

DESARROLLO DE APLICACIONES MULTIPLATAFORMA 2

Contenido

1.- Evolución de la interfaz visual	2
2.- Interfaz de usuario	5
Las funciones principales	5
Tipos	6
Según la forma de interactuar del usuario.	6
• Interfaz de línea de comandos	6
• Interfaces gráficas de usuario	6
• La interfaz natural de usuario.....	7
Según su construcción.	8
Valoración.....	8
3.- La ISO 9241	10
4.- Introducción a las WCAG.....	11
5.- WYSIWYG.....	14
6.- Características de una aplicación, de principio a fin.	15
7.- La forma de diseñar el interfaz de usuario	17
• Modelo del usuario:.....	17
• Modelo del diseñador:.....	17
• Modelo del programador:.....	18
9.- IDE - Entorno de Desarrollo Interactivo.	19
Características entre diferentes plataformas computacionales.....	19

1.- Evolución de la interfaz visual

La interfaz gráfica de los sistemas operativos modernos es uno de los apartados más cuidados por los desarrolladores, y mucho ha llovido desde los primeros avances en este campo. ¿Cómo han evolucionado estos entornos gráficos? Lo comprobamos en un repaso de 33 versiones distintas de sistemas operativos legendarios como AmigaOS o OS/2, además de las evoluciones de Windows, GNU/Linux y el prodigioso Mac OS.

El diseño de la Interfaz Gráfica de Usuario (GUI) ha sido uno de los apartados más cuidados por los desarrolladores de sistemas operativos desde hace casi 3 décadas. El revolucionario descubrimiento de este componente por parte del Xerox Palo Alto Research Center (PARC) fue poco a poco trasladado a sistemas operativos comerciales que comenzaban su andadura en un terreno desconocido, pero que a la vez acercaría como nunca la informática al usuario final.

De hecho, gracias a la interfaz gráfica de usuario los sistemas operativos se convirtieron en soluciones mucho más visuales y accesibles para el gran público, algo que hasta la fecha era complicado debido a las funciones de unos sistemas basados en modo texto que resultaban demasiado limitados para un gran número de escenarios. A continuación os mostramos la evolución de esas interfaces gráficas de usuario a partir del excelente artículo original de Webdesigner Depot.

1. Xerox Alto

El primer ordenador personal que utilizó una interfaz gráfica de usuario moderna fue el Xerox Alto, que se desarrolló en 1973. Sin embargo, no era un producto comercial, y estaba destinado únicamente a la investigación en universidades.

3. Apple Lisa Office System 1 (1983)

También conocido como OS, que en este caso significaba Office System (y no Operating System). Este desarrollo fue producido por Apple con la intención de convertirlo en una estación de trabajo para el procesamiento de todo tipo de documentos. Desafortunadamente este desarrollo fue "canibalizado" por el sistema operativo de los Apple Macintosh, que era más económico.

4. VisiCorpo Visi On (1984)

Este desarrollo fue la primera GUI disponible para el IBM PC. La idea era destinarlo a grandes corporaciones, y su precio por entonces era muy alto. Para utilizarlo se usaba un ratón, y además disponía de un instalador y un sistema de ayudas, y no hacía uso de iconos.

5. Mac OS System 1.0 (1984)

System 1.0 fue la primera GUI desarrollada para el Macintosh, y disponía de varias de las prestaciones de un sistema operativo moderno, incluidas el uso de ventanas e iconos. Era posible mover ventanas de un lado a otro, y también se introdujo la función de arrastrar y soltar iconos de un origen a un destino.

6. Amiga Workbench 1.0 (1985)

El sistema operativo del Amiga y su interfaz gráfica se adelantaron a su tiempo, y ofrecían características imbatibles para la época. Cuatro colores (negro, blanco, azul y naranja), multitarea preemptiva -que no adoptaron muchos otros sistemas hasta casi 10 años después-, sonido estéreo e iconos multi-estado (seleccionados y no seleccionados).

7. Windows 1.0x (1985)

Ese mismo año Microsoft por fin logró desarrollar su primer entorno gráfico para su sistema operativo MS-DOS, al que llamó Windows 1.0 y que disponía de iconos de 32x32 píxeles y gráficos en color. Una de las características más originales fue el icono animado del reloj, que más tarde desaparecería.

9. IRIX 3 (1986)

Aunque se presentó en 1984, la interfaz gráfica de usuario del potente sistema Unix de Silicon Graphics no se lanzaría realmente hasta 1986, y daría aún más potencia al sistema operativo de 64 bits IRIX, creado a partir de UNIX. Esta GUI ofrecía soporte para iconos vectoriales, una característica que por ejemplo también haría su aparición muchos años después en Mac OS X.

13. NeXTSTEP / OPENSTEP 1.0 (1989)

La marcha de Steve Jobs de Apple le sirvió para crear una nueva empresa y un nuevo sistema operativo que fue concebido para gobernar los ordenadores de investigación de universidades y centros científicos, algo que le llevaría a fundar su compañía NeXT Computer Inc. El primer ordenador de esa empresa se presentó en 1988, pero en 1989 fue cuando se presentó la GUI de NeXTSTEP, que más tarde evolucionaría hacia OPENSTEP. Los iconos de dicha interfaz eran más grandes (48x48) y con más colores, y se convirtió en un verdadero preludio de lo que serían las interfaces gráficas de usuario de los siguientes años.

14. OS/2 1.20 (1989)

La siguiente versión de OS/2 fue presentada un año después, pero las mejoras visuales eran notables en muchos apartados, como el de los iconos, más estilizados, y las ventanas, también con mejor aspecto visual.

15. Windows 3.0 (1990)

Esta versión demostró el interés que Microsoft tenía en los entornos visuales, y de hecho aprovechó las prestaciones de los procesadores 386, que podían acceder a más de 640 KB y con las que lograron por ejemplo resoluciones de hasta 1024x768 píxeles. La interfaz también cambió de diseño e iconos, con un aspecto notablemente mejor al de las ediciones previas.

17. Mac OS System 7 (1991)

Esta versión de Mac OS incluyó la primera GUI de Apple con soporte para colores, además de que se añadieron distintas tonalidades de gris, azul y amarillo a los iconos del sistema.

18. Windows 3.1 (1992)

La versión de Windows fue la primera en incluir tipografías TrueType, que estaban instaladas por defecto y que convirtieron a Windows en una plataforma eficiente para autoedición por primera vez, un tipo de tarea que los Macintosh y Apple llevaban dominando durante años. En Windows 3.1 también se incluyó un nuevo esquema de color denominado Hotdog Stand, que contenía contrastes de rojo, amarillo y negro para darle mayor relieve a los iconos y ventanas.

19. OS/2 2.0 (1992)

Esta fue la primera interfaz gráfica de usuario que se sometió a la aceptación internacional, además de realizarse pruebas de usabilidad y accesibilidad. Toda ella fue desarrollada utilizando un diseño orientado a objetos, y cada fichero y carpeta era un objeto que podía ser relacionado con otros ficheros, carpetas y aplicaciones. También se introdujo la función de arrastrar y soltar ficheros.

20. Windows 95 (1995)

Este fue el primer sistema operativo de Microsoft que integraba una interfaz gráfica de usuario nativa en el sistema -aunque la base siguiera siendo MS-DOS-, que también se mantuvo en W98 y Me. En Windows 95 se incluyeron numerosas mejoras sobre las anteriores versiones de Windows, y entre otras cosas se les dio varios estados a los iconos. El famoso botón de Inicio (Start) apareció por primera vez, pero también lo hicieron otros conceptos que aún hoy se mantienen en Windows Vista o Windows 7 y que simplemente han visto modificado su aspecto visual.

21. OS/2 Warp 4 (1996)

El lanzamiento de esta versión de OS/2 Warp supuso una revolución para este desarrollo, que se convirtió para muchos en una referencia en el mercado de sistemas operativos aunque su popularidad se viese ensombrecida por la arrolladora máquina de márketing de Microsoft y sus Windows 95. La interfaz de OS/2 Warp 4 introdujo cambios interesantes como el posicionamiento de iconos en el escritorio, donde también se podían incluir ficheros y carpetas. La papelera de reciclaje no mantenía los ficheros para un posterior borrado definitivo, sino que una vez que los depositábamos allí se borraban para siempre automáticamente.

24. KDE 1.0 (1998)

La primera versión de un entorno gráfico para el sistema operativo GNU/Linux marcó un punto de inflexión crucial en las distribuciones Linux, y ofreció características que antes sólo era posible encontrar en otras plataformas o que eran complejas de integrar en los gestores de ventanas que existían previamente para la interfaz gráfica del sistema operativo GNU/Linux. En muchos apartados KDE trataba de emular el comportamiento de menús y ventanas de Windows 95, quizá como medio para tratar de convencer a usuarios de Windows de que migraran a soluciones Open Source.

26. GNOME 1.0 (1999)

La otra gran alternativa en el mercado de los entornos gráficos de escritorio para GNU/Linux también surgió ese año, y aunque aparecería originalmente para la distribución Red Hat Linux pronto se utilizaría en un gran número de distribuciones, que lo tomarían como solución por defecto. Ubuntu, por ejemplo, lo aplicó desde su nacimiento, aunque obviamente también existen versiones como Kubuntu en las que el entorno de escritorio es KDE.

27. Mac OS X (2001)

Probablemente el más importante de los sistemas operativos modernos en este apartado, Mac OS X supuso un cambio radical respecto a la evolución que hasta entonces habían marcado los sistemas operativos de Apple. La interfaz visual, conocida como Aqua, introdujo múltiples cambios que hoy en día son imitados por el resto de alternativas en el mercado. Iconos de gran tamaño con anti-aliasing y semitransparencias, la aparición del Dock o el diseño de las ventanas fueron algunas de las novedades, que no tuvieron buena recepción inicial por el gran cambio que supusieron, pero que a posteriori se han convertido en referentes de un sistema operativo y una interfaz que son una referencia absoluta en el mercado actual.

31. Mac OS X Leopard (2007)

La sexta generación del sistema operativo de Apple fue una de las más importantes en cuanto a cambios visuales, con un Dock con aspecto 3D, muchas más animaciones e interactividad y la popularización de métodos de visualización como su famoso CoverFlow en un buen montón de aplicaciones, incluido el Finder.

33. KDE 4.x (2008)

Aunque las primeras versiones de este entorno de escritorio fueron objeto de muchas críticas por su estado aún demasiado experimental, la versión 4.2 que ha debutado hace pocas semanas ha permitido demostrar que KDE 4 ha madurado de una forma sobresaliente, y hoy en día se considera una solución estable y eficiente en la mayoría de los apartados. Muchos usuarios siguen considerándolo aún un cambio demasiado radical -como Mac OS X lo fue en su día- pero es probable que el paradigma introducido con esta versión sea muy pronto una referencia muy relevante en este mercado. Los cambios visuales e internos -con una arquitectura modular potentísima- hacen de KDE 4 un entorno gráfico con un futuro realmente prometedor.

Ejercicio: ¿Cómo han evolucionado las GUI de 2008 a 2015? Tactil, NUI, ...

2.- Interfaz de usuario

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.

Normalmente suelen ser fáciles de entender y fáciles de accionar, aunque en el ámbito de la informática es preferible referirse a que suelen ser "amigables e intuitivos" porque es complejo y subjetivo decir "fácil".

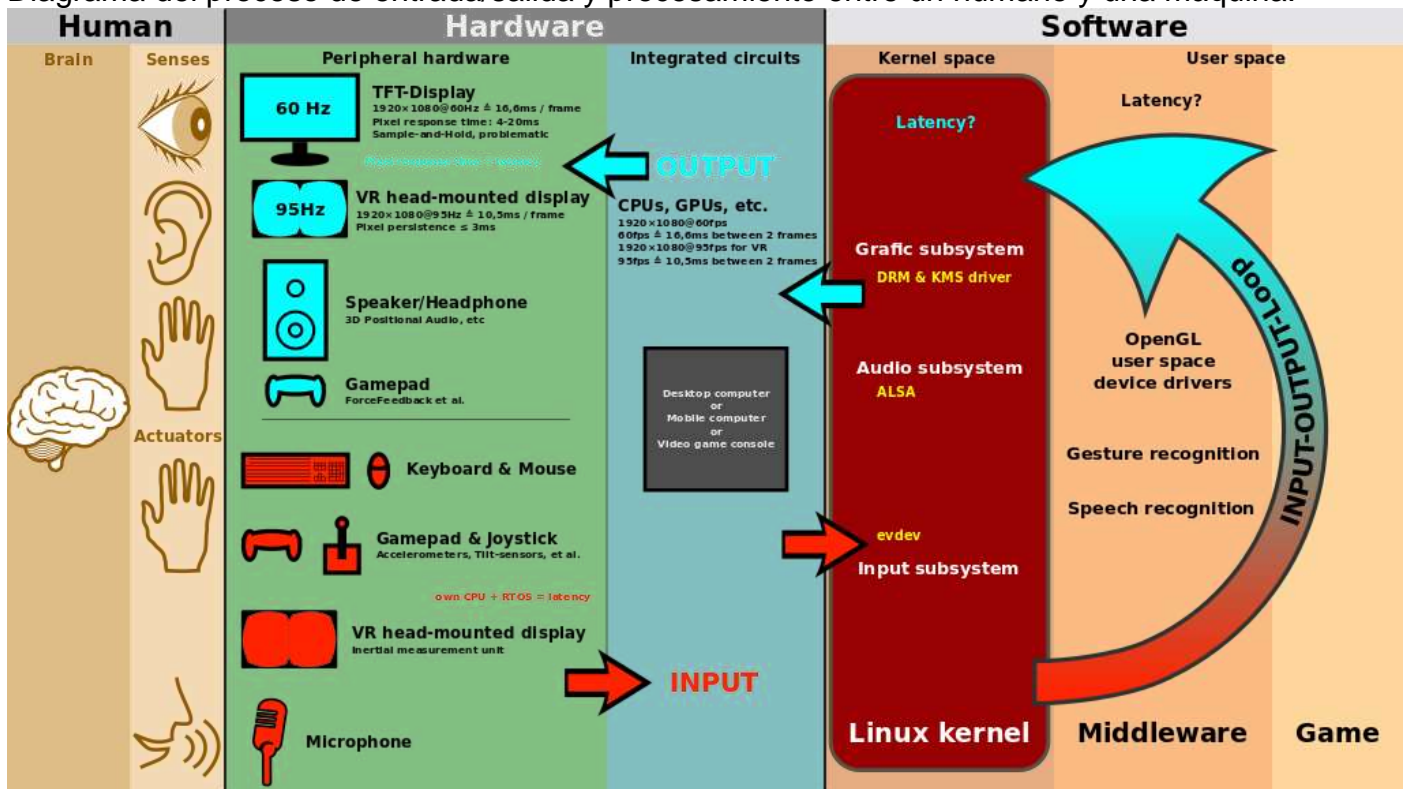
Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, contenido gráfico, cursor, los *beeps* y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora.

La mejor interacción humano-máquina a través de una adecuada interfaz (de usuario), que le brinde tanto comodidad, como eficiencia.

Las funciones principales son las siguientes:

- Puesta en marcha y apagado.
- Control de las funciones manipulables del equipo.
- Manipulación de archivos y directorios.
- Herramientas de desarrollo de aplicaciones.
- Comunicación con otros sistemas.
- Información de estado.
- Configuración de la propia interfaz y entorno.
- Intercambio de datos entre aplicaciones.
- Control de acceso.
- Sistema de ayuda interactivo.

Diagrama del proceso de entrada/salida y procesamiento entre un humano y una máquina.



Tipos

En las interfaces de usuario se pueden distinguir básicamente tres tipos:

- Una interfaz de hardware, a nivel de los dispositivos utilizados para ingresar, procesar y entregar los datos: teclado, ratón y pantalla visualizadora.
- Una interfaz de software, destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario observa habitualmente en la pantalla.
- Una interfaz de software-hardware, que establece un puente entre la máquina y las personas, permite a la máquina entender la instrucción y al hombre entender el código binario traducido a información legible.

Según la forma de interactuar del usuario.

Atendiendo a como el usuario puede interactuar con una interfaz, existen varios tipos de interfaces de usuario:

- **Interfaz de línea de comandos** (Command-Line Interface, **CLI**): Interfaces alfanuméricas (intérpretes de comandos) que solo presentan texto.
- **Interfaces gráficas de usuario** (Graphic User Interface, **GUI**): Permiten comunicarse con la computadora de forma rápida e intuitiva representando gráficamente los elementos de control y medida.
Interfaces táctiles, que representan gráficamente un «panel de control» en una pantalla sensible al tacto que permite interactuar con el dedo de forma similar a si se accionara un control físico.

El manejo de las GUI requiere de un dispositivo señalador (por ejemplo, un ratón). La historia reciente de la informática está indisolublemente unida a las interfaces gráficas, puesto que los sistemas gráficos han ocasionado grandes consecuencias en la industria del software y del hardware.

Las interfaces gráficas surgen de la necesidad de hacer las computadoras más accesibles para el uso de los usuarios comunes. La mayoría de computadoras domésticas, requerían conocimientos de BASIC (el 95 % de ellos incorporaban un intérprete BASIC como entorno operador) u ofrecían una interfaz de línea de órdenes (como los sistemas operativos CP/M o los diferentes OS del Apple II), lo que requería conocimientos técnicos si se deseaba hacer algo más que usarlo como consola de videojuegos.

Esta limitación fue salvada gracias al desarrollo de los entornos gráficos, que permitieron que las personas pudieran acceder a una computadora sin tener que pasar por el tortuoso proceso de tener que aprender a manejar un entorno bajo línea de órdenes.

Tipos de interfaces gráficas de usuario.

- Las GUI y ZUI

Los tipos de interfaces gráficas de usuarios (GUI) que se encuentran en juegos de computadora, y las GUI avanzados basados en realidad virtual, se usan con frecuencia en tareas de investigación. Muchos grupos de investigación en Norteamérica y Europa están trabajando actualmente en la interfaz de enfoque del usuario o ZUI (Zooming User Interface), que es un adelanto lógico de las GUI, mezclando 3D con 2D. Podría expresarse como «2 dimensiones y media en objetos vectoriales de una dimensión».

- Interfaz de usuario de pantalla táctil

Algunas GUI son diseñados para cumplir con los rigurosos requisitos de los mercados verticales. Estos se conocen como las GUI de uso específico. Un ejemplo de un GUI de uso específico es la ahora familiar pantalla táctil o touchscreen (pantalla que al ser tocada efectúa los comandos del ratón en el software). Se encuentra actualmente implementado en muchos restaurantes y en muchas tiendas de autoservicio de todo el mundo. Fue iniciado por Gene Mosher en la computadora del ST de Atari en 1986, el uso que él especificó en las GUI de pantalla táctil ha encabezado una revolución mundial en el uso de las computadoras a través de las industrias alimenticias y de bebidas, y en ventas al por menor.

Otros ejemplos de GUI de uso específico, relacionados con la pantalla táctil son los cajeros automáticos, los kioscos de información y las pantallas de monitoreo y control en los usos industriales, que emplean un sistema operativo de tiempo real (RTOS). Los teléfonos móviles y los sistemas o consolas de juego también emplean las pantallas táctiles. Además la domótica no es posible sin una buena interfaz de usuario, o GUI.

- Interfaz Natural de Usuario (NUI)

Las NUI naturales son aquellas en las que se interactúa con un sistema, aplicación, etcétera, sin utilizar dispositivos de entrada como ratón, teclado, lápiz óptico, etc.

- La **interfaz natural de usuario** (en inglés natural user interface, **NUI**) es el tipo de interfaz de usuario en las que se interactúa con un sistema, aplicación, etcétera, sin utilizar sistemas de mando o dispositivos de entrada (como en las interfaces gráficas de usuarios, sería un ratón, teclado alfanumérico, lápiz óptico, panel táctil, joystick, etcétera), y en su lugar, se utilizan las manos o las yemas de los dedos, se hace uso de movimientos gestuales del cuerpo o de alguna de sus partes tales como las manos, sirviendo de mando de control. En el caso de pantallas capacitivas multitáctiles, la operación o control es por medio de la yemas de los dedos, en uno o varios contactos y asimismo el control cercano a la pantalla, pero sin tocarla.

También existe el control de sistemas operativos por medio de la voz humana, denominado control por reconocimiento del habla o reconocimiento de voz, como por ejemplo Siri, Google Now u OK Google.

Un ejemplo de dispositivo con NUI es el Xbox Kinect.

Evolución de las interfaces de usuario.



Según su construcción.

Pueden ser de hardware o de software:

- Interfaces de hardware: Se trata de un conjunto de controles o dispositivos que permiten que el usuario intercambie datos con la máquina, ya sea introduciéndolos (pulsadores, botones, teclas, reguladores, palancas, manivelas, perillas) o leyéndolos (pantallas, diales, medidores, marcadores, instrumentos).
- Interfaces de software: Son programas o parte de ellos, que permiten expresar las órdenes a la computadora o visualizar su respuesta.

Valoración

El principal objetivo de una interfaz de usuario es que este pueda comunicar información a través de ella hacia algún tipo de dispositivo o sistema. Conseguida esta comunicación, el siguiente objetivo es el que dicha comunicación se desarrolle de la forma más fácil y cómoda posible para las características del usuario que utiliza el servicio.

Sin embargo, las interfaces no siempre cumplen todos los objetivos, por ejemplo: como es el caso de las consolas de línea de órdenes (CLI), que se encuentran en algunos sistemas de encaminadores (como los NOS de los routers); algunas consolas de administración con sus intérpretes de comandos (shell) de Unix, DOS, etcétera; y también en las consolas de administración de algunos servidores dedicados como Microsoft Exchange Server. Estas interfaces son fáciles de usar, sin embargo se necesita un amplio conocimiento de la persona que las utiliza. Por lo que, tanto su curva de aprendizaje, como el conocimiento técnico previo a su uso impiden que puedan ser utilizadas por cualquier persona.

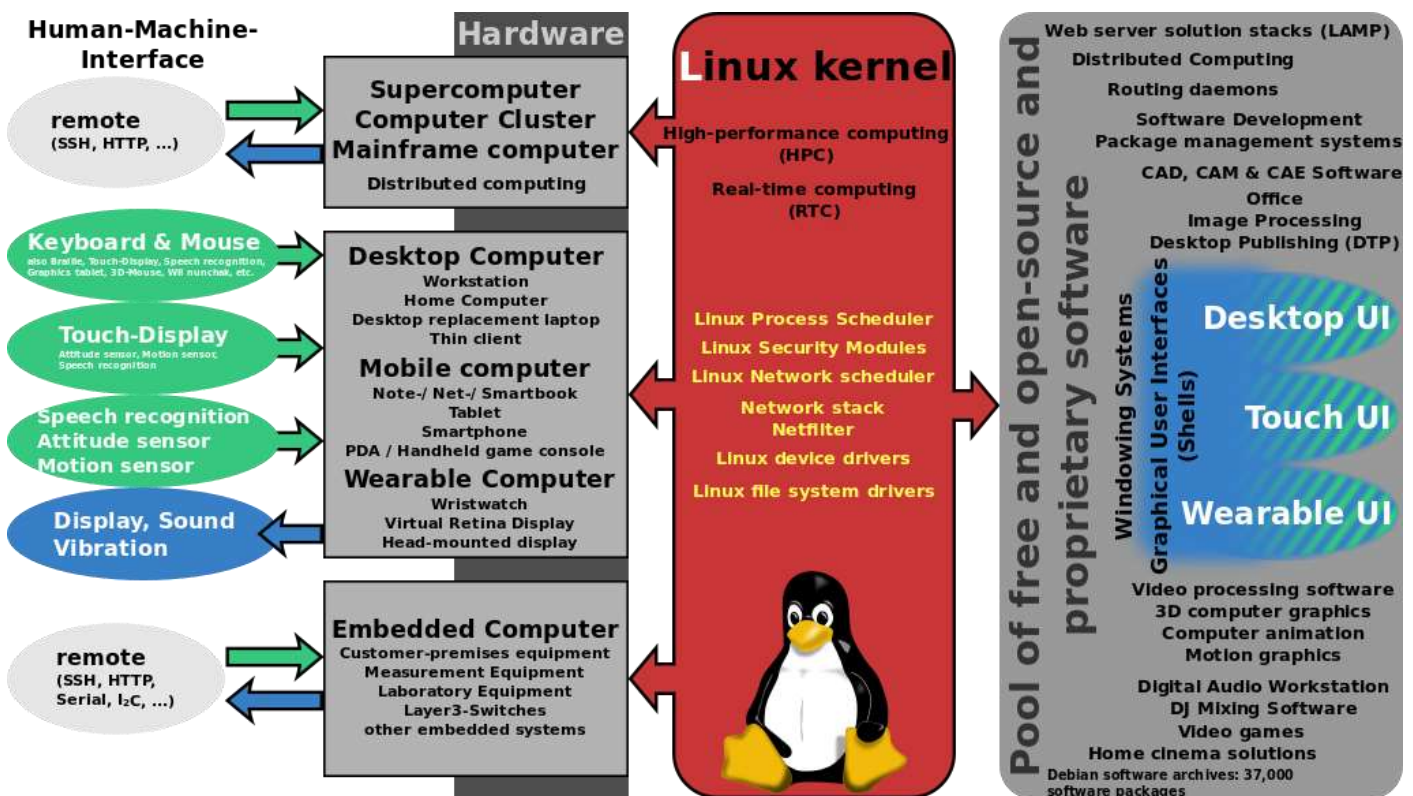
Si bien estas interfaces son las primeras que utilizaron las computadoras, y muchos usuarios podrían considerarlas anticuadas, siguen siendo incluidas en nuevos dispositivos y sistemas gracias a las ventajas que ofrecen al permitir automatizar acciones complejas mediante la creación de pequeños programas de bajo nivel (conocidos como Script o Batch).

Por otra parte, existen interfaces que reducen significativamente la curva de aprendizaje y permiten que usuarios sin experiencia y sin conocimientos técnicos puedan obtener resultados notables, por ejemplo: la interfaz táctil utilizada por los sistemas operativos de iOS y Android.

Si bien el diseño de la interfaz es crítico para el manejo del dispositivo, los diseñadores al momento de su creación ponen especial énfasis en determinar el tipo de usuario, su conocimiento y su experiencia. Esto marcará importantes diferencias entre la interfaz de línea de comandos de un gestor de correo electrónico, los menús flotantes para una aplicación de diseño gráfico o bien el despliegue de información mediante una línea de tiempo en una red social.

La tendencia a futuro se vislumbra con una importante separación entre interfaces para la creación de contenidos e interfaces para el consumo de contenidos. Como puede ser cotejado con el uso de sistemas de código de barras, sistemas de acceso RFID Etiquetas o bien Social Networks ER, para la creación de contenidos; y dispositivo como los llamados smartWatch, smartTV y tabletas.

La interfaz gráfica depende de periférico de entrada.



3.- La ISO 9241

Es un estándar que proporciona requisitos y recomendaciones relacionados con los atributos del Hardware, software y el entorno que contribuyen a la usabilidad y los principios ergonómicos que subyacen.

- UNE-EN ISO 9241-12:1999. Requisitos ergonómicos para trabajos de oficina con pantallas de visualización de datos (PVD). Parte 12: Presentación de la información. Proporciona recomendaciones ergonómicas para la **presentación de la información** así como propiedades específicas de la información **presentada en interfaces, gráficas o basadas en texto**, utilizadas para tareas de oficina. En cuanto al color, se limita a recomendaciones ergonómicas en el empleo del color para destacar y jerarquizar la información.
- UNE-EN ISO 9241-14:1999. Requisitos ergonómicos para trabajos de oficina con pantallas de visualización de datos (PDV). Parte 14: Diálogos mediante menús. Proporciona recomendaciones condicionales para los menús utilizados en diálogos usuario-ordenador para realizar tareas de oficina típicas. Comprende los **menús presentados mediante diferentes técnicas: ventanas, paneles, botones, campos, etc.** Las recomendaciones hacen referencia a los tres componentes principales de diseño de interfaces de usuario: diálogo, entrada y salida.
- UNE-EN ISO 9241-129:2011. Ergonomía de la interacción hombre sistema. Parte 129: Directrices sobre la individualización de software. Proporciona directrices de ergonomía para la individualización en los sistemas interactivos, incluyendo recomendaciones sobre cuándo la individualización puede ser o no apropiada y cómo aplicarla. Se centra en la **individualización de la interfaz de usuario de software** para apoyar las necesidades de los usuarios como individuos o como miembros de un grupo en concreto, proporcionando orientación sobre cómo los diversos aspectos de la individualización pueden resultar utilizables y accesibles, pero sin especificar qué individualizaciones han de ser incluidas en un sistema, ya que éstas dependerán del contexto específico de uso del mismo. Algunas de las recomendaciones pueden aplicarse también a las interfaces de usuario de hardware y a las interfaces de usuario que combinan software y hardware.
- UNE-EN ISO 9241-151:2008. Ergonomía de la interacción hombre-sistema. Parte 151: Directrices para las interfaces de usuario Web. Proporciona directrices sobre el diseño centrado en el usuario, para las **interfaces de usuario web** con el objetivo de aumentar su usabilidad.
- EN ISO 9241-154:2013. Ergonomics of human-system interaction - Part 154: Interactive voice response (IVR) applications. Proporciona orientaciones y requisitos para el diseño de **la interfaz de usuario de aplicaciones de respuesta de voz interactiva (IVR)**. Se ocupa tanto de los sistemas de IVR que emplean como mecanismo de entrada comandos de teclado como de los que emplean reconocimiento de voz automático (ASR). Es igualmente aplicable a los casos en los que la persona que llama o el sistema IVR por sí mismo inicia la llamada (por ejemplo, en algunas aplicaciones de telemarketing). Está destinado a ser utilizado en conjunto con la norma ISO / IEC 13714 (Information technology -- Document processing and related communication -- User interface to telephone-based services -- Voice messaging applications).

4.- Introducción a las WCAG

La Iniciativa de Accesibilidad a la Web del W3C (WAI), fundada en 1997, es un grupo de trabajo permanente del W3C (Consortio World Wide Web - <http://www.w3c.es>).

La W3C es una organización internacional que TRABAJA en el desarrollo de estándares web, y que recibe el apoyo de los principales actores de la industria y los gobiernos del mundo.

LA WAI se dedica a promover soluciones de accesibilidad en la web para personas con discapacidades. Actúa principalmente sobre cinco áreas de trabajo:

- Asegurar que las tecnologías web den soporte a la accesibilidad
- Desarrollar pautas de accesibilidad
- Crear herramientas de evaluación y corrección de la accesibilidad web
- Desarrollar materiales para la educación y difusión
- Coordinar proyectos de investigación y desarrollo

La accesibilidad web incluye los contenidos y aplicaciones, los navegadores y reproductores multimedia, las herramientas de autor y las tecnologías XML. La WAI ha propuesto para cada una de estas necesidades unas pautas a seguir.

En 1999 se publica las **Pautas de Accesibilidad para el Contenido Web**, WCAG 1.0 (Web Content Accessibility Guidelines, WCAG), que representan el primer y más grande esfuerzo por establecer unas pautas de diseño ACCESIBLE.

El 11 de diciembre de 2008 se publica la nueva recomendación, las WCAG 2.0.

Las WCAG 1.0 y las WCAG 2.0 están organizadas y estructuradas de distinta manera.

Las WCAG 1.0 se organizan en 14 pautas que constituyen los principios generales del diseño accesible. Cada una de estas pautas tiene asociados x puntos de verificación (65 en total) que explican cómo se aplica la pauta. A su vez, cada punto de verificación tiene asignada una prioridad (1, 2, 3).

El nivel de adecuación de accesibilidad (nivel de conformidad) será:

- Simple - A (A): cuando cumple todos los puntos de verificación de prioridad 1.
- Doble - A (AA): cuando cumple todos los puntos de verificación de prioridad 1 y 2.
- Triple - A (AAA): cuando cumple todos los puntos de verificación de prioridad 1, 2 y 3.

Las WCAG 2.0 se organizan en 4 principios fundamentales para la accesibilidad del contenido:

- PERCEPTIBLE: La información y los componentes de la interfaz de usuario deben ser PRESENTADOS a los usuarios de modo que ellos puedan percibirlos.
- OPERABLE: Los componentes de la interfaz de usuario y la NAVEGACIÓN deben ser operables.
- COMPRENSIBLE: La información y el manejo de la interfaz de usuario deben ser comprensibles.
- ROBUSTO: El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo los productos de apoyo.

A su vez, cada uno de estos grandes principios tiene asociadas unas pautas. En total son 12 pautas: los dos primeros principios tienen 4 pautas asociadas, el tercero tiene 3 y el último 1 pauta. Estas pautas no son testeables en sí, sino que proporcionan las metas básicas para hacer el CONTENIDO ACCESIBLE, y sirven para comprender los criterios de conformidad e implementarlos.

Cada una de estas pautas tiene asociados x criterios de conformidad (61 en total) que se han de cumplir y que sí son testeables. Los criterios de conformidad están ordenados según su nivel de cumplimiento asociado (A, AA y AAA).

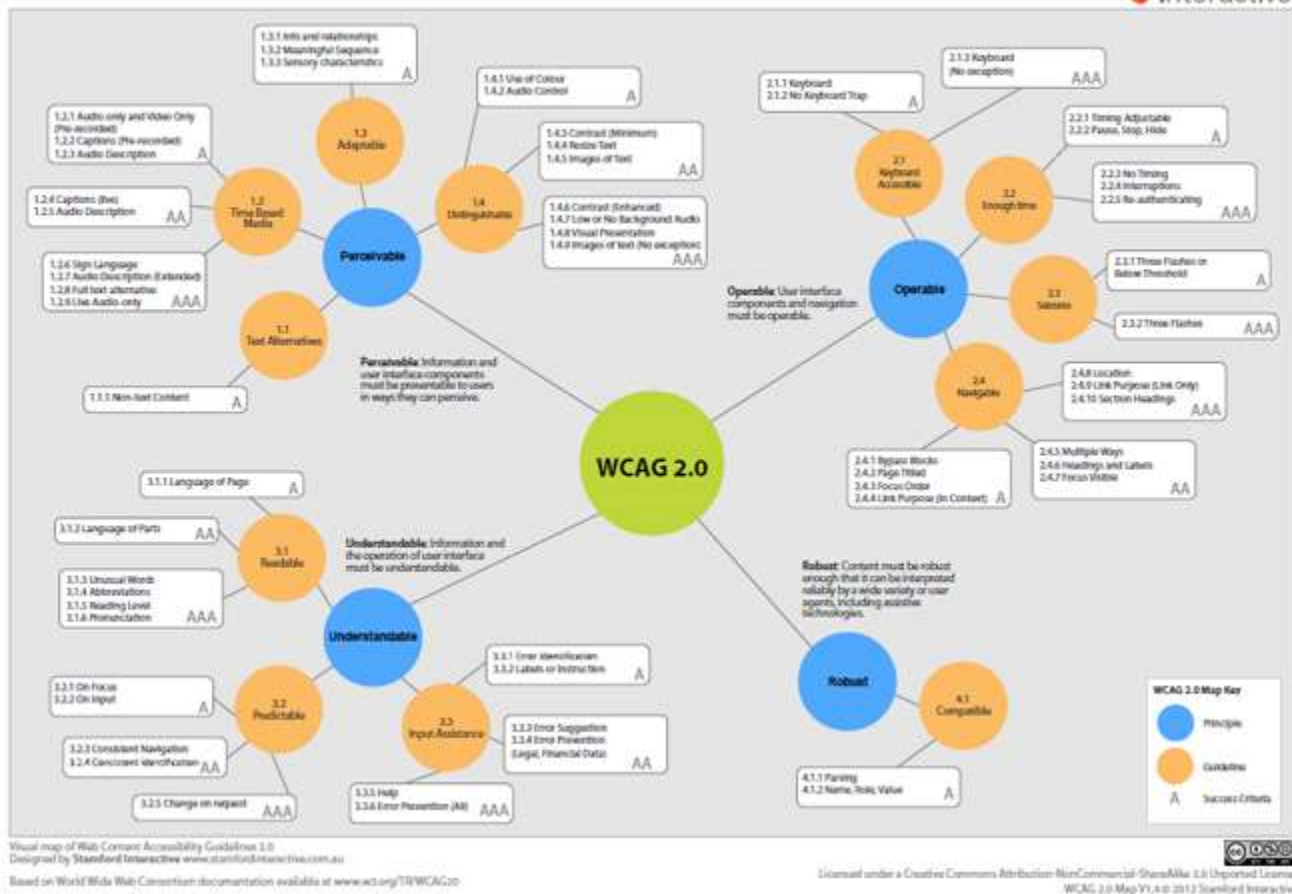
Cada criterio de conformidad tiene además dos enlaces:

- "Comprender...": enlaza con la página de información asociada a ese criterio en el documento "Comprender las WCAG 2.0" (Understanding WCAG 2.0) donde se explica el criterio, los usuarios a los que beneficia, se listan ejemplos y se incluyen los principales errores asociados a ese criterio. Se indican también una serie de técnicas informativas para resolver el criterio de conformidad que se pueden dividir en dos categorías:
 - las que son "suficientes" para resolver los criterios de conformidad. La mayoría de los criterios tienen varias técnicas o conjunto de técnicas disponibles. Para cumplir con el criterio de conformidad se puede usar cualquiera de estas técnicas o conjunto de técnicas que sean aplicables a tu contenido y a la tecnología que estás usando, sin necesidad de aplicarlas todas.
 - las que son "recomendables" y permiten conferir a la página un mayor grado de accesibilidad. Pero no son suficientes para cumplir con el criterio de conformidad, porque no pueden ser verificadas o porque son técnicas apropiadas para determinadas circunstancias pero no son efectivas en otras.
- "Cómo cumplir...": enlace al documento "Cómo cumplir con las WCAG 2.0" (How to meet WCAG 2.0), la guía rápida donde se listan SÓLO las técnicas suficientes y recomendables así como los errores asociados al criterio

Las técnicas son SÓLO informativas, no normativas, pues el criterio de conformidad podría resolverse de otro modo no documentado en las técnicas o podría haber técnicas aplicables a una tecnología todavía no documentada. Por eso estos documentos no son normativos y están en constante evolución. Según se vayan descubriendo nuevas técnicas, con el avance de las tecnologías web y productos de apoyo, se podrán ir documentando y añadiendo a la lista ya existente.

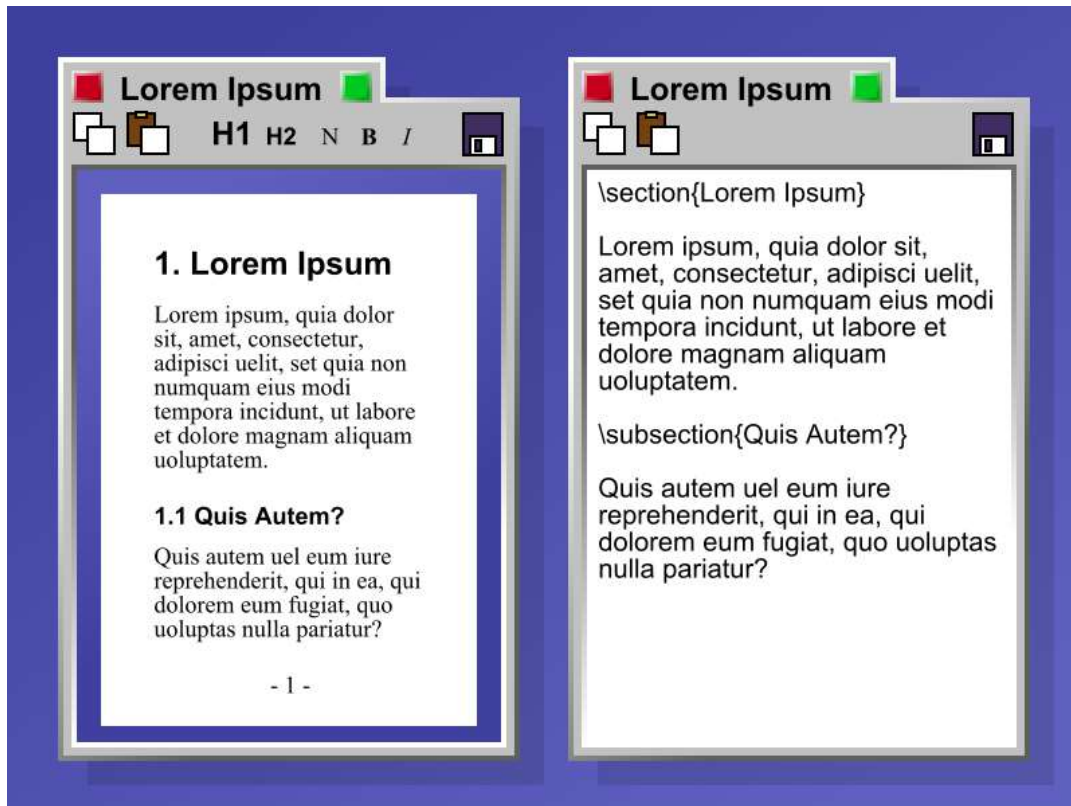
Por ejemplo, actualmente se está TRABAJANDO en las Techniques/HTML5 o en las Mobile WCAG Techniques.

WCAG 2.0 Map



5.- WYSIWYG

Es el acrónimo de What You See Is What You Get (en español, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.



El programa de la izquierda usa un editor WYSIWYG para producir un documento. El programa de la derecha contiene el código LaTeX, que cuando se compile producirá un documento que se verá muy similar al documento de la izquierda. Compilar código de formato no es un proceso WYSIWYG.

Ejemplos de editores HTML tipo WYSIWYG son: Dreamweaver, NVU/Kompozer, las versiones de Composer de Netscape y Mozilla, Amaya, Writer (de OpenOffice.org), Adobe Golive, Frontpage y Microsoft Word. También existen editores que se pueden integrar en formularios de páginas web como FCKeditor, TinyMCE, FreeRichTextEditor.

En el área de diseño web existen también herramientas WYSIWYG, dentro de los llamados CMS (Content Management System); aunque un CMS no tiene porqué ser WYSIWYG, es decir puede ser un gestor de contenidos cuyo panel de gestión no se corresponda con el diseño final, sino que en este caso se utiliza un back-end o panel de gestión para crear-modificar los contenidos, que serán reflejados en el front-end o parte final que ven los usuarios. En el grupo de los CMS WYSIWYG hay varias aplicaciones, incluso alguna de ellas con tecnología flash, como por ejemplo Easy Site Manager.

En el entorno servidor serían:

- **Visual Studio** es la IDE de Microsoft. (Visual Basic,...)
- **NetBeans** es la IDE de Oracle (Java, XML, HTML, PHP, Groovy, JavaDoc, JavaScript y JSP)
Inicialmente fue de Sun Microsystems
- **Eclipse** es la IDE de una Fundación (Java, c / C++, PHP)

6.- Características de una aplicación, de principio a fin.

Esta serie de principio a fin te ayuda a agregar características a tu aplicación para atraer a los clientes y crear el tipo de experiencias que llaman la atención de los usuarios y les encantan.

1. Comienzo: aplicación básica

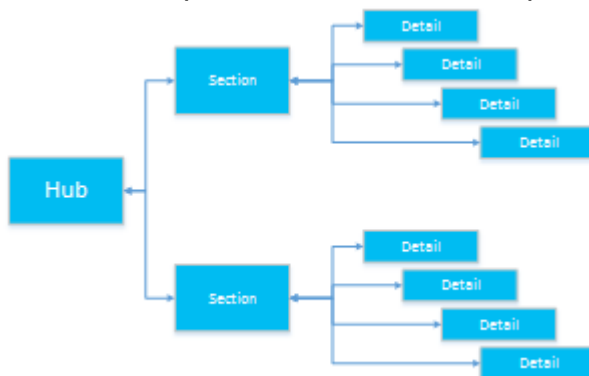
- Navegación plana

Barra de la aplicación, estado de la aplicación, configuración y ayuda.



- Navegación jerárquica

Barra de la aplicación, estado de la aplicación, configuración y ayuda.



- Conceptos básicos de la interfaz de usuario de la aplicación
Agrega contenido y controles a tu aplicación con un diseño dinámico que resulte fácil de usar y navegar en cualquier dispositivo o tamaño de pantalla.

Estado, datos y almacenamiento de la aplicación

- Estado y datos de la aplicación
Guarda y restaura los datos y el estado de la aplicación para que los usuarios puedan retomar una tarea en el lugar donde la dejaron, en cualquier momento y en cualquier dispositivo.
- Administración de archivos
Agrega a tu aplicación funcionalidades de administración de archivos tales como usar selectores de archivos y carpetas, enumerar los archivos de una carpeta, crear archivos, leer y escribir archivos, obtener las propiedades de un archivo y trabajar con archivos on line.
Guía de administración de archivos

Interacción del usuario

- Interacción del usuario básica
Proporciona experiencias del usuario intuitivas y coherentes en todos los dispositivos de entrada usando los comportamientos de interacción integrados de los controles de la plataforma de aplicaciones. Los controles de plataforma pueden administrar y responder a la entrada desde diferentes orígenes, entre ellos, entrada táctil, panel táctil, ratón, pluma/lápiz y teclado.

- **Interacción del usuario personalizada**
Procesa la entrada del usuario desde diferentes orígenes, tales como entrada táctil, panel táctil, mouse, pluma o lápiz y teclado, para crear una experiencia de interacción personalizada y con capacidad de respuesta.
- **Reproducción de multimedia**
Agrega características multimedia para reproducir audio y vídeo (segundo plano, con barras de menú, tamaño personalizado,...) y que, al mismo tiempo, ahorren batería y optimicen la representación.
- **Geovalla**
Define una región geográfica para que puedas enviar notificaciones a los usuarios cuando entren o salgan de esa área o estén cerca de familiares o amigos. Con las geovallas, la aplicación puede enviar un recordatorio a un usuario que sale del trabajo o llega a casa. Si configuras una serie de geovallas, podrías mostrar cupones cada vez que el usuario esté cerca de una cafetería. Con servicios en la nube como Servicios móviles de Azure, también podrías crear una aplicación que te notifique cuando tus amigos estén cerca.

EJERCICIO: Enuncia ejemplos de aplicaciones de lo expuesto.

7.- La forma de diseñar el interfaz de usuario

Dependerá de los objetivos que le inspiren, generalmente orientado a favorecer la facilidad de uso del sistema o programa. Los interfaces de usuario se convierten en unas poderosas herramientas de navegación, pero hay que tener presente algunas premisas:

Si la IU está bien diseñada, el usuario encontrará la respuesta que espera a su acción. Si no es así puede ser frustrante su operación, ya que el usuario habitualmente tiende a culparse a sí mismo por no saber usar el objeto.

Los programas son usados por usuarios con distintos niveles de conocimientos, desde principiantes hasta expertos. Es por ello que no existe una interfaz válida para todos los usuarios y todas las tareas. Debe permitirse libertad al usuario para que elija el modo de interacción que más se adecúe a sus objetivos en cada momento. La mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

Existen tres puntos de vista distintos en una IU: el del usuario, el del programador y el del diseñador (analogía de la construcción de una casa). Cada uno tiene un modelo mental propio de la interfaz, que contiene los conceptos y expectativas acerca de la misma, desarrollados a través de su experiencia.

El modelo permite explicar o predecir comportamientos del sistema y tomar las decisiones adecuadas para modificar el mismo. Los modelos subyacen en la interacción con las computadoras, de ahí su importancia.

- **Modelo del usuario:** El usuario tiene su visión personal del sistema, y espera que éste se comporte de una cierta forma. Se puede conocer el modelo del usuario estudiándolo, ya sea realizando tests de usabilidad, entrevistas, o a través de una realimentación. Una interfaz debe facilitar el proceso de crear un modelo mental efectivo.
- **Modelo del diseñador:** El diseñador mezcla las necesidades, ideas, deseos del usuario y los materiales de que dispone el programador para diseñar un producto de software. Es un intermediario entre ambos.

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

- La presentación es lo que primero capta la atención del usuario, pero más tarde pasa a un segundo plano, y adquiere más importancia la interacción con el producto para poder satisfacer sus expectativas. La presentación no es lo más relevante y un abuso en la misma (por ejemplo, en el color) puede ser contraproducente, distrayendo al usuario.
- La segunda parte del modelo define las técnicas de interacción del usuario, a través de diversos dispositivos.
- La tercera es la más importante, y es donde el diseñador determina la metáfora adecuada que encaja con el modelo mental del usuario. El modelo debe comenzar por esta parte e ir hacia arriba. Una vez definida la metáfora y los objetos del interfaz, los aspectos visuales saldrán de una manera lógica y fácil.

- **Modelo del programador:** Es el más fácil de visualizar, al poderse especificar formalmente. Está constituido por los objetos que manipula el programador, distintos de los que trata el usuario (ejemplo: el programador llama “base de datos” a lo que el usuario podría llamar “agenda”). Estos objetos deben esconderse del usuario. Los conocimientos del programador incluyen la plataforma de desarrollo, el sistema operativo, las herramientas de desarrollo y especificaciones. Sin embargo, esto no significa necesariamente que tenga la habilidad de proporcionar al usuario los modelos y metáforas más adecuadas.

A los usuarios les gusta poseer el control en el ordenador, y manipular directamente los objetos en pantalla.

- Indicar claramente cómo arrastrar, editar o hacer clic sobre los objetos.
- Proporcionar ayuda visual cuando el usuario manipule algo.
- Si un usuario inicia un proceso que tardará unos cuantos segundos, se indicará visualmente (reloj) que el sistema está ocupado.
- Diseñar interfaces consistentes e intuitivos. Mantener la misma ubicación de los botones o barras de herramientas y dotarles de las mismas funciones en todas las pantallas, para evitar la confusión del usuario.
- Permitir que el usuario pueda personalizar el interfaz, que cometa errores y pueda deshacer sus acciones.
- Proporcionar hiperenlaces de texto descriptivo.
- En definitiva, el usuario debería ser capaz de responder en cada pantalla a las siguientes preguntas: ¿ dónde estoy? ¿ a dónde puedo ir? ¿ cómo llego allí?.

A pesar de todo, estas recomendaciones aquí expuestas no garantizan que se cubran todas las expectativas de los usuarios, pero en alguna medida pueden contribuir a solventar y mejorar las deficiencias en el diseño de los programas hipermedia, su capacidad de interacción y navegación en función de los potenciales usuarios.

El diseño de recursos hipermediales, bien sea en soporte CD-Rom o WWW, debe tener presente algunos aspectos:

Ofrecer una amplia oferta de información. Dotar de una gran libertad de movimientos. Sin embargo, esa flexibilidad por un lado se puede favorecer la autonomía en el usuario, y por otro, se corre el riesgo de que la navegación indiscriminada a través de la información no relevante aumente la carga cognitiva que supone una navegación compleja. De ahí que se precise de programas que faciliten el desarrollo de habilidades y destrezas para una navegación selectiva en función de lo que se entiende como información significativa. En definitiva, habrá que determinar cuáles son los mecanismos y procesos necesarios para favorecer la navegación conceptual a través del entorno específico del que se trate.

Diseño Orientado a Objetos.pdf

9.- IDE - Entorno de Desarrollo Interactivo.

Un ambiente de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen auto-completado inteligente de código (IntelliSense). Algunos IDE contienen un compilador, un intérprete, o ambos, tales como **NetBeans y Eclipse**; otros no, tales como SharpDevelop y Lazarus.

El límite entre un IDE y otras partes del entorno de desarrollo de software más amplio no está bien definido. Muchas veces, a los efectos de simplificar la construcción de la interfaz gráfica de usuario (GUI, por sus siglas en inglés) se integran un sistema controlador de versión y varias herramientas. Muchos IDE modernos también cuentan con un navegador de clases, un buscador de objetos y un diagrama de jerarquía de clases, para su uso con el desarrollo de software orientado a objetos.

Los IDE están diseñados para maximizar la productividad del programador proporcionando componentes muy unidos con interfaces de usuario similares. Los IDE presentan un único programa en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de software.

Algunos IDE están dedicados específicamente a un lenguaje de programación, permitiendo que las características sean lo más cercanas al paradigma de programación de dicho lenguaje. Por otro lado, existen muchos IDE de múltiples lenguajes tales como [Eclipse](#), ActiveState Komodo, IntelliJ IDEA, MyEclipse, Oracle JDeveloper, [NetBeans](#), Codenvy y [Microsoft Visual Studio](#). [Xcode](#), Xojo y [Delphi](#) están dedicados a un lenguaje cerrado o a un tipo de ajustes de tipos de lenguajes de programación.

Algunos IDE soportan múltiples lenguajes, tales como:

- [GNU Emacs](#) basados en **C**
- [Emacs Lisp](#), y [Eclipse](#), IntelliJ IDEA, MyEclipse o [NetBeans](#), todos basados en **Java**
- [MonoDevelop](#), basados en **C#**.

Normalmente, el soporte para lenguajes alternativos regularmente es proveído por un *plug-in*, permitiéndoles ser instalados en el mismo IDE, al mismo tiempo.

[Eclipse](#), y [Netbeans](#) tienen plugins para **C / C++**, **Ada**, (por ejemplo AdaGIDE), **Perl**, **Python**, **Ruby**, y **PHP**, los cuales son seleccionados entre extensión de archivos, ambientes o ajustes de proyectos.

Características entre diferentes plataformas computacionales.

Las herramientas GNU de software libre ([GNU Compiler Collection](#) (GCC), [depurador GNU](#) (gdb), [GNU make](#)) están disponibles en muchas plataformas, incluyendo Windows. Los desarrolladores que prefieren herramientas orientadas a la línea de comandos pueden utilizar los editores con soporte para muchos estándares de Unix y herramientas de construcción GNU, construyendo una IDE con programas como Emacs o Vim.

Con el advenimiento de la computación en nube, algunos IDE están disponibles en línea y se ejecutan dentro de los navegadores web. Desarrolladores de páginas web entorno cliente (HTML + JAVACRIPT + CSS)

User Interface Toolkits

JavaFX - AWT - **Swing** - Java 2D - Accessibility - Drag and Drop - Input Methods
Image I/O - Print Service - Sound

SWING, implementa un conjunto de componentes para la construcción de interfaces gráficas de usuario (GUI) y la adición de una rica funcionalidad de gráficos e interactividad a las aplicaciones Java. Los componentes Swing se implementan en su totalidad en el lenguaje de programación Java. La apariencia y la sensación incorporable le permite crear interfaces gráficas de usuario que pueden parecerse a las mismas plataformas o pueden asumir la apariencia de la plataforma de sistema operativo actual (como Microsoft Windows, Solaris TM o Linux).

<http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784>

JavaFX es un conjunto de gráficos y paquetes multimedia que permite a los desarrolladores diseñar, crear, probar, depurar y desplegar **aplicaciones de cliente enriquecido** que operan constantemente a través de diversas plataformas.

<http://docs.oracle.com/javase/8/docs/>

Oracle tiene dos productos que implementan **Java Platform Standard Edition (Java SE) 8**: Java SE Development Kit (JDK) 8 y Java SE Runtime Environment (JRE) 8.

JDK 8 es un superconjunto de JRE 8, y contiene todo lo que está en JRE 8, además de herramientas como los compiladores y depuradores necesarios para el desarrollo de applets y aplicaciones. JRE 8 proporciona las bibliotecas, la máquina virtual de Java (JVM) y otros componentes para ejecutar applets y aplicaciones escritas en el lenguaje de programación Java. JRE incluye componentes no requeridos por la especificación Java SE, incluyendo ambos componentes Java estándar y no estándar.

El siguiente diagrama conceptual ilustra los componentes de Java SE de Oracle:

[Description of Java Conceptual Diagram](#)

