



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

Licenciatura En Informática

Programación de Interfaces

*Autor: L.I. María de Lourdes Isabel
Ponce Vásquez*

AGOSTO-DICIEMBRE 2005



Contenido

UNIDAD 1. Introducción a la Programación de Interfaces.....	3
Objetivos:	3
Introducción.....	3
1.1 La disciplina.....	3
1.2 Definición	4
1.3 Historia de la PI	4
1.3.1 Preliminares	4
1.3.2 Los orígenes de la PI	5
1.4 Objetivos de la PI	9
1.5 La interfaz de usuario	9
1.5.1 Psicología	11
1.5.2 Diseño.....	12
1.5.3 Etnografía - Sociología.....	12
1.5.4 Ergonomía o factores humanos	12
1.5.5 Programación.....	13
1.5.6 Inteligencia artificial.....	14
1.5.7 Ingeniería de software.....	14
1.6 Usabilidad	14
1.6.1 ¿Por qué es importante la usabilidad?	15
1.6.2 Comentarios habituales que se hacen respecto de la no necesidad de la usabilidad	15
1.6.3 Principios generales de la usabilidad	15
1.7 El diseño centrado en el usuario.....	18



UNIDAD 1. Introducción a la Programación de Interfaces

Objetivos:

- Entender y describir qué es una interfaz
- Conocer la importancia de la disciplina
- Aprender los conceptos básicos
- Conocer los objetivos de la PI
- Entender qué quiere decir que un sistema es usable
- Aprender a analizar si un sistema es usable
- Entender la importancia de que las computadoras sean accesibles a todos los seres humanos
- Saber que hay muchas disciplinas que contribuyen a la PI y valorar su aportación

Introducción

Cuando los seres humanos y las computadoras interactúan lo hacen a través de un medio o interfaz. Una interfaz es *“una superficie de contacto que refleja las propiedades físicas de los que interactúan, las funciones a realizar y el balance de poder y control”*.

En el caso de la Interacción Persona - Computadora, la interfaz es el punto en el que seres humanos y computadoras se ponen en contacto, transmitiéndose mutuamente tanto información, órdenes y datos, como sensaciones, intuiciones y nuevas formas de ver las cosas. Por otro lado, la interfaz es también un límite a la comunicación en muchos casos, ya que aquello que no sea posible expresar a través de ella permanecerá fuera de la relación mutua. Ahora bien, mientras que a veces esos límites derivan del estado actual de nuestros conocimientos acerca de cualquiera de las partes implicadas, en muchos casos la interfaz se convierte en una barrera debido a un pobre diseño y una escasa atención a los detalles de la tarea a realizar.

Este curso se centra en transmitir qué es una superficie usable, quiénes son los protagonistas que están a uno y otro lado y cuáles son las disciplinas que nos pueden ayudar en este trabajo.

La interfaz forma parte de un entorno cultural, físico y social y por tanto existen una serie de factores que debemos tener en cuenta en el momento de diseñarlas.

1.1 La disciplina

¿Por qué es tan importante estudiar la interfaz de usuario? Porque tal como hemos dicho antes es una parte muy importante del éxito o fracaso de una aplicación interactiva. Así por ejemplo, según los estudios realizados por MYERS a través de una encuesta hecha a desarrolladores, alrededor de un 48% del código de la aplicación esta dedicado a la interfaz.

No obstante, a pesar de su importancia, el estudio de la interfaz de usuario es una de las disciplinas con menos dedicación en los estudios universitarios de Informática. Por otro lado, otros estudios universitarios están demostrando un interés por el tema, aportando conocimientos, herramientas y técnicas acerca de los diversos aspectos que confluyen en el diseño de la interfaz de usuario.



Más adelante volveremos al tema de las diferentes disciplinas que pueden contribuir y contribuyen a esta disciplina.

1.2 Definición

La ACM, *Association for Computer Machinery*, es, posiblemente, la organización internacional de investigadores y profesionales interesados en todos los aspectos de la computación más importante del mundo. Esta asociación tiene un grupo especial de trabajo en temas de PI denominado SIGCHI, *Special Interest Group in Computer Human Interaction*, el cual propuso la siguiente definición de Programación de Interfaces:

“Es la disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de seres humanos, y con el estudio de los fenómenos más importantes con los que está relacionado”.

El tema principal de esta disciplina está en la interacción y mas específicamente en la interacción entre uno o más seres humanos y uno o más computadoras. Aunque la situación clásica sería una persona usando un programa sentado en su puesto de trabajo, alrededor de las palabras interfaz de usuario hay un amplio conjunto de opciones posibles. Por ejemplo, las computadoras pueden formar parte de monitores de ruta en coches, teléfonos móviles, bloc de notas, libros electrónicos, sistemas de realidad virtual, etc. Por otro lado, los usuarios pueden formar parte de grupos u organizaciones y por tanto necesitaremos de interfaces para sistemas distribuidos o trabajo cooperativo. O también, el sujeto puede querer utilizar su computadora mientras camina hacia una reunión para consultar su agenda, las llamadas telefónicas pendientes y las últimas cotizaciones en bolsa. También, el usuario puede tener la visión o la movilidad disminuida y utiliza un software de lectura de la pantalla o de reconocimiento de la voz para manejar su computadora. Todas estas situaciones podrían ser manejadas por esta disciplina que llamamos Programación de Interfaces.

1.3 Historia de la PI

1.3.1 Preliminares

Toda la Interfaz de Usuario se produce dentro de un contexto social y organizacional. Se requieren distintos tipos de aplicaciones para distintos propósitos.

A finales de los años 1970 y principios de los 80, los psicólogos empiezan a interesarse en los aspectos de procesamiento de la información del diseño de sistemas de información. Empiezan a aparecer áreas de estudio como la de *diseño de menús*, que utilizan tópicos como el de nombres de menús o profundidad versus anchura. A mediados de los 80, la *usabilidad* de los sistemas monousuario empieza a hacerse un hueco, como respuesta a la explosión de las PC. Hacia finales de los 80 y en los 90, la tendencia se centra en las potentes estaciones de trabajo (*workstations*) multiusuario y en las PC más potentes. Se hace hincapié en las comunicaciones más avanzadas, en las aplicaciones multimedia, en las computadoras multitarea y en la realidad virtual. Asimismo, la comunidad PI siempre se ha mostrado de acuerdo en que, aunque es importante entender la interacción de un usuario con una máquina mediante una interfaz, también es necesario que se consideren otros tópicos para aprovechar al máximo las nuevas tecnologías. Estos tópicos que se deben estudiar son: trabajo en grupo, integración e interacción de medios y el impacto de estas tecnologías en el lugar de trabajo, en la casa y en la sociedad en general.

La PI surgió como un campo entrecruzado de disciplinas: gráficos por computadora, sistemas operativos, factores humanos, factores ergonómicos, ingeniería industrial, psicología cognitiva, y por



supuesto la ingeniería informática, conocida en aquellos días en el mundo anglosajón como *Computer Science*.

- **Gráficos por computadora** (Gráficos Interactivos). Nacen con el uso del lápiz (*pen*) y de los monitores de rayos catódicos (*CRT*). Esto lleva a que se desarrollen varias técnicas de la PI. Algunas de estas técnicas, que marcaron el inicio de los gráficos por computadora como disciplina, datan del año 1963, de la tesis de SUTHERLAND “*Sketchpad*”. El trabajo dentro de esta disciplina ha continuado desde aquellos inicios, con el desarrollo de algoritmos y de hardware que permita la manipulación y la representación de objetos más realistas (*CAD/CAM – Computer Aided Design, Computer Aided Manufacturing*). Se desarrolló un conjunto muy importante de componentes de la PI, por ejemplo: el ratón, los *bitmapped displays*, los PC, las ventanas (*windows*), la metáfora del escritorio, y los editores *point & clic*.
- **Sistemas operativos** (Sistemas de Gestión de la Interfaz de Usuario y Kits de Herramientas —*toolkits*— de la Interfaz de Usuario). El trabajo en los sistemas operativos desarrolló técnicas para las interfaces de los dispositivos de E/S, técnicas para el *tunning* del tiempo de respuesta del sistema frente al tiempo de interacción humana, técnicas de multiproceso y técnicas para soportar entornos *windows* y de animación.
- **Ergonomía**. Esta disciplina es similar a la anterior pero más centrada en estudios de trabajo. Se da énfasis a factores relacionados con el estrés como pueden ser: rutina en el trabajo, postura en sentarse o diseño de visión de las pantallas CRT, entre otros.

1.3.2 Los orígenes de la PI

La investigación de PI a lo largo de los años ha sido fructífera. Por ejemplo, la interfaz gráfica utilizada en Microsoft Windows 2000 y XP, está basada en las interfaces de Macintosh, que a su vez están basadas en los trabajos del Xerox PARC, que se basan a su vez en los trabajos del SRI (*Stanford Research Institute*) y del MIT (*Massachusetts Institute of Technology*). O por ejemplo, casi toda la creación actual de software se hace con ayuda de *User Interface Toolkits* y de *Interface Builders* (por ejemplo, *Microsoft Visual Studio*). Cabe decir, que estos conceptos fueron utilizados primero en el mundo universitario.

Por otro lado, incluso el importante crecimiento de la World-Wide Web se debe, en parte, a los avances en investigación en el mundo de la PI, como sucede por ejemplo con la tecnología hipertexto.

A continuación hablaremos de los cambios más significativos en el mundo de la PI:

- Manipulación directa de objetos gráficos
- El ratón (*mouse*)
- Las ventanas (*windows*)
- Programas de dibujo y pintura
- Edición y procesamiento de texto
- Hoja de cálculo
- Hipertexto
- Reconocimiento de gestos

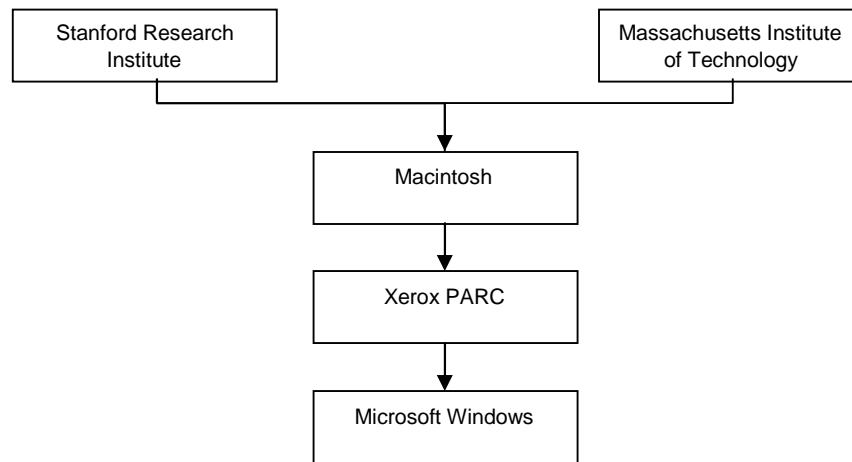


Figura 1 Evolución de los sistemas GUI

1.3.2.1 Manipulación Directa de Objetos Gráficos

- 1963. *IVAN SUTHERLAND*. Tesis doctoral en el MIT: *SketchPad*. El sistema *SketchPad* soportaba la manipulación de objetos gráficos mediante un lápiz óptico, permitiendo tomar los objetos, moverlos y cambiarles el tamaño utilizando algunas restricciones. Fue producido en los laboratorios Lincoln (*Lincoln Labs*) con el soporte de las fuerzas aéreas y la NSF.
- 1966-1967. *WILLIAM NEWMAN*. *William Newman's Reaction Handler*. Este sistema fue creado en el *Imperial College* de Londres y permitía la manipulación directa de gráficos. Introdujo los "*Light Handles*", una forma de potenciómetro gráfico que fue, con toda probabilidad, el primer *widget*.
- 1968. *MIT's Lincoln Labs*. *AMBIT/G*. Este sistema incluyó representaciones de iconos, reconocimiento de gestos, menús dinámicos con la ayuda de dispositivos apuntadores y selección de iconos apuntándolos.
- 1975. *DAVID CANFIELD SMITH*. Tesis doctoral en Stanford: *Pygmalion*. Acuñó por primera vez el término *Icono*.
- Años 70. *Xerox PARC*. Muchas de las técnicas de interacción de interfaces de manipulación directa (por ejemplo, la forma como se selecciona el texto, como se manipulan los objetos, etc.) se hicieron populares a raíz de la investigación llevada a cabo por el laboratorio de *Xerox PARC*. Otro importante hito de aquellas investigaciones fue la idea de *WYSIWYG* (*what you see is what you get*), idea que ha llegado hasta nosotros a través de procesadores de texto como *Microsoft Word*.
- 1977. *ALAN KAY*, *Xerox PARC: Dynabook*. En este artículo, *ALAN KAY* acuñó el término de interfaces de manipulación directa.
- 1981. *Xerox Star*. Sale al mercado el primer sistema comercial que hace extenso el uso de la Manipulación Directa: *Xerox Star*. Le siguen el *Apple Lisa* en 1982 y el *Macintosh* en 1984.
- 1982. *BEN SHNEIDERMAN*, Universidad de Maryland. *BEN SHNEIDERMAN* acuñó el término "Manipulación directa", identificando además los distintos componentes.

1.3.2.2 El ratón

Es el segundo dispositivo de entrada más utilizado. El mouse o ratón es arrastrado a lo largo de una superficie para maniobrar un apuntador en la pantalla del monitor. Fue inventado por Douglas Engelbart y su nombre se deriva por su forma la cual se asemeja a la de un ratón.



Otro tipo de dispositivo similar es el lápiz óptico, la tableta digitalizadora o incluso las pantallas táctiles.

- 1965. *NLS*. El primer ratón se desarrolló en los laboratorios SRI, como parte del proyecto *NLS* (financiado por *ARPA*, *NASA* y *Rome ADC*), como sustituto económico de los lápices ópticos que habían sido usados desde 1954.
- 1968. *DOUG ENGELBART*. Algunos de los usos actuales del ratón fueron demostrados ya en 1968, dentro del proyecto *NLS*, en una película.
- 1970. *Xerox PARC* popularizó el ratón como un dispositivo de entrada práctico.
- 1981. *Xerox Star*. Se comercializa por primera vez como parte del *Xerox Star*. Más tarde aparece en el *Apple Lisa* (1982), y en el *Apple Macintosh* (1984).

1.3.2.3 Las ventanas

- 1968. *ENGELBART*. El uso de múltiples ventanas fue demostrado por *ENGELBART*, dentro del proyecto *NLS*.
- 1969. *ALAN KAY*. Tesis doctoral en Universidad de Utah. *ALAN KAY* propuso la idea de ventanas traslapadas.
- 1974. *Stanford University*. Sistema *COPLOT*.
- 1974. *MIT*. Editor de textos *EMACS*.
- 1974. *Xerox PARC*. Los sistemas *Smalltalk* e *InterLisp*, introdujeron el uso de las ventanas traslapadas.
- 1979. *Lisp Machines Inc.* Uno de los sistemas comerciales que primero introdujo el uso de ventanas fue en *Lisp Machines Inc. (LMI)* y en *Symbolics Lisp Machines*.
- 1981. *Xerox PARC*. El *Cedar Window Manager* fue el gestor de ventanas con más relevancia de la época.
- 1981. *Xerox Star*. *Xerox Star* fue el primer sistema comercial que popularizó las ventanas. En sus primeras versiones no utilizaba ventanas traslapadas. Sólo permitía ventanas principales y no superpuestas.
- 1982. *Apple Lisa*. Junto al *Xerox Star* (1981), el *Apple Lisa* también ayudó a popularizar el uso de ventanas. Este sistema utilizaba ventanas traslapadas.
- 1983. *Carnegie Mellon University's Information Technology Center*. Al anterior gestor de ventanas, le siguió el *Andrew*.
- 1983. *Lee Lorenzen & Dan Meyer*. Primera versión de *GEM* (Graphical Environment Manager) Desktop para Atari.
- 1984. *Apple Macintosh*. El *Apple Macintosh*, junto con sus predecesores (*Xerox Star* en 1981 y *Apple Lisa* en 1982), ayudó a popularizar el uso de las ventanas en las aplicaciones informáticas. Este sistema utilizaba ventanas traslapadas.
- 1984. *MIT*. El sistema de ventanas estándar *X Window* (que utilizan muy a menudo los sistemas *UNIX*), fue desarrollado en 1984 por investigadores del *MIT* [SCH86].
- 1985. *Lee Lorenzen & Dan Meyer*. Segunda versión de *GEM* Desktop, para Atari Amstrad PC1512.
- 1985. *MicroSoft*. Sale la primera versión de *MS-Windows* (1.0) sobre *MSDOS*.
- 1988. *Lee Lorenzen & Dan Meyer*. Tercera –y última– versión de *GEM* Desktop. Disponible para PC y ya disponía de un *Programmers Toolkit* y de un *Editor de Textos*.
- 1992. *MicroSoft*. *MS-Windows for WorkGroups* (3.1).
- 1995. *MicroSoft*. *MS-Windows NT 3.51* y *MS-Windows 95*.



1.3.2.4 Programas de dibujo y pintura

- 1963. *IVAN SUTHERLAND*. Tesis doctoral en el MIT: *SketchPad*. La gran mayoría de la tecnología actual que utilizan los programas gráficos, fueron demostrados en 1963 por SUTHERLAND en el sistema Sketchpad.
- 1965. *NLS*. Uno de los periféricos imprescindibles en el mundo de las aplicaciones de dibujo es el ratón. El uso de este dispositivo de entrada fue demostrado en 1965 por *NLS*.
- 1968. *KEN PULFER & GRANT BECHTHOLD*, *National Research Council of Canada*. En 1968 se construyó un ratón que se utilizó en un sistema de animación de largometrajes para dibujar los fotogramas de una película.
- 1971. *National Film Board of Canada*. El film “*Hunger*” ganó muchos premios después de ser dibujado utilizando una tableta financiada por el *National Film Board of Canada*, en vez de un ratón.
- 1975. *WILLIAM NEWMAN*, *Xerox PARC's Alto*. El *Markup* de WILLIAM NEWMAN fue el primer programa de dibujo seguido de cerca por el *Draw* de PATRICK BAUDELAIRE, que además gestionaba líneas y curvas.
- 1974-1975. *DICK SHOUP*, *Xerox PARC*. El primer programa de computadora para pintar fue el *Superpaint*.
- 1985. *DIGITAL RESEARCH, INC.* GEM Paint Version 2.01. Programa de dibujo de Mapa de Bits.
- 1985. *DIGITAL RESEARCH, INC.* GEM Draw 1.0. Programa de dibujo Vectorial con fuentes de Mapa de Bits. Podía abrir dos ventanas y realizar “drag & drop” entre ellas.

1.3.2.5 Edición y procesamiento de textos

- 1962. *ENGELBART*, *Stanford Research Lab*. Se implementó un procesador de textos con *word wrap* automático, buscar y reemplazar, macros definidas por el usuario, *scroll* y comandos para mover, copiar y borrar caracteres, palabras o bloques de texto.
- 1965. *Stanford University*. *TVEdit* fue uno de los primeros editores basados en CRT que fue ampliamente usado.
- 1967. *Brown University* (financiado por *IBM*). El *Hypertext Editing System*, disponía de editor de pantalla y formateo de cadenas de caracteres de tamaño arbitrario con la ayuda de un lápiz óptico.
- 1967. *MIT*. *TECO* fue uno de los primeros editores de texto de pantalla.
- 1968. *NLS*. Se demuestra la edición con la ayuda del ratón.
- 1974. *MIT*. Editor de textos *EMACS*.
- 1974. *Butler Lampson & Charles Simony*, *Xerox PARC*. El *Bravo* fue el primer editor-formateador WYSIWYG. Los primeros editores comerciales WYSIWYG fueron el *Star*, el *LisaWrite* y el *MacWrite*.
- mediados de los 80s. Varios procesadores de texto fueron populares en entornos PC entre los que cabe destacar el “Word Star”, el “MS-Word” —el precursor del mayormente utilizado hoy en día—.
- 1990. *Ventura Software Inc.* Desarrollan este postprocesador de textos que usaba la técnica de WYSIWYG muy avanzada. Aunque ya empezaron a aparecer versiones en 1986 fue en 1990 cuando se puede utilizar en entorno PC junto con GEM. Actualmente ha derivado a la familia de productos Corel.

1.3.2.6 Hoja de cálculo

- 1977-1978. *FRANKSTON & BRICKLIN*, *MIT & Harvard Business School*. Estos estudiantes realizaron la primera hoja de cálculo de la historia. Se llamaba *VisiCalc* y fue desarrollada para el *Apple II*. La solución fue basada en un algoritmo de



backtracking (*dependency-directed backtracking*) diseñado por SUSSMAN y STALLMAN en el MIT AI Lab.

- 1982. *LOTUS DEVELOPMENT CORPORATION*. Se funda la compañía y introduce la primera versión de Lotus 1-2-3.

1.3.2.7 Hipertexto

- 1945. *VANNEVAR BUSH*. Se introdujo la idea de Hipertexto (*MEMEX* idea).
- 1965. *TED NELSON*. Acuñó la palabra hipertexto.

1.4 Objetivos de la PI

Los objetivos de la PI son desarrollar o mejorar la seguridad, utilidad, efectividad, eficiencia y usabilidad de sistemas que incluyan computadoras. Cuando decimos sistemas no nos referimos tan solo al hardware y al software sino también a todo el entorno.

Para hacer sistemas interactivos hace falta :

1. Comprender los factores tales como psicológicos, ergonómicos, organizativos y sociales, que determinan como trabaja la gente y hace uso de las computadoras y trasladar esta comprensión para
2. Desarrollar herramientas y técnicas que ayuden a los diseñadores a conseguir que los sistemas informáticos sean los idóneos según las actividades a las cuales se quieran aplicar, para
3. Conseguir una interacción eficiente, efectiva y segura, tanto a nivel individual como de grupo.

Es muy importante comprender que los usuarios no han de cambiar radicalmente su manera de ser, sino que los sistemas han de ser diseñados para satisfacer los requisitos del usuario.

1.5 La interfaz de usuario

La interfaz es una superficie de contacto entre dos entidades. En la interacción persona-computadora estas entidades son la persona y la computadora.

NEGROPONTE en su libro *“Being digital”* da una definición muy sencilla: *La interfaz es el sitio donde los bits y las personas se encuentran.*

En la vida cotidiana tenemos muchos ejemplos de interfaz. En el caso de la puerta, la perilla de la puerta es la interfaz entre la puerta y la persona. El volante, el acelerador y otros instrumentos y herramientas son la interfaz entre un auto y el conductor. Es muy importante darse cuenta en un primer nivel de que la interfaz refleja las cualidades físicas de las dos partes de la interacción. La perilla está hecha de un material sólido y está bien pegada a la puerta, la cual por otra parte, como tiene que interaccionar con la mano, esta puesta a la altura de ésta, y tiene la forma que se le adapta. Esta es una idea muy importante en el diseño que puede concretarse en dos conceptos:

1. visibilidad: para poder realizar una acción sobre un objeto ha de ser visible, y
2. comprensión intuitiva, o propiedad de ser evidente la parte del objeto sobre la que hemos de realizar la acción y cómo hacerlo.

Este principio se conoce como *affordance*.



Como hemos visto, una interfaz es una superficie de contacto que refleja las propiedades físicas de los que interactúan, y en la que se tienen que intuir las funciones a realizar y nos da un balance de poder y control.

CHI dice que *“es un lenguaje de entrada para el usuario, un lenguaje de salida para la computadora y un protocolo para la interacción”*.

Obviamente, aparte de la interacción física entre usuario y computadora hemos de añadir un nivel cognitivo referido a que, es necesario que el lado humano comprenda el protocolo de interacción y actúe sobre la interfaz e interprete sus reacciones adecuadamente. Podemos decir por tanto que:

“La interfaz de usuario de un sistema consiste de aquellos aspectos del sistema con los que el usuario entra en contacto, físicamente, perceptivamente o conceptualmente. Los aspectos del sistema que están escondidos para el usuario se denominan la implementación”.

GERRIT VAN DER VEER nos define la interfaz como *“el conocimiento que los usuarios pueden y deberían tener para poder utilizar satisfactoriamente el sistema”*.

Esta interfaz debería ser visible y de comprensión intuitiva para estar de acuerdo con los principios de diseño anteriormente esbozados.

La interfaz de usuario es el principal punto de contacto entre el usuario y la computadora; es la parte del sistema que el usuario ve, oye, toca y con la que se comunica. El usuario interacciona con la computadora para poder realizar una tarea.

Dependiendo de la experiencia del usuario con la interfaz, el sistema puede tener éxito o fallar en ayudar al usuario a realizar la tarea. El tipo de problemas que origina una interfaz de usuario pobre incluye la reducción de la productividad, un tiempo de aprendizaje inaceptable y niveles de errores inaceptables que producen frustración y probablemente el desechar el sistema.

En la interfaz hemos de tener en cuenta también cómo está sentado el usuario, cómo es la organización de la que forma parte el usuario, el ámbito cultural, etc. Añadiremos por tanto que en la interfaz también se ha de tener en cuenta el entorno y el ámbito cultural.

En las siguientes imágenes podemos ver diferentes ejemplos de interfaces de dispositivos interactivos.



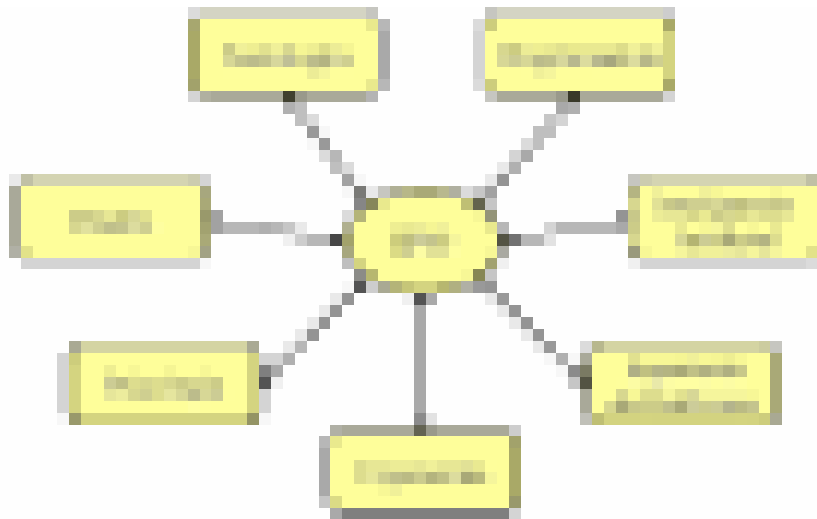
Nuestra idea de computadora hasta hace poco era la de un dispositivo con el que interaccionamos a través de una pantalla con un teclado y un ratón, sentados en una silla. Esta es una situación que ha



cambiado radicalmente. La computadora se ha incrustado en todas nuestras actividades cotidianas y se encuentra en muchas formas, teléfonos móviles, cajero en un supermercado, balanza en la farmacia, en una cámara fotográfica, en dispositivos médicos y, en situaciones especiales como en la realidad virtual, en el futuro llegará a estar en perfecta simbiosis con nuestro cuerpo.

Esta es una reflexión fundamental en el momento de acercarse a una disciplina como la PI, para abordarla con una perspectiva suficientemente amplia respecto a cuales van a ser el tipo de interfaces que vamos a encontrar.

Para poder diseñar interfaces, además del aspecto informático, hace falta tener en cuenta otras disciplinas. Necesitamos trabajar los aspectos psicológicos del usuario, la ergonomía del equipamiento, los aspectos sociales, temas de diseño, etc. Como vemos, hemos de tener en cuenta muchas disciplinas para comprender toda la problemática que supone el desarrollo de interfaces. Esto supone que, a menudo, tengamos que pensar en un equipo interdisciplinario para el desarrollo de sistemas interactivos, cuando el problema a tratar es de la suficiente envergadura. En otros casos, no obstante, cuando el equipo de trabajo sea reducido, las mismas personas deberán jugar los diferentes papeles y tratar de hacerlo bien por sí mismos.



En la figura anterior se presentan un esquema en que aparecen las principales disciplinas que pueden contribuir a los diferentes aspectos en el diseño de interfaces:

1.5.1 Psicología

La psicología es la ciencia que estudia el comportamiento y los estados de la conciencia de la persona humana, considerada individualmente o bien al mismo tiempo como miembro de un grupo social.

Cabe distinguir entre:

1. *Psicología cognitiva*. Trata de comprender el comportamiento humano y los procesos mentales que comporta.
2. *Psicología social*. Trata de estudiar el origen y las causas del comportamiento humano en un contexto social.

La psicología contribuye a la PI mediante conocimientos y teorías acerca de cómo los sujetos se comportan, procesan la información y actúan en grupos y organizaciones, pero también proporciona metodologías y herramientas para evaluar y determinar el grado de satisfacción de éstos a nuestros



diseños. De este modo, la psicología proporciona una manera de comprobar que nuestras interfaces son tan efectivas como deseamos.

1.5.2 Diseño

Actividad encaminada a conseguir la producción en serie de objetos útiles y bellos. Tal como se entiende actualmente, pretende actuar sobre el entorno físico del hombre con tal de mejorarlo en su conjunto. Esta es una disciplina muy importante dentro de la PI, tal como podemos ver en su definición para conseguir programas usables.

1.5.3 Etnografía - Sociología

Es la ciencia que estudia las costumbres y las tradiciones de los pueblos. En los últimos años, algunas de las mayores compañías americanas están reclutando antropólogos para comprender mejor a sus clientes y a sus trabajadores y para diseñar productos que reflejen mejor las tendencias culturales emergentes. Estas compañías están convencidas que las herramientas de investigación etnográfica — observación detallada, entrevistas sutiles, documentación sistemática—, pueden responder a aquellas cuestiones sobre organizaciones y mercados que otros métodos no pueden.

1.5.4 Ergonomía o factores humanos

Su propósito es definir y diseñar herramientas y artefactos para diferentes tipos de ambiente: trabajo, descanso y doméstico. Su objetivo es maximizar la seguridad, la eficiencia y la fiabilidad para simplificar las tareas e incrementar la sensación de confort y satisfacción. Las radiaciones de las pantallas, por ejemplo han sido una tema de trabajo importante en los últimos años.

La ergonomía o factores humanos se centra en el estudio de las características físicas de la interacción, por ejemplo el entorno físico donde se produce esta y la forma y las peculiaridades físicas de la pantalla.

Describir completamente la ergonomía podría ocupar casi un curso completo, tanto como para hacer una disciplina aparte.

Algunos de los aspectos que se podrían considerar son los siguientes:

1. *Organización de los controles y pantallas.* Los aspectos físicos de como están organizados los controles y las pantallas son importantes. En general han de permitir una acción rápida al usuario. El usuario ha de poder acceder fácilmente a todos los controles y ver toda la información que se le presenta sin mover excesivamente el cuerpo.
2. *La información más importante ha de estar situada a la altura de los ojos.* Los controles se han de disponer espaciados con tal de que el usuario se acomode perfectamente. En cuanto a la iluminación, hace falta prevenir los reflejos.
3. *El entorno físico de la interacción.* La ergonomía también estudia el entorno físico donde se realiza la interacción.
4. *Aspectos de salud.* Aunque el uso de las computadoras no es una actividad arriesgada, es preciso tener en cuenta determinados aspectos de salud que afectan directamente la calidad de la interacción y las prestaciones del usuario, la posición física, la temperatura, la luz, el ruido, el tiempo de permanencia delante de la computadora, etc. Actualmente las grandes compañías ya están teniendo en cuenta los aspectos de prevención de riesgos laborales en el mundo de la informática.
5. *El uso de los colores.* Los colores utilizados para visualizar han de ser tan diferentes como sea posible y su distinción no se debe ver afectada por los cambios de contraste.



1.5.5 Programación

Una computadora por si sola no es capaz de hacer nada, sino que solo puede hacer lo que se la ha indicado, o mejor dicho, programado que haga. La programación es la herramienta que nos permite “decirle al computadora lo que debe hacer”. Todas las acciones que una computadora realiza cuando un usuario interactúa con ella son respuestas programadas por un programador.

El programador de computadoras es el encargado de diseñar soluciones a problemas y de escribirlos como programas de computadora.

Un programa:

- debe funcionar
- no debe tener dificultades
- debe estar bien documentado
- debe ser eficiente

Dentro de la programación podemos englobar tanto los propios lenguajes de programación (Java, C, C++, Pascal, Fortran, BASIC, etc.) como todas las herramientas de diseño y programación que permiten el desarrollo de las interfaces.

Básicamente podemos clasificar los tipos de programación en cinco grandes grupos:

1. **ORIENTADA a OBJETOS.** La programación Orientada a Objetos (POO) esta basada en los conceptos de Clases y Objetos.
Clases: Una Clase es una colección de Datos y Métodos que operan sobre los datos. Los datos y métodos tomados juntos sirven para definir contenido y capacidades de algunos Objetos.
Objetos: Un Objeto es una instancia de una Clase.
Es la programación mayormente utilizada hoy en día.
Ejemplos: C++, Java
2. **IMPERATIVA** (Entrada, procesamiento y salidas de Datos): Es llamada así porque esta basada en comandos que actualizan variables que están en almacenamiento. Es en 1950 cuando los programadores reconocen que las variables y comandos de asignación constituyen una simple pero útil abstracción de memoria que se actualiza.
Debido a su relación cerrada con las arquitecturas de las máquinas, los lenguajes de Programación Imperativa pueden ser implementados eficientemente. La razón fundamental de la Programación Imperativa está relacionada a la naturaleza y propósito de la programación.
Su principal característica es que están *orientados a la arquitectura del computador*.
Ejemplos: PASCAL-ADA
3. **FUNCIONAL:** La principal característica de este tipo de programación reside en que “los datos son funciones y los resultados pueden ser un valor o una función”. En programación Funcional la aplicación de valores de entrada para valores de salida es alcanzado directamente.
El programa en sí es una Función (o un grupo de funciones). Usualmente compuesto de Funciones mas simples.
La relación entre las funciones son muy simples: Una Función puede llamar a otra Función, o el resultado de una Función puede ser usado como el argumento de otra Función. Los programas son escritos enteramente dentro del lenguaje de expresiones, funciones y declaraciones.
La Programación Funcional se caracteriza por el uso de Expresiones y Funciones mientras que la Imperativa por el uso de variables, comandos y procedimientos.



Ejemplos: LISP, ML, HOPE

4. **DECLARATIVA:** La programación Declarativa está basada en la noción de Relación debido a que la relación es el concepto más general de una Aplicación. Los lenguajes declarativos están basados en el modo de pensar de los humanos en lugar de la computadora.

Principales ventajas que aportan:

- Elegancia, claridad, sencillez, potencia y concisión.
- Semánticas claras, simples y matemáticamente bien fundadas.
- Cercanos al nivel de abstracción de las especificaciones formales/informales de los problemas a resolver.
- Referencialmente transparentes: Comportamiento matemático adecuado que permite razonar sobre los programas.
- Soportan técnicas muy avanzadas de desarrollo, mantenimiento y validación de programas.
- Altas dosis de paralelismo implícito.
- Aplicaciones variadas y de gran interés.

Ejemplos: PROLOG, HASKEL, MODULA-2.

5. **CONCURRENTE:** Este Tipo de programación esta aún en proceso de investigación. La Programación Concurrente es una disciplina relativamente nueva y, por ello, va tomando muchas aproximaciones diferentes según va avanzando su investigación.

La Programación Concurrente debe frecuentemente estar acompañada por el avance del Hardware y sus arquitecturas correspondientes.

Ejemplos: ADA es un lenguaje familiar y tiene una aproximación a la Concurrencia.

1.5.6 Inteligencia artificial

La inteligencia artificial trata de diseñar programas de computadora inteligentes que simulen diferentes aspectos del comportamiento humano inteligente.

La inteligencia artificial ha sido utilizada en el diseño de tutores y sistemas expertos en interfaces inteligentes y en el diseño de interfaces de lenguaje natural utilizando la voz. El desarrollo de agentes inteligentes para ayudar en la navegación a los usuarios y reducir las tareas más normales es un tema nuevo en los desarrollos multimedia.

1.5.7 Ingeniería de software

Es importante tener en cuenta la ingeniería de software en el desarrollo de un sistema interactivo. Esta disciplina estudia técnicas de diseño y desarrollo del software. Para realizar diferentes aplicaciones se deben utilizar procedimientos propios de ingeniería. Sólo con estos procedimientos y técnicas vamos a obtener un software de calidad.

1.6 Usabilidad

Para que un sistema interactivo cumpla sus objetivos tiene que ser *usable* y, además, debido a la generalización del uso de las computadoras, *accesible* a la mayor parte de la población humana.

La utilidad de un sistema, como medio para conseguir un objetivo, tiene una componente de funcionalidad (la llamada utilidad funcional) y otra basada en el modo en que los usuarios pueden usar dicha funcionalidad. Es esta última la que nos interesa ahora.



Podemos definir la usabilidad como la medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

1.6.1 ¿Por qué es importante la usabilidad?

El establecimiento de unos principios de diseño en ingeniería basados en la usabilidad han tenido como consecuencia probada:

- reducción de los costos de producción
- reducción de los costos de mantenimiento y apoyo
- reducción de los costos de uso
- mejora en la calidad del producto

Aun así cabría preguntarse: ¿por qué nos tenemos que preocupar?, ¿porqué las cosas son difíciles de utilizar?

El problema radica en el desarrollo del producto, en el énfasis de la tecnología, en vez del usuario, la persona para la cual esta hecho el dispositivo.

La Interfaz de Usuario, es la puerta del usuario a la funcionalidad del sistema subyacente. Que las interfaces de usuario estén mal diseñadas es un factor que frena el uso de las funcionalidades. Por tanto, es muy importante diseñar interfaces de usuario usables.

Podemos entender la usabilidad como “*aquella característica que hace que el software sea fácil de utilizar y fácil de aprender*”. Un software es fácil de utilizar si realiza la tarea para la que lo estamos usando de una manera fácil, eficiente e intuitiva. La facilidad de aprendizaje se puede medir por lo rápido que realizamos una tarea, cuantos errores se cometen y la satisfacción de la gente que lo utiliza.

También incluye aspectos como que sea seguro, útil y que tenga un costo adecuado.

Una aplicación usable es la que permite que el usuario se concentre en su tarea y no en la aplicación.

Las interfaces se ponen en el medio. No quiero concentrar mis energías en la interfaz, me quiero concentrar en mi trabajo.

1.6.2 Comentarios habituales que se hacen respecto de la no necesidad de la usabilidad

A pesar de lo expuesto anteriormente, siguen apareciendo críticas respecto la usabilidad:

- Los usuarios no necesitan mejores interfaces sino un entrenamiento mejor.
- La usabilidad es subjetiva —no se puede medir.
- El diseño de la interfaz de usuario está implícito en el diseño del software. No se tiene que planificar expresamente y valorar su costo.
- Si el diseñador esta familiarizado con guías de estilo y principios de diseño, se harán buenas interfaces.
- El diseño de la interfaz de usuario no es necesario hasta el diseño detallado.
- La usabilidad aumenta los costos de desarrollo y el tamaño del ciclo de desarrollo.

1.6.3 Principios generales de la usabilidad



Los principios generales que se pueden aplicar a un sistema interactivo para mejorar la usabilidad son:

1.6.3.1 Facilidad de aprendizaje.

- Que sea mínimo el tiempo necesario que se requiere desde el no conocimiento de una aplicación a su uso productivo.
- Reducir el tiempo necesario para ayudar en las sesiones de aprendizaje.
- Proporcionar ayuda a usuarios intermedios. Esta característica permite que los usuarios novatos comprendan como utilizar inicialmente un sistema interactivo y, a partir de esta utilización, llegar a un nivel de conocimiento y uso del sistema máximo.

Para que el sistema sea fácil de aprender, éste debe ser:

- *Sintetizable*. El usuario tiene que poder evaluar el efecto de operaciones anteriores en el estado actual. Es decir, cuando una operación cambia algún aspecto del estado anterior, es importante que el cambio sea captado por el usuario.
- *Familiar*. Los nuevos usuarios de un sistema poseen una amplia experiencia interactiva con otros sistemas. Esta experiencia se obtiene mediante la interacción en el mundo real y la interacción con otros sistemas informáticos. La familiaridad de un sistema es la correlación que existe entre los conocimientos que posee el usuario y los conocimientos requeridos para la interacción en un sistema nuevo.

1.6.3.2 Consistencia.

Este es un concepto clave en la usabilidad de un sistema informático. Un sistema es consistente si todos los mecanismos que se utilizan son siempre usados de la misma manera, siempre que se utilicen y sea cual sea el momento en que se haga.

El problema que se nos plantea es que, ya recién acabada una aplicación, el mercado ya nos presiona para hacer una nueva, con nuevas metáforas, objetos y comportamiento. ¿Es posible, en estos casos, tener una consistencia completa? La verdad es que es difícil, pero hemos de intentar mantener la máxima consistencia entre versiones o como mínimo, que sea lo máximo consistente posible y que el usuario pueda utilizar nuevas posibilidades interactivas sin perder las que ya conocía.

Recomendaciones para diseñar sistemas consistentes:

- Seguir guías de estilo siempre que sea posible.
- Diseñar con un “look & feel” común.
- No hacer modificaciones si no es necesario hacerlas.
- Añadir nuevas técnicas al conjunto preexistente, en vez de cambiar las ya conocidas.

1.6.3.3 Flexibilidad.

La flexibilidad se refiere a la multiplicidad de maneras en que el usuario y el sistema intercambian información.

Los parámetros que miden la flexibilidad son:

A- Control del usuario.

- Permite a los usuarios conducir la interacción.
- Permite a los usuarios no estar forzados a trabajar para la aplicación.
- Hace a los usuarios inteligentes para que resulte obvio cómo proceder.

Como dar control al usuario:

- Dar a los usuarios la posibilidad de poder deshacer.



- Dar a los usuarios control para empezar y acabar las operaciones siempre que sea posible. Cuando el proceso no se pueda interrumpir, advertir a los usuarios y visualizar mensajes apropiados durante el proceso.

B- Migración de tareas.

La migración de tareas está relacionada con la transferencia del control entre el usuario y el sistema. Tanto el usuario como el sistema han de poder pasar de una tarea a la otra o promocionarla, de manera que pueda ser completamente interna o compartida entre los dos.

Por ejemplo, el corrector ortográfico es una tarea que puede ser realizada por el usuario, automatizada por el sistema o compartida entre los dos.

C- Capacidad de substitución.

La capacidad de substitución permite que valores equivalentes puedan ser substituidos los unos por los otros.

Por ejemplo, si queremos introducir el valor que determina el margen de una carta, se puede preferir ingresarlo en centímetros o en pulgadas, explícitamente o como resultado de calcularlo según los valores que previamente hemos ingresado, tal como la anchura de la carta u otros o, finalmente, decidirlo visualmente.

Eliminando cálculos innecesarios al usuario, se pueden minimizar errores y esfuerzo cognitivo.

D- Adaptabilidad.

Adaptabilidad es la adecuación automática de la interfaz al sistema. Las decisiones para poder hacerlo pueden estar basadas en la experiencia del usuario o en la observación de la repetición de ciertas secuencias de tareas. Se puede preparar a un sistema para reconocer el comportamiento de un experto o de un usuario novel y, de acuerdo con esto, ajustar automáticamente el control del diálogo o el sistema de ayuda con tal de adaptarlo a las necesidades del usuario actual.

Existen técnicas para obtener adaptabilidad, como

- Permitir a los usuarios suspender una acción y comenzar otra para atender un trabajo inesperado.
- Tener atajos y soporte de navegación por teclado para no tener que utilizar siempre el ratón.

1.6.3.4 Robustez.

La robustez de una interacción cubre las características para poder cumplir sus objetivos y su asesoramiento.

1.6.3.5 Recuperabilidad.

Grado de facilidad que una aplicación permite al usuario para corregir una acción una vez está reconocido un error.

1.6.3.6 Tiempo de respuesta.

Se define generalmente como el tiempo que necesita el sistema para expresar los cambios de estado del usuario. Es importante que los tiempos de respuesta sean soportables para el usuario.

1.6.3.7 Adecuación de las tareas.

Grado en que los servicios del sistema soportan todas las tareas que el usuario quiere hacer y la manera en que éstas las comprenden.

1.6.3.8 Disminución de la carga cognitiva.

Esto significa que:

- Los usuarios tienen que confiar mas en los reconocimientos que en los recuerdos.
- Los usuarios no tienen que recordar abreviaciones y códigos muy complicados.



1.7 El diseño centrado en el usuario

Tal como destacan diversos autores, el diseño de sistemas interactivos implica realizar un diseño pensando en el usuario, centrando nuestro sistema de desarrollo en él e implicarlo tanto como sea posible, hasta pensar en incluir usuarios en el equipo de diseño. Actualmente es fácil encontrar implantaciones de sistemas de información en que los usuarios están totalmente integrados dentro del equipo de diseño. Por ejemplo, los ya muy conocidos ERP, cuando se adaptan a una empresa suelen necesitar de la interacción con los usuarios finales de la aplicación. Muchas veces está presente la negativa de algunos empleados de algunas empresas en colaborar. Los estudios actuales informan que esta poca predisposición a ayudar en su implantación es debida a la poca adaptabilidad del ser humano a los cambios drásticos. Así pues, el usuario final de la aplicación, ya sea de modo consciente o inconsciente, tiene miedo al nuevo sistema, a las nuevas reestructuraciones del departamento que se pueden llegar a realizar, a que el sistema reduzca el trabajo a realizar por los empleados y por tanto sean necesarios los despidos o las jubilaciones anticipadas, etc. Por tanto es necesario tener en cuenta estas negativas con las que nos podemos encontrar.



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

Licenciatura En Informática

Programación de Interfaces

*Autor: L.I. María de Lourdes Isabel
Ponce Vásquez*

AGOSTO-DICIEMBRE 2005



Contenido

UNIDAD 2. El Factor Humano.....	3
Objetivos:	3
Introducción.....	3
2.1 Cognición	3
2.2 Los Sentidos.....	3
2.2.1 La visión.....	4
2.2.1.1 Agudeza visual y campo visual	5
2.2.1.2 Color.....	6
2.2.1.3 Matiz.....	6
2.2.1.4 Luminosidad	6
2.2.1.5 Saturación	7
2.2.1.6 Otras consideraciones	7
2.2.1.7 Consideraciones de diseño	7
2.2.2 El oído.....	9
2.2.2.1 Tono y Sonoridad	10
2.2.3 Tacto.....	11
2.2.4 Gusto	11
2.2.5 Olfato	11
2.3 Capacidades mentales de los usuarios	12
2.3.1. Un Vistazo a la Memoria	12
2.3.1.1 Memoria de trabajo o memoria de corto plazo	13
2.3.1.2 Memoria a largo plazo	13
2.3.1.3 Memoria Sensorial.....	14
2.3.1.4 Memoria de episodios y memoria semántica	14
2.4 La Memoria en Acción.....	15
2.5 Experimentos de Memoria.....	15
2.5.1 El hilado de dígitos.....	15
2.5.2 Agrupar información (Chunking).....	16
2.5.3 Efectos primacía y reciente	17
2.5.4 Conclusión	18
2.5.5 Maximizar la memoria	18
2.6 Guía para crear interfaces	19
2.6.1 Consistencia	19
2.6.2 Sugerencias visuales y auditivas.....	20
2.7 Memoria y Aprendizaje	20
2.8 Sistemas de interacción Humano-computadora.	20



UNIDAD 2. El Factor Humano

Objetivos:

- Profundizar en las capacidades del usuario y sugerir formas de diseñar sistemas de cómputo que puedan maximizar estas capacidades
- Examinar las capacidades mentales de los usuarios, en especial la memoria y sugerir algunas guías de desarrollo de sistemas.

Introducción

Los seres humanos estamos sujetos a pérdidas de concentración, cambios en el carácter, motivación y emociones. También tenemos prejuicios y miedos, y cometemos errores y faltas de juicio.

Al mismo tiempo podemos protagonizar hechos remarcables, percibir y responder rápidamente a estímulos, resolver problemas complejos, crear obras de arte y coordinar acciones con otros en una orquesta, hacer volar aviones o rodar películas.

En el pasado los diseñadores de sistemas informáticos no habían dado importancia al elemento humano porque se suponía que sin mucho esfuerzo, los usuarios podían aprender y hacer uso de los sistemas y las aplicaciones desarrolladas. No obstante y como probablemente todos conocemos por la experiencia, el uso de los sistemas es muchas veces difícil, complicado y frustrante.

Por eso es importante dedicar un poco de tiempo a intentar comprender los aspectos humanos de la informática y dentro de los mismos el sistema cognitivo, ya que puede ser muy útil para definir modelos de interfaces que se adapten más fácilmente a los modelos cognitivos del ser humano.

2.1 Cognición

El término cognición se usa para describir la interpretación de la información desde el mundo exterior que es recibido a través de los sentidos. Es aquello que hace posible la percepción de los objetos y eventos e interpretarlos. Esta interpretación es extremadamente importante para los seres humanos. Ver un objeto y escuchar los sonidos que emite no es suficiente, esta información debe ser interpretada minuciosamente y actuar en relación. La interpretación incorrecta de la información podría ser fatal.

Entre los sistemas cognitivos humanos hay dos que son de una especial relevancia en la PI y que constituyen cada uno de ellos niveles independientes. Uno es el sistema perceptivo y otro es el sistema sensorio motor.

En cuanto al sistema Perceptivo cabe decir que es el que hace referencia a aspectos tan importantes como las características de las interfaces visuales y las auditivas y las interfaces de manipulación directa. Por su parte, los sistemas de entrada-salida de las interfaces deben ser estudiados en relación con los sistemas sensoriales y motores del ser humano.

2.2 Los Sentidos

La interacción persona-computadora ocurre cuando hay un intercambio de información entre ambos. La computadora presenta cierta información en un formato físico determinado y la persona debe captarla a través de sus sentidos para después procesarla. De la misma manera, la persona transmite cierta información a la computadora quien la capta a través de sus sistemas de entrada.

Cuando nos ocupamos de la transmisión de información de la computadora a la persona tendremos que considerar el funcionamiento de los sistemas sensoriales humanos.



La información acerca del mundo es recibida mediante los cinco sentidos antes de que pueda ser interpretada. Estos sentidos se clasifican en:

- Vista
- Oído
- Gusto
- Olfato
- Tacto

Sin embargo, para la mayoría de los seres humanos, aquellos que no tienen serios defectos de visión, el sentido más importante es el de la vista –la habilidad de ver. Si se le pide a alguien que describa un objeto poco familiar, generalmente lo describirá primero en términos de su apariencia física y después en algunos casos describirá otras características como el sabor o la textura. Cuando queremos saber dónde está un objeto perdido, preguntamos si alguien lo ha visto. Esta referencia de haberlo visto, también implica conocer su localización. Tan importante es la vista para nosotros que nuestro lenguaje se maneja a su alrededor. La apariencia de los objetos es muy importante para nosotros. Cuando se habla de algo que nos gusta estéticamente, generalmente significa que se ve agradable, no que sabe o que tiene textura agradable. Si quisiéramos indicar eso, alargaríamos generalmente la oración.

Generalmente, las interfaces presentan la información en un formato visual y también, aunque menos frecuentemente, en formato auditivo. Por esta razón, el sentido visual ha sido el más tratado en PI, seguido a distancia por el sentido auditivo.

Sin embargo, recientemente, se están empezando a diseñar interfaces donde la información es presentada en otros formatos físicos para que el usuario la reciba a través de sus otros sentidos. Por ejemplo, actualmente se están investigando las posibilidades que ofrecen las interfaces olfativas, lo que nos obligará a considerar también el funcionamiento de estos canales sensoriales.

2.2.1 La visión

Cuando la luz entra en el ojo tiene dos características principales: brillo (intensidad) y color (longitud de onda). El rango de objetos vistos por el ojo es un resultado directo de estos dos fuentes de información.

Como sabemos, el ojo es como una cámara. La luz entra en el ojo y es recibida por células sensibles en el fondo del ojo que forman la retina. La retina es análoga a la cinta en la cámara. Los lentes del ojo, como los de una cámara, acomodan la luz con forme pasa a través de ellos, mediante su curvatura, para hacer la imagen tan clara como sea posible. Este enfoque se realiza mediante una serie de músculos (ciliares) en el ojo. Los lentes cambian de forma de acuerdo a la proximidad del objeto. Con forme envejecemos, la habilidad de los lentes de cambiar de forma disminuye.

La retina consiste de una serie de capas de células nerviosas incluyendo una capa receptora compuesta de conos y bastones que al contacto con la luz segregan sustancias químicas que estimulan a las neuronas. Existen 130 millones de bastones y 6 o 7 millones de conos. Los bastones son capaces de operar en condiciones de poca luz pero son incapaces de detectar el color; los conos requieren buena luz para poder operar y pueden detectar el color. Esto significa, por ejemplo que, en la oscuridad, los seres humanos son capaces de ver formas pero no son capaces de detectar su color. Los conos se concentran al frente de la retina y el área llamada fovea y los bastones a los lados. Esto implica que la detección del color es buena cuando las imágenes se colocan frente al ojo pero no si un objeto se coloca al lado de una persona. Esto puede ser importante en el desarrollo de sistemas que involucran grandes pantallas, o muchas pantallas o controles que se colocan en la visión periférica.

La retina responde a la luz enfocándola al producir un impulso eléctrico que se pasa al cerebro para ser interpretado. Los receptores en la retina estimulan las células bipolares y éstas estimulan las células ganglionares. Los mensajes son finalmente enviados desde el ojo a través del nervio óptico.

Esto es una simplificación del proceso. Es importante recordar que la visión no es simplemente un producto del ojo sino que involucra también una interpretación por parte del cerebro. Esta interpretación es la responsable de algunas ilusiones ópticas. Obviamente, en el desarrollo de sistemas de cómputo, es importante evitar crear



cualquier efecto confuso. Esto puede ocurrir con cierto tipo de iconos tridimensionales, por ejemplo, o puede ser causado por el abuso del color. Por ejemplo, una línea recta puede aparecer distorsionada si se presenta con un fondo de líneas curvas o radiantes.

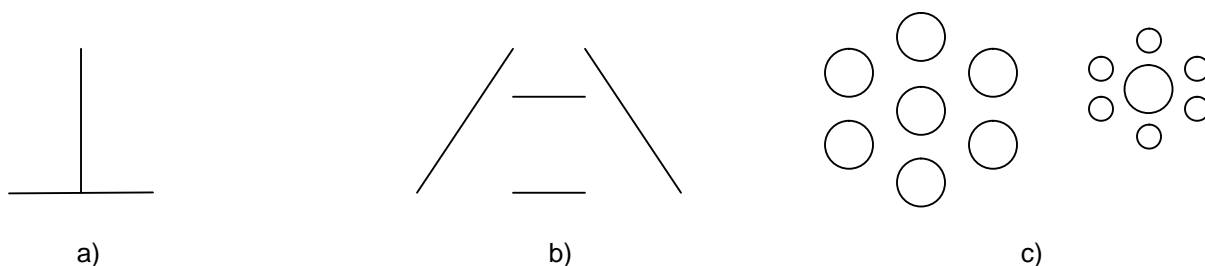


Figura 1. Ilusiones ópticas.

a) ambas líneas son iguales, b) las líneas horizontales son iguales, c) los dos círculos centrales son iguales

2.2.1.1 Agudeza visual y campo visual

La habilidad visual humana debe permitir a una persona distinguir entre proyecciones separadas en la retina. La gente debe ser capaz de hacer esto claramente, sin ver un objeto borroso. Esta habilidad es llamada agudeza visual. En condiciones de luz de día normales, la agudeza visual es mejor al frente del ojo –en la fovea. En este punto, los receptores son más densos. Para examinar un objeto, los ojos se moverán hasta que el objeto se enfoque al frente de la retina. El la periferia de la visión, una persona puede estar conciente de la presencia de un objeto sin reconocer qué es. Para ver el objeto es necesario voltear la cabeza para que el frente de los ojos pueda enfocar ese objeto.

El campo visual es el rango discernible en grados para el promedio de los seres humanos. Este varía de acuerdo a si la cabeza permanece o no estacionaria. Si los ojos o la cabeza se mantienen en la misma posición esto causará severas restricciones al usuario de un sistema de cómputo interactivo. Generalmente, se puede realizar movimiento y esto proporciona alrededor de 100-120 grados desde la posición de la cabeza. El campo visual es importante para la definición del tamaño de una pantalla de despliegue, el formato de despliegues y de cualquier control. Demasiado movimiento de la cabeza puede causar incomodidad y puede incrementar la fatiga ocular si se requiere reenfocar los objetos.

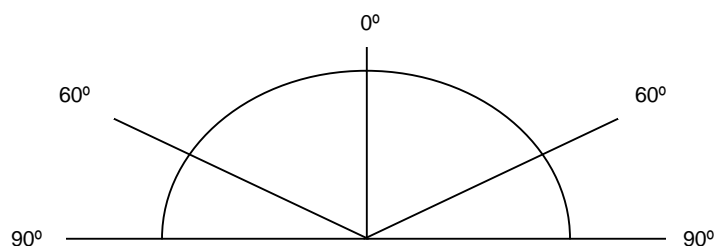


Figura 2. El campo visual

Sin embargo, esto no significa que el usuario de un sistema de cómputo debe sentarse sin movimiento todo el día. Esto podría causar fatiga en el cuerpo al menos. Los sistemas necesitan ser diseñados de modo que el se requiera el mínimo movimiento de cabeza y ojos pero el usuario no debe permanecer sin movimiento por largos periodos de tiempo. Los sistemas de cómputo deben construirse para minimizar la incomodidad, pero el patrón de trabajo del individuo debe incluir movimiento y cambios de actividad. Los optometristas han reportado casos de usuarios jóvenes que sufren ceguera temporal por trabajar todo el día frente a un monitor. En el diseño de sistemas de interacción hombre-computadora es importante comprender que provocar que el usuario enfoque en el mismo punto todo el día sin permitir que el ojo se ajuste a la distancia, está creando un ambiente que puede afectar el bienestar del usuario.



2.2.1.2 Color

La luz visible es una pequeña parte del espectro electromagnético. Nuestro sistema visual es capaz de responder a longitudes de onda en la región de 400 a 750 nanómetros (1 nanómetro = 1 millonésima de milímetro). Esto es llamado el espectro visual. En 400 nm el sistema visual ve violeta y en 750 nm da la sensación de rojo. El color se clasifica de acuerdo al matiz, brillo y saturación.

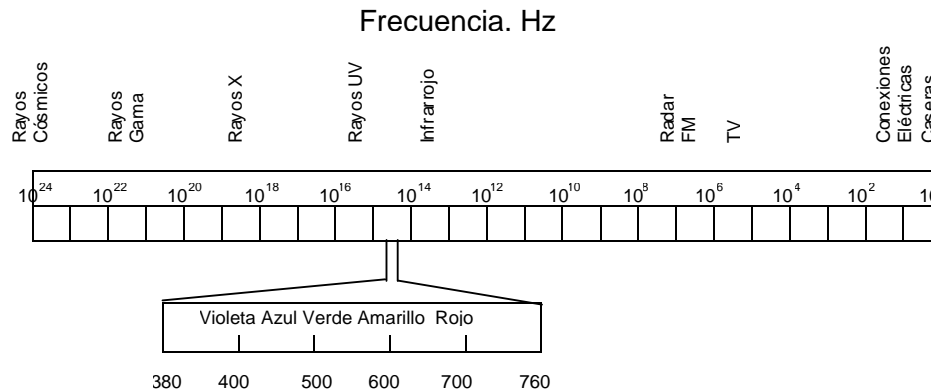


Figura 3. El espectro electromagnético. Las luces visibles se encuentran en el rango entre 400 y 750 nanómetros

2.2.1.3 Matiz

El matiz es muy similar a lo que normalmente pasa cuando la palabra “color” es usada en una conversación. El matiz es un aspecto de los colores cromáticos (rojo y verde). Los colores no cromáticos (negro, blanco y gris neutral) no tienen matiz. El matiz varía de acuerdo a la longitud de onda. En alrededor de 465 nm se experimenta azul puro (sin rojo o verde). El verde puro (sin azul o amarillo) se experimenta en alrededor de 500 nm. El amarillo puro (sin verde o rojo) ocupa el rango alrededor de 570nm. Finalmente, el rojo puro (sin azul o amarillo) se produce por una mezcla de longitudes de onda. Es llamado extraespectral porque no se produce con una sola longitud de onda como en el caso de los otros colores mencionados.

Obviamente, la experiencia de estos colores varía de acuerdo a cómo los perciba un individuo.

2.2.1.4 Luminosidad

La luminosidad es la respuesta subjetiva a la luz, no tiene significado real la medición de niveles absolutos de luminosidad. Nuestra percepción de luminosidad es subjetiva y depende de nuestras preferencias individuales.

Donde las áreas oscuras y claras se juntan es posible ver efectos extraños –ilusiones ópticas. Por ejemplo, en la “Rejilla de Hermann” (Figura. 4) la mayoría de la gente ve puntos negros donde ocurre la intersección de las líneas blancas y puntos blancos donde la intersección de las líneas negras ocurre.

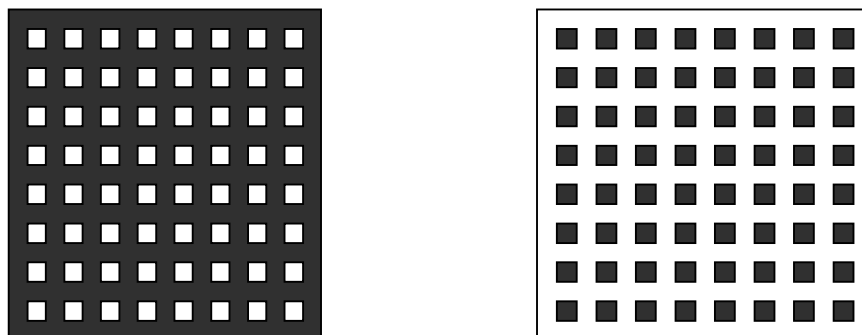


Figura 4. Rejilla de Hermann



2.2.1.5 Saturación

La saturación se refiere a la extensión donde el color es cromático y no acromático. Entre más negro, blanco o gris se mezcle con el color entonces menor saturación habrá. Cuando el color es totalmente gris, la saturación se dice que está en cero.

2.2.1.6 Otras consideraciones

El **contraste** es la relación entre la luz emitida desde un objeto y la luz emitida desde el fondo. La figura 5 muestra el efecto de cuatro diferentes fondos con un cuadro gris idéntico sobre ellos. Los cuatro cuadros grises aparecen más brillantes u oscuros de acuerdo al fondo. Entre más claro es el fondo, mas oscuro se ve el cuadro gris.

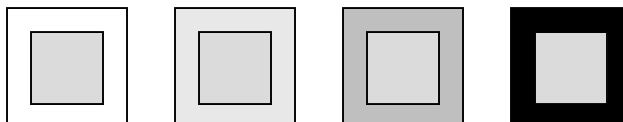


Figura 5. Contrastes

La **iluminación** es la luz reflejada desde la superficie de un objeto y se mide en candelas por metro cuadrado. Entre mayor sea la iluminación de un objeto, mayor es la habilidad del ojo para ver los detalles. El diámetro de la pupila disminuye con forme se incrementa la iluminación de un objeto y esto incrementa la profundidad de enfoque. También, el ojo se hace más sensible al parpadeo con forme la iluminación de un objeto disminuye. La iluminación relativa de una pantalla definirá parcialmente los niveles de parpadeo.

2.2.1.7 Consideraciones de diseño

Una persona con visión normal de color es capaz de distinguir cerca de 7 millones de colores diferentes. Por otro lado, sólo ocho de diez diferentes colores pueden ser identificados claramente, sin entrenamiento previo, cuando se examinan aisladamente por una persona con visión normal de color. Sin embargo, se debe decir que no sería prudente producir un sistema que usara todos estos colores a la vez por varias razones.

La sensibilidad al color no es uniforme a través de todo el campo de visión. Por ejemplo, el ojo no es sensible al color en la periferia. Si se pidiera a una persona que sostuviera un objeto a un lado y gradualmente lo acercara al hombro, si se mantienen los ojos al frente, eventualmente se notará la presencia del objeto pero no sería posible establecer su color.

Claramente, la discriminación del color sólo es posible entre $\pm 60^\circ$ de la posición de la cabeza (teniendo la cabeza y los ojos fijos). En la periferia de la visión de color, el ojo es menos sensible al rojo, verde y amarillo claro y más sensible al azul. Por esta razón, el azul es un buen color de fondo, especialmente donde la pantalla es grande.

Al frente del ojo, donde la visión del color es la mejor, el ojo es más sensible al rojo y amarillo y es menos sensible al azul. Los objetos pequeños azules tienden a desaparecer en la pantalla, y esto es particularmente cierto cuando el azul es pálido. Pequeños cambios en el matiz del azul son difíciles de distinguir pero el ojo es sensible a pequeños cambios en rojo. En efecto, el ojo es sensible a las diferencias entre colores en varios grados y el discernimiento de las diferencias de color no es uniforme a lo largo del espectro. Los diferentes colores también requieren reenfocar así que colores espectralmente extremos no deben colocarse juntos. El ojo humano encontrará difícil competir entre rojo y azul juntos, ya que esto requiere un reenfoque que inevitablemente generará cansancio. Sin embargo, rojo, naranja y amarillo pueden ser vistos juntos cómodamente.

El ojo no es un corrector de color, así que los colores cálidos (rojos y amarillos) serán vistos como viniendo hacia la vista y los colores fríos se ven como si se alejaran de la vista. Esto puede usarse en el desarrollo de interfaces gráficas de usuario para sugerir niveles de actividad.



Un pequeño porcentaje de gente son ciegos al color –cerca de 8 por ciento de los hombres y cerca de 1 por ciento de las mujeres. (para que una mujer sea ciega al color, se necesita que ambos padres tengan el gen de ceguera de color.) Es evidente que algunas personas no conocen su grado de daltonismo y han adoptado diferentes significados para identificar los colores. Esto se asocia probablemente con el matiz y profundidad del tono. La forma más común de daltonismo es rojo-verde pero existen otras formas raras. Un pequeño número de personas tienen daltonismo total.

Algunas personas encuentran la etiqueta “daltónico” como algo muy desagradable y no admiten la ceguera del color. Los constructores de interfaces deben ser sensibles a este factor y deben evitar colores que son factibles de causar problemas.

Se debe recordar que la clasificación de los colores es determinada tanto física como culturalmente. Las sociedades clasifican los colores de manera diferente. Hay sociedades que asocian el rojo con calor, peligro o detenerse, o el blanco con frío, etc.

En los individuos varían los niveles de color amarillo en el ojo, haciendo su percepción de color diferente. Más aún, el color detectado por un usuario puede depender del nivel de luminosidad de la pantalla, de modo que, si un usuario tiene la iluminación apagada puede ver los objetos amarillos de color café.

Por lo tanto no se debe confiar en el color como un elemento único y se debe usar como un medio para reforzar.

La elasticidad de los ojos disminuye con la edad. También, los ojos amarillos con la edad se van haciendo menos susceptibles al azul. Los ojos de algunas personas son naturalmente menos elásticos y más propensos al amarillo y por tanto la sensibilidad a la iluminación y al azul varía de un individuo a otro. Se debe recordar que la gente mayor puede ser menos sensible al azul y a ver colores menos brillantes generalmente. Adicionalmente, los usuarios mayores generalmente requieren mayor nivel de iluminación para su área de trabajo.

Ciertamente, el uso del color debe ser considerado con limitación y cuidado. Aún cuando el uso del color puede reducir el tiempo de búsqueda, el despliegue de múltiples colores puede incrementar ese tiempo. Es probable que para la mayoría de las personas es mejor ningún color que demasiados colores –efecto llamado Las Vegas- pero para los niños que les gustan los colores vivos puede no ser el caso. Para los niños, el color puede atrapar su imaginación y se puede posible experimentar mucho más con las interfaces diseñadas para el mercado joven.

Los juegos de cómputo tampoco requieren restringir el uso del color, pero se debe tener cuidado.

Es probablemente cierto decir que los diseñadores de sistemas deben, sin embargo, experimentar con los colores hasta encontrar la combinación de colores que mejor satisfaga las pruebas.

La apariencia estética del formato de la pantalla también será resultado de su impacto visual. Ya que la visión es un resultado de la interpretación por parte del cerebro de la información enviada por los ojos, es importante asegurarse de que el formato de la ventana no sea un desorden y que las opciones del menú se ordenen de forma agradable. Es importante mantener el mismo tipo de letra, por ejemplo, ya que la mezcla de diferentes tipos de letra en una pantalla pueden crear un efecto confuso. Por ejemplo, este texto se fue escrito usando un mismo tipo de letra para las descripciones y para los encabezados. Sería muy difícil de leer si se cambiara el tipo de letra en cualquier sitio sin razón aparente.

Sin embargo, el tamaño de la pantalla dictará, a cierto nivel, cuánta información puede colocarse en ella cómodamente. Algunas veces puede ser necesario restringir el número de cosas desplegadas a la vez. Por ejemplo, las opciones del menú principal pueden desplegarse y una vez que se selecciona una opción mostrar una lista de subopciones. Una pantalla confusa es difícil de interpretar porque hay demasiado que absorber por parte de los ojos y del cerebro. Demasiada confusión puede ser irritante porque el usuario es bombardeado con información que debe filtrar para realizar selecciones razonables sobre lo que el usuario desea hacer. El diseño de las pantallas debe finalmente ofrecer simplicidad y mantener la confusión al mínimo.

También, es importante que los elementos se desplieguen de acuerdo a su prioridad y deben ocupar el mismo espacio en la pantalla, a menos que su prioridad cambie. No es conveniente alterar la posición de un objeto a



menos que se deba a una buena razón. La siguiente tabla muestra algunas sugerencias para atrapar la atención visual.

Usar elementos visuales para atrapar la atención cuando:
La información es larga y compleja.
La información será referida posteriormente.
Para alertar y prevenir de algo que no necesite atención inmediata del usuario.
El sistema auditivo del usuario esté sobrecargado.
El usuario no esté en movimiento.
No se requiera una respuesta verbal.
La iluminación es buena.

Tabla1. Uso de los sistemas visuales para atrapar la atención.

La extensa investigación llevada a cabo en Psicología sobre este tema ha permitido que hoy tengamos guías para la selección del color en las interfaces. Un ejemplo de estas guías la podemos ver en la Tabla 2.

En general
1. Elegir combinaciones de colores compatibles. Evitar rojo-verde, azul-amarillo, verde-azul, rojo-azul
2. Usar contrastes altos de color entre la letra y el fondo
3. Limitar el número de colores a 4 para los novatos y a 7 para los expertos
4. Usar azul claro sólo para las áreas de fondo
5. Usar el blanco para la información periférica
6. Usar códigos redundantes (formas además de colores)
Para la Pantallas de visualización de datos
1. La luminosidad disminuye en este orden: blanco, amarillo, cian, verde, magenta, rojo y azul
2. Usar blanco, cian o verde sobre fondos oscuros
3. Para vídeos inversos usar nada (negro), rojo, azul o magenta
4. Evitar colores muy saturados

Tabla 2. Guía de selección de colores.

Como indica la regla general 3, es necesario tener presente que no se debe abusar de los colores como medios de codificación porque los problemas de visión del color son muy comunes. En este sentido conviene saber cuales son estas deficiencias de la visión cromática (Tabla 3).

Tipo	Descripción
Tricrómata	Visión cromática normal
Dicromático Protanopa	Insensible al rojo
Dicromático Deuterópata	Insensible al verde
Tritanopa	Insensible al azul y al amarillo
Monocrómata	Sin visión del color

Tabla 3. Deficiencias de visión cromática

2.2.2 El oído

Para la gente con visión normal, el oído es el segundo sentido más importante después de la vista. Para los ciegos parcial o totalmente, el oído o tacto pueden ser mucho más importantes que la visión. En la interacción hombre-computadora, el segundo medio de comunicación más importante, para gente con visión normal, es el sonido. El oído o audición, como el tacto es una respuesta a la presión. Sin embargo, nos informa acerca de los cambios que ocurren a varias distancias de nosotros. Estos cambios causan un desplazamiento de las partículas del aire que después retornan a su posición original. Aún cuando cada partícula se mueve una fracción ínfima esto causa una serie de variaciones de presión sucesivas. Cuando estas ondas de sonido llegan



a las orejas, causan una serie de cambios de presión mecánicos que eventualmente alimentan los receptores auditivos. Esto causa respuestas en el cerebro y la sensación de oír.

Las ondas de sonido varía en: **amplitud**, la altura de la cresta de la onda; **longitud de onda**, la distancia entre las crestas de las ondas; y **frecuencia**, el número de ondas por segundo. La amplitud y frecuencia son objetivas, dimensiones físicas. Sin embargo, estos son traducidos por el cerebro en sonoridad y tono. Un sonido parece aumentar su volumen cuando la amplitud se incrementa; el tono incrementa con el incremento de la frecuencia. El tono es que tan alto o bajo está un sonido.

2.2.2.1 Tono y Sonoridad

La frecuencia se mide en **hertz** (Hz). La mayoría de la gente puede detectar sonido en el rango de frecuencia de 20 a 20,000 Hz pero la habilidad para detectar los rangos superiores e inferiores se deterioran con la edad y también se afectan con la salud. El oído es más sensible en el rango 1000 a 4000 Hz; esto corresponde a las dos octavas más altas del teclado del piano.

La sonoridad de un sonido se describe en **decibeles**. El umbral del oído es definido como 0 decibeles. 20 decibeles es un susurro y 50 a 70 decibeles es una conversación normal. Puede ocurrir un daño al oído cuando el sonido excede 140 decibeles.

Sonido	Decibeles
Lanzamiento aéreo (a 150 pies)	180
Banda de rock	160
Trueno	120
Grito	100
Conversación	50
Susurro	20
Umbral del oído	0

Tabla 4. Algunos sonidos y su sonoridad

El oído es sensible a cambios de frecuencia bajo 20 decibeles. La sensibilidad a la frecuencia y sonoridad varía entre individuos y puede ser diferente para la misma persona en diferentes momentos. Por ejemplo, el oído es frecuentemente disparejo si un individuo tiene gripe.

El deterioro auditivo afecta a cerca del 10 por ciento de la población y no se toma tan en serio como los defectos de vista, es posible porque los seres humanos tienden a darle más importancia a la vista. El deterioro auditivo es muy frecuente entre los ancianos; especialmente a los sonidos de altas frecuencias. La gente con deterioro auditivo generalmente son reacias a admitirlo por lo que no se proporciona la ayuda para mejorar la calidad de la audición.

El sonido se usa como un localizador. De forma interesante, para los usuarios de teléfonos móviles en ocasiones es difícil identificar de quién es el teléfono que suena. Esto es probablemente porque la posición de cada teléfono no es suficientemente distante. Sin embargo, los humanos son normalmente adeptos a usar sonidos como un medio para localizar cosas, de modo que en algunas circunstancias se deben emplear otras tácticas. Por ejemplo, los buzos no pueden localizar la dirección de un sonido porque las ondas de sonido viajan más rápido en agua que en aire, por lo que están entrenados para voltear alrededor cuando hay un sonido que los alerta.

El sonido es una buena forma de atraer la atención ya que la gente responde más rápidamente a las señales auditivas que a las visuales.

Los sonidos pueden ser una fuente de incomodidad y distracción. Es invasiva ya que no se puede decidir no escuchar a diferencia de que se puede decidir no mirar. Se debe usar escasamente como la sal. La siguiente tabla sugiere algunas formas en las cuales el sonido puede ser usado y dónde es más apropiado.



Uso de sonido para atraer la atención cuando:
La información es corta y simple.
La información no será referida posteriormente.
Para alertar y prevenir de algo que necesite atención inmediata del usuario.
El sistema visual esté sobrecargado.
El usuario esté en movimiento.
Se requiera una respuesta verbal.
Se tenga iluminación pobre lo que hace difícil la visión.

Tabla 5. Uso del sistema auditivo para atraer la atención.

2.2.3 Tacto

El sentido del tacto es probablemente un poco desestimado en lo que se refiere al desarrollo de sistemas de interacción hombre-computadora. El tacto es un sentido muy importante para los ciegos o aquellos que tienen deterioros de la vista. Es útil en ambientes ruidoso (ya sea visual o auditivamente) donde se requiere otro canal para atraer la atención. El uso del tacto es importante en el uso del teclado, por ejemplo. Los usuarios pueden quejarse de cómo se siente un teclado si no obtienen la respuesta adecuada. Es una de las mayores quejas hechas acerca de los sistemas de procesamiento de texto. Incidentalmente, frecuentemente es el resultado de que la persona incorrecta realice las pruebas de un sistema y, lo compre una persona que no será la que lo va a usar.

¿Por qué nos debemos preocupar por el sentido del tacto en la PI? Podemos decir que existen dos razones fundamentales. En primer lugar, el tacto se está convirtiendo en un canal sensitivo importantísimo en el diseño de sistemas de Realidad Virtual donde el usuario explora mundos virtuales con sus manos. En este sentido, el diseñador debe saber como el tacto activo funciona cuando exploramos objetos y tratamos de descubrir sus propiedades. Al tacto activo también se le llama percepción táctil y es más preciso y útil que el tacto pasivo. En segundo lugar, para personas con discapacidades visuales o auditivas es muy útil el sistema táctil. En este sentido, los diseños de interfaces son un desarrollo que se basa en la larga historia de sistemas alternativos de comunicación (ej. Braille).

La piel es nuestro sistema sensorial más grande. Contiene muchas clases de receptores, los cuales tienen terminaciones nerviosas libres o encapsuladas. No existe una clara división entre todos los tipos de receptores de la piel y sus funciones.

2.2.4 Gusto

El sentido del gusto proporciona información acerca de lo que se debe o no comer. Intenta evitar que las criaturas vivas coman cosas que pueden ser venenosas. Los receptores en este caso son las papilas gustativas que responden a químicos que se disuelven en agua. Algunos estudios indican que existen cuatro sabores diferentes: amargo, salado, agrio y dulce; y se cree que todas las otras sensaciones de sabor se producen por una combinación de estos sabores. En promedio, un ser humano tiene 10,000 papilas gustativas, principalmente en la lengua, también existen algunas en otras regiones en la boca. Los mensajes son pasados desde los receptores al cerebro.

2.2.5 Olfato

El olfato proporciona información sobre los químicos en el aire. Estos excitan los receptores localizados en la cavidad nasal.

El olfato es relativamente poco importante para los seres humanos a diferencia de los animales como los perros para los que es de vital importancia. Cualquiera que ha tenido un perro, sabe que si se le muestra algo a un perro, primero lo olerá y después probablemente intentará probarlo. Si por otro lado, se le muestra algo a un ser humano, primero lo observará. Si se desea que lo huela, generalmente se le tendrá que pedir que lo haga.

Sin embargo se debe indicar que el olfato tiene probablemente más efecto sobre nosotros del que nos damos cuenta.



El sentido del olfato ha comenzado a ser explorado en PI por las posibilidades que ofrecen los olores para crear mundos virtuales parecidos a los reales. Además, en el contexto de las 'interfaces emocionales', el sentido del olfato adquiere una gran importancia porque tiene conexiones nerviosas directas con el sistema límbico que es el encargado de procesar las emociones. Sin embargo, las investigaciones sobre este sentido en PI está comenzando tímidamente debido a las grandes dificultades que tiene para ser utilizado en el diseño de interfaces.

El olfato y el gusto, son sentidos químicos. Los receptores detectan la presencia de moléculas en el aire. Las características más importantes de los receptores olfativos y que los hacen tan difíciles de tratar en PI son las siguientes:

1. **Adaptación**, si los receptores son expuestos durante mucho tiempo a un mismo olor pierden selectivamente la sensibilidad a ese olor;
2. Existe una **gran variación individual** en la sensibilidad al olor, lo que hace que sea difícil diseñar interfaces olfativas para que sean usadas universalmente.

2.3 Capacidades mentales de los usuarios

Comprender la memoria humana es uno de los aspectos más importantes del procesamiento de la información humana para aquellos que están relacionados con el desarrollo de sistemas de interacción hombre-computadora. Algunas aplicaciones modernas de cómputo operan comúnmente como si los usuarios tuvieran una memoria increíble. Estas esperan que el usuario recuerden toda clase de datos insignificantes. Peor aún, un conjunto de datos irrelevantes aprendidos para una aplicación en particular generalmente será totalmente inútil para otra aplicación, para la cual se deberá aprender un conjunto de datos irrelevantes. Todo esto incrementa la carga de las ya sobrecargadas memorias y caen en la ineficiencia causando olvidos y probablemente ansiedad y tensión en el trabajo.

Como mucho de la psicología humana, la memoria es un tema altamente complejo y fascinante. Sin embargo, en esta unidad sólo se revisarán algunos conocimientos básicos de qué es la memoria humana y cómo parece operar. Lo que es más importante, es que los estudiantes de PI desarrollen un entendimiento de lo que los usuarios pueden esperar recordar razonablemente cuando trabajan con un sistema de cómputo. Lo más importante es no sobrecargar la memoria del usuario. Pero al mismo tiempo, la intención no es crear un sistema que no maximice el potencial del ser humano y del sistema de cómputo. El objeto es por tanto proporcionar un entendimiento de las fortalezas y debilidades de la memoria humana para evitar su sobrecarga.

Es importante comprender que no existe algo así como memoria promedio. Antes de iniciar, vamos a empezar con un experimento corto. Lea la serie de letras que siguen e intente memorizarlas. Se requiere aprenderlas exactamente como aparecen en la página, en orden y posición. Emplea un par de minutos para aprenderlos y cuando este listo, continúe la lectura.

gnihsub
potksed

2.3.1. Un Vistazo a la Memoria

La memoria humana probablemente consiste de una serie de sistemas conectados. No existe un acuerdo general sobre la forma que toma, pero una de las descripciones más útiles para propósitos de la interacción humano-computadora consiste del reconocimiento de tres tipos de memoria. Estas son:

- Memoria a largo plazo
- Memoria de trabajo
- Memoria sensorial

La memoria a largo plazo es donde se almacenan los recuerdos. La memoria de trabajo es donde se realiza el proceso operativo consciente. La memoria sensorial es el área de memoria relacionada con las entradas desde los sentidos.

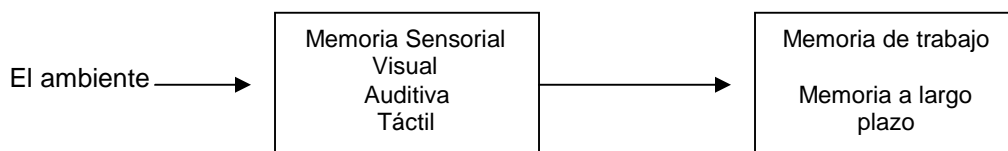


Figura 6. Tipos de memoria

2.3.1.1 Memoria de trabajo o memoria de corto plazo

Existe mucho debate acerca de la naturaleza de la memoria, si consiste de sistemas separados o es un sistema unificado, operando en diferentes formas de acuerdo a sus necesidades. En el pasado, los psicólogos identificaron la memoria a corto plazo (STM –short term memory) y la memoria a largo plazo (LTM –long term memory). La STM fue vista como un área de memoria que era capaz de mantener información limitada por un tiempo corto y esta información puede ser olvidada si no se refresca constantemente. Hoy, el trabajo de los psicólogos como Alan Baddeley han llevado al desarrollo de una vista alterna de la memoria. Brevemente, Baddeley sugiere que la memoria consiste de LTM y algo que llama memoria de trabajo (WM –working memory). Él argumenta que en efecto la STM se exhibe en experimentos de memoria y es un aspecto de la WM, y sus limitaciones. Para el propósito de la PI, aunque es conveniente pensar en tipos de memoria separados, es probablemente más fácil ver a la memoria como una entidad única que exhibe diferentes aspectos de sí misma dependiendo de las circunstancias bajo las cuales esté operando.

El área de la memoria consciente donde el proceso actual toma lugar ha sido etiquetado como memoria de trabajo por algunos psicólogos en un esfuerzo de ilustrar una diferencia entre esta y la memoria a largo plazo donde los conceptos se almacenan. Es suficiente para propósitos de comprender el uso de un sistema de cómputo decir que la memoria de trabajo actúa como un lugar donde la información puede ser trabajada por el proceso de pensamiento consciente. Puede ser alimentada con los recuerdos almacenados en la LTM o recibir información de varios receptores sensoriales. Es limitada y es esta limitación la que afecta el tipo de interfaz de cómputo que puede ser desarrollada para usuarios particulares.

Lo que fue etiquetado como memoria a corto plazo (STM) puede ser visto por tanto como un aspecto de la memoria cuando está en un modo de trabajo. La STM y por tanto la WM están limitadas en capacidad mientras que la LTM no. Estas limitaciones de la STM se decía anteriormente que eran de siete piezas de información, mas o menos dos;

2.3.1.2 Memoria a largo plazo

La memoria a largo plazo (LTM) es un área de memoria donde la información se almacena y puede ser recuperada después de periodos de tiempo largos. Para todos los propósitos, la memoria a largo plazo (LTM) es infinita. En otras palabras, es poco probable que un individuo sea capaz de llenar su memoria durante su vida. Sin embargo, los rastros de memoria parecen volverse menos accesibles y parece que los humanos olvidan cosas. Freud creía que la información nunca se pierde realmente de la memoria a largo plazo pero se vuelve cada vez menos accesible, también creía que bajo las circunstancias correctas, el material puede recuperarse pero como existe poco o ningún control sobre este proceso, el estado de olvido puede ser visto como algo probablemente permanente. El porque la gente olvida no es realmente comprendido aunque algunas veces parece ser un bloque en particular de memoria. Freud creía que el olvido era algunas veces una acción deliberada como parte del pensamiento del individuo pero inconsciente.

El estímulo eléctrico de varias partes del cerebro ha causado que varios recuerdos hayan venido a la mente cuando ocurre el estímulo; sin embargo, los datos disponibles son inconclusos. Realmente, hoy existen muchos argumentos acerca de cómo se almacenan los recuerdos. Se ha sugerido que no se almacenan en lugares, como almacenamos cartas en un cajón, sino que se difunden por todas partes del cerebro. Puede ser conveniente pensar en los recuerdos como si se almacenaran datos en un arreglo con cada paquete consistente de una fracción de la memoria entera y un apuntador a donde está el siguiente elemento almacenado. Sin embargo, para propósitos de la interacción humano-computadora es suficiente hablar acerca almacenamiento de recuerdos pero es sensible recordar que la comprensión del cerebro humano está aún lejos de completar. Puede ser que a futuro se cambie la forma de pensar acerca de los aspectos de la memoria humana.



No existe control sobre la forma en la cual el material se almacena y relaciona y no existe un control completo sobre lo que será recordado en la memoria a largo plazo ya que frecuentemente recordamos eventos que preferiríamos olvidar y olvidamos cosas que nos gustaría recordar. Es importante tener presente que todas estas representaciones del comportamiento humano son muy simplistas. Aún cuando es posible diseñar sistemas que sean fáciles de usar y fáciles de aprender no es posible tomar en cuenta al individuo que olvida porque tiene un bloqueo psicológico causado por problemas psicológicos. Esta falta de habilidad puede necesitar ser considerada cuando se prueba un sistema.

Los elementos de la LTM recuperados y llevados a la memoria de trabajo antes de que puedan ser usados; es el proceso de activación y, obviamente, toma tiempo. El equivalente paralelo en términos computacionales sería la diferencia entre un arreglo y un archivo. El arreglo almacena menos información pero se almacena en la memoria volátil, así que el tiempo requerido para procesarla es rápido. Por otro lado, un archivo puede almacenar grandes cantidades de información, pero esa información debe colocarse en la memoria volátil antes de que pueda ser usada y esto requiere mas tiempo de procesamiento. La LTM y la memoria de trabajo son como ellas. Los elementos en la memoria de trabajo pueden ser procesados más rápido que la información que primero debe ser recuperada de la LTM. Así cualquier sistema que requiera recuperar de la memoria a largo plazo hará más lenta la actividad del usuario.

2.3.1.3 Memoria Sensorial

La memoria sensorial es un área de la memoria consciente que trata con información desde los sentidos, esto es, desde los ojos, oídos, lengua y canal táctil. Parece que existe un almacenamiento separado para cada uno, ya que la información recibida desde los sentidos puede ser “recreada” o invocada a voluntad y pareciera estar separada de la memoria de trabajo. Uno puede ser alertado del olor de algo que se quema antes de procesar esta información. Algunas memorias sensoriales parecen ser mas durables y efectivas que otras. Por ejemplo, la memoria auditiva es mejor que la memoria visual a este respecto.

La memoria del gusto y el olfato es probablemente aún mas fugaz aunque es claro que los individuos pueden variar mucho y algunas personas, por ejemplo, los perfumistas, han desarrollado sus sentidos del olfato a un nivel notable y pueden identificar varios componentes de un olor en particular. Esto puede ser una sorpresa al principio pero si se considera a un animal como el perro, por ejemplo es menos sorprendente. El sentido del olfato de un perro es muy superior al humano. Un perro recordará a una persona evocando su olor individual. Esta información puede ser almacenada por años de la misma forma que los seres humanos pueden recordar la cara de alguien. Lo que es cierto para la mayoría de las personas es que los olores pueden evocar fuertes recuerdos y algunos olores y sabores pueden traer estímulos satisfactorios. Aún así, los sistemas de cómputo no usan los olores y sabores ya que no es aún muy claro cómo esto puede ser usado. Sin embargo, la realidad virtual puede eventualmente necesitar considerar todos los sentidos y usarlos para crear una atmósfera de una experiencia real. Más aún se debe notar que la demanda hecha al sentido de la vista es considerable, y generalmente en detrimento de ella. Puede ser que en el futuro, sea deseable reducir la carga en el sistema visual e incrementarla en otros sistemas.

2.3.1.4 Memoria de episodios y memoria semántica

Además de la existencia de la memoria a largo plazo, memoria de trabajo y memoria de los sentidos, una distinción útil puede ser hecha entre los tipos de memoria que el ser humano puede tener, por ejemplo la memoria de episodios y la memoria semántica.

La *memoria de episodios* involucra la memoria de eventos, por ejemplo, lo que uno cenó, o a dónde fue el sábado.

La *memoria semántica* es el conocimiento del mundo, por ejemplo, lo que significan las diferentes palabras, cuántos centímetros hay en un metro, etc.

Estas proporcionan un panorama inicial al trabajo de la memoria.



2.4 La Memoria en Acción

Es posible ahora considerar el papel de la memoria en el procesamiento humano, en más detalles. Por ejemplo, la memoria de trabajo juega un papel importante en la comprensión del lenguaje y el cálculo. En ambos casos, existe la necesidad de mantener ideas en la cabeza mientras el proceso de interpretación de las palabras o el cálculo se lleva a cabo. Por ejemplo, para comprender una oración se necesita recordar el principio mientras se lee o escucha el resto. Si no se pudiera hacer eso, seríamos incapaces de recordar el sujeto de la oración. Hay personas con daño cerebral a las que les es difícil o imposible realizar esta tarea y es gracias al estudio de estas personas que el entendimiento de la memoria ha avanzado. En el proceso de comprensión de una oración, la memoria de trabajo debe participar, de otra forma no se podría obtener sentido al mundo. Una vez que se obtiene el material, puede ser almacenado o eliminado conforme se requiera. Algunas ideas serán efectivamente perdidas pero otras serán recordadas y pueden ser evocadas pensando sobre ellas o repitiéndolas a otras personas.

Por otro lado, la memoria sensorial está involucrada en el procesamiento de la información que se recibe de los sentidos. Por ejemplo, cuando se ve una película, se ve un movimiento continuo de la escena aunque en realidad se están viendo una serie de imágenes fijas continuas con periodos de oscuridad. El sistema visual almacena la información desde una imagen hasta que llega la siguiente. Estas se ponen juntas por el cerebro de forma que parecen una escena continua. Esto es porque nuestra comprensión del mundo nos dice que una persona no aparece repentinamente en un lugar o en otro; se tienen que mover de un área a la siguiente.

Por ejemplo cuando se ve una antorcha que gira; se puede notar que se ve un círculo porque el ojo recuerda el rastro y el cerebro convierte esto en una imagen continua. Similarmente, es posible revocar palabras que se escuchan, si alguien dice algo y uno no lo comprende, se pueden evocar nuevamente lo dicho de forma similar a si se reprodujeran en una grabadora. La memoria sensorial para los oídos parece ser más durable que la visual ya que por experimentos con la memoria se sabe que recordar una lista que se lee será más difícil que recordar una lista que se escucha y se escribe, esto probablemente porque la información se codifica de dos formas diferentes.

2.5 Experimentos de Memoria

Se han hecho muchos experimentos de memoria y estos han permitido obtener una imagen de cómo está constituida la memoria humana. Cuando sea posible, se debe intentar experimentar uno mismo. Aunque no se pueda llegar a alguna conclusión, esto ayuda a comprender cómo trabaja la memoria.

2.5.1 El hilado de dígitos

Para considerar la memoria podría ser conveniente observar algunos aspectos que han surgido de experimentos llevados a cabo desde hace más de cien años. La cuestión de la capacidad de la memoria a corto plazo (STM) fue estudiada por un profesor de una escuela de Londres, J. Jacobs, en 1887. Estos experimentos, realizados en sus estudiantes, le permitieron definir una técnica llamada el hilado de dígitos. Esta tiene efectos profundos ya que el tipo de experimentación llevada a cabo en la memoria y su legado pueden ser vistos en la experimentación de memoria moderna. En efecto, la sociedad tiende a clasificar a la gente como que tienen buena o mala memoria basados en los resultados de pruebas de memoria como los que Jacobs realizó. Aquellas personas que pueden hilar muchos dígitos son clasificadas como que tienen buena memoria.

El hilado de dígitos se determina como sigue. Se le presenta al sujeto una serie de dígitos y se le pide que los repita en el mismo orden. El número de dígitos es gradualmente incrementado hasta que finalmente el sujeto falla al recordar la lista correctamente. La longitud de la secuencia hasta la cual estuvo bien el promedio de las veces es el hilado de dígitos.

Realiza tu mismo(a) la prueba. Has anotaciones de cuando te equivocas en la secuencia para encontrar tu hilado de dígitos. La siguiente figura proporciona un conjunto de números para hacer la prueba.



4579
3825
6514
83192
68259
54281
913825
648371
596382
7958423
5316842
5271468
86951372
51739826
31598247
827419435
783164952
615933826
9152338162
7157996193
3829667318

Figura 7. El hilado de dígitos

2.5.2 Agrupar información (Chunking)

Los experimentos de George Miller mostraron que la capacidad de la memoria a corto plazo es de 7 más menos 2 trozos de información. Más puede ser recordado si los elementos pueden ser separados en porciones. Esto significa que las piezas de información se agrupan juntas de modo que forman un elemento a ser recordado. Miller se refirió a esto como el número mágico 7 mas o menos 2 (Miller, 1956).

Obviamente esto tiene implicaciones para el diseño de un sistema de cómputo. Esto significa, por ejemplo, que sería conveniente no presentar mas de 7 mas o menos 2 elementos al usuario, para que el usuario los recuerde. También, las claves de más de 5-9 caracteres serán difíciles de recordar para la mayoría de la gente a menos que el agrupado de información sea posible. Este agrupamiento no siempre es obvio o predecible, de modo que lo que es fácil para una persona es difícil para otra. Es importante no asumir que porque una persona encuentra una tarea fácil, otra será capaz de realizar esa tarea también fácilmente. La capacidad de la memoria varía de una persona a otra e incluso en un mismo individuo hay variaciones. Algunas personas parecen tener buena memoria para las caras o para los números, etc.

El chunking significa agrupar la información en secciones que tienen sentido para el individuo y son vistas como entidades por ese individuo. Esto no necesariamente significa que el agrupamiento será el mismo para todos, aunque algunas veces esto será indudablemente el caso. Por ejemplo, la siguiente secuencia de número puede ser difícil de recordar si se tratan como una entidad:

051594737

Sin embargo, si se tratan como una serie de tres grupos serán más fáciles de recordar:

051-594-737

La mayoría puede recordar números telefónicos de esa longitud o más largos.

Similarmente, será difícil recordar las letras:

G G N N T I H I A U

Pero será más fácil recordar lo siguiente:

H U N G T I N G



La razón de que HUNGATING es más fácil de recordar que GGNNTIHIAU que las letras de la primera pueden dividirse en dos grupos HUN y GATING. Si se intenta recordar una palabra real será aún más fácil porque el material tendrá sentido y está por lo tanto en la forma de un grupo de información. Una oración consiste de muchas palabras que serán vistas como una pieza de información porque están agrupadas y pueden ser fácilmente recordadas. Todos hemos sido capaces de repetir algo que se nos dijo antes; mientras que las palabras absurdas de una longitud similar son más difíciles de recordar.

2.5.3 Efectos primacia y reciente

a partir de la experimentación en la memoria es posible hacer algunas generalizaciones acerca de la forma que la cual la memoria humana trabaja. La siguiente tabla muestra algunos conjuntos de palabras. Lee cada lista una vez de forma continua y después trata de recordar tantas como puedas escribiéndolas con forme las recuerdes al terminar la lectura. No se debe hacer muy lento, se deben tomar sólo unos minutos.

<i>Lista 1</i>	<i>Lista 2</i>	<i>Lista 3</i>	<i>Lista 4</i>
barrera	bebés	tarro	archivo
arma	sofá	postal	corazón
corbata	vestíbulo	lámpara	espantapájaros
periódico	reloj	chinchilla	estilo
concha	pulir	borrador	antojo
tomate	dintel	sombrero	felpudo
apología	perro	gasa	mosca
tabla	muñeca	florero	lápiz
planta	oasis	popurrí	francachela
química	festival	lapicero	Neptuno
identidad	mosquito	fogata	magnífico
filtro	cortinas	vela	clip
salsa	entrar	vino	capturista
azulejos	precinto	mecedora	metro
directorío	argumento	pasa	accidente
Recordar...	Recordar...	Recordar...	Recordar...

Tabla 6. Prueba de Memoria.

Recordar las listas en cualquier orden es conocido como **recuerdo libre** y es más fácil que recordar las listar en orden. En el ejercicio de recuerdo libre existe una tendencia a recordar la primera o las dos primeras palabras bien; esto se llama **efecto primacia**. Este sugiere que las primeras palabras son repetidas varias veces y el sujeto efectivamente las aprende, o estas ingresan a la memoria a largo plazo. A continuación, las palabras en el centro serán menos recordadas mientras que las palabras al final serán generalmente bien recordadas; esto es llamado es el **efecto reciente**. Las palabras al final se recuerdan porque están aún presentes en la memoria operativa o en la memoria auditiva si estas fueron escuchadas. Las palabras raras son más difíciles de recordar que las palabras comunes, y las palabras que crean una imagen fuerte (como elefante) se recuerdan fácilmente porque probablemente se codifican como una palabra y como una imagen visual. Cuando ocurre esta doble codificación el concepto es mucho más fácil de recordar.

La forma en la que la información puede ser agrupada puede también ayuda a almacenar y recuperar palabras. Por ejemplo, al principio de esta sección se pidió aprenderse una serie de letras –intenta recordarlas ahora. Probablemente no se noto que se puede leer como “desk top publishing” al revés. Si se hubiera hecho así, esto habría sido más fácil de recordar por estarían agrupadas como una unidad. Sin embargo, se pueden usar mnemónicos, cosas que ayudan a recordar, tales como hacer una historia, nombres familiares, con el objeto de ayudar a recordarlos.

Antes de continuar se puede realizar otro experimento. Esta vez antes de recordar la lista, realizar un cálculo después de cada lista. Hacer la suma en la cabeza, y entonces recordar la lista como se hizo anteriormente. La siguiente tabla servirá para el ejercicio:



<i>Lista 1</i>	<i>Lista 2</i>	<i>Lista 3</i>
violeta	rescatar	angora
viruela	lápida	establo
elefante	flor	saludos
teléfono	fuelle	canasta
pendiente	estatua	curso
huevo	tonto	bolsa
melancolía	pulgón	gusano
queso	sorpresa	brincolín
tarro	impresora	biscocho
ninfa	cenotafio	bastón
lancha	madriguera	genoma
rastro	magneto	cocinar
muelle	ley	chocolate
vagón	cochecito	parcela
macabro	emparedado	tetera
6	7	1+
+5	+2	+6
+3	+5	-4
+8	-3	+9
-5	-2	-2
-1	+9	+8
+7	+4	+9
-3	-6	+6
+9	+8	-3
Recordar...	Recordar...	Recordar...

Tabla 7. Prueba de Memoria con Cálculo.

Probablemente esta vez fue más difícil. Cuando los sujetos reciben una lista y luego se les pide realizar un cálculo, o cualquier otra actividad, antes de recordar, la otra actividad interfiere con la recuperación del material y tiende a destruir la memoria de las palabras al final de la lista. En otras palabras, las palabras al principio de la lista serán recordadas pero el resto de la lista puede ser menos fácil de recordar. El efecto primacía todavía opera –las pocas primeras palabras generalmente serán recordadas correctamente. Sin embargo, el efecto reciente será destruido por la tarea de tener que hacer el cálculo.

2.5.4 Conclusión

Cuando se ve un número telefónico en el directorio y se repite antes de marcarlo, una vez que se marque se tiene una sensación de descanso. Esto es llamado **conclusión** y es muy importante recordar esto cuando se construyen sistemas de cómputo interactivo. Por ejemplo, cuando se firma en un sistemas de cómputo, si se logra estará a gusto con el mensaje de bienvenida. Se sabe que está listo para proceder a la siguiente etapa. A los seres humanos les gusta saber cuándo ha sido completada una tarea, esto les indica que son libres de moverse a la siguiente etapa y olvidarse de la tarea que acaban de realizar. Esto mismo se debe haber sentido cuando ya no se tienen que recordar las series de letras al inicio de la sección. Todo el tiempo que se estuvo tratando de no olvidarlas probablemente se sintió cierta tensión. Es difícil concentrarse totalmente en otra tarea cuando se está desesperado tratando de recordar algo que no es particularmente fácil de recordar.

Es importante construir conclusiones en los sistemas ya que esto actúa como un medio para permitir que el procesamiento se agrupe en la memoria. El proceso se convertirá después en un solo grupo para el usuario experto. La acción de proporcionar una conclusión también le dará un respaldo al usuario novato y ayudará en su proceso de aprendizaje reduciendo la sobrecarga de memoria.

2.5.5 Maximizar la memoria

Existe evidencia de que la información que se codifica en más de una forma, es más fácil de recordar que la información que se codifica en sólo una forma. Por ejemplo, si se le presenta al sujeto una serie de imágenes, estas pueden ser más fácilmente recordadas que las palabras correspondientes. Existe mucha evidencia para



soportar la consideración de que los seres humanos son capaces de recordar imágenes mejor que las palabras. Esta tendencia es usada como un argumento para el desarrollo de interfaces de manipulación directa o iconos.

Es importante que el usuario se siente cómodo cuando está usando un sistema. En estado de excitación – temor, angustia, gozo, o cualquier otra emoción extrema, es menos probable que el ser humano recuerde cosas. Los experimentos realizados por Loftus han mostrado que los testigos de crímenes pueden ser poco confiables. Esto es porque comúnmente los testigos están distraídos por el factor de la violencia o no son capaces de recuperar la información.

Una interfaz de usuario debe fomentar una actitud relajada ya que el usuario tendrá más éxito con un sistema si no está bajo tensión. Sin embargo, la relajación no implica que el usuario se quede dormido. Tristemente, no existe evidencia que soporte la idea de que el aprendizaje puede realizarse durante el sueño y el usuario debe estar despierto y alerta. No se pueden usar técnicas de aprendizaje durante el sueño como medio para revisar las interacciones humano-computadora. La experimentación en aprendizaje durante el sueño tienden a sugerir que el “aprendiz” puede recordar sólo aquellas pequeñas porciones de información que escucharon antes de dormir; no aprendieron durante el sueño.

Baddeley (1994) ha sugerido por sus experimentos con buzos que la gente recuerda más información fácilmente si son capaces de recordarla en la misma situación en la cual fue aprendida. Los buzos encuentran más fácil recordar elementos que han aprendido bajo el agua cuando están bajo el agua y las cosas aprendidas en tierra fueron mejor recordadas en tierra. Esto también aplica a estados físicos. Para los sujetos a los que se pidió recordar donde habían dejado cosas cuando estaban intoxicados, les fue más fácil de recordar su localización cuando estaban intoxicados nuevamente que cuando estaban sobrios. Esto pudiera sugerir que aprender a usar un sistema puede ser mejor en el mismo ambiente/ estado en el lugar en el que será usado el sistema. Este es un punto que se debe tomar en cuenta cuando se hagan las pruebas y evaluación de un sistema. Es mejor entrenar a los usuarios y evaluar un sistema en el ambiente donde operará.

Anderson sugiere que el humor es también importante, Los sujetos encuentran más fácil recordar eventos felices cuando son felices y tristes cuando están tristes.

Debido a las limitaciones de la memoria operativa (WM) es necesario presentar elementos al usuario de forma que minimice las fallas. Por ejemplo, para la mayoría de la gente, las claves necesitan ser de no más de 9 caracteres (el número mágico 7 más o menos 2). La necesidad de recordar información para entradas posteriores debe mantenerse al mínimo para no sobrecargar la WM. Esto significa que el sistema de cómputo no debe pedir información innecesaria o que pueda derivar. Después de cada operación completada, debe haber alguna operación que actúe como conclusión y ofrezca un respaldo al usuario. Es importante que la memoria del usuario no sea sobrecargada.

Más aún, es importante pedir material que sea relevante. Los usuarios son más cooperativos con el sistema de cómputo que hace solicitudes que tienen sentido. Solicitar información irrelevante causa irritación y no ayuda al usuario a aprender cómo operar el sistema. Es muy fácil ser irritado por formularios que hacen preguntas que la persona puede considerar irrelevantes o hasta impertinentes. Los sistemas de cómputo evitar colocar sobrecargas innecesarias al usuario y evitar causar irritación o incomodidad.

Es posible entrenar la memoria para ejecutar hazañas notables. Sin embargo, generalmente hablando, en el diseño de sistemas de cómputo es deseable reducir cualquiera de esas necesidades. No se les debe solicitar a los usuarios que hagan cosas que puedan causarles sobrecarga de memoria indebidamente. Algunas personas pueden encontrar fácil recordar algunos grupos de información mientras que otras no serán capaces de realizar esa tarea.

2.6 Guía para crear interfaces

2.6.1 Consistencia

Las interfaces necesitan ser consistentes. Es difícil enfatizar esto suficientemente. No se debe esperar que el usuario recuerde una secuencia diferente de eventos por acciones similares. Por ejemplo, el teclado de la computadora sigue la secuencia qwerty, sería raro y además sería difícil entrenar a un capturista para que



reaprenda un conjunto diferente. Como esto, cualquier solicitud de información debe ser solicitada en un formato que es usual para el usuario. En México, por ejemplo, la gente podría esperar ingresar la fecha en el orden de día, mes y año (01/01/01), si se le solicita ingresar la fecha con un formato xx/xxx/xx (01/ene/01) les será muy difícil adivinar lo que el sistema requiere.

2.6.2 Sugerencias visuales y auditivas

Las sugerencias visuales y auditivas deben darse para confirmar que el usuario ha terminado adecuadamente o el procesamiento llegó al objetivo. Por ejemplo, la Apple Macintosh usa un sonido cuando ocurre un error. Se proporciona una confirmación visual desplegando una barra que se llena gradualmente conforme se avanza en la copia de un archivo. Otro tipo de confirmación visual son los porcentajes de avance en la pantalla. Estos elementos visuales ayudan a dar confianza al usuario que el procesamiento se dirige al objeto de la forma esperada. Esa confianza es especialmente importante para los principiantes ya que ayudan a confiar en el sistema. Los principiantes generalmente están ansiosos sobre lo que puede pasar y algunas veces esa ansiedad está bien infundada. Los sistemas deben ofrecer más información sobre los que está pasando.

La información dada verbalmente es más fácil de recordar que la información leída. Esto es porque la información se codifica en dos formas. Entre más formas de codificar la información existan, será más fácil de recordar. Tanto como sea posible, los sistemas de cómputo deben ser contruidos considerando la forma en que el ser humano opera. Con el desarrollo de entradas y salidas habladas probablemente algunos de los problemas asociados actualmente con la interacción humana con los sistemas de cómputo podrán solucionarse.

2.7 Memoria y Aprendizaje

La memoria y el aprendizaje están muy relacionados y es importante considerar las formas en que la gente aprende si deseamos construir sistemas humano-computadora efectivos. Los sistemas humano-computadora involucran tanto a la memoria de la computadora como a la memoria humana y es importante no pedirle al humano que realice tareas que pueden ser ejecutadas más fácilmente por la computadora. La computadora no sufre de disminución de la memoria de la misma forma que los seres humanos, por lo que es razonable que recuerde cosas que para los humanos es difícil de recordar. Se debe solicitar que cada parte del sistema realice lo que hacen de manera natural y no sobrecargar alguna de las partes. Recordar que si a los sistemas de cómputo se les pide hacer algo que no pueden hacer, no lo harán. Los seres humanos pueden intentarlo y terminar frustrados y enojados en el proceso.

Es muy fácil equivocarse con el diseño de sistemas de cómputo. Por ejemplo, hacer una lista de colores usando una pluma azul o negra. Hacer la lista usando el color apropiado por cada color, con rojo escribir rojo, verde con verde, etc. Después hacer una lista de colores usando el color inapropiado, rojo con azul, azul con amarillo, etc. Después hacer una lista de palabras cualesquiera con colores múltiples. Intentar leer las palabras en cada lista y decir los colores que están escritos. Probablemente se experimentará alguna dificultad. Este fenómeno es llamado *efecto Stroop*. El efecto Stroop reconoce que una vez que la gente ha aprendido a leer, tiene la tendencia a leer lo que sea que esté frente a ellos y cuando se encuentran con fuentes de información contradictorias, en este caso un color escrito en el color equivocado, existe un conflicto.

Es importante no pensar que la interacción entre los sistemas de cómputo y los usuarios está aislada. El usuario no solo es un usuario del sistema sino un empleado con una extensión telefónica, un RFC, un domicilio, una cuenta de banco, una lista de cumpleaños a recordar, etc.

2.8 Sistemas de interacción Humano-computadora.

Esta no es una lista completa de los factores humanos que se necesitan considerar en el diseño de interfaces. No se toman en consideración todas las capacidades humanas y aunque la vista es el aspecto más importante del usuario.

Aunque este es el estado actual de investigación, esto no significa que no puede haber cambios en la construcción futura de sistemas de cómputo. Por ejemplo, se podrían usar más los pies de los usuarios más de



lo que se usan actualmente. La mayoría de la gente es capaz de usar sus pies cuando conducen un auto y se pueden usar para manejar controles sensibles. Kroemer ha mostrado que la gente puede usar sus pies para controlar casi de forma tan efectiva como usan las manos. Osborne a remarcado que las manos, como el sistema visual, son casi siempre sobrecargados, así que el uso efectivo de controles de pie puede ser deseable.

Existen sistemas de cómputo especiales para usar por aquellos que no pueden usar sistemas convencionales. Comúnmente estos sistemas usan habilidades humanas que no se desarrollan en la mayoría de la población y estos otros métodos de control pueden desarrollar altamente a estas personas usándolos. Por ejemplo, los artistas son capaces de pintar con la boca o los pies. Los teclados pueden ser operados con un dispositivo colocado en la cabeza pero la mayoría de la gente cree que esto no existe.

Sin embargo, una comprensión del estado actual de la tecnología usada en la construcción de los sistemas actuales no puede prever lo que sucederá en el futuro. Es necesario ver al ser humano como un todo con otros talentos que no son explotados por los sistemas modernos de cómputo. El estudio de PI debe prepararnos para pensar en otros medios de interacción y prepararnos para modificarlos con el tiempo.



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

Licenciatura En Informática

Programación de Interfaces

*Autor: L.I. María de Lourdes Isabel
Ponce Vásquez*

AGOSTO-DICIEMBRE 2005



Contenido

UNIDAD 3. METÁFORAS, ESTILOS Y PARADIGMAS.....	4
Objetivos:	4
Introducción	4
3.1 Metáforas.....	4
3.1.1 Metáforas verbales	5
3.1.2 Metáforas visuales.....	5
3.1.3 Metáfora del mundo real.....	6
3.1.4 Metáfora global.....	6
3.2 Metáfora del escritorio.....	6
3.2.1 Historia.....	7
3.2.2 Opiniones críticas.....	7
3.2.3 Metáforas compuestas	7
3.3 Metodología de creación de metáforas	7
3.3.1 Definición funcional.....	8
3.3.2 Identificación de los problemas del usuario	8
3.3.3 Generación de la metáfora	8
3.3.4 Evaluación de las metáforas	8
3.3.5 Otra versión	8
3.4 Lenguaje Visual para el Diseño de Metáforas.....	9
3.4.1 Objetos familiares.....	9
3.4.2 Verbos	9
3.4.3 Ejemplo de construcción de una metáfora	9
3.5 Ejemplos de metáforas	10
3.6 Interacción	11
3.6.1 Interacción multimodal.....	11
3.6.2 Estilos de Interacción.....	11
3.6.2.1 Interfaz por línea de comandos	12
3.6.2.2 Menús y navegación.....	12
3.6.2.3 Manipulación directa.....	13
3.6.2.4 Interacción asistida	14
3.7 Agentes.....	15
3.7.1 Integración de agentes con otras aplicaciones	16
3.7.2 Aspectos a considerar en la integración agente–aplicación	16
3.7.3 Ejemplo: El agente de Microsoft.....	16
3.8 Asistentes	16
3.8.1 Ejemplo de asistente	16
3.8.2 Computadoras emocionales	17
3.9 Paradigmas.....	17
3.9.1 Computadora de escritorio	17
3.9.2 Entornos virtuales y realidad virtual.....	17
3.9.2.1 La cueva (CAVE)	18
3.9.2.2 Beneficios y problemas.....	18
3.9.2.3 Ejemplos de uso de la realidad virtual:.....	18
3.9.3 Computación ubicua.....	18
3.9.4 Realidad aumentada.....	19
3.9.4.1 Corrientes existentes	20
3.9.4.2 Líneas de trabajo	20
3.9.5 Diferencias entre Realidad Aumentada y Realidad Virtual	20
3.9.6 Computadoras corporales	21
3.9.7 Dispositivos utilizados en RA	21
3.9.8 Aplicaciones: Algunos escenarios de RA	22
3.9.8.1 El fontanero del futuro	22
3.9.8.2 Mantenimiento mecánico y reparación.....	22



3.9.8.3 Diseño interior.....	23
3.9.8.4 Construcción exterior	23
3.10 Comparación de los Paradigmas de Interacción.....	23



UNIDAD 3. METÁFORAS, ESTILOS Y PARADIGMAS

Objetivos:

- Aprender qué es una metáfora
- Conocer qué son las metáforas verbales
- Conocer qué son las metáforas visuales
- Conocer las metáforas globales
- Conocer la metáfora del escritorio y su historia
- Ver el uso de las metáforas en las interfaces actuales
- Discutir los problemas y las ventajas de su uso
- Aprender cómo diseñarlas y cómo aplicarlas con la máxima efectividad
- Entender y aprender qué es un estilo de interacción
- Tener una visión general y comparativa de los estilos y paradigmas de interacción
- Conocer el estado actual y la futura evolución de los estilos de interacción
- Aprender a elegir entre los diferentes paradigmas y dentro de estos paradigmas, qué estilos de interacción utilizar para una determinada aplicación

Introducción

Este capítulo está dedicado a dar una perspectiva de los escenarios y conceptos genéricos en que se organizan los sistemas interactivos: metáforas, estilos y paradigmas.

Hasta hace poco, la mayor parte de los intercambios de información entre las personas y las computadoras se hacían tecleando texto y recibiendo posteriormente las respuestas visualizadas en la pantalla y todavía es una manera de interactuar con la computadora muy utilizada.

Posteriormente, debido a la necesidad de su uso en las empresas, se comenzó a interactuar usando menús y formularios copiados de los existentes en las empresas. Desde hace un cierto tiempo, con la proliferación de la computadora personal, se ha generalizado la interacción por manipulación directa utilizando interfaces gráficas: en vez de teclear texto, manipulamos objetos visuales y los modificamos utilizando representaciones gráficas de estos objetos.

En el futuro se verán nuevas formas de interactuar con las computadoras: la realidad virtual, el uso del lenguaje natural, la migración de la interacción de las pantallas y el teclado al entorno que nos propone la realidad aumentada.

3.1 Metáforas

¿Qué es exactamente una metáfora y por qué se considera tan importante en el diseño de una buena interfaz? El término metáfora está tradicionalmente asociado con el uso del lenguaje. Cuando queremos comunicar un concepto abstracto de una manera más familiar y accesible utilizamos el recurso de las metáforas. Por ejemplo cuando hablamos del tiempo, que es un termino abstracto, lo expresamos en relación con dinero que es un termino concreto y hablamos de ahorrar, gastar, desaprovechar el tiempo, de hecho una gran parte del lenguaje está basado en el uso de metáforas.

En el diseño de las interfaces actuales, las metáforas tienen un papel dominante. La metáfora del escritorio introducida por la computadora Macintosh y de uso generalizado actualmente, supuso un cambio en la usabilidad de las computadoras. En esta metáfora los objetos en la pantalla, los nombres que se dan a las órdenes de comando, conceptos como escritorio, icono, menús, ventanas están basados en temas familiares y lo mismo sucede con las acciones a realizar, arrastrar, soltar, pegar, etc.



El uso de las metáforas puede asistir a los desarrolladores en conseguir maneras más eficientes y efectivas de desarrollar programas que permitan ser usados por comunidades de usuarios más diversas.

3.1.1 Metáforas verbales

Mucha gente cree que las metáforas son una especie de lenguaje floreado que aparece en la poesía y en las novelas baratas. Esto evidentemente no es correcto. Las metáforas son una parte integrante de nuestro lenguaje y de nuestro pensamiento y aparecen en nuestras conversaciones cotidianas, aunque muchas veces no nos demos cuenta.

La metáfora se utiliza cuando queremos comunicar un concepto abstracto de una manera familiar y accesible. Utilizamos conceptos bélicos para describir argumentos, decimos que tenemos ideas que pueden ser defendidas o atacadas; si una posición es indefendible nos podemos retirar, podemos tener puntos débiles que pueden ser destruidos. Como vemos, una parte del lenguaje está basado en las metáforas.

Cuando nos encontramos delante de una nueva herramienta tecnológica —como una computadora— se tiende a compararlo con alguna cosa conocida. Un ejemplo clásico de cuando se utiliza una computadora por primera vez, se compara inmediatamente con una máquina de escribir de una manera metafórica. Cuando se utiliza un tratamiento de texto pensamos que se comporta igual que una máquina de escribir.

Estos enlaces metafóricos proporcionan los fundamentos por los cuales el usuario desarrolla su modelo de computadora propio. Los conocimientos de los elementos y sus relaciones en un dominio familiar se traspasan a los elementos y sus relaciones en un dominio no familiar.

En el ejemplo anterior, los elementos son el teclado, la barra espaciadora y la tecla de retorno de carro. Las relaciones entre los elementos nos dicen, por ejemplo que *solo se puede presionar una tecla a la vez y presionar una tecla significa ver un carácter visualizado en la pantalla*. Lo importante a tener en cuenta es que basado en este conocimiento previo se puede desarrollar un conocimiento del nuevo dominio más rápidamente.

No obstante, hay que tener en cuenta que hay diferencias, por ejemplo, la tecla de retorno de carro mueve el carro físicamente hacia atrás mientras que en el procesador de texto se borra el carácter. No obstante, una vez asimiladas estas diferencias, el usuario construye un nuevo *modelo mental*.

Análogamente podemos aplicar otras metáforas, como por ejemplo, explicar el funcionamiento de los archivos con la metáfora de funcionamiento de un archivero.

En general las metáforas verbales pueden ser herramientas útiles para ayudar a los usuarios a iniciarse en el aprendizaje de un nuevo sistema.

3.1.2 Metáforas visuales

Xerox fue una de las primeras empresas que se dio cuenta de la importancia de diseñar interfaces simulando el mundo físico concreto que nos es familiar. En vez de pensar en interfaces verbales, como medio de hacer entender a los usuarios la interfaz, fue un poco más lejos y desarrolló *una interfaz visual basada en la oficina física*. Esta interfaz fue desarrollada para el sistema Star. La base de la metáfora consistió en crear objetos electrónicos simulando los objetos físicos en una oficina. Esto incluía papel, carpetas, bandejas y archiveros. La metáfora de organización global presentada por pantalla fue el escritorio y el área de trabajo se parecía a una típica mesa de oficina.

Los archivos fueron transformados en representaciones pictóricas en substitución de entidades abstractas con nombres arbitrarios.

Esta metáfora fue aplicada a la computadora Lisa de Apple y más tarde al Macintosh que tuvo un éxito de ventas excepcional.

Posteriormente fue aplicada a las computadoras personales, Windows sobre MS-DOS, Presentation Manager para OS/2 y en las estaciones de trabajo UNIX con las interfaces basadas sobre X-Windows.



La metáfora visual normalmente es una imagen que nos permite representar alguna cosa y que el usuario puede reconocer lo que representa y por extensión puede comprender el significado de la funcionalidad que recubre.

Las metáforas pueden variar de pequeñas imágenes puestas en botones de barras hasta pantallas completas en algunos programas. Las personas entendemos las metáforas por intuición. El diccionario define intuición como:

«Intuición. Cognición inmediata. Conocimiento de una cosa obtenida sin recurrir a inferencia o razonamiento.»

Se comprende el significado de las metáforas de los controles que se encuentran en la interfaz, porque se conectan mentalmente con otros procesos que previamente se han aprendido. Las metáforas no funcionan como la comprensión. A veces funciona, a veces no. Las metáforas se basan en asociaciones percibidas de manera similar por el diseñador y el usuario. Si el usuario no tiene la misma base cultural que el desarrollador es fácil que la metáfora falle, incluso teniendo la misma base cultural pueden haber faltas de comprensión importantes, por ejemplo la metáfora de la figura se ha utilizado como una metáfora de correo electrónico sin tener en cuenta que este tipo de buzón de correo es un elemento de la cultura americana, difícilmente comprensible para nuestra cultura.

3.1.3 Metáfora del mundo real

Las personas organizamos la información espacialmente. Si recibimos una llamada de teléfono en la oficina, mientras hablamos, escribimos, por ejemplo, el número de teléfono en un papel y lo colgamos en una pizarra de corcho por ejemplo. Una semana después quieres llamar a esa persona y piensas dónde dejaste el papel y tienes una imagen espacial en la mente de donde dejaste la nota. Este es un mapa espacial que todos nosotros tenemos en nuestra cabeza para registrar esta base de datos que llamamos mundo.

3.1.4 Metáfora global

La metáfora global es una metáfora que nos da el marco para todas las otras metáforas del sistema.

Por ejemplo la metáfora del escritorio, desarrollada originalmente para la computadora Xerox Star y generalizada con la computadora Apple Macintosh se puede considerar como un primer ejemplo de amplia difusión de metáfora global.

En un título de CD para niños, explorando el sistema solar, la metáfora global del centro de control de una nave espacial, constituye una opción que permite la inclusión del resto de metáforas dependiendo de la metáfora global.

3.2 Metáfora del escritorio

La metáfora del escritorio fue la primera metáfora global y está siendo utilizada prácticamente en todas las interfaces gráficas. La idea era reproducir un escritorio de oficina y todos los objetos que tenemos en ella y en sus alrededores. En aquellos momentos el uso de las computadoras estaba dirigido a los oficinistas, por tanto esta metáfora correspondería exactamente al entorno que la mayor parte de los usuarios trabajaban.

La base de la metáfora consiste en crear objetos electrónicos que simulan los objetos físicos en una oficina, lo que incluye papel, carpetas, bandejas, archiveros.

La metáfora de la organización global presentada en la pantalla es la del escritorio de una típica mesa de oficina. Los archivos, las carpetas se transforman en representaciones pictóricas en substitución de entidades abstractas con nombres arbitrarios.



Figura 1. Metáfora de las carpetas



¿Cómo funcionan estas metáforas? Las carpetas son contenedores de documentos en el mundo real y en el virtual: Puedes abrir una carpeta para tomar o dejar alguna cosa; se pueden poner carpetas dentro de carpetas y dentro de carpetas, podemos mover las carpetas por todo el escritorio. Algunas propiedades físicas están ausentes, no pesan no hacen ruido cuando se abren. Pero por otra parte tienen propiedades mágicas, se puede poner el mismo documento en dos carpetas al mismo tiempo, aunque se pueda pensar que se pueden pasar por una copiadora virtual, se puede reproducir un conjunto de carpetas y sus documentos automáticamente, se pueden ordenar las carpetas alfabéticamente, etc...

Actualmente es la metáfora dominante en las computadoras personales y las estaciones de trabajo.

3.2.1 Historia

Todo comenzó en los años '70 en el laboratorio Xerox Park en California. Un grupo de investigadores dirigidos por Alan Kay sintetizaron sus trabajos en la computadora Xerox Star, que fue caro, complejo y muy lento, y, como es fácil de entender, un fracaso comercial. Debido a este fracaso comenzó un éxodo de investigadores, principalmente a Apple y Microsoft. Steve Jobs y los refugiados de Xerox aplicaron los conceptos de la computadora Star a la computadora Lisa que aunque mejoraba al Star, todavía era caro y lento. Finalmente en 1984 apareció el Macintosh, alcanzando un éxito excepcional, debido a su facilidad de uso.

Posteriormente fue aplicado a las computadoras personales, Windows para MSDOS, Presentation Manager para OS/2 y en las estaciones de trabajo UNIX con los sistemas X-Windows, que tienen diferentes sistemas de ventanas como Open-Look y Motif.

3.2.2 Opiniones críticas

Nos han hecho creer que estamos trabajando con una metáfora de un escritorio, pero nunca se ha visto que apuntar a una pieza de papel que se encuentra mas abajo haga que suba encima de la pila, o que poner una hoja de papel en una carpeta haga que la hoja desaparezca dentro. Este tipo de oficina no existe .

¿Qué cosas son falsas en los sistemas actuales? Considerando la metáfora del escritorio que intenta simplificar operaciones de archivos comunes presentándolos en el lenguaje familiar del mundo basado en papel (documentos de papel como archivos, carpetas como subdirectorios, papelería de reciclaje). Aunque la metáfora ha tenido éxito hasta cierto punto (se le tiene que explicar a un nuevo usuario que el escritorio de la computadora es como un escritorio real), el modelo basado en papel es una base pobre para organizar información.

3.2.3 Metáforas compuestas

La metáfora del escritorio ha sido combinada con otras metáforas para permitir que los usuarios puedan ejecutar un amplio abanico de tareas en la computadora. Un ejemplo es la barra de desplazamiento, es una metáfora basada en la idea del rollo, es decir como un papiro que se desplaza para leer.

Otros ejemplos de metáforas son el menú, las ventanas y el *cortar* y *pegar* basadas en el diseño de páginas de una imprenta.

Desde un punto de vista cognitivo, uno de los problemas que se pueden presentar es cómo interpretar estas metáforas compuestas. De hecho se ha comprobado que el usuario desarrolla modelos mentales múltiples de la interfaz.

3.3 Metodología de creación de metáforas

Las metáforas visuales son un aspecto importante del diseño de un sistema interactivo con interfaz visual y evidentemente ha de formar parte del diseño, por tanto aunque se va a describir la metodología de diseño de metáforas, siempre se ha de tener en cuenta su relación con el diseño global.



3.3.1 Definición funcional

En esta primera fase se parte del trabajo realizado en la recopilación de requisitos, que permite disponer de los primeros datos para establecer las primeras metáforas o del análisis de tareas en que se pueden precisar las funcionalidades del sistema.

3.3.2 Identificación de los problemas del usuario

En esta etapa se hace un estudio de los usuarios para ver en qué tienen problemas y que aspectos de la funcionalidad les implican. La mejor manera es ver a los usuarios utilizando funcionalidades similares y ver qué problemas tienen. Explicar lo que quieren hacer y ver si lo entienden, enseñándoles el prototipo y observando como lo utilizan.

3.3.3 Generación de la metáfora

Una buena manera de empezar esta tarea es hacer un examen cuidadoso de la manera tradicional de realizar la tarea. Hay que perder tiempo en oficinas, fábricas, escuelas y observar los problemas que los usuarios tienen y qué herramientas utilizan para resolverlos en su trabajo diario y en su educación.

Una vez identificados los problemas y las herramientas que se utilizan, ver cuales de ellas envuelven algunas de las características que los usuarios encuentran difíciles de comprender. Esta son buenas candidatas para nuevas metáforas.

3.3.4 Evaluación de las metáforas

Una vez se han generado varias metáforas, es el momento de evaluar cual elegir para expresar la nueva funcionalidad. Aquí se presentan cinco puntos para evaluar la utilidad de una metáfora de la interfaz.

- **Volumen de la estructura.** Cuanta estructura nos proporciona la metáfora. Una metáfora con poca estructura será poco útil.
- **Aplicabilidad de la estructura.** Qué parte de la estructura aplicable es relevante para el problema. Lo que es importante no lo que sea irrelevante, sino lo que pueda llevar al usuario en la dirección incorrecta o le pueda hacer caer en falsas expectativas.
- **Representatividad.** ¿Es la metáfora fácil de representar? Las metáforas ideales tienen representación visual, auditiva y palabras asociadas.
- **Adaptabilidad a la audiencia.** Los usuarios tienen que entender la metáfora, porque aunque cumpla los otros criterios no sirve.
- **Extensibilidad.** ¿Qué más pueden hacer las metáforas? Una metáfora puede tener otras partes de la estructura que pueden ser útiles mas adelante.

3.3.5 Otra versión

Para establecer una metáfora, se deben seguir los siguientes pasos:

- 1) **Identificar el tipo de comparación.** Debemos encontrar una relación entre la información familiar y la nueva. Por ejemplo la metáfora del escritorio puede ser adecuada para la descripción del sistema de computadora. Los objetos en un escritorio se pueden coger, ubicar y manipular. En la computadora, los objetos (representados por iconos) que se asemejan a papeles y carpetas pueden ser manipulados. Coger un documento es equivalente a presionar el botón del ratón sobre el icono y arrastrar.

En general, un tono emocional en la metáfora incide sobre la actitud del usuario (deben ser compatibles). No todas las comparaciones son adecuadas. Ejemplo. *El sistema ha finalizado su actividad: metáfora de funeral o de parada.*



- 2) **Grado de ajuste.** Estudiar el grado de coincidencia y diferencias que existen. Ejemplo. *Máquina de escribir: Posición de teclas igual, pero no existe las opciones hacia atrás ni suprimir un carácter.*

El problema surge cuando no existe una relación completa entre la tarea conocida y la nueva. ¿Cómo se podría explicar mediante la metáfora de escritorio la expulsión de un disco en un Mac? No existe nada análogo. Además, la forma de realizarlo no es nada intuitiva, ya que se debe poner el disco sobre la papelera para expulsarlo (confuso).

3.4 Lenguaje Visual para el Diseño de Metáforas

Las metáforas pueden conseguir su efectividad a través de la asociación con organizaciones, podemos asociar estructuras, clases, objetos, atributos a nombres o operaciones, podemos asociar procesos y algoritmos a verbos.

3.4.1 Objetos familiares

Estas son algunas de las familias de objetos que pueden resultarnos más familiares:

- Escritorio: dibujos, archivos, carpetas, papeles, clips, notas de papel.
- Documentos: libros, capítulos, marcadores, figuras; periódicos, secciones; revistas, artículos; cartas; formularios.
- Fotografía: álbumes, fotos, portarretratos.
- Televisión: programas, canales, redes, anuncios comerciales, guías.
- Disco compacto, cassette, grabaciones, pistas, jukeboxes.
- Pila de cartas: cartas, pilas.
- Juegos: reglas del juego, piezas del juego, tablero de juego.
- Películas: rollos, bandejas de transparencias, presentaciones, rollos, películas, teatros.
- Contenedores: estanterías, cajas, compartimentos.
- Árboles: raíces, tronco, ramas, hojas.
- Red, diagrama, mapa: nodos, enlaces, hitos, regiones, etiquetas, bases (fondos), leyenda.
- Ciudades: regiones, hitos, caminos, casas, habitaciones, ventanas, mesas.

3.4.2 Verbos

Ejemplos típicos de conceptos de acción y las relaciones con los objetos:

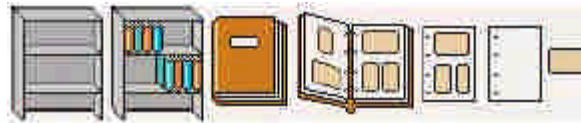
- Mover: navegar, conducir, volar
- Localizar: apuntar, tocar, enmarcar elemento(s)
- Seleccionar: tocar elemento, grabar elemento, poner dedo en elemento y moverlo
- Crear: añadir (nuevo), copiar
- Borrar: tirar, destruir, perder, reciclar, borrar (temporal o permanentemente)
- Evaluar: Mover botón, desplazar puntero, rodar, girar
- Vaciar, flujo: agua (tubos, ríos), electricidad

3.4.3 Ejemplo de construcción de una metáfora

En este ejemplo se plantea el diseño de un conjunto de metáforas para el mantenimiento de un archivo de fotografías digitales.

Los pasos que se seguirán son:

- 1) Elegir los objetos que están implicados. Se han elegido estos: estante, álbum, página, foto.
- 2) Asociar un elemento visual a los objetos anteriores. La asociación es:



- 3) Seleccionar los verbos asociados a las acciones que se pueden dar:
 - Añadir estantería, álbum, página, foto.
 - Borrar estantería, álbum, página, foto.
 - Mover álbum, página, foto.
 - Seleccionar álbum, página, foto.
- 4) Construir un elemento visual para las páginas anteriores:
 - Crear



- Borrar



- Mover



- Seleccionar



Con este último paso acaba el diseño de la metáfora del álbum de fotos.

3.5 Ejemplos de metáforas

- Cortar:



Las tijeras representan rápidamente su funcionalidad de cortar papeles, ropa, etc. Esta metáfora aporta la idea de cortar un trozo de documento, una parte de un dibujo, etc.

- Papelera:

La papelera es una herramienta habitual en la mayoría de las culturas avanzadas. Sirve para poner todos los papeles o otros elementos que no sirven, para después tirarlo como basura. El símbolo de reciclaje se está haciendo habitual en nuestra cultura. Esto nos permite ver que tenemos una papelera que permite reciclar los objetos que ponemos en ella.





➤ Pintar:

El bote de pintura es una herramienta muy común y de fácil comprensión. En esta metáfora del bote de pintura que se esta vaciando se desea hacer comprender al usuario que lo que hace es llenar de color el interior de un determinado objeto.



➤ Portapapeles:

El portapapeles es un componente del sistema operativo que nos permite pasar información entre aplicaciones, es decir pasar información entre aplicaciones diferentes.

Es fácil de intuir como utilizar el portapapeles, pero si nos limitamos estrictamente a la metáfora es muy pobre. No se puede recoger más de una cosa, no se sabe de donde vienen los documentos u objetos, no sabe todo lo que ha ido recibiendo. Todas estas posibilidades no están cubiertas por la metáfora y por tanto tienen que ser aprendidas. Esta metáfora, presenta problemas de cobertura.

➤ Correo:

El correo es un elemento habitual en nuestra cultura, sirve para enviar información escrita con papel a un destinatario normalmente lejano. Utilizamos esta metáfora para la mensajería de correo electrónico.



3.6 Interacción

Las interacciones son todos los intercambios que suceden entre la persona y la computadora.

3.6.1 Interacción multimodal

La mayoría de los sistemas actuales interactúan a través de un teclado y una pantalla y normalmente también un ratón.

Cada uno de estos dispositivos se puede considerar canales de comunicación del sistema y corresponden con ciertos canales de comunicación humanas (tacto, vista, etc..)

Una interacción es multimodal cuando usa múltiples canales de comunicación simultáneamente. Cada canal del usuario se puede considerar una modalidad diferente de interacción. Los sistemas actuales tienden a tener múltiples canales de comunicación de entrada/ salida. Los seres humanos procesan la información simultáneamente por varios canales. Por ejemplo se puede ajustar el movimiento de un ratón mediante la voz.

3.6.2 Estilos de Interacción

Un estilo de interacción es un término genérico para agrupar las diferentes maneras en que los usuarios se comunican o interactúan con la computadora.

Los estilos de interacción predominantes son:

- 1) La interfaz por línea de comandos
- 2) Menús y formularios
- 3) Manipulación directa
- 4) Interacción asistida



3.6.2.1 Interfaz por línea de comandos

La interfaz por línea de comandos fue el primer estilo de interacción de uso generalizado y a pesar de todos los estilos de interacción que se describen después, todavía es muy usado.

Es una manera de dar instrucciones directamente a la computadora. Pueden tener la forma de teclas de función, un carácter, abreviaciones cortas, palabras enteras o una combinación de las dos primeras. La ventaja de utilizar un solo carácter o una tecla de función es que presionando una o dos veces la tecla se ejecuta la orden. La desventaja es que es más difícil de recordar que un nombre bien elegido. Dar a las órdenes nombres apropiados es importante porque ayuda a recordar a qué hacen referencia.

La interfaz por línea de comandos es potente porque ofrece acceso directo a la funcionalidad del sistema. También es muy flexible: la orden normalmente tiene una serie de opciones o parámetros para modificar su comportamiento y puede ser aplicado a muchos objetos a la vez haciéndolo útil para tareas repetitivas.

No obstante, esta flexibilidad y potencia es al mismo tiempo el problema de su dificultad de aprendizaje. Para poder escribir un comando en la línea hace falta tenerla memorizada porque no hay una indicación visual de la orden que se necesita. Es evidente por tanto que son mas útiles para usuarios expertos que para usuarios noveles.

Este problema puede reducirse utilizando nombres de comandos con sentido y consistentes aunque normalmente debido a los estándares establecidos en el inicio estos nombres normalmente son crípticos y varían entre diferentes sistemas.

Las interfaces orientadas a comandos más conocidas son:

- El shell de UNIX
 - \$ ls -la
 - \$ mv mmail
- MSDOS
 - C:>pwd
 - Comando o nombre de archivo incorrecto
 - C:>ls -la *.c
 - Comando o nombre de archivo incorrecto
 - C:>dir *.c

Sus ventajas son:

- Flexibilidad
- Permite la iniciativa del usuario
- Es atractivo para usuarios expertos
- Potencialmente rápido para tareas complejas
- Capacidad para hacer macros

Sus desventajas son:

- Requiere memorización y entrenamiento importante
- Difícil de memorizar
- Gestión de errores pobre

3.6.2.2 Menús y navegación

Un menú es un conjunto de opciones visualizadas en la pantalla, que se pueden seleccionar y la selección de una de ellas o más supone la ejecución de una orden subyacente y normalmente un cambio en el estado de la interfaz. A diferencia de la interfaz por línea de comandos, los usuarios tienen la ventaja de no tener que recordar ni palabras, ni sintaxis, siempre y cuando los textos que acompañan a los menús sean significativos, lo que no siempre se produce.



Uno de los problemas que tienen los menús es que ocupan mucho espacio en la interfaz. Para resolver el problema surgieron los menús desplegables y menús de contexto (pop-up). En el primer caso el menú consiste en un texto inicial que define un conjunto de opciones y al interactuar con él despliega la lista de opciones hacia abajo y en el momento que seleccionamos un elemento el menú vuelve al estado inicial y el sistema ejecuta la acción correspondiente. Los menús de contexto aparecen a partir de una acción realizada en un punto cualquiera de la pantalla hasta que se selecciona una acción o se desactiva.

Los menús se pueden seleccionar mediante una acción del ratón o a través de una combinación de teclas. Existe una opción estandarizada, por ejemplo, de pulsar la tecla *Alt* para acceder al menú y después utilizar las letras que estén subrayadas en el menú.



Figura 2. Acceso a menú por teclado

La estructura de un menú de una aplicación se organiza de forma jerárquica, existen guías de estilo que explican cómo hacer la estructura básica de un menú. El número de opciones ideal en un menú es entre 3 y 8.

3.6.2.3 Manipulación directa

La interfaz por línea de comandos nació de los teletipos y la navegación por menú de las pantallas de texto. Las pantallas gráficas de alta resolución y los dispositivos de apuntar como el ratón han permitido la creación de los entornos de manipulación directa que crean una representación visual del *mundo de las acciones* que incluye visualización de objetos y acciones de interés. Estas ideas fueron aplicadas a la computadora Xerox Star y posteriormente al Apple Macintosh que fue la primera computadora de manipulación directa que tuvo éxito comercial.

El término *manipulación directa* describe sistemas que tienen las siguientes características:

- Representación continua de los objetos y de las acciones de interés
- Cambio de una sintaxis de comandos compleja por la manipulación de objetos y acciones
- Acciones rápidas, incrementales y reversibles que provocan un efecto visible inmediatamente en el objeto seleccionado.

Típicamente, los sistemas de manipulación directa tienen iconos representando objetos que pueden ser movidos por la pantalla y manipulados controlando un cursor con un ratón. Las ventajas de un sistema de manipulación directa las explica BEN SCHNEIDERMAN, profesor de la Universidad de Maryland, quien acuñó dicho término.

- Los nuevos usuarios pueden aprender las funcionalidades básicas más rápido.
- Los usuarios experimentados pueden trabajar rápidamente para hacer una amplia variedad de tareas.
- Los usuarios intermitentes pueden memorizar los conceptos operacionales. Los mensajes de error son poco necesarios.
- Los usuarios pueden ver inmediatamente si sus acciones responden a sus objetivos y no cambiar.
- Los usuarios tienen menos ansiedad porque el sistema es comprensible y porque las acciones son reversibles.
- Los usuarios tienen confianza porque cuando inician una acción, controlan y pueden predecir la respuesta.



Uno de los problemas con la manipulación directa es que no todas las tareas pueden ser descritas por objetos concretos y no todas las acciones se pueden hacer directamente. Por ejemplo, uno de los primeros problemas que se presentaron fue cómo representar el concepto de buffer, que se solucionó utilizando el concepto de *cortar y pegar*.

La interfaz WIMP

Éste es, en la actualidad, el entorno más común de manipulación directa. WIMP quiere decir Ventanas, Iconos, Menús y Apuntadores (*Windows, Icons, Menus and Pointers*).

- **Ventanas.** Las ventanas son áreas de la pantalla que se comportan como si fuesen terminales independientes, permitiendo que tareas independientes puedan verse al mismo tiempo. Los usuarios pueden dirigir su atención a las diferentes ventanas. Si una ventana se coloca sobre la otra, la segunda queda cubierta por la primera, la cual se recupera al volverla a seleccionar.
- **Iconos.** Las ventanas se pueden cerrar y perderse o reducirlas a un icono. Esto permite tener muchas ventanas en la pantalla. EL icono ahorra espacio en la pantalla y sirve de recordatorio al usuario que puede continuar el diálogo restaurando la ventana a su tamaño original. Los iconos también se pueden utilizar en otros aspectos del sistema como una papelera para enviar archivos que se quieren borrar. Los diseños de los iconos pueden ser muy variados, pero lo importante es que su representación sea significativa.
- **Apuntadores.** El apuntador es un componente muy importante de la metáfora WIMP, ya que la manipulación directa se apoya en apuntar y seleccionar objetos. Tal como se ha comentado previamente el ratón, el joystick y el trackball permiten hacer estas tareas. El usuario ve un cursor en la pantalla que se controla con el dispositivo apuntador.
- **Menús.** Presenta una lista alternativa de operaciones o servicios a seleccionar. Los menús nos dan indicaciones de intenciones en forma de una lista. Esto representa que los nombres de las órdenes en el menú han de tener significado y ser informativas. El dispositivo de selección sirve para indicar la opción deseada. Cuando se mueve el apuntador a la posición de un ítem, este es modificado por un cambio de color o de vídeo inverso para indicar que es el candidato potencial a ser seleccionado. Finalmente la selección requiere otra acción, por ejemplo un botón del ratón. Los menús normalmente disponen de aceleradores de teclado, que son combinaciones de teclas que tienen el mismo efecto que seleccionar el menú. Esto permite que usuarios expertos familiarizados con el sistema, trabajen mas rápido.

3.6.2.4 Interacción asistida

Somos conscientes que la computadora se está utilizando en un conjunto muy amplio de actividades diarias: búsqueda de información, correo, entretenimientos, etc. Al mismo tiempo que la computadora se hace más habitual en la vida diaria un número creciente de nuevos usuarios comienza a utilizarlo y este número continuará creciendo, aunque desgraciadamente la manera como la gente interacciona con las computadoras no responde a estas necesidades.

El estilo de interacción dominante, el de manipulación directa, requiere que el usuario indique explícitamente todas las tareas y controle todos los eventos. Este estilo ha de cambiar si queremos que los nuevos usuarios hagan un buen uso de las computadoras y redes del futuro.

Actualmente comienza a utilizarse un nuevo estilo de interacción denominado *interacción asistida* que utiliza la metáfora del asistente personal o agente, que colabora con el usuario en el mismo ambiente de trabajo y el usuario en vez de dirigir la interacción, trabaja en un entorno cooperativo en que el usuario y los agentes o asistentes se comunican, controlan eventos y realizan tareas.

Este tipo de interacción permitirá reducir el esfuerzo necesario para realizar tareas, debido a que en el caso de la manipulación directa para realizar una tarea hay que seleccionar objetos y seleccionar acciones. En el caso de la interacción asistida se pueden provocar cambios en los objetos que no corresponden una por una con las acciones del usuario. La idea de utilizar agentes en la interfaz fue introducida por visionarios como Negroponte



y Alan Kay. También la empresa Apple produjo un vídeo donde exponía su visión de la computadora del futuro con un agente antropomórfico de soporte.

3.7 Agentes

Según Henry Lieberman un agente es un programa que puede ser considerado por el usuario como un asistente o programa que le ayuda y no se le considere una herramienta desde el punto de vista de una interfaz de manipulación directa.

Un agente tiene que presentar algunas (quizás no todas) las características que asociamos con la inteligencia humana: Capacidad de aprender, inferencia, adaptabilidad, independencia, creatividad, etc. Según Pattie Maes el usuario delega una tarea a un agente mas que ordena al agente realizar la tarea.

Las interfaces de manipulación directa visualizan representaciones de objetos físicos o conceptuales y permiten a los usuarios hacer acciones que cambien el estado de los objetos. En una interfaz de manipulación directa los cambios en el estado de la pantalla son más o menos, una a una, con las acciones hechas explícitamente por el usuario.

Un agente de la interfaz es como un programa que también puede afectar los objetos de una interfaz pero sin instrucciones explícitas del usuario. El agente de la interfaz lee la entrada que el usuario presenta en la interfaz y puede hacer cambios en los objetos que el usuario ve en la pantalla, aunque no necesariamente una-a-una con las acciones del usuario. El agente puede observar muchas interacciones del usuario antes de hacer una acción o con una sola interacción puede lanzar una serie de acciones y actuar en determinados períodos de tiempo que le hemos fijado.

Los agentes son más discretos que los asistentes, trabajan en segundo plano y actúan por propia iniciativa, cuando encuentran información que puede ser relevante para el usuario. Se pueden añadir agentes especializados a entornos existentes.

Las características de los agentes son:

- Autonomía. Trabajan en segundo plano y si no se les pide explícitamente observan al usuario y a las fuentes de información accesibles.
- Inteligencia. Actúan por su propia iniciativa y pueden trabajar en entornos heterogéneos adaptándose a múltiple situaciones (no necesariamente utilizan la misma estrategia de resolución cada vez).
- Uso personal. Se adaptan y aprenden del usuario y no insisten en una determinada solución si el usuario decide otra (la reglamentación por parte del usuario es una característica de los agentes inteligentes). Los agentes pueden tomar decisiones en situaciones que no son cómodas para el usuario, pero nunca han de forzar a tener un cierto comportamiento.

La implementación de agentes es una tarea difícil, que se puede realizar en entornos orientados a objetos hasta con técnicas que utilicen sistemas basados en conocimientos o aprendizaje (por ejemplo sistemas expertos o redes neuronales). Estas técnicas son especialmente importantes para modelar entornos en tiempo real o de sentido común. Esta habilidad hará que los agentes sean útiles para usuarios no técnicamente entusiastas.

Los agentes son el camino para que el conocimiento de fuentes de información se puedan utilizar efectivamente, considerando la complejidad de los nuevos sistemas de información. La idea de utilizar agentes no es nueva. Hace más una década, Alan Kay, Marvin Minsky y otros ya trabajaban en el tema, pero hoy día es una necesidad real.

Con el uso de los agentes, los usuarios se liberan de muchas tareas rutinarias —como la de hacer copias de seguridad o la búsqueda de noticias o determinados temas— evitando fuentes de errores como resultado de la pereza natural hacia las necesidades no humanas de la computadora.

Los agentes permitirán hacer las computadoras usables y útiles para personas que no están motivadas por la tecnología. Un aspecto que los asistentes y agentes tienen en común es la multitarea, porque normalmente trabajan en segundo plano o sin preferencia.



3.7.1 Integración de agentes con otras aplicaciones

Veamos cómo usar mecanismos de comunicación entre aplicaciones para implementar agentes que sean independientes de aplicaciones.

Nos centraremos en agentes que aprenden observando las acciones del usuario con la interfaz y que son capaces de generar procedimientos generalizados de automatizar tareas para el usuario. Estos mecanismos de comunicación actuales están lejos de ser una plena cooperación agente–aplicación.

Inicialmente se describen las propiedades que deben tener las aplicaciones para poder interactuar con agentes.

- **Escripable.** Una aplicación es escripable si nos da un medio (a través de un lenguaje script o mediante un API a un agente externo de llamar las órdenes de la aplicación.
- **Recordable.** Una aplicación es recordable si es capaz de informar a un agente externo que el usuario es capaz de realizar una función por menú, por icono o por teclado.
- **Examinable.** Una aplicación es examinable si se pueden revisar periódicamente las estructuras de datos de la aplicación y tratar de inferir las acciones que se están realizando con la interfaz de usuario comparándolo con otros estados de las estructuras de datos.

3.7.2 Aspectos a considerar en la integración agente–aplicación

Granularidad del protocolo de eventos. En el diseño de la aplicación hace falta ponerse de acuerdo con el conjunto de eventos que la aplicación ha de pasar al agente y que el agente ha de aceptar.

Compartir la interfaz. Otro aspecto a considerar es la necesidad de que el agente pueda compartir o modificar la interfaz.

3.7.3 Ejemplo: El agente de Microsoft

Tecnología ActiveX® para agentes interactivos.

Microsoft Agent permite incorporar una nueva forma de interacción conocida como interfaces conversacionales que incorporan aspectos de la comunicación social.

Además la entrada por teclado y ratón, incluye un soporte opcional para el reconocimiento de voz, de forma que las aplicaciones puedan responder a órdenes de voz. Los agentes pueden responder utilizando voz sintetizada, voz grabada o texto en forma de llamada de cómic.

3.8 Asistentes

Los asistentes son entidades computacionales que nos asisten en el uso de las aplicaciones existentes. Los asistentes exponen de una manera fácil qué se ha de hacer y pueden entender palabras escritas o habladas o acciones gráficas e interpretarlas. Interpretar quiere decir que el asistente puede hacer acciones complejas u ordenes cortas. Un requerimiento importante para los asistentes es que han de ser muy flexibles en la forma en que reciben las instrucciones. Los asistentes son muchas veces mas flexibles que los menús y las macros porque el usuario nada mas dice lo que quiere hacer. Los asistentes necesitan mucha información durante las interacciones gráficas del usuario y en consecuencia habrían de ser capaces de aprender del usuario. El usuario activa un asistente seleccionando órdenes y accionando un botón de asistencia o haciendo una acción gráfica encima de él.

Los asistentes son bastante habituales en las aplicaciones actuales y a veces alguna aplicación tiene mas de uno.

3.8.1 Ejemplo de asistente



Un ejemplo de asistente es el que permite a través de la realización de una serie de tareas la conversión de una presentación PowerPoint a una presentación basada en WEB en formato HTML. Dispone de una serie predefinida de formatos, se elige uno y finalmente deja toda la información en un directorio que se selecciona.

3.8.2 Computadoras emocionales

Los últimos resultados científicos indican que las emociones desempeñan un papel esencial en la toma de decisión, opinión, aprendizaje, etc. —es decir, influyen los mismos mecanismos del pensamiento racional. Demasiada o poca emoción puede deteriorar la toma de decisión. Según Rosalind Picard, si se quiere que las computadoras sean genuinamente inteligentes y que trabajen naturalmente con los humanos, se les debe dar a las computadoras la capacidad de reconocer, de entender, tener y expresar emociones.

3.9 Paradigmas

La palabra paradigma viene del latín *paradigma* y este del griego *parádeigma* que quiere decir ejemplo, modelo y en este aspecto los paradigmas de interacción representan los ejemplos o modelos de los que se derivan todos los sistemas de interacción.

Es una abstracción de todos los posibles modelos de interacción organizados en grupos con características similares.

Los paradigmas interactivos actuales son:

- 1) La computadora de escritorio
- 2) La realidad virtual
- 3) La computación ubicua
- 4) La realidad aumentada

3.9.1 Computadora de escritorio

Actualmente es el paradigma dominante. La interacción se realiza normalmente aislado del entorno, el usuario básicamente interacciona sentado en una mesa con una computadora de escritorio de manipulación directa.

Hasta hoy día las computadoras personales no se corresponden con su nombre. La mayor parte de las computadoras están puestos en una mesa e interaccionan con sus propietarios, nada más una pequeña fracción del día. Las computadoras portátiles han hecho la movilidad un poco mas factible, pero continúan utilizando el mismo paradigma, aunque comienzan a modificar el escenario.

3.9.2 Entornos virtuales y realidad virtual

Los términos entorno virtual y realidad virtual describen una amplia variedad de estilos de interacción que van desde interfaces en tres dimensiones con las que se puede interactuar y actualizar en tiempo real hasta sistemas en que el nivel de autonomía, interacción y sensación de presencia es prácticamente igual al del mundo real.

Las condiciones para hablar realmente de un sistema de realidad virtual son las siguientes:

- 1) Sensación de presencia física directa. El usuario tiene una sensación de presencia física o experiencia directa mediante indicaciones sensoriales creadas por la tecnología. Estas indicaciones normalmente visuales pueden ser también de audio o táctiles o una combinaciones de las tres.
- 2) Indicaciones sensoriales en tres dimensiones. La información normalmente visual se presenta en tres dimensiones.
- 3) Interacción natural. Normalmente los sistemas de realidad virtual permiten manipular los objetos virtuales utilizando los mismos gestos que se utilizan para manipular objetos reales: Tomar objetos, girar, tirar, etc..



Una de las formas mas comunes de interacciones virtuales es mediante el uso de cascos y ratones 3D. El casco permite la visualización tridimensional y el posicionamiento mediante los 6 grados de libertad y el ratón permite la interacción mediante su visualización como una mano virtual que podemos mover y realizar operaciones de selección y acciones como abrir puertas, tomar objetos mediante el uso de los botones del ratón.

El dataglove permite ampliar y mejorar estas acciones añadiendo la posibilidad de realizar acciones con las posiciones de los dedos. El uso de dispositivos táctiles añade a estas posibilidades el poder experimentar fuerza sobre los objetos.

3.9.2.1 La cueva (CAVE)

La cueva es un espacio cúbico de 3x3x3 metros, delimitado por tres paredes (una frontal y dos laterales). Las tres paredes y el suelo son pantallas de proyección: en las paredes es retroproyección y en el suelo es proyección frontal del techo. Las imágenes proyectadas son en tres dimensiones y es preciso utilizar las gafas de cristal líquido con tal de percibir un efecto de estereoscopia. El usuario se puede mover dentro de este espacio mirando las imágenes proyectadas (en las paredes y en el suelo) y por esta razón, es preciso que tenga un sensor de posición y orientación que permita a la computadora saber dónde está el usuario. De esta forma, la computadora puede generar las imágenes con la proyección perspectiva correcta para el plano de proyección, de forma que la percepción estereoscópica sea efectiva.

El usuario interactúa con los entornos mediante un dispositivo tipo *joystick aéreo* denominado *wand*, con una serie de botones y un sensor de posición, orientación.

El espacio también está provisto de un sistema que genera sonido cuadra fónico. La característica más importante de la cueva es que es un sistema multiusuario, es decir hasta cinco personas con gafas se pueden mover dentro de este espacio. Un usuario dirige la interacción mediante el uso del *wand*.

3.9.2.2 Beneficios y problemas

- Simulaciones imposibles en otro estilo
- Alto coste
- Cansancio del usuario

3.9.2.3 Ejemplos de uso de la realidad virtual:

- Entrenamiento de operarios de una central nuclear
- Entrenamiento de bomberos
- Reconstrucciones virtuales de patrimonio histórico

3.9.3 Computación ubicua

En 1991, Mark Weiser (Xerox PARC) ilustró un nuevo paradigma de interacción publicando un artículo sobre su visión acerca de la *Computación Ubicua*. En su visión, Weiser explica que la *computación ubicua* trataría de extender la capacidad computacional al entorno del usuario, permitiendo que la capacidad de información esté presente en todas partes en forma de pequeños dispositivos muy diversos que permiten interacciones de poca dificultad, conectados en red a servidores de información. El diseño y localización de estos dispositivos deben ser ideados especialmente para la tarea objeto de interacción. La computación, por tanto, deja de estar localizada en un único punto para pasar a diluirse en el entorno. La computadora queda delegada a un segundo plano, intentando que resulte lo más transparente posible al usuario. Esta idea suele referirse con el término de "omnipresencia" de la computación. Alan Kay denomina a la computación ubicua el tercer paradigma.

Por tanto ya no existirá nunca más una estación de trabajo y una sola pantalla donde interactuar, sino una serie de visualizaciones por todas partes permitiendo interacciones de poca dificultad. Los dispositivos necesarios para acceder a este sistema estarían disponibles en cualquier parte.



Descrita de esta manera la computación ubicua no es la realidad virtual con la que a veces se confunde. La realidad virtual trata de crear un mundo virtual no existente en favor de uno imaginario, la computación ubicua por contra forma parte del mundo real y lo mejora.

Respecto al paradigma del escritorio, Donald Norman en su libro *The invisible computer* insiste en la idea de que la computadora personal es probablemente la tecnología mas frustrante jamás fabricada. Propone que la computadora tiene que ser vista como una infraestructura. Tiene que ser silenciosa, invisible y no obstructiva pero es demasiado visible, demasiado exigente y controla nuestro destino.

Su complejidad y sus frustraciones son debidas a la concentración de demasiadas funciones en una caja que está en nuestra escritorio.

El paradigma de computación ubicua está inspirado en el acceso constante a la información y en las numerosas capacidades computacionales.

Dada la variedad de dispositivos existentes hoy en día —insignias activas, marcas, WAPs palmtops, tabletas, pizarras electrónicas, entre otros—, podemos decir que la visión de Weiser ha pasado a ser una realidad tecnológica. Así, podemos hablar hoy en día de entornos en los que los usuarios no interaccionan directamente con computadoras, sino que opera en un entorno integrado coherente que incorpora computadoras y dispositivos de interacción de todos los tamaños y tipos. Avances recientes en comunicaciones sin hilos y computadoras móviles podrán extender la computación ubicua a cualquier sitio, a cualquiera, y en cualquier momento.

Los avances en hardware que se han producido no son suficientes para que se realice el cambio de paradigma de el escritorio a la computación ubicua. No podemos simplemente adaptar entornos de software realizados para soportar una persona, una pantalla, un entorno de computadora (ya sea escritorio, portátil o PDA). El problema es crear un entorno de software/hardware que soporte la multi-persona, el multi-dispositivo, diferentes lugares de una manera integrada, hecho a partir de la tecnología existente pero aportando coherencia sobre el entorno como un todo.

Esto implica al mismo tiempo integración del sistema de forma que todos los dispositivos puedan ínter operar y una integración de la interacción.

3.9.4 Realidad aumentada

La Realidad Aumentada (RA) es un paradigma de interacción que trata de reducir las interacciones con la computadora utilizando la información del entorno como una entrada implícita. Con este paradigma el usuario será capaz de interactuar con el mundo real, el cual aparece aumentado por la información sintética de la computadora. Con ello se consigue integrar los dos mundos (el real y el computacional), obteniendo como resultado una disminución importante del costo interactivo. La situación del usuario será automáticamente reconocida utilizando un amplio conjunto de métodos de reconocimiento.

Con todo esto se puede afirmar que con el paradigma de la RA se consigue asistir y mejorar la interacción entre los humanos y el mundo real. Permite la integración del uso de la computadora en la mayoría de las actividades de la vida cotidiana, posibilitando el acceso a usuarios diversos y no especializados, dado que los objetos de la vida cotidiana se convierten en verdaderos objetos interactivos.

Así, la RA permite al usuario permanecer en contacto con su entorno de trabajo de forma que su foco de atención no esté en la computadora, sino en el mundo real, refiriéndonos a éste como *mundo real aumentado*. Explotando las habilidades visuales y espaciales de los usuarios, la RA traslada información adicional al mundo real, en vez de introducirlos en el mundo virtual de la computadora. Se pueden utilizar métodos de reconocimiento como tiempo, posición o reconocimiento de objetos utilizando la visión por computadora. También se puede hacer más comprensible el mundo real para la computadora usando, por ejemplo, códigos de barra.

El papel que juega la computadora es el de asistir y mejorar las relaciones e interacciones entre las personas y el mundo real. Este paradigma de la interacción y visualización de información constituye el centro de una nueva y muy prometedora tecnología para muchas aplicaciones. La RA puede ser muy útil en multitud de



sectores como medicina, arquitectura, diseño interior, construcción, ingeniería civil, diseño de automóviles, mantenimiento mecánico y reparación, etc. De todas formas, las aplicaciones reales para RA imponen una fuerte demanda de tecnología que todavía no se ha alcanzado. En qué se diferencia la realidad aumentada de la computación ubicua? La diferencia es muy clara. Mientras que la computación ubicua continua utilizando dispositivos informáticos de bajo coste interactivo, la realidad aumentada utiliza la propia realidad para interactuar, sin incurrir en la necesidad de adoptar nuevos procedimientos de trabajo. Todo lo contrario, se mantienen los métodos tradicionales que garantizan la familiarización del usuario con el entorno.

3.9.4.1 Corrientes existentes

Existen dos corrientes importantes en este nuevo paradigma. Por un lado el derivado de aplicar la realidad virtual en el mundo real, y por otro lado el uso de dispositivos que aumentan la realidad e interaccionan directamente con ella.

En el primer caso se trata, pues, de una nueva tecnología que aumenta o mejora la visión que el usuario tiene del mundo real con información adicional sintetizada mediante un modelo computerizado, las cuales se superponen mediante el uso, generalmente, de unas gafas especializadas. Los usuarios pueden trabajar y examinar objetos 3D reales mientras reciben información adicional sobre estos objetos o sobre la tarea que se está realizando.

En el segundo caso el usuario será capaz de interactuar con el mundo real, el cual estará aumentado por la información sintetizada por la computadora. No se trata de superponer la información real con la virtual, como es el caso de la anterior, sino de hacer participar diversos objetos de la vida cotidiana – ya sea un bolígrafo o un bloc de notas – que automáticamente interaccionan con el sistema sin que para ello sea necesario realizar ninguna acción específica. Esto es importante puesto que evitamos que el usuario tenga que familiarizarse con un nuevo entorno de trabajo.

3.9.4.2 Líneas de trabajo

Superficies interactivas. Transformación de la superficie dentro de un espacio arquitectónico (paredes, mesas, puertas, ventanas) como una interfaz activa entre el mundo físico y el mundo real.

Acoplamiento de bits y átomos. Acoplamiento sin interrupciones entre los objetos de cada día que se pueden tomar (tarjetas, libros, modelos) con la información digital que está relacionado con ellos.

Medio ambiente. Uso del medio ambientes como el sonido, la luz, corrientes de aire o movimientos de agua como una interfaz de fondo.

Ejemplos: La realidad aumentada en esta segunda línea es un tema muy nuevo y no hay muchas implementaciones realizadas. Existen algunos ejemplos descritos en el artículo de HIROSHI ISHII del MIT que describe tres ejemplos: Metadesk, Ambient room y Transboard.

3.9.5 Diferencias entre Realidad Aumentada y Realidad Virtual

La Realidad Virtual (RV) ha sido un tema de gran interés en los últimos tiempos. Menos atención se ha puesto en el campo de la Realidad Aumentada (RA) a pesar de su similar potencial. En los sistemas de RV el usuario está completamente inmerso en un mundo artificial y no hay manera alguna de interactuar con objetos del mundo real. En contraposición con la RV, en la tecnología de la RA los usuarios pueden interactuar con una mezcla de mundo real y virtual de una forma natural.

Así, la diferencia entre RV y RA está en el tratamiento que hacen del mundo real. La RV sumerge al usuario dentro de un mundo virtual que completamente reemplaza el mundo real exterior, mientras que la RA deja al usuario ver el mundo real alrededor de él y aumenta la visión que éste tiene de su entorno mediante la superposición o composición del objetos 3D virtuales. Idealmente esto daría la ilusión al usuario de que los objetos del mundo real y virtual coexisten.



Se podría decir que los sistemas de RA llevan la computadora al entorno de trabajo real del usuario, mientras que los sistemas de RV intentan llevar el mundo al interior de la computadora. Dicho de otro modo, la RA lleva información dentro del mundo real del usuario en vez de llevar al usuario dentro del mundo virtual de la computadora.

3.9.6 Computadoras corporales

La computadora debe poder llevarse encima, tal y como se llevan unas gafas o un vestido, y debe interactuar con el usuario según el contexto en el que se encuentre.

El objetivo de la computadora corporal debe ser el conseguir enlazar la información del espacio personal del usuario (su entorno), con la propia información de un sistema informático. Para conseguirlo, debe ofrecer toda su funcionalidad de forma natural y sin obstáculos (cómoda), permitiendo al usuario concentrar toda su atención en la tarea que está realizando sin que sufra distracciones por la utilización del propio sistema.

Una computadora corporal puede usarse sin tener que emplear las manos (o reduciendo al mínimo esa interacción con la utilización de dispositivos de entrada especiales, y sin necesidad de desviar la mirada para ver la pantalla del dispositivo electrónico. La computadora corporal responde a órdenes habladas (reconocimiento de voz), se integra con la vestimenta, y la pantalla donde se visualiza la información está colocada de tal manera que no obliga a desviar la vista del trabajo que se está realizando.

En la actualidad, los chips van siendo cada vez más pequeños y baratos, pero a su vez más potentes. Uno de los desafíos de los científicos, todavía lejos de conseguir, es hacer la computadora corporal invisible a la vista, integrándolo totalmente en la vestimenta humana o en complementos a la misma (relojes, pendientes, gafas, zapatos, etc.). Es decir, en cualquier sitio en el que un par de centímetros se encuentren disponibles.

Las computadoras corporales abrirán la puerta de cientos de posibles aplicaciones, permitiendo al usuario escapar de la oficina y aventurarse al mundo exterior sin tener que llevar su computadora portátil. Médicos, secretarías, hombres de negocios, padres, profesores, policías, bomberos y estudiantes son unos pocos ejemplos de grupos para los que se están buscando las posibilidades de aplicación de las computadoras corporales.

3.9.7 Dispositivos utilizados en RA

La siguiente lista de dispositivos no tiene porqué estar presentes en todas las aplicaciones de la tecnología de la RA, pero sí que se podría decir que son específicos o indicativos del uso de RA. Dependerá de cada uso concreto, los dispositivos que se usarán y la configuración de los mismos.

Un equipo estándar podría ser el siguiente: computadora (a ser posible con aceleración de gráficos 3D), un sistema GPS diferencial, unas gafas o un sistema de visión montado en casco (con seguidor de orientación) o head-trackers, etc.

El problema clave en la RA es el del registro de los objetos. Se ha de almacenar la información sobre el objeto virtual para superimprimirla posteriormente sobre el objeto. En los sistemas típicos de RA se utilizan seguidores del movimiento de la cabeza para procesar la orientación conjuntamente con sensores de sonar y scanners para detectar los objetos reales. El problema principal es la falta de sensibilidad, la mayoría de "head-trackers" comerciales no proporcionan la suficiente.

Uno de los dispositivos que suelen estar presentes y que dará una idea más clara de lo que implica la RA implica son las:

Gafas montadas en casco (see-through head-mounted display). En el espíritu de las primeras "see-through head-mounted display" desarrolladas por SUTHERLAND. Como su nombre indica van montadas sobre la cabeza y permiten mostrar gráficos superimpuestos sobre lo que puede ver la persona que las lleva. Se utilizan en computadoras corporales y en aplicaciones tales como de construcción, de mantenimiento mecánico y reparación, etc. Normalmente incorporan un seguidor de orientación "head-tracker".



3.9.8 Aplicaciones: Algunos escenarios de RA

Mencionar todas las áreas de aplicación de la tecnología de la RA sería un tarea interminable, por este motivo, se expone una lista incompleta de escenarios de aplicación en los que la RA se prevé que será muy beneficiosa. Algunos de estos escenarios ya son una realidad hoy en día.

3.9.8.1 El fontanero del futuro

Suponiendo que es el año 2025 y que cierta compañía quisiera remodelar una habitación para colocar un fregadero de cocina. Para instalar las tuberías necesarias de agua caliente y fría se contrata a un fontanero y éste visita la compañía para hacer los planos requeridos y presupuestar la obra.

El "fontanero del futuro" llega llevando consigo una pequeña computadora y después de discutir dónde se colocará el fregadero, abre la maleta de su computadora personal y retira lo que parecen ser unas gafas de sol gigantes. Las gafas de sol están conectadas con un cable a la computadora personal. Cuando se pone las gafas, éstas actúan exactamente como unas gafas de sol normales, es decir, él ve una versión oscurecida de la habitación alrededor suyo. Pero después de pulsar unas cuantas teclas en su computadora, cuando mira de nuevo alrededor, además de la habitación ve unos gráficos sobreimpresos de muchos objetos que normalmente son invisibles.

Así, él ve unas líneas amarillas dentro de los muros que representan la localización de líneas de corriente y líneas verdes que muestran por donde circulan las líneas de teléfono. También ve representaciones de varias estructuras de acero de soporte concretas dentro de los muros tanto como la localización de los conductos de calor y aire acondicionado.

Después de mirar alrededor de la habitación el fontanero no puede ver lo que está buscando, así que vuelve a teclear unos comandos en su computadora.

Ahora es capaz de mirar a través de los muros y ver la disposición de las habitaciones adyacentes. También encuentra lo que estaba buscando: dos gruesas líneas azules y rojas que representan las tuberías principales de agua caliente y fría a través del edificio. En este momento, él, que también está conectado por un cable a su computadora personal, hace una selección mediante un dispositivo apuntador. Debido a que el dispositivo físico de apuntar no puede ser proyectado físicamente a través de un muro, el fontanero ve una versión extendida del apuntador generada por la computadora y que puede ser ahora utilizada para indicar posiciones dentro del muro. Usando este apuntador extendido el fontanero traza un plano para las tuberías de agua fría y caliente de la cocina propuesta. Después de evaluar su plano, hace unas pequeñas modificaciones para simplificar la instalación de las nuevas tuberías y entonces le pide a la computadora que le proporcione un costo estimado.

Después de conseguir la aprobación final del presupuesto el fontanero transmite los planos desde su computadora a su centro de trabajo, donde todas las tuberías necesarias son precortadas y ensambladas. Al día siguiente, uno de sus operarios llega con las partes prefabricadas y, utilizando una computadora similar revisa el plan y realiza la instalación.

En el relato del fontanero del futuro queda patente un problema actual en la construcción: Después de que un edificio haya sido construido, su mantenimiento y renovación a menudo requieren que se efectúen agujeros y modificaciones. Sabiendo la localización exacta de las tuberías o de los cables eléctricos en un muro se podrían evitar fugas de agua o pérdidas de fluido eléctrico. La información sobre el cableado eléctrico y de las tuberías de agua está disponible en los planos de construcción, pero es una faena tediosa para el personal de mantenimiento medir de nuevo las localizaciones exactas. Con la tecnología de RA se puede simplificar el proceso enormemente determinando automáticamente las medidas y sobreimprimir gráficamente los aspectos relevantes de un plano en un muro.

3.9.8.2 Mantenimiento mecánico y reparación

Otra área de aplicación que se está explorando actualmente es en el campo del mantenimiento mecánico y la reparación asistida por computadora. En este escenario un mecánico es asistido por un sistema de RA



mientras, por ejemplo, examina y repara un complejo motor. El sistema puede presentar una gran variedad de información al mecánico. Las anotaciones pueden identificar el nombre de las piezas o partes, describir su función y/ o presentar otra información importante tal como datos sobre el mantenimiento o de los posibles proveedores. La RA puede guiar al mecánico a través de una tarea específica resaltando partes que tienen que ser retiradas secuencialmente y mostrar su sentido de extracción. El sistema también podría proveer de información sobre seguridad. Por ejemplo, las partes que están calientes o electrificadas pueden ser resaltadas para recordar constantemente al mecánico el peligro de tocarlas.

3.9.8.3 Diseño interior

Antes de que un edificio se construya, se renueve o se redecore se emplea mucho tiempo diseñando y discutiendo los cambios a realizar. Muchas personas, en particular clientes y el público en general, tienen problemas para visualizar lo que se les propone. Además, personas diferentes pueden visualizar cosas diferentes debido a especificaciones imprecisas, sin darse cuenta de este fallo de comunicación o entendimiento hasta que alguna versión de lo propuesto se realiza finalmente. Normalmente, los arquitectos utilizan modelos en 3D, modelos en realidad virtual y versiones de fotografías retocadas para convencer a los clientes de lo atractivo de sus diseños. La RA iría un pasó más allá en esta dirección, permitiendo a los clientes interactivamente diseñar y visualizar "in situ" lo que se les está proponiendo.

Así, por ejemplo un arquitecto de interiores diseña, remodela y visualiza una habitación utilizando modelos de muebles de una base de datos que son sobreimpresos sobre imágenes de vídeo de la habitación.

Para hacer más comprensible esta aplicación imaginemos el siguiente escenario para esta aplicación que consiste en un jefe de oficinas que trabaja con un diseñador de interiores sobre la disposición de una oficina. El jefe pretende cambiar el mobiliario de la oficina. Con un sistema de RA el jefe de oficina podría ver la habitación tal y como quedaría con diferente mobiliario y desde diferentes puntos de vista.

Interactuando con varios proveedores sobre una red, seleccionarían el mobiliario consultando bases de datos utilizando un paradigma gráfico. El sistema provee descripciones y gráficos del mueble que están disponible desde los diferentes proveedores quienes a su vez, han hecho que sus modelos estén disponibles en sus bases de datos. Piezas o grupos de muebles que acuerdan ciertos requerimientos como color, proveedor o precio pueden ser solicitados El usuario elige piezas de este "catálogo electrónico" y presentaciones en 3D de este mueble aparecen visibles a los ojos del usuario. El mueble es posicionado utilizando un apuntador 3D. Un mueble puede ser suprimido, añadido y reorganizado hasta que los usuarios están satisfechos con el resultado; ellos ven estas piezas tal y como aparecen en la habitación actual. Si ellos varían su posición pueden ver la habitación amueblada desde diferentes puntos de vista.

Este sistema también puede ser utilizado en otras áreas de diseño como automóviles, moda y arquitectura. Incluso puede ser útil en áreas donde los usuarios necesiten comunicarse y discutir un problema en un contexto de 3D, como por ejemplo en el campo de la electrónica y reparación mecánica o la medicina.

3.9.8.4 Construcción exterior

A veces se desperdicia mucho tiempo y dinero porque se malinterpretan unos planos, o se utilizan versiones obsoletas, o simplemente la información es transferida de forma imprecisa del plano a la construcción real.

3.10 Comparación de los Paradigmas de Interacción

- 1) En una computadora de escritorio (utilizando una IGU) la interacción entre el usuario y la computadora está aislada de la interacción entre el usuario y el mundo real.
- 2) En la realidad virtual la computadora cubre totalmente el usuario y la interacción entre el usuario y el mundo real desaparece.
- 3) En la computación ubicua el usuario interacciona con el mundo real pero también puede interactuar con las computadoras de los que dispone en el mundo real.
- 4) La realidad aumentada soporta la interacción entre el usuario y el mundo real utilizando la información aumentada de la computadora.



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

Licenciatura En Informática

Programación de Interfaces

*Autor: L.I. María de Lourdes Isabel
Ponce Vásquez*

AGOSTO-DICIEMBRE 2005



Contenido

UNIDAD 4. DISEÑO	3
Objetivos:	3
Introducción.....	3
4.1 Prototipado	5
4.1.1 Dimensiones del prototipado	5
4.1.2 Tipos de prototipos:.....	5
4.1.2.1 Prototipo de papel	5
4.1.2.2 Storyboard.....	6
4.1.2.3 Escenario	6
4.1.2.4 Vídeo	7
4.1.2.5 Simulaciones	7
4.1.3 Prototipo de software	7
4.1.4 Problemas potenciales	7
4.2 Análisis centrado en el usuario	8
4.2.1 Los usuarios.....	8
4.2.2 Las tareas	9
4.2.3 El escenario	9
4.3 Ciclo de vida de la interfaz de usuario	10
4.4 Aproximaciones al diseño	11
4.4.1 Modelo mental y modelo conceptual	11
4.4.1.1 Modelo de procesador humano	12
4.4.1.2 Modelo de desarrollo de tareas	13
4.4.1.3 Modelo objeto–acción sintáctico–semántico (SSOA)	13
4.4.2 Estructura del modelo conceptual	13
4.5 Análisis de tareas	15
4.5.1 Proceso de obtención y análisis	16
4.5.2 Métodos de análisis de tareas.....	16
4.5.2.1 Análisis jerárquico de tareas (HTA)	17
4.5.2.2. GOMS	19
4.5.2.3 KLM.....	21
4.5.2.4 TAG.....	22
4.5.2.5 UAN.....	24
4.5.2.6 Concur Task Trees (CTT)	25
4.6 Modelos arquitectónicos	28
4.7 Modelos abstractos	28



UNIDAD 4. DISEÑO

Objetivos:

- Conocer el proceso de diseño de sistemas interactivos
- Estudiar notaciones y métodos para el análisis de la interfaz de usuario
- Conocer y aplicar análisis de tareas
- Introducir conceptos de diseño orientado a objetos
- Analizar estrategias de diseño

Introducción

En diferentes temas anteriores se expone que el prototipado y la evaluación constituye una parte básica a lo largo de todo el proceso de diseño de un sistema centrado en el usuario. Si se deja la validación del diseño para el final no se podrá saber si un diseño o un sistema cumple las expectativas de los usuarios y se adapta a su contexto social, físico y organizativo.

Los sistemas interactivos se caracteriza por la importancia del diálogo con el usuario. La interfaz de usuario es por tanto, una parte fundamental en el proceso de desarrollo de cualquier aplicación y por tanto se tiene que tener en cuenta su diseño desde el principio. La interfaz es la parte (hardware y software) del sistema informático que facilita al usuario el acceso a los recursos de la computadora. En este sentido, Thimbleby sugiere que la interfaz determinará en gran medida la percepción e impresión que el usuario poseerá de la aplicación. El usuario no está interesado en la estructura interna de la aplicación, sino en cómo usarla. No se puede realizar la especificación, diseñar las funciones y estructuras de datos y escribir el código y una vez casi terminado el proceso de desarrollo de la aplicación plantearse el diseño de la interfaz de usuario. Siguiendo esta forma de trabajo lo mas seguro es que se obtengan diseños de interfaces muy dependientes de los diseños que se han realizado de las datos y de las funciones, sin tener en cuenta que esos datos han de ser obtenidos y representados por y para el usuario.

Una vez que se tenga hecha la especificación, propuesto un diseño y el código esté implantado, es muy difícil cambiar las características de la interacción y presentación de la información, excepto pequeñas cosas. Por tanto, se debe empezar con un idea clara de cómo se quiere la interfaz y cómo serán las interacciones con el usuario para después, desarrollar las especificaciones funcionales que sirvan de guía al diseño posterior.

En el desarrollo de aplicaciones interactivas se podrán aplicar las técnicas de la ingeniería de software, pero teniendo en cuenta que se han de modificar algunos aspectos de los métodos de diseño clásico para adaptarlos a las peculiaridades de estos sistemas. Hay que tener en cuenta que un aspecto fundamental es el análisis y diseño de la parte interactiva, y que para realizarlo, se necesitarán aplicar técnicas de análisis y diseño específicas.

El desarrollo de un sistema interactivo deberá tener en cuenta a los participantes que van a intervenir en el mismo: **el usuario**, que posee la capacidad de elección y actuación, **la computadora**, que ofrece un programa y mecanismos para su acceso, y **el diseñador**, el encargado de anticipar las



posibles acciones del usuario y codificarlas en el programa. Todo ello se articula a través de la interfaz de usuario de la aplicación.

La tendencia hacia interfaces de usuarios fáciles de usar provoca que su diseño sea cada vez más complejo. La interfaz de usuario, como medio de comunicación entre el humano y la computadora se caracteriza por su apariencia (presentación) y su capacidad de procesamiento del diálogo. Se puede encontrar multitud de productos que permiten la descripción y generación automática de la apariencia externa de una aplicación mediante la utilización de paletas de recursos (botones, menús, etc.) herramientas visuales, *toolkits*, etc. Sin embargo, estas herramientas no proporcionan suficiente ayuda en el análisis del comportamiento dinámico de la interfaz, en su descripción y sobre todo, no aseguran su corrección. A continuación se introducirá una aproximación de ingeniería para el diseño de sistemas interactivos.

Tal como destacan diversos autores el diseño de sistemas interactivos implica realizar un diseño centrado en el usuario, haciendo que en el modelo de proceso sea un elemento fundamental e implicándolos tanto como sea posible hasta pensar en incluir usuarios en el equipo de diseño.

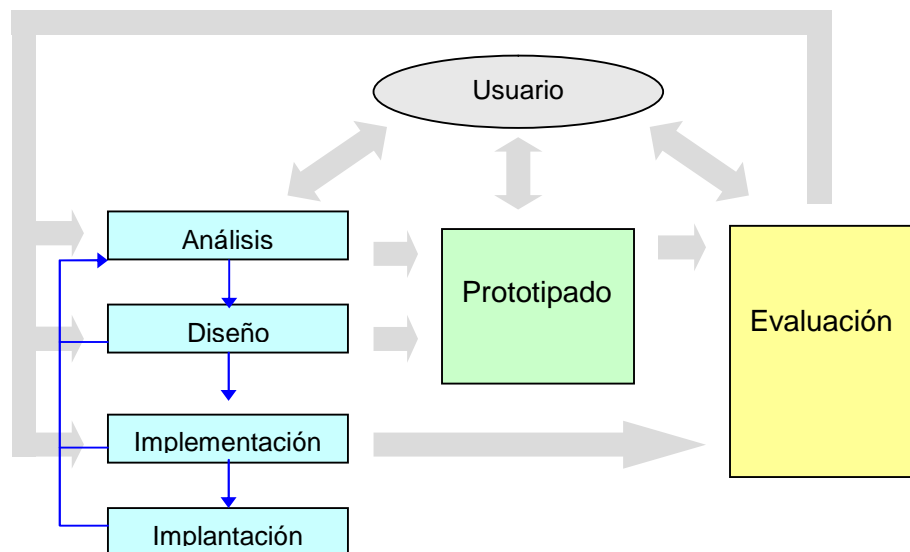


Figura 1 Ciclo de Vida del Sistema Interactiva Centrado en el Usuario

El trabajo se iniciará observando la práctica habitual de actividades (el comportamiento), para poder modelarlo realizando escenarios, prototipos o maquetas con el fin de poder ir evaluando el diseño a lo largo del ciclo de vida. Esto se puede realizar a través de un modelo de proceso o de ciclo de vida iterativo.

En el esquema anterior se muestra un modelo de ciclo de vida en el que en todas las etapas interviene la evaluación, ya sea inicialmente evaluando al usuario y su puesto de trabajo, o bien posteriormente realizando algún tipo de prototipado en que se cuestiona la usabilidad.



4.1 Prototipado

Tal como se ha planteado en el modelo de proceso centrado en el usuario, no se puede empezar una implementación a gran escala del sistema a partir de un diseño inicial de la interfaz de usuario. Para poder realizar evaluaciones de la usabilidad en etapas iniciales se han de crear prototipos, que pueden ser implementados mucho mas rápidamente, ser mas baratos y que se puedan cambiar muchas veces. Los prototipos son documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final. El prototipo es una herramienta muy útil para hacer participar al usuario en el desarrollo y poder evaluar el producto ya en las primeras fases del desarrollo (Modelo del ciclo de vida basado en prototipos).

4.1.1 Dimensiones del prototipado

La razón principal del uso de los prototipos es la reducción en costo y tiempo que supone su uso en la implementación del futuro sistema, esta reducción se puede conseguir ya sea reduciendo el número de características o bien reduciendo el nivel de implementación de las funcionalidades de las características, esto define dos dimensiones que se denominan prototipos horizontal y vertical:

- **Prototipado vertical.** El resultado de este tipo de prototipo es un sistema que tiene implementadas pocas características, pero sus funcionalidades están totalmente implementadas. Un prototipo vertical puede probar por tanto una parte limitada del sistema, pero puede ser probado en profundidad bajo circunstancias reales.
- **Prototipado horizontal.** Incluye toda la interfaz de todas las características del sistema pero no contiene funcionalidad subyacente. Un prototipo horizontal es una simulación de la interfaz donde no se puede realizar ningún trabajo real.

4.1.2 Tipos de prototipos:

4.1.2.1 Prototipo de papel

Este tipo de prototipo se basa en la utilización de papel, tijeras, lápiz o instrumentos que se puedan utilizar para describir un diseño en un papel. Este sistema permite una gran velocidad y flexibilidad.

- **Cómo se realiza un prototipo de papel.** Para poder simular las diferentes interacciones que se van a realizar con el sistema, se realiza una hoja para cada uno de los diferentes escenarios que se van a tener como resultado de las diferentes posibles interacciones que se pueden realizar. Se apilarán estas hojas que permitirán simular la aplicación.
- **Uso.** Para utilizar el prototipo de papel se debe situar en un escenario de uso de futuro en el que el diseñador actúa como coordinador. El prototipo será analizado por un posible usuario e intentará realizar algunas de las tareas que se pretende diseñar. En voz alta se irán realizando las interacciones y le irán cambiando las hojas de papel en función de las interacciones que vaya realizando.
- **Ventajas.** El costo es muy reducido, necesitando únicamente los recursos humanos dedicados a la realización del prototipo. Los cambios se pueden realizar muy rápidamente y sobre la marcha. Si el diseño no funciona se pueden rescribir las hojas erróneas o rediseñarlas y volver a probar la tarea a realizar. Los usuarios o los actores se sienten más cómodos para poder realizar críticas al diseño debido a la sencillez del mismo por lo que no se sienten cohibidos a dar sus opiniones.



4.1.2.2 Storyboard

Un storyboard es una narración gráfica de una historia en cuadros consecutivos. Se puede utilizar este concepto que se emplea en el diseño cinematográfico, teatro, etc. para la realización de un escenario de interacción que puede ser evaluado con diferentes técnicas. Una de las opciones que se tienen en un storyboard para una aplicación es que se pueden indicar los enlaces a diferentes páginas del storyboard a partir de los resultados de las interacciones del usuario.

4.1.2.3 Escenario

Las computadoras son algo mas que funcionalidades, siempre reestructuran actividades humanas, creando nuevas posibilidades, al mismo tiempo que dificultades. Por otro lado, cada contexto en el que el ser humano tiene experiencia y actúa, proporciona restricciones para el desarrollo de sistemas de información. En el momento que se tenga que analizar y diseñar software, se necesita una forma de ver cómo estos nuevos sistemas pueden transformar y ser restringidos por los contextos actuales de la actividad humana. Una aproximación directa es imaginando y documentando las actividades típicas y significativas en etapas iniciales y continuamente durante el proceso de desarrollo.

Estas descripciones se denominan escenarios. Los escenarios son historias, historias sobre personas y sus actividades. Los escenarios destacan objetivos sugeridos por la apariencia y comportamiento del sistema; qué es lo que las personas quieren hacer con el sistema; qué procedimientos se usan, cuáles no se usan, se realizan o no satisfactoriamente y qué interpretaciones hacen de lo que les sucede.

Los escenarios tienen elementos característicos:

- **Configuración** (*setting*). Por ejemplo una oficina, con una persona sentada enfrente de una computadora trabajando con una hoja electrónica. Es importante concretar por ejemplo que la persona es el contador y que los objetos de trabajo son balances y presupuestos.
- **Agentes o actores**. El contador es el único agente en el escenario que se está describiendo, pero es normal que en las actividades humanas participe mas de un actor, cada uno típicamente con sus objetivos.

Cada escenario implica por lo menos un agente y al menos un objetivo. En el caso de que haya mas de un actor y objetivo ha de haber un objetivo y un agente principales.

Los escenarios tienen un diagrama que incluye secuencias de acciones y eventos, cosas que hacen los actores, cosas que les suceden, cambios en las circunstancias de la configuración y otras.

Representando el uso de un sistema o una aplicación con un conjunto de escenarios de interacción el uso se hace explícito y por tanto orienta el diseño y el análisis.

Las representaciones de escenarios pueden ser elaboradas como prototipos, a través del uso de storyboards, videos y herramientas de prototipado rápido.



Herramientas de diagramación

- **Narrativa.** Una historia completa de la interacción hecha con la existente o con un diseño nuevo.
- **Flowchart.** Una representación gráfica de las series de acciones y decisiones extraídas de la narrativa.
- **Texto procedural.** Una descripción paso a paso de las acciones del usuario y las respuestas del sistema.

4.1.2.4 Vídeo

El vídeo permite el rodaje de un escenario en el que se pueden realizar manipulaciones durante el postproceso para simular algunas características del diseño de las que todavía no se disponen. El prototipo en vídeo puede ser muy útil en el diseño de interfaces multimodales en el que por ejemplo se realiza una interacción por voz o en el diseño de escenarios futuros de los que todavía no se dispone de la tecnología. El vídeo se analiza después por el equipo de desarrollo y posibles usuarios y puede ser evaluado por ejemplo con las técnicas de hablar alto. Un ejemplo interesante de escenario es el vídeo *Starfire* rodado por Sun Microsystems que plantea como será la interacción en el futuro.

4.1.2.5 Simulaciones

Algo de funcionalidad tiene que ser incluida en el prototipo para demostrar el trabajo que la aplicación tiene que realizar. Los prototipos presentados hasta ahora no son suficientes para este propósito, para poder realizar este trabajo parte o toda la funcionalidad del sistema tiene que ser simulada por el equipo de diseño. Añadiendo un soporte de programación para simular permite al diseñador construir objetos interactivos textuales y gráficos que añaden cierto comportamiento a estos objetos que simulan las funcionalidades del sistema. Una vez construida la simulación se puede evaluar y cambiar en función de los resultados de la evaluación.

4.1.3 Prototipo de software

En este caso se pueden realizar diferentes tipos de prototipos utilizando las herramientas de desarrollo. Dix plantea los siguientes:

- **Maqueta para tirar.** Es un tipo de prototipo parecido al de papel por ejemplo, que sirve para realizar una evaluación con el usuario y posteriormente se desecha.
- **Incremental.** El producto final se construye como componentes separados. Cada vez se va probando uno y finalmente se realiza una prueba final.
- **Evolutivo.** En este caso el prototipo no es eliminado y se utiliza como base para una próxima iteración en el diseño.

4.1.4 Problemas potenciales

Sommerville plantea problemas en el uso de prototipos de los que se debe estar conscientes:



- Trabajar con prototipos requiere su tiempo y además si se trabaja con maquetas que se tiran, se puede ver desde la administración como la pérdida de un tiempo precioso que se saca del diseño de tareas reales.
- Puede ser que el administrador del proyecto no tenga la experiencia necesaria para planear y deducir el costo del proceso de diseño con prototipos.
- Algunas veces las características mas importantes de un sistema pueden ser no funcionales como la seguridad y la fiabilidad, y estas son precisamente las características que hay que sacrificar en un diseño con prototipos.
- El proceso de diseño esta normalmente basado en un contrato entre el cliente y el diseñador, por lo que hay que definir claramente el modelo de diseño que se adapte al uso de prototipos y obtener un documento como resultado del prototipado.

4.2 Análisis centrado en el usuario

El diseño de un sistema interactivo debe satisfacer las demandas de los usuarios que lo van a utilizar. La computadora es una herramienta para realizar un determinado trabajo o actividad, y para que sea una buena herramienta, deberá ser adecuada, cómoda y eficiente para realizar estos cometidos. Para lograr un buen diseño, se debe realizar un análisis profundo del contexto donde se desarrolla el trabajo. Para ello se deberán analizar las características del usuario, las actividades que realiza y el escenario donde se desempeña su actividad. Todos estos factores permitirán conocer los requisitos que se deben satisfacer en el diseño del sistema.

4.2.1 Los usuarios

En primer lugar, a la hora de diseñar el sistema, se deben tener en cuenta las peculiaridades de los usuarios potenciales del mismo. Esta necesidad de incorporar el factor humano en el diseño viene dada por el reconocimiento del mal diseño que se ha hecho en gran cantidad de aplicaciones y el deseo de crear productos que ayuden de forma efectiva al usuario. Además, las características de los usuarios pueden afectar al modo de trabajo y condicionar el proceso de comunicación con el sistema. Por ejemplo, los factores humanos pueden condicionar el tiempo de aprendizaje, el rendimiento (tiempo para realizar una tarea), la frecuencia de errores cometidos, grado de retención (memoria de uso) o de satisfacción del usuario. A la hora de diseñar la aplicación, se puede realizar por encargo directo (por lo que existe un cliente), o bien, dirigirlo a un colectivo más o menos amplio de potenciales usuarios (niños, profesionales, estudiantes, etc.).

El análisis del usuario implica conocer aspectos tales como:

- **Habilidades físicas y sensoriales.** Estas habilidades determinarán en gran medida la adaptación del entorno de trabajo a las características del usuario (tamaño de los botones, tipo de dispositivos, etc.). Se pueden dar casos en los que el diseño debe ser preferentemente ergonómico por las limitaciones en movilidad de los usuarios, como por ejemplo, la discapacidad por parálisis cerebral, o tener en cuenta pequeñas alteraciones como por ejemplo el daltonismo.
- **Habilidades cognitivas.** Estas diferencias en la capacidad de razonamiento y conocimiento están motivadas por el grado de experiencia que posee el usuario tanto de su propio trabajo como del uso



de la computadora como herramienta. Se puede tener una gran variedad de usuarios desde los expertos a los novatos, usuarios cotidianos u ocasionales, motivados o no, etc.

- **Diferencias de personalidad.** Las diferencias en la personalidad puede provocar alteraciones en la propia comunicación. Así, personas tímidas tendrán un comportamiento más cauto y prudente ante la computadora que una persona extrovertida y nerviosa.
- **Diferencias culturales.** También se pueden encontrar diferencias motivadas por el entorno socio-cultural donde se encuentra el usuario, que puede afectar al lenguaje utilizado, expresiones y terminología, modo de trabajar, etc.

Este conjunto de características relevantes de los usuarios serán de gran ayuda en las etapas posteriores de diseño. Para ello, se puede partir de una tabla en la cual se indiquen los distintos tipos de usuarios (secretaria, director, técnico..) y sus características relevantes (grado de utilización del sistema, nivel de experiencia, etc.).

4.2.2 Las tareas

Otro factor importante a tener en cuenta en el diseño son las tareas que realizan los usuarios. Nuestra forma de actuar está dirigida por objetivos (*goals*) como se recoge en el modelo de Norman. Para lograr ese objetivo (por ejemplo comer), se debe llevar a cabo una serie de actividades (encender, obtener, poner...) sobre unos objetos (microondas, pizza, temporizador...) encaminadas a lograr ese objetivo. A la hora de realizar estas tareas mediante un sistema interactivo se deberá tener en cuenta que sigan siendo familiares al usuario, es decir, la forma de llevarlas a cabo, su representación así como la secuencia de acciones debe ser similar a la que realiza en el entorno real. Si esto no se satisface, el usuario requerirá un esfuerzo adicional para comprender las tareas que realiza cotidianamente.

4.2.3 El escenario

Las personas no realizan su trabajo de forma aislada, sino que se ven condicionadas por el escenario donde se desempeña esta labor. Los aspectos más relevante a tener en cuenta son:

- **Entorno físico.** El entorno es fundamental para poder trabajar. Se debe prestar atención a las características ergonómicas del mismo (tipo de ubicación, iluminación, espacio, etc.) así como las peculiaridades del entorno (ruido, contaminación, calor, etc.). Puede haber casos de especial importancia como sitios de alto riesgo (central nuclear) o condiciones extremas (submarino, aeronave, etc.)
- **Entorno social.** El entorno social implica el trabajo dentro de un grupo donde existen unas normas de comportamiento. Se pueden encontrar situaciones en las cuales pueda haber cooperación para el trabajo (ayuda), compartir datos o recursos, dependencias jerárquicas, etc.

Algunas de estas características pueden condicionar el diseño, ya que un trabajo en equipo fuertemente acoplado (con alto nivel de cooperación y compartición de datos) requerirá de una aplicación groupware para trabajo en grupo.



4.3 Ciclo de vida de la interfaz de usuario

La construcción de un sistema interactivo implica un proceso cíclico de diseño, desarrollo y evaluación. La realimentación que proporciona la evaluación sobre el diseño es fundamental para refinar y pulir aspectos que son muy dependientes de los usuarios finales (el factor humano) una vez que el sistema se ha puesto en marcha. En la figura del ciclo de vida se muestra la importancia del usuario (tanto en la fase de análisis como de evaluación) y la naturaleza cíclica del diseño (con continua realimentación a partir de la evaluación).

Las primeras interfaces las realizaban los propios programadores para los programas que ellos mismos utilizaban. Sin embargo, los diseños deben ir dirigidos a usuarios con diferentes habilidades, y no necesariamente tienen que ser expertos en informática. Las computadoras son herramientas con las cuales las personas pueden realizar sus tareas, por lo que esto se deberá tener en cuenta a la hora del diseño, ya que si el usuario percibe que algo es difícil de usar, cometerá errores, o bien no realizará la tarea adecuadamente. Para que esto no suceda, es muy importante basar el diseño del sistema sobre aquellos conceptos que maneja el usuario y fundamentarse sobre criterios consistentes y fundamentos teóricos y no en meros juicios intuitivos.

Un buen diseño depende del conocimiento (fundamentos) y experiencia de los diseñadores. Esta información se puede organizar y estructurar para que pueda servir a otros diseñadores. Se puede disponer de varias fuentes de información con diferente grado de rigor y normativa, entre las que destacan:

- **Principios.** Son objetivos generales que pueden ser útiles para organizar el diseño. Aconsejan al diseñador cómo debe proceder. Sin embargo, no se especifican métodos para obtener esos objetivos, y está limitado al uso práctico (por ejemplo: conocer al usuario, minimizar el esfuerzo para realizar una tarea, mantener la consistencia, etc.).
- **Guías (*guidelines*).** Conjunto de recomendaciones que deben ser aplicados a la interfaz y que son cuantificables. Deben ser generales para que puedan ser aplicadas en diferentes contextos. Pueden deducirse de teorías cognitivas, ergonomía, sociología, de la experiencia etc. (por ejemplo, no disponer mas de siete ítems en un menú).
- **Estándares.** Son principios y guías que se deben seguir por imposición industrial. Existen estándares de facto (Macintosh Toolbook, MS Windows, IBM SAA/CUA). Estos estándares se diseñan para proteger la uniformidad y la línea de productos desarrollados. Con ello, mejoran la eficiencia del usuario (beneficio de una interfaz común para muchos programas). Existen otros estándares en otros ámbitos: ANSI, ISO, DIN, MIL–STD, NASA–STD.

Este conocimiento puede ayudar en el diseño, aunque sin embargo no es suficiente, por lo que se debe partir de los requisitos del sistema, conocimiento del usuario y aplicar una metodología para un desarrollo efectivo del sistema. Se deberán aplicar técnicas de análisis y especificación para la descripción de aquellos aspectos que sean relevantes dentro del sistema.

Un diseño centrado en el usuario requiere de una continua evaluación del producto a desarrollar. Por este motivo, cobran gran importancia los siguientes aspectos:

- **Métodos formales.** Permiten una especificación precisa y sin ambigüedad del diseño a generar. Permite una verificación formal de propiedades y en algunos casos se puede generar la implementación automáticamente.



- **Herramientas de desarrollo de modelos de interfaces (MB–UID).** Estas herramientas obtienen la interfaz a partir del análisis de los requisitos de usuario. Su labor fundamental es la generación de aplicaciones a partir del diseño aunque también se pueden considerar como herramientas de prototipado. Actualmente los lenguajes de programación visuales también disponen de bibliotecas (ej. AWT y SWING en Java) que permiten implementar las técnicas de interacción y presentación de la información.
- **Prototipado.** Los prototipos son documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final. El prototipo es una herramienta muy útil para hacer participar al usuario en el desarrollo y poder evaluar el producto ya en las primeras fases del diseño (modelo del ciclo de vida basado en prototipos).

No obstante, el desarrollo de sistemas interactivos sigue siendo una labor difícil y con un alto costo en tiempo y esfuerzo. Un motivo de esta complejidad es por la necesidad de adaptar el diseño a una gran variedad de usuarios, a diferentes objetivos y sobre diferentes contextos.

4.4 Aproximaciones al diseño

El desarrollo de Sistemas Interactivos es una tarea compleja para la cual se necesitarán herramientas y metodologías que permitan realizar un diseño satisfactorio centrado en el usuario. Existen dos aproximaciones para realizar el diseño:

- **Aproximación empírica.** El diseño se basa en la propia experiencia del diseñador o bien en la de otros diseñadores que se unifican mediante compendios de recomendaciones (guías, *reglas de oro*, estándares, etc.) más o menos relevantes para la construcción de un interfaz con éxito. Estos resultados generalmente están avalados por unos estudios de evaluación por el usuario (tests de usabilidad).
- **Aproximación metodológica.** Se basa en unos fundamentos teóricos y en la aplicación de una serie de pasos para la realización del diseño.

La aproximación metodológica posee bastantes aportaciones de otras disciplinas, sobre todo de las teorías cognitivas ya que aportan mecanismos para la descripción del conocimiento que el usuario posee del sistema. De hecho, la aproximación empírica se basa en las aportaciones más relevantes (enunciadas como reglas de diseño) de las aportaciones teóricas. Esta unidad se centrará en una aproximación metodológica para el desarrollo de sistemas interactivos, analizando las peculiaridades de este tipo de sistemas y los mecanismos existentes para su análisis y diseño.

En el ámbito de los sistemas interactivos se ha utilizado el término diseño con muchas connotaciones. De hecho, el concepto de diseño abarca desde aspectos de análisis (de usuarios, tareas, del entorno, propiedades), aspectos de modelado (arquitectura) hasta cuestiones relativas propiamente de diseño (apariencia, codificación, etc.).

4.4.1 Modelo mental y modelo conceptual

Un aspecto muy importante en el diseño de sistemas interactivos es el factor humano, por lo que se deberá partir de modelos cognitivos que permitan estudiar y representar cómo es asimilada y procesada la información por una persona. La obtención del conocimiento acerca de una aplicación



basada en computadoras se realiza mediante un *aprendizaje*. Para ello, se introducen dos términos para identificar el grado de asimilación y comprensión del usuario del entorno:

- **Modelo conceptual:** Es una abstracción externa que describe, mediante diagramas y notaciones más o menos formales, el conocimiento que debe poseer una persona acerca de un sistema. Este modelo es realizado por el analista y debe ser completo, consistente y exacto (sin ambigüedad).
- **Modelo mental** (o modelo de usuario): Es la abstracción del conocimiento interno que posee el usuario. Este modelo da una medida real de lo que el usuario piensa/ conoce acerca del sistema informático. Este modelo guía las intenciones del usuario para realizar una tarea en el sistema. Además, este modelo mental se puede ir modificando conforme se interacciona con el sistema.

El modelo conceptual está basado en un conjunto de elementos y de relaciones que se pueden observar en un determinado sistema, representando el conocimiento que cualquier usuario debería adquirir sobre el sistema. Este modelo se deberá definir mediante una notación formal y comprensible que evite la ambigüedad del lenguaje.



Figura 2 Inconsistencia en el modelo mental

El modelo conceptual debe proporcionar información al usuario acerca de **qué** hace el sistema y los mecanismos para llevarlo a cabo. Su importancia radica en que debe favorecer el *aprendizaje* del sistema, es una guía para predecir el comportamiento del sistema, y además, el usuario utilizará este modelo para establecer estrategias encaminada a resolver sus problemas. Los principios en los que debe estar basado el modelo conceptual serán por tanto que sea **asimilable** (mediante el uso de conceptos familiares), **consistente** (coherente y bien formulado) y **simple** (uso de descripciones comprensibles por un usuario medio).



Figura 3 Modelo conceptual (con notación formal)

Para poder realizar el modelo conceptual de un sistema, se deberán conocer y aplicar modelos teóricos cognitivos que están fundamentados en el mecanismo de razonamiento humano. Los más relevantes son:

4.4.1.1 Modelo de procesador humano

Card y Moran presentan este modelo en el que se expone la forma de percibir, procesar y manipular la información. Este modelo identifica diferentes procesadores y sistemas de memoria, donde cada uno de ellos tiene asignado parámetros cuantitativos importantes como ciclos de tiempo o capacidades. El modelo del procesador humano está compuesto de tres sistemas: el sistema



perceptual, que maneja los estímulos sensoriales externos, el sistema motor, que controla las acciones y por último, el sistema cognitivo, que suministra el conocimiento suficiente para conectar ambos.

4.4.1.2 Modelo de desarrollo de tareas

Norman, en 1986, propone un modelo de desarrollo de tareas que identifica siete etapas de ejecución y evaluación de acciones de usuario. El modelo representa las etapas de actividad mental que implica que el usuario alcance un objetivo y que son:

1. establecer el objetivo que se quiere alcanzar,
2. formalizar la intención para la acción que alcanzará el objetivo,
3. especificar la secuencia de acción correspondiente a la intención,
4. ejecutar la acción,
5. percibir el estado del sistema,
6. interpretar el estado, y por último
7. evaluar la interpretación del estado con respecto al objetivo inicial.

Este modelo proporciona una base para representar y entender las consecuencias cognitivas de diseños particulares.

4.4.1.3 Modelo objeto–acción sintáctico–semántico (SSOA)

Este modelo descrito originalmente por Shneiderman en 1980, propone que los usuarios poseen un conocimiento sintáctico y semántico del dominio del problema y de los mecanismos de interacción. En este conocimiento se almacenan detalles de los dispositivos (tipos, modo de uso), conocimientos semánticos sobre las actividades y conceptos de la computadora. Este conocimiento se estructura mediante una colección de objetos que componen el sistema (cursor, icono, ventana..) y de las acciones que se pueden llevar a cabo sobre cada uno de esos objetos (mover, cambiar, redimensionar, etc.).

4.4.2 Estructura del modelo conceptual

El modelo conceptual es muy importante, ya que permiten identificar, organizar y realizar razonamientos sobre los componentes y comportamiento de un sistema interactivo, será la guía para el proceso de diseño del software y puede usarse posteriormente como una referencia para evaluar un diseño particular, razonar sobre la solución realizada y el posible espacio de soluciones existente. Por tanto, la correcta especificación del modelo conceptual será crucial en toda la etapa del proceso de diseño. Algunas de estas notaciones del modelo conceptual están basadas en métodos formales (con un fundamento basado en lógica matemática), lo que permitirá una descripción precisa y sin ambigüedad.

Partiendo de las teorías cognitivas presentadas anteriormente, se podría realizar la descripción conceptual del sistema mediante uno de estos modelos:

- **Modelo de caja negra:** El usuario no tiene idea del funcionamiento interno, y simplemente conoce que ciertas entradas producen una serie de resultados. Este es una visión “mágica” del sistema, en la cual el usuario no tiene bases para predecir nuevos comportamientos ni causas que provocan



los errores. El usuario se ve forzado a considerar los resultados verdaderos, y no sabe cómo juzgar su validez.

- **Modelo funcional jerárquico:** Las funciones suministradas por el sistema se agrupan en jerarquías, permitiendo reducir la complejidad del sistema mediante la aplicación de técnicas de partición en el dominio del problema (método de *divide y vencerás*).
- **Modelo basado en estados:** El sistema se define como un conjunto de estados. Las transiciones son provocadas por eventos claramente definidos. El usuario puede observar los cambios en el estado del sistema. Un ejemplo es el sistema de comunicación por teléfono (diferentes pitidos para estados del sistema: ocupado, llamada, etc.)
- **Modelo basado en objetos y acciones:** Se trabaja directamente sobre entidades (físicas o abstractas), sobre las cuales se pueden realizar acciones. El usuario debe conocer la existencia de objetos, de sus posibles atributos y acciones aplicables. Por ejemplo, los iconos (acciones asociadas y atributos).

Con estas posibles estructuraciones de la información que residen en el modelo conceptual, se podría optar por centrarnos en la descripción del conocimiento que el usuario debe tener del sistema (siendo irrelevante la arquitectura del sistema) o viceversa (dando más importancia al modelo del sistema respecto al conocimiento del usuario). A menudo, estas son dos alternativas (en algunos casos complementarias) para el diseño de sistemas interactivos. Una descripción basada en el conocimiento del usuario llevará a un **modelo de tareas**, mientras que una descripción del sistema conducirá a un **modelo arquitectónico**.

Los **modelos de tareas** analizan y describen el conocimiento que el usuario debe poseer acerca del sistema para su correcta utilización. En ese sentido, se ha trabajado en dos vertientes. Por un lado, se debe caracterizar el proceso de adquisición de la información por parte del usuario, y por otro, se busca un mecanismo para expresar el rendimiento humano para la ejecución de unas determinadas actividades. Estos métodos se basan en el análisis de tareas y generalmente usan una descripción funcional jerárquica.

Los **modelos arquitectónicos** representan la estructura interna del sistema. Se describe la composición del sistema en base a una descripción modular que facilita la composición de componentes simples para la definición de elementos mas complejos. Estos modelos se basan en el concepto de *iterador*, *objeto activo* o *agente interactivo* como un objeto especializado que va a formar parte del sistema interactivo y que posee un estado y reacciona ante eventos (estímulos externos al Objeto). Normalmente se usa una descripción basada en estados o bien en objetos y acciones (aproximación que se adapta bien a las metodologías orientadas a objetos existentes).

Otra alternativa diferente son los **modelos abstractos** (basados en un modelo de caja negra), los cuales se utilizan para describir las propiedades más relevantes del sistema (consistencia, visibilidad, etc.) en base a las entradas y salidas que se producen en el sistema, y sin tener en cuenta su estructura interna. El modelo PIE propuesto por A. Dix es el más conocido.

Estos tres modelos no tienen por qué ser excluyentes entre sí, ya que básicamente diferencian los aspectos relevantes del estudio y en el nivel de abstracción con el que se analiza el sistema.



4.5 Análisis de tareas

El análisis de tareas es un término que cubre diferentes técnicas orientadas a describir las interacciones entre las personas y los entornos de una manera sistemática. El análisis de tareas se puede definir como *el estudio de lo que un usuario tiene que realizar en términos de acciones y/o procesos cognitivos para conseguir un objetivo*. Es por tanto una metodología que esta soportada por un conjunto de técnicas para ayudar en el proceso analítico de obtención de información, organizarlo y usarlo para realizar valoraciones o decisiones de diseño.

Conceptos iniciales:

- Una de las premisas de cualquier aproximación con la que se aborde el diseño es la de *conocer al usuario* y las actividades que realiza.
- Esta información se obtiene en la fase de **análisis de las tareas** con una notación que permita su formalización y estudio.
- Para ello, se considerará una tarea como una unidad significativa de trabajo en la actividad de una persona.

El análisis de tareas proporciona información muy relevante acerca del sistema a diseñar que a menudo no se obtiene con las técnicas de requisitos tradicionales de la ingeniería del software. Dentro del proceso de análisis de tareas, hay dos fases muy importantes:

- Obtención de la información necesaria para comprender las actividades que realiza el usuario (fase de análisis).
- Representación de esta información sobre un modelo adecuado (fase de modelado)

Mediante estos pasos, se obtiene una descripción formal del conjunto de acciones que debe realizar el usuario para la consecución de un determinado objetivo o finalidad. Estos métodos parten de las teorías cognitivas para realizar una representación del usuario y su interacción con la interfaz. Se modela su comprensión, conocimiento, intenciones y mecanismo de procesamiento. El nivel de representación depende del modelo concreto (desde tareas y objetivos hasta el análisis de la actividades motoras). En concreto, esta información puede ser muy útil para:

- Comprender el dominio de la aplicación: identificación de las actividades más importantes y sus relaciones.
- Facilitar discusiones interdisciplinarias: el conocimiento de las tareas puede ser útil al diseñador, usuarios, analistas, psicólogos, sociólogos, etc.
- Diseño de la nueva aplicación de forma consistente con el actual modelo conceptual, preservando las características más relevantes del funcionamiento lógico.
- Análisis y evaluación de usabilidad. Se puede predecir el rendimiento humano y para identificar problemas de uso.

Definiciones básicas:



- **Objetivo:** Es el estado o logro que el usuario quiere alcanzar dentro de una aplicación
- **Tarea:** Es la actividad necesaria para conseguir un objetivo. Pueden ser tanto actividades mentales como físicas.
- **Acción:** Es cada uno de los pasos a seguir para cumplimentar una tarea. Se puede considerar una acción como una tarea que no implica una solución de un problema o una estructura de control.

4.5.1 Proceso de obtención y análisis

En el análisis de tareas se deberán identificar las tareas más relevantes del sistema. La obtención de esta información se puede realizar mediante las siguientes técnicas:

- Entrevistas y reuniones
- Cuestionarios
- Observación del usuario en su trabajo
- Identificación de actividades en el contexto del entorno
- Estudio de la documentación actual, programas de formación, etc.

Mediante estas técnicas, se obtendrá información relevante para identificar las tareas. En concreto, se deberá centrar en:

- Información que necesita el usuario para realizar la tarea (qué hacer)
- Terminología y símbolos del dominio del problema (elementos).
- Descripción de cómo esas tareas se realizan actualmente (cómo).
- Casos de uso (situaciones)
- Tipos de usuarios

El resultado de este análisis es una lista de tareas relevantes con algún tipo de información adicional (atributos, restricciones, preferencias, etc.). De esta información se deben abstraer aquellos conceptos que son relevantes para el diseño de la aplicación como son:

- El modelo de diálogo: cómo se va a realizar la comunicación persona-computadora, bajo qué paradigma y estilo.
- Modelo de tareas: Especificación de las tareas en el sistema nuevo.
- Dominio de sistema: Descripción de los componentes y arquitectura del sistema.
- Modelo de usuarios: Identificación del tipo de usuarios, papel que desempeñan en el sistema y sus interrelaciones.
- Propiedades del sistema. Estudio de las características del sistema y de los requisitos que satisface (seguridad, robustez, etc.).

4.5.2 Métodos de análisis de tareas

Para llevar a cabo el análisis de tareas, se pueden usar diferentes métodos que se diferencian en el grado de formalismo de su notación, poder de expresividad y finalidad. Si bien todos ellos representan las tareas del sistema, la finalidad del estudio puede ser diferente:

- Métodos de competencia o cognitivos. Estos métodos identifican secuencias de comportamiento correctas, representando el tipo de conocimiento que debe poseer un usuario acerca del uso



del sistema. Partiendo de la descripción de tareas generan una especificación del conocimiento del usuario.

- **Métodos predictivos** para la evaluación del rendimiento humano. Describen secuencias de comportamiento y el conocimiento que necesita el usuario para su ejecución. Análisis centrado en rutinas de comportamiento.
- **Métodos descriptivos.** Permiten obtener una descripción más o menos completa del sistema a partir de la información obtenida de las tareas.

La siguiente tabla detalla algunos de estos métodos con sus características más relevantes.

Método	Tipo	Notación	Especificación	Comentarios
HTA	Cognitivo	Gráfico	Semi-Formal	Modelo de descomposición del conocimiento
GOMS	Cognitivo	Textual	Semi-Formal	Familia de lenguajes para describir el conocimiento
UAN	Cognitivo	Gráfico	Semi-Formal	Notación para el estilo de manipulación directa
KLM	Predictivo	Textual	Tiempo	Medición del rendimiento humano
TAG	Predictivo	Textual	Esquemas	Medida de la consistencia
CTT	Descriptivo	Gráfico	Lógica temporal	Herramientas de soporte al análisis y verificación.

4.5.2.1 Análisis jerárquico de tareas (HTA)

El análisis jerárquico de tareas (*HTA Hierarchical Task Analysis*) desarrollado por Annett y Duncan, es la técnica de análisis de tareas más conocido y más antiguo.

En HTA se realiza una descripción de tareas en términos de *operaciones* y *planes*. Las operaciones (descomposición en subtareas) son actividades que realizan las personas para alcanzar un objetivo, y los planes son una descripción de las condiciones que se tienen que dar cuando se realiza cada una de las actividades. Las operaciones se pueden descomponer de forma jerárquica y se asigna un plan a cada una de las subtareas que aparecen. Se define un objetivo como un estado determinado del sistema que quiere alcanzar el usuario. Aunque se habla de objetivos y tareas, la representación que se realiza describe únicamente la descomposición jerárquica en subtareas de las tareas que aparecen en el sistema.

El formato gráfico se parece a un árbol con ramas y subramas en función de las necesidades. A la hora de describir la descomposición de una tareas en subtareas se pueden representar cuatro tipos de descomposiciones:

- **Secuencia.** Descomposición en un conjunto ordenado temporalmente de una secuencia de tareas.
- **Selección.** Conjunto de tareas de las que se tendrá que elegir una de ellas.
- **Iteración.** Repetición de un subconjunto de tareas.
- **Tarea unitaria.** Actividad indivisible (según el nivel de detalle dado)

El análisis de tarea implica tres etapas enlazadas: obtención de información, diagramación y análisis. Los procedimientos de obtención de información incluyen la revisión de la documentación existente (por ejemplo, manuales de funcionamiento, procedimientos, informes de seguridad, estudios de análisis de tareas previos, diseños, imágenes, prototipos, etc.), que permitan establecer qué hacen 17



las personas en circunstancias específicas (normales y anormales), entrevistas y cuestionarios (descripciones por parte de personas experimentadas como hacen las cosas, que informaciones necesitan y como determinan si la tarea se puede realizar satisfactoriamente).

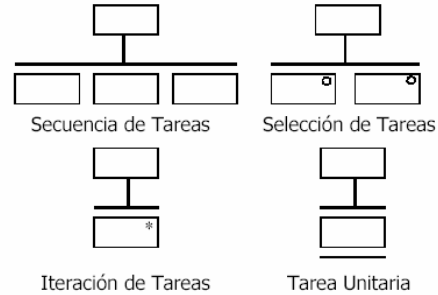


Figura 4 Notación de HTA

Algunas tareas se pueden desglosar con mayor detalle en secuencias. Un plan describe el conjunto de operaciones necesarias para llevar a cabo una actividad, o bien, muestra las circunstancias por las que una operación es realizada antes que otra. Estos planes se añaden a la tabla jerárquica.

La descripción de la información se realiza en forma de tabla o en forma de diagrama de árbol que describa las relaciones entre tareas y subtareas.

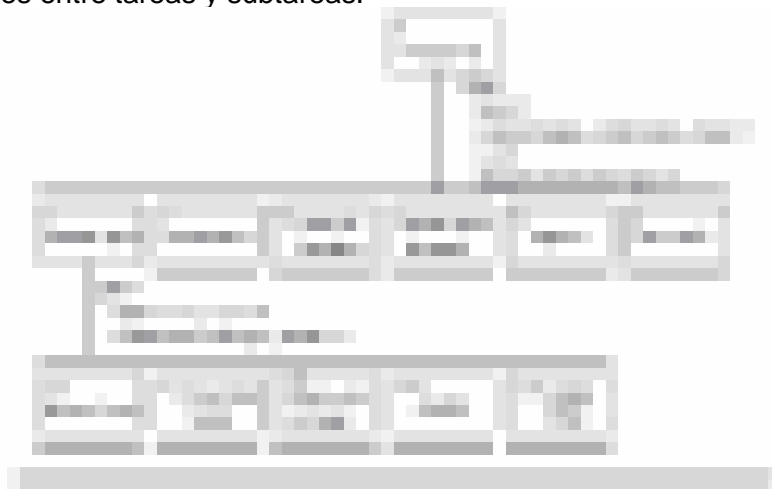


Figura 5 Descripción en HTA de la preparación del té

- 0. Hacer té
 - 1. Calentar el agua
 - 1.1 Llenar cazo
 - 1.2 Encender fuego
 - 1.3 Poner cazo en fuego
 - 1.4 Esperar
 - 1.5 Apagar fuego
 - 2. Vaciar tetera
 - 3. Poner hojas de té en tetera
 - 4. Verter el agua
 - 5. Esperar



6. Servir el té

Plan 0: hacer 1.
si tetera está llena entonces
hacer 2 al mismo tiempo
hacer 3-4-5
Cuando el té ha reposado, hacer 6

Plan 1: hacer 1.1-1.2-1.3-1.4
Cuando el agua está hirviendo, hacer 1.5

El análisis de la información es la fase final. Esta información se usará como base para decisiones de diseño y puede servir como guía para las actividades de diseño. La metodología para utilizar HTA como análisis de tareas sería la siguiente:

- Etapa inicial. Definición de la tarea principal, que puede ser dividida entre cuatro y ocho subtareas.
- Etapa intermedia. Decidir el nivel de detalle que se requiere y en que punto acabar la descomposición.
- Parte final. Revisión y evaluación del trabajo realizado para comprobar su consistencia.

4.5.2.2. GOMS

GOMS (propuesto por Card/ Moran) comprende a una familia de lenguajes (que incluye a NGOMSL, KLM) que se basan en la visión del usuario como un sistema procesador de información (modelo de procesador humano). El modelo GOMS se basa en el mecanismo de razonamiento humano para la resolución de problemas y realiza la formalización de aquellas actividades (físicas y mentales) que intervienen en esa labor. Para cada tarea se describe el objetivo a satisfacer (*Goal*), el conjunto de operaciones (*Operations*) que el sistema pone a disposición del usuario para la interacción, los métodos disponibles para llevar a cabo esas operaciones (*Methods*) y por último, un conjunto de reglas de selección (*Selection*) para determinar la alternativa más conveniente en cada caso (descritas mediante estructuras de control if-then). Cada tarea se podría descomponer en otras tareas primitivas formando un árbol jerárquico.

Los objetivos son las metas que se propone el usuario (lo que desea obtener). Los objetivos pueden servir como un punto de referencia en caso de un error. Un objetivo contiene información de la intención del usuario. Para ello, debe realizar una serie de operaciones básicas. Las operaciones son unidades elementales de percepción, motoras o actos cognitivos cuya ejecución es necesaria para cambiar algún aspecto del modelo mental del usuario, o bien, para modificar el entorno.

Este tipo de acciones puede afectar al sistema (pulsar una tecla) o bien, sólo al estado mental del usuario (leer el cuadro de diálogo). Existe un grado de flexibilidad acerca de la granularidad de las operaciones (amplitud de cada operación). Para llevar a cabo estas operaciones, existen varias posibilidades de descomposición de una tarea en subtareas. Por ejemplo, en un procesador de ventanas, se puede cerrar la ventana mediante ratón en un menú o teclado (atajo). Cada una de estas posibilidades será un método.

GOAL: CERRAR-VENTANA



```
[select GOAL: USAR-METODO-RATON
      MOVER-RATON-A-MENU-VENTANA
      ABRIR- MENU
      CLICK-SOBRE-OPCION-CERRAR
GOAL: USAR-METODO-TECLADO
      PULSAR-TECLAS-ALT-F4]
```

Cuando hay más de una alternativa, se puede indicar una serie de condiciones y reglas para tomar la mejor alternativa (método):

```
METHODS: IF (USUARIO-EXPERTO)USAR-METODO-TECLADO
           ELSE USAR-METODO-RATON]
```

Los objetivos se pueden descomponer en subobjetivos.

```
GOAL: EDITAR-DOCUMENTO
GOAL: ABRIR-DOCUMENTO
```

La descomposición de tareas suministra una comprensión de estrategias para resolver problemas del dominio de la aplicación. El objetivo del análisis jerárquico de tareas es la de producir una descomposición de tareas, de modo que se pueda seguir paso a paso el método de resolución.

GOMS puede servir también para medir rendimientos. La profundidad de subtarear se puede usar para estimar los requerimientos de la memoria de corto plazo (MCP) e incluso para estimar tiempo de respuesta (asumiendo tiempos constantes para cada operación).

Este ejemplo describe la tarea de mover una pieza en una partida de ajedrez.

```
GOAL: MOVER-PIEZA
  GOAL: DETERMINAR-TURNO
    [select GOAL: CONOCER-ULTIMO-MOVIMIENTO
            MOVER-A-LA-HISTORIA-DE-MOVIMIENTOS
            DETERMINAR-ULTIMO-MOVIMIENTO
            COMPROBAR-SI-NO-ERES-TU
    GOAL: CONOCER-MOVIMIENTO-SIGUIENTE
            MOVERSE-AL-TABLERO
            IDENTIFICAR-POSICION-DEL-RELOJ
            COMPROBAR-SI-RELOJ-ESTA-EN-TU-POSICION]
  GOAL: ELEGIR-MEJOR-ESTRATEGIA
  GOAL: REALIZAR-MOVIMIENTO
    GOAL: SELECCIONAR-PIEZA-ADECUADA
      [select GOAL: IDENTIFICAR-PIEZA
              SELECCIONAR-TECLADO
              ESCRIBIR-IDENTIFICACION-PIEZA
              CONFIRMAR
      GOAL: SELECCIONAR-PIEZA
              MOVER-CURSOR-A-PIEZA
              PULSAR-BOTON-RATON]
    GOAL: ELEGIR-DESTINO
      [select GOAL: IDENTIFICAR-DESTINO
              MOVER-CURSO-ARRASTRANDO-PIEZA
```



ESCRIBIR-IDENTIFICACION-POSICION
CONFIRMAR
GOAL: SOLTAR-PIEZA
MOVER-CURSOR-ARRASTRANDO-PIEZA
SOLTAR-BOTON-RATON]
GOAL: CONFIRMAR-MOVIMIENTO
[select GOAL: TECLA-CONFIRMACION
PULSAR-ENTER
GOAL: PARAR-RELOJ
MOVER-CURSOR-RELOJ
PULSAR-BOTON-RATON]
Selection Rule for GOAL: DETERMINAR-TURNO
Si es una visualización gráfica, usar el método CONOCER-MOVIMIENTO-SIGUIENTE
De lo contrario usar el CONOCER-ULTIMO-MOVIMIENTO
Selection Rule for GOAL: SELECCIONAR-PIEZA-APROPIADA
Si no tienes ratón usar el método IDENTIFICAR-PIEZA,
De lo contrario usar el método SELECCIONAR-PIEZA
Selection Rule for GOAL: ELGIR-DESTINO
Si no tienes ratón usar el método IDENTIFICAR-DESTINO,
De lo contrario usar SOLTAR-PIEZA
Selection Rule for GOAL: CONFIRMAR-MOVIMIENTO
Si estas usando el teclado usar el método TECLA-CONFIRMACION,
De lo contrario usar el método PARAR-RELOJ

El modelo GOMS fue uno de los primeros métodos utilizados para el diseño de interfaces de usuario, teniendo gran repercusión en investigaciones posteriores. Permite describir cómo un experto realiza tareas y las descompone en subtareas. Sin embargo, una de sus puntos débiles es que sólo considera comportamientos sin errores y tareas secuenciales.

4.5.2.3 KLM

Uno de los modelo GOMS más simple es el KLM (*KeyStroke Level Mode*), que simplifica el conjunto de operaciones disponibles a la pulsación de teclas y movimiento de ratón. Esta simplificación permite obtener predicciones del tiempo que una persona empleará para la realización de una tarea.

Estas mediciones parten de unos valores experimentales que determinan mediciones concretas para la realización de actividades elementales. A partir de estos resultados experimentales, se puede deducir que por término medio:

- El tiempo de planificación de una tarea se puede estimar que dura 2-3 seg. si ya está definida, y de 5 a 30 seg. si hay que pensarla.
- El tiempo de respuesta del usuario para una acción varía notablemente según el tipo de dispositivo y de la destreza del usuario. Se podría crear una tabla donde aparecen las operaciones más usuales sobre los dispositivos (pulsar tecla, mover ratón, etc.) y su tiempo en función de las habilidades del usuario.

Operador	Descripción	Segundos
K	Pulsación de teclado: Buen mecanógrafo (135 ppm)	0.08
	Habilidoso (90ppm)	0.12



	Normal (40ppm)	0.28
	Malo	1.20
P	Apuntar con el ratón	1.10
H	Ubicar las manos en teclado	0.40
D(N _b ,l _b)	Dibujar un trazo (N: nº segmentos, l: longitud)	0.9N _b +0.016l _b
M	Preparación mental	1.35

Por ejemplo, se puede comparar el tiempo de respuesta entre el diseño de dos menús con mucha profundidad o con mucha ramificación. En un menú basado en ordenes numeradas (16 opciones). Partiendo de las siguientes hipótesis:

- Opciones con igual probabilidad
- Usuario experto mecanógrafo

Se podría concluir con la siguiente medición. Una selección de un menú por su posición (1-16) necesitaría de los siguientes operadores:

- Operador M para buscar y visualizar el menú
- Uno o dos operadores K para introducir el número de ítem + 1K (pulsación de tecla ENTER)
 $M + K \text{ (1er dígito)} + 0.44K \text{ (2º dígito)} + K \text{ (Enter)}$
 $1.35 + 0.20 + 0.44(0.20) + 0.20 = 1.84 \text{ seg.}$
 ·Usado para las opciones de la 10-16. Probabilidad $7/16 = 0.44$

Se puede comparar con la utilización del ratón para la elección del menú. Para ello bastarán los siguientes operadores:

$$M + P + K \text{ (click ratón)}$$

$$1,35 + 1.10 + 0.20 = 2.65 \text{ seg.}$$

Para este caso KLM podría predecir que la selección por teclado sería más rápida (efectiva) que la selección por menú.

Se puede utilizar para analizar igualmente el balanceo de los menús (profundidad), selecciones de elementos por identificador o apuntando, etc. Otra utilidad que posee este método es la de hacer explícito el *conocimiento* implícito que debe poseer el usuario acerca del sistema.

4.5.2.4 TAG

Este es otro tipo de formalismo para representar el conocimiento del usuario. TAG (Task Action Grammar) describe el conocimiento del usuario para realizar una determinada tarea en base a una gramática con características. El conocimiento que posee el usuario del tipo "*como puedo hacer...*" se puede expresar mediante un esquema que engloba a un conjunto de reglas individuales.

Un esquema estará formado por una estructura sintáctica definida a partir de un conjunto de características. Las características (*features*) son atributos que definen la semántica de la tarea. Las tareas que tenga estructuras sintácticas diferentes tendrá un esquema diferente. A partir de esta aproximación, se podrá medir la complejidad de la interfaz mediante la identificación de



características en la acción de tareas y de su agrupamiento en esquemas. El número de esquemas puede dar una orientación de la consistencia del lenguaje de la interfaz; a menos esquemas, el usuario puede inferir el comportamiento esperado del sistema a partir de un conocimiento parcial. Se parte del hecho que la simplificación del esquema es comprensible por el usuario. De este modo, la consistencia ayuda al aprendizaje, ya que se pueden inferir nuevas tareas a partir de las conocidas.

Para abordar la especificación de un interfaz mediante TAG se deben seguir los siguientes pasos:

- Identificar tareas simples que el usuario puede realizar sin resolución de problemas (sin incluir estructuras de control).
- Describir las tareas simples en un diccionario (como conjuntos de componentes semánticos), reflejando la categorización de las tareas.
- Definición de reglas de reescritura que trasladan tareas simples en una especificación de acciones, donde los tokens son etiquetados con características semánticas procedentes del diccionario de tareas simples. Las etiquetas permiten la generalización.

Por ejemplo, se puede utilizar TAG para estudiar la consistencia del conjunto de órdenes de un Sistema Operativo (copy, rename, delete). Para ello, se parte de la descripción gramatical de estas tareas, y se analizará cómo se pueden agrupar en esquemas:

Tareas simples

Copiar archivo nombre_archivo nombre_archivo
Copiar disco nombre_disco nombre_disco
Renombrar archivo nombre_archivo nombre_archivo
Borrar archivo nombre_archivo

Primero, se identifican las características que aparecen en estas tareas y sus posibles valores. En este caso, las características corresponden con el tipo de atributo y sus posibles valores. También se debe tener en cuenta cuando el objeto fuente y destino es el mismo, ya que en tal caso, se produce un cambio en el objeto (y es una característica a tener en cuenta):

Objeto_antiguo [archivo, disco, vacio]
Objeto_nuevo [archivo, disco, vacio]
Produce_cambio [si, no]

Una vez obtenidas las características y sus valores, se deberán describir las tareas simples mediante estas características, de modo que definan unívocamente su significado:

Copiar archivo [Obj_antiguo=archivo, Obj_nuevo=archivo, cambio= no]
Copiar disco [Obj_antiguo=disco, Obj_nuevo=disco, cambio= no]
Renombrar archivo [Obj_antiguo=archivo, Obj_nuevo=archivo, cambio= si]
Borrar archivo [Obj_antiguo=archivo, Obj_nuevo=vacio, cambio= no]

A partir de estas tareas se podría obtener un único esquema a partir de tres características que lo definen.

Task [obj_ant, obj_nuevo, cambio] :=



Orden[obj_ant, obj_nuevo, cambio]+param1[Obj_ant]+param2[obj_nuevo]

Cada orden vendría dada por un nombre (de la tarea a realizar) y dos parámetros. Los valores para estos argumentos vendrían dados unívocamente por el valor de las características:

```
tarea[obj_ant=archivo, obj_nuevo=archivo, cambio=si] := "ren"
tarea[obj_ant=archivo, obj_nuevo=archivo, cambio=no] := "copy"
tarea[obj_ant=disco, obj_nuevo=disco, cambio=no] := "diskcopy"
tarea[obj_ant=archivo, obj_nuevo=vacio, cambio=no] := "delete"
param1[obj_ant=vacio] := NULL
param1[obj_ant=archivo] := drive-id + "nombre_archivo"
param1[obj_ant=disco] := drive-id
param2[obj_nuevo=vacio] := NULL
param2[obj_nuevo=archivo] := drive-id + "nombre_archivo"
param2[obj_nuevo=disco] := drive-id
drive-id := NULL | "A:" | "B:" | "C:" | ...
```

TAG no realiza predicciones absolutas de rendimiento. Mas bien, se puede utilizar para generar hipótesis que deben ser probadas experimentalmente.

TAG se ha utilizado para medir la consistencia de lenguajes de órdenes (UNIX, MSDOS), aunque también se puede generalizar para acciones de manipulación directa, ubicación de órdenes en menús, etc.

Consistente	Inconsistente
copy src_file target_file	copy target_file src_file
rename src_file target_file	rename target_file in src_file
delete src_file	make src_file deleted

La evaluación de la consistencia del lenguaje del interfaz de usuario puede ser muy sencillo de estimar con TAG. Así por ejemplo, se podrían detectar rápidamente inconsistencias en el lenguaje de órdenes como en este ejemplo, donde las órdenes no poseen la misma estructura. El esquema tiende a ser menos consistente conforme aumenten los esquemas. Como consecuencia, el número total de reglas no es importante, sólo el numero de esquemas.

La consistencia está muy relacionada con la facilidad de aprendizaje. Se podría estimar que existe una relación directa entre el tiempo de aprendizaje necesario y el número de esquemas que aparecen del lenguaje.

4.5.2.5 UAN

Las técnicas basadas en *gramáticas* no resultan adecuadas para la descripción de la interacción en interfaces gráficas de usuario basadas en el paradigma de **manipulación directa** ya que no permiten describir la realimentación visual (*feedback*) del sistema. A esto se une la dificultad de especificar tareas basadas en el arrastre de iconos ya que su semántica depende del lugar donde se suelte. En este sentido, **UAN** (*User Action Notation*) es una notación centrada en el usuario para descripción de tareas. Su principal característica es la descripción física de las acciones a realizar en el proceso de interacción.



Una especificación en UAN se realiza mediante una tabla dividida en 3 columnas describiendo las acciones de usuario, la realimentación de la interfaz y el estado del sistema tras la acción. Las acciones de usuario y la realimentación poseen una notación donde explícitamente se designa la manipulación que se realiza sobre los objetos de la interfaz. Así por ejemplo, las acciones **Mv** y **M[^]** denotan el efecto de pulsar y soltar el botón del ratón, **~[archivo]** representa el movimiento del ratón en el contexto del objeto *archivo* mientras que **archivo > ~** significa que el objeto *archivo* se está arrastrando. Del mismo modo se puede especificar la respuesta del sistema (*feedback*) para cada acción tales como la una selección en vídeo inverso de un icono (**archivo!**), vídeo normal (**archivo-!**), o un parpadeo (**archivo!!**). La siguiente descripción especifica la tarea de borrar un archivo mediante técnicas de arrastre.

	UAN	Realimentación	Estado
1)	~[arch] Mv	arch!, forall(arch!): arch-!	Selección = arch
2)	~[x,y]*	outline(arch) > ~	
3)	~[papelera]	outline(arch) > ~, palelera!	
4)	M [^]	Borrar(arch), papelera!!	Selección = null

Esta especificación permite especificar la tarea de arrastrar un archivo a la papelera mediante el proceso que se muestra gráficamente en la siguiente figura.

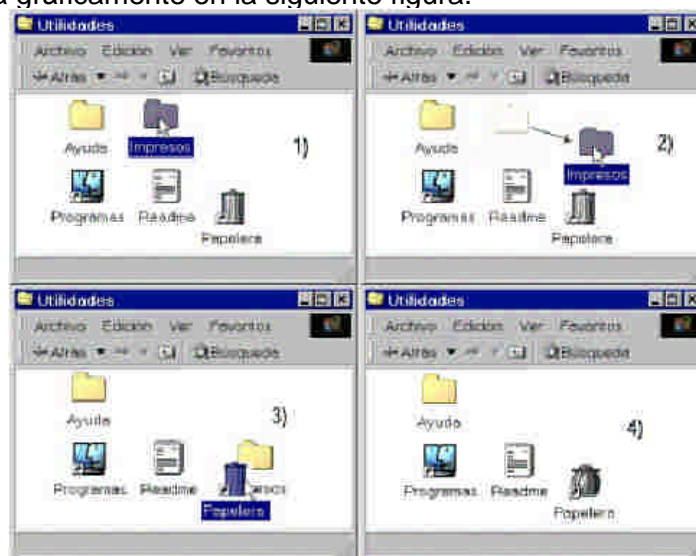


Figura 6 Manipulación directa descrita mediante UAN

4.5.2.6 Concur Task Trees (CTT)





CTT es una notación desarrollada por Fabio Paternó cuya principal finalidad es la de poder representar las relaciones temporales existentes entre las actividades y usuarios que son necesarios para llevar a cabo en las tareas. En concreto, esta notación es especialmente útil para aplicaciones CSCW (Computer Supported Cooperative Work – Soporte por computadora para trabajo en equipo).

Una de las principales ventajas de esta notación es su facilidad de uso, lo que hace que sea aplicable a proyectos reales con aplicaciones de un tamaño medio-largo y que con llevan especificaciones de cierta complejidad. La notación genera una representación gráfica en forma de árbol de la



descomposición jerárquica de las tareas existentes en el sistema. Se permite la utilización de un conjunto de operadores, sacados de la notación de LOTOS, para describir las relaciones temporales entre tareas (secuencialidad, concurrencia, recursión, iteración...)

Se pueden reutilizar partes de especificación para la creación de “árboles de tareas concurrentes” e identificarlo como un patrón de tarea. Existen 4 categorías de tareas en función del actor que la llevará a cabo.

	Tareas del usuario. Tareas realizadas completamente por el usuario, son tareas cognitivas o físicas que no interactúan con el sistema. Describen procesos realizados por el usuario usando la información que recibe del entorno (por ejemplo seleccionar dentro de un conjunto de información la que se necesita en un instante determinado para la realización de otra tarea).
	Tareas de la aplicación. Tareas realizadas por la aplicación y activadas realizadas por la propia aplicación. Pueden obtener información interna del sistema o producir información hacia el usuario. Como ejemplo, una tarea que presente los resultados obtenidos de una consulta a una base de datos.
	Tareas de interacción. Son tareas que realiza el usuario interactuando con la aplicación por medio de alguna técnica de interacción. Un ejemplo puede ser seleccionar un elemento de una lista desplegable.
	Tareas Abstractas . Tareas que requieren acciones complejas y que por ello no es fácil decidir donde se van a realizar exactamente. Son tareas que van a ser descompuestas en un conjunto de nuevas subtareas.

Para la descripción se utilizan además una serie de operadores temporales que facilitan la descripción de las relaciones temporales existentes entre tareas. Estos operadores se han obtenido como una extensión de los operadores existentes en LOTOS. El uso de estos operadores facilita la descripción de comportamientos complejos. Los operadores temporales que se pueden usar son:

T1 ||| T2. Entrelazado (Concurrencia independiente). Las acciones de las dos tareas pueden realizarse en cualquier orden.

T1 [|] T2. Sincronización (Concurrencia con intercambio de Información). Las dos tareas tiene que sincronizarse en alguna de sus acciones para intercambiar información.

T1 >> T2. Activar (enabling). Cuando termina la T1 se activa la T2. Las dos tareas se realizan de forma secuencial.

T1 [|>> T2. Activar con paso de información. Cuando termina T1 genera algún valor que se pasa a T2 antes de ser activada.

T1 [] T2. Elección. Selección alternativa entre dos tareas. Una vez que se esta realizando una de ellas la otra no esta disponible al menos hasta que termine la que esta activa.

T1 [> T2. Desactivación. Cuando se da la primera acción de T2, la tarea T1 se desactiva.

T1 [|> T2. Desactivación con paso de información. Igual que la anterior pero pasando información de una tarea a la otra.



T1*. Iteración. La tarea T1 se realiza de forma repetitiva. Se estará realizando hasta que otra tarea la desactive.

T1(n). Iteración finita. La tarea T1 puede darse n veces. Se utiliza cuando el diseñador conoce cuantas veces tiene que realizarse la tarea.

[T1]. Tarea opcional. No es obligatorio que se realice la tarea. Cuando se describen las subtareas existentes en la tarea de rellenar un formulario algunas de las subtareas pueden ser opcionales (las de los campos que sean opcionales).

A la descripción jerárquica de las tareas se le añade la descripción de los objetos que son manipulados por las distintas tareas. Estos objetos pueden clasificarse en dos grupos:

- **Objetos perceptibles.** Son objetos gráficos que sirven para presentar información al usuario (ventanas, tablas, gráficos...) o elementos sobre los que el usuario puede interactuar (menús, iconos...).
- **Objetos de aplicación.** Elementos que pertenecen al dominio de la aplicación. La información que almacenan estos objetos tiene que ser mapeada a alguno de los otros para poder ser percibida por el usuario.

Cada objeto puede ser manipulado por mas de una tarea. Como ejemplo de especificación utilizando los Concur Task Trees se puede ver una parte de una aplicación para acceder a información sobre un museo. En la siguiente figura se muestra el modo de introducir la información por parte del usuario sobre el tipo de artista que está buscando y el sistema le presenta la información asociada a el.

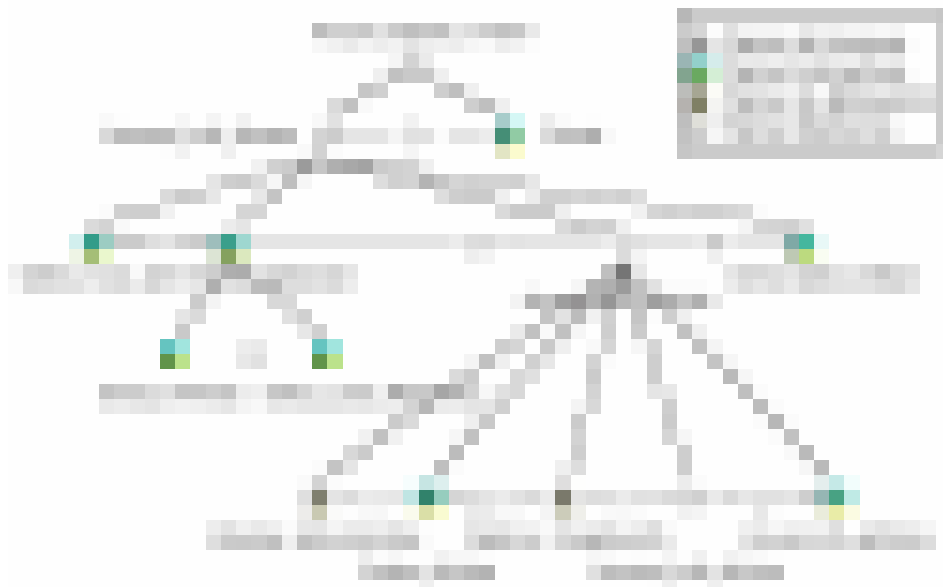


Figura 7 Tarea de acceso a un museo mediante CTT

Se comienza con la tarea “*Acceso_Museo_Virtual*” que puede ser interrumpida en cualquier momento ([>]) por la tarea “*Cerrar*”. En un siguiente nivel se puede ver la tarea de interacción “*Selecc_Tipo_Arte*” con la que se describe la selección del tipo de arte sobre el que se requiere



información, seguido (>>) por la selección del periodo histórico en el que está encuadrada la obra del artista. La siguiente tarea a realizar es el acceso a la información del artista para ello se ha obtenido información de las tareas anteriores ([]>> operador activación con paso de información). Algunas de estas tareas se van a descomponer en nuevas tareas de manera jerárquica. Así la tarea “*Selecc_adicional*” permite seleccionar un artista mediante una lista alfabética o mediante una lista ordenada por periodos históricos.

La descripción de tareas cooperativas consiste en tareas que requieren de la participación de más de un usuario para poder llevarla a cabo. Esta descripción se puede realizar creando un árbol donde aparecerán las tareas cooperativas, que se denotarán con un identificador especial. Por ejemplo, una solicitud de información se podría modelar del siguiente modo:

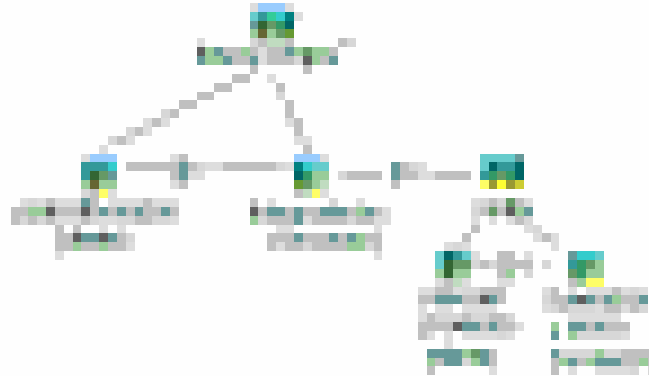


Figura 8 Tarea cooperativa con CTT

4.6 Modelos arquitectónicos

Los modelos arquitectónicos son una alternativa (en algunos casos complementaria) para la representación de sistemas interactivos. En este caso, se pretende obtener un modelo del sistema a desarrollar, centrándose en los aspectos computacionales básicos en los que debe estar basado.

Los primeros modelos que se proponen en este sentido son para describir el sistema interactivo en su conjunto obteniendo sus características esenciales. Si bien los primeros modelos como Seeheim o ARCH se centraban en aspectos funcionales, las últimas propuestas definen una arquitectura modular basada en componentes como MVC o PAC. También se pueden encontrar propuestas encaminadas hacia un diseño basado en modelos concretos del sistema con herramientas para su obtención automática como puede ser MOBI-D, HUMANOID, etc.

Los aspectos más relevantes de estas propuestas se centran en la modelización de los componentes interactivos (estructura) y mecanismo de interacción (diálogo).

4.7 Modelos abstractos

El estudio de los sistemas interactivos como sistemas reactivos permite estudiar propiedades relacionadas con estos sistemas tales como la ausencia de interbloques o inanición. Sin embargo, desde el punto de vista de la interacción con el usuario, las propiedades deseables del sistema están relacionadas con la parte de interacción con el humano, tales como: la propiedad de comenzar nuevamente (*Restartability*), deshacer la última acción en cualquier momento (*Undo*), utilizar toda



la funcionalidad de la aplicación (*completitud*), capacidad para cancelar en cualquier momento la ejecución de una tarea, observación en todo momento del estado del sistema (*Observable*), etc.. Básicamente este tipo de propiedades de interés se pueden resumir en los siguientes casos:

1) **Predecible**. Se trata de reconocer y predecir cual será el efecto futuro del sistema ante una nueva interacción. Esta propiedad permite medir la consistencia del sistema.

2) **Alcanzable**. Esta propiedad permite razonar y determinar si el usuario tiene acceso en todo momento a la funcionalidad del sistema. Esta propiedad permite medir la *completitud* del sistema.

En este sentido, los métodos que se utilizan son meramente descriptivos. La notación y el fundamento matemático en el cual están basados permiten expresar propiedades y por tanto, se podrán utilizar para razonar y verificar ese tipo de propiedades.



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

Licenciatura En Informática

Programación de Interfaces

*Autor: L.I. María de Lourdes Isabel
Ponce Vásquez*

AGOSTO-DICIEMBRE 2005



Contenido

UNIDAD 5. ESTRATEGIAS DE DISEÑO.....	3
Objetivos:	3
Introducción.....	3
5.1 Tareas de interacción	3
5.1.1 Tarea de posicionamiento	4
5.1.2 Tarea de selección.....	4
5.1.3 Tarea de introducción de texto.....	6
5.1.4 Tarea de introducción de valor.....	6
5.1.5 Tarea de arrastre	7
5.2 Procesamiento de entradas del usuario.....	7
5.2.1 Cola de eventos	7
5.2.2 Comunicación entre objetos.....	8
5.3 Criterios para un buen diseño.....	9
5.3.1 Control de usuario.....	9
5.3.2 Retroalimentación	10
5.3.3 Personalización.....	11
5.3.4 Dirección.....	12
5.3.5 Consistencia	12
5.3.6 Claridad	13
5.3.7 Estética.....	17
5.3.8 Indulgencia	20
5.3.9 El factor humano.....	21
5.3.10 Procesamiento de errores	21
5.4 Cohesión de ventanas.....	22
5.4.1 Cohesión funcional.....	22
5.4.2 Cohesión secuencial	23
5.4.3 Cohesión comunicativa	24
5.4.4 Cohesión procedural	24
5.4.5 Cohesión temporal.....	25
5.4.6 Cohesión lógica	25
5.4.7 Cohesión coincidental	25
5.4.8 Valoración de los niveles de cohesión.....	26



UNIDAD 5. ESTRATEGIAS DE DISEÑO

Objetivos:

- Analizar algunas estrategias y criterios de diseño

Introducción

En unidades anteriores se han mostrado algunos elementos importantes al momento de diseñar interfaces de usuario, en esta unidad, se pretende analizar algunos criterios estándares para el buen diseño aceptados en la industria, dar algunos consejos de experiencias propias en relación a algunas características de las interfaces e intentar unir los conceptos en situaciones concretas.

Con los modelos que se disponen de las tareas y de la arquitectura del sistema, se deberá guiar el diseño para obtener una implementación correcta. Este proceso en algunos casos puede ser automático, pero en la mayoría de los casos, requerirá de un profundo reconocimiento de los aspectos más delicados del proceso de diseño y que están directamente relacionados con el diálogo con la máquina y la presentación de la información. Se mostrarán los mecanismos básicos de interacción y el diálogo con la aplicación y la capa de presentación.

5.1 Tareas de interacción

Cuando el usuario realiza una interacción con la computadora, introduce una unidad de información que posee significado en el contexto de la aplicación. El tipo de interacciones que se pueden realizar en el sistema son:

- **Posicionamiento.** Obtención de una posición u orientación. La posición puede ser en cualquier dimensión (2D, 3D). Se puede usar para dibujo, situar ventanas, arrastrar elementos, etc.
- **Valor.** Obtención de un dato cuantificable, ya sea numérico (por ejemplo, el número de página, nº de copias, etc.) o porcentual (p.e. barra de progresión).
- **Texto.** Introducción de un texto o identificador (nombre de archivo, apellidos, etc.)
- **Selección.** Obtención de una alternativa (de entre varias posibles). Se puede distinguir entre la selección sobre un conjunto finito o bien de tamaño variable, en el cual las opciones pueden ser alternativas disyuntivas o no.
- **Arrastre.** Secuencia de posiciones entre una posición inicial y otra final (por ejemplo movimiento de una carpeta con el ratón).

A cada una de estas interacciones se le puede aplicar un conjunto de técnicas que permitan facilitar y flexibilizar el modo de obtención de la información.



5.1.1 Tarea de posicionamiento

Esta tarea consiste en la obtención de una coordenada 2D o 3D. La acción que se realiza es la de mover un cursor por pantalla para introducir un valor, o introducir la coordenada directamente. Algunos de los aspectos a tener en cuenta son:

- Sistema de coordenadas. El movimiento del objeto puede ser en función de un sistema de coordenadas del propio objeto, de la pantalla o del mundo. Este factor es muy importante en los sistemas de modelado de posicionamiento 3D (Realidad Virtual).
- Resolución: En caso de movimiento con un dispositivo (cursor) las posiciones pueden ser discretas o continuas. Habrá que tener en cuenta la relación Control/Display (C/D) para mecanismos de posicionamiento indirecto, es decir, la distancia que hay que mover el apuntador (ratón) en la mesa para obtener una nueva posición diferente en pantalla.
- Restricciones. Se pueden utilizar elementos que ayudan al posicionamiento como la **rejilla** (*grid*) que facilitan la introducción de los puntos ajustados a valores. Esta rejilla puede ser direccional (sobre una única dirección), modular (restringido a una retícula) o gravitacional (restringida a unos puntos sensitivos).
- Retroalimentación. La retroalimentación puede ser espacial (visualización gráfica del cursor en pantalla), relacionada con otros elementos, o lingüística (representando el valor numérico en coordenadas cartesianas).

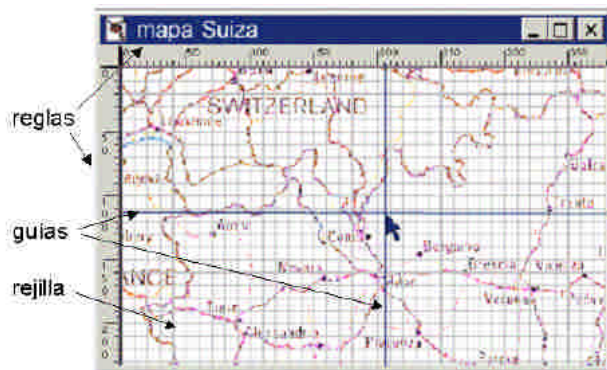


Figura 1 Diferentes técnicas de ayuda en el posicionamiento

5.1.2 Tarea de selección

Esta tarea básica consiste en la selección de un elemento de entre un conjunto (de órdenes, atributos, objetos..). El conjunto puede ser:

- Tamaño fijo: Elementos invariables (órdenes, etc.)
- Tamaño variable: Elementos de la aplicación (objetos)

La tarea de selección se puede realizar de los siguientes modos:

- Mediante identificador: La selección se realiza introduciendo el identificador del objeto (nombre)
- Apuntando: Búsqueda espacial del objeto mediante un dispositivo apuntador (ratón)

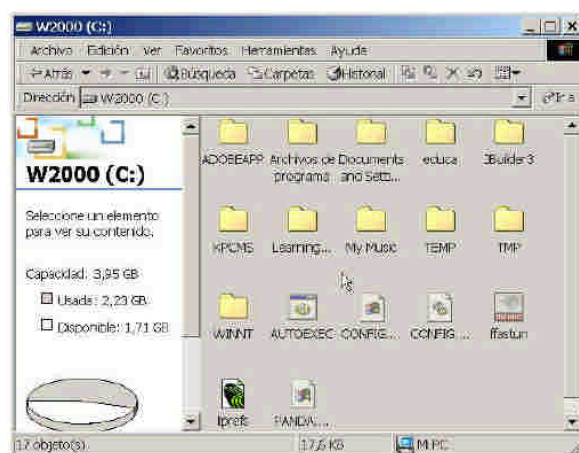


Figura 2 Selección en lista variable de objetos (gráficos)

La selección sobre conjunto de objetos de tamaño variable se organiza sobre listas dinámicas (combobox) o bien en ventanas (que actúan como contenedores) donde se disponen los elementos con diferentes alternativas de presentación (gráfica, textual, icónica, etc.) y modo de ordenación.

La selección sobre conjuntos de tamaño fijo utiliza otros mecanismos de representación. Uno de los más utilizados es la presentación mediante menús y se usa para la selección de órdenes. Cada orden es un ítem dentro del menú, y se pueden utilizar diferentes técnicas para estructurar el conjunto de elementos (jerarquías, separaciones, atajos, etc.).

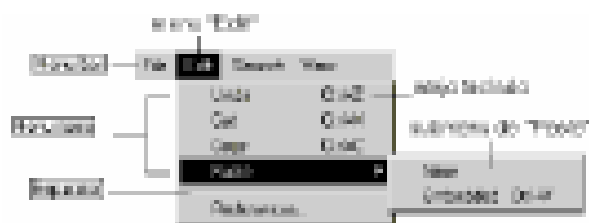


Figura 3 Organización de un menú

Otras posibles representaciones para la selección son los botones (o conjuntos de botones agrupados en barras de iconos). Los botones pueden tener una representación gráfica icónica o bien estar basado en el texto. Se pueden inhabilitar botones (representados en un tono claro) en caso que la orden no esté activa en ese momento.



Figura 4 Barra de botones (icónica)

Otras estructuras para la selección son la listas en sus diferentes representaciones (desplegables) o fijas.

Para el caso de selección sobre valores lógicos, se utilizan elementos que representen dos posibles estados (seleccionado o no). Generalmente se utiliza la casilla de verificación como elemento gráfico que representa un valor lógico (si no aparece seleccionada equivale a falso/no). En caso de seleccionarse la casilla, aparece un símbolo (✓) que recuerda a una marca hecha con pluma.



En caso de casillas de verificación, se pueden encontrar conjuntos de cuestiones donde las opciones son mutuamente excluyentes (sólo se puede seleccionar una de las casillas). Para diferenciar estos dos tipos de casillas gráficamente, tradicionalmente se ha utilizado una representación circular para las selecciones excluyentes y cuadrada para las casillas de comprobación. Estas casillas pueden estar incluidas en los menús.



Figura 5 Casillas de verificación

En ambos casos de selección, ya sea de tamaño fijo o variable, un factor muy importante es la ubicación espacial de los ítems (órdenes, objetos) para su rápida selección, el texto del identificador (para su reconocimiento) y la organización semántica (para recordar su posición).

5.1.3 Tarea de introducción de texto

Esta tarea consiste en la introducción de información textual. Existen alternativas a la introducción de texto por teclado como son los reconocedores de caracteres (OCR) y de acciones. Un aspecto importante relacionado con el texto es su resolución (tamaño en pantalla), que se mide en puntos o píxels. Se deben utilizar tamaños y tipos de letra que sean legibles y proporcionales a la resolución de pantalla que posea en ese momento el usuario. El texto puede ser de longitud variable y ocupar más de una línea, por lo que se consideran dos tipos de presentación, una entrada de tamaño fijo (con posible control del formato) o un área de texto (con uso de deslizadores).

5.1.4 Tarea de introducción de valor

Esta tarea consiste en la introducción de un dato cuantificable. En caso de identificar un número, la forma habitual es mediante el teclado numérico, aunque en determinados casos, se utilizan representaciones gráficas (diales, deslizadores, etc.) que ayudan a una introducción mediante ratón.

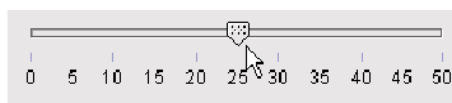


Figura 6 Deslizador

En caso de valores porcentuales (sobre un valor de tamaño variable o de tiempo para completar una acción) también se pueden utilizar representaciones, pero en este caso, no son directamente manipulables por el usuario (son informativas).



Figura 7 Barra de progresión y de posición



5.1.5 Tarea de arrastre

Esta tarea consiste en la introducción de una secuencia de posiciones que denotan un movimiento. Esta tarea típicamente se ha utilizado para describir explícitamente manipulaciones de objetos gráficos (mover, rotar, escalar), y que ha sido utilizada para dar significado a las operaciones de un sistema de escritorio (*drag and drop*) y en operaciones de diseño gráfico. Esta tarea requiere de una retroalimentación continua del objeto desplazado (puede ser una instancia, una línea/ caja elástica, un punto de una curva, etc.).

5.2 Procesamiento de entradas del usuario

El procesamiento de entradas se puede hacer mediante distintas técnicas de interacción entre aplicación–dispositivo. Esta interacción se puede realizar en diferentes modos: pregunta–respuesta, ordenes, etc. Los eventos es el principal mecanismo para la comunicación entre el usuario y el sistema interactivo. Cuando el usuario interacciona con los dispositivos, estas acciones se trasladan a eventos de software que se distribuyen a la ventana apropiada (en un sistema de ventanas). Todos los eventos están identificados (mediante un tipo de evento) que permite al software que los recibe distinguir la información que se pretende comunicar. Se pueden establecer tres mecanismos de comunicación entre usuario y aplicación:

- **Petición (*request*).** El programa espera hasta que se produzca una entrada. Los Dispositivos están en espera. Es un diálogo dirigido por la aplicación.
- **Muestreo (*sample*).** Ambos trabajan concurrentemente. Consulta del estado del dispositivo al realizar una petición. Los datos no son almacenados, por lo que se consulta el estado actual.
- **Evento (*event*).** Se provee de una cola de sucesos por parte del dispositivo. La aplicación está dirigida por los datos, y permite entradas asíncronas.

Los sistemas interactivos son programas dirigidos por eventos, y su estructura difiere de las aplicaciones tradicionales de procesamiento y de cálculo. El cuerpo principal del programa es simplemente un ciclo de obtención de eventos.

```
Inicialización
While(not salir) {
    Obtener siguiente evento E
    Procesar evento E
}
```

5.2.1 Cola de eventos

Existen diferentes modelos para distribuir los eventos a los objetos. Al manipular el sistema interactivo, los eventos se ponen en una cola de eventos. Todos los sistemas de ventanas poseen rutinas para obtener el siguiente evento de la cola. A menudo, los eventos no poseen suficiente información como para ser procesados o son irrelevantes (movimiento continuado del ratón, pulsación de teclas no habilitadas), por lo que se deben proveer mecanismos de filtrado para eliminar aquellos eventos que no son significativos.



Algunos de los tipos de eventos que se pueden encontrar son:

- **Eventos de entrada.** Son los generados por el usuario. Clasificación: *eventos del ratón*. El evento del ratón siempre posee la posición actual del ratón. *Teclado*: Se puede considerar un arreglo de botones de ratón.
- **Eventos de las ventanas.** Se reciben de la propia ventana. La mayoría de los sistemas de ventanas envían un evento de creación/ destrucción de la ventana, que permite al código de la aplicación procesar la acción pertinente. Tipos de eventos: crear/ destruir, abrir/ cerrar, iconificar/ deiconificar, redimensionar, mover, etc.
- **Eventos definidos por el usuario.** Eventos de alto nivel creados por software.

5.2.2 Comunicación entre objetos

Un punto importante del proceso de eventos es la comunicación con otros objetos interactivos. Un ejemplo de ello es el movimiento de la barra de deslizamiento, que provoca el cambio del texto presentado (relación entre dos componentes interactivos). Esta comunicación se realiza mediante pseudo-eventos (eventos que han sido creados por la comunicación entre objetos, y no son realmente eventos de entrada).

En sistemas de ventanas no orientados a objetos, un evento consiste en un registro de un tipo (entero), información estándar y otra adicional suministrada por el objeto que origina el evento. Existen tres modelos básicos para este tipo de comunicación:

- **Modelo de llamadas (*callbacks*).** Permite asociar un procedimiento (código C) a un widget en tiempo de ejecución. Por ejemplo, se utiliza en sistemas de ventanas como Motif, y primeras versiones de XWindows y lenguajes como Xtk, TK. Un callback es un procedimiento que se ejecuta inmediatamente después que se produce el evento. En Tcl/Tk, la conexión entre objetos (interactivos) evento y procedimiento se realiza mediante la siguiente orden `bind .boton1 <Left-Button> { puts "Boton %b situado en %x,%y" }`. Con esta orden se asocia al objeto interactivo “.boton1” cuando sucede el evento `<Left-Button>` (pulsación del botón izquierdo del ratón) una secuencia de órdenes entre llaves.
- **Modelo de notificación.** Cada componente interactivo notifica a su ventana padre (de la que depende jerárquicamente) la ocurrencia de un evento significativo. Este modelo se utiliza en las primeras versiones de MS Windows. Suministra un mecanismo interactivo más estructurado. Los componentes notifican a la ventana de nivel superior el evento sucedido. La ventana padre posee todo el código para decidir qué realizar con los eventos recibidos. Se puede prestar atención a eventos específicos (por ejemplo a eventos de aceptar–rechazar dentro de una ventana de opciones), que dirigen el modo de consulta y obtención del resto de eventos.
- **Modelo de conexión o delegación.** Permite a un objeto comunicarse con cualquier otro objeto (u objetos) mediante registro de los objetos receptores para un determinado evento. Modelo utilizado en NextStep o Java. Permite a los objetos comunicarse directamente entre ellos. Por ejemplo: comunicación entre la barra de desplazamiento con el editor de textos, invocando un método del objeto editor de textos.

La organización de la información afecta a la impresión general del interfaz. Debe incluir los siguientes elementos:



- Diseño (formato de pantalla, organización general)
- Representación de la información (métodos de codificación)
- Retroalimentación (seguimiento)
- Comunicación con el usuario (errores, mensajes de estado, etc.)

En el nivel de presentación se debe mostrar la información de forma que sea comprensible y asimilable por el usuario. Se puede pecar de exceso de información (provocando fatiga y desbordamiento al usuario) o demasiado sobria (la falta de información puede causar incertidumbre). Hay que tener en cuenta que el espacio donde se presenta la información es escaso, por lo que se deberá hacer un uso racional de este recurso. La distribución (no oclusiva) de la información es un factor muy importante, y se puede jugar con diferentes tipos de diseño: uso de zonas fijas o que se puedan ocultar (de forma dinámica), disposición del área trabajo: menús, mensajes, información de estado, etc.

Para la representación de objetos se deberá usar una simbología y codificación para que sea fácilmente identificable, económica en recursos (tamaño en pantalla) y consistente (iconografía, colores, texto, etc.).

5.3 Criterios para un buen diseño

Antiguamente los monitores monocromáticos estaban basados en caracteres, por lo que se dividían en cuadrículas de 80 caracteres de ancho y 24 líneas de alto. Cada celda de la cuadrícula desplegaba un solo carácter. Al programar, se debía indicar qué carácter se mostraría en cada intersección de columna y renglón. Debido a estas limitaciones casi toda la información se manejaba con códigos y abreviaturas, en cambio actualmente, gracias a los sistemas cliente/ servidor GUI que permiten definir tipos de letra dimensionables, ventanas emergentes y listas desplegables que ha incrementado el espacio disponible en la interfaz y ya que el costo de almacenamiento en disco a disminuido, se pueden evitar estas técnicas.

Los monitores monocromáticos basados en caracteres también tenían una capacidad limitada para gráficas. La mayoría de las imágenes se presentaban con guiones y líneas; actualmente, los gráficos sólo están limitados por la resolución cada vez mayor de píxeles en el monitor de color.

En las terminales de mainframe las pantallas se presentaban de una a una. Se escribía código procedural para colocar algo en la pantalla. El procesador, incapaz de detectar teclazos individuales, debía esperar hasta que el usuario presionara Enter o alguna tecla de función antes de obtener el contenido de la pantalla y procesar los resultados. Esto difiere de las PC, donde el objeto ventana contiene código para detectar eventos cuando suceden. Cada teclazo y clic de ratón tiene capacidad de comunicar información al procesador. Se requiere gran potencia de procesamiento local en cada terminal para que esto sea posible.

Muchos de los principios que están tras el buen diseño GUI no son nuevos. La mayoría de los usuarios preferiría usar una pantalla en monitor monocromático bien elaborada, en vez de una GUI con un diseño pobre.

5.3.1 Control de usuario

El usuario debe tener el control completo de la GUI, la retroalimentación inmediata y obvia para cada



acción mejora el sentimiento de control y reduce la frustración con el sistema. Aunque el usuario tiene más control que antes, demasiado control puede ser peligroso. Llevado al extremo, el usuario tiene la habilidad para moverse libremente de ventana a ventana y hacer cualquier cosa que desee.

El ratón hace que sea irrelevante el control del orden de tabulación, debido a que no hay garantía de que un campo sea dado antes que otro. Obviamente en una aplicación de negocios, la integridad de los datos se puede ver amenazada, y al usuario no se le puede dar simplemente el dominio libre que se podría esperar en un procesador de palabras. Debido a que el ambiente de ventanas permite realizar “cualquier cosa”, la tarea del diseñador de aplicaciones empresariales es restringir qué puede hacer el usuario en cualquier momento. El usuario debe tener el control, pero no al grado de pueda dañar los datos.

Algunos usuarios prefieren el ratón y otros no. El ratón es bueno para las tareas de selección y navegación de apuntar y disparar, pero puede degradar la velocidad de mecanografía de un capturista. Una buena aplicación GUI debe tener facilidad de uso para un “ratonero” o para un “buen mecanógrafo”. Por esto, siempre se debe incluir el orden de tabulación y las teclas aceleradoras (letra subrayada de un elemento de menú o botón que proporciona el equivalente de teclado al hacer clic sobre el elemento), para que cualquier acción que pueda realizarse con el ratón también se pueda con el teclado sin el ratón. La desventaja de esto es que la interfaz debe probarse tanto para las acciones de teclado como para las del ratón, lo que aumenta el esfuerzo de aseguramiento de calidad.

5.3.2 Retroalimentación

La aplicación debe indicar claramente si obtiene el control ó lo quita al usuario. La retroalimentación es de gran importancia en los sistemas interactivos, ya que el usuario debe estar informado de forma tangible e inmediata de las acciones que realiza. Cuando por ejemplo una tarea tarda más tiempo del razonable, se deberá informar mediante algún tipo de mensaje de ese proceso para no provocar incertidumbre. Esto se logra, por ejemplo, cambiando el apuntador del ratón a un reloj de arena o a un indicador de espera, y atrapando cualquier clic de ratón adicional que realice el usuario y que atrape el ambiente operativo. Sin embargo, un problema que se deberá tener en cuenta es que se tenga un procesamiento rápido, ya que en tal caso puede no coincidir el estímulo con la respuesta del sistema. Algunos ejemplos de retroalimentación son:

- **Sobre las órdenes.** Mostrar efectos, errores, confirmación, indicadores.
- **Sobre la selección.** Resaltar de forma no ambigua la orden activa.

Esta retroalimentación debe ser fácil de leer y entender. Para ello se debe fomentar la consistencia, y en algunos casos puede condicionar la estructura de datos del modelo, porque será necesario almacenar información adicional para realizar el feedback.

Para su diseño, se deben estudiar las acciones de cada tarea y ver cómo es la interacción (retroalimentación de la propia acción), confirmación (selección, mensajes, iluminación) y posibles errores (pantalla de aviso) La retroalimentación informa al usuario acerca de su interacción con el sistema. Indica qué está haciendo y le ayuda a realizarlo correctamente. Se puede utilizar cualquier combinación de canales sensoriales (sonoro, visual, táctil, etc.)

La retroalimentación se puede clasificar por su dimensionalidad temporal como:

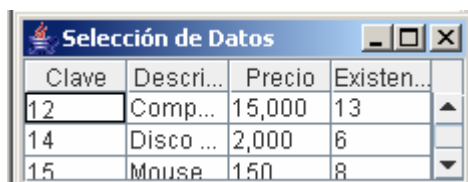


- **Futura.** Retroalimentación de una acción antes de llevarla a cabo. Indica qué sucederá si se realiza una acción (por ejemplo, etiqueta informativa de los botones).
- **Presente.** Retroalimentación durante la interacción. Indica qué está sucediendo actualmente (p. e. borrado de archivos, formateando, mover cursor, etc).
- **Pasada.** Información de lo que ha sucedido, y cómo ha cambiado el sistema (p. e. información de finalización de tareas).

Las generaciones actuales se aburren rápidamente. Si la aplicación procesa durante mucho tiempo, se puede usar una barra deslizante para indicar el porcentaje de trabajo que se ha realizado, o distraer al usuario con algún mensaje informativo en la barra de estado. Cuando el sistema no responde rápidamente, muchos usuarios lo interpretan como falla del sistema y volverán a arrancar la computadora a mitad del trabajo.

5.3.3 Personalización

La personalización es una característica importante de las interfaces gráficas de usuario. No todos los usuarios tienen las mismas necesidades, y el programador puede dejar muchas de las tareas de personalización a los usuarios. Es útil permitir, por ejemplo, que los usuarios reorganicen y redimensionen las columnas en un conjunto de resultados grande, ya que cuando hay muchas columnas no se pueden desplegar todas en pantalla, por lo que el usuario puede reorganizar dependiendo de cuáles columnas son más importantes para él. Los usuarios deben poder almacenar sus preferencias para que la siguiente vez que recuperen la ventana, su orden siga intacto. En cualquier momento debe poder regresar el orden de las columnas a la posición predeterminada por el programador. Esta técnica es muy útil en conjuntos de resultados de sólo lectura. Es recomendable tener precaución con el uso de tablas para ventanas de captura de datos, ya que el usuario puede eliminar de la vista alguna columna crítica.



Clave	Descri...	Precio	Existen...
12	Comp...	15,000	13
14	Disco ...	2,000	6
15	Mouse	150	8

Figura 8 Ventana estilo tabla

Muchos lineamientos GUI publicados dicen que los usuarios deben ser capaces de personalizar los colores de la interfaz, pero existen algunos problemas con esto, ya que esto puede ser bueno para un sistema operativo, pero no se debe permitir a los usuarios que usan aplicaciones críticas del negocio que hagan menos visible la información importante de la interfaz, algunas personas se quejan de que sus pantallas se han puesto en blanco, cuando cambian los rótulos de los campos al mismo color de fondo.

Para aplicaciones críticas se deben bloquear los colores. Los colores pueden usarse como pistas visuales para informar al usuario si un campo es opcional, requerido o prohibido. Un esquema razonable es hacer que los campos opcionales sean blancos, los prohibidos gris claro y los requeridos de un color reconocible pero pasivo, como el cian pálido. Los campos de datos son



colocados sobre un fondo gris suave que es cómodo para los ojos. Bajo estas circunstancias, la modificación de los colores cambia el significado de la aplicación.

5.3.4 Dirección

El concepto de dirección es bien soportado por el paradigma objeto-acción. Es mucho más intuitivo localizar el objeto que se desea y manipularlo directamente que el emitir un comando críptico y declarar el objeto al que se aplica un comando. El ratón es una herramienta excelente para este tipo de diálogo “apunte y dispare”. Por ejemplo, existe mucha diferencia entre borrar un archivo con un comando de MS-DOS y hacerlo en un ambiente GUI, la estructura del comando es inversa con respecto a MS-DOS.

Cuando se ha seleccionado un objeto, se pueden iniciar diversas acciones. Cada acción que se puede realizar debe estar disponible ya sea en un elemento de menú o en un botón de comando. La característica de arrastrar y soltar al borrar un objeto va más allá, agregando una pista visual (el bote de basura), para que el usuario no tenga que buscar en el menú la opción deseada.

La dirección puede ser aplicada también en las aplicaciones de negocios. Una disposición típica para una ventana objeto-acción es donde el usuario puede recuperar una lista de objetos, por ejemplo pedidos, y aplicar acciones desde una barra de botones o un menú para cualquier renglón que seleccione.

Las pistas visuales, tales como los iconos y barras de herramientas, también pueden agregarse a aplicaciones de negocios para mejorar la facilidad de uso y dirección de las aplicaciones. Se debe tener cuidado de no exagerar en el número de iconos muy agradables en el sistema. Los iconos muy vistosos pueden saturar una aplicación. Los iconos con ilustraciones y las barras de herramientas deben agregarse a una aplicación sólo cuando mejoran la habilidad del usuario para lograr sus tareas.

5.3.5 Consistencia

Una aplicación debe ser consistente con el mundo en que los usuarios viven y trabajan diariamente. Cuando la tecnología cliente/ servidor invade áreas de la compañía que no estaban automatizadas, éste puede ser un reto para el analista y diseñador. Las unidades de negocios que realizan la misma función a veces tienen un vocabulario de la industria y terminología completamente diferentes para las mismas ideas. Las compañías construidas mediante fusiones y adquisiciones encuentran este problema particularmente molesto. El analista tiene dos opciones, personalizar la aplicación para mapear la cultura histórica de cada lugar o aplicar reingeniería al negocio para obtener un consenso sobre la terminología y los procesos del negocio. Este tipo de asuntos por lo general requieren guía de un comité de dirección.

Para lograr consistencia en los rótulos de datos a lo largo de la aplicación, el proyecto debe tener un conjunto de nombres de rótulos estándar para cada atributo del modelo de información. Muchas de las herramientas de desarrollo de GUI principales soportan este tipo de extensiones al modelado de datos. Se debe incluir un rótulo estándar a usar: los rótulos pueden estar a la izquierda del campo en un formato libre y encima del campo en un formato de columnas. Para atributos como el dinero, porcentajes y números telefónicos, también se deben especificar los formatos y máscaras



predeterminadas. Cuando se deba abreviar un rótulo se deberá usar una lista de abreviaturas estándar para que los rótulos siempre sean consistentes en toda la aplicación.

Otro aspecto importante de la consistencia es que las aplicaciones deben ser funcionalmente consistentes dentro de la misma aplicación. Esto se logra mediante el apego a estándares para el aspecto y sensación, nombres de menús, nombres y colocación de botones de comando y uso racional y consistente de otras características GUI. Muchos de estos estándares GUI pueden ser comprados de vendedores de terceros o tomados de otros proyectos. Un proyecto GUI nuevo no debe perder tiempo argumentando sobre el tamaño de letra para la barra de título. Cuando se trata de los estándares se deben recordar los mandamientos del diseñador:

- Deberás pedir prestado
- Deberás tomar prestado
- Deberás robar
- Deberás desear la aplicación de tu prójimo

Las aplicaciones de negocios también deberán ser consistentes con el aspecto, sensación y lenguaje de otras aplicaciones. Es mucho más fácil entrenar a un usuario para usar una aplicación empresarial si es visual y funcionalmente consistente con su procesador de palabras, hoja de cálculo y correo electrónico. Con la computación cliente/ servidor, la aplicación no es solamente el fragmento que se construye. Es todo lo que el usuario necesita en su escritorio para realizar su trabajo.

Siempre se puede ir a los paquetes populares, como procesadores de texto y hojas de cálculo, para buscar lineamientos sobre la manera de definir conceptos de menú y botones de comando, sin embargo, no todos los proveedores son consistentes con sus propios paquetes. Han surgido algunos estándares de facto que siempre deben seguirse, tales como palabras reservadas y la colocación de determinados conceptos de menú como Archivo, Ventana y Ayuda. La violación de los estándares aceptados de la industria puede dar como resultado aplicaciones extrañas.

5.3.6 Claridad

La información que se presenta en la interfaz debe ser inmediatamente comprensible, y el uso de la aplicación debe ser visualmente evidente. La claridad puede ser un reto significativo para la aplicaciones GUI que están remplazando la tecnología monocromática. Una aplicación GUI debe tener un uso limitado y consistente de abreviaturas para valores de datos y rótulos. Esta es una oportunidad para comenzar a dismantelar la montaña de códigos crípticos, abreviaturas y códigos mnemónicos que se han estado juntando desde que las computadoras ingresaron a las organizaciones.

El concepto de consistencia dice que hay que usar el lenguaje del mundo real de los usuarios. El concepto de claridad toma en cuenta que algunos de los lenguajes de los usuarios son surrealistas, y por lo general es resultado del diseño de sistemas heredados. Se debe tratar de alejar a los usuarios del vocabulario del sistema antiguo, ya que no tiene sentido en el nuevo sistema (usar un número para un reporte en vez de un nombre que lo identifique).

Los sistemas anteriores tenían muchos códigos que se han metido en el lenguaje diario del negocio. No todos son malos, pero sí deben ser cuestionados. La siguiente figura muestra una forma de pedido de un sistema antiguo. En la base de datos existe número de cliente, código de transporte,



número de planta y código de vendedor como referencias a claves externas de las tablas cliente, métodos de transporte, planta y vendedor, respectivamente.

ID: 0E023	Encabezado de Pedido	10:20:03

NUM_ORDN:	102304	
NUM_CLIEN:	56023	[Fábrica Inter.]
MO_TRANS:	03	FECHA PEDIDO: 12/21/96
NUM_PLAN:	12	INIC_VENDE: BST
>:		
PF1-AYUDA	PF9-PRIM	PF10-ANT
PF11-SIG	PF12-ULT	

Figura 9 Pantalla monocromática de Pedido

Pedido					
Num_pedido	Num_cte	Cod_Transp.	Fecha_ped	Num_planta	Cod_per_ven
102304	56023	03	21-12-96	12	BST

Figura 10 Renglón parcial de la tabla de pedidos

No existe problema con el número de pedido, pero si sería molesto si la única forma de obtener un pedido en pantalla fuera teclear el número exacto. En un sistema bien elaborado que soporta el paradigma objeto-acción, los usuarios deben ser capaces de obtener una lista de pedidos por diversos criterios de selección en caso de que no sepan su número.

En cuanto al número de cliente, parece que la forma de obtener un cliente en un pedido es teclear su número. El nombre, que está mal abreviado en el sistema antiguo, se despliega a la derecha sólo cuando se teclea un número válido. Esto causa que el usuario consulte una lista de números de clientes, o lo peor, los memorice. No existe problemas con los números de cliente en sí mismos, es una forma cómo de proteger el sistema en caso de que el cliente cambie su nombre. Sin embargo, en una interfaz moderna, el usuario debe tener la habilidad de buscar el nombre del cliente para identificar al cliente. Un método razonable es permitir la introducción de una cadena parcial en el campo nombre de cliente y regresar una lista de clientes candidatos en una ventana de respuesta desde la cual el usuario puede elegir uno. Si el usuario teclea correctamente el nombre completo, y se encuentra en la base de datos, no hay que molestarlo con la ventana de respuesta. El número de cliente puede usarse como un campo opcional, sólo para presentarlo o eliminarlo completamente de la ventana.

El método de transporte es el peor código. Hay varias formas en que esta empresa envíe sus productos terminados. Si se observa la tabla de control para los métodos de transporte se encontrarán los registros:

Métodos de Transporte	
Código	Descripción
01	Camión



02	Ferrocarril
03	Contenedor
04	Carga a granel
05	Aéreo

Figura 11 Tabla de métodos de transporte

En las tablas de control que contienen solamente un código y descripción, se deben seguir ciertas reglas para desaparecer los códigos. Para implementar un conjunto de valores de atributos restringidos sin usar una tabla de control se deben satisfacer las siguientes condiciones:

1. **Los valores deben ser permanentes.** Los códigos son ampliamente usados en lugares donde el nombre tienen alguna probabilidad de cambiar durante el tiempo de vida del sistema. Si es extremadamente improbable que lleguen a cambiar los valores, tal vez no se necesite el código.
2. **Los valores no deben ser demasiado largos.** Otra razón por la que se emplean códigos es porque la descripción ocupa demasiados caracteres, haciendo que sea difícil ponerla en ventanas y reportes. Antes algunas organizaciones creaban un código para cualquier descripción que fuera de más de seis o diez caracteres. Actualmente muchos proyectos han elevado ese límite a 20 o más. La decisión debe tomarse de acuerdo al espacio de las ventanas y los reportes y no al espacio en disco.
3. **Los valores no deben tener otros atributos.** Si el atributo en cuestión tiene atributos adicionales propios, es una entidad por sí y debe permanecer en una tabla separada. Sin embargo, el que se use el valor literal del dato como clave primaria dependerá principalmente de los conceptos 1 y 2.
4. **Hay un código en la aplicación que depende de los valores.** Una razón principal para la creación de tablas de control para valores de datos es permitir que el usuario pueda extender el conjunto de valores válidos. La extensibilidad definida por el usuario funcionará solamente si no hay código dependiente de los valores actuales. En el caso del método de transporte del ejemplo, se puede encontrar algo de código en el sistema que siga las siguientes reglas de negocio:

```
If método_de_transporte = "camión" then
    //Se requiere nombre de compañía transportista
else If método_de_transporte = "ferrocarril" then
    //Se requiere nombre de compañía transportista
    //Se requiere plan de ruta del vehículo
else If método_de_transporte = "contenedor" then
    //Se requiere registro de contenedor
else If método_de_transporte = "carga a granel" then
    //Se requiere registro de carga a granel
    //Se requiere dimensiones del embarque (altura, longitud, anchura)
else If método_de_transporte = "aéreo" then
    //Se requiere que el peso no exceda de parámetro de sistema (límite de peso aéreo);
```

Cuando se tiene este tipo de código dependiente de valores en el sistema, no hay forma en que el usuario pueda agregar valores a la tabla sin una intervención del programador para alterar la aplicación.

5. **El nuevo sistema no necesita conservar traducción de código para la comunicación con otros sistemas.** Si el nuevo sistema debe comunicarse con otros sistemas antiguos que requieren los códigos antiguos, entonces las tablas de traducción de código tendrán que ser parte del nuevo sistema. Aunque éste sea el caso, no se debe molestar a los usuarios con los códigos antiguos. 15



Se deben guardar y desplegar los valores literales de los datos y usar una tabla de traducción para comunicarse con los sistemas antiguos.

Dado este conjunto de criterios, pareciera que se puede eliminar el código de método de transporte. Una lista desplegable en la interfaz permitirá que los usuarios seleccionen el método sin tener que memorizar el código. La lista desplegable no hará más lentos a los que teclean por tacto, debido a que los conceptos pueden seleccionarse desde la lista simplemente tecleando la primera letra del campo. La lista de valores puede cargarse a partir de una tabla y ser mantenida por el administrador del sistema o, si es completamente permanente, codificarla directamente. Otra ventaja de este método es que los renglones de pedido contienen ahora el valor literal, lo cual hace que sean más fáciles los reportes y las consultas de usuario final al eliminar la unión de tablas.

En relación al número de planta, es común el uso de un número como clave primaria de la tabla planta. Puede ser que en el mundo real las personas hagan referencia a una planta con un número. En el nuevo sistema GUI se esperaría al menos que el nombre de la planta se desplegara junto con el número de la planta y, al igual que el nombre del cliente, permitir que se seleccione de una lista. Dos de las desventajas del uso de números es que los nuevos empleados tienen que memorizar la lista y se tienen huecos en la secuencia si se cierran algunas plantas.

El último código es el de vendedor. Pareciera que el diseñador del sistema antiguo uso las iniciales de los vendedores, lo cual en Estados Unidos puede cambiar con el matrimonio, pero los códigos de vendedor no pueden modificarse debido a que está por todos lados del sistema como clave externa en cada pedido colocado.

Los códigos mnemónicos basados en nombre hacen complicadas las claves primarias para cualquier tabla en donde es probable que cambie el nombre. En el nuevo sistema, el personal de ventas deberá tener identificadores aleatorios que nunca son desplegados ante los usuarios. Éstos pueden ser generados por el sistema de administración de base de datos cuando se inserte un nuevo renglón. La interfaz debe permitir la selección entre una lista de nombres actual. Para los reportes donde los nombres son muy largos se puede usar una concatenación de sus iniciales actuales.

La siguiente figura muestra la nueva ventana Encabezado de pedido como aparecería en una GUI. La claridad es mejorada mediante el lenguaje del mundo real para conceptos que previamente requerían la memorización de códigos. Se han ocultado los códigos de la vista en lugares donde no hay razón para que el usuario los vea. Con el método de transporte se eliminó el código completamente y se almacenarán los valores literales en la base de datos, debido a que los valores de los datos no cambian a través del tiempo, no son muy largos, no tienen otros atributos, hay códigos dependientes de los valores y no se tiene que mantener una tabla de traducción a otros sistemas que usen los códigos antiguos.

Número de Pedido:	102304	
Nombre Cliente:	Fábrica Internacional, S.A. ?	# Cte: 551123
Modo Transporte:	Contenedor	
Fecha Pedido:	21/12/1996	
Número Planta:	12 - Centro ?	
Nombre Vendedor:	Beatriz Soto Torres	

Guardar Cerrar



5.3.7 Estética

La composición y disposición de una ventana debe ser visualmente agradable. Deberá atraer la vista hacia la información que es más importante. Como se indicó anteriormente, la fijación y movimiento del ojo humano muestran que la mayoría de la gente ve primero la parte superior izquierda del centro de la pantalla y rápidamente hace un barrido en una dirección en sentido del reloj. Por medio de algunas investigaciones han derivado principios que definen un sentido de estética agradable para culturas occidentales. Estos principios incluyen una sensación de balance, el uso de elementos espaciados y alineados con regularidad, simetría, patrones predecibles, economía de estilos, colores y técnicas, arreglos secuenciales de elementos para guiar la vista, unidad de ideas relacionadas, un sentido de proporción, simplicidad y el agrupamiento de elementos relacionados.

El significado de una imagen puede ser más fácilmente percibido por el observador si posee claridad visual. Se deberá por tanto enfatizar la organización lógica de la información. Para conseguir una buena organización se pueden utilizar las reglas de Gestalt, que permiten mejorar la claridad visual de la información. Estas reglas se basan en cómo organiza el observador los estímulos visuales (de forma global) y se pueden resumir en los siguientes principios:

- **Similitud.** Elementos visuales con propiedad común se interpretan como agrupados.
- **Proximidad.** Objetos similares próximos se interpretan como una representación conjunta/agrupada.
- **Cierre (clausura).** Elementos visuales que tienden a cerrar un área se interpreta como cerrada.
- **Continuidad (determinación de formas).** Discriminación de elementos diferentes según la continuidad natural.

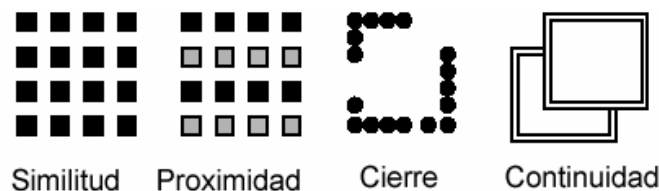


Figura 8 Reglas de claridad visual (Gestalt)

Estas reglas se aplican frecuentemente al diseño visual de los sistemas gráficos, como por ejemplo en la colocación de los botones, elementos de menú, organización general del interfaz, etc.

Cuando las ventanas comienzan a estar muy saturadas, hay varias técnicas para limpiarlas. Se puede recurrir al modelo de información para ver si hay una división lógica entre dos entidades diferentes que comparten la ventana. Una relación, como la clásica encabezado/ detalle, es una forma de dividir una ventana.

Si la ventana maneja más de un evento del negocio probablemente se puede dividir en ventanas separadas (continuas). Puede ser que la ventana maneje diferentes permutaciones del mismo evento de negocios. Se debe analizar si todos los datos se necesitan para una transacción normal. Los datos que se requieren para el evento de excepción, en vez de los normales, pueden separarse en otra ventana.



Si el usuario realmente debe tener todos los datos residentes en pantalla, tal vez puedan dividirse en dos ventanas sincronizadas, pero traslapantes, con los datos más importantes ubicados en la ventana superior. Crear marcos alrededor de grupos de datos en una ventana saturada, reduce la cantidad de elementos que debe percibir el ojo a primera vista.

Los artistas y fotógrafos han mostrado que el color y la composición, pueden dirigir la vista. Ningún color llama más la atención que el rojo. Es probable que una ventana con una diversidad de iconos coloreados repartidos por todos lados genere tensión visual. Se debe usar el color de forma económica y sensata, y reservar los iconos para las áreas donde realmente se necesitan pistas visuales.

La claridad visual afecta a la impresión general de la interfaz. Al reforzar la claridad visual, se promueven las relaciones lógicas entre elementos (por ejemplo, minimizando el movimiento ocular para obtener información).

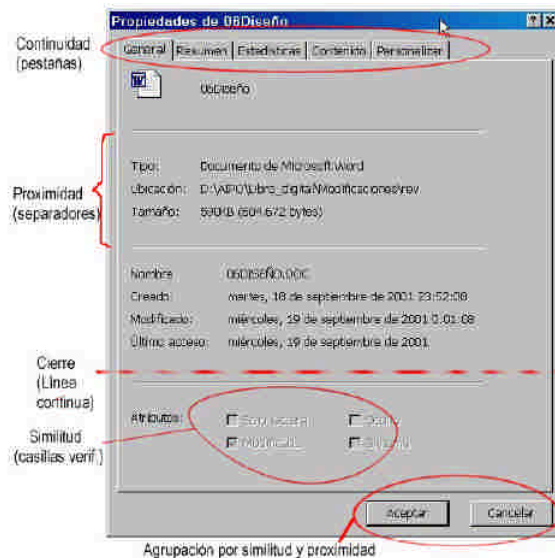


Figura 9 Aplicación de reglas a ventana de diálogo

Se puede organizar la pantalla de la interfaz siguiendo algunas reglas efectivas de diseño.

- **Balanceado.** Consiste en el ajuste de la visión con el área de visualización. El balanceado es la búsqueda de equilibrio entre los ejes horizontal y vertical en el diseño. Si se asigna un peso a cada elemento visual, se debe conseguir que la suma en cada eje sea similar. Se debe buscar un centro de gravedad en sentido horizontal y vertical, ya que de lo contrario, se crearía una *inestabilidad*.

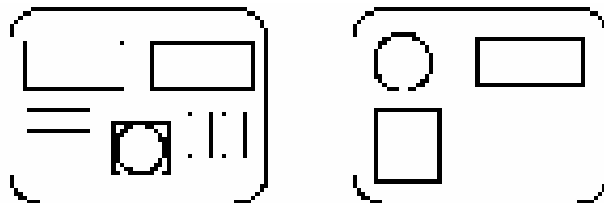


Figura 10 Pantalla balanceada (izda.) e inestable (dcha.)



- **Simetría.** Consiste en duplicar la imagen visual a lo largo de un eje de simetría. Esta técnica automáticamente asegura el balance.

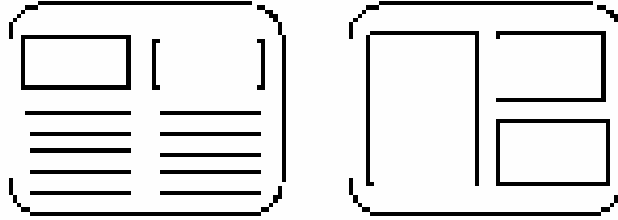


Figura 11 Pantallas con diferentes simetrías

- **Regularidad.** Técnica visual para establecer uniformidad ubicando los elementos de acuerdo con una distribución regular en filas–columnas.
- **Alineamiento.** Puntos de alineación que existen en el diseño. Se debería minimizar.
- **Enrejillado.** Separación y acentuación la organización entre áreas.

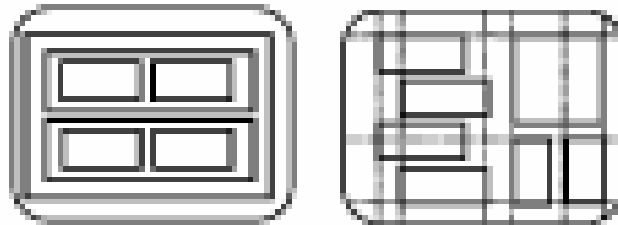


Figura 12 Pantalla con enrejillado y alineamiento (izda.)

Para obtener estas distribuciones de contenedores y componentes dentro de la pantalla, se deberán usar los controladores geométricos (*layout manager*) que facilitan las bibliotecas de diseño de interfaces (GridLayout, BorderLayout, FlowLayout, etc. en Java). Se pueden utilizar desde los más simples (posiciones absolutas, ordenación de izda. a dcha.) a los más complejos, que asignan proporciones relativas para cada elemento, tamaño de expansión, offset (desplazamiento), etc.

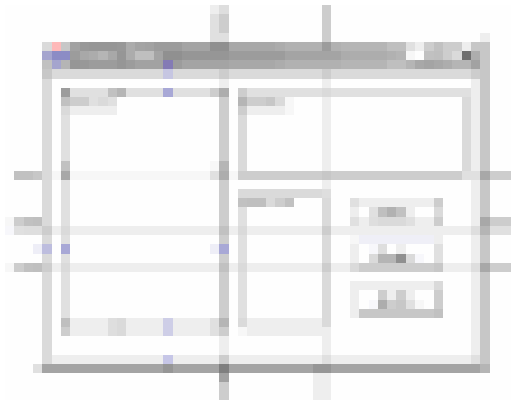


Figura 13 Controlador geométrico reticular con restricciones

El diseñador GUI necesita poner límites a la cantidad de características diferentes que coloca en una aplicación y usarlas consistentemente.



5.3.8 Indulgencia

Un buen diseño de interfaz debe motivar la exploración. El usuario debe sentirse libre para conocer la aplicación y visualizar las diversas ventanas y características. Esto es bastante fácil de lograr, pero requiere un poco más de trabajo del que están acostumbrados a realizar la mayoría de los diseñadores de aplicaciones.

El nivel de acceso del usuario está controlado, generalmente, por un conjunto de tablas en el servidor. Cuando la aplicación está ejecutando se consultan estas tablas para determinar si un usuario está autorizado para realizar diversas acciones, como realizar un pedido o crear una factura. Una forma para que se cumplan estas reglas es permitir que el usuario intente una acción y luego rechazarla o aceptarla cuando la transacción es enviada al servidor de base de datos. Aunque este esquema es fácil de programar, puede ser molesto para el usuario.

Otra aproximación es leer las tablas de autorización de usuario al momento en el que el usuario ingresa a la aplicación y usarlas para activar o desactivar conceptos en el menú principal durante su sesión. A los usuarios sólo se les da acceso a aquellas opciones donde están autorizados para trabajar. La desventaja de esto es que desmotiva la exploración de áreas que están fuera de la descripción de trabajo actual del usuario.

El esquema más liberal es permitir que todos entren a cualquier ventana, pero desactivando los botones y conceptos de menú para impedir que realicen acciones en ventanas donde no tienen autorización. Algunos usuarios nunca se apartarán de sus actividades cotidianas, pero otros explorarán el sistema completo y aprenderán mucho más acerca del negocio y las actividades que se realizan alrededor de ellos.

Siempre se debe dar al usuario una forma agradable por si decide abandonar su transacción. La siguiente es una lista de características de un sistema indulgente:

Característica	Descripción
Deshacer	Esta característica, generalmente colocada bajo el menú edición, se usa para regresar un solo campo a su valor original, siempre y cuando el usuario no haya guardado su trabajo. Es muy agradable poder deshacer a nivel campo, pero no es una característica obligatoria en las aplicaciones.
Cancelar	Se usa en ventanas de respuesta donde el sistema pregunta al usuario si se debe continuar con la acción seleccionada o abandonarla. CANCELAR se combina, por lo general, con un botón ACEPTAR o botones SI y NO. Es vital que se dé a los usuarios una oportunidad para cancelar este tipo de acciones.
Confirmar cambios en Cerrar y salir	El comando Cerrar se usa para cerrar ventanas dentro de una aplicación. El comando salir se usa para abandonar una aplicación completa. Cuando el usuario selecciona Cerrar y Salir, el sistema debe revisar si hay cambios no guardados. Si hay cambios sin guardar debe preguntar al usuario si desea guardar su trabajo antes de cerrar la ventana o salir de la aplicación.
Confirmar en Eliminar	El comando Eliminar se usa para quitar datos de la base de datos de forma permanente. Cuando el usuario selecciona Eliminar, el sistema debe preguntar "¿Esta usted seguro?" y permitir continuar o cancelar la eliminación.
Guardar explícito	Cada acción de Guardar en la base de datos debe ser explícita y obvia. Nunca se deben guardar los datos con un comando Aceptar u OK. El usuario siempre debe estar consciente de que está actualizando un registro permanente.



El Guardar explícito es una consideración importante. Muchas herramientas GUI permiten y promueven, que se realicen cambios a la base de datos cada vez que se cambia el foco de un registro. Al usuario se le permite actualizar una ventana tipo hoja de cálculo y cada vez que el cursor abandona un renglón, ese renglón se actualiza permanentemente sin notificación explícita, violando el concepto de Guardar explícito.

La técnica puede ser útil para reducir la cantidad de teclazos de una aplicación de escritorio, pero deja a la base de datos totalmente abierta a los errores.

Un método más benévolo es permitir que el usuario actualice tantos elementos de la ventana como quiera y luego hacer que guarde su trabajo explícitamente, realizando los cambios a todos los renglones de una sola vez. Esto permite cerrar la ventana sin guardar en caso de que se arrepienta en cualquier momento.

5.3.9 El factor humano

Una de las consideraciones en una interfaz son las limitaciones humanas. Es más fácil reconocer una pista visual o comando que el memorizarlo. En una GUI el usuario no requiere memorización. Cada comando o acción posible debe estar disponible ante el usuario por medio de botones, opciones de menú o icono. La microayuda en línea y el uso efectivo de la barra de estado pueden servir para ayudar al usuario a navegar por el sistema sin tener que memorizar códigos de acción.

Se debe recordar el límite humano de siete mas o menos dos, así, los menús no deben tener más de siete opciones, no deben haber más de nueve conceptos en la barra de menú, de preferencia máximo seis, en una ventana no debe haber más de cinco botones.

De forma similar, la cantidad de eventos de negocio que pueden ser reconocidos por una sola ventana también debe respetar el límite humano de siete mas o menos dos. Conforme se incrementa la cantidad de temas diferentes manejados por una ventana, disminuye la habilidad de la persona para usarla en forma efectiva. La complejidad de la ventana también es inversamente proporcional a la habilidad del programador para mantener y probar la aplicación a lo largo del tiempo.

5.3.10 Procesamiento de errores

Sobre un sistema, un factor crítico desde el punto de vista del usuario, es cómo se organizan los mensajes de error y su explicación. Normalmente ocurren por un desconocimiento por parte del usuario que puede ser motivado por diferentes causas:

- **Errores por acciones del usuario.** Error en la traslación entre la intención del usuario y su acción (intención correcta, pero realización incorrecta). La solución es mejorar el diseño ergonómico, mejorar los aspectos físicos del diseño (ubicación de menús, tamaño, posición, visibilidad, color, etc.)
- **Errores por las intenciones del usuario.** El usuario realiza una acción equivocada. El modelo de usuario no es correcto. La solución es mejorar el modelo mental. Es importante buscar posibles causas ya que el usuario está asumiendo un modelo mental incorrecto.

El usuario, ante un error, debe reconocer qué ha sucedido, para evitar confusión. Algunas técnicas que se deben evitar en los mensajes de error son:



- **Tono imperativo.** Aumenta la ansiedad del usuario. Dificulta la corrección del error ("Acción ilegal", "error fatal", "terminación anormal del programa").
- **Mensajes genéricos o confusos.** Ofrecen poca información ("error sintáctico", "run time error n. XXXX"). Cuando se agregue un cuadro de mensaje a la aplicación, se debe convertir cualquier mensaje críptico generado por la computadora en una retroalimentación agradable comprensible por el usuario.

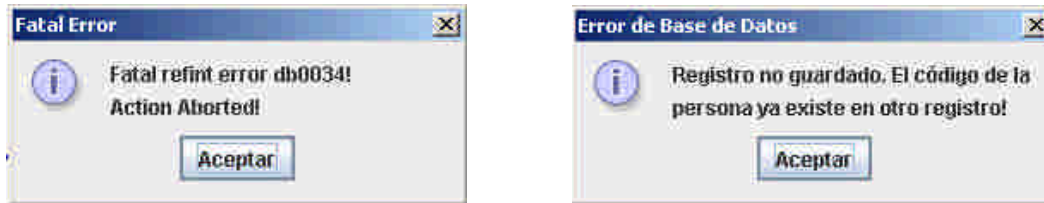


Figura 14 Mensajes de error crípticos contra comprensibles.

Un estudio de la distribución y frecuencia de errores puede ayudar al diseñador, al mantenimiento del producto y al posible usuario mediante una conveniente documentación.

5.4 Cohesión de ventanas

En el enfoque de análisis y diseño estructurado, se promovió la idea de que los módulos de un programa que ejecutan una idea altamente cohesiva tienden a ser más fáciles de comprender y más baratos de mantener a lo largo del tiempo, que los que tienen muchas ideas diferentes; también descubrieron que a mayor cohesión interna tiende a reducirse la comunicación o acoplamiento entre módulos. La alta cohesión y el bajo acoplamiento llegaron a ser la meta para los módulos bien diseñados. Esto también incrementó la probabilidad de la reutilización de código.

Estos conceptos se pueden aplicar en el ambiente orientado a objetos de una interfaz de ventanas. La cohesión de una ventana puede afectar la habilidad del usuario para comprenderla y usarla, y la habilidad del programador para mantenerla. La cohesión de ventanas puede evaluarse por la cantidad y tipo de eventos de negocios que se reconocen y manejan en una ventana o conjunto de ellas en una aplicación.

Enseguida se presentan los siete niveles de cohesión en orden del más deseable al menos deseable.

5.4.1 Cohesión funcional

Una ventana o conjunto de ellas funcionalmente cohesivas manejan un evento a nivel negocio. Suponiendo una aplicación para vender artículos de limpieza en sus locales y por medio de una campaña de correo directo. Cuando un cliente compra artículos en su tienda, el empleado realiza los siguientes eventos:

- El empleado registra/ actualiza el nombre y dirección del cliente.
- El empleado registra el pedido del cliente

El cliente puede pagar en efectivo, con cheque, o con tarjeta. Dependiendo del método de pago del cliente, los eventos finales son:



- El empleado aplica el pago de efectivo/ cheque al pedido o,
- El empleado carga el pedido a la cuenta del cliente.

Hay varias formas de diseñar un conjunto de ventanas para manejar esas transacciones. Usando el concepto de cohesión funcional se puede diseñar una ventana para cada uno de los eventos. Esto probablemente daría como resultado cuatro ventanas, una para los datos del cliente, otra para el pedido, otra para el pago en efectivo o cheque y una última para el pago con tarjeta.

La ventaja de este enfoque es que se tienen ventanas eficientes y especializadas, menos complejas y, por tanto, más fáciles de usar. El código interno ejecuta una sola idea, lo que hace que sea más fácil de comprender y más fácil de mantener. El potencial de la reutilización también se incrementa. Cada ventana puede ser reutilizada para otras aplicaciones.

La desventaja es que generalmente incrementa la cantidad de ventanas de la interfaz. Esto requiere que el usuario navegue entre varias ventanas para realizar el conjunto de tareas completas.

Se debe tener cuidado de establecer que, debido a que la definición de un evento a nivel negocios deja mucha amplitud, una ventana puede manejar varias permutaciones muy relacionadas del mismo evento de negocio y todavía ser considerada como que se apega a la cohesión funcional. Por ejemplo, en evento “el empleado registra/ actualiza el nombre y dirección del cliente” podría incluir funciones para la creación de un nuevo cliente y funciones para la actualización de uno existente. No se debe pensar que como las ventanas funcionalmente cohesivas son buenas, se deben diseñar ventanas separadas para crear, leer, actualizar y eliminar todo. Dada la tecnología esto sería excesivo.

5.4.2 Cohesión secuencial

Una ventana secuencialmente cohesiva es aquella donde los eventos están agrupados porque suceden en secuencia. El primer evento sucede como predecesor del siguiente, y ese del siguiente, y así sucesivamente. En el mismo ejemplo, agrupando los eventos en una sola ventana:

- El empleado registra/ actualiza el nombre y dirección del cliente.
- El empleado registra el pedido del cliente
- El empleado aplica el pago de efectivo/ cheque al pedido o,
- El empleado carga el pedido a la cuenta del cliente.

Primero, el empleado se asegura de que nombre y dirección del cliente estén actualizados, luego captura el pedido y después aplica el pago o el cargo.

La ventaja de la cohesión secuencial es que se mapea muy de cerca con el flujo de trabajo manual. La aplicación conserva el tecleado de navegación poniendo todos los eventos relacionados en una sola ventana. La cohesión secuencial es adecuada para las ventanas que manejan tareas altamente repetitivas donde la velocidad de captura y la economía de tecleado son los principales objetivos de la calidad.

La desventaja es que ahora es una mezcla más compleja de código no relacionado. Una parte corresponde a los datos del cliente, otra al pedido y otra a la forma de pago. El diseñador debe decidir si la acción de Guardar se aplica a todas las partes o si se requieren tres comandos Guardar.



La ventana es más compleja de usar y más costosa de mantener. Estéticamente está más amontonada, y es poco probable que cualquier parte de ella sea reutilizable sin que requiera modificaciones. La cohesión secuencial supone que la secuencia de eventos siempre es ordenada. Cuando suceden excepciones el diseñador debe proporcionar una ventana diferente o el usuario estará forzado a hacer la secuencia normal que ha sido codificada en el sistema.

5.4.3 Cohesión comunicativa

una ventana comunicativamente cohesiva es aquella en la que los elementos han sido agregados porque afectan al mismo objeto. A diferencia de la cohesión funcional o secuencial donde se puede afectar al mismo objeto, con la cohesión comunicativa el objetivo es compartir datos para agrupar los eventos.

La ventaja de tener este tipo de cohesión es que los eventos comparten el mismo código de acceso de datos y las mismas reglas de negocio. Estas ventanas también se adaptan bien al paradigma objeto-acción de seleccionar un objeto y luego desplegar una selección de acciones para varios usuarios.

El problema con esta cohesión es que las ventanas frecuentemente sirven a más de un usuario. Varios grupos de usuarios diferentes, responsables de diferentes aspectos del mismo objeto, son forzados a compartir una misma aplicación que trata de satisfacer a todas las partes. Esto da como resultado ventanas muy complejas en las que ningún usuario tiene la autoridad para hacer todo. Para cada usuario de la aplicación puede ser irrelevante un subconjunto de los campos, y algunos botones u opciones de menú están siempre desactivadas.

5.4.4 Cohesión procedural

Una ventana proceduralmente cohesiva organiza las tareas de acuerdo a la descripción de trabajo de un usuario particular. Los eventos se agregan para dar al usuario todo lo que necesita en una ventana. Por ejemplo, el coordinador de producción:

- aprueba un pedido
- programa el transporte para el pedido
- transfiere el pedido de una máquina a otra
- abre una aplicación de mensaje telefónico

Los primeros tres eventos están altamente relacionados. Si la ventana sólo manejara los primeros tres eventos, probablemente calificaría para cohesión comunicativa, ya que cada evento actualiza algún aspecto del pedido.

El cuarto evento es diferente. La única razón por la que está en la ventana es porque el coordinador de producción también hace el papel de recepcionista, y el diseñador decidió ahorrar un teclazo permitiendo que el usuario abriera la aplicación de mensaje telefónico desde la ventana que utiliza más frecuentemente.

Esta cohesión puede conducir a una cohesión muy extraña de temas no relacionados. Estos da como resultado ventanas complejas, saturadas, lentas en responder y extremadamente sensibles a los cambios organizacionales.



Una mejor forma de lograr “mantener todo en un lugar” para los usuarios, es poner juntas las ventanas en un grupo funcional, secuencial o hasta comunicativamente cohesivo en un menú personalizado. Bajo este escenario, las ventanas individuales serán menos complejas, más fáciles de reutilizar y tendrán mayores oportunidades de resistir los cambios en la descripción del puesto del usuario. Es mucho más fácil cambiar la estructura de un menú que dividir ventanas individuales.

5.4.5 Cohesión temporal

La cohesión temporal sucede cuando los eventos están agrupados en una ventana sólo porque suceden al mismo tiempo. Por ejemplo:

- Se registra en el sistema
- Revisa el correo electrónico
- Imprime el reporte de ventas del día anterior

El desarrollador pone estas actividades en una sola ventana porque son “cosas que el usuario hace cuando llega en la mañana”. La relación de estas tareas es muy débil.

La cohesión temporal es rara. Por lo general aparece en sistemas sólo para iniciar actividades de contabilidad de cierre de mes y están relacionadas sólo porque se necesita que sucedan en un momento específico. Estas ventanas también son casi imposibles de reutilizar y existen para la conveniencia de un pequeño grupo de usuarios en un momento particular.

5.4.6 Cohesión lógica

La cohesión lógica sucede cuando los eventos están agrupados para compartir el mismo código. Por ejemplo en una ventana con criterios de selección para un reporte de propósito general. Esta ventana puede contener casi todos los campos que podría llegar a necesitar un usuario para dividir y repartir los datos para diversos reportes. El usuario indica primero el reporte que desea entre una gran lista de reportes disponibles. Los criterios de selección le permiten estrechar el conjunto de resultados antes de iniciar el trabajo de reporte. El propósito de la ventana es evitar la proliferación de ventanas de selección de reportes similares por toda la aplicación, y se pretende que reduzca el efecto de propagación de onda de los cambios que puedan afectar varias ventanas.

El problema es este ejemplo es que probablemente, no todos los campos de criterios de selección se aplican a todos los reportes. Aquellos que son irrelevantes son simplemente descartados por el programa. Si el diseñador encuentra que esto es inaceptable, la aplicación tendrá que manejarlos campos que aplican activando y desactivando las capacidades de actualización para cada reporte que use la ventana de selección.

La cohesión lógica puede crear comportamientos raros en la interfaz y problemas de mantenimiento. El programador de mantenimiento deberá evaluar constantemente el efecto de cualquier cambio solicitado contra cada uno de los eventos que se manejan por la ventana para asegurarse que el cambio no genere resultados indeseables debido al código compartido. En muchos casos, esta complejidad adicional se contrapone con el benéfico pretendido de la reducción del efecto de propagación de onda. Frecuentemente es más fácil hacer cambios claros y simples en más de un lugar que desenmarañar y volver a probar una maraña de código complejo.

5.4.7 Cohesión coincidental

Mientras que las demás medidas de cohesión al menos tienen un objetivo de diseño válido, la cohesión



coincidental es simplemente caprichosa. Cualquier relación entre eventos en una ventana coincidentalmente cohesiva existe sólo en la mente del programador que la creó.

Aunque esta cohesión es afortunadamente rara en los sistemas de negocios, no es difícil encontrarla en Internet donde se ponen enlaces a todo tipo de cosas que no tienen nada que ver, ventas, clima, horóscopos, curriculums, etc.

5.4.8 Valoración de los niveles de cohesión

Cualquier aplicación grande tiene una mezcla saludable de diversos niveles de cohesión entre las diversas ventanas. La mayoría de las ventanas en una aplicación empresarial bien diseñada tendrán alguno de los tres primeros niveles de cohesión. La cohesión funcional produce las mejores ventanas reutilizables, comprensibles y mantenibles de todas, pero genera muchas ventanas si se usa exclusivamente. La cohesión secuencial es un método excelente para diseñar ventanas donde el usuario ejecuta una serie de tareas regularmente. La cohesión comunicacional hace un buen trabajo al mantener el acceso a los datos y las reglas del negocio visualmente evidentes en un solo lugar, pero incrementa la complejidad del manejo de la activación y desactivación de la interfaz.

Las ventanas con cohesión procedural generan el riesgo adicional de ser inflexibles ante los cambios de la descripción del puesto del usuario. Esto no hace que lo procedural sea malo por sí mismo. El desarrollador puede estar bajo muy fuertes presiones de los usuarios para usar cohesión procedural. Antes de agregar eventos muy poco relacionados con base en la descripción del puesto, se deben considerar cuidadosamente las consecuencias tanto para el usuario como para el programador de mantenimiento de la complejidad resultante.

Los dos siguientes niveles de cohesión tienen utilidad limitada. La cohesión temporal es adecuada para ventanas que inician rutinas de cierre de mes o tareas similares relacionadas con el cliente. Para otras áreas del sistema debe evitarse. Aunque la cohesión lógica se inicia con la mejor intención de compartir código, por lo general resulta en un comportamiento problemático de la interfaz y termina siendo más molesto que lo que vale. Una ventana lógicamente cohesiva probablemente deberá ser rediseñada en varias ventanas, donde cada una tenga cohesión de mayor nivel.

El último nivel de cohesión coincidental, no tiene cabida en una aplicación seria.



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

Licenciatura En Informática

Programación de Interfaces

*Autor: L.I. María de Lourdes Isabel
Ponce Vásquez*

AGOSTO-DICIEMBRE 2005



Contenido

Contenido	2
UNIDAD 6. ESTÁNDARES Y GUIAS	3
Objetivos:	3
Introducción.....	3
6.1 Principios y directrices	4
6.1.1 Principios	4
6.1.2 Directrices	7
6.2 Estándares	8
6.2.1 Estándar de iure	9
6.2.2 Estándares de iure en PI.....	12
6.2.2.1 ISO/IEC 9126	14
6.2.2.2 ISO 9241	14
6.2.2.3 ISO/IEC 10741	15
6.2.2.4 ISO/IEC 11581	15
6.2.2.5 ISO 11064	16
6.2.2.6 ISO 13406	16
6.2.2.7 ISO/IEC 14754	16
6.2.2.8 ISO/IEC 15910	17
6.2.2.9 ISO 13407	17
6.2.2.10 ISO/IEC 14598	17
6.2.2.11 ISO TR 18529.....	18
6.2.2.12 ISO 10075	18
6.2.3 Estándar de facto	18
6.4 Guías de estilo	18
6.4.1 Guías de estilo comerciales	19
6.4.1.1 Apple	19
6.4.1.2 CUA.....	19
6.4.1.3 Microsoft.....	20
6.4.1.4 Common Desktop Environment (CDE)	20
6.4.1.5 Motif	20
6.4.1.6 Open Look.....	21
6.4.1.7 Java Look and Feel	21
6.4.2 Guías de estilo para la Web	22
6.4.2.1 Apple	22
6.4.2.2 IBM.....	22
6.4.2.3 Sun.....	23
6.4.2.4 W3C	23
6.4.2.5 Yale Center for Advanced Instructional Media	24
6.4.2.9 National Cancer Institute (NIC)	24
6.4.3 Guías de estilo corporativas	24
6.5 Consideraciones sobre los estándares y las guías de estilo	25



UNIDAD 6. ESTÁNDARES Y GUÍAS

Objetivos:

- Conocer la diferencia entre principios y directrices
- Conocer los diferentes estándares existentes relacionados con PI
- Conocer las diferentes guías de estilo existentes
- Ver la necesidad y utilidad de los estándares y las guías de estilo

Introducción

Antiguamente el software era diseñado con poca consideración hacia el usuario: el usuario debía adaptarse al sistema. Sin embargo, esto hoy en día no puede ocurrir. Está claro que la interfaz de un software bien diseñado debería establecer, un enlace o relación con los usuarios, guiándoles en el aprendizaje y haciéndoles que disfruten de lo que están haciendo.

Evidentemente, para conseguir esto es necesario un buen entendimiento del modelo mental del usuario, así como de sus habilidades psíquicas, físicas y psicológicas. Sin embargo, los diseñadores generalmente no son expertos en estos temas. Es por eso que esta información, consensuada por la mayoría de los expertos en la materia, se ha plasmado en unos principios generales de diseño de interfaces de usuario.

Estos principios son conceptos de muy alto nivel que deberían ser empleados en el diseño del software. Sin embargo, para ciertos productos y situaciones unos principios pueden entrar en conflicto con otros. Son necesarias entonces unas reglas de diseño que guíen al diseñador con el fin de incrementar la usabilidad del producto a diseñar.

Estas reglas de diseño pueden clasificarse en estándares y directrices, dónde de una forma aproximada se puede decir que, los estándares son altos en autoridad y limitados en aplicación, mientras que las directrices son más bajas en autoridad y más generales en aplicación.

La mayoría de los sistemas de interfaces gráficas de usuario han publicado directrices que describen cómo asociar estos principios abstractos a entornos de programación específicos. Estas directrices reciben el nombre de guías de estilo y reflejan que no son reglas estrictas, sino convenciones sugeridas para programar en dicho entorno.

Las guías de estilo permiten a los diseñadores tener marcos generales de diseño que les pueden ayudar a tomar decisiones correctas en sus diseños. Estas guías pueden adoptar una gran variedad de formas y se pueden obtener en diferentes sitios, como por ejemplo en artículos de revistas académicas, profesionales o comerciales que dan una buena referencia del estado actual en cuanto a práctica y experiencia.

Una empresa con el fin de mantener su imagen corporativa puede disponer también de una guía de estilo propia que recibe el nombre de guía de estilo corporativa.



6.1 Principios y directrices

6.1.1 Principios

Un principio es una sentencia en un sentido muy amplio que normalmente está basada en la investigación hecha de cómo las personas aprenden y trabajan.

Estos principios están basados en ideas de alto nivel y de aplicación muy general. Son bastante abstractos. Un ejemplo de un principio relacionado, por ejemplo, con la ayuda a los usuarios podría ser:

Asistencia: ayudar al usuario en la realización de las diferentes tareas.

A continuación se describen principios expuestos por diferentes autores, y que pueden dar una idea de lo que opina la comunidad de PI en este aspecto:

SIMPSON:

- Definir los usuarios
- Dejar el control a los usuarios
- Minimizar el trabajo de los usuarios
- Hacer programas sencillos
- Mantener la consistencia
- Proporcionar realimentación
- No cargar la memoria de trabajo
- No abusar de la memoria a largo plazo

PERECE:

- Estudiar la población de usuarios. Éste es un aspecto difícil de conseguir teniendo en cuenta los diferentes tipos de usuario que se pueden presentar. Esto lleva, por ejemplo, a la conveniencia de emplear atajos (shortcuts) normalizados para usuarios experimentados, de forma que dispongan de más de una opción para desarrollar una tarea.
- Reducir la carga cognitiva. Aspecto relacionado con el hecho de que el usuario no tenga que aprender una gran cantidad de detalles.
- Aplicar ingeniería para resolver la problemática del error humano. Una excusa que normalmente se utiliza es que los errores surgen debido a errores humanos. No obstante, las personas siempre cometen errores de los cuales posteriormente aprenden.
- Mantener consistencia. La consistencia emerge de operaciones y representaciones estándar, así como por el empleo de las metáforas apropiadas.

MANDEL:

- Colocar a los usuarios en el control de la interfaz. Esto se traduce en una serie de principios más concretos, entre los que se encuentran:



- Emplear los modos adecuadamente.
 - Permitir a los usuarios emplear el teclado y el ratón
 - Permitir a los usuarios cambiar la atención
 - Mostrar mensajes y texto descriptivos
 - Proporcionar acciones inmediatas, reversibles y realimentación
 - Acomodar a los usuarios con diferentes niveles de habilidad
 - Hacer la interfaz de usuario transparente
 - Permitir al usuario personalizar la interfaz
 - Permitir a los usuarios manipular los objetos de la interfaz
- Reducir la carga de memoria de los usuarios. La memoria humana tiene una serie de capacidades y limitaciones que deben considerarse cuando se están diseñando las interfaces. Existen una serie de principios que reducen la carga de memoria del usuario:
- Aliviar la memoria a corto plazo
 - Confiar en el reconocimiento
 - Proporcionar pistas visuales
 - Proporcionar opciones por omisión
 - Proporcionar atajos
 - Promover la sintaxis objeto–acción
 - Emplear metáforas del mundo real
 - Emplear la revelación progresiva para evitar abrumar al usuario.
 - Promover la claridad visual
- Hacer la interfaz consistente. Lo que permitirá que los usuarios puedan transferir su conocimiento y aprendizaje a un nuevo programa siempre que éste sea consistente con otros programas que ya haya usado. Los principios que ayudan a lograr una interfaz consistente son:
- Preservar el contexto de trabajo de los usuarios
 - Mantener la consistencia dentro y entre productos
 - Conservar los resultados para las mismas interacciones
 - Animar la exploración de la interfaz

DIX:

- Facilidad de Aprendizaje. De forma que el usuario novel comprenda cómo utilizar inicialmente un sistema interactivo, y cómo a partir de esta utilización puede llegar a un máximo nivel de conocimiento y rendimiento del sistema. Los principios que colaboran para conseguir este objetivo son:
- Capacidad de predicción
 - Capacidad de síntesis
 - Familiaridad
 - Generalización
 - Consistencia



- Flexibilidad. Haciendo referencia a la multiplicidad de formas en las que el usuario y el sistema intercambian información. Para esto se identifican varios principios:
 - Iniciativa en el diálogo
 - Multi-hilo
 - Migración de tareas
 - Capacidad de sustitución
 - Configurabilidad
- Robustez. Que hace referencia a las características que permiten cumplir los objetivos y su evaluación. Los principios para conseguir dicha solidez son:
 - Capacidad de observación
 - Capacidad de recuperación
 - Tiempos de respuesta aceptables
 - Adecuación de las tareas

SCHNEIDERMAN:

- Reconocer la diversidad. Antes de comenzar un diseño se debe realizar la caracterización de los usuarios y de las situaciones, de forma tan precisa y completa como sea posible.
- Prevenir los errores antes de que ocurran.
- Emplear, entre otras, las siguientes reglas en el diseño de las interfaces:
 - Consistencia
 - Permitir atajos a los usuarios experimentados
 - Proporcionar realimentación informativa
 - Ofrecer prevención de errores y una administración de errores sencilla
 - Permitir que se puedan deshacer acciones
 - Reducir la carga cognitiva de la memoria a corto plazo

IBM:

- Simplicidad. No sacrificar la usabilidad del programa por la funcionalidad del mismo.
- Apoyo. Hay que proporcionar el control al usuario sobre el sistema y suministrarle asistencia para facilitar la realización de las tareas.
- Familiaridad. Construir el producto según el conocimiento previo del usuario, lo que le permitirá progresar rápidamente.
- Evidencia. Hacer los objetos y sus controles visibles e intuitivos. Emplear siempre que se pueda representaciones del mundo real en la interfaz.
- Estímulo. Hacer las acciones previsibles y reversibles. Las acciones de los usuarios deberían producir los resultados que ellos esperan.
- Satisfacción. Crear una sensación de progreso y logro en el usuario.
- Disponibilidad. Hacer todos los objetos disponibles de forma que el usuario pueda usar todos sus objetos en cualquier secuencia y en cualquier momento.
- Seguridad. Evitarle errores al usuario proporcionándole diferentes tipos de ayuda bien de forma automática o bien a petición del propio usuario.



- Versatilidad. Soportar diversas técnicas de interacción, de forma que el usuario pueda seleccionar el método de interacción más apropiado para su situación.
- Personalización. Permitiendo a los usuarios adaptar la interfaz a sus necesidades.
- Afinidad. Permitir que a través de un buen diseño visual que los objetos sean afines a otros de la realidad cotidiana. Dicho principio incluye en cierto modo los siguientes:
 - Diseño substractivo: eliminar cualquier elemento que no ayude a una comunicación visual.
 - Herencia Visual: que establece una jerarquía visual de las tareas de usuario por orden de importancia. Emplea la posición relativa y el contraste en el color y tamaño para incrementar la prominencia visual de un objeto.
 - Similitud: los usuarios deben determinar de una forma intuitiva el uso de un objeto al relacionarlo con otro del mundo real.
 - Esquema Visual: se debe diseñar un esquema visual que se asemeje al modelo del usuario.

6.1.2 Directrices

Las directrices recomiendan acciones basándose en un conjunto de principios de diseño. Generalmente son más específicas que los principios y requieren menos experiencia para entenderlas e interpretarlas que éstos.

Un ejemplo de una directriz relacionada con el principio de ayuda al usuario mencionado anteriormente podría ser:

“Proporcionar ayuda contextual para cada opción y objeto sobre el que pueda posicionarse el cursor”.

Cada principio en general es un objetivo, pero no dice como conseguirlo. Las directrices son objetivos más específicos que los especialistas en PI concretan a partir de los principios para usuarios, entornos y tecnologías diferentes.

Las directrices permiten asegurar consistencia a través de las diferentes partes de un sistema, o a través de una familia de sistemas. Por este motivo es tan importante para las organizaciones que desarrollan software, disponer de unas directrices que puedan seguir sus desarrolladores.

Brown en su libro Human Computer Interface Guidelines resume los objetivos de las directrices e indica que proporcionan una aproximación sistemática a:

- Aprovechar la experiencia práctica
- Difundir e incorporar experiencia experimental aplicable
- Incorporar reglas de sentido común
- Promover consistencia entre los diseñadores responsables de partes diferentes de la interfaz
- Como a veces pueden provocar conflictos siempre es importante aplicar test de usabilidad para tratar de resolverlos.



6.2 Estándares

Un estándar es un requisito, regla o recomendación basada en principios probados y en la práctica. Representa un acuerdo de un grupo de profesionales oficialmente autorizados a nivel local, nacional o internacional.

Los estándares pueden ser, por tanto:

- **Locales:** diseño o práctica aceptada desde una industria, organización profesional o entidad empresarial.
- **Nacionales:** convención aceptada por una amplia variedad de organizaciones dentro de una nación.
- **Internacionales:** consenso entre organizaciones de estándares a nivel mundial.

El objetivo de los estándares es hacer las cosas más fáciles, definiendo características de objetos y sistemas que se utilizan cotidianamente. El diseño de un teclado de teléfono es un estándar que se utiliza continuamente. El teclado QWERTY es otro estándar importante que permite interactuar con cualquier computadora.

Hay estándares en todas las industrias, así por ejemplo, en la industria de la construcción, los estándares permiten al arquitecto y al constructor comunicarse con todo el mundo que participa en la construcción de una obra. Pueden transferir sus conocimientos porque todo el mundo reconoce los estándares involucrados en el proceso de construcción. Hay estándares mecánicos, eléctricos, etc. que facilitan la realización de las tareas.

La industria informática también tiene estándares, y esos estándares pueden aplicarse tanto al hardware como al software.

Existen definiciones de estándares de pantallas, teclados, unidades centrales y hasta de mobiliario. Por ejemplo, si se necesita un cable para conectar la computadora a la impresora seguramente será un cable paralelo estándar con los conectores estándar. La mayoría de los estándares de hardware reflejan la importancia de la ergonomía en las interfaces de usuario. Por ejemplo, un estándar hardware internacional enuncia: "la inclinación del teclado debería estar entre 0 y 25 grados".

Los estándares de software se aplican generalmente a características básicas de la interfaz de usuario. Con el hecho de desarrollar estándares para la interfaz se intenta conseguir un software más fácil y seguro, estableciendo unos requisitos mínimos de fabricación y eliminando inconsistencias y variaciones innecesarias en las interfaces. Se pueden entender los estándares como una manera de asegurar que los factores humanos de calidad estarán incorporados en el sistema.

Se podrían resumir los beneficios que supone la utilización de estándares diciendo lo que éstos favorecen:

- Una terminología común. Esto permite que los diseñadores sepan que están discutiendo los mismos conceptos, con lo que se pueden hacer valoraciones comparativas.



- El mantenimiento y la evolución. Porque todos los programas tienen la misma estructura y el mismo estilo.
- Una identidad común. Lo que hace que todos los sistemas sean fáciles de reconocer.
- Reducción en la formación. Los conocimientos son más fáciles de transmitir de un sistema a otro si por ejemplo, las teclas de comandos están estandarizadas.
- Salud y seguridad. Si los sistemas han pasado controles de estandarización es difícil que tengan comportamientos inesperados.

Hay dos tipos de estándares: estándares de iure y estándares de facto, que se examinan a continuación con más detalle.

6.2.1 Estándar de iure

Los estándares de iure son generados por un comité con estatus legal y están avalados por el apoyo de un gobierno o institución para producir estándares. Para hacer un estándar de iure se debe seguir un proceso complejo. Primeramente, se confecciona un documento preliminar que se debe hacer público, después cualquier persona o empresa puede presentar enmiendas a los borradores del documento. Estas enmiendas deben ser comentadas y resueltas. Después de un cierto tiempo, a veces años, se consigue un consenso y se acepta el nuevo estándar.

En informática existen una serie de comités que han participado en la creación de muchos estándares de iure, como por ejemplo: ANSI, API, ASME, ASQ, ASTM, AWS, BSI, CSA, DIN, EIA, FORD, GM, ICEA, IEC, IEEE, IPC, ISA, ISO, JSA, NEMA, SAE, TIA, UL... De todos estos se destacan los más importantes:

1. ISO: Organización Internacional para Estándares



URL: <http://www.iso.ch/iso/en/ISOOnline.opennerpage>

La Organización Internacional para Estándares (International Organization for Standardization, ISO) con sede en Ginebra, es una federación mundial de cuerpos de estándares nacionales de más de 130 países.

Fundada en 1947, ISO es una organización no gubernamental cuya misión es promover el desarrollo de la estandarización y actividades relacionadas a nivel mundial, con la intención de facilitar el intercambio internacional de bienes y servicios, y para desarrollar la cooperación en el ámbito económico, científico, tecnológico e intelectual. El ámbito de un estándar ISO incluye todos los campos excepto básicamente la ingeniería eléctrica (responsabilidad de IEC).



El trabajo de ISO acaba en acuerdos internacionales que son publicados como estándares internacionales.

2. IEC: Comisión Electrotécnica Internacional



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

URL: <http://www.iec.ch>

La Comisión Electrotécnica Internacional (International Electrotechnical Commission, IEC) fue fundada en 1906, y es una organización no gubernamental compuesta por comités en más de 40 países. Su trabajo es realizado por 88 comités técnicos, más de 100 subcomités y varios cientos de grupos de trabajo, donde cada grupo es responsable del desarrollo de estándares para un sector específico de la tecnología. Su misión es preparar y publicar estándares internacionales de temas relacionados con la ingeniería eléctrica y electrónica, y tecnologías relacionadas.

IEC coopera estrechamente con ISO. La coordinación de ambas asociaciones es responsabilidad del Comité Técnico de Conexión (Joint Technical Committee, JTC). El JTC para los estándares en el campo de la tecnología de la información es el número 1.

3. ANSI: Instituto Nacional Americano para Estándares



URL: <http://www.ansi.org>

El Instituto Nacional Americano para Estándares (American National Standards Institute, ANSI) se dedica desde hace más de 80 años a administrar y coordinar la estandarización voluntaria del sector privado de la industria de los Estados Unidos. Creado el 18 de Octubre de 1918 por cinco sociedades de ingeniería y tres agencias gubernamentales, sigue su papel hoy en día como un Instituto de carácter privado sin ánimo de lucro, dedicado a la representación de los intereses de casi 1.000 compañías, organizaciones y agencias del gobierno. Su principal misión consiste en ampliar la competitividad de las empresas de los Estados Unidos a través de la generación de estándares.

4. IEEE: Instituto de Ingenieros Eléctricos y Electrónicos Americano. Asociación para Estándares



URL: <http://standards.ieee.org>



El Instituto de Ingenieros Eléctricos y Electrónicos Americano (Institute of Electrical and Electronics Engineers, IEEE) es una asociación profesional técnica, sin ánimo de lucro, con socios en más de 150 países. La actividad de IEEE en relación con los estándares es desarrollar y publicar estándares generalmente aceptados, que promoverán la teoría y la práctica de la ingeniería eléctrica, electrónica e informática, así como del resto de ramas de la ingeniería o artes y ciencias relacionadas. Trabaja conjuntamente con otros cuerpos de estandarización, nacionales e internacionales, con el fin de difundir estándares en el campo de la electrotecnología.

Dentro de IEEE existe una Asociación para Estándares (IEEE Standards Association) que es la responsable de gestionar los estándares. Dicha asociación tiene dos órganos de gobierno: el consejo directivo y el consejo de estándares. Éste último es el encargado de fomentar y coordinar el desarrollo y la revisión de estándares, así como la aprobación de proyectos.

5. CEN: Comité Europeo para la Estandarización



URL: <http://www.cenorm.be>

El Comité Europeo para la Estandarización (Comité Européen de Normalisation, CEN) es el principal proveedor de estándares europeos y especificaciones técnicas. Es la única organización europea reconocida para planear y adoptar estándares europeos en todas las áreas de la actividad económica, con la excepción de la electrotecnología (European Committee for Electrotechnical Standardization, CENELEC) y las telecomunicaciones (European Telecommunications Standards Institute, ETSI). Su misión es promover la coordinación técnica voluntaria en Europa en conjunción con cuerpos internacionales y sus socios en Europa.

6. W3C: Consorcio para World Wide Web



URL: <http://www.w3.org>

El Consorcio para World Wide Web (World Wide Web Consortium, W3C) fue creado en 1994 por Tim Berners-Lee (inventor de la Web) en el Instituto Tecnológico de Massachusetts (MIT, EE.UU.), y con la colaboración del Instituto Nacional de Investigación en Informática y Automática (INRIA, Francia) y la Universidad Keio de Japón. Su objetivo es llevar el World Wide Web a su pleno potencial, desarrollando protocolos comunes que promuevan su evolución y aseguren su interoperabilidad.

W3C está constituido por más de 500 organizaciones alrededor del mundo, y contribuye a estandarizar las tecnologías Web produciendo especificaciones (llamadas 'recomendacio-



nes') que describen los bloques de construcción de la Web. El consorcio tiene grupos de trabajo que crean recomendaciones en áreas relacionadas con las interfaces de usuario tales como la accesibilidad, la internacionalización, etc. Estas recomendaciones y otros informes técnicos están disponibles de forma totalmente gratuita.

6.2.2 Estándares de iure en PI

La publicación de estándares de la interfaz es un tema relativamente reciente, aunque es una preocupación desde hace muchos años. En este tema probablemente la organización para hacer estándares internacionales ISO, ha sido la más activa.

En este apartado se desea mencionar la existencia de estándares de iure que están relacionados con el diseño de sistemas interactivos. De estos estándares algunos están ya completamente publicados, mientras que otros están en proceso de elaboración, ya que como se mencionó con anterioridad, la creación de un estándar es un proceso complejo y largo en el que los documentos van pasando por diferentes estados. En la siguiente tabla, y a modo de ejemplo, se reflejan los diferentes estados implicados en el proceso de desarrollo de un estándar ISO.

Estado	Tipo documento	Documento	Descripción
1	AWI	Approved Work Item Elemento de trabajo aprobado	Previo a un borrador de trabajo
2	WD	Working Draft Borrador de trabajo	Borrador preliminar para su debate por un grupo de trabajo
3	CD	Committee Draft Borrador de comité	Borrador completo para votación y comentarios técnicos por cuerpos nacionales
	CD TR o TS	Committee Draft, Technical Report/Specification Borrador de comité de un informe o especificación técnica	
4	CDV	Committee Draft for Vote Borrador de comité para votación (IEC)	Borrador final para votación y comentarios editoriales por cuerpos nacionales
	DIS	Draft International Standard Borrador de un estándar internacional	
	FCD	Final Committee Draft Borrador final del comité (JTC1)	



	DTR o DTS	Draft Technical Report/Specification Borrador de una especificación o informe técnico	
5	FDIS	Final Draft International Standard Borrador final de un estándar Internacional	Texto para la publicación por aprobación fina
6	ISO	International Standard Estándar internacional	Documento publicado
	ISO TR o TS	Technical Report or Specification Especificación o informe técnico	

Tabla 1 Estados implicados en el proceso de desarrollo de un estándar ISO

A continuación se muestra una posible clasificación de los estándares relacionados con PI, para pasar posteriormente a describir brevemente algunos de los más relevantes y que ya están completamente publicados, o bien tienen alguna de sus partes ya publicadas.

Conviene destacar que el proceso de confección de estándares está en continua evolución con lo que parte de la información aquí presentada está sujeta a modificaciones constantes.

Tipo	Principios y recomendaciones	Especificaciones
Contexto	ISO/IEC 9126–1	ISO AWI 20282 Usabilidad de productos cotidianos
	ISO/IEC TR 9126–4	
	ISO 9241–11	
Interfaz e Interacción	ISO/IEC TR 9126–2	ISO 9241. Partes 3–9
	ISO/IEC TR 9126–3	ISO/IEC 10741–1
	ISO 9241. Partes 10–17	ISO/IEC 11581
	ISO 11064	ISO 13406
	ISO 14915 Ergonomía del software para interfaces de usuario multimedia	ISO/IEC 14754
	IEC TR 61997 Directrices para las interfaces de usuario en equipos multimedia para usos de propósito general	ISO/IEC FDIS 18021 Tecnología de la información – Interfaces de usuario para herramientas Móviles
		ISO AWI 18789 Requisitos ergonómicos y técnicas de medida para elementos visuales electrónicos



Documentación	ISO/IEC WD 18019 Directrices para el diseño y preparación de la documentación	ISO/IEC 15910
Proceso de Desarrollo	ISO 13407	ISO/IEC 14598
	ISO CD TR 16982 Métodos de usabilidad que soportan diseño centrado en el usuario	
Capacidad	ISO TR 18529	
Otros	ISO 9241-1	
	ISO 9241-2	
	ISO 10075	
	ISO WD TR 16071 Guía sobre accesibilidad para interfaces persona-computadora	

Tabla 2 Posible clasificación de los estándares relacionados con PI

6.2.2.1 ISO/IEC 9126

Este estándar define la usabilidad como una contribución relativamente independiente a la calidad del software, asociada con el diseño y la evaluación de la interfaz de usuario y la interacción. Define métricas para la usabilidad y la calidad de uso.

ISO/IEC 9126 Evaluación de productos software – características de calidad y directrices para su uso		
		Estado
Parte 1	Modelo de calidad	ISO
Parte 2	Métricas externas	CD
Parte 3	Métricas internas	CD
Parte 4	Métricas para la calidad de uso	CD

6.2.2.2 ISO 9241

ISO 9241 es un estándar de iure relacionado con los requisitos ergonómicos para trabajar con terminales de presentación visual (Visual Display Terminals, VDT). Las tareas de la oficina incluyendo procesamiento de textos y datos son cubiertas por este estándar.

Aunque este estándar se centró originalmente en los requisitos hardware para los VDT, pronto evolucionó para incluir requisitos de software. Está compuesto de 17 partes que tratan cuatro áreas principales:

- Introducción y descripción general (partes 1–2)
- Requisitos hardware (partes 3–9):
- Requisitos del entorno (parte 6)
- Requisitos software (partes 10–17)



ISO 9241 Requisitos ergonómicos para trabajar con terminales de presentación visual (VDTs)		
		Estado
Parte 1	Introducción general	ISO
Parte 2	Orientación sobre los requisitos de las tareas	ISO
Parte 3	Requisitos de la presentación visual	ISO
Parte 4	Requisitos de teclado	ISO
Parte 5	Diseño de estaciones de trabajo y requisitos de las posturas	ISO
Parte 6	Orientación sobre el entorno de trabajo	ISO
Parte 7	Requisitos para la visualización con reflejos	ISO
Parte 8	Requisitos para colores visualizados	ISO
Parte 9	Requisitos para dispositivos de entrada no-teclado	ISO
Parte 10	Principios de diálogos	ISO
Parte 11	Orientación sobre usabilidad	ISO
Parte 12	Presentación de información	ISO
Parte 13	Orientación del usuario	ISO
Parte 14	Diálogos de menús	ISO
Parte 15	Diálogos de comandos	ISO
Parte 16	Diálogos de manipulación directa	ISO
Parte 17	Diálogos para completar formularios	ISO

6.2.2.3 ISO/IEC 10741

Este estándar define cómo deben iniciar, controlar y monitorizar los usuarios las posibilidades del sistema para controlar los cursores en sistemas de texto por medio de funciones de control.

Inicialmente tenía dos partes, y finalmente la parte 2 fue cancelada permaneciendo como estándar internacional desde 1995 la parte 1, que especifica cómo debería moverse el cursor sobre la pantalla en respuesta al uso de las teclas de los cursores.

ISO/IEC 10741 Interacción de diálogos		
		Estado
Parte 1	Control del cursor para edición de textos	ISO
Parte 2	Control del cursor para hojas de cálculo	Cancelada

6.2.2.4 ISO/IEC 11581

Este estándar internacional comprende seis partes que se aplican a los iconos que son visualizados en la pantalla de la computadora. Estos iconos representan datos o funciones del sistema con los que los usuarios pueden interactuar o manipular.



ISO/IEC 11581		
Símbolos y funciones de los iconos		
		Estado
Parte 1	Iconos-general	ISO
Parte 2	Iconos de objetos	ISO
Parte 3	Iconos de punteros	ISO
Parte 4	Iconos de controles	CDO
Parte 5	Iconos de herramientas	FCD
Parte 6	Iconos de acciones	ISO

6.2.2.5 ISO 11064

Este estándar está dividido en ocho partes que contienen principios ergonómicos, recomendaciones y directrices para el diseño de los centros de control.

ISO 11064		
Diseño ergonómico de centros de control		
		Estado
Parte 1	Principios para el diseño de centros de control	ISO
Parte 2	Principios para la organización del control	ISO
Parte 3	Disposición del sitio de control	ISO
Parte 4	Disposición y dimensiones de las estaciones de trabajo	CD
Parte 5	Interfaces persona – sistema	WD
Parte 6	Requisitos del entorno para los sitios de control	CD
Parte 7	Principios para la evaluación de los centros de control	WD
Parte 8	Requisitos ergonómicos para aplicaciones específicas	WD

6.2.2.6 ISO 13406

Este estándar establece los requisitos ergonómicos de la calidad de la imagen para el diseño y evaluación de presentaciones visuales basadas en paneles planos. También especifica métodos para determinar la calidad de la imagen.

ISO 13406		
Requisitos ergonómicos para trabajar con presentaciones visuales basadas en paneles planos		
		Estado
Parte 1	Introducción	ISO
Parte 2	Requisitos ergonómicos para presentaciones visuales en paneles planos	ISO

6.2.2.7 ISO/IEC 14754

Este estándar comprende una única parte que define un conjunto de comandos de acciones básicas y reacciones para interfaces basadas en lápiz. Las acciones que incluye son: seleccionar, borrar, insertar espacio, saltar línea, mover, copiar, cortar, pegar, desplazar y deshacer.



ISO/IEC 14754		
Gestos comunes para la edición de textos con sistemas basados en lápiz		
		Estado
	Gestos comunes para la edición de textos con sistemas basados en lápiz	ISO

6.2.2.8 ISO/IEC 15910

Este estándar especifica el proceso mínimo para crear documentación para el software que tiene una interfaz de usuario. Incluye documentación impresa (por ejemplo, manuales de usuario y de referencia rápida) y documentación en línea.

ISO/IEC 15910		
Proceso de documentación de software de usuario		
		Estado
	Proceso de documentación de software de usuario	ISO

6.2.2.9 ISO 13407

Este estándar proporciona una orientación sobre las actividades de diseño centradas en la persona a lo largo del ciclo de vida de sistemas interactivos basados en computadoras. Describe el diseño centrado en el usuario como una actividad multidisciplinar que incorpora factores humanos, y técnicas y conocimientos ergonómicos, con el fin de conseguir efectividad y eficiencia, y mejorar las condiciones de trabajo para las personas.

ISO 13407 Procesos de diseño centrados en la persona para sistemas interactivos		
		Estado
	Procesos de diseño centrados en la persona para sistemas interactivos	ISO

6.2.2.10 ISO/IEC 14598

Este estándar comprende seis partes que especifican el proceso a seguir para evaluar software. La primera parte incluye la definición original de la calidad en el uso.

ISO/IEC 14598		
Evaluación de productos software		
		Estado
Parte 1	Visión general	ISO
Parte 2	Planificación y gestión	ISO
Parte 3	Proceso para los desarrolladores	ISO
Parte 4	Proceso para los adquirientes	ISO
Parte 5	Proceso para los evaluadores	ISO
Parte 6	Documentación de los módulos de evaluación	ISO



6.2.2.11 ISO TR 18529

Este estándar puede ser empleado para evaluar el punto al que una organización es capaz de llevar el diseño centrado en la persona. Contiene una lista formalizada y estructurada de procesos centrados en la persona: especificar el usuario y los requisitos de la organización, producir las soluciones de diseño, evaluar los diseños a partir de los requisitos, etc.

ISO TR 18529		
Descripciones de los procesos del ciclo de vida centrados en la persona		
		Estado
	Descripciones de los procesos del ciclo de vida centrados en la persona	ISO

6.2.2.12 ISO 10075

Este estándar comprende tres partes. Las dos primeras, reconocidas como estándares internacionales, especifican definiciones y términos relacionados con la sobrecarga mental y enuncian determinados principios ergonómicos relacionados con la misma. La tercera parte, todavía no aprobada como estándar, es la encargada de la medida y valoración de la sobrecarga.

ISO 10075		
Principios ergonómicos relacionados a la sobrecarga mental		
		Estado
Parte 1	Términos y definiciones generales	ISO
Parte 2	Principios de diseño	ISO
Parte 3	Medida y valoración de la sobrecarga mental	WD

6.2.3 Estándar de facto

Son estándares que nacen a partir de productos de la industria que tienen un gran éxito en el mercado, o bien a partir de desarrollos hechos por grupos de investigación de universidades y que tienen una gran difusión. Estos productos o proyectos de investigación llegan a tener un uso muy generalizado, convirtiéndose, por tanto, en estándares de facto (por ejemplo el sistema X Window). Su definición se encuentra en los manuales, libros o artículos. Son técnicamente muy valiosos y muy utilizados. La mayoría también son llamadas guías de estilo comercial.

6.4 Guías de estilo

Para poder asegurar consistencia a través de las diferentes partes de un sistema o a través de una familia de sistemas, es fundamental para los desarrolladores basar sus diseños en un conjunto de principios y directrices. Esto permite transferir sus conocimientos a los usuarios de la interfaz, dentro de un producto y a todas las aplicaciones en que trabaja. Por este motivo es tan importante para las organizaciones que desarrollan software disponer de una guía que puedan seguir sus desarrolladores. Estas guías se denominan *guías de estilo* y varían mucho en sus objetivos.

Las hay de dos tipos: *guías de estilo comercial*, producidas por fabricantes de software y hardware,



que son en general estándares de facto, y *guías de estilo corporativas*, creadas por las empresas para su propio uso.

La ventaja más evidente de las guías de estilo es que aseguran una mejor usabilidad mediante la consistencia que imponen. En el lenguaje industrial al hablar de un cierto tipo de guía de estilo se le da la denominación de *look and feel*.

6.4.1 Guías de estilo comerciales

Son guías de estilo diseñadas por las empresas de software. Están basadas en principios y contienen directrices que se concretan a muy bajo nivel. Las guías de estilo más relevantes son en las que están basadas los entornos operativos más importantes: Macintosh, OS/2, Windows y UNIX. Estos sistemas son propiedad de las empresas informáticas más importantes y cada una de ellas ha publicado su guía de estilo propia.

6.4.1.1 Apple

Apple publicó su primera guía el año 1985, Human Interface Guidelines: The Apple Desktop Interface en Inside Macintosh, con más de 600 páginas. Su nueva guía, Macintosh Human Interface Guidelines, publicada en 1992 es un libro muy ameno y con muchos ejemplos y gráficos. Acompañando a esta guía se han incorporado una serie de presentaciones multimedia en CD-ROM, Making it Macintosh, que contienen ejemplos animados que demuestran el empleo correcto de los elementos de interfaz de Macintosh, incluyendo ejemplos visuales que ilustran la apariencia y comportamiento de menús, diálogos, iconos, etc.

Combinando ambos elementos, está disponible actualmente la Electronic Guide to Macintosh Human Interface Design, que contiene enlaces hipertexto para una navegación sencilla entre el libro y las presentaciones multimedia de Making It Macintosh.

Más recientemente, Apple ha publicado Mac OS 8 Human Interface Guidelines, que describe las incorporaciones y los cambios a realizar en Macintosh Human Interface Guidelines relacionados con la versión 8 del Mac OS, pero que no reemplaza a ésta guía en absoluto.

6.4.1.2 CUA

La primera guía de las normas CUA que describía el estilo de interfaz de usuario Common User Access (CUA) de IBM fue publicada en 1987. En 1989, aparece la segunda versión de dicha guía con el nombre Advanced Interface Design Guide.

Hasta 1991 IBM y Microsoft trabajaron juntos, de forma que, cuando se publicaron estas normas todavía estaban juntos, por lo que las interfaces de usuario de los dos sistemas, DOS/Windows y OS/2, siguieron dichas normas. Esto supuso un éxito inmediato de estas normas y su reconocimiento como estándares de facto, debido sobre todo a la fuerza de IBM como primera empresa informática mundial.



En 1992, IBM publica Object–Oriented Interface Design: IBM Common User Access Guidelines, donde está completamente documentado uno de los primeros entornos de trabajo orientados a objeto.

6.4.1.3 Microsoft

En 1992, Microsoft publicó su primera guía en solitario: Microsoft. The Windows Interface: An Application Design Guide. En 1995 publica la siguiente versión The Windows Interface Guidelines for Software Design, que cubre el diseño de interfaces para el entorno Windows 95.

En el año 2000 se publica The Microsoft Windows User Experience, que incluye información para diseñar interfaces de usuario para aplicaciones que se ejecutan bajo Microsoft Windows 98 y Microsoft Windows 2000. Se incluye también en esta guía el soporte para las nuevas interfaces y convenciones empleadas en estos sistemas operativos. Esta guía hace referencia, sin ser una guía para el diseño de páginas web, recomendaciones para aplicaciones con estilo Web.

6.4.1.4 Common Desktop Environment (CDE)

CDE es una interfaz gráfica de usuario para UNIX en sus diferentes variantes (AIX, Digital UNIX, HP/UX, Solaris, ...), que ha sido desarrollada conjuntamente por HP, IBM, Novell y Sun, y que ha sido aprobada por la organización de estándares X/Open. Proporciona una especificación y una implementación de referencia de un entorno de escritorio, tanto para desarrolladores como usuarios de UNIX. Su objetivo es facilitar el uso de UNIX.

CDE está basado en estándares de facto de la industria como X.11, Motif y Tooltalk. De hecho, para escribir una aplicación conforme con el entorno CDE se deben considerar las guías de estilo de CDE (Common Desktop Environment: Style Guide 1.0 and Certification Checklist) y de Motif (OSF/Motif Style Guide). Sin embargo, con la versión 2.1 de Motif la guía de estilo que se publica incluye ya las guías de estilo de ambos, Motif 2.1 y CDE 2.1.

6.4.1.5 Motif

Motif es una interfaz gráfica de usuario para el sistema X Window desarrollada por la Open Software Foundation (OSF), un consorcio de empresas de informática entre las cuales se encontraban Hewlett–Packard y Microsoft. Proporciona una interfaz consistente para el entorno UNIX, permitiendo que los usuarios trabajen con múltiples aplicaciones que tiene características similares y consistentes.

Es importante, por tanto, tener en cuenta el hecho de que Motif, Windows y OS/2 parten de la misma guía de estilo original, las normas CUA, lo que les da un interés especial por el hecho de que asegura una consistencia entre todas las aplicaciones desarrolladas para estos entornos, y que constituyen el 90% de los sistemas informáticos existentes. Aunque esta uniformidad parecía muy útil, la ruptura entre IBM y Microsoft implicó que posteriormente cada uno publicara sus propias guías por separado, apareciendo así algunas diferencias con relación a las normas CUA originales.

Casi paralelamente a las diferentes versiones de Motif, se han ido generando guías de estilo para ellas. En 1992 se publica OSF/Motif Style Guide referenciando la versión 1.2 de Motif. Actualmente, la versión de Motif es la 2.1, que permite la generación de aplicaciones CDE 2.1. Es por ello que la



guía de estilo de Motif 2.1 incluye también la guía de estilo para CDE 2.1: Motif and CDE 2.1 Style Guide.

6.4.1.6 Open Look

Open Look es una interfaz gráfica de usuario definida por una especificación y una guía de estilo descrita en el libro Open Look Graphical User Interface Functional Specification, y desarrollada básicamente por Sun y AT&T. Emplea el protocolo de comunicación X11 del sistema X Window, al igual que lo hace su competidora Motif, pero presenta un look and feel diferente al el suyo.

Una de las implementaciones más importantes de Open Look es la hecha por Sun en su sistema Open Windows.

6.4.1.7 Java Look and Feel

El lenguaje y la plataforma Java permiten la ejecución de un mismo programa en diferentes plataformas, y para cada una de ellas Java utiliza la interfaz gráfica de la plataforma sobre la que se está ejecutando. Así, los que se ejecuten en Windows tendrán esa apariencia y los que se ejecuten en UNIX tendrán apariencia Motif. Esto ocurre siempre que los componentes empleados para construir la interfaz gráfica del programa sean AWT (Abstract Window Toolkit).

Sin embargo, la aparición del conjunto de componentes Swing, como parte de una nueva biblioteca de clases llamada JFC (Java Foundation Classes), permite la selección de esta apariencia gráfica, independientemente de la plataforma en la que se esté ejecutando; tan es así que, la apariencia por omisión de los componentes Swing se denomina Metal y es propia de Java. Además del Look&Feel Metal, Swing incorpora Look&Feel para Windows, Macintosh y Motif.

Una de las ventajas que representa el Look and Feel de Java es la posibilidad que se brinda a las empresas de poder crear una interfaz gráfica estándar y corporativa, ya que con el crecimiento de las intranets se están soportando muchas aplicaciones propias que deben ejecutarse en diferentes plataformas.

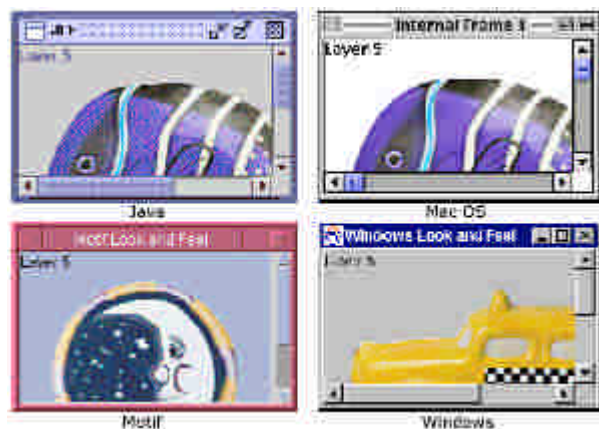


Figura 1 Diseños Look and Feel predefinidos disponibles bajo Swing



El Look and Feel de Java representa por tanto, la interfaz por omisión para las aplicaciones construidas con JFC, y se ha publicado una guía de estilo Java Look and Feel Design Guidelines que proporciona información esencial para construir interfaces gráficas de usuario para aplicaciones y applets con el lenguaje de programación Java y JFC. La primera versión de estas guías fue publicada en julio de 1999, y en marzo del 2001 ha sido publicada la segunda versión. A principios del 2002 apareció la publicación de una tercera versión más actualizada.

6.4.2 Guías de estilo para la Web

Diseñar para la Web es diferente que diseñar interfaces de usuario para el software tradicional, aunque también hay similitudes: ambos son sistemas interactivos y ambos son diseño de software, no diseño de objetos físicos. Es por ello, que algunos de los principios de diseño de interfaces de usuario tradicionales pueden ser directamente aplicables, pero obviamente la Web exige otra serie de consideraciones, partiendo del hecho de que la metáfora empleada para la Web ya no está basada en la metáfora del escritorio como en las interfaces gráficas tradicionales.

Una característica importante de la Web es la falta de interfaces de usuario comunes, ya que normalmente la prioridad es conseguir una interfaz atractiva, diferenciada de las otras, para que los usuarios visiten el sitio Web y lo vuelvan a visitar.

Con el fin de afrontar este problema las principales empresas informáticas (IBM, Apple, Sun, etc.), así como determinadas organizaciones han publicado y están publicando sus guías de diseño para Web. La mayoría de ellos acuden a su propio sitio Web para su publicación. Sin embargo, las diferencias entre estas guías son considerables. Algunas de estas guías son únicamente una colección de principios generales a seguir, como la propuesta por Apple, sin embargo, en otras como la propuesta por la Universidad de Yale el nivel de detalle reflejado en la guía es mayor.

6.4.2.1 Apple

Apple ha habilitado un sitio Web con el nombre Apple Web Design Guide que contiene principios de diseño de interfaces de usuario generales que pueden aplicarse al diseño de páginas Web. Esta información ha sido generada a partir de, la guía Macintosh Human Interface Guidelines, la información recogida desde la propia página Web, y la experiencia adquirida por la empresa durante años.

6.4.2.2 IBM

Proporciona una guía de diseño Web Design Guidelines disponible en el sitio Web Easy of Use de IBM. Las directrices que proporciona la convierten en una guía de diseño de nivel medio: abarca desde principios abstractos a convenciones.

Esta guía está organizada de acuerdo al proceso para desarrollar sitios Web: planificación, diseño, producción y mantenimiento, y ofrece además una sección especial para comercio electrónico. Su contenido es producto de diferentes fuentes:

- La experiencia de emplear el proceso de diseño centrado en el usuario (UCD – User-Centered Design) para crear sitios IBM.



- Los estudios realizados durante los últimos años sobre las diferentes secciones del sitio Web de IBM (incluyendo el sitio Easy of Use).
- Estudios de los usuarios que interactúan con sitios no IBM (ventas de libros, vídeos, etc.).
- Otras investigaciones publicadas por otros expertos en el campo.

6.4.2.3 Sun

La guía de Sun: Sun Guide to Web Style es un libro de recetas para ayudar a la gente a crear mejores páginas Web. Las directrices presentadas aquí representan las opiniones y preferencias de un pequeño grupo de gente dentro de Sun que han creado algunas páginas Web y han examinado muchas más; conocen la literatura existente sobre diseño de interfaces y usabilidad, y se encargan de los test de usabilidad sobre las páginas Web actuales. La información que proporciona está organizada en categorías (objetivo, audiencia, enlaces, longitud de la página, gráficos, etc.) pero es también bastante general.

6.4.2.4 W3C

Como se comentó con anterioridad el W3C es un consorcio internacional que promueve la evolución e interoperabilidad en la Web. Uno de sus dominios de actuación es la accesibilidad, y por ello alberga la Iniciativa de Accesibilidad Web (Web Accessibility Initiative, WAI), que está patrocinada por una gran variedad de organizaciones preocupadas por la accesibilidad.

Las guías juegan un papel muy importante para conseguir una Web accesible explicando como emplear tecnologías Web para crear sitios Web accesibles, browsers o herramientas autorizadas. WAI tiene tres guías diferentes para solucionar estas necesidades: Web Content Accessibility Guidelines, Authoring Tool Accessibility Guidelines y User Agent Accessibility Guidelines.

- Web Content Accessibility Guidelines (WCAG). WCAG contiene principios de diseño para hacer los sitios Web accesibles. Estudian escenarios corrientes que pueden ocasionar problemas para usuarios discapacitados. Por ejemplo, la primera directriz explica como se pueden hacer imágenes accesibles, ya que algunos usuarios pueden no ser capaces de ver las imágenes, otros pueden emplear browsers basados en texto que no soportan imágenes, etc.

La versión 1.0 de estas guías ha sido reconocida ya como una recomendación de W3C desde 1999, y existe una versión 2.0 como un borrador de trabajo desde Agosto del 2001.

- Authoring Tool Accessibility Guidelines (ATAG). ATAG son reconocidas en su versión 1.0 como recomendaciones W3C desde Febrero del 2000. El objetivo de estas guías es asistir a los constructores en el diseño de herramientas autorizadas que producen contenidos Web accesibles, y asistir a los desarrolladores en la creación de una interfaz autorizada accesible. Como la mayoría del contenido de la Web se crea empleando herramientas autorizadas, éstas juegan un papel importante para asegurar la accesibilidad de la Web, y además, ya que la Web es un medio tanto de recibir como de comunicar información, es importante que tanto el contenido producido, como la herramienta sean accesibles.

Es por eso que los objetivos de estas guías son, por un lado, conseguir que las herramientas autorizadas sean accesibles para los autores, independientemente de su



discapacidad, y por otro lado, que produzca un contenido accesible por omisión y que esto anime al autor en la creación de contenidos accesibles.

- User Agent Accessibility Guidelines (UAAG). Actualmente W3C ha publicado UAAG en su versión 1.0 como recomendación candidata. UAAG 1.0 explica como la navegación con teclado, las opciones de configuración, la documentación, la comunicación con software especializado como por ejemplo los sintetizadores de voz, y otras características de las interfaces de usuario, benefician a la gente con discapacidades visuales, auditivas, físicas, cognitivas y neurológicas.

6.4.2.5 Yale Center for Advanced Instructional Media

El Yale Center for Advanced Instructional Media (C/AIM) ha construido una guía de estilo que describe los principios de diseño empleados para crear las páginas dentro del sitio Web del C/AIM. Esta guía se encuentra disponible en el sitio Web del C/AIM, y además ha sido recientemente publicada con el nombre Web Style Guide: *Basic Design Principles for Creating Web Sites*.

Esta guía es una de las más reconocidas y cubre todos los elementos básicos que se ven implicados en la creación de un sitio Web centrándose en la interfaz y en los principios de diseño gráfico subyacentes al diseño de un sitio Web. Aconseja sobre la planificación y organización de los objetos, el diseño de estrategias para un sitio o sobre el diseño de páginas individuales.

6.4.2.9 National Cancer Institute (NIC)

El NIC publica en su sitio Web: Web Design & Usability Guidelines, un conjunto de aproximadamente unas 50 directrices sobre la usabilidad y diseño Web, basadas en investigaciones e información contrastada sobre el tema.

El NIC, principal agencia del gobierno federal americano para la investigación contra el cáncer, diseña y gestiona un gran número de sitios Web relacionados con el cáncer. Estas guías fueron desarrolladas para ayudar a los gestores Web, diseñadores y autores a mejorar sus esfuerzos de diseño y ayudarles a tomar sus decisiones. Sin embargo, como los principios para un buen diseño y usabilidad son aplicables a cualquiera que trabaje con sitios Web, el NIC se decidió a poner estas guías a disposición del público en general.

6.4.3 Guías de estilo corporativas

Las guías de estilo corporativas se centran en presentaciones comunes, comportamientos y técnicas que deben ser implementadas por todos los productos en una compañía. Un objetivo de las guías de estilo corporativas es mantener y reforzar la identidad corporativa, esto es, el uso de colores, gráficos, e iconos que presenten una imagen visual consistente del logotipo de la compañía a través de todas las interfaces de los productos.

El método sugerido para la construcción de una guía de estilo corporativa consiste en basarse en una guía de estilo de la industria. Un ejemplo para la construcción de una guía de estilo corporativa puede ser el expuesto por Theo Mandel, en el que para desarrollar una guía de estilo corporativa para una empresa parte de la guía de diseño de Microsoft Windows 95, pero le añade nuevos elementos



siempre que éstos pertenezcan a una de estas tres categorías: suplemento, complemento o contradicción. Una entrada como suplemento proporciona información que mejora o clarifica un tema en la guía de diseño de Microsoft. Una entrada como complemento proporciona información que no estaba contenida en un tema concreto en la guía de Microsoft, y una contradicción contiene información para los desarrolladores que supone una excepción a la guía de diseño de Microsoft. En este caso, deberían seguirse las recomendaciones de las guías en vez de la guía de Microsoft.

Además de una guía de estilo corporativa, se pueden necesitar guías de diseño para un grupo de productos relacionados (product suite guide), o para productos individuales (product style guides). Todos estos documentos deberían estar basados en los estándares subyacentes, guías de diseño, y guías de estilo corporativas ya desarrolladas.



Figura 2 Pirámide implicada en la construcción de guías de estilo corporativas y de producto

6.5 Consideraciones sobre los estándares y las guías de estilo

Los estándares y guías son bloques de construcción sobre los que se pueden basar los esfuerzos de diseño y de desarrollo. Sin embargo, el hecho de que sean seguidos no garantiza la usabilidad del producto, al igual que una casa construida de acuerdo a los estándares de construcción no garantiza que sea una casa habitable. De hecho, un error común es la creencia de que si se han seguido los estándares y las guías, la interfaz es usable, y el producto no necesita hacer ningún test de usabilidad.

Es importante dejar constancia de que los estándares y las guías son parte del proceso de diseño que incluye, además de principios de diseño, una metodología de diseño, test de usabilidad, y un entendimiento pleno de los usuarios. De hecho, si los test de usabilidad indican que se debería realizar una desviación de las guías, entonces hay que sopesar los resultados de los test de usabilidad con los beneficios de las guías seguidas, y tomar una decisión.

No obstante, aunque el empleo de guías y estándares no garantiza que la interfaz sea usable, es evidente que es mejor seguirlos que no seguirlos, porque aunque a veces se pueda hacer un mejor diseño sin ellos, son muchas más las ventajas que aportan que las desventajas.

Por otro lado, para poder aplicar las guías de estilo correctamente hace falta dar facilidades a los diseñadores y programadores. Así, los ejemplos, son muy útiles en la presentación de las guías. También es fundamental el hecho de contar con herramientas que permitan que las interfaces sean diseñadas de acuerdo a las guías de interfaz de usuario vigentes.