



El futuro digital  
es de todos

MinTIC

# Unidad 5

14 - Componentes Básicos  
SWING

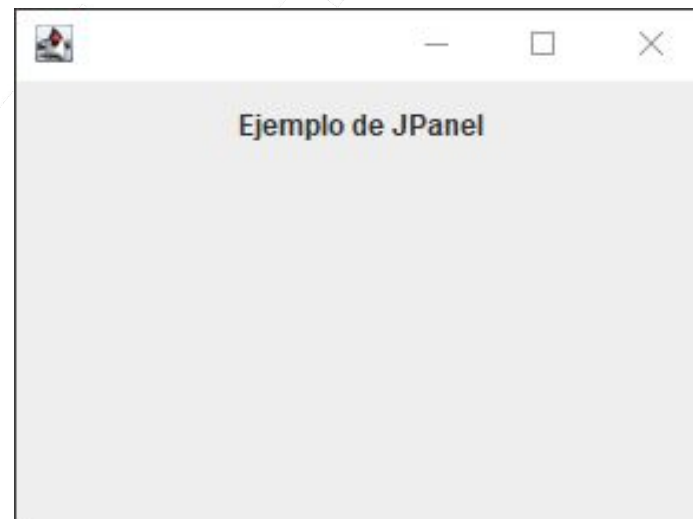


# JPanel y JLabel

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame{
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JPanel p = new JPanel();
        p.add(new JLabel("Ejemplo de JPanel"));
        c.add(p);
        setSize(200 ,200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana() ;
    }
}
```



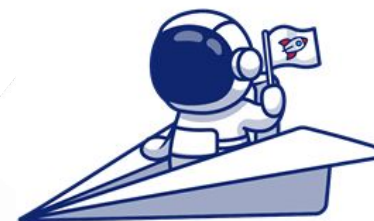
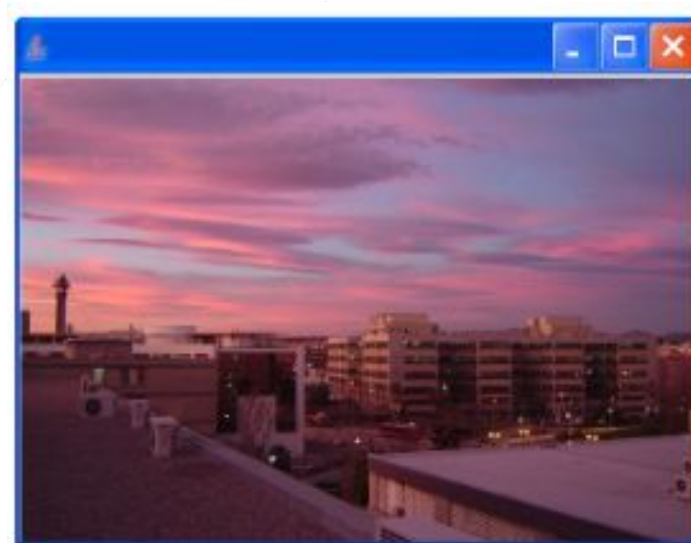


# ImageIcon

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;

public class Ventana extends JFrame{
    public Ventana(String fich) {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        ImageIcon ii = new ImageIcon(fich);
        c.add(new JLabel("", ii, JLabel.CENTER));
        setSize(650 ,500);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana(args[0]);
    }
}
```



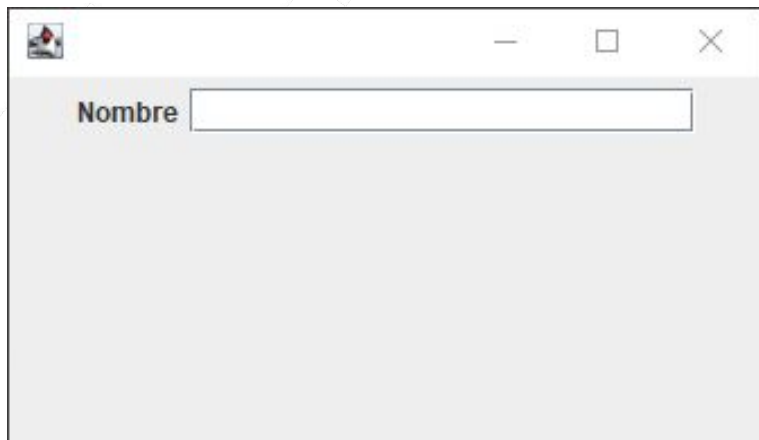


# JTextField

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JTextField campoTexto = new JTextField(20);
        c.add(new JLabel("Nombre"));
        c.add(campoTexto);
        setSize(350, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```





# JTextArea

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JTextArea area = new JTextArea(8, 20);
        c.add(new JLabel("Observaciones"));
        c.add(area);
        setSize(350, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```



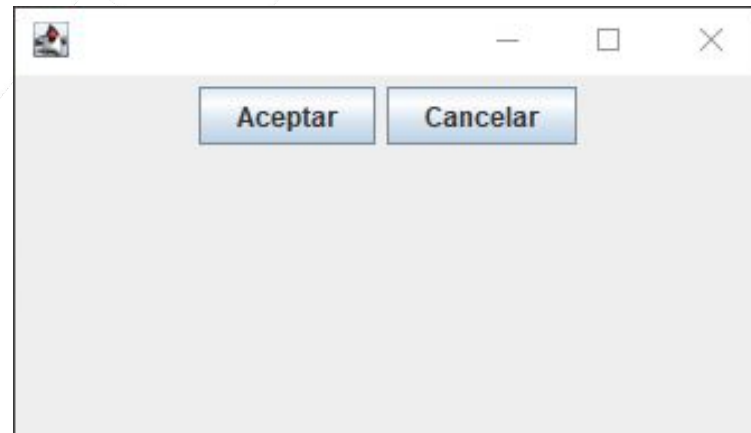


# JButton

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JButton b1 = new JButton("Aceptar");
        JButton b2 = new JButton("Cancelar");
        c.add(b1);
        c.add(b2);
        setSize(350, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```



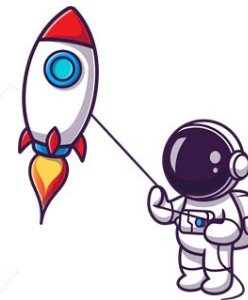
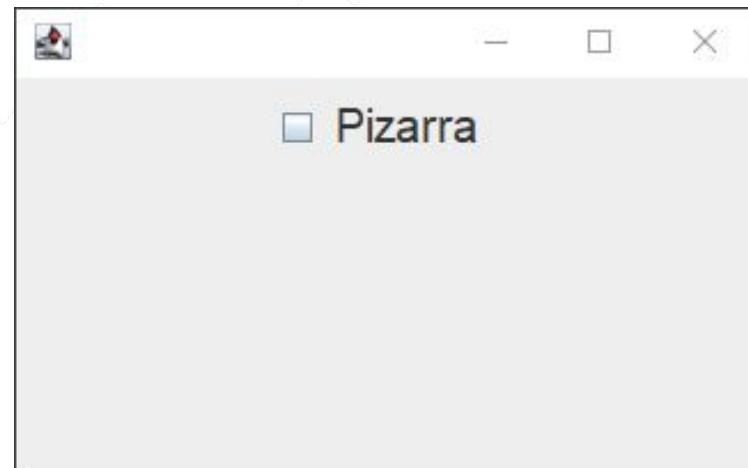


# JCheckBox

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JCheckBox cb = new JCheckBox("Pizarra");
        cb.setFont(new Font("Arial", Font.PLAIN, 20));
        c.add(cb);
        setSize(200, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```





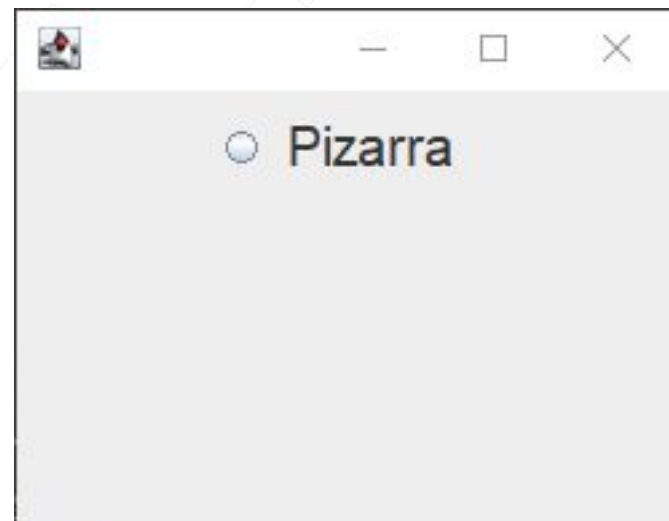


# JRadioButton

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JRadioButton cb = new JRadioButton("Pizarra");
        cb.setFont(new Font("Arial", Font.PLAIN, 20));
        c.add(cb);
        setSize(200, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```





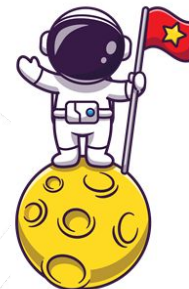
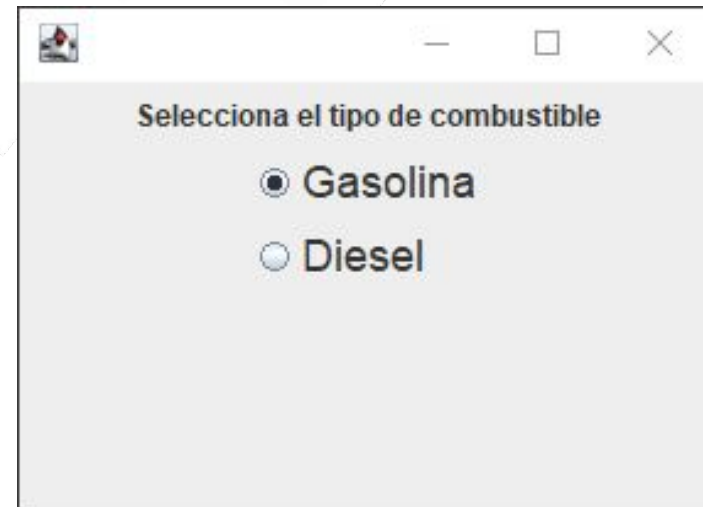


# ButtonGroup

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        c.add(new JLabel("Selecciona el tipo de combustible"));
        Font fuente = new Font("Arial", Font.PLAIN, 18);
        JRadioButton gas = new JRadioButton("Gasolina");
        gas.setFont(fuente);
        JRadioButton die = new JRadioButton("Diesel");
        die.setFont(fuente);
        // Agrupamos los botones
        ButtonGroup grupo = new ButtonGroup();
        grupo.add(gas);
        grupo.add(die);
        JPanel radioPanel = new JPanel();
        radioPanel.setLayout(new GridLayout(0, 1));
        radioPanel.add(gas);
        radioPanel.add(die);
        c.add(radioPanel);
        setSize(300, 300);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```



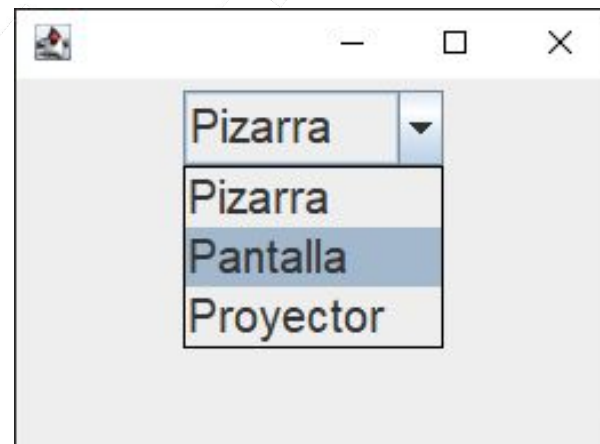


# JComboBox

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JComboBox<String> cb = new JComboBox<>();
        cb.setFont(new Font("Arial", Font.PLAIN, 20));
        cb.addItem("Pizarra");
        cb.addItem("Pantalla");
        cb.addItem("Proyector");
        c.add(cb);
        setSize(200, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```



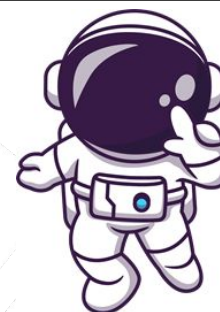
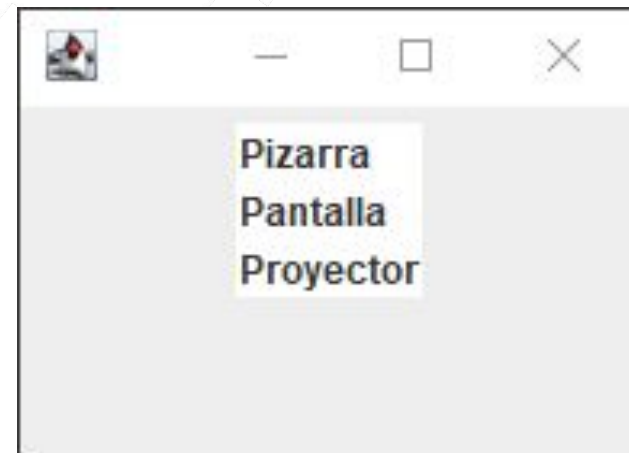


# JList

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;

public class Ventana extends JFrame {
    public Ventana() {
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        String[] datos = { "Pizarra", "Pantalla",
        "Proyector"};
        JList<String> lista = new JList<>(datos);
        c.add(lista);
        setSize(200, 200);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Ventana();
    }
}
```





# JOptionPane

- Permite crear nuevos cuadros de diálogo o usar algunos de los más comunes:
  - Message, para informar al usuario sobre algún hecho relevante
  - Confirm, para realizar una pregunta al usuario con las posibilidades básicas de respuesta de Sí, No o Cancelar.
  - Input, para solicitar una entrada del usuario
  - Option, permite crear una ventana personalizada de cualquiera de los tipos anteriores





# JOptionPane.showMessageDialog

```
//default title and icon
```

```
JOptionPane.showMessageDialog(frame, "Eggs are not supposed to be green.",  
                                "Message", JOptionPane.INFORMATION_MESSAGE);
```



```
//custom title, warning icon
```

```
JOptionPane.showMessageDialog(frame, "Eggs are not supposed to be green.",  
                                "Inane warning", JOptionPane.WARNING_MESSAGE);
```



# JOptionPane.showMessageDialog

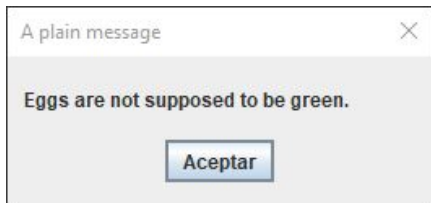
```
// custom title, error icon
```

```
JOptionPane.showMessageDialog(frame, "Eggs are not supposed to be green.",  
                                "Inane error",JOptionPane.ERROR_MESSAGE);
```



```
// custom title, no icon
```

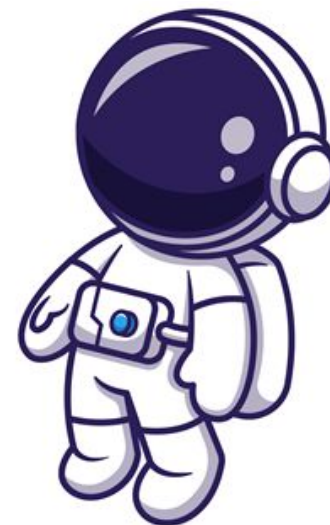
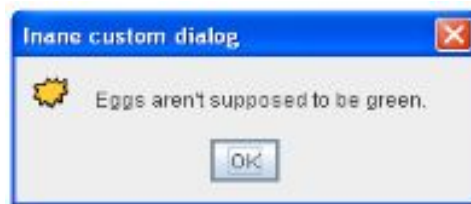
```
JOptionPane.showMessageDialog(frame, "Eggs are not supposed to be green.",  
                                "A plain message",JOptionPane.PLAIN_MESSAGE);
```





# JOptionPane.showMessageDialog

```
// custom title, custom icon  
JOptionPane.showMessageDialog(frame, "Eggs are not supposed to be green.",  
    "Inane custom dialog", JOptionPane.INFORMATION_MESSAGE, icon);
```







# JOptionPane.showConfirmDialog

```
// default icon, custom title  
int option = JOptionPane.showConfirmDialog(frame, "Would you like green eggs and  
ham?", "An Inane Question", JOptionPane.YES_NO_OPTION);
```



```
Object[] options = { "Yes, please", "No way!" };  
int option = JOptionPane.showOptionDialog(frame, "Would you like green eggs and  
ham?", "A Silly Question", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE,  
null, options, options[0]);
```



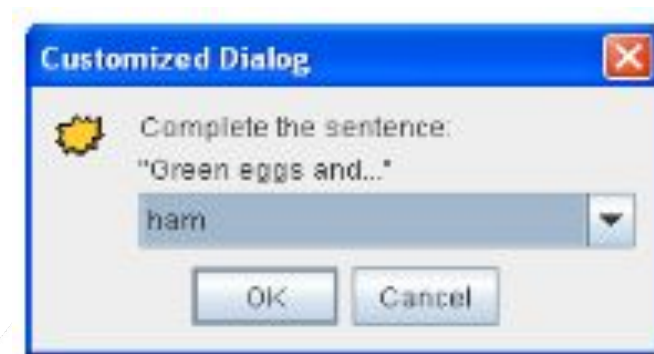


# JOptionPane.showInputDialog

```
Object[] possibilities = { "ham", "spam", "yam" };
String s = (String) JOptionPane.showInputDialog(frame, "Complete the sentence:\n"
    + "\"Green eggs and...\\"", "Customized Dialog", JOptionPane.PLAIN_MESSAGE,
    icon, possibilities, "ham");

// If a string was returned, say so.
if ((s != null) && (s.length() > 0)) {
    etiqueta.setText("Green eggs and... " + s + "!");
    return;
}

// If you're here, the return value was null/empty.
etiqueta.setText("Come on, finish the sentence!");
```





# JOptionPane.showOptionDialog

```
// Custom button text
Object[] options = { "Yes, please", "No, thanks", "No eggs, no ham!" };

int option = JOptionPane.showOptionDialog(frame,
    "Would you like some green eggs to go with that ham?",
    "A Silly Question", JOptionPane.YES_NO_CANCEL_OPTION,
    JOptionPane.QUESTION_MESSAGE, null, options, options[2]);
```





# Valores de retorno

- Si el método showXXXDialog devuelve un entero, los posibles valores de retorno son:
  - JOptionPane.YES\_OPTION
  - JOptionPane.NO\_OPTION
  - JOptionPane.CANCEL\_OPTION
  - JOptionPane.OK\_OPTION
  - JOptionPane.CLOSED\_OPTION
- En otro caso, devolverá un String con la opción seleccionada (como en los ejemplos que hemos visto)

