

<http://britton.disted.camosun.bc.ca/goldslide/bgoldslide.htm>

## Tema 5.3

# Diseño físico

Interfaces Persona Computador

Depto. Sistemas Informáticos y Computación. UPV

### DOCENCIA VIRTUAL

#### Finalidad:

Prestación del servicio Público de educación superior (art. 1 LOU)

#### Responsable:

Universitat Politècnica de València.

**Derechos de acceso, rectificación, supresión, portabilidad, limitación u oposición al tratamiento conforme a políticas de privacidad:**

<http://www.upv.es/contenidos/DPD/>

#### Propiedad intelectual:

Uso exclusivo en el entorno de aula virtual.

Queda prohibida la difusión, distribución o divulgación de la grabación de las clases y particularmente su compartición en redes sociales o servicios dedicados a compartir apuntes.

La infracción de esta prohibición puede generar responsabilidad disciplinaria, administrativa o civil



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



En este tema vamos a ver cómo obtener el diseño físico de una interfaz de usuario.

## Objetivos

- ▶ El Diseño Físico es el diseño real de las pantallas de nuestra aplicación utilizando los controles disponibles
- ▶ Conocer las diferencias entre principios de diseño, estándares de interfaces de usuario y guías de estilo
- ▶ Conocer, reconocer y ser capaz de aplicar los principios de diseño más comunes
- ▶ Conocer los controles más comunes y saber cómo usarlos de forma efectiva en una interfaz gráfica de usuario



2

Después del análisis de requisitos y del diseño conceptual necesitamos decidir cómo transformar nuestros contenedores en una interfaz física. Normalmente nos basaremos en principios de diseño y en estándares y guías de estilo existentes, pero también necesitaremos aplicar los conceptos vistos hasta ahora, por ejemplo, cómo es la percepción humana, qué principios de diseño necesitamos aplicar para obtener aplicaciones usables y accesibles, etc.. En este tema profundizaremos en algunos de estos conceptos, y también veremos cuáles son los controles adecuados a utilizar en una interfaz.

# Índice

- ▶ Introducción
- ▶ Principios de diseño
- ▶ Estándares de interfaces de usuario
- ▶ Guías de estilo
- ▶ Principios de una buena composición
- ▶ Diseñando una interfaz gráfica de usuario

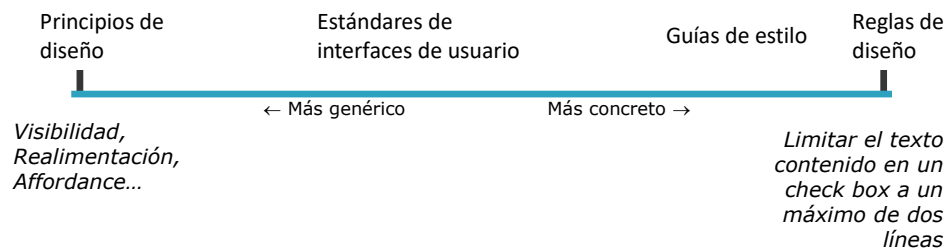


3

Empezaremos hablando de los principios de diseño, de los estándares y guías de estilo, y por último hablaremos de los principios, o criterios a seguir, para conseguir una buena composición. Estos principios no hay que confundirlos con los principios de diseño.

# Introducción

- ▶ Diseñar una interfaz de usuario consiste en tomar muchas decisiones
- ▶ Las decisiones de diseño se pueden basar en muchas fuentes:



Recordamos que los principios de diseño representan objetivos generales, no indican cómo alcanzarlos. Para poder alcanzar estos objetivos necesitaremos basarnos en recomendaciones más o menos concretas. Dependiendo de su grado de concreción estaremos hablando de estándares (más genéricos) o de Reglas de estilo, mucho más concretas. Por ejemplo, la guía de estilo de Windows 8 indica: “el texto máximo de un check box no debe exceder de dos líneas” (*“Limit check box text content to no more than two lines”*, pp. 100 Windows 8. Design and coding guidelines. Updated for Windows 8.1)

# Principios de Diseño

- ▶ Los principios de diseño son:
  - Generales, aplicables y duraderos
  - Requieren aclaración
- ▶ Principios
  - Visibilidad
  - Affordance
  - Realimentación
  - **Simplicidad**
  - **Estructura**
  - **Consistencia**
  - **Tolerancia**



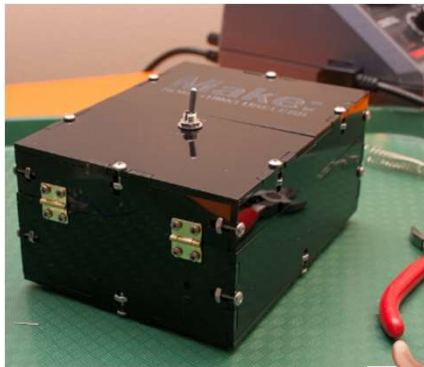
5

Ya en el tema 3 se hablo de los principios de diseño más importantes basados en las observaciones de Don Norman . En este tema los vamos a repasar y presentaremos otros 4: Simplicidad, estructura, consistencia y tolerancia.

# Principios de Diseño

## ► Visibilidad

- Los controles deberían ser fáciles de encontrar



makershed.com/products/useless-box-kit



6

Recordamos que el principio de Visibilidad nos dice que: los controles para las funciones principales del sistema deberían ser claramente visibles o fáciles de encontrar.

# Principios de Diseño

## ► Visibilidad

- Una forma de hacer que el uso del sistema sea más fácil para los usuarios es incrementando la visibilidad de su interfaz
  - Crear accesos directos para las tareas más utilizadas
  - Pero, por otra parte, esto incrementa el número de controles y hace que la interfaz parezca más compleja



Conviene señalar que alcanzar el principio de visibilidad no siempre es fácil. En una situación de máxima visibilidad todos los controles deberían estar a la vista, pero eso haría que la interfaz fuera mucho más compleja, lo que contradice otros principios, como el de simplicidad. Por tanto, vamos a dar prioridad a la visibilidad de los controles para la realización de las tareas más importantes o más frecuentes.

# Principios de Diseño

## ► Affordance

- Debería ser obvio cómo utilizar la interfaz



El segundo principio es el de affordance. Una interfaz debería poderse utilizarse de una forma intuitiva. En la imagen de la izquierda vemos un pulsador que nos permite solicitar que el semáforo se ponga en verde para los peatones. Está claro cómo usar un botón. Sólo hay que pulsarlo.

En la imagen de la derecha vemos una puerta, una puerta es algo que todos conocemos y debería ser sencillo de usar. Si embargo, en el ejemplo se muestra una puerta automática que tiene asas, lo que nos induce a pensar que para abrirla se debería tirar o empujar.

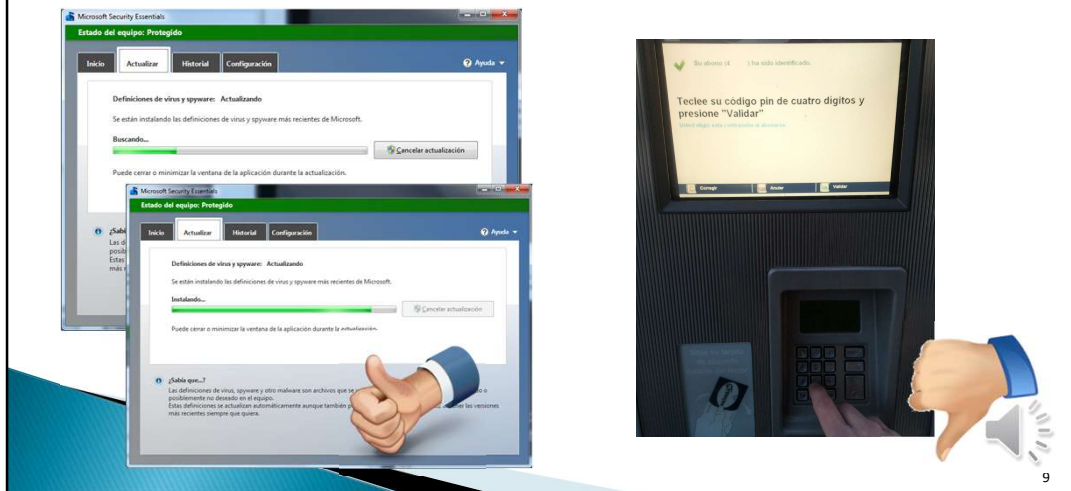
<https://www.youtube.com/watch?v=qtCEoGyfsxk#action=share>



# Principios de Diseño

## ► Realimentación

- El sistema debería decirnos qué está haciendo en cada momento



El tercer principio es el de Realimentación. Cada acción que se realice sobre el sistema debe tener una reacción que le permita al usuario conocer el estado del sistema o saber qué acción se ha realizado. En la imagen de la izquierda la función de actualizar del Microsoft Security Essentials no sólo te dice cuánto queda (más o menos) para completar la tarea, sino que también te dice qué está haciendo (Buscando o Instalando). En la imagen de la derecha se está introduciendo un pin. En realidad, sí está dando feedback, pero lo está dando en la pantalla pequeña, de manera que puede pasar desapercibido para el usuario.

# Principios de Diseño

## ► Simplicidad

- La IU debe ser tan sencilla como sea posible
- La interfaz debería utilizar el mismo lenguaje que los usuarios
- Utilizar acciones, iconos, palabras y controles que sean naturales para los usuarios
- Descomponer tareas complejas en subtareas más sencillas



El cuarto principio es el Simplicidad. Este principio nos indica que una interfaz debe ser lo más sencilla posible. Tomar este principio al pie de la letra no es buena idea. Llevado al extremo, en cada pantalla podríamos tener un único control, lo que haría que la interacción con la aplicación fuera imposible. Aunque es cierto que hay aplicaciones que por su naturaleza no permiten hacer ventanas realmente sencillas, siempre podemos mejorar su simplicidad utilizando términos, controles o iconos conocidos para el usuario, o se puede descomponer las tareas en subtareas más sencillas, aunque esto suponga incrementar el número de ventanas necesarias.

# Principios de Diseño

## ► Simplicidad

- Uno de los principales problemas de las interfaces modernas es su complejidad. Deberíamos intentar crear interfaces simples
  - ¿Es necesario que ese control esté ahí? ¿Es necesario que el botón de búsqueda esté animado? ¿Es realmente necesario que suene una melodía cada vez que se ejecuta la aplicación?
- Por otro lado, hay sistemas complejos que requieren interfaces complejas



[digitalavmagazine.com/2012/06/11/la-tecnologia-led-full-hd-de-barco-ayuda-a-vigilar-el-trafico-en-valencia/](http://digitalavmagazine.com/2012/06/11/la-tecnologia-led-full-hd-de-barco-ayuda-a-vigilar-el-trafico-en-valencia/)



[wired.com/2011/10/cockpit-tour-of-all-nippon-airways-boeing-787-dreamliner/](http://wired.com/2011/10/cockpit-tour-of-all-nippon-airways-boeing-787-dreamliner/)

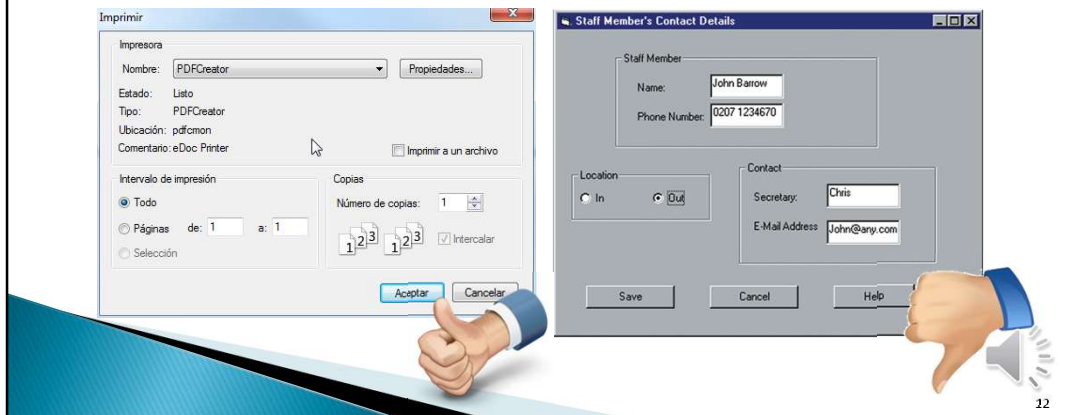
11

Nos tenemos que plantear si es necesario todo lo que aparece en la interfaz. No deberíamos utilizar controles que no se van a utilizar o que no son adecuados para la tarea. Cada elemento de la interfaz debe tener un propósito. Por ejemplo, añadir algún, sonido o melodía, o incluir efectos visuales como parpadeos, cambios de color o de tamaño, tienen que estar justificados. Recordad que la interfaz es un medio para conseguir un objetivo, y por tanto no debe suponer una distracción o molestia para el usuario.

# Principios de Diseño

## ► Estructura

- Organizar la IU de forma que tenga significado y sea útil
- Las cosas que los usuarios piensan que están relacionadas deberían aparecer juntas
- Utilizar metáforas para proporcionar una estructura reconocible



El principio de Estructura nos indica que la interfaz se debe organizar de forma que tenga sentido y ayude al usuario a completar la tarea de la forma más eficiente posible. Las leyes Gestalt que se vieron en el tema 3 son una buena forma de conseguir una buena estructura. Por ejemplo en la imagen de la derecha no se ha alineado ni agrupado bien la información lo que además de transmitirnos desorden nos puede inducir a error.

# Principios de Diseño

## ► Consistencia

- Enfatizar la importancia de la uniformidad en la apariencia, colocación y comportamiento, para construir interfaces que sean fáciles de aprender y recordar
- Diferentes tipos de consistencia: acciones, objetos, color, composición, iconos, fuentes, etc.



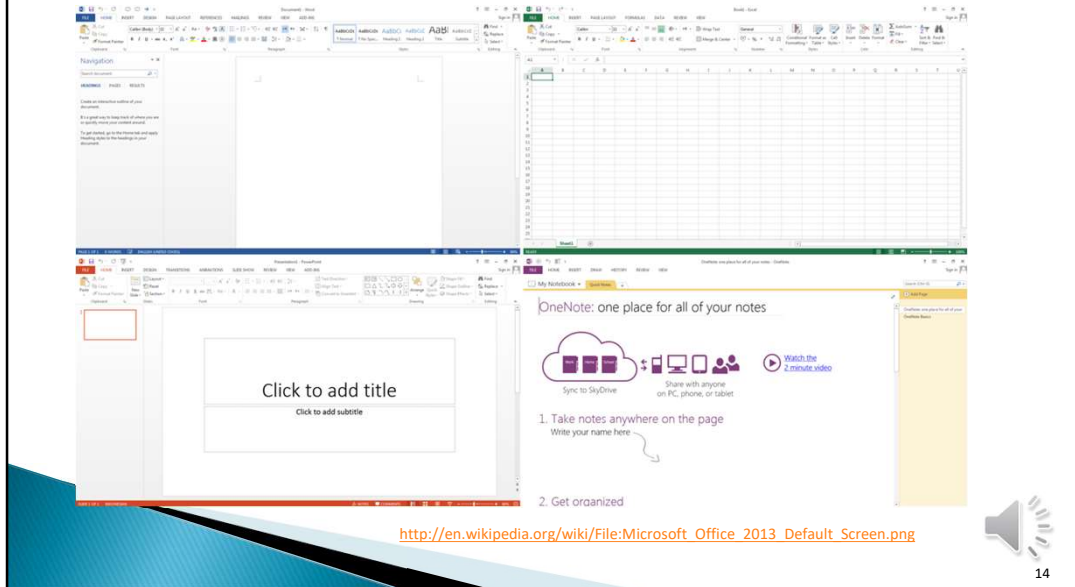
El siguiente principio es el de Consistencia. La consistencia nos puede ayudar a prevenir errores de usuario causados por el principio psicológico “los usuarios ven lo que esperan ver”

El ejemplo de la imagen de la derecha contradice este principio, ya que no es posible utilizar YES, NO, Cancel y OK, Cancel a la vez, ni cambiar el orden o color de los botones en cada ventana.

La posición de los menús, de las barras de herramientas, hay que situarlos en el lugar habitual para ellos. Hay que facilitar encontrar las opciones y situarlas siempre que sea posible, en la misma posición en las diferentes vistas.

# Principios de Diseño

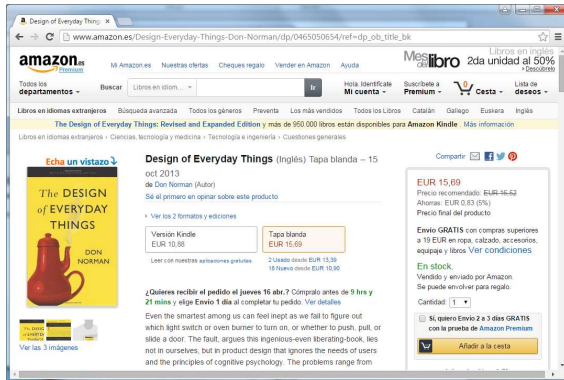
## ► Ejemplo de consistencia: Microsoft Office



Aquí vemos como las diferentes herramientas de Microsoft Office utilizan un lenguaje común, una estructura y aspecto similar... lo que hace que una vez hemos aprendido a manejar una de ellas aprender cualquiera de las demás nos va a resultar mucho más sencillo.

# Principios de Diseño

- En ocasiones algo de inconsistencia es útil



15

Por otra parte, hay que tener en cuenta, que una consistencia absoluta, mismos colores, mismo tamaño, etc. no es útil si queremos llamar la atención sobre algún elemento concreto. Para dirigir la atención del usuario a un elemento determinado podemos utilizar colores que se salgan de la gama de colores utilizado o cambiar estilos y tamaños de letra. Fijaos que aquí estaríamos aplicando la ley de Gestalt de separación figura-fondo para incrementar el contraste.

# Principios de Diseño

## ► Tolerancia

- Diseñar la interfaz para prevenir que el usuario cometa errores y facilitar la recuperación del sistema
- Deshabilitar los botones u opciones de menú que no estén disponibles, mostrar el formato de entrada requerido...



Vamos a hablar ahora del principio de tolerancia. Este principio nos indica que hay que diseñar interfaces que prevengan los errores de usuario, y que en caso de cometerse, nos podamos recuperar del error de una forma sencilla. La mejor forma de prevenir un error es eliminando la posibilidad de que ocurra: por ejemplo, desactivar botones u opciones de menú que no deben estar disponibles. También se debe informar al usuario de los campos que requieren un formato especial, o como mínimo proporcionar mensajes de error que expliquen bien el error cometido.

En el ejemplo de la intranet de la UPV, introducir la letra del DNI es un error. Sin embargo, no se especifica en ningún sitio que no se debe introducir y el mensaje de error que muestra no es adecuado. Para seguir el principio de tolerancia se debería aceptar el DNI con y sin letra.

En el ejemplo de Powerpoint, no hay ayuda para establecer el rango de impresión. Para imprimir las transparencias 7 y 20 no se usa la coma, sino el punto y coma. No se dice en ningún sitio, el icono de la ayuda no hace nada, y el mensaje de error es genérico y no da pistas sobre cuál es el problema o cómo resolverlo.



# Principios de Diseño

## ► Tolerancia

- Recuperabilidad: cómo de fácil es para los usuarios corregir un error
  - Recuperación hacia delante
    - Aceptar el error del usuario y ayudarlo a corregirlo
  - Recuperación hacia atrás:
    - Permitir a los usuarios deshacer sus acciones



17

En caso de que se cometan errores de usuario, dado que no siempre se pueden evitar, el sistema nos tiene que facilitar recuperarnos de ese error.

Básicamente hay dos formas de recuperarse de un error, aceptar el error y posteriormente llevar a cabo otra acción que lo solucione, lo que sería recuperación hacia delante, o deshacer la acción errónea, lo que sería recuperación hacia atrás.

Un ejemplo de recuperación hacia delante sería cuando escribimos mal un apalabra en un editor de textos. Aunque nos deja escribirla y continuar nos indica que es un error y deja en manos del usuario el llevar a cabo alguna acción que lo subsane. Un ejemplo de recuperación hacia atrás, es cuando borramos un texto sin querer. El sistema nos permite deshacer la operación y volver a la situación inmediatamente anterior a la acción de borrado .

# Principios de Diseño

## ► Tolerancia

- Los buenos mensajes de error ayudan a los usuarios a corregir sus errores
- Guías para crear mensajes de error:
  - Explicar los errores para ayudar al usuario a corregirlos
  - Proporcionar información adicional si el usuario lo solicita
  - Utilizar un lenguaje que entienda el usuario
  - Utilizar un lenguaje positivo, no amenazante
  - Utilizar términos específicos y constructivos (ej. “Rango de meses de 1 a 12” en lugar de “Valor inválido”)
  - Asegurarse de que el sistema asuma la culpabilidad de los errores



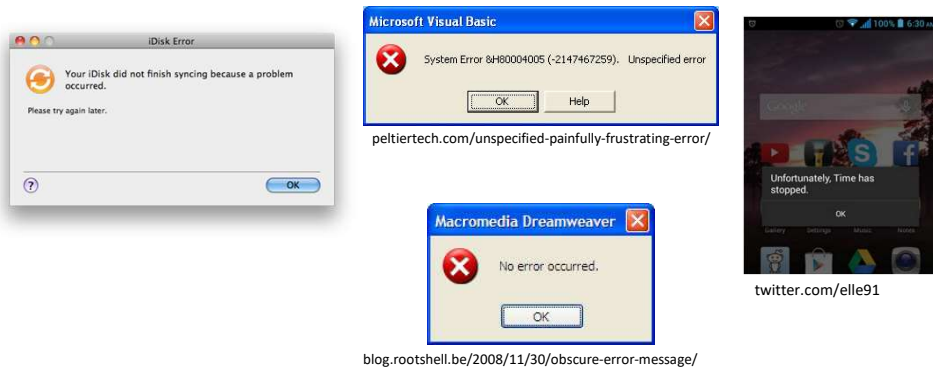
18

La tolerancia también está relacionada con los mensajes de error. Para crear un buen mensaje de error que nos ayude a recuperarnos de un error, se deben seguir ciertas directrices o recomendaciones que ayuden al usuario a comprender la causa del error y a solucionarlo. El lenguaje utilizado en los mensajes debe ser comprensible para el usuario y no le deben generar inseguridad, ansiedad o miedo a cometer nuevos errores.

# Principios de Diseño

## Tolerancia

### ► Mensajes de error poco útiles:

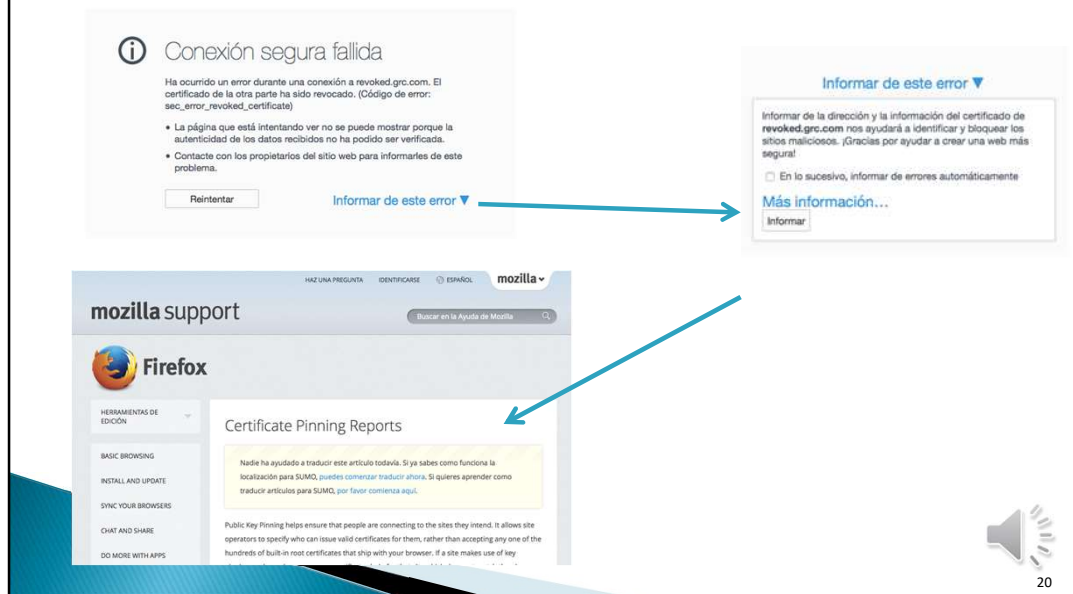


Aquí vemos algunos ejemplos con mensajes de error que no siguen el principio de tolerancia ni las recomendaciones para la creación de buenos mensajes de error. Los mensajes que vemos aquí no indican al usuario cuál es la causa del error ni cómo resolverlo.

# Principios de Diseño

## Tolerancia

### ► Ejemplo



Por otra parte este ejemplo sí que sigue el principio de tolerancia. En primer lugar nos explica la causa del error, nos permite informar a un tercero del error ocurrido y nos permite ampliar la información sobre el error.

# Principios de Diseño

- ▶ Heurísticas de usabilidad de Jakob Nielsen para el diseño de la interacción (1995) (<http://www.nngroup.com/articles/ten-usability-heuristics>)

1. Visibilidad del estado del sistema

- El sistema debería mantener siempre informado al usuario sobre lo que está pasando, mediante una realimentación apropiada dentro de un tiempo razonable

2. Relacionar el sistema y el mundo real

- El sistema debería hablar el lenguaje del usuario, con palabras, frases y conceptos familiares para el mismo, en vez de usar términos tecnológicos. Seguir las convenciones del mundo real, haciendo que la información aparezca de una forma natural y lógica.

3. Control y libertad del usuario

- A menudo los usuarios seleccionan funciones del sistema por error y necesitan una “salida de emergencia” claramente indicada para abandonar el estado actual sin tener que realizar muchos pasos. Implementar las opciones de deshacer y rehacer.

4. Consistencia y estándares

- Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones son lo mismo. Sigue las convenciones de la plataforma.



21

Hay muchos autores que han propuesto principios de diseño similares . Uno de ellos es Jakob Nielsen, un renombrado experto en diseño web para obtener una buena experiencia de usuario (lo que se conoce como diseño UX). Nielsen propuso 10 principios heurísticos de usabilidad, que como vamos a ver, son consistentes con los principios de diseño propuestos por Don Norman.

1. Visibilidad del estado del Sistema: se refiere a que se debe proporcionar información al usuario sobre lo que está pasando en el sistema. Fijaos que, a pesar del nombre, este principio de Nielsen no se corresponde con el principio de Visibilidad de Don Norman, sino con el principio de **Realimentación**.
2. Relacionar el Sistema y el mundo real: Se trata de utilizar un lenguaje conocido para el usuario utilizando iconos y metáforas similares al mundo real, como por ejemplo el icono de la papelera para borrar ficheros.
3. Control y libertad de usuario: es un principio relacionado con la recuperación frente a errores del usuario. Este principio se correspondería con el principio de **Tolerancia** de Don Norman.
4. Consistencia y estándares: este principio se corresponde con el principio de consistencia visto anteriormente . Nilsen añade la conveniencia de ajustarse a estándares o las convenciones propias de la plataforma.

# Principios de Diseño

## ► Heurísticas de usabilidad de Jakob Nielsen para el diseño de la interacción

### 5. Prevención de errores

- Mejor incluso que buenos mensajes de error es un diseño cuidadoso que previene que los problemas ocurran. Eliminar las condiciones propensas a producir errores o pedir al usuario una confirmación antes de realizar la acción.

### 6. Reconocer en vez de recordar

- Minimizar la necesidad de que el usuario recuerde cómo realizar tareas haciendo que los objetos, acciones y opciones sean visibles. El usuario no debería tener que recordar información de una parte de la interfaz para usarla en otra. Las instrucciones de uso del sistema deberían estar visibles o deberían poder mostrarse fácilmente cuando fuera necesario.

### 7. Flexibilidad y eficiencia de uso

- Los aceleradores –invisibles para el usuario novato– a menudo permiten a los usuarios expertos trabajar más rápido, por lo que el mismo sistema se adapta a ambos tipos de usuario. Permitir a los usuarios personalizar las acciones frecuentes.

### 8. Estética y diseño minimalista

- Los diálogos no deberían contener información irrelevante o que se necesita raramente. Cada unidad de información adicional en un diálogo compete con las unidades de información relevantes y disminuye la visibilidad relativa de las mismas



22

5. Prevención de errores: se trata de evitar, en la medida de lo posible que el usuario cometa errores. Por ejemplo, se podrían activar o desactivar botones para evitar que el usuario realice acciones no permitidas. Este principio también se correspondería con el principio de **Tolerancia** de Don Norman.

6. Reconocer en vez de recordar: se trata de poner en práctica el principio de la psicología humana que se vio en el tema 3. Para conseguir esto deberíamos diseñar nuestra interfaz de manera que las acciones más frecuentes o importantes sean visibles y fácilmente identificables por el usuario. Este principio se corresponde con el principio de Visibilidad de Don Norman.

7. Flexibilidad y eficiencia de uso: si es posible se proporcionarán diferentes formas de interacción, por ejemplo usar teclas de aceleración o atajos, de manera que nuestra aplicación se adapte a diferentes tipos de usuario.

8. Estética y diseño minimalista: Las interfaces muy recargadas son complejas de entender lo que hace que el usuario no se centre en las características más importantes. Una forma de reducir la complejidad puede ser utilizando etiquetas cortas y comprensibles. Por ejemplo, en la etiqueta de un campo de entrada, en lugar de poner “introduzca su nombre” podríamos poner simplemente “Nombre”. Esta regla se correspondería con el principio de Simplicidad visto anteriormente.

# Principios de Diseño

## ► Heurísticas de usabilidad de Jakob Nielsen para el diseño de la interacción

9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de un error
  - Los mensajes de error deberían expresarse en lenguaje sencillo (sin códigos), indicando de forma precisa el problema, y sugiriendo de forma constructiva una solución
10. Ayuda y documentación
  - Aunque es mejor que el sistema se pueda usar sin documentación, puede ser necesario proporcionar ayuda y documentación. Debería ser sencillo buscar dicha información, debería estar centrada en las tareas del usuario, enumerar los pasos concretos a seguir, y no ser demasiado larga.



23

En los dos últimos principios Nielsen recomienda ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores. Para ello hay que proporcionar mensajes de error y herramientas de ayuda adecuados. En cualquier caso, hay que recordar que siempre es mejor que el sistema se pueda utilizar sin recurrir a la documentación.

# Principios de Diseño

## Principios de diseño en Windows 7

- ▶ Reducir los conceptos para aumentar la confianza
- ▶ Las cosas pequeñas importan, para lo bueno y para lo malo
- ▶ Sé bueno en la apariencia y en la acción
- ▶ Soluciona las distracciones, no la capacidad de descubrir
- ▶ Experiencia de usuario antes que consejos y preguntas
- ▶ Personalización, no customización
- ▶ Valorar la experiencia en el ciclo de vida
- ▶ El tiempo importa, diseña para la gente en movimiento



video

Más información y dos presentaciones en:

<http://channel9.msdn.com/Events/MIX/MIX09/C26F>



24

En esta transparencia tenemos la lista de principios para el diseño de aplicaciones que corren bajo Windows 7.

Como: Hacer interfaces sencillas; prestar atención a los detalles, ya que hacer que las cosas funcionen no es suficiente, la vista también tiene que ser agradable al usuario; Recordamos que en el tema 3 se habló de las dificultades que tenemos las personas en centrarnos en más de una actividad a la vez, por tanto deberíamos diseñar para prevenir estas distracciones o para facilitar la vuelta a la tarea que estábamos realizando; aprovechar la experiencia de usuario de manera que se minimicen las necesidades de ayuda, Primar la personalización de la interfaz (el usuario no tiene que decir explícitamente cuáles son sus preferencias), frente a la customización (el usuario define sus preferencias explícitamente). Valorar no solo el uso de la interfaz, sino cuidar también la instalación o mantenimiento de la aplicación y diseñar aplicaciones que permitan realizar las tareas de forma eficiente.

Para más información podéis seguir el enlace que tenéis aquí:

<http://channel9.msdn.com/Events/MIX/MIX09/C26F>



# Estándares de Interfaces de usuario

- ▶ Estándares

- Normalmente también incluyen principios de diseño y buenas prácticas y reglas ampliamente aceptados

- ▶ Estándares para HCI y usabilidad

- ISO 9241: Ergonomic requirements for office work with visual display terminals (VDTs)
- ISO 14915: Software ergonomics for multimedia user interfaces
- ISO 13407: Human-centered design processes for interactive systems
- ISO/CD 20282: Ease of operation of everyday products



25

Los estándares se pueden utilizar como punto de partida del diseño. Generalmente incluyen principios de diseño junto con buenas prácticas y reglas. En ocasiones, el ajustarse a un determinado estándar viene impuesto por el dominio de la aplicación o es parte de los términos en los que se ha definido el proyecto.

En Poliformat tenéis una copia de la versión en español del estándar ISO 9241-171 sobre la ergonomía de la interacción persona-sistema.

# Estándares de Interfaces de usuario

## ► Ejemplo:

- U.S. Nuclear Regulatory Commission. Human-System Interface Design Review Guidelines: <http://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0700/>

### 1 INFORMATION DISPLAY 1.1 General Display Guidelines

#### 1.1-28 Grouping of Related Information

Related information should be organized into groups.

*Additional Information:* Information needed by the operator to accomplish a task should be grouped and perceptually related, when possible. To minimize the disadvantages of divided attention, the number of attention shifts should be minimized, both within a display page and between them.<sup>6690, 6633</sup>

#### 1.1-29 Spatial Proximity for Related Information

Information that must be compared or mentally integrated should be presented in the close spatial proximity.

*Additional Information:* If possible, the information items should be contained in the same display page and grouped together. Spatial proximity may also be achieved by presenting the display pages in adjacent display windows or on adjacent display devices that can be viewed together.<sup>6690</sup>

#### 1.1-30 Chromatic Proximity for Related Information

If information must be mentally integrated, similar color codes should be used for the information items.

*Additional Information:* Information items may be easier to identify if a similar color coding scheme has been applied to them. This may be particularly important if the information items are spatially separated.<sup>6690</sup>

- Más ejemplos en:

<http://www.hfes.org/Web/Standards/GovernmentStandardsIndex.pdf>



26

Aquí tenéis un extracto del estándar de diseño de interfaces persona-sistema que utiliza la Comisión de regulación nuclear de Estados Unidos

## Guías de estilo

- ▶ Proporcionan las convenciones básicas para productos específicos o para una familia de productos. Normalmente incluyen:
  - Una descripción e ilustración de los estilos de interacción necesarios y de los controles de la interfaz de usuario
  - Guía de cuándo y cómo utilizarlos
  - Plantillas que muestran cómo debería ser el aspecto de las pantallas
- ▶ Dos tipos de guías de estilo:
  - Comerciales
  - Personalizadas



27

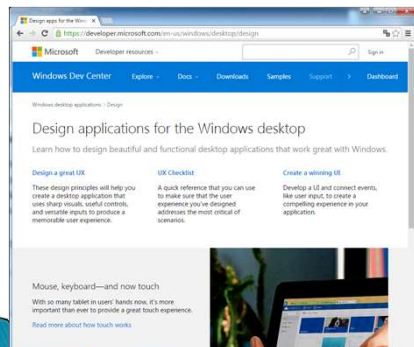
Las guías de estilo pueden ser personalizadas, para un producto concreto, o comerciales, para una familia de productos. Tanto unas como otras incluyen reglas más específicas que los principios de diseño.

Generalmente se utilizan para conseguir una imagen de marca de manera que, a través del diseño, el usuario pueda identificar la compañía que hay detrás de un producto.

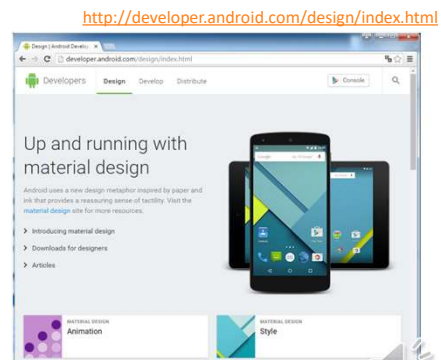
# Guías de estilo

## Guías de estilo comerciales

- ▶ Guías de estilo comerciales
  - Publicadas por una compañía
  - Compuestas por reglas de diseño muy específicas
  - Sólo aplicables a una plataforma concreta



<https://developer.microsoft.com/en-us/windows/desktop/design>



Aquí podemos ver dos guías comerciales, una es de Microsoft y la otra de Google, a las que se tienen que ajustar los desarrolladores de aplicaciones para Windows o Android.

# Guías de estilo

## Guías de estilo comerciales

### Windows 8 Design and coding guidelines

Updated for Windows 8.1

- ▶ Incluye principios y directrices
- ▶ Especifica, por ejemplo:
  - Cómo elegir un control
  - Cómo asignar el tamaño y separación de los controles
  - Cómo asignar el color, tamaño y fuentes para el texto
  - Cómo organizar las ventanas: posición, tamaño, etc.
  - Cómo elegir y construir los diálogos de mensaje
  - Cómo usar los mensajes: error, aviso, confirmación, etc.



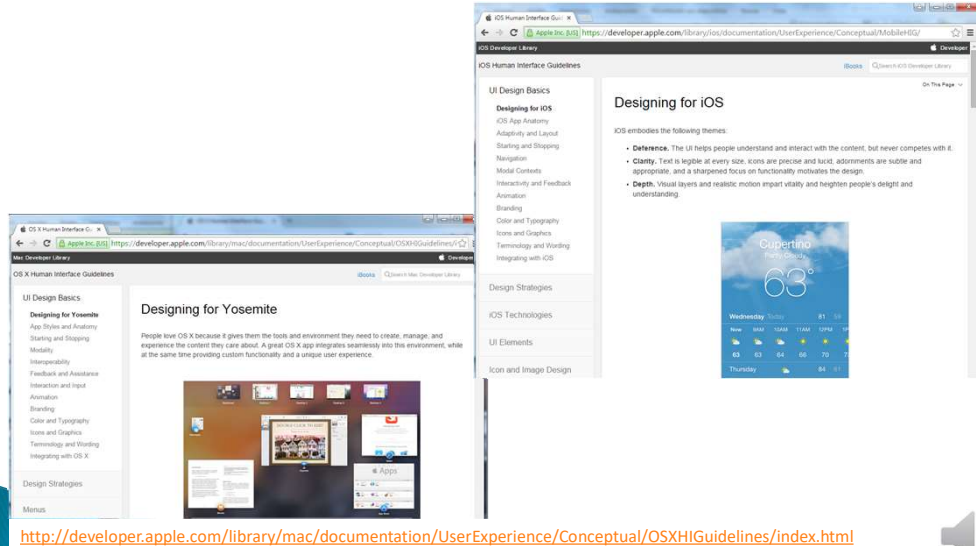
29

Este tipo de guías incluyen principios y reglas específicas, como el tamaño y separación de los controles, cómo elegir los colores, guías sobre la definición de los mensajes de error, etc.

# Guías de estilo

## Guías de estilo comerciales

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG>



<http://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/OSXHIGuidelines/index.html>



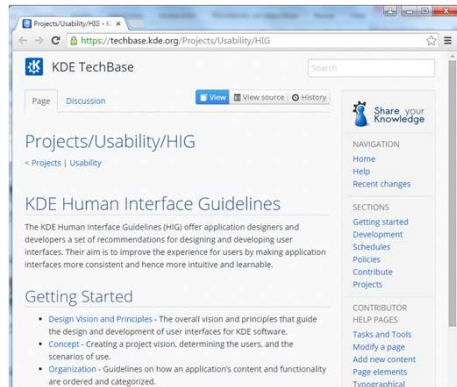
30

Generalmente, todas las plataformas ofrecen a los desarrolladores algún tipo de guía de estilo comercial o documentación.

Aquí vemos los ejemplos de las guías de diseño para los sistemas operativos iOS y Yosemite de Apple,

# Guías de estilo

## Guías de estilo comerciales



<https://techbase.kde.org/Projects/Usability/HIG>



<https://developer.gnome.org/hig/stable>



31

... y aquí vemos las guías de diseño de interfaces para KDE y para Gnome.

# Guías de estilo

## Guías de estilo personalizadas

- ▶ Guías de estilo personalizadas
  - Consiste en crear una guía de estilo específica para un proyecto
    - Se pueden basar en estándares, principios de diseño, otras guías de estilo, etc.
  - Se define en los primeros momentos del proceso de desarrollo y puede ayudar durante la captura de requisitos y la toma de decisiones
  - Promueven la consistencia a través de la interfaz de usuario
  - Si se utiliza en toda la organización, ayuda a crear una imagen corporativa
  - Se pueden hacer a medida dependiendo de las circunstancias del proyecto (características de los usuarios, sus tareas y entorno)



32

Además de las guías de estilo comerciales también podemos encontrar otras más específicas que son las guías de estilo personalizadas. Están relacionadas con la obtención de una imagen de marca o para adaptar un producto a un dominio específico, o para adaptarse a las características y tareas de los usuarios.

Estas guías son más concretas que las comerciales ya que no están creadas para ser aplicadas a todas las posibles aplicaciones de un sistema operativo, sino solo a un subconjunto de ellas.



# Principios de una buena composición

## ► Principios de una buena composición

### 1. Crear grupos naturales

- Tener en cuenta la estructura de la información, crear grupos lógicos (ej. en una base de datos de estudiantes, detalles de la dirección de contacto, cursos en los que está matriculado en la actualidad, expediente académico, etc.)
- Diferentes colores de fondo, líneas de separación, espacio en blanco, fuentes diferentes,... (recordar las Leyes de Gestalt)
- En una IGU, agrupar los controles relacionados (menús, iconos, botones, etc.)

### 2. Separar los componentes de la actividad activa actual

- Enfatizar lo que el usuario está haciendo en este momento
- Por ejemplo, Windows resalta la ventana con la que el usuario está interactuando utilizando un color más brillante en la barra del título y situándola encima de las otras ventanas



33

Hasta ahora hemos estado hablando de principios generales para crear una interfaz usable, pero no hemos hablado todavía de cómo disponer los controles en la interfaz de manera que no ayuden a alcanzar esos objetivos. Para ello vamos a basarnos en una serie de principios que nos van a facilitar obtener vistas con una configuración óptima.

En primer lugar crearemos grupos naturales, no solo en los formularios, sino también en las opciones de los menús, en las barras de herramientas y en general en todos los controles que aparezcan en las ventanas.

En segundo lugar, hay que enfatizar los componentes de la tarea que el usuarios está llevando a cabo. De esta manera se consigue que el usuario sepa en todo momento lo que está haciendo, o que retome la tarea en caso de interrupción.

## Principios de una buena composición

3. Resaltar los componentes importantes
  - Subrayar los componentes más importantes (ej., los botones de emergencia generalmente son rojos), pero nada más
  - ¿Cómo? Color, tipo, tamaño, posición en la pantalla, animación...
  - Combinar efectos refuerza el resultado (ej. alarmas sonoras y luces intermitentes)
4. Utilizar el espacio en blanco de forma efectiva
  - Con frecuencia el espacio en blanco es más efectivo que las líneas
  - Esto puede implicar que la información se divida en varias pantallas



34

En tercer lugar resaltar los componentes más importantes, por ejemplo utilizando fuentes o colores diferentes, o aplicando algún efecto

En cuarto lugar hay que utilizar el espacio en “blanco” de forma efectiva. Conviene aclarar que el espacio en blanco no se refiera al color, sino a lo que sería el espacio vacío de la interfaz. Esto está relacionado con una buena estructura y agrupación de controles. Si separamos determinados elementos estamos indicando al usuario que están relacionados con diferentes conceptos o acciones.

## Principios de una buena composición

5. Hacer que los controles sean visibles
  - Los controles en pantalla deberían sugerir cuáles son sus funciones, aprovechando el conocimiento que tienen los usuarios de otros IU y del mundo en general
6. Equilibrar la estética y la usabilidad
  - Hay que mantener un equilibrio entre los diseños atractivos y la usabilidad. Por ejemplo, un fondo vivo puede resultar atractivo, pero podría cansar la vista



35

En quinto lugar, hay que hacer que los controles sean visibles. Esto está relacionado con el principio de Visibilidad que hemos visto al comienzo del tema.

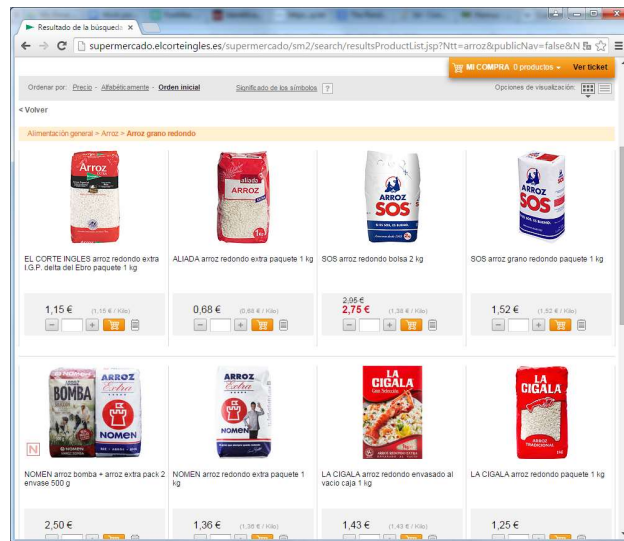
El usuario debe conocer que controles hay en la vista para interactuar con ellos. El diseño de estos elementos debería ser similar a otras aplicaciones de manera que se pueda aprovechar el conocimiento previo del usuario.

Y en sexto lugar hay que balancear la estética y la usabilidad. Hay que llevar cuidado con el contraste ya que un mal uso del mismo puede hacer que el texto no sea legible o que algunos controles nos pasen desapercibidos.

Igualmente hay que elegir el fondo de manera que no distraiga al usuario en su interacción con la aplicación. También tenemos que recordar que hay usuarios con diferentes capacidades, por lo que conviene no utilizar únicamente el color para informar al usuario de las opciones habilitadas y deshabilitadas, acciones no permitidas, elementos no disponibles, etc.

# Principios de una buena composición

1. Crear grupos naturales
2. Separar los componentes de la actividad activa actual
3. Resaltar los componentes importantes
4. Utilizar el espacio en blanco de forma efectiva
5. Hacer que los controles sean visibles
6. Equilibrar la estética y la usabilidad



En este ejemplo vemos cómo se han aplicado estos principios:

1. Hay una cuadrícula donde cada celda contiene una foto y la descripción de un producto, junto con los controles para comprarlo. Los controles asociados a comprar un producto están juntos (añadir o quitar unidades, comprar, y poner en la lista de la compra). Los controles están peligrosamente lejos de su producto y cerca del producto de abajo. Sin embargo, las líneas aplican un cierre que hacen que quede claro quién va con quién.
2. El resumen de la compra (el carro de arriba a la derecha) siempre está visible, aunque hagamos scroll. Así, podemos saber cuántos productos hemos comprado en todo momento.
3. Vemos que tanto el botón de comprar como el carro de la compra tienen un color que resalta con el resto de la interfaz.
4. Se usan líneas sutiles y espacio para separar cada producto y sus controles asociados, creando una cuadrícula.
5. Los controles son bastante auto-explicativos. Al pulsar + se incrementa el contador de productos, al pulsar - se decrementa, el botón con el carrito introduce el número de productos seleccionados al carrito, etc. Las fotografías ayudan a saber exactamente qué se está comprando.
6. En general el diseño es bastante minimalista, con un fondo blanco plano. No hay una gran variedad de colores en los elementos de la interfaz, aparte de las fotografías, que son de muy buena calidad.

## Principios de una buena composición



37

Vamos a analizar ahora los ejemplos que tenemos aquí, que corresponden a dos máquinas expendedoras de bebidas calientes.

1. Respecto al los grupos: los botones están agrupados por columnas (cafeinado, descafeinado y otros). Los soft-buttons están pegados a la pantalla. La imagen correspondiente a cada botón está claramente identificada
2. El principio de separar los componentes de la actividad activa actual es mejorable ya que: el elemento con el foco de interacción no está marcado. Quizá estaría bien que el monedero tuviera una luz que estuviera encendida cuando se espera que el usuario introduzca el dinero. Tampoco hay ninguna indicación cuando la máquina devuelve el cambio. Una vez que se ha seleccionado el café, ya no importa, puesto que la interacción está casi acabada (sólo queda recoger el vaso).
3. El elemento más importante está resaltado, puesto que los botones están claramente marcados e iluminados. El monedero está mejor identificado en el diseño de la izquierda (aunque en el de la derecha tampoco hay problema para encontrarlo)
4. El espacio en blanco se utiliza de forma efectiva: hay espacios entre los botones que ayudan a separar cada botón. El monedero y la caja de devoluciones están más separados. El botón de + y – para seleccionar la cantidad de azúcar está cerca de la pantalla.
5. Los controles importantes son visibles, ya que el diseño de los botones deja claro dónde hay que pulsar (los ejemplos que vimos de las botoneras de ascensores donde no queda claro qué es el botón y qué es el indicador son contraejemplos de este principio)
6. En cuanto al equilibrio entre la estética y la usabilidad, quizá el diseño en negro del embellecedor del monedero reduce su visibilidad. Un diseño como el de la izquierda mejoraría la visibilidad. El fondo es sobrio y negro, lo que hace resaltar los elementos interactivos.

## Diseñando una interfaz gráfica de usuario

- ▶ Componentes de una IGU
  - Ventanas, cuadros de diálogo, pestañas...
  - Menús, barras de herramientas, botones...
  - Radio buttons, check boxes, list boxes, cajas de texto...
- ▶ ¿Cómo elegimos el control adecuado? ¿Cómo utilizamos el control de forma efectiva? ¿Cómo combinamos los controles?



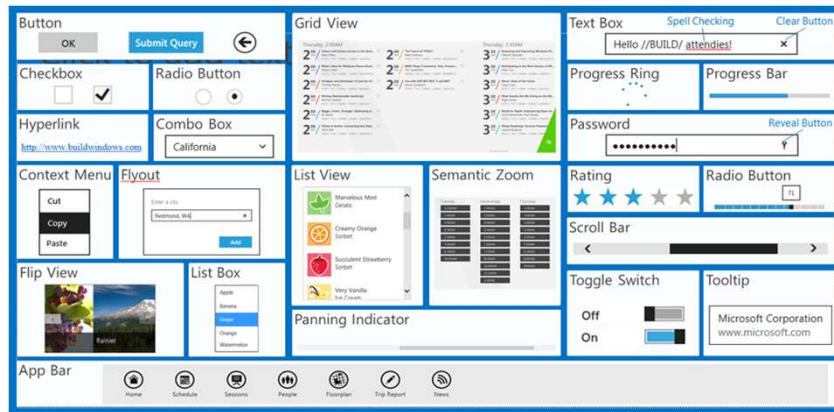
38

Bien, ya conocemos los componentes propios de una IGU y conocemos los principios y guías de diseño que necesitamos seguir para conseguir una buena interfaz. Ahora nos falta decidir qué componentes vamos a utilizar. ¿Cómo elegimos los controles adecuados? ¿Cómo los combinamos?

Vamos a verlo.

# Diseñando una interfaz gráfica de usuario

- ▶ Cada SO tiene su propio conjunto de controles



Windows 8 Metro



39

En primer lugar deberíamos elegir los controles usados habitualmente en el sistema operativo sobre el que va a correr nuestra aplicación. Recordad que cada sistema operativo tiene sus propias guías de estilo. Tenemos diferentes conjuntos de controles para Windows...

# Diseñando una interfaz gráfica de usuario

- ▶ Cada SO tiene su propio conjunto de controles



Mac OS X



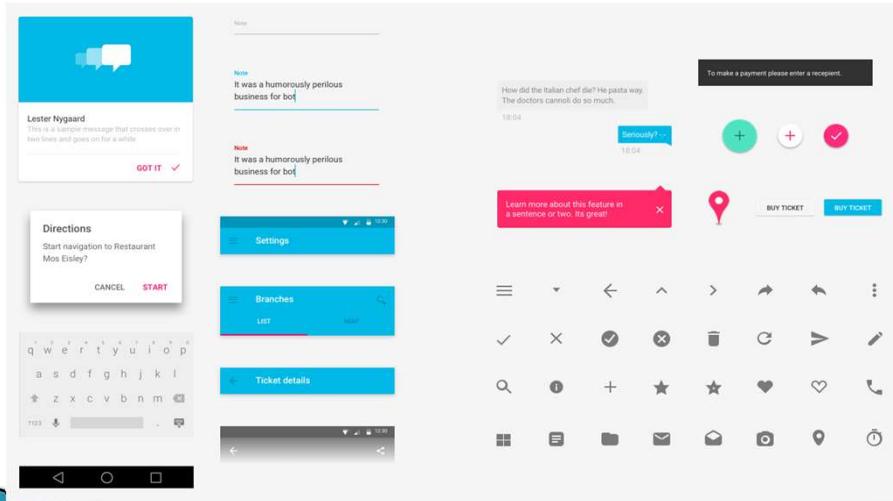
40

...para Mac...



# Diseñando una interfaz gráfica de usuario

- ▶ Cada SO tiene su propio conjunto de controles



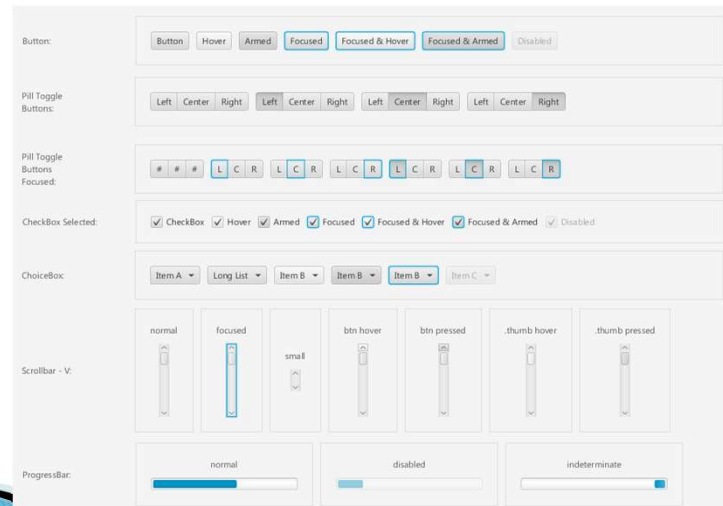
Android Lollipop

41

...para Android

# Diseñando una interfaz gráfica de usuario

- ▶ Cada SO tiene su propio conjunto de controles



JavaFX Modena



42

o para Java.

## Diseñando una interfaz gráfica de usuario

- ▶ Hay que traducir el diagrama de contenidos construido durante la fase de diseño conceptual (y los otros elementos recopilados durante la fase de análisis de requisitos) a una interfaz
- ▶ Pasos:
  1. Elegir los controles para estructurar la interacción
  2. Elegir los controles para controlar la interacción
  3. Elegir los controles para introducir la información



43

Una vez hemos decidido el conjunto de controles a utilizar, vamos a decidir cuáles utilizamos en función de nuestras necesidades. Así, elegiremos los controles adecuados para estructurar la interacción, posteriormente para controlar la interacción y por último los controles para introducir información.

# Diseñando una interfaz gráfica de usuario

## Elegir los controles para estructurar la interacción

- ▶ La mayoría de los IGU se organizan utilizando un contenedor de alto nivel, como las ventanas, cuadros de diálogo o pestañas
- ▶ Ventana Principal
  - Contiene un marco, una barra de título, menús, barras de scroll, etc.
  - Generalmente representan objetos principales de tarea (el documento en un procesador de texto, o la fotografía que se está editando)
  - Normalmente sólo hay unas pocas ventanas principales a las que vuelve el usuario constantemente
  - En ocasiones hay una ventana principal que actúa como lanzadera



44

En general, estructuraremos nuestra interfaz utilizando varias ventanas y cajas de diálogo.

Las ventanas principales contendrán un marco, un título, menús, barras de scroll, etc. Normalmente en estas ventanas se incluirán las acciones que se habían identificado en los contenedores, y llevarán a cabo las tareas principales.

En la actualidad, siguiendo la estética minimalista, muchas aplicaciones se diseñan sin marco.

En otras ocasiones la ventana principal no lleva a cabo ninguna tarea, simplemente actúa de lanzadera, de manera que solo presenta un menú de selección.

# Diseñando una interfaz gráfica de usuario

## Elegir los controles para estructurar la interacción

### ▶ Ventanas secundarias

- Proporciona funcionalidad adicional y apoyo al usuario
- Cuadros de mensaje
  - Muestran mensajes, generalmente sobre algún problema que el usuario tiene que solucionar antes de seguir trabajando
  - Los cuadros de mensaje normalmente son modales (se bloquea la interacción con el resto de la aplicación hasta que se cierran)
  - Los cuadros de mensaje no modales permiten al usuario interactuar con otras ventanas de la aplicación



45

Las ventanas secundarias suelen proporcionar funcionalidad adicional o de apoyo, como los cuadros de mensaje o diálogos.

Cuando el usuario necesita interactuar con la ventana principal y la secundaria al mismo tiempo utilizaremos ventanas o diálogos no modales, lo que en los enlaces del diagrama de contenidos vendrá representado por una doble flecha. Sin embargo, si necesitamos una respuesta de la ventana secundaria para que se pueda continuar la ejecución se utilizarán ventanas o cajas de diálogo modales.

.

# Diseñando una interfaz gráfica de usuario

## Elegir los controles para estructurar la interacción

### ► Pautas para los cuadros de mensaje de Windows 8

#### Dos and don'ts

- Use message dialogs to convey urgent information that the user must see and acknowledge before continuing. An example is, "Your trial period for advanced features has expired."
- Use message dialogs to present blocking questions that require the user's input. A blocking question is a question where the application cannot make a choice on the user's behalf, and cannot continue to fulfill its value proposition to the user. A blocking question should present clear choices to the user. It is not a question that can be ignored or postponed.
- Use message dialogs to ask for explicit action from the user or to deliver a message that is important for the user to acknowledge. Examples of usages of dialogs are the following:
  - The user is about to permanently alter a valuable asset
  - The user is about to delete a valuable asset
  - The security of the user's system could be compromised
- Use custom dialogs when the app or the system must invest a significant amount of time in the ensuing actions such that an accidental dismiss would be detrimental to the user's confidence.
- All dialogs should clearly identify the user's objective in the first line of the dialog's text (with or without a title).
- Don't use message dialogs when the app needs to confirm the user's intention for an action that the user has taken. Instead, a flyout is the appropriate surface.
- Don't use message dialogs for errors that are contextual to a specific place on the page, such as validation errors (in password fields, for example), use the app's canvas itself to show inline errors.



46

Aquí podéis ver una guía para definir cuadros de mensajes en Windows 8. Como podéis ver, no solo dice que se debe hacer, sino también que no se debe hacer.

# Diseñando una interfaz gráfica de usuario

## Elegir los controles para estructurar la interacción

### ► Pautas para los cuadros de mensaje de Windows 8

#### Additional usage guidance

All message dialogs should clearly identify the user's objective in the first line of the dialog's text. The following guidelines explain how to use the "title" and "content" fields of the message dialog to convey information effectively.

- **Title (main instruction, optional)**
  - Use a short title to explain what people need to do with the dialog. Long titles do not wrap and are truncated.
  - If you're using the dialog to deliver a simple message, error or question, you can optionally omit the title. Rely on the content text to deliver that core information.
  - Make sure the title relates directly to the button choices.
- **Content (descriptive text)**
  - Present the message, error, or blocking question as simply as possible without extraneous information.
  - When a title is used, use the content area to provide more detail or define terminology. Don't repeat the title with slightly different wording.
- **Buttons**
  - Use buttons with text that identifies specific responses to the main instruction or content. An example is, "Do you want to allow AppName to access your location?", followed by "Allow" and "Block" buttons. Specific responses can be understood more quickly, resulting in efficient decision making.
  - Avoid using generic patterns such as "OK/Cancel".
  - Specify the default button, which should be the action you most want the user to take. In the above example, "Allow" is the default choice.
    - If not specified, the default is the leftmost button.
    - Put the safest, most conservative choice on the rightmost position. In the above example, "Block" is the rightmost choice as it is more conservative.
- **Color**
  - Message dialog backgrounds are always white. The primary colors of the app that owns the dialog are used for controls.



47

También nos proporciona información sobre cómo se debe configurar el diálogo.

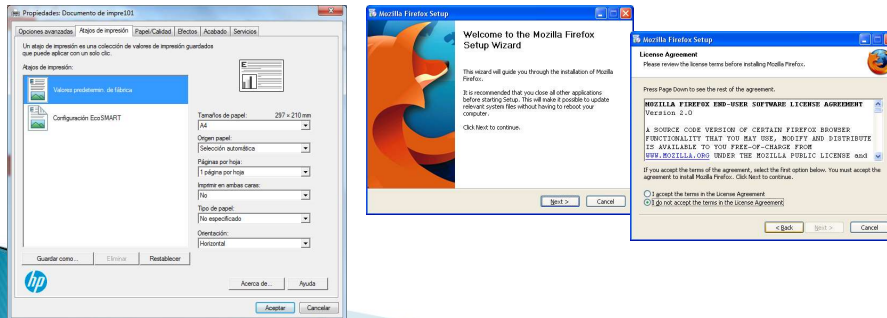
# Diseñando una interfaz gráfica de usuario

## Elegir los controles para estructurar la interacción

### ► Ventanas secundarias

#### ◦ Cuadros de diálogo

- Son invocados por el usuario
- Se suelen utilizar para pedir información adicional
- Pueden ser pantallas complejas con campos de texto, botones, etc.
- Un asistente (*Wizard*) es una serie de cuadros de diálogo en un orden determinado que guía al usuario en la realización de tareas complejas



Otro tipo de ventanas secundarias son los diálogos.

Este tipo de ventanas se utilizan para añadir información, ya sea en forma de texto, eligiendo opciones dentro de un menú o pulsando algún botón...

Por ejemplo, para loguearse en un sistema, para asignar parámetros de configuración, etc.

Los cuadros de diálogo también se pueden utilizar para dirigir al usuario en la realización de tareas compleja.

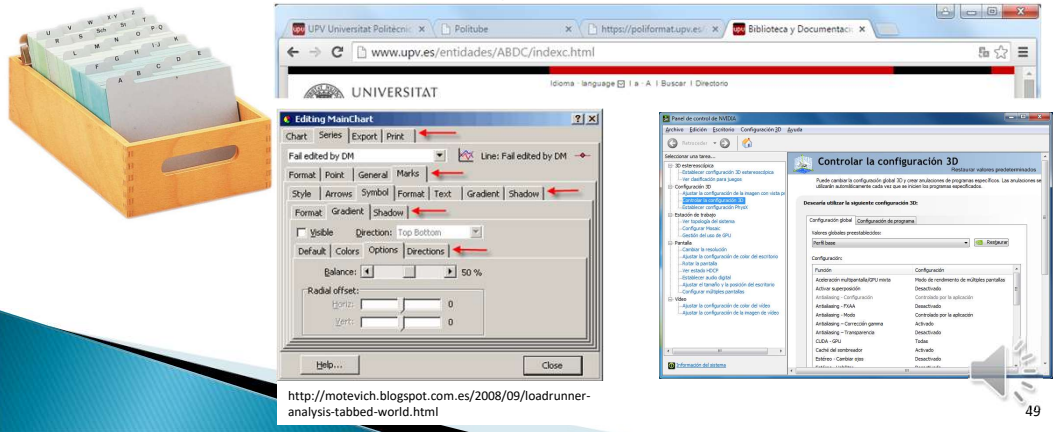


# Diseñando una interfaz gráfica de usuario

## Elegir los controles para estructurar la interacción

### ► Pestañas

- Son útiles para clasificar las propiedades de los objetos de tarea representados en una ventana
- La información en cada pestaña debería ser independiente
- Retos: Utilizar demasiadas pestañas. Los usuarios pueden olvidar completar o pasar por alto la información en alguna de ellas



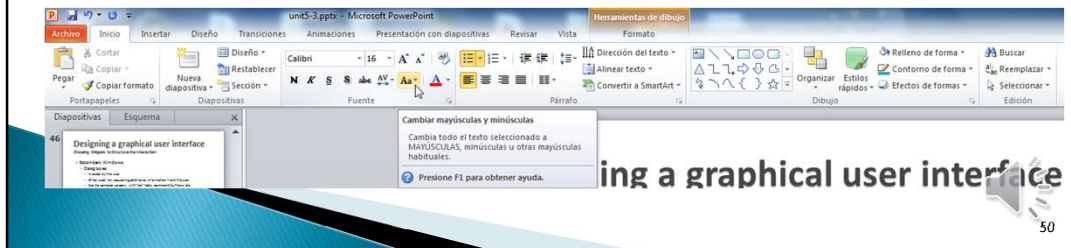
Otra forma de estructurar la información es utilizando pestañas. Las pestañas son útiles si no hay muchas opciones pero, si como en la ventana de la izquierda, hay muchas, es mejor utilizar menús anidados.

Las pestañas también pueden dar problemas con los lectores de pantallas, por tanto, si la aplicación debe ser accesible, puede ser más adecuado utilizar botones o menús.

# Diseñando una interfaz gráfica de usuario

## Elegir controles para controlar la interacción

- ▶ Elegir controles para controlar la interacción
  - Permiten al usuario controlar la interacción
    - Menús. Ya estudiados en una unidad anterior
    - Barras de herramientas
      - Complementan la jerarquía de menús
      - Contienen una serie de comandos usados frecuentemente, representados por iconos
      - La utilidad de un botón se explica en un *tooltip*
      - Hay varias barras de herramientas clasificadas en grupos lógicos
      - Puede ser complicado seleccionar los iconos apropiados
      - Las cintas combinan menús y barras de herramientas



Posteriormente a estructurar la interacción debemos elegir los controles adecuados para controlar la interacción.

Para ello podemos utilizar menús o barras de herramientas (que ya ese estudiaron en el apartado de estilos de interacción), y son elementos útiles para seguir el principio de diseño de reconocer en lugar de recordar.

En este caso es crucial seleccionar iconos familiares al usuario y agrupar las funciones dependiendo del objeto sobre el que actúan o el tipo de acción que realizan.

# Diseñando una interfaz gráfica de usuario

## Elegir controles para controlar la interacción

### ► Elegir controles para controlar la interacción

#### ◦ Botones

- Normalmente usados para controlar la operación de los cuadros de diálogo
- Es importante usar etiquetas comprensibles
  - Windows: Aceptar (botón por defecto), Cancelar
  - Mac OS X: Cancelar, Imprimir
- Tamaño: para agrupar visualmente los botones deberían tener la misma forma y tamaño. Botones en una fila pueden tener diferentes anchuras



51

Los botones también son elementos utilizados para controlar la interacción.

La posición y nombre de los botones se deben elegir de forma que sigan el principio de consistencia

Por ejemplo, en la Plataforma Windows, el botón “**Cancelar**” está siempre a la derecha del botón “**Aceptar**”.

Las guía de los sistemas Apple (MacOS), dicen que “Un botón que inicia una acción debe estar en el extremo de la derecha. El botón **Cancelar** debe estar a la izquierda de éste”. De este modo, para los usuarios de MacOS el botón “**Cancelar**” está a la izquierda del botón “**Aceptar**”.

A la hora de nombrar los botones, suele ser mejor explicar qué hace en lugar de usar una etiqueta genérica como “OK”, es decir utilizar, siempre que sea posible un nombre en lugar de “Si” o “No”, ya que dan sentido a los botones independientemente del contexto. Por otra parte el color de los botones nos puede ayudar a dotarlos de significado “positivo” o “negativo”

# Diseñando una interfaz gráfica de usuario

## Elegir controles para introducir información

### ► Elegir controles para introducir información

- Estudiados en una unidad anterior
- *Radio buttons* and *check boxes*
  - Preguntas a considerar:
    - ¿Cuántas opciones son aceptables para que puedan ser utilizadas? ¿Cómo deberían agruparse y disponer en la pantalla? ¿Cómo se deberían ordenar en la pantalla? ¿Cómo se deberían etiquetar las opciones?

Más información en:

<https://uxplanet.org/radio-buttons-ux-design-588e5c0a50dc>

<https://www.nngroup.com/articles/checkboxes-vs-radio-buttons/>



52

Por ultimo vamos a ver algunos criterios para elegir los controles para introducir información.

Recordad que los botones de radio (*radio buttons*) se utilizan cuando hay una lista de dos o más opciones *mutuamente excluyentes*. Si no son excluyentes, es decir, si el usuario puede elegir cero, una o varias opciones, elegiremos “*check boxes*”. En ambos casos, la lista de opciones debe tener un *orden lógico*, por ejemplo dependiendo de la probabilidad de ser seleccionado, o de operaciones más sencillas a más complejas, o de menor a mayor riesgo.

En general, el orden alfabético no suele ser adecuado porque depende del idioma.

Las opciones deben ser comprensibles y claramente distintas, siempre deben tener una opción por defecto, que debería ser la más probable o más conveniente.

Es más adecuado mostrar la lista verticalmente y si es posible utilizar “*radio buttons*” en lugar de menús desplegables

En el caso de que tengas menos de 7 opciones se debe considerar utilizar *radio buttons*. Utiliza los *radio buttons* para cambiar el valor de los parámetros, no para ejecutar comandos.

# Diseñando una interfaz gráfica de usuario

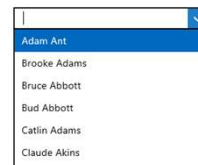
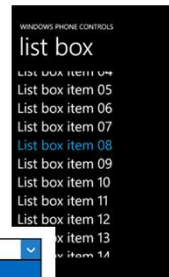
## Elegir controles para introducir información

### ◦ List Boxes

- Permiten al usuario elegir entre un gran número de opciones
- Selección simple/múltiple
- Desplegable/permanente, dependiendo del espacio disponible
- Son más flexibles que *check boxes*/*radio buttons*
- Se deberían utilizar valores por defecto adecuados para acelerar el uso del programa
- Se pueden combinar con cajas de texto, que pueden actuar como filtro (*combo box*)

Más información en:

<https://www.nngroup.com/articles/drop-down-menus/>  
<https://uxplanet.org/ux-design-drop-downs-in-forms-e6943ec30037>



53

No utilices cajas de listas (list boxes) cuando tengas menos de 7 opciones. En ese caso es mejor utilizar radio buttons o checkboxes. Además, si tienes más de 15 opciones considera utilizar otro tipo de controles, como los combo boxes, que permiten introducir el texto a buscar o parte de él en el caso de que no conozcamos el nombre exacto de la opción que estamos buscando. Por ejemplo, son una buena opción para la selección de países.

Conviene evitar los cuadros desplegables para los datos que son muy familiares para el usuario, como el día, mes o año de su nacimiento. Tener que encontrar estas opciones en un menú largo es tedioso y puede crear aún más trabajo para los usuarios.

También es importante mantener a la vista la etiqueta o la descripción del menú que esté abierto. Los títulos de menú recuerdan a los usuarios lo que están eligiendo. Los menús o cuadros desplegables deben admitir tanto la entrada por ratón como por teclado, lo que nos permite movernos entre las opciones.

# Diseñando una interfaz gráfica de usuario

## Elegir controles para introducir información

- Cajas de texto
  - Es el control más flexible para introducir información
  - No es adecuado para introducir información con formato (fechas, códigos postales, opciones de una lista predefinida...)
  - Pautas
    - El tamaño de la caja de texto debería indicar la cantidad de información esperada (tanto número de líneas como longitud de la línea)
    - Si el usuario puede introducir una gran cantidad de texto, utilizar una caja de texto multilínea con barras de scroll



54

Las cajas de texto o campos de entrada son los controles más flexibles para introducir información. El tamaño de la caja debe ser proporcional a la entrada esperada. Los campos de entrada deben destacarse e indicar a los usuarios que pueden introducir información. Por tanto debería haber un cierto contraste entre la zona de introducción de información y el área que la rodea. Los campos pueden tener uno de los siguientes estados: valor por defecto, foco, error o deshabilitado. Cada estado se debe diferenciar claramente de los demás de manera que los usuarios los puedan percibir de un vistazo. Es muy importante que se utilice un diseño que permita diferenciar a los campos de información de los demás controles, por ejemplo, evitar que se parezcan a los botones. El aspecto que tengan los controles nos debería dar una idea de cómo se interactúa con él, y por tanto un buen diseño nos permitirá mejorar el *affordance* de la interfaz.

Cada campo de texto también debe tener una etiqueta que indique al usuario qué información se debe introducir. Estas etiquetas también son necesarias para obtener interfaces accesibles. Deben ser concisas, cortas y descriptivas con únicamente una o dos palabras.

## Bibliografía

- ▶ D. Stone, C. Jarrett, M. Woodroffe. User Interface Design and Evaluation. Morgan Kaufmann, 2005.
  - Capítulos 9, 14, 16
- ▶ [UX Planet. https://uxplanet.org/](https://uxplanet.org/) (Free. Search options after sign-in)
- ▶ [Nielsen Norman Group. https://www.nngroup.com/articles/](https://www.nngroup.com/articles/)



55

Y con la bibliografía y un par de enlaces, por si queréis ampliar la información, terminamos el tema de diseño físico de interfaces.