Fundamentos teóricos – Práctica 3

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Principales tipos de contenedores

- Marco (JFrame)
- **Cuadro de Diálogo (JDialog)**
- Panel (JPanel)
- Panel de Scroll (JScrollPane)
- Panel de Pestañas (JTabbedPane)

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

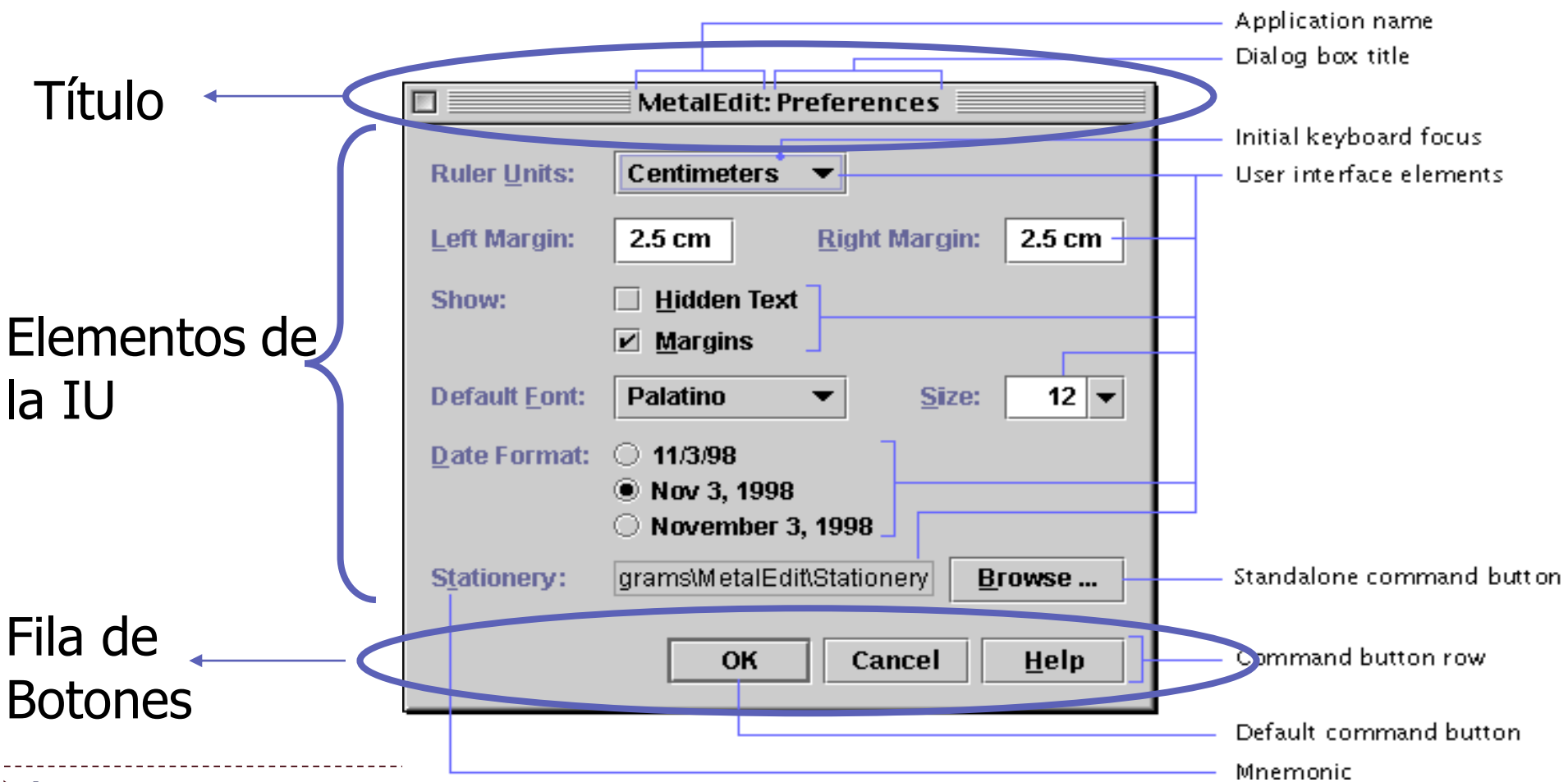
Diálogos (JDialog)

- Generalmente se emplean para recoger datos del usuario y mostrar mensaje de advertencia.
- Derivan de algún otro componente.
- No pueden contener barra de menús.
- Los diálogos pueden ser
  - **Modales.** Impiden que los usuarios interactúen con la aplicación hasta que el diálogo sea cerrado, sin embargo, no impiden la interacción con otras aplicaciones mientras el diálogo está abierto.
  - **No modales**. No impiden a los usuarios interactuar con la aplicación con la que están, o con otras, mientras el cuadro esté abierto.

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

JDialog: Diálogos definidos por el usuario

Elementos de un Diálogo



Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

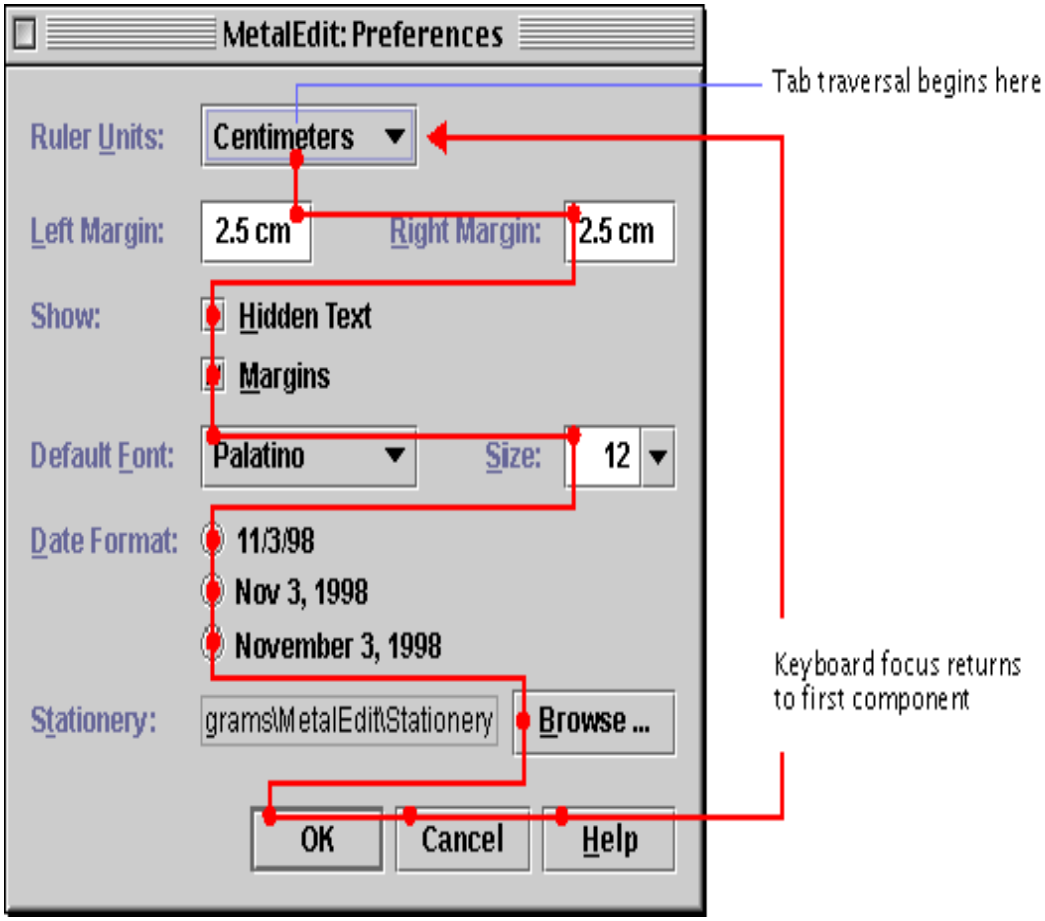
Características Generales

- El título del cuadro de diálogo, mostrado en la barra del título, debe tener la forma:
  - "Nombre de la aplicación: Título del cuadro"
- Hay que incluir mnemotécnicos para todos los elementos excepto el botón por defecto y el botón de cancelación
- Al abrir un cuadro de diálogo el foco debe aparecer sobre el componente sobre el que se espera actuar en primer lugar

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Orden de Tabulación

- El orden de tabulación debería concordar con el orden de lectura (del último pasar al primero)

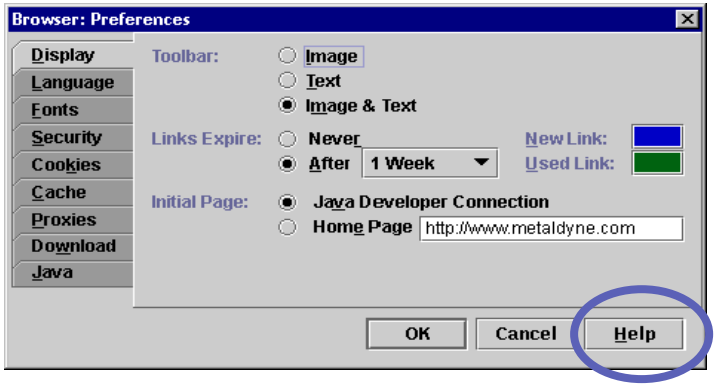


Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Situación de los Botones de Comando en el Cuadro de Diálogo

- Los botones que se aplican a todo el cuadro de diálogo deben colocarse en una fila en la parte inferior del cuadro de diálogo y alineados a la derecha

- Si se emplea botón de Ayuda, para mostrar información adicional sobre el cuadro de diálogo, éste debe ser el último (el de más a la derecha)



Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Botón por Defecto en un Cuadro de Diálogo

- Se activa cuando el usuario presiona Enter y desencadena la ejecución de las acciones asociadas a dicho botón (las realizadas más a menudo).
- Una opción no segura (que ocasione la pérdida de datos) nunca puede ser el botón por defecto.
- El botón por defecto no necesita tener el foco cuando el usuario presiona Intro
- Si el cuadro de diálogo tiene botón por defecto éste debe ser el primer botón de comando en el grupo de botones del cuadro.
- El botón por defecto no lleva mnemónico
- No es obligatorio tener un botón por defecto en cada cuadro de diálogo
- En java: `getRootPane().setDefaultButton(nombreBoton);`

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Botón de Cancelación en un Cuadro de Diálogo

- Se activa al pulsar la tecla Escape y provoca la ejecución de las acciones asociadas al botón identificado como de cancelación
- A diferencia del anterior es necesario implementar este comportamiento, es decir, no existe una única instrucción que permita indicar de forma sencilla cual es el botón de cancelación en un diálogo.
  - Solución: comprobar en cada momento si la tecla pulsada es Escape y si es así invocar el código asociado al botón de cancelación (*gestión de evento de teclado*)

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Diálogos (II)

- En Swing hay varias clases que soportan los diálogos estándar:
  - **JOptionPane**
  - JColorChooser
  - JFileChooser
- Todos ellos son modales

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

JOptionPane

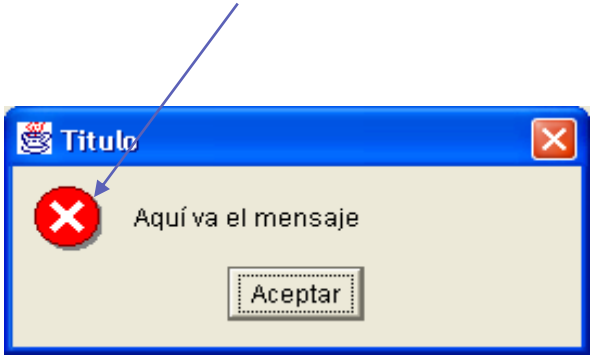
- Permite crear y adaptar varias clases de diálogos, especificando por ejemplo los iconos (propio, ninguno o uno de los cuatro estándar), el título y texto de los diálogos y el texto de los botones. Se puede especificar también donde aparecerá sobre la pantalla.
- Los iconos estándar son: *question*, *information*, *warning* y *error*
- Métodos estáticos principales:
  - *showMessageDialog*
  - *showConfirmDialog*
  - *showInputDialog y*
  - *showOptionDialog* // permite mayor personalización especificando por ejemplo el título de los botones

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

JOptionPane.showMessageDialog

- Muestra un cuadro de diálogo modal con un solo botón etiquetado como 'Aceptar'
- Permite especificar el mensaje, el icono y el título que muestra el diálogo
- Ejemplos de uso:
  - `JOptionPane.showMessageDialog(this, "Mensaje");`
  - `JOptionPane.showMessageDialog(this,"Mensaje","Titulo",JOptionPane.WARNING_MESSAGE);`
  - `JOptionPane.showMessageDialog(this,"Mensaje","Titulo",JOptionPane.ERROR_MESSAGE);`
  - `JOptionPane.showMessageDialog(this,"Mensaje","Titulo",JOptionPane.INFORMATION_MESSAGE);`
  - `JOptionPane.showMessageDialog(this,"Mensaje","Titulo",JOptionPane.QUESTION_MESSAGE);`
  - `JOptionPane.showMessageDialog(this,"Mensaje","Titulo",JOptionPane.PLAIN_MESSAGE);`

Especifica el componente padre (*parent*). Por lo general siempre es un frame (this) y por tanto el cuadro de diálogo siempre aparece desplegado sobre el centro del frame. Sin embargo, se puede especificar como *parent* algún otro componente que esté dentro del frame (ej. jButton) y esto hará que el cuadro de diálogo se despliegue centrado sobre ese componente.

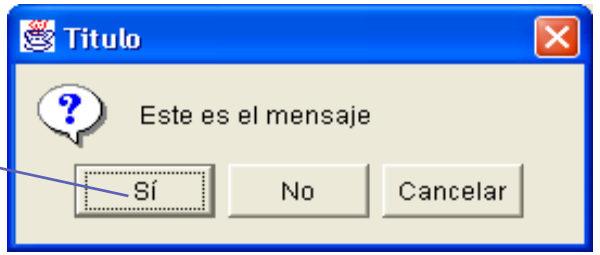


Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

JOptionPane.showConfirmDialog

- Muestra un cuadro de diálogo modal para pedir una confirmación al usuario
- Permite especificar el mensaje, el icono y el título que muestra el diálogo y el número de botones (dentro de un conjunto fijo de ellos)
- Ejemplos de uso:
  - `int resp = JOptionPane.showConfirmDialog(this, "Mensaje");`
  - `int resp = JOptionPane.showConfirmDialog(this,"Mensaje","Titulo", JOptionPane.YES_NO_OPTION);`
  - `int resp = JOptionPane.showConfirmDialog(this,"Mensaje","Titulo", JOptionPane.YES_NO_CANCEL_OPTION);`
  - `int resp = JOptionPane.showConfirmDialog(this,"Mensaje","Titulo", JOptionPane.OK_CANCEL_OPTION);`
  - `int resp = JOptionPane.showConfirmDialog(this,"Mensaje","Titulo", JOptionPane.DEFAULT_OPTION);`

```
if (resp == JOptionPane.YES_OPTION) {
      . . .
}
```

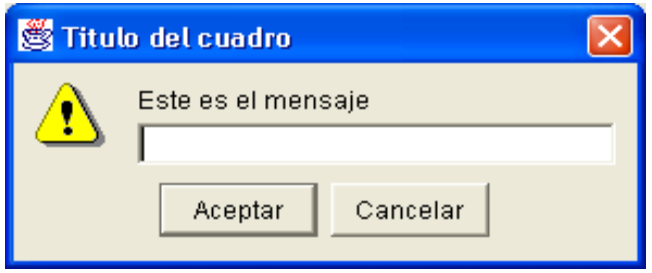


Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

JOptionPane.showInputDialog

- Muestra un cuadro de diálogo modal que toma una cadena introducida por el usuario.
- Este cuadro debe emplearse con bastante prudencia, ya que la validación del dato sólo se puede realizar cuando se cierra el cuadro de diálogo.
- Ejemplos de uso:

```
String valor = JOptionPane.showInputDialog(this, mensaje);
String valor = JOptionPane.showInputDialog(this,"Mensaje","Titulo",JOptionPane.PLAIN_MESSAGE);
String valor = JOptionPane.showInputDialog(this,"Mensaje","Titulo",JOptionPane.INFORMATION_MESSAGE);
String valor = JOptionPane.showInputDialog(this,"Mensaje","Titulo",JOptionPane.WARNING_MESSAGE);
String valor = JOptionPane.showInputDialog(this,"Mensaje","Titulo",JOptionPane.QUESTION_MESSAGE);
String valor = JOptionPane.showInputDialog(this,"Mensaje","Titulo",JOptionPane.ERROR_MESSAGE);
```



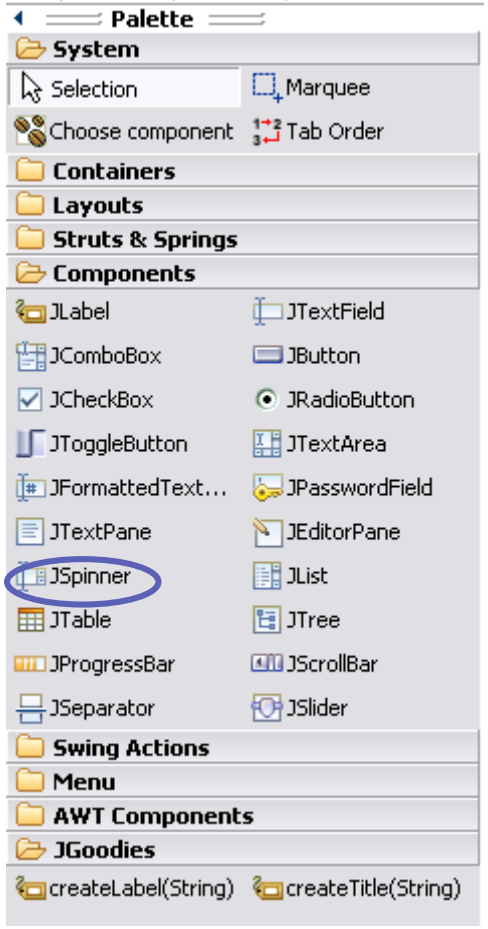
Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Otros Componentes

- Menus (*JMenuBar*)
- Listas (*JList*)
- Deslizadores (*JSlider*)
- **Spinners (*JSpinner*)**

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Spinner (JSpinner)



- Permiten seleccionar un valor entre un rango de opciones posibles.
- Los valores cambian al pulsar los botones de desplazamiento. También se puede introducir un valor directamente.
- Este componente tiene un modelo asociado para la gestión de los datos en el que se indican los valores inferior y superior y el incremento entre valores cada vez que el usuario interactúe con el spinner.
- Estos valores pueden ser cambiados en tiempo de ejecución para adaptar el modelo del spinner a los datos que se deseen recoger en el mismo.

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

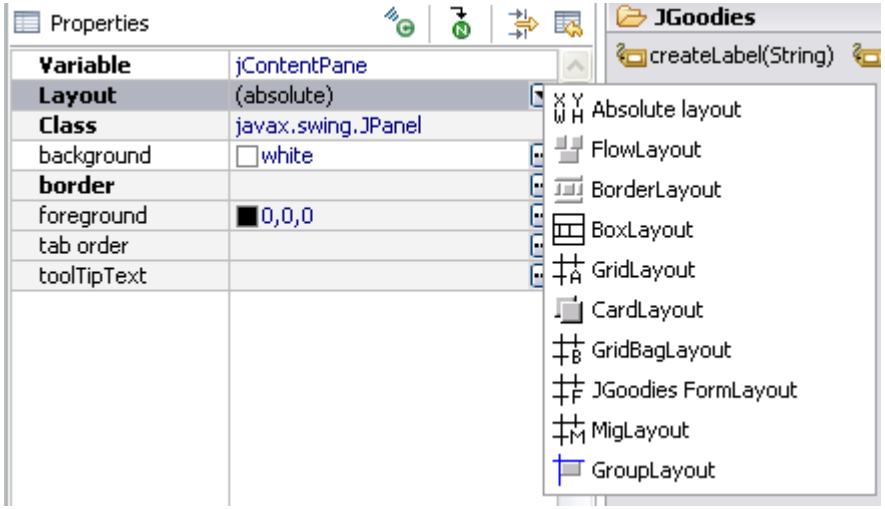
Layouts

- Indican la forma de organizar los componentes dentro de un contenedor, determinando el tamaño y la posición de los mismos.
- Se debe elegir el layout que mejor se adecúe a las necesidades de la aplicación a desarrollar
- Para utilizar un layout:
  - Crear el contenedor
  - Establecer el layout
  - Agregar los componentes al contenedor

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

Tipos de Layouts

- Los más importantes son los siguientes:
  - **FlowLayout**
  - BorderLayout
  - CardLayout
  - GridLayout
  - BoxLayout
  - GridBagLayout
- Por defecto:
  - JFrame, JDialog → BorderLayout
  - JPanel, JScrollPane → FlowLayout



Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente

FlowLayout

- Es el más simple y el que se utiliza por defecto en todos los paneles.
- Los componentes añadidos a un contenedor con FlowLayout se disponen una o más filas, de izquierda a derecha y de arriba abajo.
- Se crean nuevas filas si es necesario.
- Si se modifica el tamaño del contenedor los componentes se redistribuyen.
- Se puede seleccionar la alineación de los componentes respecto al contenedor y el espaciado entre los mismos.
- Propiedades:
  - alignment: izquierda, derecha, centro
  - horizontalgap, verticalGap

Ana Belén Martínez Prieto
Mª del Carmen Suárez Torrente