Interfaces gráficos

Área de Ingeniería Telemática

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la “bestia”
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Línea de comandos (CLI – Command Line Interface)

- El diseñador del interfaz hace poco más que permitir al usuario hacer lo que quiera
- El usuario debe memorizar los comandos
- Una vez conocido es fácil de utilizar, pero no es fácil de aprender a utilizar
- Pregunta-respuesta:
  - No hace falta memorizar opciones, el programa pregunta por cada una de ellas
  - No puedo cambiar el orden.
  - Ejemplo: especificar nombre de fichero

Satisfacción/frustración

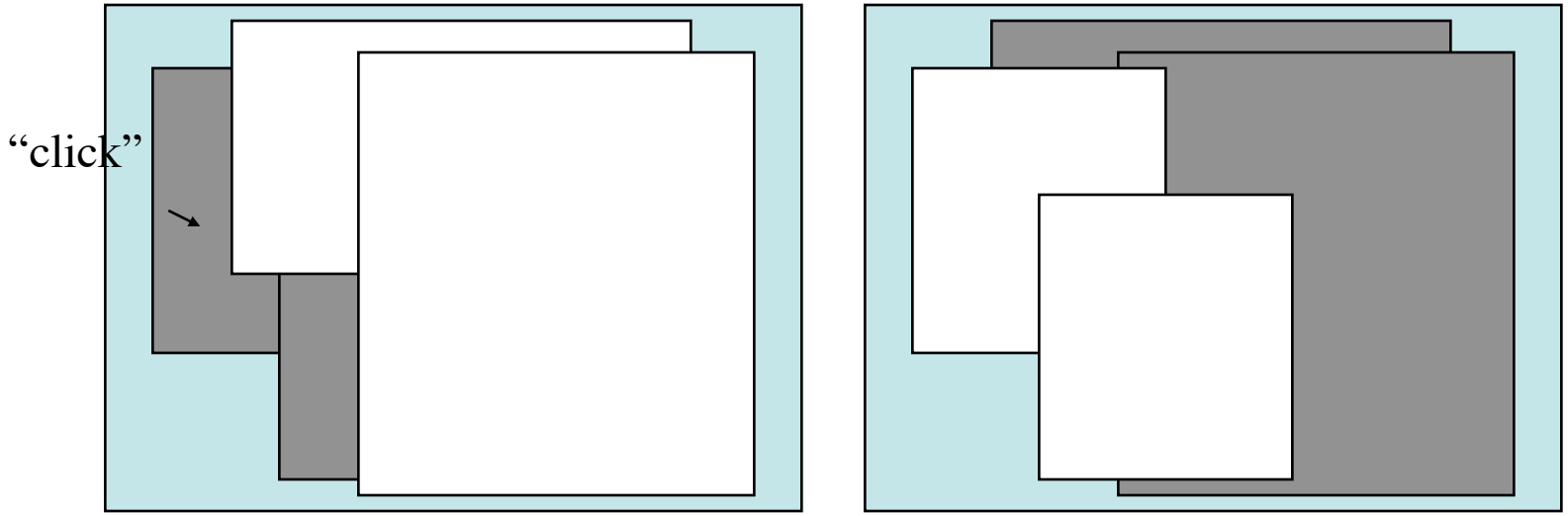
- Las pequeñas frustraciones se acumulan
- Más feliz cuanto más se sienta en control del entorno
- Desagrado cuando sucede algo que no se puede controlar
- Un interfaz de usuario está bien diseñado cuando el programa se comporta exactamente como el usuario espera que se comporte
- El usuario no quiere pensar en cómo funciona. El interfaz será mejor cuanto menos haga pensar al usuario.

Modelos

- El usuario no es una *tábula rasa*. Al empezar a utilizar un programa ya cuenta con unas expectativas de cómo funciona
- Esto es el ***modelo del usuario***
- Cómo funciona el programa es el ***modelo del programa***
- Un interfaz de usuario está bien diseñado cuando el modelo del programa se ajusta al modelo del usuario
- Debemos ajustar el modelo del programa al modelo del usuario
- Ejemplos:
  - Documentos en blanco
  - Editores de páginas HTML vs editores de texto.

¿Cómo es el modelo del usuario?

- Pregúntale a él/ellos
- Opiniones diferentes
- Habrá una opinión mayoritaria
- Basta con probarlo con unos pocos usuarios
- El usuario asume el modelo más simple posible
- Ejemplo:
  - Cambio de aplicación activa con aplicaciones con varias ventanas



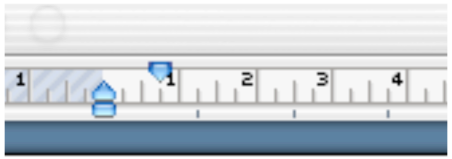
Opciones

- Cada vez que se ofrece una opción se le está pidiendo al usuario que tome una decisión
- El problema está en pedir decisiones sobre algo que al usuario no le importa. Esas decisiones son tarea del diseñador
- Al usuario le interesan las decisiones relacionadas con la tarea que desea completar
- Muchas veces no se saben tomar las decisiones en el diseño y el resultado son cuadros de Opciones o Preferencias plagados de alternativas
- Ejemplos:
  - Búsqueda de ayuda en Windows
  - Toolbars y menú flotante
  - Cambios de aspecto sin cambios de comportamiento.

Metáforas

- Cuando el modelo de usuario está incompleto el programa puede emplear metáforas para enseñarle su modelo
- Enseñarle al usuario cómo funciona el programa dándole pistas mediante metáforas
- Deben comportarse de forma predecible, como los objetos de la vida real, si no sólo añaden confusión
- Ejemplos (buenos y malos):
  - Escritorio
  - Zoom
  - Papelera
  - Marca de sangrado
  - Carpetas dentro de carpetas.





Elementos autoexplicativos

- Ejemplos:
  - Chapa de empujar para abrir puerta
  - Cajas de CDs y DVDs
  - Botones 3D
  - Cuadro de opciones de preferencias del Netscape vs pestañas...

Consistencia vs creatividad

- Consistencia entre diferentes programas los hace más fáciles de utilizar para el usuario
- Básicamente el modelo del programa se adapta al modelo aprendido del usuario
- Empleada inteligentemente nos permite evitar "reinventar la rueda"

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Usabilidad

- Usabilidad: aceptación de un sistema web por parte de los usuarios

- Los usuarios visitan y vuelven a un website si:
  - Encuentran fácilmente la información útil
  - La estructura de la información permite la navegación y el acceso
  - La estructura de la presentación es adecuada

- La calidad de las aplicaciones es un factor muy importante.

Usabilidad

- Hay normativa (alguna estandarizada y otra aceptada de facto):
  - ISO 9241: La usabilidad es la mezcla de 3 propiedades:
    - Efectividad (*effectiveness*): con qué precisión y totalidad logran sus objetivos los usuarios
    - Eficiencia (*efficiency*): los recursos usados para lograr los objetivos
      - Tiempo
      - Número de *Clicks*
      - Número de transiciones entre páginas
    - Satisfacción *(satisfaction)*: la comodidad y aceptación del uso

Usabilidad

- Jakob Nielsen (guru de la usabilidad)
  - Facilidad de aprendizaje (*learnability*): facilidad para entender el funcionamiento y comportamiento del sistema.
  - Eficiencia (*efficiency*): provecho que le puede sacar el usuario una vez aprendido a usarlo.
  - Facilidad para memorizarlo (*memorability*): facilidad para recordar el funcionamiento del sistema
  - Gestión de errores (*few errors*):
    - Capacidad para tener una tasa de error pequeña
    - Si existen errores el sistema tiene que ser robusto para darles respuesta adecuada a estos
    - Capacidad de recuperación de errores
  - Satisfacción de los usuarios (*satisfaction*): cuanto de agradable les resulta la navegación a los usuarios
    http://www.useit.com/alertbox/20030825.html

La usabilidad informática en las películas

1. El héroe puede usar inmediatamente cualquier UI
2. Los viajeros del tiempo pueden usar diseños actuales
3. Las UI 3D
4. La integración es fácil, interoperatibidad de datos
5. Acceso denegado / Acceso concedido
6. Fuentes grandes
7. La computadora habladora de Star Trek
8. Controladores remotos
9. Tienes un email es siempre una buena noticia
10. "Esto es Unix, es fácil"

http://www.useit.com/alertbox/film-ui-bloopers.html

Pautas de Shneiderman

1. Solidez del diseño
   - Estructura fuerte y sólida para los contenidos del menú, la maquetación y la terminología

2. Poner atajos (*short-cuts*)
   - Para minimizar y agilizar número de interacciones

3. Feedback
   - Mostrar avisos según la acción

4. Al hacer acciones dar información
   - Dar información de que se ha logrado el objetivo para que el usuario tenga sensación de tranquilidad

5. Gestión de errores robusta
   1. Intentar que el usuario no cometa errores graves y si suceden que existan mecanismos para recuperarse de él

Pautas de Shneiderman

6. Opción de anular acciones
   - Ofrecer la opción "Deshacer"

7. Control del sistema
   - Tiene que dar la sensación de que el usuario controla al sistema, no que el sistema controla al usuario. El sistema debe funcionar como consecuencia de la acciones del usuario.

8. Reducción de carga de la memoria a corto plazo
   - **Memoria a corto plazo**: "sistema de nuestra memoria que maneja la información para interactuar con el entorno". Si no se repasa puede usar 7±2 elementos a la vez.
   - Por tanto, usar poco controles y textos simples.

Heurísticos de Nielsen
(10 pautas para la homepage)

1. Visibilidad del estado del Sistema
2. Adecuación entre el sistema y el mundo real
   - Utilizar un lenguaje apto para usuarios poco especializados
3. Control y libertad del usuario
4. Consistencia y estándares
5. Prevención de errores
6. Reconocer mejor que recordar
7. Flexibilidad y eficiencia de uso
8. Estética y diseño minimalista
9. Ayudar a los usuarios a reconocer, diagnosticar y solucionar errores
10. Ayuda y documentación

http://www.useit.com/papers/heuristic/heuristic_list.html

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Accesibilidad

- Garantizar que todos los usuarios pueden usar y acceder al sistema web. Este acceso tiene que ser independiente de:
  - Discapacidades individuales (permanentes o temporales)
    - Físicas
    - Sensoriales
    - Cognitivas
  - Del uso contextual
    - Dispositivo de acceso: PC, teléfono móvil, PDA...
    - Conexión
    - Situación: conduciendo coche, medio ruidoso...
    - Software

Accesibilidad

- Algunos autores basados en esta definición consideran la accesibilidad un atributo de la usabilidad, pero no es así:
  - Un website usable puede no ser accesible
  - Un website accesible puede no ser usable

- Eso si, la usabilidad y la accesibilidad están muy relacionadas

Accesibilidad

- Está mucho más regulada que la usabilidad
- Existe una ley a nivel nacional:
  - LSSICE (Ley de Servicios de la Sociedad de la Información y del Comercio Electrónico): http://www.lssi.es/

Disposición adicional quinta. Accesibilidad para las personas con discapacidad y de edad avanzada a la información proporcionada por medios electrónicos.

Uno. Las Administraciones públicas adoptarán las medidas necesarias para que la información disponible en sus respectivas páginas de Internet pueda ser accesible a personas con discapacidad y de edad avanzada de acuerdo con los criterios de accesibilidad al contenido generalmente reconocidos antes del 31 de diciembre de 2005.

Asimismo, podrán exigir que las páginas de Internet cuyo diseño o mantenimiento financien apliquen los criterios de accesibilidad antes mencionados.

Dos. Igualmente, se promoverá la adopción de normas de accesibilidad por los prestadores de servicios y los fabricantes de equipos y <<software >>, para facilitar el acceso de las personas con discapacidad o de edad avanzada a los contenidos digitales.

- Generalmente se entiende como que tiene que aplicar la normativa WCAG (prioridad 2) del grupo de trabajo WAI del W3C.

WAI y WCAG

- Web Accessibility Initiative: http://www.w3.org/WAI/
  - Una de las iniciativas y ambitos de trabajo del W3C
  - Se dedica a promover la accesibilidad
    - Desarrollando materiales, guía, reglas, …
- Web Content Accessibility Guidelines: http://www.w3.org/WAI/intro/wcag.php
  - Grupo específico de trabajo de WAI
  - Actualmente en vigor las reglas WACG 1.0
  - En desarrollo las WACG 2.0
    - Más estrictas
    - Más concretas y detalladas
    - Menos ambiguas

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Pruebas de usabilidad

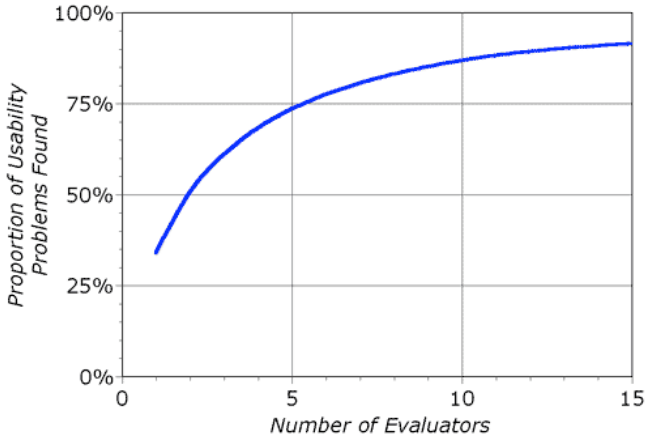
- Seguir las directrices de usabilidad no asegura la usabilidad del producto final => necesario hacer pruebas con usuarios.
- No son tan importantes los resultados estadísticos como detectar fallos en el interfaz
- Las evaluaciones de usabilidad tiene 3 objetivos:
  - Medir la funcionalidad de la aplicación
  - Estudiar la influencia de las interfaces en el usuario
  - Identificar problemas imprevistos
- Generalmente al hacer las pruebas el software no está terminado, lo cual genera sus propios problemas

Pruebas de usabilidad

- 2 opciones:
  - Con usuarios "normales"
    - Hacen falta muchos usuarios => caro
    - Es el método más eficaz => se encuentran todos los problemas
    - El entorno no es el habitual del usuario y además se siente "examinado"
    - Gente falla el test por no saber ni por dónde empezar
    - Por ello se inventaron los wizards o los asistentes:
      - No resuelven el problema de usabilidad, lo enmascaran
      - Son como las preguntas de los programas antiguos pero hacen que tenga éxito el test
      - Si el usuario quiere hacer algo diferente vuelve el problema
      - Ejemplo: balloon sobre el botón de inicio en windows

Pruebas de usabilidad

- Con "expertos" en usabilidad
  - Se hace con pocos "expertos" => barato
  - Con 5 se sacan el 75% de los problemas que verían los usuarios.
  - Saben lo que tienen que hacer
  - Cada "experto" hace un informe y luego se juntan.



Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Diseñar para situaciones extremas

- ¿Entiende y puede usar el interfaz un experimentado informático?
- ¿Entiende y puede usar el interfaz un anciano con vista cansada y parkinson?
- Diseñándolo para casos extremos será más sencillo de utilizar en situaciones normales

El usuario no sabe leer

- Puede que el usuario no tenga el manual
- Aunque tuviera el manual, el usuario no lo leería
- De hecho tampoco va a leer lo que aparezca en el interfaz si puede hacer lo que quiere sin leerlo
- Si tiene que leerlo para hacer lo que quiere será una molestia
- Puede que incluso en ese caso no lo lea y deje de intentar hacer lo que quería o cambie de programa
- Cuanto más aparezca para leer menos probabilidades hay de que lo lea
  - Los usuarios expertos asumen que saben lo que se les intenta explicar y no tienen tiempo para leerlo
  - Los inexpertos se saltan las instrucciones esperando que las selecciones por defecto les valgan
  - Lo pocos inexpertos que lo leen pueden acabar confundidos por los conceptos que se les intentan explicar
- Ejemplo: usuario intimidado por los ordenadores que intenta salir de un programa.

El usuario no sabe usar el ratón

- Una barra de menú en la parte superior de una ventana tiene menos de 1cm de anchura
- Una barra de menú en la parte superior del escritorio tiene menos de 1cm de anchura pero "una milla" de altura
- ¿Cuáles son los 5 puntos de la pantalla más fáciles de acertar con el ratón?...
  - Las 4 esquinas
  - El punto donde ya está el ratón
- Ejemplo: El botón de *Inicio* de los Windows

El usuario no sabe usar el ratón

- Los usuarios no saben usar el ratón:
  - Puede que empleen malos dispositivos: trackpad, trackball
  - Las condiciones no sean óptimas: bola sucia, mala alfombra para óptico, en un tren que se mueve
  - El usuario inexperto con ordenadores no sabe utilizar un ratón
  - Hay personas con problemas motrices
  - Doble-click sin mover el ratón es complicado para muchas personas
  - En general tener que mover con cuidado el ratón requiere energía y concentración que el usuario prefiere emplear en realizar mejor su tarea

El usuario no recuerda

- Ejemplo: nombre del fichero a abrir
- Los menús sustituyen a los comandos de CLI, ya no hay que memorizarlos.

Relatividad temporal

- Los días son segundos:
  - Partes de un interfaz que se tarda días en desarrollar solo serán experimentados por el usuario durante unos segundos
  - El programador pasa mucho más tiempo delante del interfaz. Lo que para él es evidente no lo será para alguien que solo está unos segundos ante él
- Los meses son minutos:
  - En otra escala, el desarrollo de un software completo lleva meses o años
  - Durante ese tiempo se desarrollan muchos conceptos y se incluyen muchas opciones en el programa
  - El resultado final puede ser un programa con una gran cantidad de opciones que el usuario debe aprender a usar en muy poco tiempo
  - Los mejores diseños suelen tener el menor número de opciones
- Los segundos son horas:
  - Si el programa parece lento los usuarios sienten que pierden el control
  - Trucos:
    - Responder inmediatamente a la acción del usuario. Aunque no se complete que sepa que está en curso
    - Dividir el procesamiento en pequeños bloques de forma que sean despreciables para el usuario
    - Unir los procesamientos en un gran bloque que permita al usuario cambiar de actividad (Ejemplo: preguntar todo lo necesario antes de empezar instalación lenta, no preguntas en medio).

Heurísticos

- Se intenta adivinar lo que el usuario quiere hacer
- El programa *apuesta* por lo que considera más probable
- Ejemplo: se empieza a escribir el nombre de un fichero y lo completa
- Ejemplo: :-) en word
- Se pueden volver molestos con facilidad
- Problema: cada vez que el heurístico falla crea más desagrado en el usuario que muchos aciertos
- ¿Cuánto? Depende de lo que cueste deshacer la acción del heurístico
- Un buen heurístico es obvio cómo adivina, fácil de deshacer y acierta con una alta probabilidad.

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

La Web

- Se diseñó para ir mostrando documentos uno a uno

- Un interfaz con mucha interactividad con procesamiento en el servidor se enfrenta al *lag*
  - El tiempo que transcurre entre que el usuario realiza una acción y obtiene la respuesta es largo por culpa de la red
  - Ejemplo: webmail, borrar correos uno a uno
  - En la web cada click del usuario implica al menos un RTT lo cual reduce la usabilidad. Se debe diseñar para minimizar el número de RTTs de espera experimentados

La Web

- No dispone de muchos elementos de UIs
  - Ejemplo: no hay menús, si se implementa con JavaScript hay problemas de compatibilidad y si se hacen con <select> se supone que seleccionar la opción no debería generar un cambio de URL (no lo espera el usuario)
  - Ejemplo: no se puede sacar un cuadro de diálogo. Sacar otra ventana no es igual y se confunde con publicidad
  - Ejemplo: no hay un buen campo de edición de texto. <textarea> no soporta grabar por si se cierra la ventana

- Un problema de usabilidad es muy grave porque la competencia "está a un *click* de distancia".

Modelo del usuario de la web

- Logo en la esquina superior izquierda
- Click en él lleva a la *home page*
- Los enlaces son azules y subrayados y todo lo que es azul y subrayado es un enlace
- Los botones parecen botones
- El usuario trae ya aprendido el modelo de la prensa impresa: encabezados, secciones, etc.

Comportamiento del usuario

- El usuario no lee las páginas, las escanea, busca el enlace interesante y lo pulsa
- No ve todas las opciones y escoge la mejor sino que escoge la primera *razonable* que ve
  - Puede que tenga prisa
  - Si se equivoca puede volver atrás
  - Es más satisfactorio
- Muchas veces en realidad no entiende cómo funcionan las cosas sino que se hace su propia idea de cómo funcionan. Al menos mientras le sirva. Ejemplo: creer que google o AOL son Internet.
- Seguro que el usuario usa el interfaz de alguna manera en la que no habías pensado inicialmente:
  - Plantear todos los escenarios posibles/imaginables

Navegación

- Es muy importante que el usuario vea con facilidad cómo navegar por el sitio web
- En comparación con el mundo real el usuario no tiene información espacial de dónde está. Se le debe indicar con claridad (logo, sección). Ejemplo: breadcrumbs

CNET > Downloads > Mac > **Multimedia & Design**

- El usuario no puede saber cuánto le queda por explorar, no hay sentido de la escala. Debe poder ver las opciones
- Puede llegar hasta ahí de muy diferentes formas, en vez de siguiendo un camino jerárquico puede ir con un enlace directo.

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Top Ten Web Design Mistakes of 2005 (Jakob Nielsen)

- Problemas de legibilidad
- Enlaces alejados de su formato estándar
- Flash
- Contenido no escrito para la web
- Búsquedas deficientes
- Incompatibilidades entre navegadores
- Formularios incómodos
- Ausencia de vías de contacto con los responsables del sitio web
- Maquetación con ancho fijo
- Ampliación inadecuada de las imágenes

http://www.useit.com/alertbox/designmistakes.html

Top Ten Web Design Mistakes of 2007 (Jakob Nielsen)

- Malos buscadores
- Ficheros PDF para lectura online
- No cambiar el color de enlaces visitados
- Texto no escaneable
- Tamaño de fuentes fijo
- Títulos de páginas con poca visibilidad para buscadores
- Cualquier cosa que se parezca a un anuncio
- Violar convenciones de diseño
- Abrir ventanas nuevas
- No contestar las preguntas de los usuarios

http://www.useit.com/alertbox/9605.html

8 problemas que no han cambiado desde 1997 (Jakob Nielsen)

- Links que no cambian de color una vez han sido visitados
- Romper la funcionalidad del botón de volver del navegador
- Abrir nuevas ventanas en el navegador
- Utilizar pop-ups o ventanas emergentes
- Realizar diseños de contenido que parecen banners
- Violar convenciones de la web
- Contenido poco relevante
- Contenido denso y no escaneable (ojeable)

http://www.webmonkey.com/06/24/index4a.html

Vuelta a lo básico (Jakob Nielsen)

- Que el texto sea legible
- Que los contenidos respondan a sus expectativas
- Que los sistemas de navegación y búsqueda les ayuden a encontrar lo que buscan
- Formularios más cortos y simples
- Que no haya cosas que no funcionen, enlaces que no lleven a ninguna parte, contenido desactualizado

Contenidos

- Introducción
- Usabilidad
- Accesibilidad
- Pruebas de usabilidad
- El usuario y la interacción con la "bestia"
- El usuario y la web
- Errores, problemas y soluciones de diseño (by Jacob Nielsen)
- Resumen

Resumen

- El objetivo es poner en línea el modelo del usuario y el modelo del programa
- Debemos saber como espera el usuario que funcionen las cosas para adaptar el modelo del programa
- Podemos enseñarle el modelo del programa al usuario con metáforas
- Aunque suene duro:
  - El usuario no lee
  - El usuario no sabe usar el ratón
  - El usuario no tiene buena memoria
  - Siempre hay algún usuario que se pasa de "listo"
- Diseñar para el caso mas desfavorable hará que el interfaz sea más manejable en todos los casos => accesibilidad/WAI
- Usar estándares y programación en cliente no intrusiva para

Lecturas interesantes

- *User interface design for programmers.* Joel Spolsky
- *Don't make me think: A common sense approach to web usability.* Steve Krug
- *Designing web usability: The practice of simplicity.* Jakob Nielsen
- *Homepage Usability: 50 Websites Deconstructed.* Jakob Nielsen
- *Usabilidad: http://www.useit.com/.* Jakob Nielsen