

## Tema IX

# Conceptos sobre diseño de interfaces de usuario. (R-1.0)

Programación en Entornos Interactivos.

14 de marzo de 2011

Dpto. Lenguajes y Sistemas Informáticos  
Universidad de Alicante



## Resumen

Preliminares. Principios básicos. Elementos del interfaz de usuario.  
Iconos. Consideraciones generales. Uso del color.



# Preliminares.

- Es importante prestar especial atención al interfaz gráfico de usuario (**GUI**) de la aplicación ya que representa la forma en la que los usuarios se comunican con ella y viceversa.
- *Ante dos aplicaciones similares, un usuario optará por aquella con un interfaz más sencillo de utilizar.*
- **Importante:** El código del interfaz de usuario de una aplicación requiere el 40% o más del tiempo de su desarrollo.
- Es útil hacer varias versiones de una aplicación, o parte de una aplicación, cada una de ellas con un interfaz distinto, y quedarnos con aquella que sea *mejor*, más sencilla de usar para el usuario.

# Criterios de evaluación de interfaces de usuario. (I)

Disponemos de cinco criterios para que un usuario pueda evaluar la calidad del interfaz de una aplicación:

**Concisión:** La cantidad de acciones necesarias para realizar una transacción con él tiene una cota superior pequeña.

**Expresividad:** Cuándo puede ser usado fácilmente para llevar a cabo una gran variedad de acciones, algunas incluso no pensadas inicialmente por el programador.

**Facilidad:** De uso, es inversamente proporcional a la cantidad de cosas que el usuario debe recordar para poder emplear este interfaz.

## Criterios de evaluación de interfaces de usuario. (II)

**Transparencia:** Se refiere a la cantidad de cosas que debe recordar un usuario acerca del estado de sus datos, problema o del programa cuando está empleando un interfaz sobre él.

**Programabilidad:** Indica lo fácilmente que puede ser manipulado por otras aplicaciones, por ejemplo para automatizar tareas repetitivas.

# Principios básicos.

Podemos centrar nuestra atención en la parte del interfaz de la aplicación en estos aspectos:

- Consistencia del interfaz.
- Integridad estética.
- Control por parte del usuario de la aplicación.
- Evitar en lo posible el uso de *modos* en la aplicación.
- Información por parte de la aplicación al usuario.
- Posibilidad de *deshacer* acciones que pueda realizar el usuario.

# Elementos del interfaz de usuario.

Es conveniente conocer las características de uso y comportamiento de determinados elementos que forman parte del interfaz de usuario de una aplicación: Nos centraremos en los siguientes:

menús	widgets/controles	comportamiento
ventanas	iconos	lenguaje
diálogos		color

## Menús. (I)

- Evitar el uso de menús jerárquicos. Si se usan procurar que sean sólo de un nivel.
- Hacer uso del carácter elipsis: '...' cuando necesitemos más información del usuario.
- Elegir el agrupamiento mejor de las opciones dentro de un menú. Estos grupos se separan unos de otros por un separador.
- No usar menús creados dinámicamente cuando no sepamos el número exacto de opciones que tendrán.



## Menús. (II)

- Los menús de opciones `-option-menu-` que añadamos a un diálogo:
  - ① Se pueden emplear para simular botones de radio `-radio-buttons-` cuando su número es mayor o igual a 5.
  - ② No se deben emplear para simular un grupo de `check-buttons`.
- Un menú que represente una paleta de herramientas podría ser un menú *arrancable* `-tear-off menu-`.

## Menús. (III)

Si la aplicación utiliza la metáfora del '*documento*' como entidad con la que un usuario trabaja, debería proporcionar, al menos, estos menús:

**Fichero** En este menú se agruparán las opciones relativas al documento como un todo:

- Nuevo, Abrir
- Cerrar, Guardar, Guardar como... , Recuperar
- Ajustar página, Imprimir
- Salir

**Edición** En este menú se agruparán las opciones relativas a la edición del contenido de un documento:

- Deshacer / Rehacer
- Cortar, Copiar, Pegar, Borrar, Seleccionar todo.

Y por último...

Y no por ello menos importante:

**Ayuda** En este menú estarán las opciones que proporcionarán ayuda al usuario o le dirán dónde y cómo encontrarla. La mayoría de aplicaciones aprovechan para añadir una opción conocida como **Acerca de...**

Esta opción permite al usuario conocer datos como:

- El **nombre** exacto de la aplicación que emplea.
- La **versión** de la aplicación
- La **empresa** que la crea, etc. . .

# Ventanas.

- Dejar su gestión al gestor de ventanas –movimiento, iconización, etc. . . –.
- Si se anula la funcionalidad de algún control, avisar al usuario.
- Las partes seleccionadas de un documento sólo se deben mostrar en la ventana activa o de primer plano de la aplicación, o mostrarlas de forma diferente en las ventanas de segundo plano.
- Si nuestra aplicación permite que sobre un documento existan varias **vistas**, puede ser útil permitir al usuario usar *paneles* dentro de una ventana.

# Diálogos. (I)

- Son un tipo especial de ventanas que permiten:
  - Que la aplicación suministre información al usuario.
  - Que el usuario suministre información a la aplicación.
- Distinguiremos tres tipos de diálogos:

**No modales:** Permiten al usuario interactuar con otras ventanas de la aplicación aun cuando están abiertos. **No** deben proporcionar un **botón para cerrarlos**, el usuario los debe cerrar usando alguno de los controles que tenga la ventana en la que se muestran.

## Diálogos. (II)

- Modales:** No permiten al usuario interactuar con otras ventanas de la aplicación hasta que éste no cierra el diálogo. Deben cerrarse pulsando algún botón que coloquemos para ello, normalmente tendremos dos de estos botones: **Aceptar** y **Cancelar**.
- Alertas:** También llamadas '*cajas de mensaje*', son un tipo especial de diálogo modal, informan al usuario mediante un texto breve de una determinada situación, y/o recogen ninguna o muy poca información mediante uno o varios botones que el usuario puede pulsar para cerrarlas.

# Controles o Widgets. (I)

Debemos conocer todos los que nos proporcione el entorno gráfico que utilizemos. Algunos de ellos son:

**Botones:** Hay de distinto tipo –radio, check, push, toggle, etc. . . –.

- La etiqueta asociada a un botón debe ser una sola palabra y ser un verbo.
- Asociados a un diálogo existe lo que se llama el **botón por defecto**, representa la acción más probable siempre que ésta *no sea peligrosa*. Gráficamente se resalta de alguna manera sobre los otros botones.

**Etiquetas:** Son texto no editable fijado en tiempo de compilación, pero que se puede cambiar en tiempo de ejecución.

## Controles o Widgets. (II)

**Texto editable:** Permiten la entrada de un texto –pequeño– alfanumérico o sólo numérico.

**Texto editable multilínea:** Similar al anterior, pero pensado para textos mayores. Admiten varias líneas, simulando un editor de texto.

**Sliders:** También llamados *diales*, representan gráficamente información numérica.

**Flechas:** Permiten la entrada de valores numéricos dentro de un rango, evitando así errores del usuario.



## Controles o Widgets. (III)

**Listas, Árboles:** Permiten mostrar información de manera lineal y/o jerárquica.

**Combo-box:** Es un widget compuesto de un campo de texto editable y una lista de selección asociada. Su comportamiento debería ser lo menos ambiguo posible.

**Menús:** Existen varios tipos de menús:

- Pull-Down (barra de menús).
- Pop-Up (asociados al botón derecho del ratón) también llamados menú contextuales o *context-menu*.
- Option-Menu, se usan en diálogos, tienen una representación gráfica especial como widget, al pinchar sobre él, despliegan todas sus opciones.

- Deben representar gráficamente distintos objetos de nuestra aplicación.
- Serán una representación sencilla de 'nombres, no de 'verbos' o 'acciones'.
- Debemos emplear:
  - ① Una fuente de luz consistente, p.e. desde la esquina superior izquierda de la pantalla.
  - ② Una paleta de colores común para todos los iconos de la aplicación.
- Diseñarlos teniendo en cuenta características del display, número de colores, dimensiones, etc. . .

# Consideraciones generales. (I)

Es conveniente plantearnos aspectos de 'compatibilidad cultural' de nuestra aplicación, por su uso en diferentes comunidades autónomas o, incluso, otros países.

Algunos de estos aspectos pueden ser:

- Valores culturales:
  - Frases o palabras con un significado especial.
  - Interpretaciones distintas del significado de los colores.
  - Interpretaciones distintas de los iconos, p.e. el del correo.

## Consideraciones generales. (II)

- Diferencias entre lenguajes:
  - Uso de frases hechas para representar determinadas acciones o cosas.
  - Diferencias de tamaño de las frases –en relación al espacio dejado para ellas en un diálogo–.
  - Sentido de dibujo del texto en un diálogo, p.e. de derecha a izquierda o de arriba a abajo, y como afecta a la colocación de iconos y botones en éste.
  - Necesidad de emplear más de un byte por carácter.

## Uso del color.

- Si la aplicación hace uso de elementos que pueden estar representados en color o en blanco y negro, es mejor diseñarlos primero en B/N y luego darles color.
- Hay que tener cuidado especial con el color 'azul' cuando se usa para texto, resulta el más difícil de leer, sin embargo es bueno para pintar cosas de manera no muy visible –por ejemplo una rejilla–.
- Se debe llevar cuidado con el cursor del ratón, de modo que no quede 'invisible' al ponerlo sobre una zona de color similar.
- No se debería forzar a que los widgets tuvieran un color particular, ya que el usuario elige el 'tema' que le interesa en cada caso.

# Ah sí, una cosa más. . .

Emplea siempre la **regla de la menor sorpresa**:

- Siempre haz aquello que menos sorprenda al usuario. . .
- Lamentablemente las personas sólo podemos prestar atención a una sola cosa cada vez.

En la medida de lo posible:

- Evita diseñar un nuevo modelo de interfaz.
- Trata de buscar similitudes en el interfaz entre tu aplicación y otros programas parecidos.
- Siempre que sea posible permite a los usuarios que deleguen funciones del interfaz a otros programas, y si no puedes, trata de emular a estos últimos.