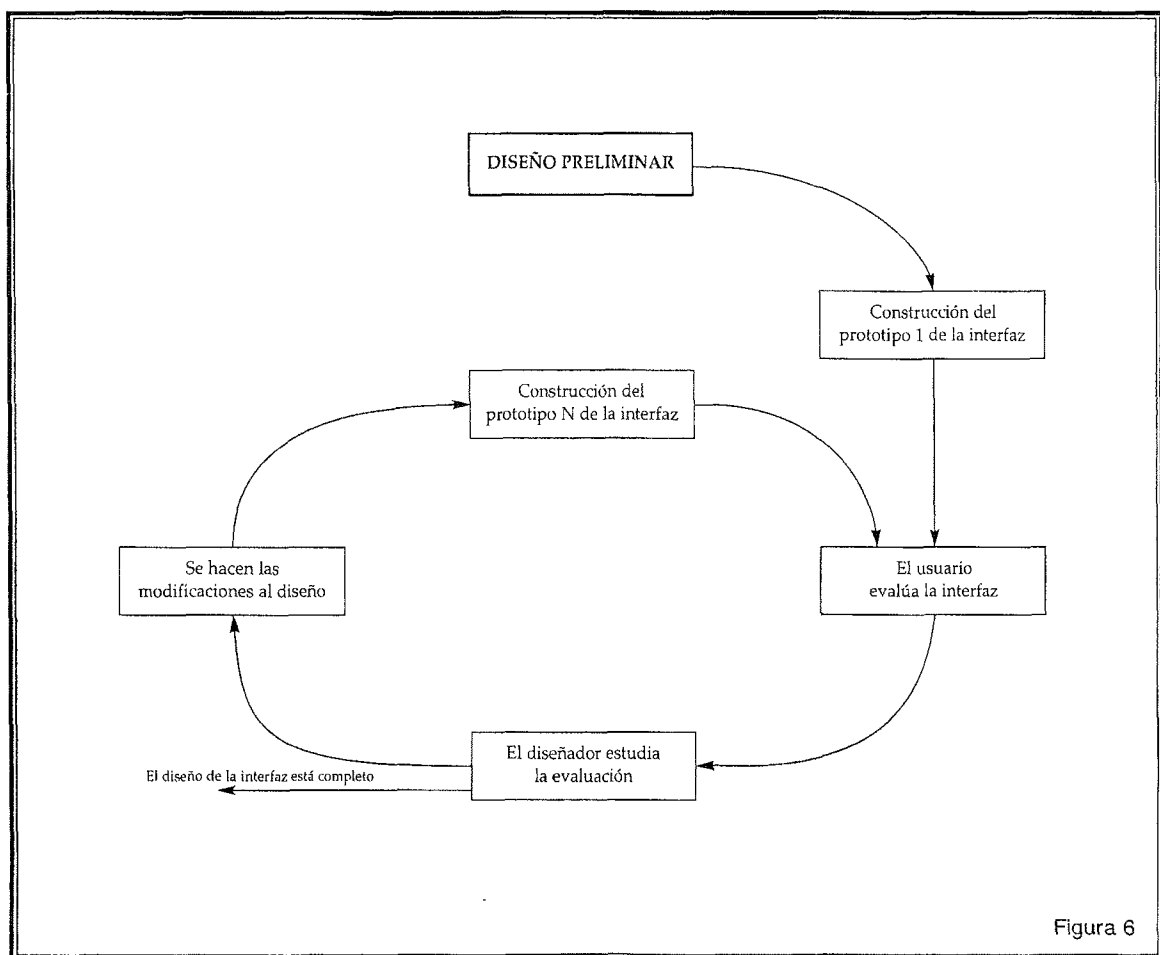


Figura 6

En el diseño preliminar se establecen los criterios de evaluación del modelo de diseño de interfaz. Si se descubren y corrigen en esta fase errores potenciales, el número de bucles se reducirá y se acortará la duración del desarrollo. Una evaluación de calidad antes de crear el prototipo proporciona indicaciones importantes a tener en cuenta:

- El tamaño y complejidad de la especificación del sistema y de la interfaz permiten obtener una indicación de la cantidad de aprendizaje necesaria.
- El número de acciones especificadas, junto al número de argumentos de cada acción da idea del tiempo de interacción.
- El número de acciones, órdenes y estados del sistema se interpreta como la carga de memorización que va a tener que soportar el usuario.
- El estilo de la interfaz, las ayudas y métodos de control de errores indican la complejidad y el grado de aceptación.

Si se utilizan técnicas formales de evaluación, el diseñador puede extraer información más útil.

Los datos que sirven de guía para la modificación de la interfaz pueden ser:

- Cuantitativos. Observando a los usuarios durante la interacción para obtener el número de tareas completas y correctas por unidad de tiempo, secuencia de acciones, frecuencia, tiempo intervenido en mirar el monitor, número de errores y su tipo, tiempo de recuperación, tiempo de utilización de la ayuda, número de referencias a la ayuda por unidad de tiempo, etc.
- Cualitativos: cuestionarios del tipo ¿Son fáciles de recordar las acciones?, ¿cuántas acciones distintas ha utilizado? Comparando con otras interfaces, etc.

REFERENCIAS

- Temario de las pruebas selectivas para ingreso en el Cuerpo Superior de Sistemas y Tecnologías de la Información de la Administración del Estado. ASTIC.
- Temario de las pruebas selectivas para el acceso al Cuerpo de Gestión de Sistemas e Informática de la Administración del Estado. MINISTERIO PARA LAS ADMINISTRACIONES PÚBLICAS.
- <http://www.w3.org>
- <http://www.useit.com>
- <http://web.mit.edu/is/usability>

