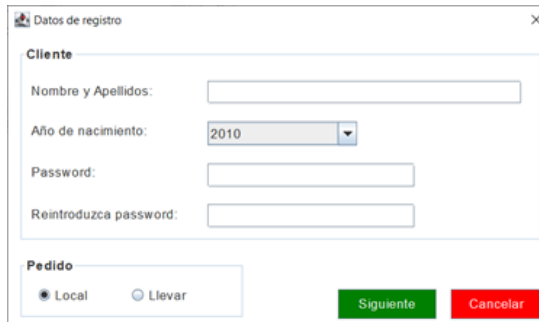


## Práctica 2

### Creación de una Aplicación con Interfaz Gráfica de Usuario utilizando WindowBuilder



#### 1. Introducción

Mediante esta práctica se persigue realizar una aplicación que presente un formulario para solicitar datos a un cliente. Se utilizarán los siguientes componentes:

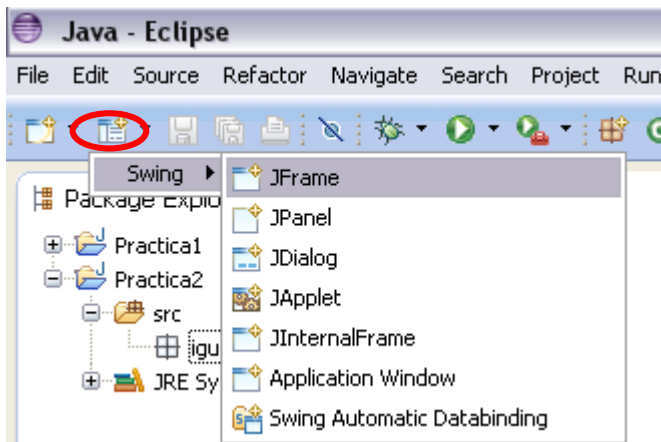
- Etiquetas (*JLabel*), Botones (*JButton*), Campos de texto (*JTextField*), Radio Botones (*JRadioButton*)
- Comboboxes (*JComboBox*)
- Campos para contraseña (*JPasswordField*)
- Paneles con título (*JPanel*)

#### 2. Creación de un proyecto en WindowBuilder con una clase visual

**Paso 1: Crear un proyecto java.** Ponle de nombre *Practica2*

**Paso 2: Añadir la clase visual**

Haz clic en el segundo icono de la barra de herramientas o selecciona *File-New-WindowBuilder-Swing Designer-JFrame*:



En el cuadro de diálogo, define la nueva clase visual:

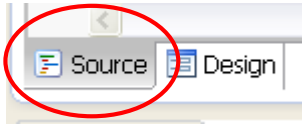


1. Introduce **uo.cpm.p2.ui** en el campo *package*
2. Introduce **VentanaRegistro** como nombre de la clase

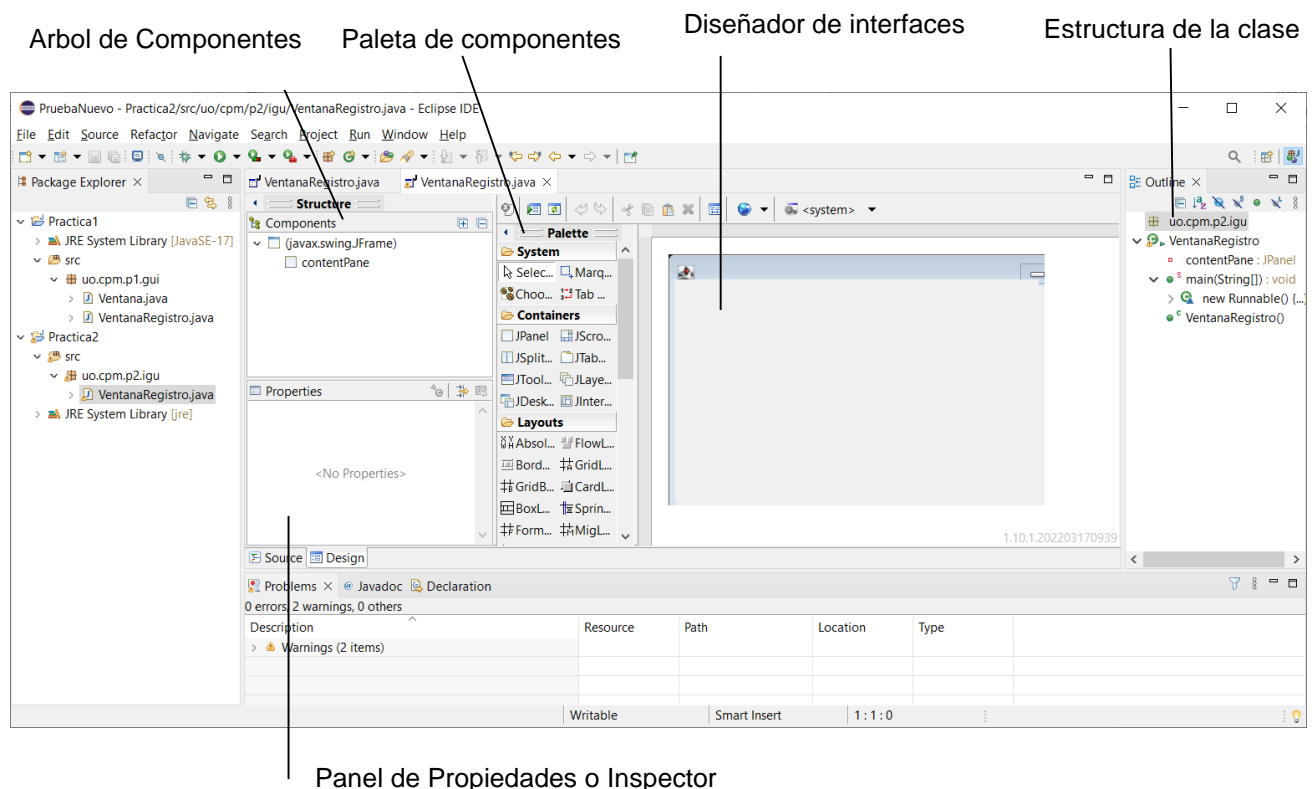
Una vez finalizada la definición de la clase, Eclipse nos presenta el código fuente generado. Este incorpora lo siguiente:

- El contenedor principal como atributo de la clase (*contentPane*)
- En el método main se crea y se hace visible una nueva ventana (*frame*)
- En el constructor de la ventana se establece la acción a realizar por defecto cuando el usuario pulse sobre el icono de cierre de la ventana, la dimensión de la misma, se crea la ventana, se dimensiona y, por último, se establece la relación entre la ventana y el panel.

El resto de los componentes los añadiremos mediante el editor visual de WindowBuilder. Para ello, seleccionaremos la pestaña Design situada en la parte inferior del panel de código.



El aspecto que presenta entonces Eclipse es el siguiente:

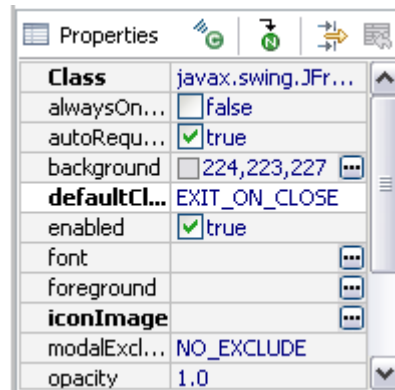


## 3. Modificación de los valores por defecto

Como ya hemos visto, WindowBuilder ya ha incorporado de manera automática una ventana o marco (*JFrame*) con un contenedor (*contentPane*) para ir incluyendo en él los componentes que formarán parte de la interfaz. Vamos a modificar alguno de los valores por defecto tanto de la ventana como del panel de contenidos o contenedor.

### 3.1. VentanaRegistro

Selecciona el marco o ventana en el panel de diseño haciendo clic sobre la barra de título de la misma. Observa como en la pestaña propiedades se muestra la relación de atributos relacionados con el objeto de tipo *JFrame*. A través de este panel pueden cambiarse los valores de los atributos; además, algunos pueden modificarse directamente sobre el editor visual como veremos en el siguiente ejemplo:



Modificamos:

1. **title:** introduce la cadena *Datos de registro*.
2. el tamaño de la ventana a (600,300) “estirando” la ventana directamente en el diseñador mediante los controles situados en los bordes de la misma. Previamente “haz sitio” modificando la disposición de los paneles de WindowBuilder para poder ver la ventana en su nueva dimensión.

## 3.2 Panel de contenidos

Como ya comentamos previamente, WindowBuilder ya ha incorporado en la ventana un contenedor principal o panel de contenidos (*contentPane*) para albergar los componentes que formarán parte de la interfaz. Es recomendable modificar el nombre del componente. Para ello, selecciona el panel de contenidos en el panel de diseño haciendo clic en el centro de la ventana que estamos diseñando. En la pestaña propiedades se muestra la relación de atributos relacionados con el objeto de tipo *JPanel*.

Modificamos:

1. **Variable:** sustituye el nombre que aparece por defecto por un nombre más significativo como *pnPrincipal*
2. **Layout:** selecciona *Absolute layout* en la lista desplegable.
3. **background** o color de fondo: haz clic sobre los puntos suspensivos para abrir el cuadro de diálogo que permite modificar el atributo color. Elige el color *white*.

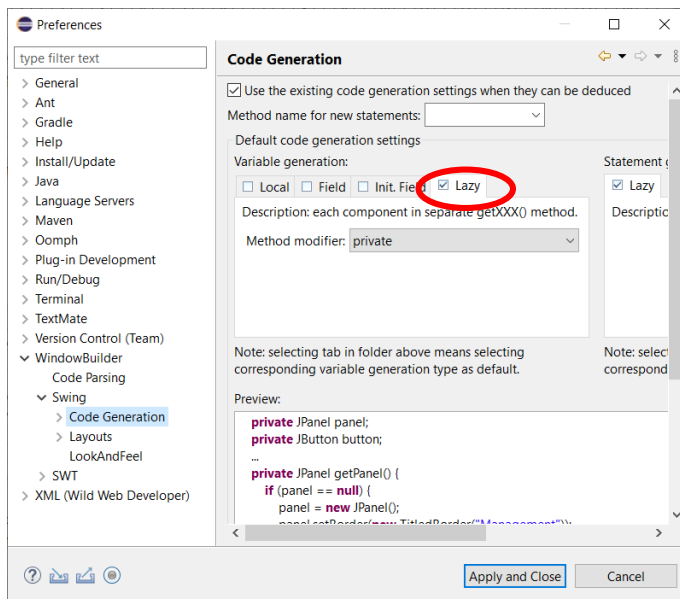
Comprueba como todos los cambios realizados se recogen en el código fuente de la aplicación.

## 4. Personalización de la interfaz de usuario

### 4.1. Incorporación de nuevos componentes a la aplicación

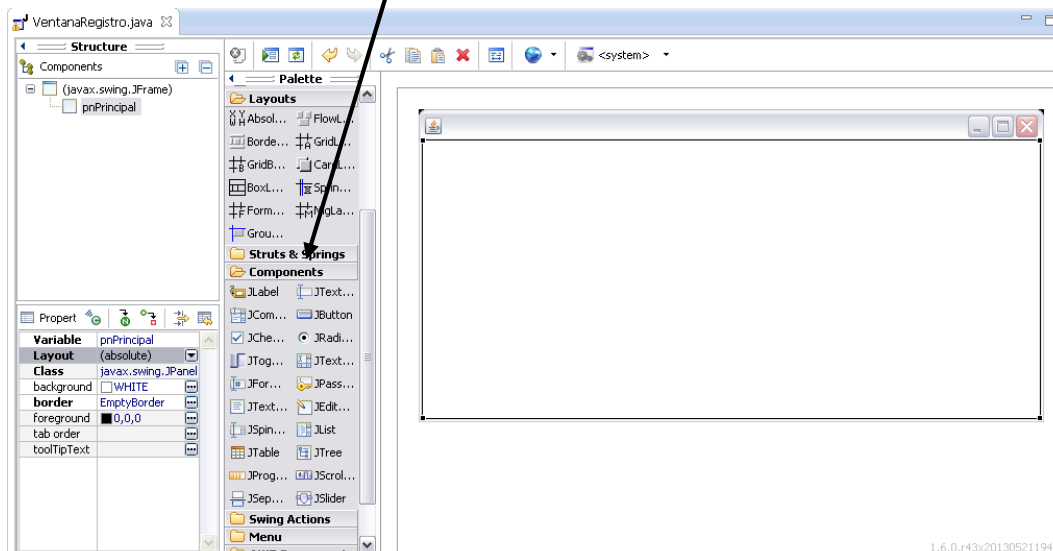
#### Paso 1

Modifica el modo en el que WindowBuilder generará el código de forma automática. Para ello, accede al menú y elige Window – Preferences - WindowBuilder (clic en >) - Swing (clic en >) – Code Generation (clic encima) – Lazy (clic en pestaña)



## Paso 2

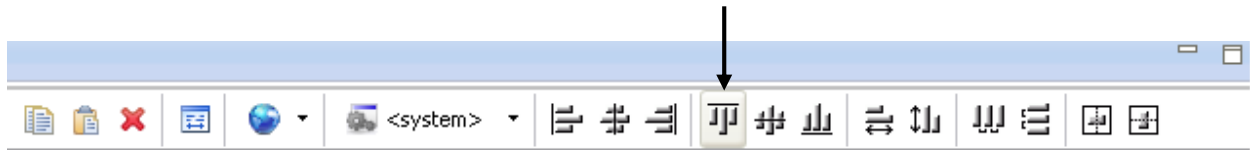
Selecciona la carpeta *Components* en la paleta de componentes. Aparecerá la relación de componentes visuales que podemos incluir en la ventana que estamos diseñando.



## Paso 3

Añade los componentes necesarios en la ventana y modifica las propiedades oportunas. Recuerda proporcionar a cada uno de ellos un nombre significativo.

Para alinear los componentes que se desee en línea vertical/horizontal se seleccionan todos a la vez (manteniendo pulsada a tecla *Ctrl*) y se elige la opción adecuada mediante los botones que aparecen en la parte superior del diseñador.



Para modificar una o varias propiedades comunes a varios componentes puede también realizarse selección múltiple y modificar la/s propiedad/es en el panel de propiedades.

En el caso del ComboBox, introduce los posibles valores de los mismos modificando el atributo *model* y añadiendo un valor en cada línea.

## 5. Añadir funcionalidad a un botón

Se trata ahora de asociar al botón *Cancelar* el código necesario para que al hacer clic sobre él con el ratón finalice la aplicación.

### Paso 1

Selecciona el botón (bien en el panel de diseño o bien en el árbol de componentes) y haz clic con el botón derecho del ratón para mostrar el menú contextual. Selecciona *Add event handler*.

### Paso 2

Vamos a definir lo que ocurrirá al pulsar sobre *btCancelar*. Selecciona el evento *ActionPerformed* (el primero de la lista). El IDE genera el código que define la función respuesta a la acción del usuario sobre el botón. En el código fuente se ha añadido el siguiente código correspondiente al evento *ActionPerformed*:

```
btnCancelar.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
    }  
});
```

### Paso 3

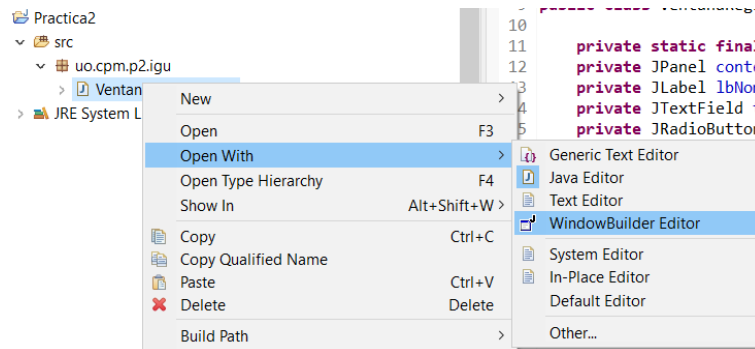
Introduce el siguiente código en el método *actionPerformed*, correspondiente a la finalización de la aplicación:

```
System.exit(0);
```

Ejecutamos de nuevo la aplicación y comprobamos que al hacer clic sobre el botón *Cancelar*, la aplicación finaliza.

## **Tareas práctica 2. IMPORTANTE FINALIZARLAS PARA LA SIGUIENTE SESION DE PRÁCTICAS**

1. **IMPORTANTE:** Estudiar los fundamentos teóricos asociados a esta práctica porque formarán parte de los contenidos a evaluar tanto en el examen teórico como en el práctico de la asignatura.
2. Una vez importado el proyecto, para **acceder al diseñador visual de una clase** (es decir, para que salga la pestaña *Design*), haz clic con el botón derecho sobre la clase y selecciona **Open With – WindowBuilder editor**



3. **Completa la interfaz** con todos los componentes indicados al inicio de la práctica. En la siguiente práctica utilizaremos esta ventana con todos sus componentes ya finalizada. Para incorporar componentes dentro de un panel con título selecciona un *JPanel* dentro de la pestaña *Containers* y colocalo en la ventana. Modifica a continuación la propiedad *border* del panel seleccionando *TitledBorder* y escribe ahí el título del panel. Antes de colocar los componentes que van dentro del panel, selecciona *AbsoluteLayout* en la pestaña *Layouts* del panel.
4. Incorpora como evento del botón *Siguiente* la llamada a uno o varios métodos (que tendrás que crear tú añadiéndolo al código fuente) para comprobar que:
  - a. El campo de texto del nombre no queda vacío
  - b. Los dos campos de contraseña coinciden.

Mostrar uno o varios mensajes al usuario en el caso de que no se cumpla alguno de los requisitos anteriores.

Nota 1: el método `getText()` de un componente *JTextField* devuelve la cadena de texto que contiene dicho componente. Si el componente es un *JPasswordField* el método a utilizar es `getPassword()` que devuelve un array de caracteres. Mediante la sentencia `String.valueOf(pwPass.getPassword())` siendo *pwPass* el nombre de un componente *PasswordField*, el array de caracteres se pasa a *String*.

Nota 2: para mostrar un mensaje al usuario, puedes utilizar la siguiente sentencia:

```
JOptionPane.showMessageDialog(null, "Texto a mostrar");
```

4. Considera TODAS las **características de usabilidad** de los componentes que hayas introducido y que están descritos en los fundamentos teóricos (ejemplo: mnemónico para un botón). Indica como comentarios en el método `get` del componente de que se trate aquellas que no sepas cómo implementar.