Componentes uno por uno

Ahora veremos los componentes de manera más detallada, la mayoría de ellos heredarán los métodos de la clase Component, por lo que tendrán muchos métodos en común.

Es indispensable tener en cuenta la siguiente página a la hora de realizar interfaces, donde podemos consultar todos los métodos. **En estos apuntes, sólo aparecen algunos de ellos.**

https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/package-summary.html

Clase JComponent

A continuación empezaremos a ver los componentes que vamos a necesitar para crear interfaces. Muchos de ellos, heredan métodos de la clase JComponent. Podemos consultarlos en la web https://docs.oracle.com/javase/8/docs/api/?javax/swing/JComponent.html
A continuación veremos los más usados.

Clase JComponent

Métodos

setVisible(boolean visible): Establece la visibilidad del componente.

isVisible(): Devuelve true si el componente es visible.

setEnabled(boolean enabled): Activa o desactiva el componente.

isEnabled(): Devuelve true si el componente está habilitado.

setSize(int width, int height): Establece el tamaño del componente en píxeles.

setForeground(Color color): Establece el color del texto o el color principal del componente. Color puede ser Color.BLUE, Color.RED...

setFont(Font font): Define la fuente del componente.

Clase JComponent

Métodos

setBorder(Border border): Establece el borde del componente usando una instancia de Border.

setToolTipText(String text): Define el texto emergente que aparece cuando se pasa el ratón sobre el componente.

add(Component comp): Añade un componente secundario.

remove(Component comp): Elimina un componente secundario.

Clase JAbstractButton

De esta clase heredarán los componentes JButton, JMenu y JToggleButton (JCheckbox e JRadioButton)

Métodos

setText(String text): Establece el texto que se muestra en el botón.

getText(): Devuelve el texto actual del botón.

setIcon(Icon icon): Asigna un icono al botón.

getIcon(): Obtiene el icono actual del botón.

setHorizontalTextPosition(int textPosition): Establece la posición horizontal del texto en relación con el icono.

setVerticalTextPosition(int textPosition): Establece la posición vertical del texto en relación con el icono.

JLabel

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JLabel

Muestra texto, imágenes o ambos en una interfaz gráfica.

Constructors

Constructor and Description
`JLabel()` Creates a JLabel instance with no image and with an empty string for the title.
`JLabel(Icon image)` Creates a JLabel instance with the specified image.
`JLabel(Icon image, int horizontalAlignment)` Creates a JLabel instance with the specified image and horizontal alignment.
`JLabel(String text)` Creates a JLabel instance with the specified text.
`JLabel(String text, Icon icon, int horizontalAlignment)` Creates a JLabel instance with the specified text, image, and horizontal alignment.
`JLabel(String text, int horizontalAlignment)` Creates a JLabel instance with the specified text and horizontal alignment.

JLabel

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JLabel

Métodos

setText(String text): Establece el texto de la etiqueta

getText(): Obtiene el texto de la etiqueta

setIcon(Icon icon): Establece un icono (imagen)

getIcon(): Obtiene el icono actual

setHorizontalAlignment(int alignment): Establece la alineación horizontal del contenido

setVerticalAlignment(int alignment): Establece la alineación vertical del contenido

JTextField

Clase JTextField

https://docs.oracle.com/javase/8/docs/api/javax/swing/JTextField.html

```
private JTextField nombre;
public NewJFrame() {
    JFrame frame = new JFrame("Formulario en Java");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 300);

    // Crear un panel para contener los elementos del formulario
    JPanel panel = new JPanel();
    // Usamos un GridLayout para organizar los campos
    panel.setLayout(new GridLayout(4, 2));


    nombre=new JTextField();//creamos un objeto de tipo JTexTField
    nombre.setText("Escribe aquí tu nombre");

    panel.add(nombre);
    // Agregar el panel del formulario al frame
    frame.add(panel, BorderLayout.CENTER);

    frame.setVisible(true);
}
```

JTextField

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.text.JTextComponent
          javax.swing.JTextField

```
public JTextField(String text)
```

Constructs a new `TextField` initialized with the specified text. A default model is created and the number of columns is 0.

```
public JTextField(String text)
```

Constructs a new `TextField` initialized with the specified text. A default model is created and the number of columns is 0.

```
public JTextField(int columns)
```

Constructs a new empty `TextField` with the specified number of columns. A default model is created and the initial string is set to `null`.

```
public JTextField(String text,
                  int columns)
```

Constructs a new `TextField` initialized with the specified text and columns. A default model is created.

JTextField

Métodos

public int getHorizontalAlignment()

setHorizontalAlignment(int alignment)

getColumns()

public void setColumns(int columns)

public void setFont(Font f)

public void addActionListener(ActionListener l)

public void removeActionListener(ActionListener l)

JTextField

```java
//SeetHorizontalAlignment: Alinear el texto en el JTextField: puede ser LEFT, CENTER, RIGHT, LEADING o TRAILING
textField.setHorizontalAlignment(JTextField.CENTER);

//getHorizontalAlignment() - Obtener la alineación del texto
System.out.println("Alineación horizontal actual: " + textField.getHorizontalAlignment());

//setColumns(int columns) - Establecer el número de columnas
textField.setColumns(20); // Esto afecta el tamaño preferido del campo de texto

//getColumns() - Obtener el número de columnas
int columns = textField.getColumns();
System.out.println("Número de columnas: " + columns);

//setFont(Font f) - Cambia la fuente
textField.setFont(new Font("Arial", Font.BOLD, 16));  // Fuente Arial, negrita, tamaño 16
```

JTextField

```
//addActionListener(ActionListener l) - Añadir un ActionListener
textField.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println("Texto añadido: " + textField.getText());
    }
});
```

JPasswordField

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.text.JTextComponent
          javax.swing.JTextField
            javax.swing.JPasswordField

Constructors

Constructor and Description
JPasswordField() Constructs a new JPasswordField, with a default document, null starting text string, and 0 column width.
JPasswordField(Document doc, String txt, int columns) Constructs a new JPasswordField that uses the given text storage model and the given number of columns.
JPasswordField(int columns) Constructs a new empty JPasswordField with the specified number of columns.
JPasswordField(String text) Constructs a new JPasswordField initialized with the specified text.
JPasswordField(String text, int columns) Constructs a new JPasswordField initialized with the specified text and columns.

JPasswordField

Métodos

char [] getPassword(): devuelve la contraseña como un array de caracteres.

void setEchoChar (char c): establece el carácter que se usará para “enmascarar” la contraseña en el campo.

char getEchoChar(): devuelve el carácter actual utilizado para enmascarar el texto introducido en el JPasswordField.

boolean echoCharIsSet(): devuelve true si se ha definido un carácter de eco, y false si no se ha configurado.

Otros métodos heredados de JTextField como getColumns, setColumns...

JPasswordField

```
public EjPasswordField() {
    JFrame frame = new JFrame("Ejemplo de PasswordField");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 200);

    JPasswordField passwordField = new JPasswordField("micontrasena");
    passwordField.setText("holamundo");
    //obsoleta, ahora se usa getPassword() por motivos de seguridad
    String password = passwordField.getText();
    char[] password2 = passwordField.getPassword();
    System.out.println("La contraseña es: "+password+" "+password2);
    frame.add(passwordField);
    frame.setVisible(true);
}
```

JButton

Constructors

Constructor and Description
`JButton()` Creates a button with no set text or icon.
`JButton(Action a)` Creates a button where properties are taken from the `Action` supplied.
`JButton(Icon icon)` Creates a button with an icon.
`JButton(String text)` Creates a button with text.
`JButton(String text, Icon icon)` Creates a button with initial text and an icon.

JButton

Métodos

void setText(String t): modifica el texto que aparece en el botón.

String getText(): devuelve el texto del botón.

void addActionListener(ActionListener l): define las acciones que se realizarán cuando el botón es presionado.

void setEnabled(boolean enabled): habilita o deshabilita el botón. Deshabilitado el botón no responde a eventos de clic.

boolean isEnabled(): devuelve true si el botón está habilitado y false si está deshabilitado.

void setIcon(Icon icon): establece un icono en el botón.

JButton

Métodos

Icon getIcon(): devuelve el icono que está en el botón

void setToolTipText(String text): establece un texto de ayuda que aparece cuando el usuario deja el cursor sobre el botón.

String getToolTipText(): devuelve el texto de ayuda que se ha establecido para el botón.

void setForeground(Color color): cambia el color del texto del botón.

void setBackground(Color color): cambia el color de fondo del botón.

void setBorder(Border border): establece un borde personalizado para el botón

JButton. Ejemplos

```
public EjJButton() {
    JFrame frame = new JFrame("JSlider Example");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 500);
    JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 100, 100));

    JButton jbPresionar = new JButton();
    //modifico el texto del botón
    jbPresionar.setText("Presionar");
    //obtengo el texto del botón
    String texto = jbPresionar.getText();
    System.out.println("Texto del botón: " + texto);
    //añado el texto que debe aparecer cuando el ratón está sobre el botón
    jbPresionar.setToolTipText("Presiona para guardar los cambios");
    String toolTip = jbPresionar.getToolTipText();
    System.out.println("Texto del tooltip: " + toolTip);
    // El texto del botón será rojo
    jbPresionar.setForeground(Color.RED);
    // El fondo del botón será verde
    jbPresionar.setBackground(Color.GREEN);
```

Continúa en la siguiente diapositiva

JButton. Ejemplos

```
    //añado un evento
    jbPresionar.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            //Muestra un mensaje cuando hace clic en el botón
            System.out.println("Botón presionado");
        }
    });

    jbPresionar.setEnabled(false);  // deshabilito el botón
    //añado el botón al panel
    panel.add(jbPresionar);
    //añado el panel al frame
    frame.add(panel);
    frame.setVisible(true);
}
```

JCheckbox

Constructors

Constructor and Description
`JCheckBox()` Creates an initially unselected check box button with no text, no icon.
`JCheckBox(Action a)` Creates a check box where properties are taken from the Action supplied.
`JCheckBox(Icon icon)` Creates an initially unselected check box with an icon.
`JCheckBox(Icon icon, boolean selected)` Creates a check box with an icon and specifies whether or not it is initially selected.
`JCheckBox(String text)` Creates an initially unselected check box with text.
`JCheckBox(String text, boolean selected)` Creates a check box with text and specifies whether or not it is initially selected.
`JCheckBox(String text, Icon icon)` Creates an initially unselected check box with the specified text and icon.
`JCheckBox(String text, Icon icon, boolean selected)` Creates a check box with text and icon, and specifies whether or not it is initially selected.

JCheckbox

Métodos

boolean isSelected(): devuelve true si el JCheckbox está seleccionado, false si no lo está.

void setSelected(boolean selected): establece el estado del JCheckbox, si pasas true, se marcará, si pasas false, se deseleccionará.

void setText(String text): establece el texto que aparece al lado del JCheckbox.

void setEnabled(boolean enabled): habilita y deshabilita el JCheckbox.

boolean isEnabled(): devuelve true si el JCheckbox está habilitado o false si está deshabilitado.

JCheckbox

Métodos

void setIcon (Icon icon)/Icon getIcon(): establece o devuelve el icono que se muestra al lado del JCheckbox.

Otros que ya vimos anteriormente como setToolTipText, getToolTipText, setForeground, setBackground...

JCheckbox. Ejemplos

```
public EjCheckbox() {
    JFrame frame = new JFrame("JSlider Example");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 500);
    //creo un panel con un layout de tipo FlowLayout
    JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 100, 100));

    JCheckBox jcTerminos = new JCheckBox("Aceptar términos");
    if (jcTerminos.isSelected()) {
        System.out.println("El checkbox está marcado.");
    }
    jcTerminos.setText("Aceptar términos");
    // selecciono el checkbox
    jcTerminos.setSelected(true);
    //Deshabilito el checkbox
    jcTerminos.setEnabled(false);
    if (!jcTerminos.isEnabled()) {
        System.out.println("El checkbox está deshabilitado.");
    }
    //vuelvo a habilitar el checkbox
    jcTerminos.setEnabled(true);
    //añado los elementos al panel y frame
    panel.add(jcTerminos);
    frame.add(panel);
    frame.setVisible(true);
}
```

JRadioButton

Constructors

Constructor and Description
`JRadioButton()` Creates an initially unselected radio button with no set text.
`JRadioButton(Action a)` Creates a radiobutton where properties are taken from the Action supplied.
`JRadioButton(Icon icon)` Creates an initially unselected radio button with the specified image but no text.
`JRadioButton(Icon icon, boolean selected)` Creates a radio button with the specified image and selection state, but no text.
`JRadioButton(String text)` Creates an unselected radio button with the specified text.
`JRadioButton(String text, boolean selected)` Creates a radio button with the specified text and selection state.
`JRadioButton(String text, Icon icon)` Creates a radio button that has the specified text and image, and that is initially unselected.
`JRadioButton(String text, Icon icon, boolean selected)` Creates a radio button that has the specified text, image, and selection state.

JRadioButton

Métodos

boolean isSelected(): devuelve true si el JRadioButton está seleccionado, false si no lo está.

void setSelected(boolean selected): establece el estado del JRadioButton, si pasas true, se marcará, si pasas false, se deseleccionará.

void setText(String text): establece el texto que aparece al lado del JRadioButton.

void setEnabled(boolean enabled): habilita y deshabilita el JRadioButton.

boolean isEnabled(): devuelve true si el JRadioButton está habilitado o false si está deshabilitado.

JSlider

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JSlider

Permite seleccionar un valor de un rango deslizándolo.

Constructors

Constructor and Description
`JSlider()` Creates a horizontal slider with the range 0 to 100 and an initial value of 50.
`JSlider(BoundedRangeModel brm)` Creates a horizontal slider using the specified BoundedRangeModel.
`JSlider(int orientation)` Creates a slider using the specified orientation with the range `0` to `100` and an initial value of `50`.
`JSlider(int min, int max)` Creates a horizontal slider using the specified min and max with an initial value equal to the average of the min plus max.
`JSlider(int min, int max, int value)` Creates a horizontal slider using the specified min, max and value.
`JSlider(int orientation, int min, int max, int value)` Creates a slider with the specified orientation and the specified minimum, maximum, and initial values.

JSlider

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JSlider

setValue(int value): Establece el valor actual del deslizador.
getValue(): Devuelve el valor actual del deslizador.
setMinimum(int min): Define el valor mínimo que puede tener el deslizador.
getMinimum(): Obtiene el valor mínimo del deslizador.
setMaximum(int max): Establece el valor máximo del deslizador.
getMaximum(): Obtiene el valor máximo del deslizador.
setExtent(int extent): Establece la distancia entre el valor mínimo y máximo representado por el "thumb" (el cursor que se arrastra).
getExtent(): Obtiene la distancia entre el valor mínimo y máximo actual del deslizador.

JSlider

java.lang.Object
java.awt.Component
java.awt.Container
javax.swing.JComponent
javax.swing.JSlider

setPaintTicks(boolean b): Controla si los "ticks" (marcas a lo largo del deslizador) deben pintarse o no.
getPaintTicks(): Devuelve true si los "ticks" están habilitados para mostrarse.
setMajorTickSpacing(int n): Establece el espacio entre los "ticks" mayores (marcas más destacadas).
setMinorTickSpacing(int n): Establece el espacio entre los "ticks" menores (marcas menos destacadas).
setPaintLabels(boolean b): Indica si se deben pintar las etiquetas junto a los "ticks".
getPaintLabels(): Devuelve true si las etiquetas están habilitadas.

JSlider

java.lang.Object
java.awt.Component
java.awt.Container
javax.swing.JComponent
javax.swing.JSlider

setOrientation(int orientation): Establece la orientación del deslizador (puede ser JSlider.HORIZONTAL o JSlider.VERTICAL).
getOrientation(): Obtiene la orientación del deslizador.
setSnapToTicks(boolean b): Si está habilitado, el deslizador se "ajustará" automáticamente a los "ticks" más cercanos cuando el usuario lo mueva.
getSnapToTicks(): Devuelve true si el ajuste automático a los "ticks" está habilitado.
setInverted(boolean b): Invierte el orden del deslizador, haciendo que los valores aumenten hacia la izquierda o abajo, dependiendo de la orientación.
getInverted(): Devuelve true si el deslizador está invertido.

JSlider

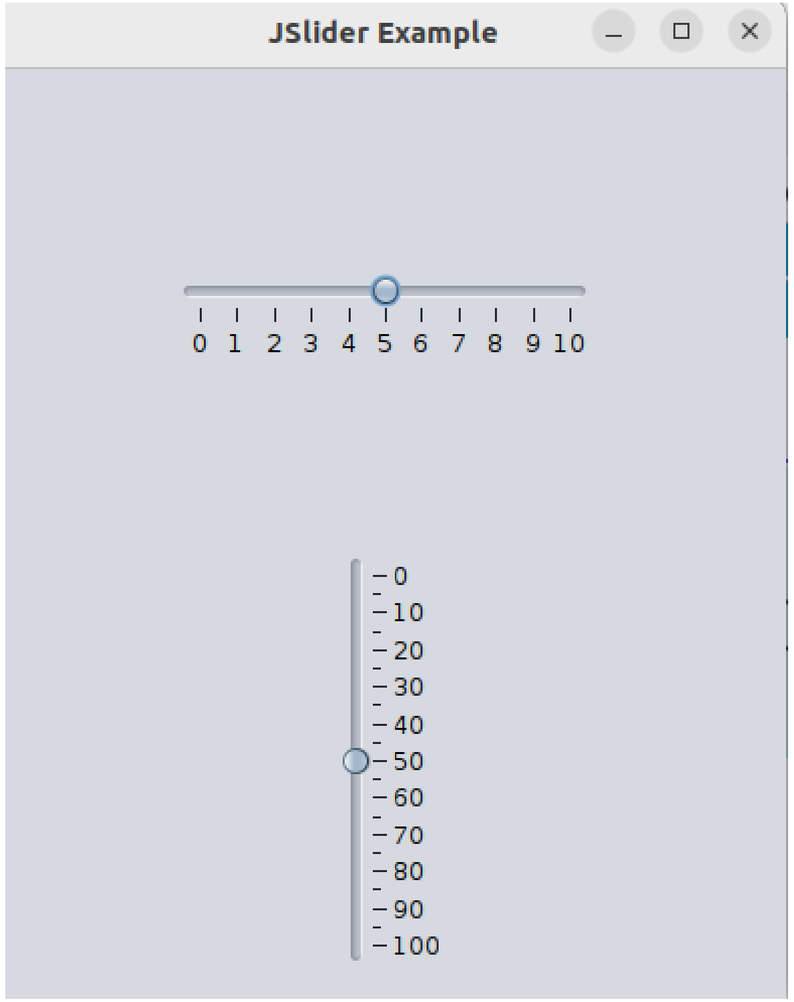
java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JSlider

addChangeListener(ChangeListener listener): Añade un oyente para eventos de cambio en el valor del deslizador.

removeChangeListener(ChangeListener listener): Elimina un oyente de eventos de cambio.

JSlider.Ejemplos

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JSlider



JSlider.Ejemplos

java.lang.Object
- java.awt.Component
  - java.awt.Container
    - javax.swing.JComponent
      - javax.swing.JSlider

```
public EjJSlider() {
    JFrame frame = new JFrame("Exemplo de JSlider");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 500);
    JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 100, 100));
    //JSlider con rango de 0 a 10 e valor inicial de 5
    JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 10, 5);

    //Configuro os "ticks" maiores e menores
    //Cada 1 unidade haberá un tick maior
    slider.setMajorTickSpacing(1);   // Cada 1 unidades habrá un tick mayor
    //Cada 1 unidade haberá un tick menor
    slider.setMinorTickSpacing(1);

    //Fago visibles os ticks e os números
    slider.setPaintTicks(true);
    slider.setPaintLabels(true);

    //Engado un ChangeListener para capturar os cambios no JSlider
    slider.addChangeListener(new ChangeListener() {
        public void stateChanged(ChangeEvent e) {
            System.out.println("Valor actual: " + slider.getValue());
        }
    });
    //Engado un ChangeListener para capturar os cambios no JSlider
    slider.addChangeListener(new ChangeListener() {
        public void stateChanged(ChangeEvent e) {
            System.out.println("Valor actual: " + slider.getValue());
        }
    });

    JSlider slider_2 = new JSlider(JSlider.HORIZONTAL, 0, 100, 50);
    slider_2.setOrientation(JSlider.VERTICAL);
    // Configurar os "ticks" maiores e menores
    slider_2.setMajorTickSpacing(10);    // Cada 1 unidade haberá un tick maior
    slider_2.setMinorTickSpacing(5);     // Cada 1 unidade haberá un tick menor
    slider_2.setInverted(true);//invertimos os números (0-100 dende enriba)
    // fago visibles os ticks e los números
    slider_2.setPaintTicks(true);
    slider_2.setPaintLabels(true);
    slider_2.setSnapToTicks(true);
    // engado un ChangeListener para capturar os cambios no jslider
    slider_2.addChangeListener(new ChangeListener() {
        public void stateChanged(ChangeEvent e) {
            System.out.println("Valor actual: " + slider_2.getValue());
        }
    });
    //engado os compoñenetes o panel e o panel ao frame
    panel.add(slider);
    panel.add(slider_2);
    frame.add(panel);
    frame.setVisible(true);
}
```

JButtonGroup

java.lang.Object
    javax.swing.ButtonGroup

Se utiliza para agrupar varios botones de opción o casillas de verificación de modo que solo uno de ellos pueda estar seleccionado a la vez.

Constructors

Constructor	Description
`ButtonGroup()`	Creates a new ButtonGroup.

JButtonGroup

java.lang.Object
javax.swing.ButtonGroup

Métodos

add(AbstractButton button): añade un botón al grupo.

remove(AbstractButton button): elimina un botón del grupo.

clearSelection(): desmarca el botón que está actualmente seleccionado en el grupo.

getElements(): devuelve una enumeración de los botones en el grupo.

getSelection(): devuelve el modelo del botón seleccionado en el grupo.

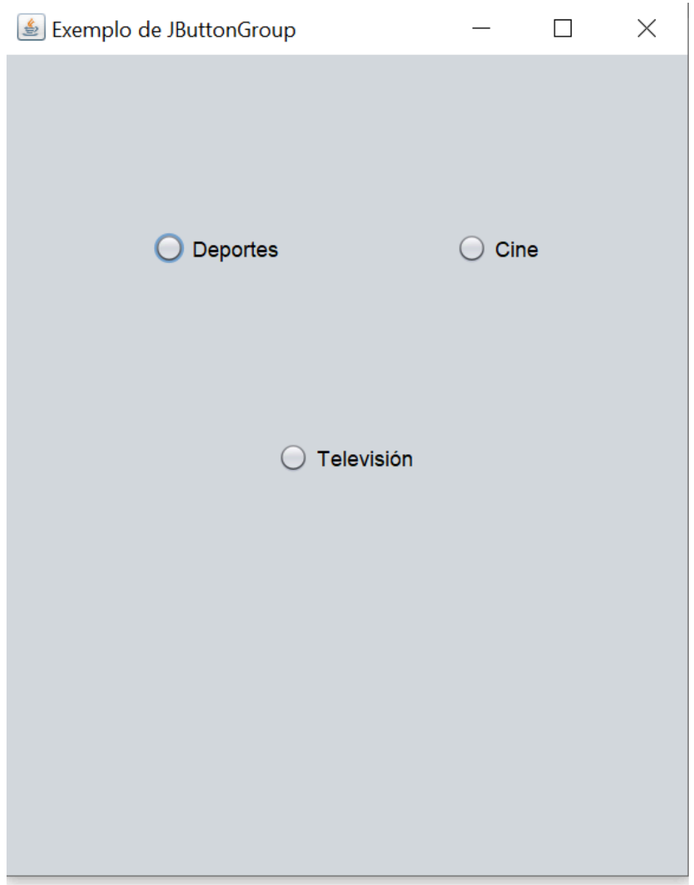
setSelected(ButtonModel model, boolean selected): selecciona o deselecciona un botón en el grupo según el modelo pasado.

isSelected(ButtonModel model): verifica si el botón con el modelo dado está seleccionado.

getButtonCount(): devuelve el número de botones del grupo.

JButtonGroup. Ejemplos

java.lang.Object
javax.swing.ButtonGroup




```java
public EjButtonGroup() {
    JFrame frame = new JFrame("Exemplo de JButtonGroup");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 500);
    JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 100, 100));

    //creo un Buttongroup que agrupará todos os RadioButton
    ButtonGroup bgAficiones = new ButtonGroup();
    //creo os radiobutton
    JRadioButton jrbDeportes = new JRadioButton("Deportes");
    JRadioButton jrbCine = new JRadioButton("Cine");
    JRadioButton jrbtelevision = new JRadioButton("Televisión");

    //engado os radioButton al ButtonGroup
    bgAficiones.add(jrbDeportes);
    bgAficiones.add(jrbCine);
    bgAficiones.add(jrbtelevision);

    //engado os compoñentes ao panel e o panel ao frame.
    panel.add(jrbDeportes);
    panel.add(jrbCine);
    panel.add(jrbtelevision);
    frame.add(panel);
    //fago visible o formulario
    frame.setVisible(true);
}
```

JComboBox.

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JComboBox<E>

Permite seleccionar un elemento de una lista desplegable.

Constructors

Constructor and Description
JComboBox() Creates a JComboBox with a default data model.
JComboBox(ComboBoxModel<E> aModel) Creates a JComboBox that takes its items from an existing ComboBoxModel.
JComboBox(E[] items) Creates a JComboBox that contains the elements in the specified array.
JComboBox(Vector<E> items) Creates a JComboBox that contains the elements in the specified Vector.

```
String[] cbAficiones = { "Deportes", "Cine", "Televisión" };
JComboBox<String> comboBox = new JComboBox<>(cbAficiones);

Vector<String> vAficiones = new Vector<>();
vAficiones.add("Opción 1");
vAficiones.add("Opción 2");
JComboBox<String> v = new JComboBox<>(vAficiones);
```

JComboBox.

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JComboBox<E>

Métodos

addItem(E item): añade un elemento al final de la lista desplegable.

insertItemAt(E item, int index): inserta un elemento en una posición específica de la lista.

removeItem(Object item): elimina un elemento específico de la lista.

removeItemAt(int index): elimina el elemento en la posición dada.

removeAllItems(): elimina todos los elementos del JComboBox.

getItemAt(int index): devuelve el número de elementos en el combo box.

getSelectedItem(): devuelve el elemento actualmente seleccionado.

setSelectedItem(Object item): establece el elemento seleccionado. Si el elemento está en la lista, lo selecciona.

getSelectedIndex(): devuelve el índice del elemento seleccionado actualmente.

JComboBox.

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JComboBox<E>

Métodos

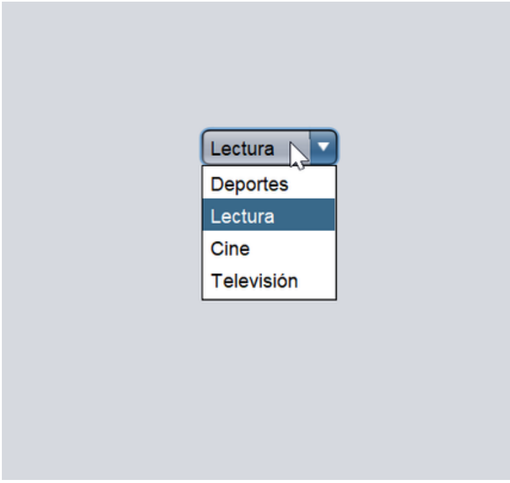
setSelectedIndex(int index): selecciona el elemento en el índice dado.

JComboBox. Ejemplos

java.lang.Object
    java.awt.Component
        java.awt.Container
            javax.swing.JComponent
                javax.swing.JComboBox<E>

```
//modificaciones en el JComboBox
cbAficiones.addItem("Televisión");//añadimos Televisión en el ComboBox
cbAficiones.insertItemAt("Lectura",1);//añadimos "Lectura" en la segunda posición del ComboBox
cbAficiones.setSelectedItem("Lectura");//establecemos "lectura" como el elemento seleccionado

//recorremos el ComboBox y obtenemos sus valores
for(int i=0;i<cbAficiones.getItemCount();i++){
    System.out.println("El elemento "+i+" es "+cbAficiones.getItemAt(i));
}
```



JOptionPane

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JOptionPane

Se utiliza para mostrar cuadros de diálogo modales para que el usuario pueda interactuar con la aplicación.

Tipos de diálogo:

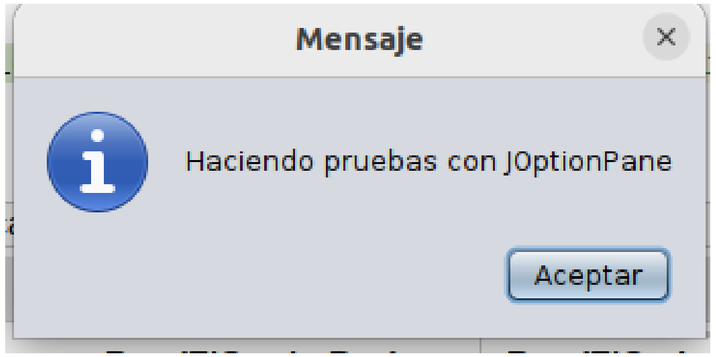
- Mostrar un mensaje simple.
- Obtener una entrada del usuario.
- Solicitar una confirmación.
- Mostrar un mensaje de advertencia o error.

JOptionPane

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JOptionPane

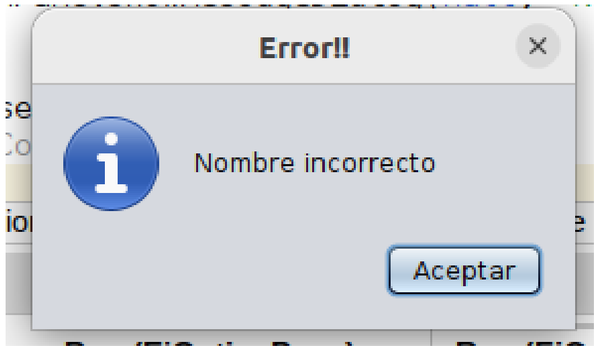
- Mostrar un mensaje simple.

```
JOptionPane.showMessageDialog(null, "Haciendo prueba con JOptionPane")
```



- Mostrar mensaje especificando el tipo de mensaje

```
JOptionPane.showMessageDialog(null, "Nombre incorrecto", "Error!!", JOptionPane.INFORMATION_MESSAGE);
```

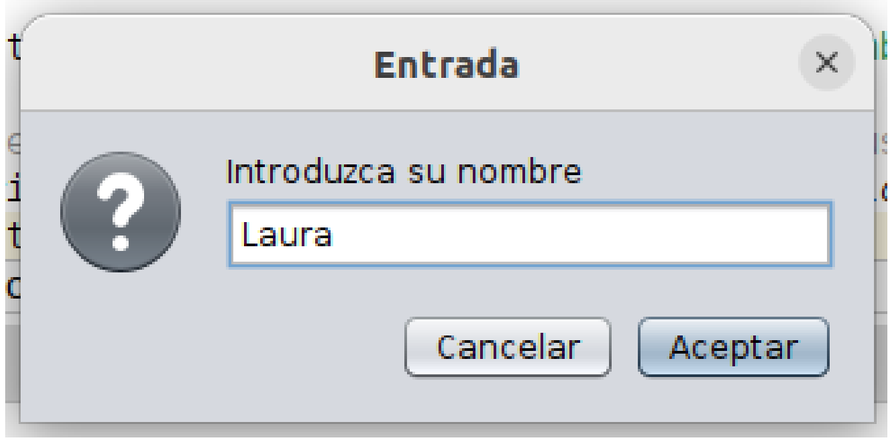


JOptionPane

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JOptionPane

- Mostrar un mensaje pidiendo una entrada del usuario

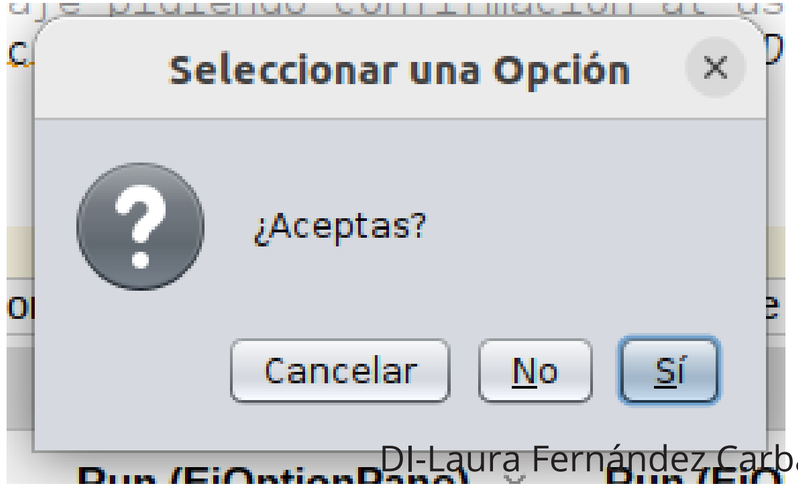
```
//mensaje para obtener una entrada del usuario
String input = JOptionPane.showInputDialog(null, "Introduzca su nombre");
System.out.println(input);
```



- Solicitar confirmación

```
//mensaje pidiendo confirmación al usuario
int opcion = JOptionPane.showConfirmDialog(null, "¿Aceptas?");

if (opcion == JOptionPane.YES_OPTION) {
    System.out.println("Seleccionaste Sí.");
} else if (opcion == JOptionPane.NO_OPTION) {
    System.out.println("Seleccionaste No.");
} else if (opcion == JOptionPane.CANCEL_OPTION) {
    System.out.println("Seleccionaste Cancelar.");
}
```



JOptionPane

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JOptionPane

- Mostrar un mensaje especificando el tipo de mensaje.
  - ERROR_MESSAGE
  - INFORMATION_MESSAGE
  - WARNING_MESSAGE
  - QUESTION_MESSAGE
  - PLAIN_MESSAGE


```
JOptionPane.showMessageDialog(null, "Ha ocurrido un error.", "Error", JOptionPane.ERROR_MESSAGE);
```

JMenuBar

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JMenuBar

Se utiliza para agregar una barra de menú en una aplicación gráfica. La barra de menú normalmente contiene múltiples elementos de menú (JMenu), y cada uno de estos puede tener varias opciones (JMenuItem)

Constructors

Constructor and Description
`JMenuBar()` Creates a new menu bar.

JMenuBar

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JMenuBar

Métodos

add(JMenu menu): Agrega un menú (JMenu) a la barra de menús.

getMenu(int index): Devuelve el menú en la posición indicada por index.

getMenuCount(): Devuelve la cantidad de menús en la barra de menús.

remove(JMenu menu): Elimina un menú específico de la barra de menús.

remove(int index): Elimina el menú en el índice especificado.

removeAll(): Elimina todos los menús de la barra de menús.

setBackground(Color color): Cambia el color de fondo de la barra de menús.

JMenu

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.AbstractButton
          javax.swing.JMenuItem
            javax.swing.JMenu

Constructors

Constructor and Description
`JMenu()` Constructs a new JMenu with no text.
`JMenu(Action a)` Constructs a menu whose properties are taken from the Action supplied.
`JMenu(String s)` Constructs a new JMenu with the supplied string as its text.
`JMenu(String s, boolean b)` Constructs a new JMenu with the supplied string as its text and specified as a tear-off menu or not.

JMenu

java.lang.Object
java.awt.Component
java.awt.Container
javax.swing.JComponent
javax.swing.AbstractButton
javax.swing.JMenuItem
javax.swing.JMenu

Métodos

add(JMenuItem menuItem): Agrega un elemento de menú (JMenuItem) al menú.

add(String itemName): Crea y agrega un JMenuItem con el texto especificado.

addSeparator(): Agrega un separador visual entre los elementos del menú.

add(JMenu menu): Agrega un submenú dentro del menú actual.

remove(JMenuItem item): Elimina el JMenuItem especificado del menú.

remove(int index): Elimina el elemento en la posición indicada.

removeAll(): Elimina todos los elementos del menú.

JMenu

java.lang.Object
java.awt.Component
java.awt.Container
javax.swing.JComponent
javax.swing.AbstractButton
javax.swing.JMenuItem
javax.swing.JMenu

Métodos

isSelected(): devuelve true si el menú está seleccionado.

getItemCount(): devuelve el número de elementos en el menú.

JMenuItem

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.AbstractButton
          javax.swing.JMenuItem

Se utiliza para crear un menú en una barra de menús (JMenuBar) o en un menú emergente (JPopupMenu). Actúa como un contenedor para elementos de menú individuales (JMenuItem), y también puede contener submenús (JMenu anidados).

JMenuItem

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.AbstractButton
          javax.swing.JMenuItem

Métodos heredados de AbstractButton

addActionListener, addChangeListener, addImpl, addItemListener, checkHorizontalKey, checkVerticalKey, createActionListener, createActionPropertyChangeListener, createChangeListener, createItemListener, doClick, doClick, fireActionPerformed, fireItemStateChanged, fireStateChanged, getAction, getActionCommand, getActionListeners, getChangeListeners, getDisabledIcon, getDisabledSelectedIcon, getDisplayedMnemonicIndex, getHideActionText, getHorizontalAlignment, getHorizontalTextPosition, getIcon, getIconTextGap, getItemListeners, getLabel, getMargin, getMnemonic, getModel, getMultiClickThreshhold, getPressedIcon, getRolloverIcon, getRolloverSelectedIcon, getSelectedIcon, getSelectedObjects, getText, getUI, getVerticalAlignment, getVerticalTextPosition, imageUpdate, isBorderPainted, isContentAreaFilled, isFocusPainted, isRolloverEnabled, isSelected, paintBorder, removeActionListener, removeChangeListener, removeItemListener, removeNotify, setAction, setActionCommand, setBorderPainted, setContentAreaFilled, setDisabledIcon, setDisabledSelectedIcon, setDisplayedMnemonicIndex, setFocusPainted, setHideActionText, setHorizontalAlignment, setHorizontalTextPosition, setIcon, setIconTextGap, setLabel, setLayout, setMargin, setMnemonic, setMnemonic, setMultiClickThreshhold, setPressedIcon, setRolloverEnabled, setRolloverIcon, setRolloverSelectedIcon, setSelected, setSelectedIcon, setText, setUI, setVerticalAlignment, setVerticalTextPosition

JMenuItem

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.AbstractButton
          javax.swing.JMenuItem

Métodos heredados de JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, hide, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingOrigin, isPaintingTile, isRequestFocusEnabled, isValidateRoot, paint, paintChildren, paintComponent, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren, printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent, processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

JMenuItem

java.lang.Object
java.awt.Component
java.awt.Container
javax.swing.JComponent
javax.swing.AbstractButton
javax.swing.JMenuItem

Métodos heredados

setText(String text), getText(), setIcon(Icon icon), getIcon(), setEnabled(boolean enabled), isEnabled(), setForeground(Color color), setBackground(Color color), setFont(Font font)

Otros métodos

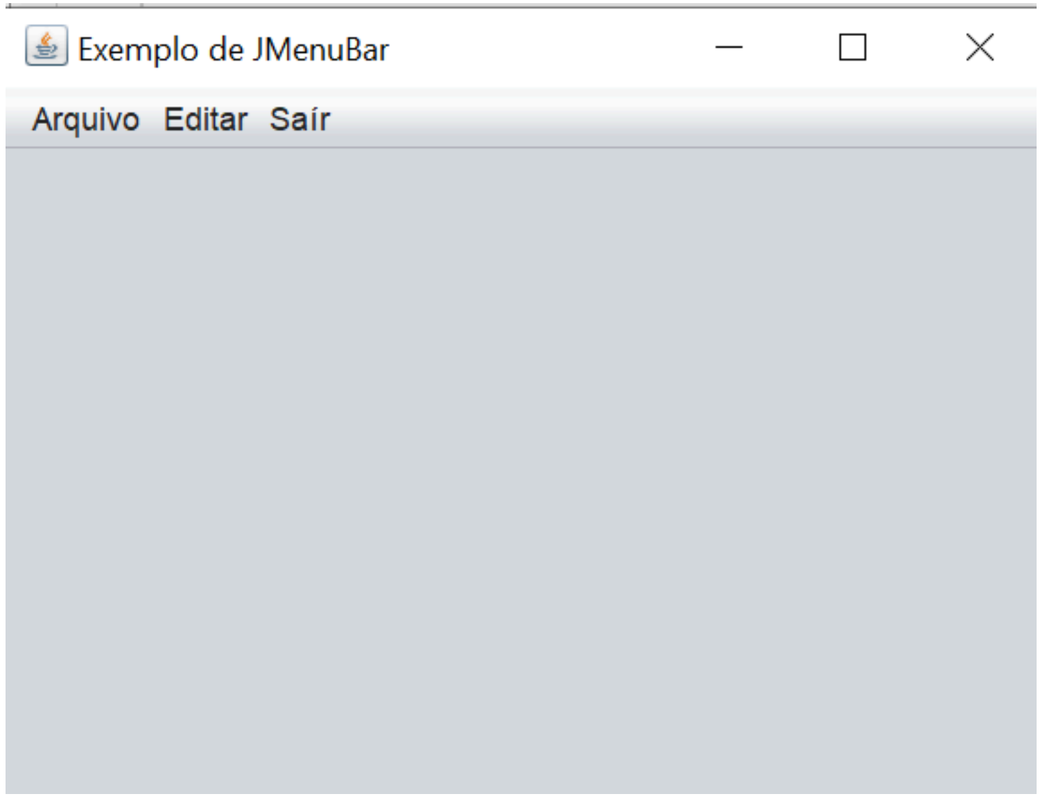
doClick(): Simula un clic en el JMenuItem, útil para automatizar acciones sin interacción del usuario.

getParent(): Obtiene el contenedor padre del JMenuItem, que suele ser el JMenu en el que está contenido.

Exemplo de JMenuBar, JMenu e JMenuItem

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.AbstractButton
          javax.swing.JMenuItem

```java
public EjMenuBar() {
    // Crear el JFrame
    JFrame frame = new JFrame("Exemplo de JMenuBar");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 300);
    //Creamos a barra de menús
    JMenuBar menuBar = new JMenuBar();
    // Crear os menús
    JMenu menuArquivo = new JMenu("Arquivo");
    JMenu menuEditar = new JMenu("Editar");
    JMenu menuSair = new JMenu("Saír");
    // Creamos os elementos para o menú de Arquivo
    JMenuItem itemNuevo = new JMenuItem("Novo Proxecto");
    JMenuItem itemAbrir = new JMenuItem("Novo arquivo");
    JMenuItem itemSalir = new JMenuItem("Abrir proxecto");
   //Engadimos elementos ao menú de arquivo
    menuArquivo.add(itemNuevo);
    menuArquivo.add(itemAbrir);
    menuArquivo.addSeparator(); // Separador visual
    menuArquivo.add(itemSalir);
    //Creamos os items para o menú de Editar
    JMenuItem itemCortar = new JMenuItem("Cortar");
    JMenuItem itemCopiar = new JMenuItem("Copiar");
    JMenuItem itemPegar = new JMenuItem("Pegar");
    //Engadimos os elementos o menú de Editar
    menuEditar.add(itemCortar);
    menuEditar.add(itemCopiar);
    menuEditar.add(itemPegar);
    //Engadimos os menus o menubar
    menuBar.add(menuArquivo);
    menuBar.add(menuEditar);
    menuBar.add(menuSair);
    // Establecer la barra de menús en el JFrame
    frame.setJMenuBar(menuBar);
    //Facemos visible o formulario
    frame.setVisible(true);
}
```



DI-Laura Fernández Carballo