

INTRODUCCIÓN A WindowBuilder

Ingeniería del Software

Práctica 5

Introducción WindowBuilder

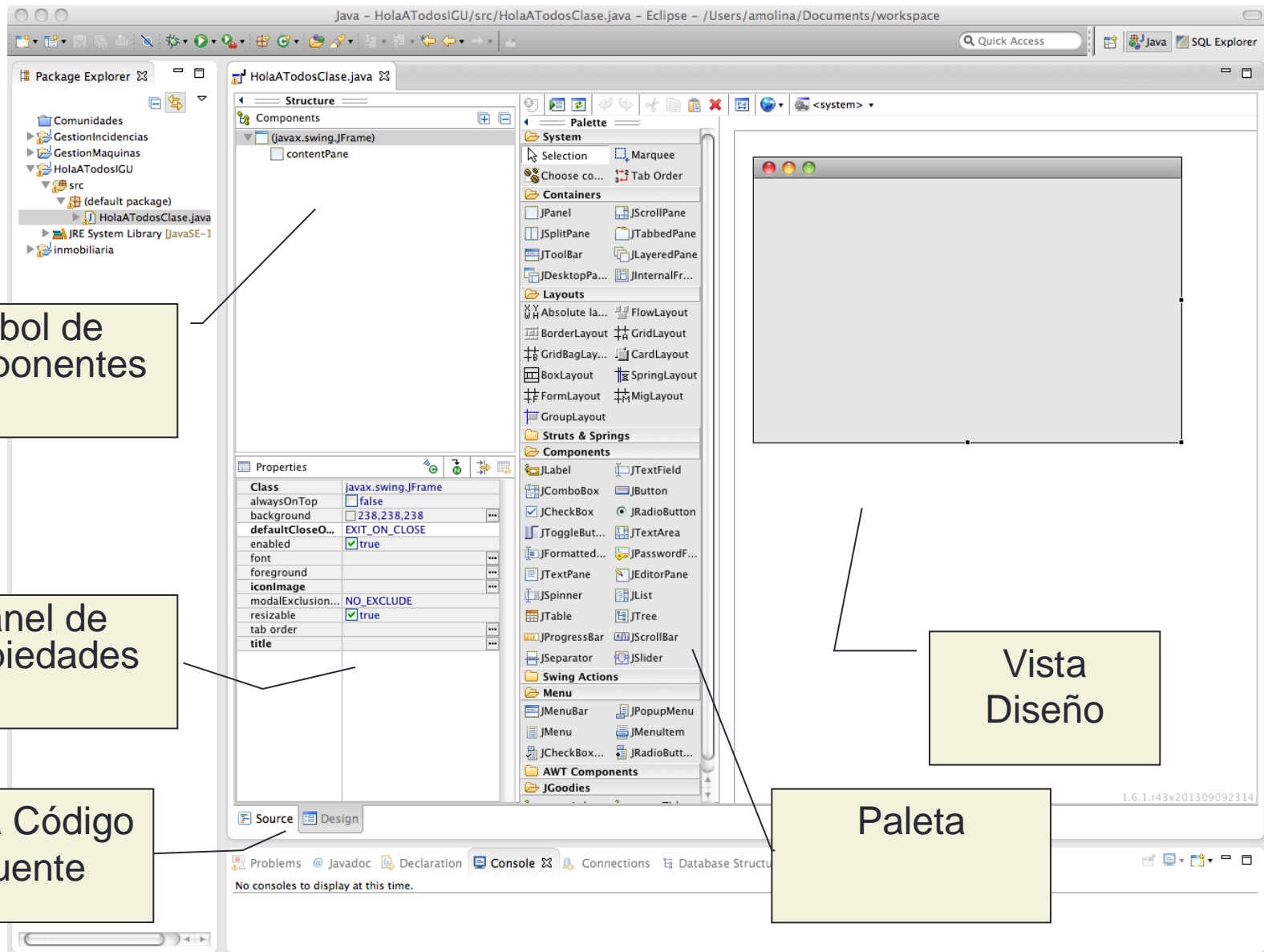
- **WindowBuilder** es un diseñador visual para el desarrollo de IGUs en Java.
- Da soporte a las librerías:
 - **Swing**
 - **SWT**
- WindowBuilder permite:
 - Diseñar la IGU arrastrando controles gráficos
 - Añadir manejadores de eventos
 - Generar el código Java de la IGU
 - Cambiar las propiedades de los controles gráficos
 - Soporte para internacionalización (i18n)
- En la página web encontrará los enlaces para instalación del plug-in y la guía de usuario

<http://www.eclipse.org/windowbuilder/>

Vistas de WindowBuilder

- **Design View:** muestra la componente visual que se está diseñando.
- **Source View:** muestra el código java de la componente visual que se está diseñando.
 - Al modificar la componente visual se actualiza el código correspondiente.
- **Structure View:**
 - **Árbol de componentes:** muestra la relación jerárquica entre los componentes de la IGU
 - **Panel de propiedades:** muestra las propiedades y eventos de los componentes y permite configurarlos.
- **Palette:** acceso a los componentes gráficos AWT, Swing y SWT.
- Además dispone de una barra de herramientas y menús contextuales para acceder a las órdenes más usadas.

Vistas de WindowBuilder



Configuración de WindowBuilder

- El menú de Preferencias de WindowBuilder permite indicar cómo se genera el código para las componentes gráficas:
 - Generación de variables
 - Generación de instrucciones
 - Generación de manejadores
 - Propiedades por defecto de los layouts
 - LookAndFeel
 - etc.
- Marque en Swing > Code Generation las siguientes opciones:
 - Variable generation: Field
 - Field modifier: private
 - Statement generation: Flat

Crear una aplicación gráfica

Este ejemplo muestra cómo crear una aplicación gráfica con WindowBuilder con los siguientes pasos:

- Paso 1: Crear un proyecto.
- Paso 2: Crear una aplicación (o una clase visual).
- Paso 3: Añadir contenedores (**containers**) a la interfaz.
- Paso 4: Añadir componentes visuales (**components**).
- Paso 5: Añadir menús.
- Paso 6: Modificar el código fuente.
- Paso 7: Asignar manejadores de eventos a un componente.

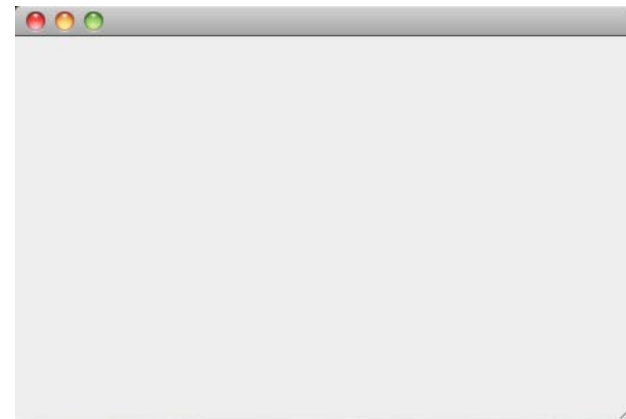
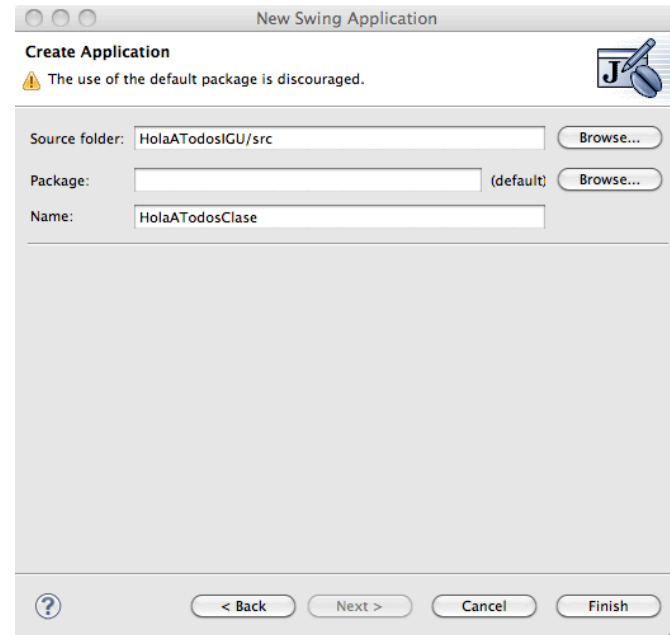
Paso 1: Crear un proyecto nuevo

- **File | New → Project ...**
- **Escoja Java Project**
- **Llámele, p.e., HolaATodosIGU**

Nota: para abrir con WindowBuilder una clase creada previamente, selecciónela en el explorador de paquetes y con el botón derecho del ratón **Open With → WindowBuilder Editor**

Paso 2: Crear una aplicación Swing

- **File | New | Other | WindowBuilder | Swing Designer → Application Window**
- **Package:** en blanco
- **Class Name:** **HolaATodosClase**, es la clase principal de la aplicación.
- Pulse **Finish**
- El código fuente generado contiene el método **main()** que inicializa el JFrame y lo hace visible
- Si ejecuta la aplicación se mostrará el frame vacío (no se ha añadido ningún componente gráfico)




Gestores de Diseño (Layouts)

- Los componentes_visuales se distribuyen en un contenedor según las reglas establecidas por el gestor de diseño (propiedad **Layout**). Algunos gestores para los contenedores Swing son:
 - **BorderLayout**: reparte los objetos en cinco lugares: North, South, East, West, Centre.
 - **FlowLayout**: reparte los objetos de izquierda a derecha horizontalmente, mientras quepan en la línea, antes de pasar a la siguiente.
 - **GridLayout**: reparte los objetos en una tabla de casillas de tamaño fijo.
 - **GridBagLayout**: igual que el anterior, pero las casillas pueden tener distintos tamaños.
 - **CardLayout**: varios paneles superpuestos que se seleccionan mediante pestañas.
 - **Absolute**: los objetos pueden colocarse libremente en cualquier posición

Paso 3: Añadir contenedores (barra de herramientas)

1. En el apartado **Containers** de la paleta de componentes, seleccione el componente **JToolBar**.
2. Para añadir el componente a la ventana puede hacerlo de dos maneras:
 - Sobre la vista de Diseño: colocando el componente sobre el JFrame. Se mostrará la división en zonas según el gestor de diseño del panel (en este caso BorderLayout). Sitúelo en la zona **North**.
 - Sobre el árbol de componentes: añadiendo el componente sobre el panel del frame (getContentPane()). Por defecto, el componente se sitúa en la zona **North**.
3. Desde el panel de propiedades puede editar aquellas que desee, p.e. el nombre de la variable asociada al componente (**Variable**), la posición en el panel (**Constraints**), colores de fondo (**background**), fuente (**foreground**), etc.
4. Cambie el tipo de borde (propiedad *border*). Por ejemplo, tipo **LineBorder**, **thickness** a 5 y a un color de la paleta. Para acceder a esta propiedad tendrá que expandir previamente la lista de propiedades.
5. Revise el código generado en la pestaña **Source**.



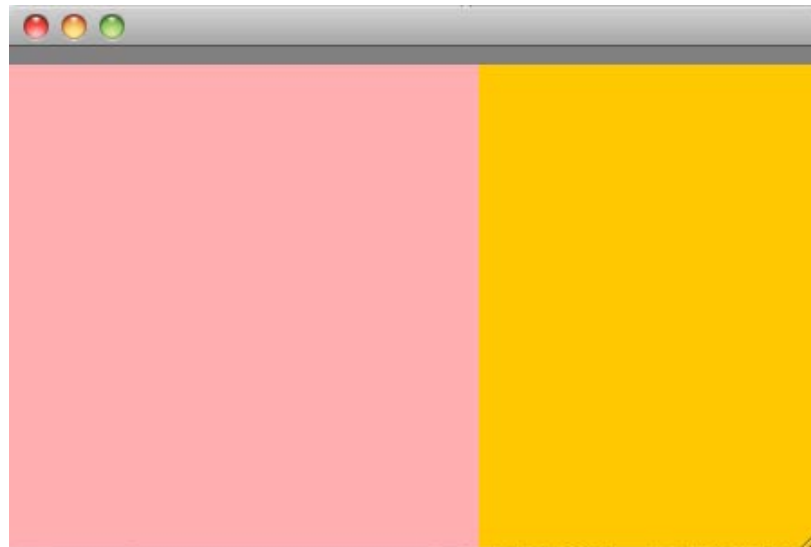
Properties	
Variable	toolBar
Constraints	North
Class	javax.swing.JToolBar
background	<input type="checkbox"/> 238,238,238 ...
enabled	<input checked="" type="checkbox"/> true
floatable	<input checked="" type="checkbox"/> true
font	Lucida Grande 13 ...
foreground	<input type="checkbox"/> 128,128,128 ...
orientation	HORIZONTAL
rollover	<input type="checkbox"/> false
toolTipText	...

Paso 3: Añadir contenedores (paneles)

1. En la pestaña **Containers** de la paleta de componentes escoja el componente **JPanel**.
2. Sitúelo en la zona **East** del panel asociado al marco de la aplicación (`getContentPane()`).
3. Llámelo **panelEste**.
4. Cambie el color del fondo del panel (propiedad **background**), a un color naranja de la paleta.
5. Cambie la dimensión del panel indicando en la propiedad *preferredSize* el ancho y alto del componente.
 - Su efecto sobre el componente dependerá del gestor de diseño del contenedor y de las restricciones.
 - En este caso podemos variar el ancho del panel, pero el alto viene determinado por las zonas Norte y Sur del gestor de diseño.
 - Pruebe con el valor: 190,0

Paso 3: Añadir contenedores (paneles)

7. Coloque otro panel:
 - En la zona **Center**,
 - Nombre: **panelCentro**,
 - background: **rosa**,
 - El contenedor situado en la zona Center se expande ocupando el espacio libre (no puede redimensionarse con preferredSize)
8. Revise el código generado y ejecute la aplicación.




Paso 3: Añadir componentes(botones)

1. En el apartado **Components** escoja el componente **JButton**
2. Sitúelo en la barra de herramientas (toolBar).
3. Llámelo **botonAbrir** (propiedad **Variable**).
4. Borre el texto que se muestra en el botón (propiedad **text**)
5. Añada el icono openFile.png (propiedad **icon**)
 - Previamente deberá añadir las imágenes de los iconos como un recurso.
 - Descargue el archivo imagenes.jar.
 - Cópielo en la carpeta src de su proyecto.
 - Añádalo al Build Path (Botón derecho sobre el archivo, opción **Build Path > Add to Build Path**).
 - Desde la propiedad **icon** puede añadir ahora la imagen seleccionando el modo **Classpath resource**.
6. Revise el código y ejecute la aplicación



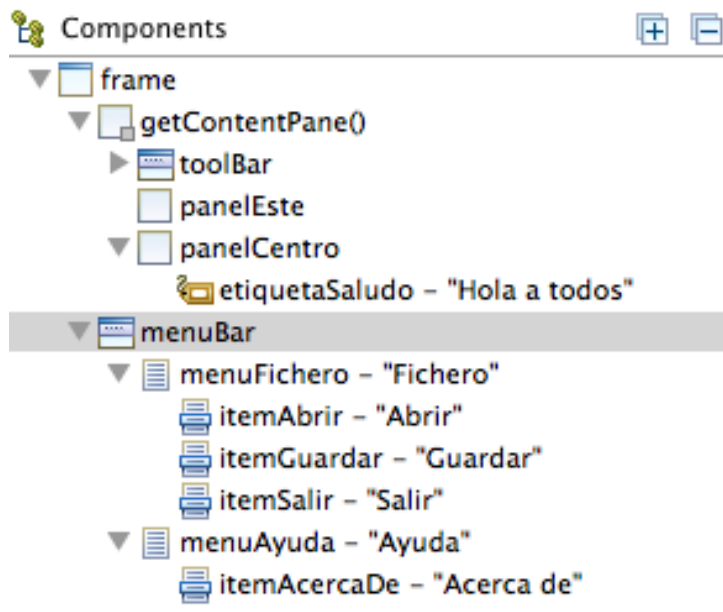
Paso 4: Añadir componentes (etiquetas)

1. Cambie la propiedad **Layout** del panel `panelCentro` a **Absolute**. Esto permitirá situar libremente cualquier componente visual en el panel.
2. En el apartado **Components** de la paleta, escoja el componente **JLabel**.
3. Sitúelo en el panel `panelCentro`.
4. Llámelo **etiquetaSaludo**.
5. Edite la propiedad **text** con el texto: "¡Hola a Todos!"
6. Cambie la fuente (propiedad **font**): Elija Serif en la lista de fuentes, marque Bold y escoja tamaño 24.
7. Si es necesario redimensione las componentes con el ratón: puede cambiar el tamaño del marco (frame) y de la etiqueta `etiquetaSaludo`.
8. Cambie el color de la fuente desde la propiedad **foreground**: seleccione el color rojo.
9. Revise el código y ejecute la aplicación



Paso 5: Añadir menús

1. Despliegue el apartado **Menu** de la paleta de componentes.
2. Primero añada la barra de menús, componente **JMenuBar**, sobre la barra de título del frame (parte superior de la ventana)
3. Para añadir un nuevo menú desplegable en la barra de menús, escoja el componente **JMenu** sitúelo en la barra de Menús.
4. Para añadir una opción de menú, escoja el componente **JMenuItem** y sitúelo sobre el menú desplegable creado anteriormente.
5. En el árbol de componentes puede observar cómo quedan organizados los menús.
6. Revise el código y ejecute la aplicación



Paso 6: Modificar el código fuente

Las modificaciones en el código se reflejan en la vista de Diseño inmediatamente.

1. Seleccione la etiqueta **etiquetaSaludo** en el Diseñador
2. Pulse sobre la pestaña **Source**. Aparecerá resaltada la línea de código donde se define el componente gráfico.
3. En la ventana de código sustituya
 `JLabel etiquetaSaludo = new JLabel("Hola a todos");`
por
 `JLabel etiquetaSaludo = new JLabel("Hola a todAs");`

Paso 7: Asignar manejadores de eventos

1. Asegúrese que el Layout del panel **panelEste** es **FlowLayout**. Este gestor distribuye los componentes por filas.
2. Coloque un botón en el panel **panelEste**.
3. Llámelo **botonCambiaColor**.
4. Edite la propiedad text: "Púlsame"
5. Para añadir un manejador de evento a un componente (p.e. JButton).
 - Botón derecho sobre el componente (en la vista de Diseño o en el árbol de componentes)
 - Opción: **Add event handler**
 - Seleccione el tipo de evento **action** y el único método propuesto **ActionPerformed**
6. Ir a la ventana del código y escriba el código para el manejador del suceso:

```
private void botonCambiaColorActionPerformed(ActionEvent evt) {  
    etiquetaSaludo.setForeground(new java.awt.Color(255,255,255));  
}
```

7. Ejecute la aplicación. Al pulsar sobre el botón cambiará el color del texto del saludo.