

# DESARROLLO DE APLICACIONES MULTIPLATAFORMA

## Desarrollo de Interfaces

Equivalencia en créditos ECTS (BOE): 9

### Tema 1: Confección de Interfaces de Usuario

*RA: Generar interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado*

# Índice

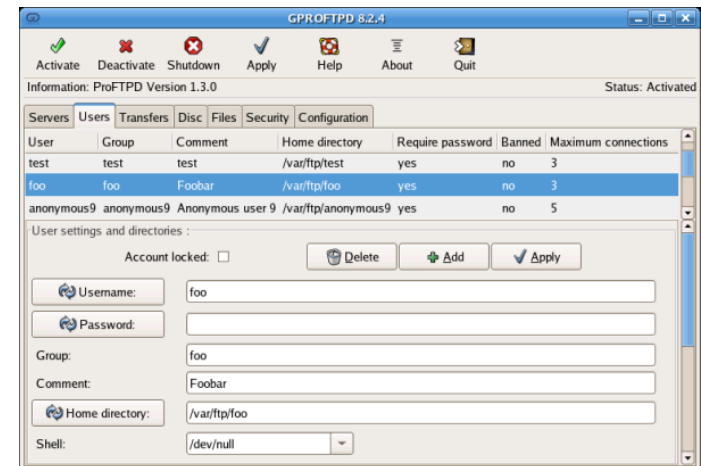
1. Interfaz de usuario
2. Herramientas de edición de interfaces
3. Librerías de componentes
4. Componentes y eventos en Gambas
5. Componentes y eventos en WindowBuilder
6. Prácticas
  - Práctica 1
  - Práctica 2

# ¿Qué es una interfaz de usuario?

- Es el medio con que el usuario se comunica con una máquina.
- Existen principalmente 3 tipos: CLI, GUI y NUI.

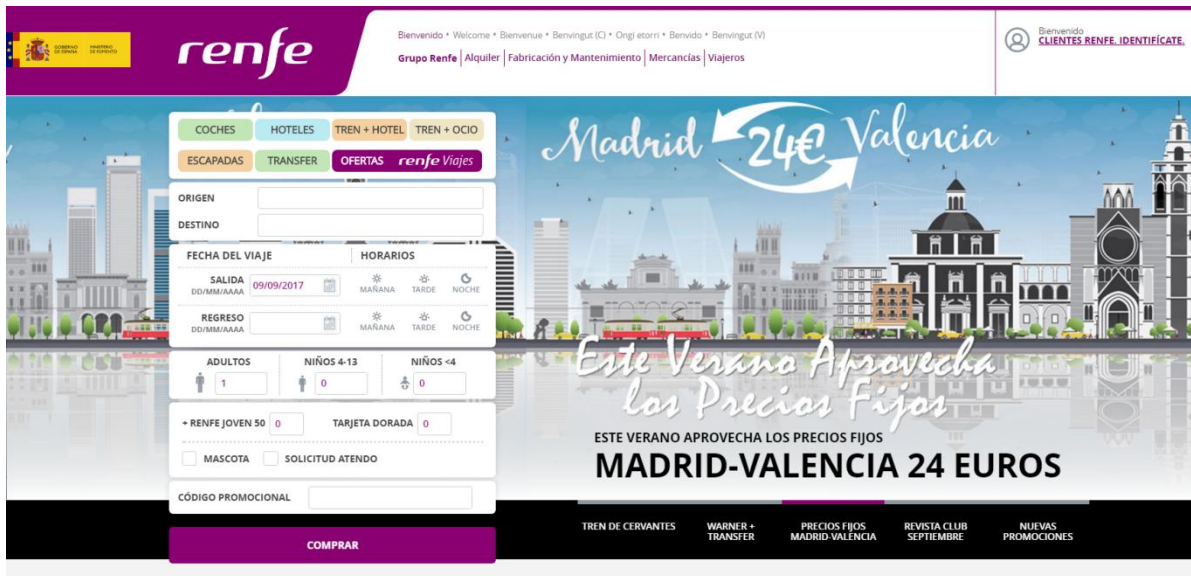


```
proxima:/usr/bin - Befehlsfenster - Konsole
-rwxr-xr-x 1 root root 44K 2005-12-06 20:53 yuvcorrect_tune
-rwxr-xr-x 1 root root 22K 2005-12-06 20:53 yuvdeinterlace
-rwxr-xr-x 1 root root 19K 2005-12-06 20:53 yuvdenoise
-rwxr-xr-x 1 root root 9.3K 2005-12-06 20:53 yuvfps
-rwxr-xr-x 1 root root 15K 2005-12-06 20:53 yuvinactive
-rwxr-xr-x 1 root root 36K 2005-12-06 20:53 yuvkineco
-rwxr-xr-x 1 root root 17K 2005-12-06 20:53 yuvmedianfilter
-rwxr-xr-x 1 root root 13K 2005-12-06 20:53 yuvplay
-rwxr-xr-x 1 root root 60K 2005-12-06 20:53 yuvscaler
-rwxr-xr-x 1 root root 8.2K 2006-02-06 20:28 yuvsplittoppm
-rwxr-xr-x 1 root root 6.5K 2006-02-06 20:28 yuvtoppm
-rwxr-xr-x 1 root root 18K 2005-12-06 20:53 yuvcynoise
-rwxr-xr-x 1 root root 9.5K 2005-12-06 20:53 yuyvtoy4m
-rwxr-xr-x 1 root root 12K 2006-05-21 19:25 zdump
-rwxr-xr-x 1 root root 6.9K 2006-02-06 20:28 zeissstopnm
-rwxr-xr-x 1 root root 73K 2005-05-26 11:47 zip
-rwxr-xr-x 1 root root 30K 2005-05-26 11:47 zipcloak
-rwxr-xr-x 1 root root 1.2K 2006-03-23 14:28 zipgrep
-rwxr-xr-x 2 root root 118K 2006-03-23 14:28 zipinfo
-rwxr-xr-x 1 root root 26K 2005-05-26 11:47 zipnote
-rwxr-xr-x 1 root root 28K 2005-05-26 11:47 zipsplit
-rwxr-xr-x 1 root root 5.1K 2005-06-28 18:18 zips_xnc
-rwxr-xr-x 1 root root 62K 2006-01-02 14:20 zjsdecode
-rwxr-xr-x 1 root root 687 2006-05-28 09:43 zoneatab2pot.py
-rwxr-xr-x 1 root root 3.1M 2006-02-13 00:32 zsnas
-rwxr-xr-x 1 root root 40K 2005-09-26 17:13 zsoelim
[22:53:57] proxima:/usr/bin$
```

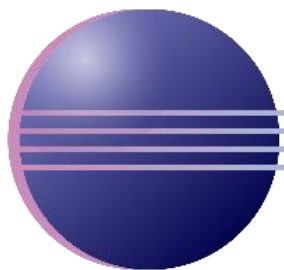


# GUI: Interfaces Gráficas de Usuario

- GUI de escritorio
- GUI Web
  - Diseño web responsive o adaptativo



# ¿Software libre o propietario?



# Ejercicio

| Herramienta    | Libre o Propietario | Sistema Operativo |
|----------------|---------------------|-------------------|
| Eclipse        |                     |                   |
| Visual Studio  |                     |                   |
| QT Creator     |                     |                   |
| Xcode          |                     |                   |
| Flash Builder  |                     |                   |
| Xamarin Studio |                     |                   |
| Android Studio |                     |                   |
| Gambas         |                     |                   |

# Lenguajes de programación



Objective-C

# Librerías para la creación de GUI



Swing y AWT



QT



Windows Forms



SDK



QT y GTK+



Xamarin Forms



Objective-C

Interface Builder



Librerías Flex

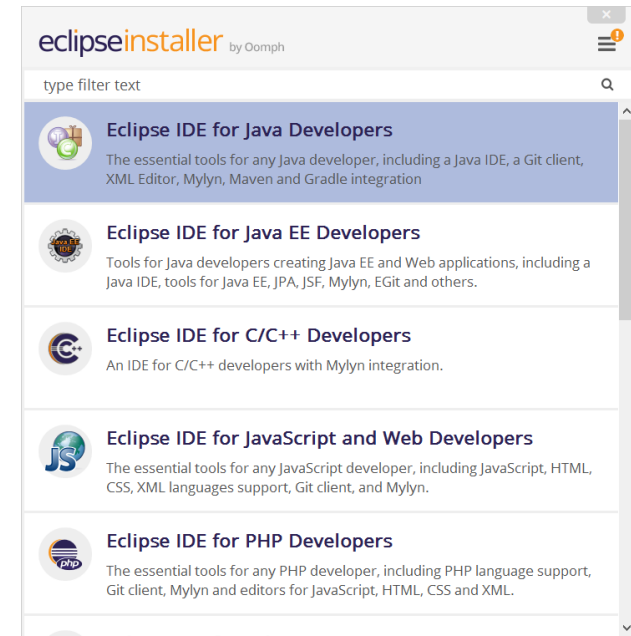
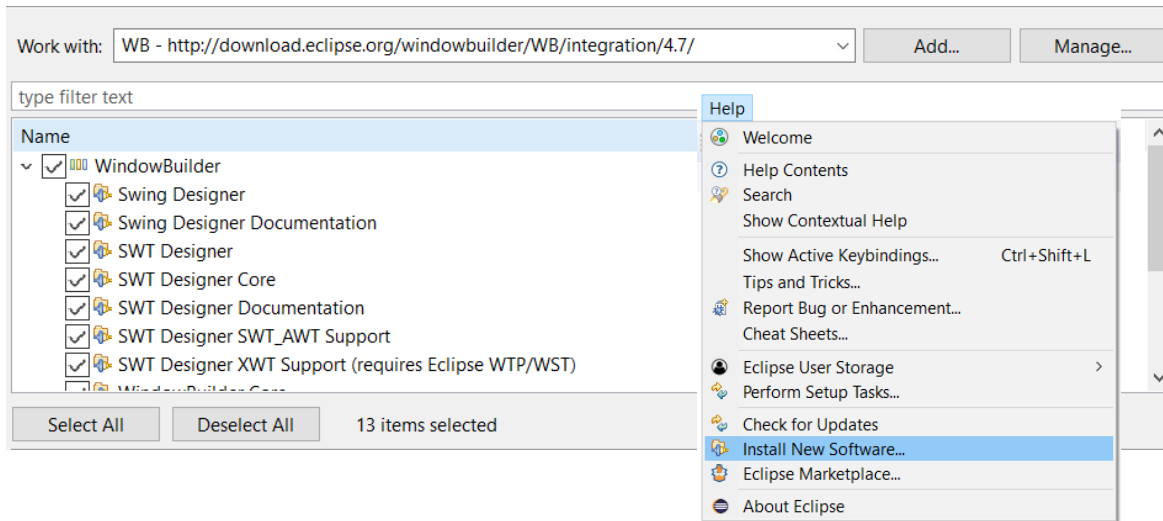


# Ejercicio

- Instalar Gambas en Ubuntu desde la CLI
  - Gambas incluye por defecto las librerías de QT
- Comprobar si está Java instalado. Si no, instalar
- Instalar Eclipse en Windows
  - Instalar el plugin WindowBuilder

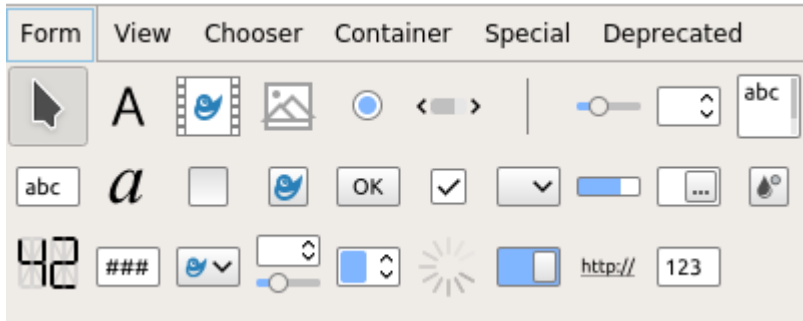


ubuntu

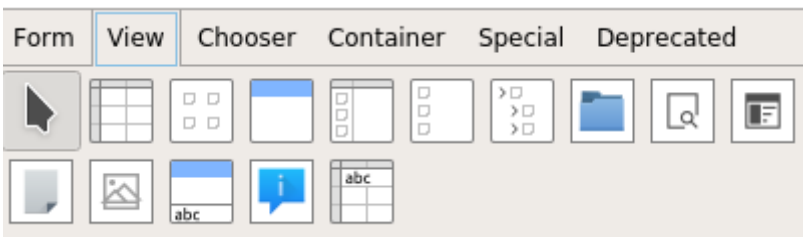


# Componentes en Gambas

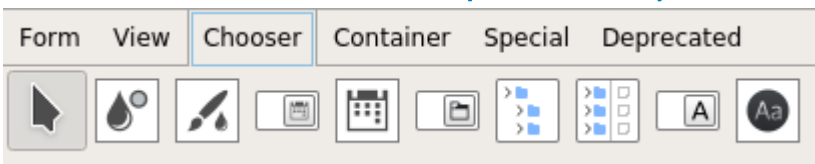
- Elementos Formulario (Form)



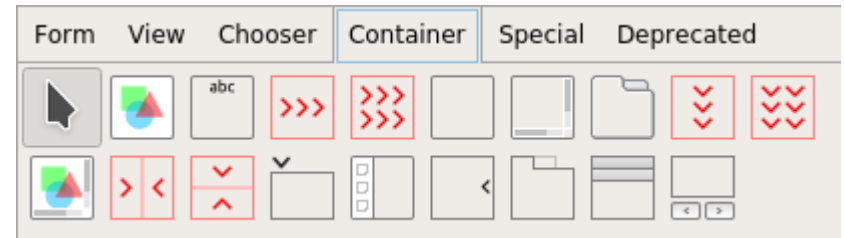
- Elementos Vista (View)



- Elementos Elección (Chooser)

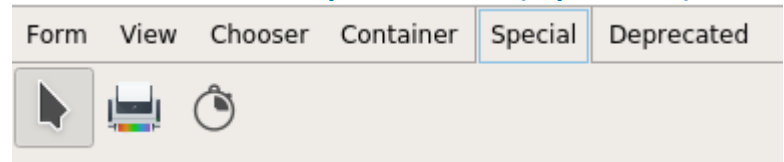


- Elementos Contenedores (Containers)

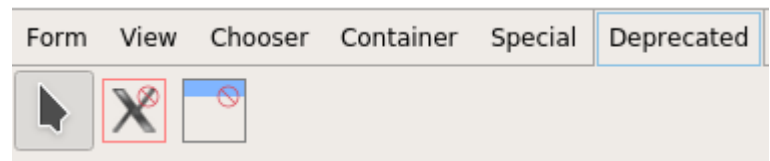


- Los contenedores son componentes que engloban a otros componentes

- Elementos Especiales (Special)



- Elementos Obsoletos (Deprecated)

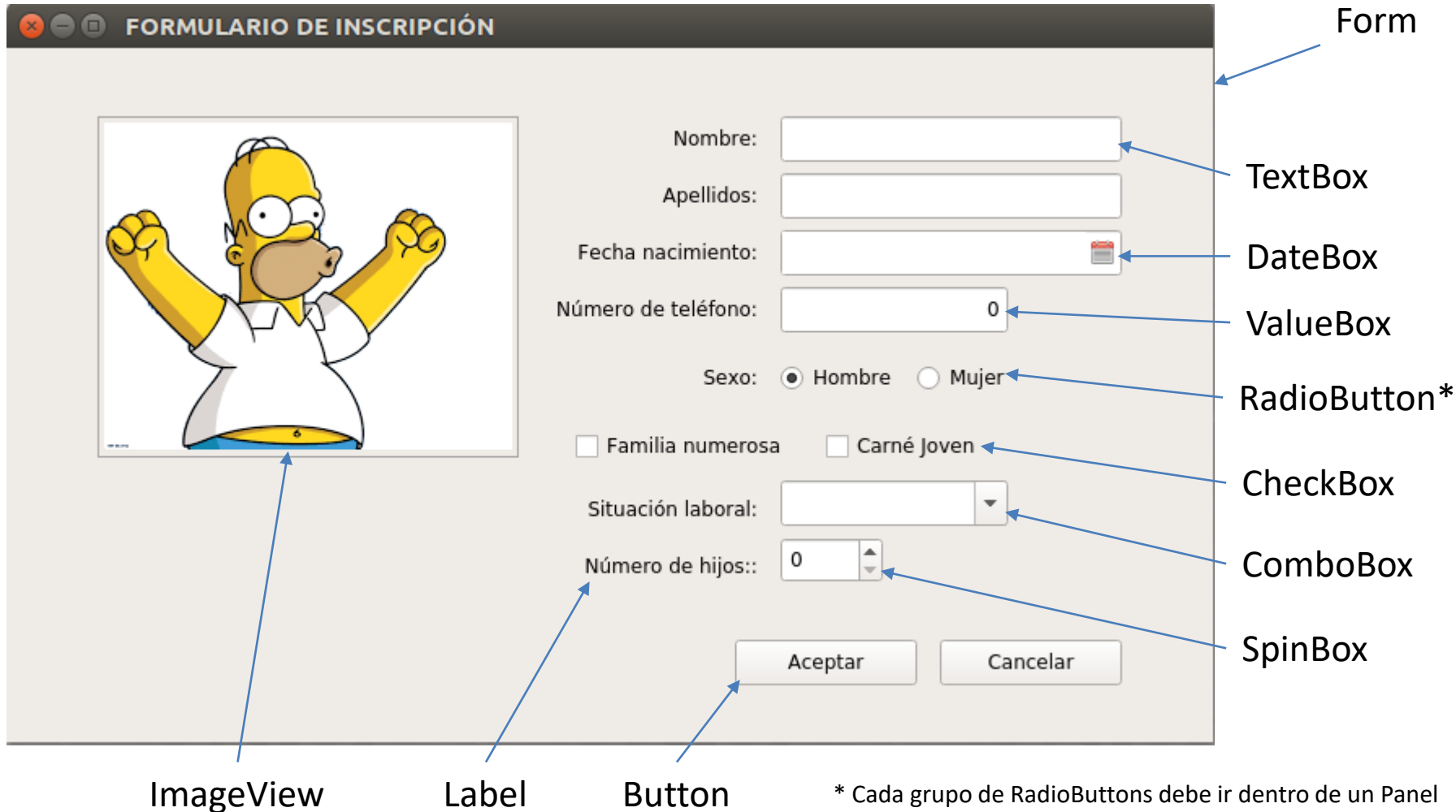


# Ejercicio

- Diseñar una GUI en Gambas haciendo uso de:
  - Un componente Container
  - Un componente Chooser
  - Un componente View, Special o Deprecated
  - Seis componentes Form diferentes
- Estudiar las diferentes propiedades de cada componente y probar a modificar valores

| Properties          | Hierarchy |
|---------------------|-----------|
| <b>Label1</b> Label |           |
| Class               | Label     |
| Name                | Label1    |
| Group               |           |
| Alignment           | Normal    |
| AutoResize          | False     |
| Background          |           |
| Border              | None      |
| Drop                | False     |
| Enabled             | True      |
| Expand              | False     |
| Font                |           |
| Foreground          |           |
| <b>Height</b>       | 28        |
| Ignore              | False     |
| Mouse               | Default   |
| NoTabFocus          | False     |
| Padding             | 0         |
| PopupMenu           |           |
| Public              | False     |
| Tag                 |           |
| Text                |           |
| ToolTip             |           |
| Tracking            | False     |
| Translate           | True      |
| Transparent         | False     |
| Visible             | True      |
| <b>Width</b>        | 168       |
| <b>X</b>            | 70        |
| <b>Y</b>            | 196       |

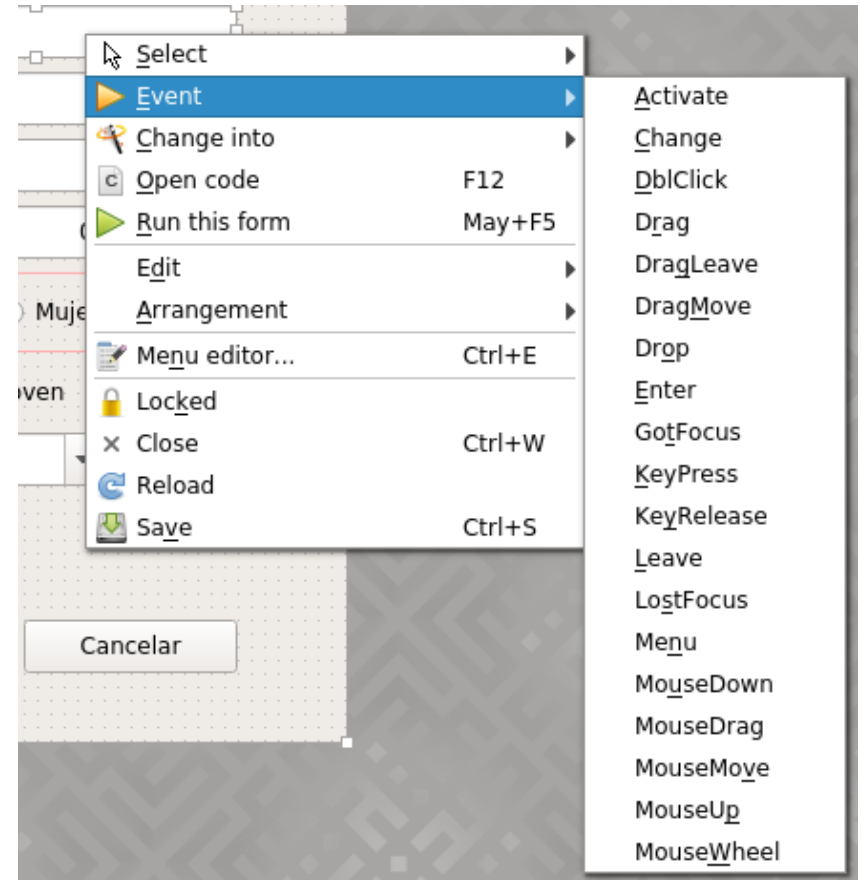
# Ejemplo de GUI en Gambas



# Eventos en Gambas

- Acción que el usuario realiza sobre un componente.

| Componentes | Eventos utilizados |
|-------------|--------------------|
| Form        | Show, Open, Close  |
| TextBox     | KeyPress, Change   |
| DateBox     | Change             |
| RadioButton | Click              |
| CheckBox    | Click              |
| ComboBox    | Click, Change      |
| SpinBox     | Change             |
| Button      | Click              |



# Cuadros de diálogo

```
Dim respuesta As Integer
'-----
'Ejemplo de uso de message.Delete (borrado)
'-----

Message.Title = "Borrar: Titulo del mensaje"
respuesta = Message.Delete("Texto", "boton 1", "boton 2", "boton 3")
Print "Mensaje Delete: respuesta elegida ", respuesta
'-----

'Ejemplo de uso de message.Warning (aviso)
'-----

Message.Title = "Warning: Titulo del mensaje"
respuesta = Message.Warning("Texto", "boton 1", "boton 2", "boton 3")
Print "Mensaje Warning: respuesta elegida ", respuesta
'-----

'Ejemplo de uso de message.Error (Error)
'-----

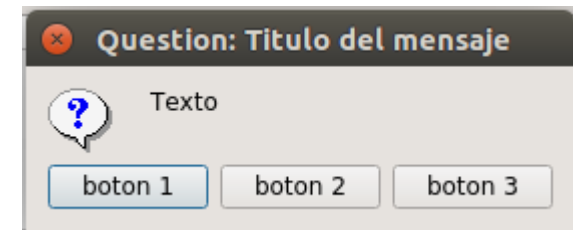
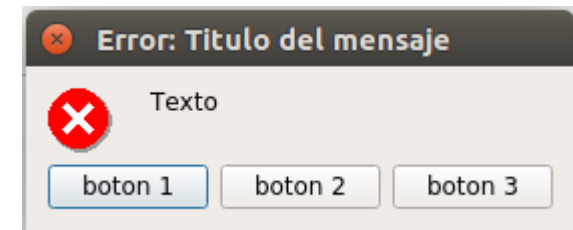
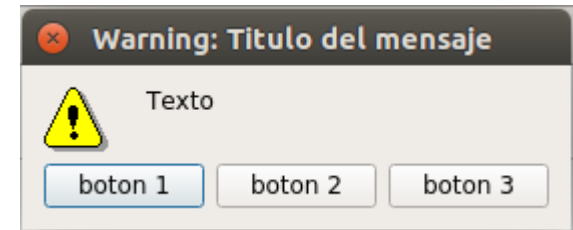
Message.Title = "Error: Titulo del mensaje"
respuesta = Message.Error("Texto", "boton 1", "boton 2", "boton 3")
Print "Mensaje Error: respuesta elegida ", respuesta
'-----

'Ejemplo de uso de message.info (informacion)
'-----

Message.Title = "Info: Titulo del mensaje"
respuesta = Message.Info("Texto", "boton 1")
Print "Mensaje Info: respuesta elegida ", respuesta
'-----

'Ejemplo de uso de message.Question (pregunta)
'-----

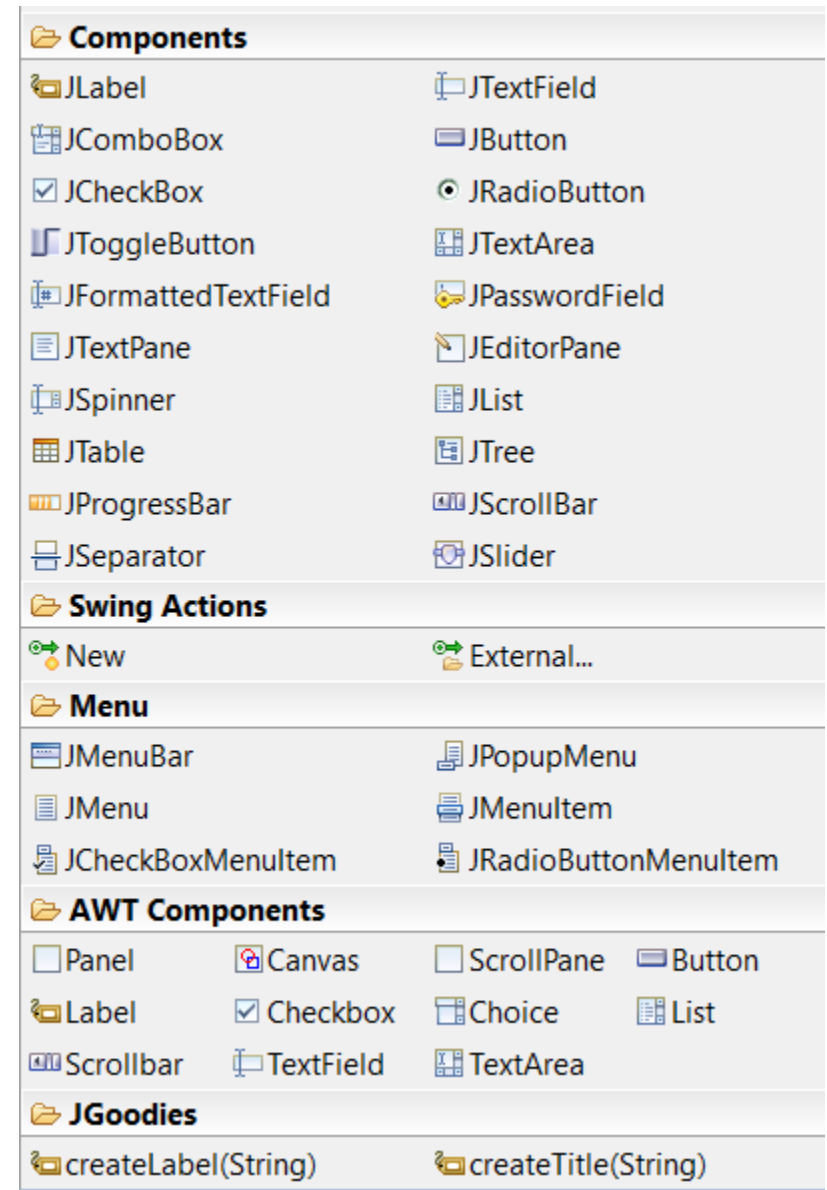
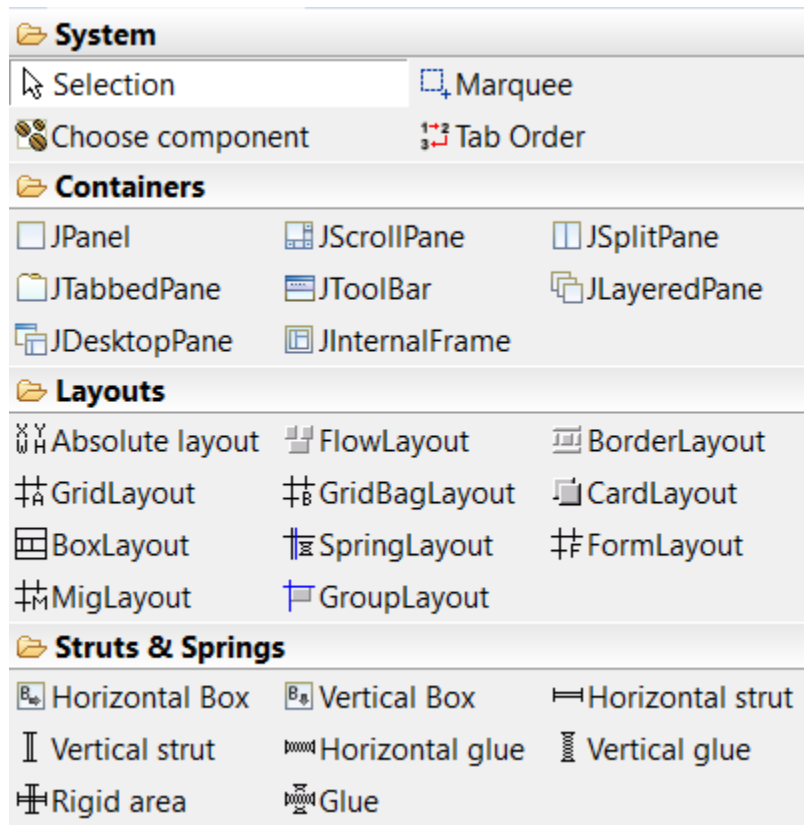
Message.Title = "Question: Titulo del mensaje"
respuesta = Message.Question("Texto", "boton 1", "boton 2", "boton 3")
Print "Mensaje Question: respuesta elegida ", respuesta
```



# Ejercicio

- Tras el diseño de la GUI realizado en la diapositiva 11, añadir a cada componente un evento.
- Diseñar la GUI de la diapositiva 12 y añadir un evento a cada componente.
- Probar a incorporar en uno de los eventos un cuadro de diálogo (mensaje, opciones, etc.).

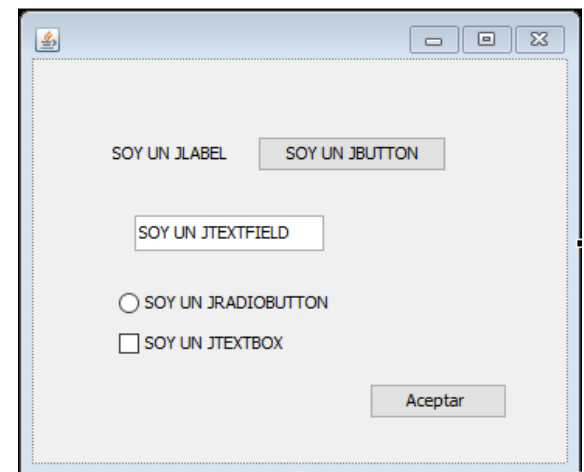
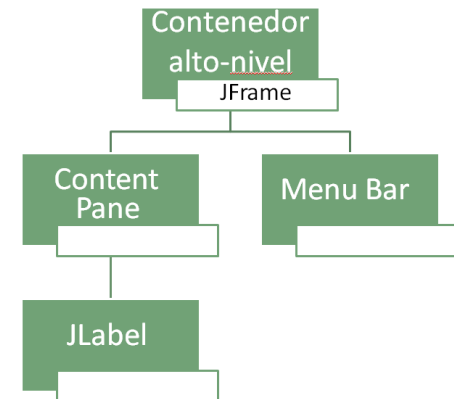
# Componentes en WindowBuilder





# Componentes: Jerarquía de clases

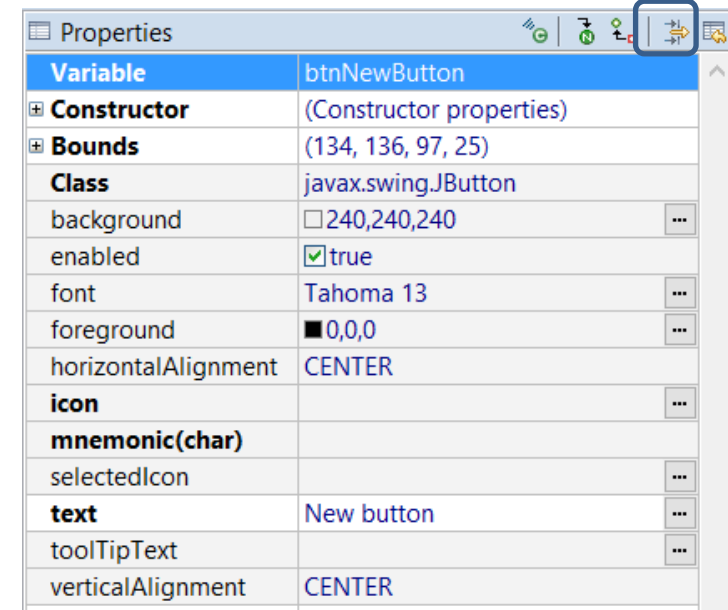
- Contenedores:
  - **JFrame**, JApplet, JWindow, JDialog
- Componentes intermedios:
  - **JPanel**, JScrollPane
  - **getContentPane(): AbsoluteLayout**
- Componentes:
  - **JLabel**, JButton, JRadioButton, ...
- Clases de soporte:
  - Graphics, Color, Font, ...



# Ejercicio

- Diseñar una GUI en WindowBuilder (Eclipse) haciendo uso de:
  - Un componente Container
  - Un componente Layout
  - Un componente Menu
  - Seis componentes (Componentes) diferentes
- Estudiar las diferentes propiedades\* de cada componente y probar a modificar valores

\* Descubrir las Propiedades ocultas

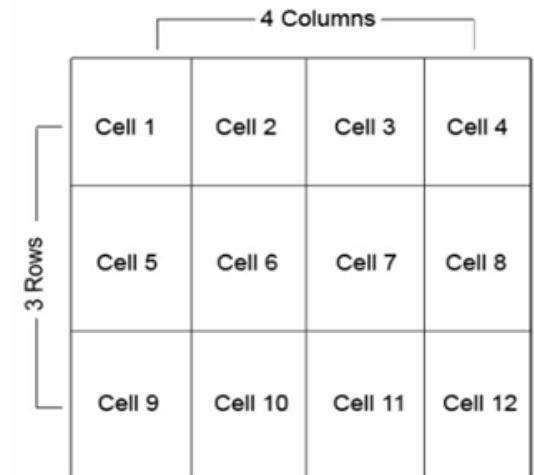
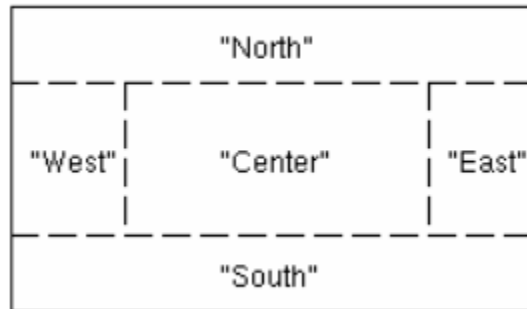


The screenshot shows the Eclipse WindowBuilder Properties view for a component named 'btnNewButton'. The 'icon' property is highlighted with a blue box, and an arrow points from the text '\* Descubrir las Propiedades ocultas' to this box. The table below lists the properties and their values.

| Variable              | btnNewButton                             |
|-----------------------|--|
| <b>Constructor</b>    | (Constructor properties)                 |
| <b>Bounds</b>         | (134, 136, 97, 25)                       |
| <b>Class</b>          | javax.swing.JButton                      |
| background            | □ 240,240,240                            |
| enabled               | <input checked="" type="checkbox"/> true |
| font                  | Tahoma 13                                |
| foreground            | ■ 0,0,0                                  |
| horizontalAlignment   | CENTER                                   |
| <b>icon</b>           |  |
| <b>mnemonic(char)</b> |  |
| selectedIcon          |  |
| <b>text</b>           | New button                               |
| toolTipText           |  |
| verticalAlignment     | CENTER                                   |

# Administradores de disposición

- **FlowLayout:** un componente tras otro de izquierda a derecha.
- **BorderLayout:** 5 regiones en el contenedor (North, South, ...).
- **GridLayout:** contenedor en filas y columnas.

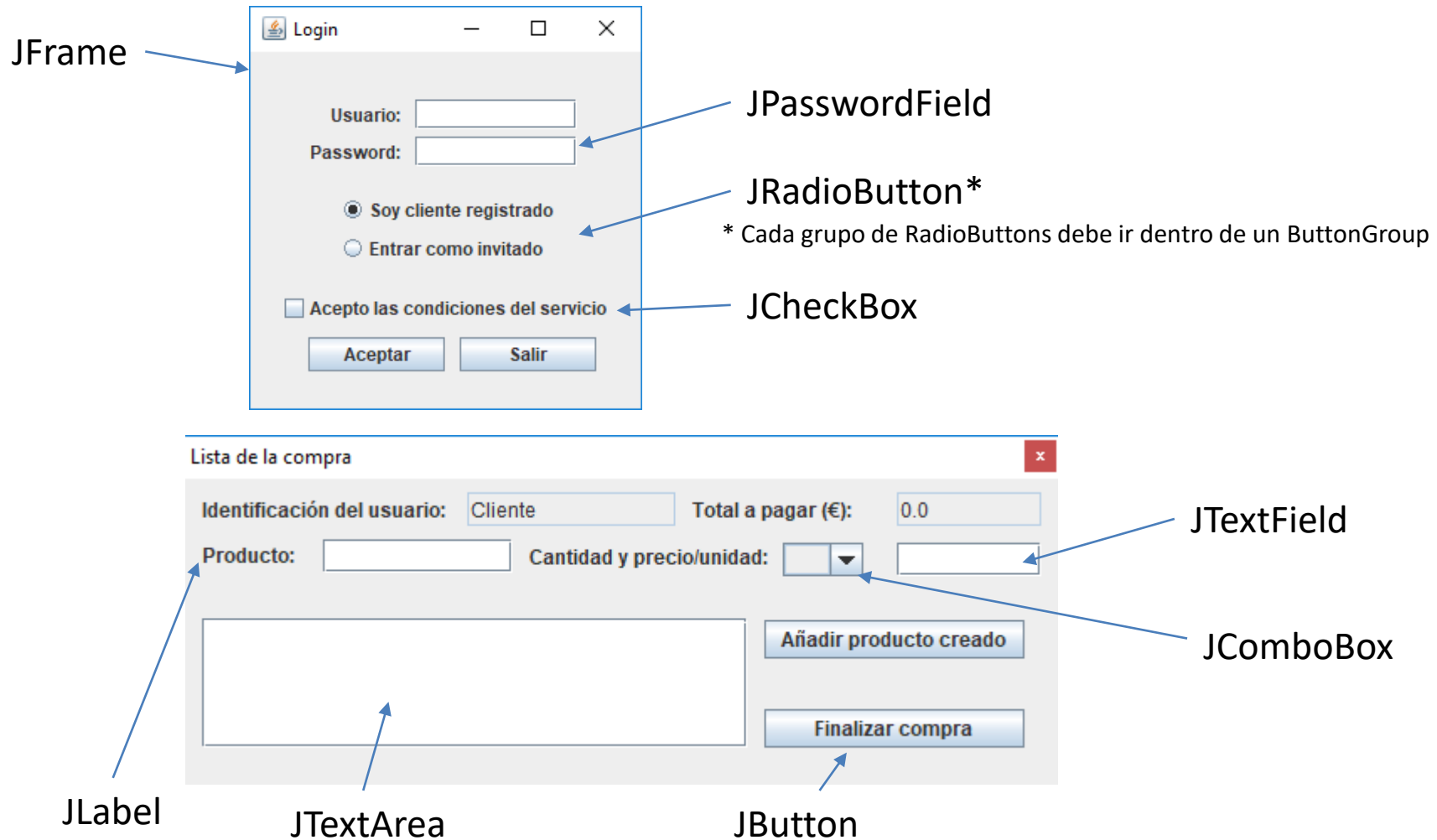


- Para la distribución Absolute Layout se establece la propiedad a null: `frame.getContentPane().setLayout(null);`

# Ejercicio

- Realizar 3 GUI de una calculadora con 3 TextBox (primer nº, segundo nº y resultado), 3 Label (para indicar estos números) y 4 botones (suma, resta, multiplicación y división) mediante:
  - FlowLayout
  - BorderLayout
  - GridLayout
- Los eventos han de estar asociados a los botones
- Si un TextBox está vacío o incluye algo distinto a un número entero positivo, sacar por pantalla este error

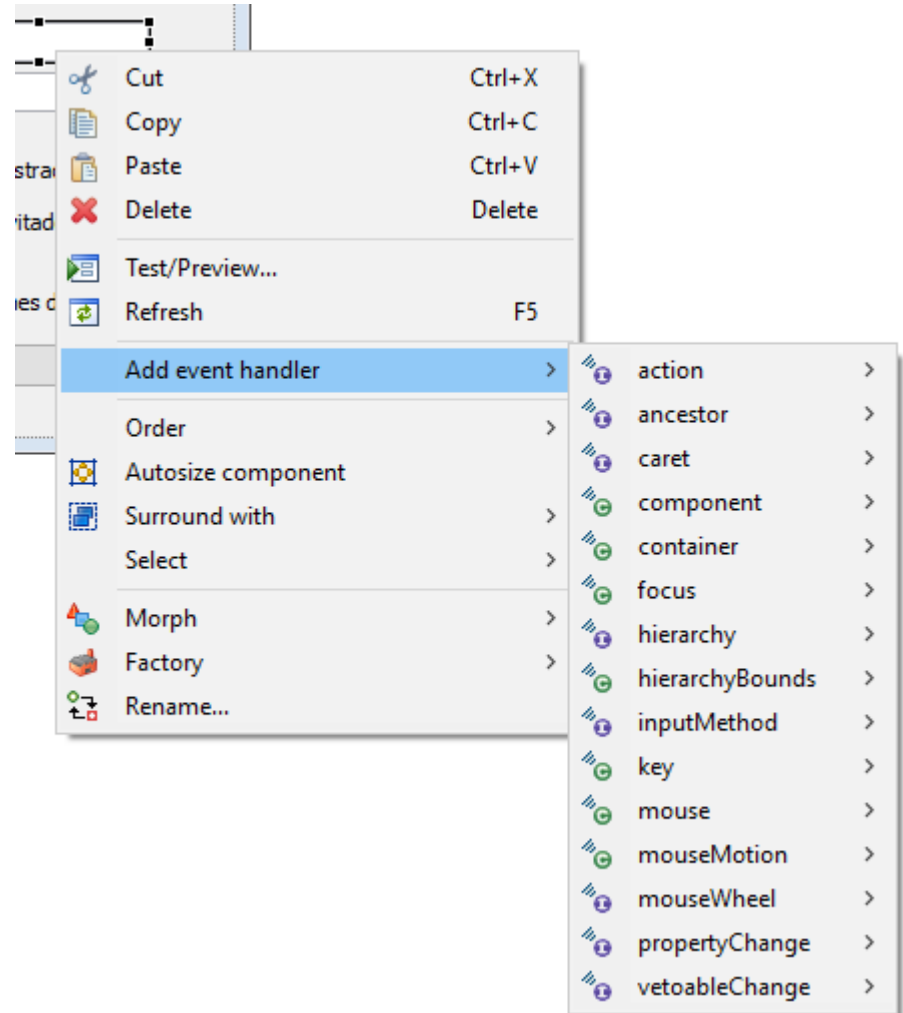
# Ejemplo de GUI en WindowBuilder



# Eventos en WindowBuilder

- Acción que el usuario realiza sobre un componente.

| Componentes    | Eventos utilizados |
|----------------|--------------------|
| JFrame         | windowOpened       |
| JTextField     | keyPressed         |
| JPasswordField | keyPressed         |
| JRadioButton   | actionPerformed    |
| JCheckBox      | actionPerformed    |
| JComboBox      | itemStateChanged   |
| JButton        | actionPerformed    |



# Cuadros de diálogo: Mensajes

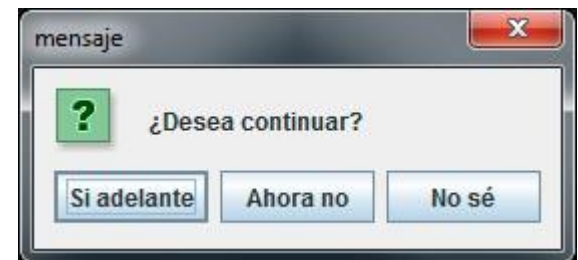
```
public class Gui18 extends JFrame {  
  
    public Gui18() {  
        super("Título de la ventana");  
        setLayout(new FlowLayout());  
        setSize(200, 100); // pack();  
        setVisible(true); // show();  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        // if ocurre algo  
        JOptionPane.showMessageDialog(null,  
            "Debe introducir datos en todos los campos",  
            "Error de entrada ",  
            JOptionPane.ERROR_MESSAGE);  
    }  
  
    public static void main(String[] args) {  
        Gui18 f = new Gui18();  
    }  
}
```



- Investigar para obtener el cuadro de diálogo que permita mostrar información al usuario.

# Cuadros de diálogo: Cuadro de opciones

```
public class Gui19 extends JFrame {  
  
    private final Container p;  
  
    public Gui19() {  
        super("Título de la ventana");  
        p = getContentPane();  
        setLayout(new FlowLayout());  
        setSize(200, 100);  
        setVisible(true);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        Object[] textoOpciones = {"Si adelante", "Ahora no", "No sé"};  
        int opcion = JOptionPane.showOptionDialog(null,  
            "¿Desea continuar?", "mensaje",  
            JOptionPane.YES_NO_CANCEL_OPTION,  
            JOptionPane.QUESTION_MESSAGE, null, textoOpciones,  
            textoOpciones[0]);  
    }  
  
    public static void main(String[] args) {  
        Gui19 f = new Gui19();  
    }  
}
```





# Ejercicio

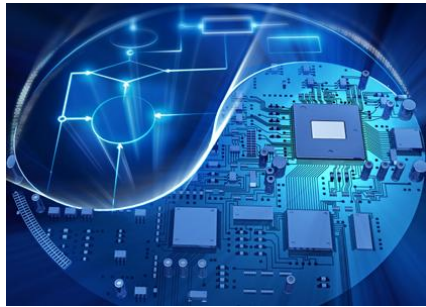
- Tras el diseño de la GUI realizado en la diapositiva 18, añadir a cada componente un evento
- Diseñar las GUIs de la diapositiva 21 y añadir un evento a cada componente. Probar a realizar una navegación de una GUI a otra (de Login a Lista de Compra)
- Probar a incorporar en uno de los eventos un cuadro de diálogo (mensaje, opciones, etc.)

# Práctica 1: GUI en Gambas

- Realizar una GUI en Gambas que contenga 12 componentes distintos, 6 eventos diferentes y 4 cuadros de diálogo vistos en clase: Warning, Error, Information y Question
- La temática será libre y los requisitos de usuario de la interfaz será aportada por el profesor. La navegación sobre los diferentes datos a introducir deben seguir una lógica con sentido común y debe haber campos obligatorios a tratar
- La valoración de esta práctica es de 1.5 puntos

# Práctica 2: GUI en WindowBuilder

- Realizar tres GUIs en WindowBuilder: Una de Login, otra de Registro y otra para realizar traducciones de palabras
- Si el logeo es correcto, se abre la interfaz de traducción donde se podrá traducir una palabra del español al inglés y/o al francés
- La navegación sobre los diferentes datos a introducir deben seguir una lógica con sentido común y debe haber campos obligatorios a tratar como tal. Se deben incluir 3 cuadros de diálogos distintos
- La valoración de esta práctica es de 2.5 puntos



# DESARROLLO DE APLICACIONES MULTIPLATAFORMA

## Desarrollo de Interfaces

Equivalencia en créditos ECTS (BOE): 9

### Bibliografía:

- Desarrollo de Interfaces. GFGS. Autor: FERRER MARTÍNEZ, JUAN. Editorial: RaMa
- Desarrollo de Interfaces. Autor: VICENTE CARRO, JUAN LUIS. Editorial: Paraninfo
- GAMBAS: <http://gambas.sourceforge.net/en/main.html>
- WINDOWBUILDER: <https://eclipse.org/windowbuilder/>
- ECLIPSE Documentation: <https://help.eclipse.org/oxygen/index.jsp>
- Apuntes proporcionados por el profesor (Moodle)