

Diseño de Interfaces de Usuario

Diseño de soluciones

Interfaces Persoa Máquina

Marcos Ortega (m.ortega@udc.es)

Etapa de diseño

- Ya conocemos nuestra audiencia y los problemas que hay que solucionar
- Ya nos hemos formado una idea de qué tipo de soluciones podrían funcionar
- ¿Cómo empezar a crear esas ideas?
 - Modelos sencillos a más complejos
 - Evitamos errores en etapas más costosas
- Primeros modelos básicos: Prototipado en papel

Prototipado en Papel

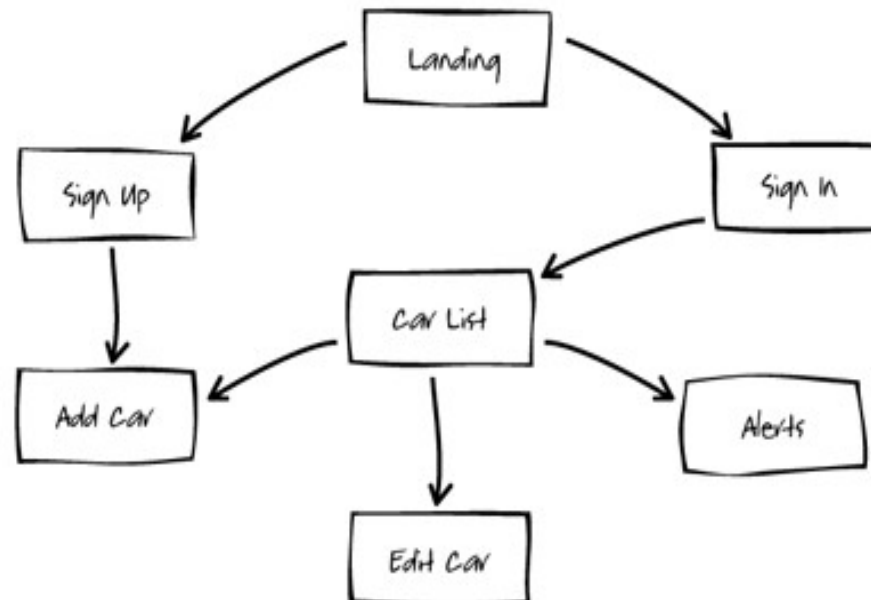
- Representaciones muy sencillas y rápidas de corregir y modificar
- Diseñando la estructura
 - Diagramas de flujo
 - Storyboards
- Diseñando pantallas
 - Wireframes
 - Mock-ups

Prototipado en Papel: Diagramas de Flujo

- Responder a la pregunta: ¿Qué tiene que hacer el usuario para conseguir lo que quiere?
- Es preferible diagramas de flujo sencillos que cubran las necesidades más relevantes

Pages

- Landing Page
- Initial Set-up / Demo
- Car Landing / Selection Page
 - Edit Car
 - Add a car
 - Input Data
- Reminders
 - Create
 - Edit
 - Dismiss



Prototipado en Papel: Storyboards

- Técnica originalmente desarrollada para planificar animaciones en películas
- Desenvuelven el camino de un usuario para una cierta acción en una serie de *snapshots*
- Se centran en los elementos de interacción
 - Es una técnica más compleja: preferible en acciones donde el diseño no sea demasiado obvio

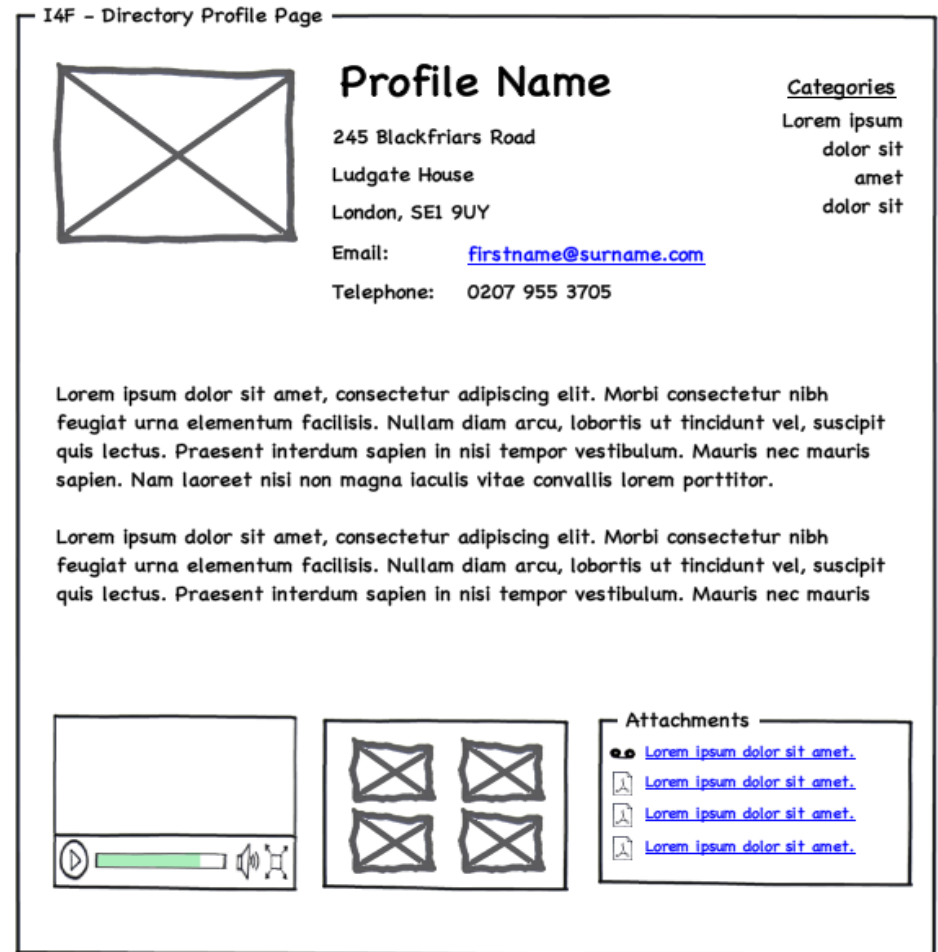
Prototipado en Papel:

Diseñar pantallas

- Una vez tenemos la estructura general, diseñamos el boceto de las pantallas individuales
- El diseño de la estructura nos permite saber ya qué funcionalidades y elementos son necesarios en cada pantalla
- Si hay guiones, pueden valer de base para el boceto
- Bocetos típicos: wireframes, mock-ups

Prototipado en Papel: Wireframes

- Representan la estructura exacta de una pantalla sin detalles de decoración (colores, sombras, imágenes,...)
- Se centran en el contenido que se quiere mostrar no en la apariencia de éste



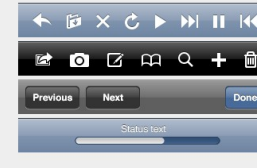
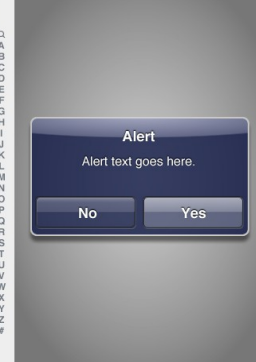
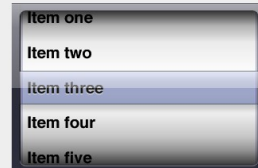
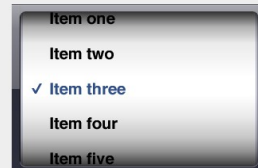
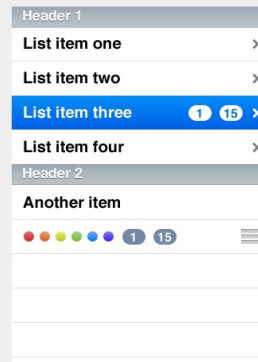
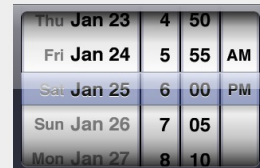
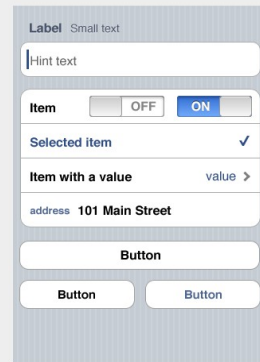
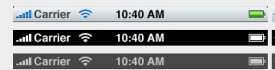
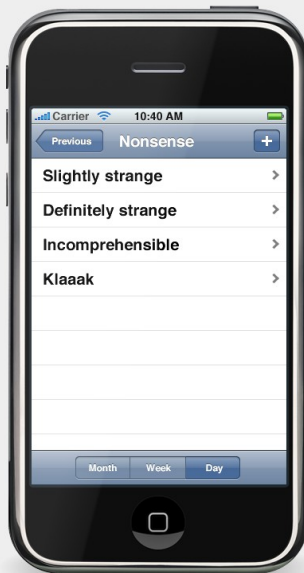
Prototipado en Papel: Mock-ups

- Los Mock-ups añaden la decoración a los wireframes
- Detalles visuales: sombras, texturas, imágenes, transparencias, etc.
- No es solo una cuestión de apariencia, también sirve para dar pistas o facilitar al usuario el acceso a funcionalidades

Prototipado en Papel: Mock-ups

iPhone mockup toolkit (CS3)
28 Jan 2009
metaspark.com (makers of Notespark)

All items have been redrawn as vectors with the exception of the picture of the iPhone device and the background of the start screen.



Prototipado en Papel: Herramientas

- Balsamiq
<http://www.balsamiq.com/products/mockups/mybalsamiq>
- Mockingbird
<https://gomockingbird.com/>
- Pencil Project
<http://pencil.evolus.vn/>
- Moqups
<https://moqups.com/>
- ...
- ¡Lápiz y papel!

Testing de prototipado en papel

- Los bocetos en papel nos permiten cambios rápidos pero ¿cómo sabemos qué cambios hacer?
- Muchas veces el simple hecho de realizar el prototipado hace que los problemas se hagan obvios
- Cuando no es obvio
 - Podemos asumir que nuestros prototipos en papel son versiones primitivas del producto
 - Podemos realizar algunos tests de usabilidad con ellos

Testing de prototipado en papel

- Una primera ronda de testeo: *Guerrilla testing* (testing informal)
- Necesitamos unos pocos voluntarios
- Utilizaremos nuestros bocetos de pantallas para mostrarle al usuario el acceso a la funcionalidad
- Podemos empezar a detectar problemas con nuestro diseño en determinadas partes

Testing de prototipado en papel

- Testing más completos o formales
- Típicamente nos centraremos en un subconjunto de la funcionalidad del sistema
 - Basado en tareas
 - Le pediremos al usuario que realice una determinada tarea
- Pasos a tener en cuenta
 - Definir las tareas
 - Crear los prototipos
 - Preparar el test
 - Ejecutar el test

Testing de prototipado en papel

- Definir tareas
 - No estamos recabando opiniones, observaremos al usuario llevando a cabo esas tareas
 - Definir lo que queremos llevar a cabo pero no el cómo (no somos el manual de usuario)
- Preparar las pantallas
 - Definir aquellas que previsiblemente puedan ser visitadas para la tarea (y alguna más...)
 - Crear elementos de pop-up (drop-down menu p.ej.)
 - Preparar un mecanismo para escribir datos
 - Escribir las tareas individualmente en papeles

Testing de prototipado en papel

- Preparar el test
 - Reclutar gente (complicado): tres o cuatro personas es suficiente
 - Mejor que no estén demasiado familiarizados con el entorno de desarrollo (otros desarrolladores de tu empresa, compañero de prácticas,...)
 - No ejecutar los tests de diferentes personas demasiado seguidos
 - Permite analizar los problemas o situaciones no previstas detectadas y corregirlas
 - Involucrar (si es posible) una tercera persona que ayude a interaccionar con el *probador*

Testing de prototipado en papel

- Ejecutando el test
 - No influenciar al usuario
 - No resolverle dudas tras haberse quedado atascado
 - Si es muy obvio que no se va a poder resolver la tarea se puede ayudar (o directamente pasar a otra)
- Analizar resultados
 - Información cualitativa, no cuantitativa
 - Identificar problemas, priorizarlos y resolverlos
- Aunque se puede iterar/refinar tanto como se desee, de nuevo hay que saber cuándo parar

Aspectos del diseño de interfaces:

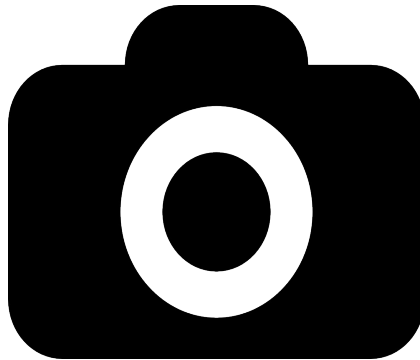
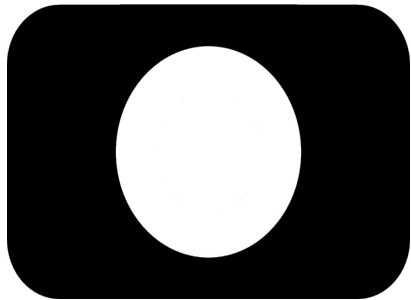
Realismo

- En general, los interfaces se diseñan para tener una apariencia realista
- Ayuda a que el usuario reconozca ciertas funciones
- Ayuda a saber qué esperar de forma intuitiva de ciertos controles
- Sin embargo, si se lleva demasiado a rajatabla puede crear confusión

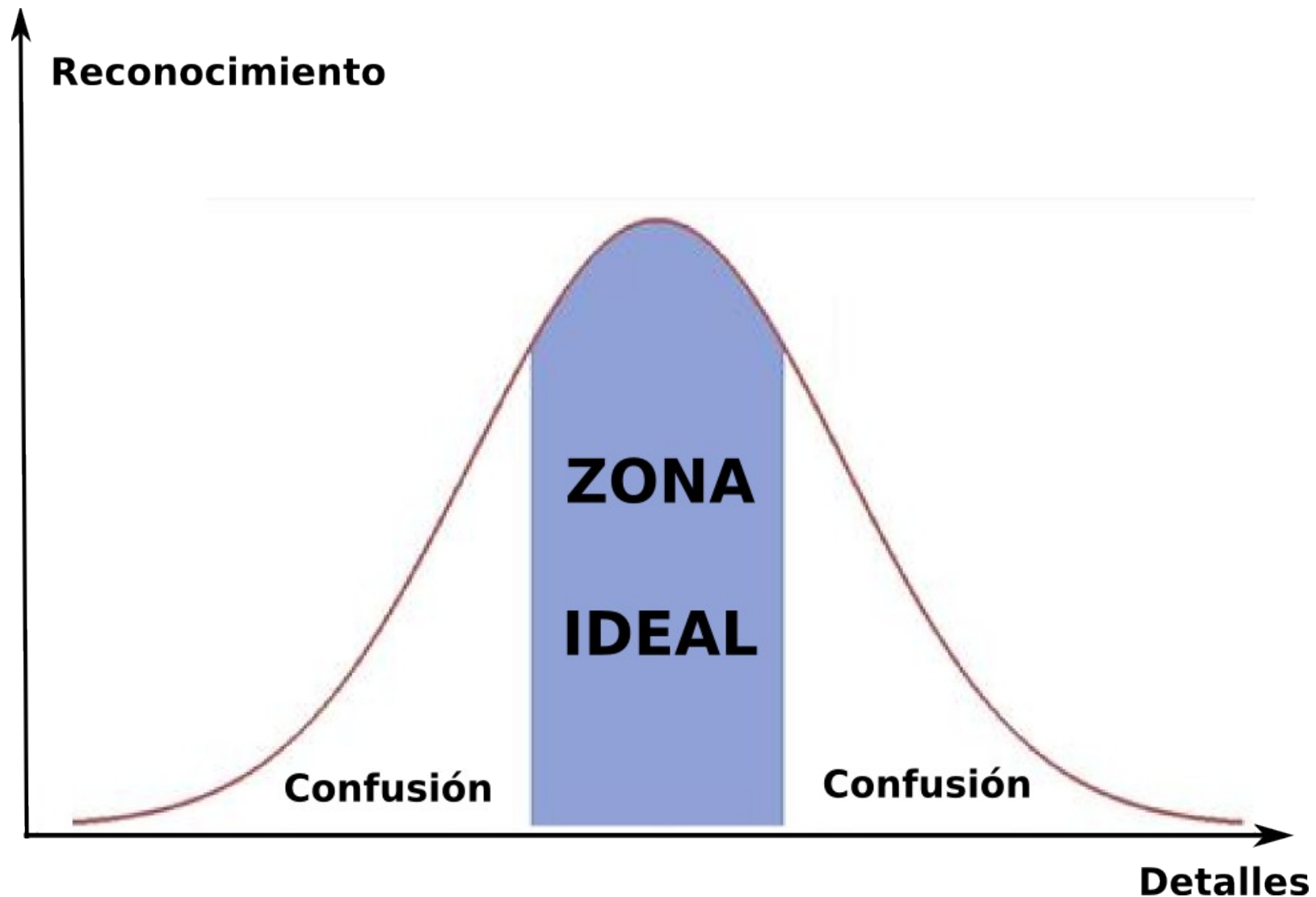
Aspectos del diseño de interfaces:

Realismo

- En general es buena idea usar símbolos para representar elementos funcionales del interfaz
- El detalle es importante (tanto para un lado como para el otro)



Aspectos del diseño de interfaces: Realismo



Aspectos del diseño de interfaces:

Realismo

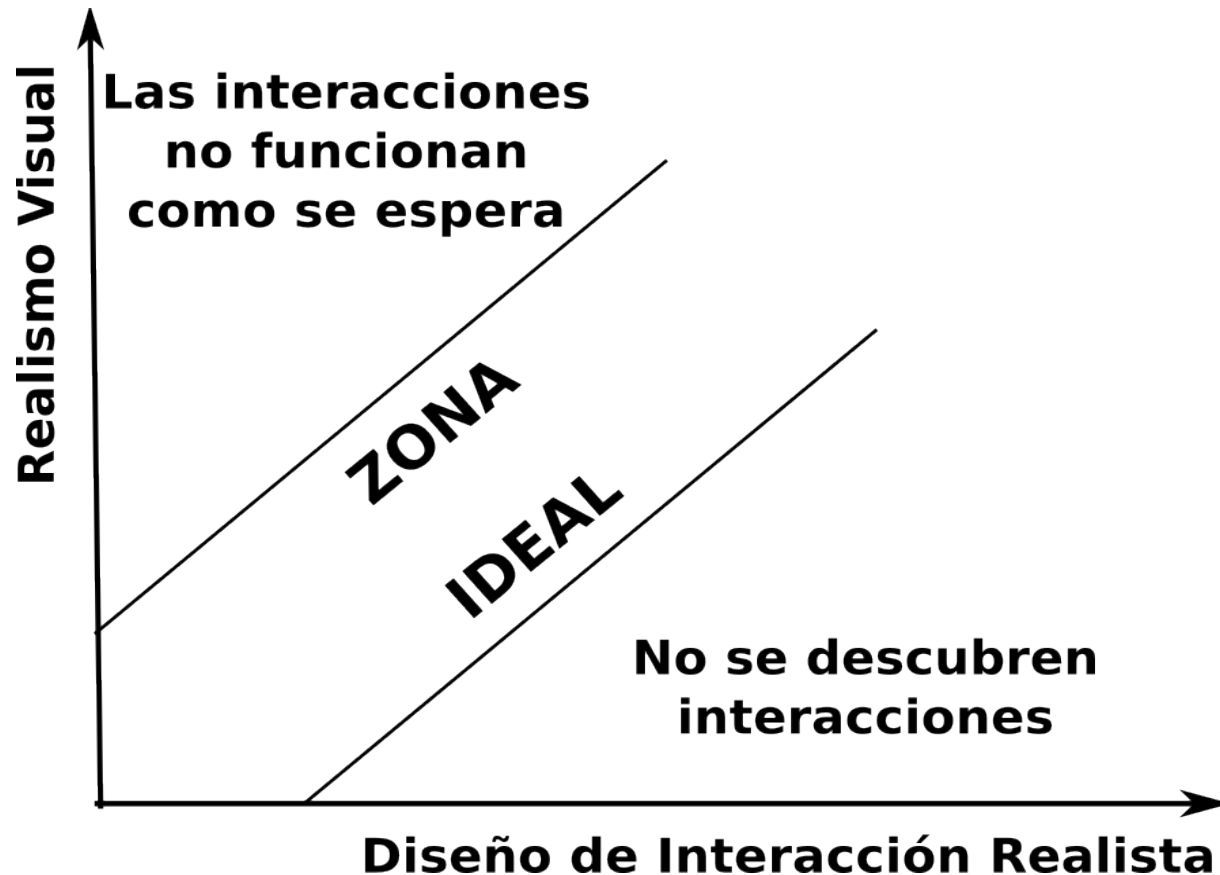
- Interacción realista
 - Simular un interfaz gráfico como un elemento del mundo real con el que comparte funcionalidad
 - Ejemplos: Reproductores de música, calculadoras...
 - Pueden limitar la funcionalidad innecesariamente
- *Skeuomorfismo*
 - Versiones nuevas de objetos que retienen elementos de versiones antiguas que ya no aportan funcionalidad

Aspectos del diseño de interfaces: Realismo



Aspectos del diseño de interfaces: Realismo

- En cuanto a la interacción realista dónde queremos estar:



Aspectos del diseño de interfaces: Interacción Natural

- Distintos paradigmas
 - Command Line Interface
 - Graphical User Interface
 - Natural User Interface
- El paradigma NUI puede resultar muy cómodo para el usuario pero es el más complicado de perfeccionar

Aspectos del diseño de interfaces: Interacción Natural

- Aspectos a considerar en un NUI:
 - Dar feedback de la manera más inmediata posible respecto de las acciones del usuario
 - Cuando tenga sentido usar gestos que activen comandos, no que manipulen objetos en la pantalla
 - Evitar el aprendizaje de gestos demasiado complejos
 - Controlar entradas accidentales

Aspectos del diseño de interfaces:

Ley de Fitts

- Afirma que el tiempo requerido (T) para alcanzar un elemento es dependiente de la distancia al elemento (D) y al ancho (W) en la dirección del movimiento

$$T = a + b \log_2 \left(1 + \frac{D}{W} \right)$$

- a y b son constantes que dependen del dispositivo: tiempo de arranque/parada y factor inverso a la velocidad de movimiento



Aspectos del diseño de interfaces:

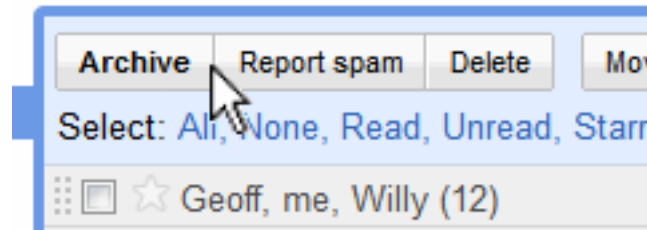
Ley de Fitt

- Un elemento en el extremo de la pantalla virtualmente tiene tamaño infinito
 - Aunque no en pantallas táctiles...
- Menús radiales de contexto reducen la distancia media
 - Típica en menús de videojuegos
 - No es muy cómodo si las opciones crecen demasiado
 - Son más intuitivas disposiciones en horizontal o vertical

Aspectos del diseño de interfaces:

Ley de Fitt

- Conviene separar elementos pequeños
 - Dejar márgenes entre ellos



- Elementos más grandes son más fáciles de clicar
 - Dejar elementos “sensibles” o destructivos en un tamaño razonablemente pequeño y/o alejado para evitar problemas

Aspectos del diseño de interfaces:

Interrupciones

- En general, las interrupciones deben ser evitadas
- Lastran la productividad a la hora de realizar una tarea
 - No solo la pérdida de atención de la tarea principal durante el período de la interrupción sino también el coste posterior de recuperar el ritmo en función del tipo de tarea

Aspectos del diseño de interfaces: Interrupciones

- Tomar decisiones por el usuario
- Generalmente la frecuencia relativa de las posibles opciones no justifica la consulta cada vez



Aspectos del diseño de interfaces: Interrupciones

- A veces es necesario consultar al usuario
 - Si es previsible varias consultas (proceso de instalación p.ej.) es mejor agruparlas todas de antemano
 - Ejemplo: Búsquedas de cadenas en procesador de texto
- Interrumpir solo para decisiones urgentes, no para informar
 - Si no es importante no la muestres
 - Si es importante pero no requiere atención inmediata notifícalo sin interrumpir

Aspectos del diseño de interfaces:

Interrupciones

- Si se abusa de las interrupciones, el usuario no va a leerlas demasiado...
- ...y se va a equivocar
- Importante dejar al usuario deshacer (*undo*) sus acciones
 - Da confianza al usuario
 - Menos *warnings* provoca una mayor concentración cuando aparecen
 - A veces, es útil un *Deshacer* temporal
 - Comunicaciones vía web: envíos de email, twitter, facebook

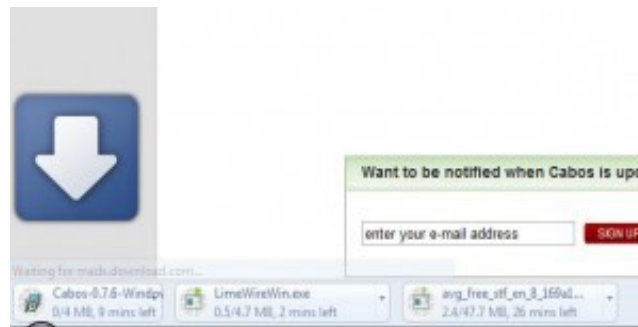
Aspectos del diseño de interfaces:

Animaciones

- A veces cambios de estado visuales de la aplicación pueden desorientar al usuario
- Frases típicas en pruebas de usabilidad
 - ¿Qué pasó?
 - ¿Cómo vuelvo atrás?
 - ¿A dónde fue la ventana?
- Animar el cambio de estado (minimizar una ventana por ejemplo) ayuda al usuario a seguir el proceso

Aspectos del diseño de interfaces: Animaciones

- Dirección de la atención del usuario
 - Caso típico: click en un enlace para descargar fichero



Aspectos del diseño de interfaces: Animaciones

- Evitar animaciones innecesarias
 - Nuestra visión periférica no puede ser ignorada
- Ayudar a los usuarios a formar modelos mentales adecuados
 - Cambios de pantalla en una cierta dirección unidos al botón de *Back* indicando una flecha en la dirección opuesta

Aspectos del diseño de interfaces:

Consistencia

- Es una palabra muy popular, incluido el ámbito de los IU
- Intuitivamente representa la armonía o lógica entre las diferentes partes del producto
- Esa armonía se contrapone a las diferencias que podemos encontrar
- También tenemos que tener en cuenta la consistencia respecto a productos que ya conocemos o usos previos de ciertos elementos

Aspectos del diseño de interfaces:

Consistencia

- Muchas veces la consistencia se usa referida a la apariencia
 - Los botones en la aplicación no parecen iguales que los que tiene el sistema por defecto
 - Ventana diferentes unas de otras
 - Formatos de texto
- Fácilmente identificables y corregibles

Aspectos del diseño de interfaces:

Consistencia

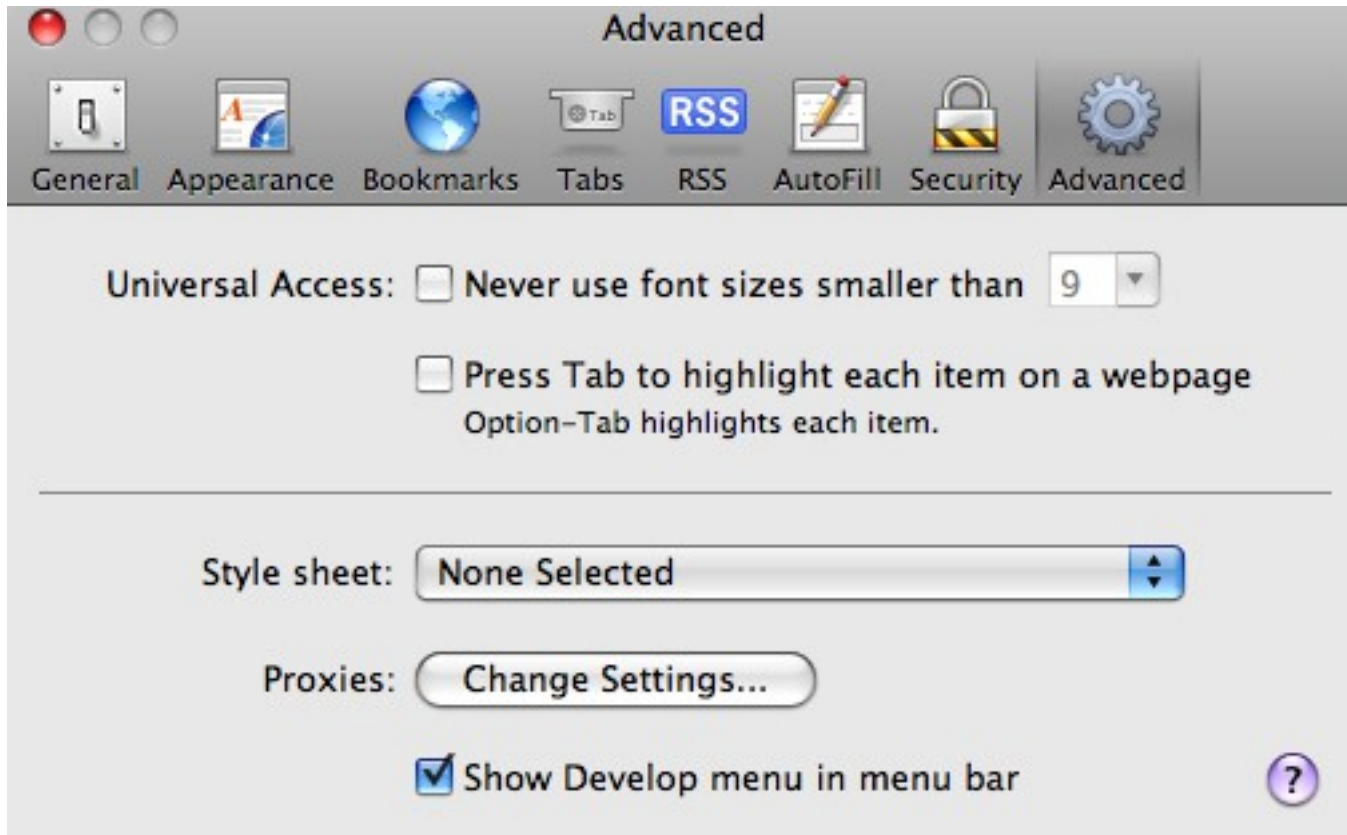
- Consistencia de comportamiento o funcional
 - Cuanto más se parezca algo a lo que el usuario previamente conoce más fácil es de despertar sentimiento de inconsistencia
 - El usuario espera que se comporte de la misma manera
 - Si vas a dar un uso algo diferente a algún elemento, hazlo suficientemente diferente/diferenciable

Aspectos del diseño de interfaces:

Encontrar Funcionalidades

- Una característica importante es la facilidad con la que los usuarios son capaces de encontrar y usar una funcionalidad.
- Para evitar la masificación en pantalla, generalmente hay un compromiso
 - Hacer una funcionalidad más “encontrable” hace que otras partes lo sea menos
 - A veces es correcto “esconderlas” y hacer que las active un usuario

Aspectos del diseño de interfaces: Encontrar Funcionalidades



Aspectos del diseño de interfaces:

Encontrar Funcionalidades

- Es importante decidir dónde encontrar dichas funcionalidades
 - Ej. Función de “rotación” de un objeto en herramienta de dibujo
 - Ej. Borrar una entrada en una agenda virtual
- ¿Cómo hacer cosas más “encontrables”
 - Jugar con propiedades: Tamaño, forma, color, posición
 - Consistencia, de nuevo: Las cosas que queremos que sean más encontrables deberían estar en situaciones/arreglos similares en todas las pantallas

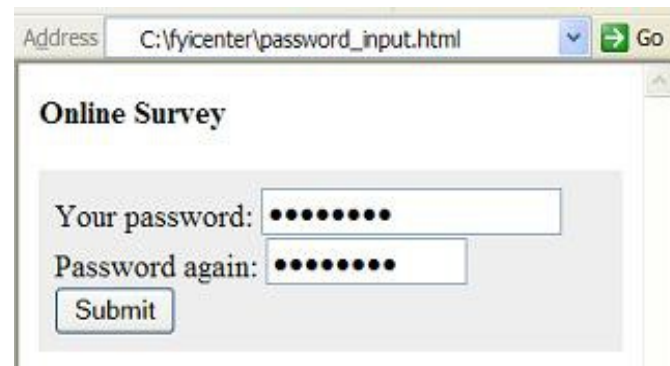
Aspectos del diseño de interfaces:

Modos

- Un *modo* es una parte de la aplicación de la que hay que entrar/salir y que restringe o acota las operaciones que se pueden llevar a cabo mientras esté activo
- Mientras una aplicación (o ventana) esté en un cierto modo, su comportamiento frente a las entradas del usuario varían
- Afectan a la usabilidad
 - No es algo natural para la gente en el mundo real
 - Si el usuario no conoce o entiende el modo actual, se producen respuestas del sistema inesperadas

Aspectos del diseño de interfaces: Modos

- Un ejemplo de modo muy conocido y sencillo: Caps Lock Key
 - En función de si está activa o no, la entrada de usuario es diferente

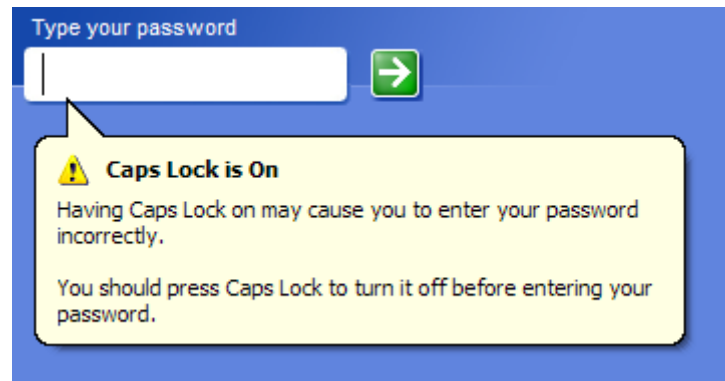


- Si el usuario no es consciente del modo, se producen errores inesperados (p.ej. el logging no funciona)

Aspectos del diseño de interfaces:

Modos no obvios

- Dado que los modos son fuente de errores comunes, es importante darle la información de forma clara al usuario
- La tecla de mayúsculas tiene una luz en el teclado
 - Fácilmente ignorable
 - La mayoría de sistemas de autenticación indican que este modo está activo



- Los modos no obvios han de ser evitados

Aspectos del diseño de interfaces:

Modos no obvios

- ¿Cómo hacemos que el modo sea obvio?
- Feedback visual al usuario para reducir errores y confusión
- Si no funcionan imágenes estáticas pueden valer animaciones sencillas
- Feedback sonoro es más difícilmente ignorable que el visual en muchos casos

Aspectos del diseño de interfaces:

Modos no obvios

- Incluso haciendo que el modo sea obvio para los usuarios, la forma de presentarlo puede llevar a confusión
- En general se recomienda usar *checkboxes* o *radiobuttons* para indicar modos activos

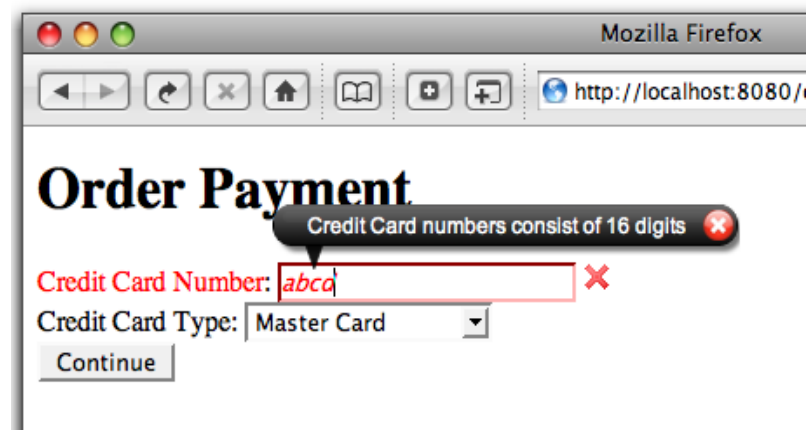
☒ HDR



- ¿Qué significa “*HDR on*”? Hay que pulsar el botón para activarlo o que está activo y si se pulsa se pone a *off*...

Aspectos del diseño de interfaces: Modos inesperados

- Los diálogos son un tipo específico de modo
- Son generalmente obvios por la forma en que aparecen...
- ...pero también inesperados generalmente (¡interrupciones!)
- Pueden *robar el foco*: entrada del usuario se malinterpreta como entrada al diálogo: notificaciones



Aspectos del diseño de interfaces:

Modos

- En general los modos no son muy bien valorados pero no siempre son inadecuados
- Hay veces que los modos son necesarios
 - Para mostrar funcionalidades sólo cuando son aplicables
 - Mientras se entre intencionadamente, se vea de forma obvia su presencia y se conozca la manera de abandonarlo no hay problema
- “Modo profesor/alumno”
 - Feedback
 - Fácil de conocer el modo activo
 - Fácil de abandonar

Aspectos del diseño de interfaces: Quasimodos

- Modos temporales solo activos mientras el usuario explícitamente los mantenga así
 - Mayúsculas manteniendo pulsado la tecla *Shift*
 - Mantener pulsado el botón del ratón
- Siempre son obvios, esperados y fáciles de abandonar
- Conviene considerar si un modo temporal resuelve nuestra necesidad de modo o es necesario algo más “persistente”

Aspectos del diseño de interfaces:

Velocidad

- Un aspecto difícilmente medible en tests con usuarios es la velocidad y respuesta de la aplicación
- Es fácil saber cuándo no entienden una etiqueta de un botón
- Pero normalmente (salvo casos flagrantes que no deberían haber llegado hasta esa prueba de usabilidad...) no se mostrará si la velocidad de respuesta es adecuada
- La velocidad es la característica más importante
- A diferencia de otros problemas vistos, en caso de aplicaciones lentas los usuarios pueden abandonarlas rápidamente

Aspectos del diseño de interfaces:

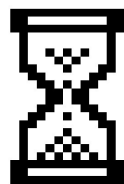
Velocidad

- La capacidad de respuesta (*responsiveness*) de un sistema es crucial
- Las investigaciones en este asunto son bastante concluyentes
 - Acciones que duran < 0.1 s se consideran instantáneas
 - Acciones < 1 s, aunque no instantáneas, no permiten perder el foco con facilidad al usuario
 - Por encima de 1s se corre ese riesgo
- La diferencia entre que el usuario perciba su interacción como con objetos físicos reales frente a dar instrucciones a una máquina

Aspectos del diseño de interfaces:

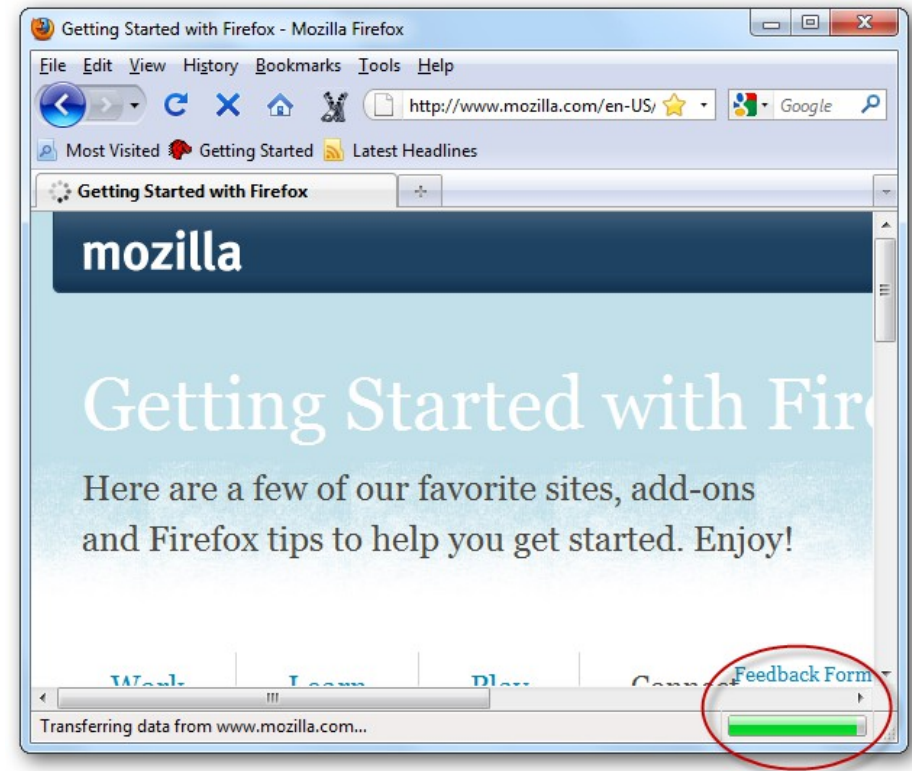
Velocidad

- Si una operación va a durar más de lo esperado es bueno dar feedback al usuario
- El tipo de feedback depende de varios aspectos
 - Cuánto dura la acción
 - Qué tipo de acción es
- Si la acción dura unos pocos segundos, es suficiente con cambiar el cursor al típico reloj de arena o un indicador de “procesando” sin indicar el estado de progreso concreto de la acción



Aspectos del diseño de interfaces: Velocidad

- En operaciones cortas, el objetivo simplemente es hacerle obvio al usuario que se ha recibido la petición y se está trabajando
- A veces si no se hace suficientemente obvio, aunque se haga igualmente, puede no producir ninguna mejora



Aspectos del diseño de interfaces:

Velocidad

- Si la acción va a durar más de unos segundos, es cuando tiene sentido indicar algún tipo de progreso



- En algunos casos es recomendable también indicarle al usuario en qué tarea se está si la acción incluye unas pocas tareas fácilmente identificables
 - Ayuda a que el usuario entienda la duración del proceso
- Si la tarea dura bastante (instalaciones, conversiones de vídeo, etc.) el usuario se centrará en otro asunto
 - No interrumpir: audio, aviso en la barra de tareas, etc.

Aspectos del diseño de interfaces:

Velocidad

- Más importante que la velocidad real es la *velocidad percibida*
- Cómo perciben los usuarios la velocidad de nuestro producto
- Maneras de mejorar la velocidad percibida
 - Mostrar resultados parciales cuando sea posible
 - No bloquear la interfaz con operaciones lentas
 - Decoraciones: efectos de ondas en las barras de progreso en sentido contrario al avance de la misma
- A veces hay que añadir barras “de pega” para dar feedback o ralentizar procesos importantes que suceden demasiado rápido para el usuario