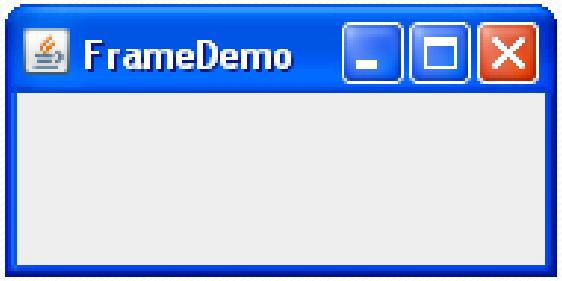
Interfaces gráficas de usuario

Swing

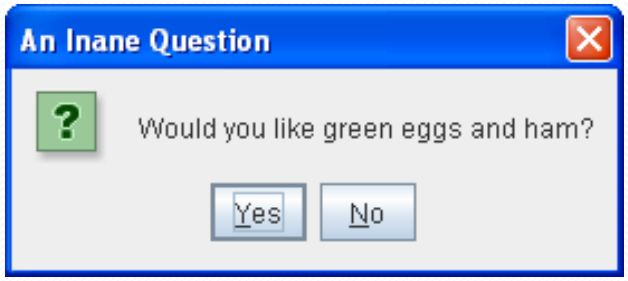
- Para crear interfaces gráficas de usuario se utiliza la librería Swing localizada en javax.swing.
- Una aplicación Swing consta de un contenedor de alto nivel y componentes que responden a eventos generados por el usuario.
- Contenedores de alto nivel: ventanas (frames) y diálogos.
- Componentes: botones, campos de texto, menús, botones de radio, etc.

Contenedores de alto nivel

JFrame



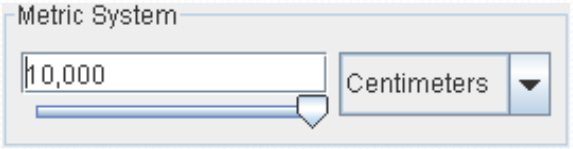
JDialog



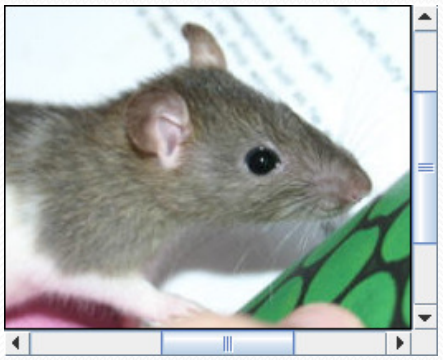
- En una aplicación tiene que haber al menos un contenedor de alto nivel.

Contenedores de propósito general

JPanel



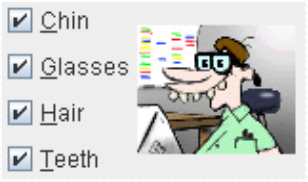
JScrollPane



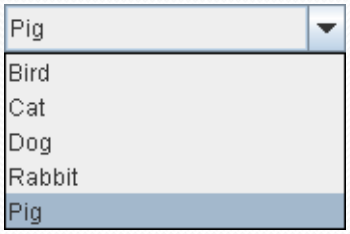
Controles básicos

JButton

JCheckBox

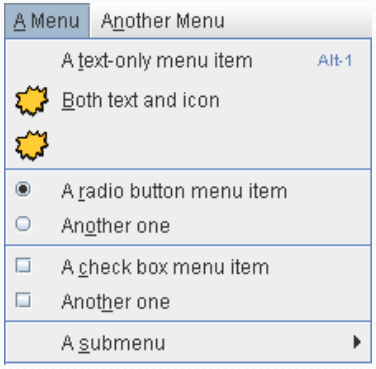


JComboBox

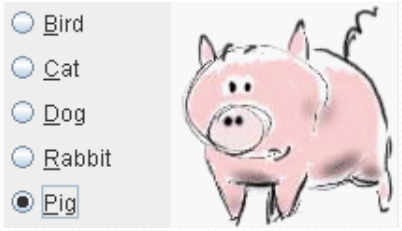


JList

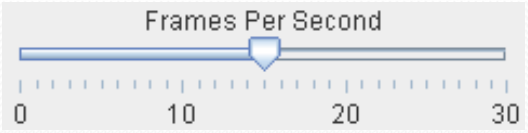
JMenu



JRadioButton



JSlider



Controles básicos

JSpinner

JTextField

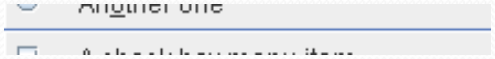
JPasswordField

Desplegar información no editable

JLabel

JProgressBar

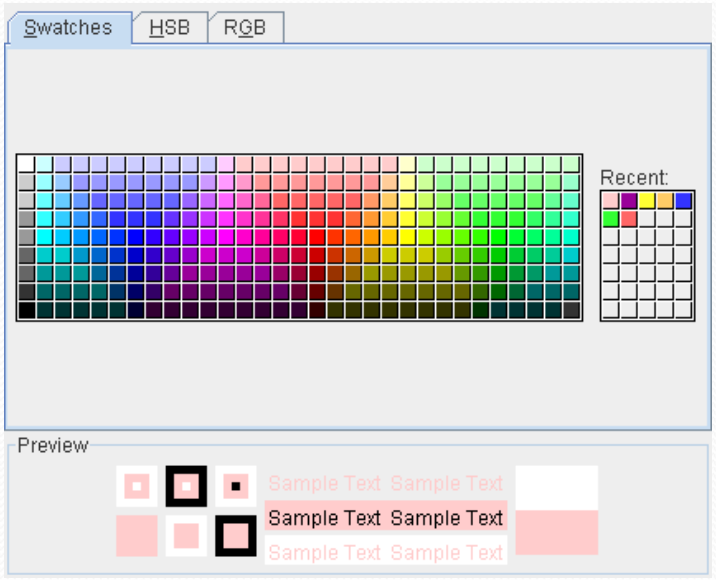
JSeparator



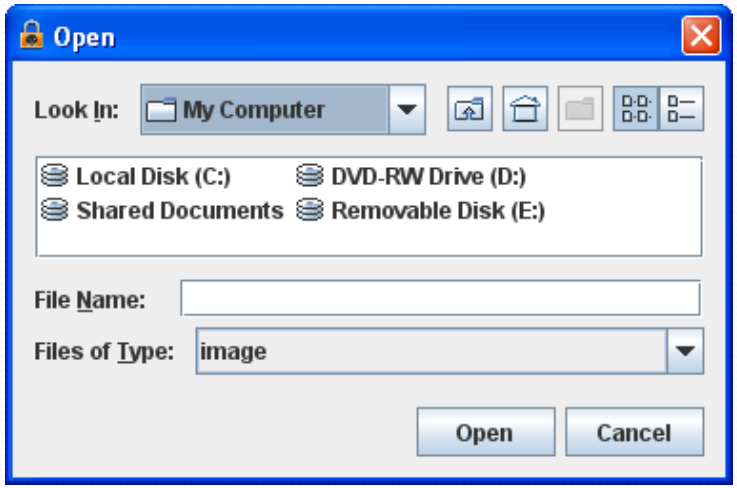
JToolTip

Desplegar información interactiva

JColorChooser

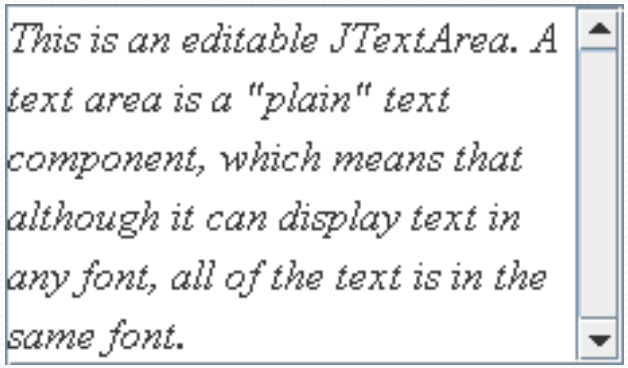


JFileChooser



Desplegar información interactiva

JTextArea



A Visual Guide to Swing Components

- http://www.mat.uson.mx/~havillam/pa/Common/VGSC/index.html

Listeners

- **Listener**. Objeto que responde a un evento.
- ActionListener: responde al clic en un botón o en un ítem de menú, el enter en un campo de texto.
- AdjustmentListener: responde al deslizamiento de un scrollbar.
- ComponentListener: responde al movimiento, cambio de tamaño y aparición/ocultamiento de un componente.
- FocusListener: responde cuando un componente obtiene o pierde el foco del teclado.
- ItemListener: responde a selecciones en una lista, checkbox, etc.

Listeners

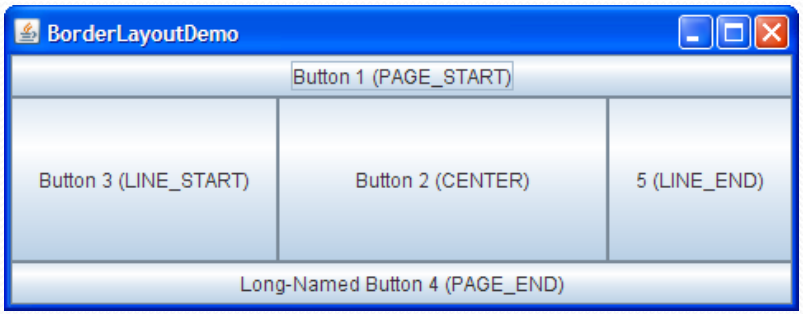
- KeyListener: responde a eventos del teclado (p. e. tecla presionada/soltada/escrita).
- MouseListener: responde a los eventos del ratón (p. e. entrar/salir/hacer clic).
- MouseMotionListener: responde al movimiento del ratón (p.e. arrastre).
- TextListener: responde a cambios de texto en campos/áreas de texto.
- WindowListener: responde a eventos de ventana (p. e. abrir/cerrar).

Layouts

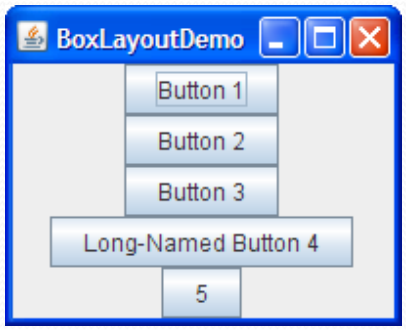
- **Layout**. Objeto que determina la forma en que los componentes se distribuyen en un contenedor.
- Cada contenedor (JFrame, JDialog, JPanel, etc.) puede tener su propio layout.
- Solo puede haber un layout activo a la vez por contenedor.

Layouts

BorderLayout

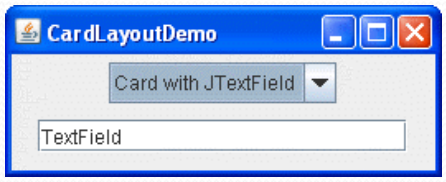


BoxLayout

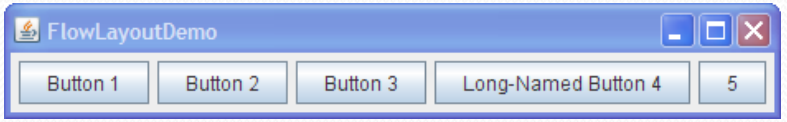


Layouts

CardLayout

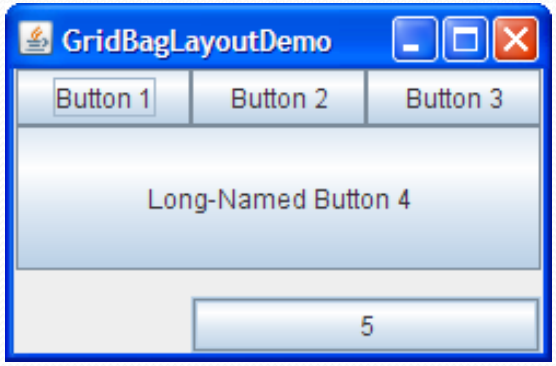


FlowLayout

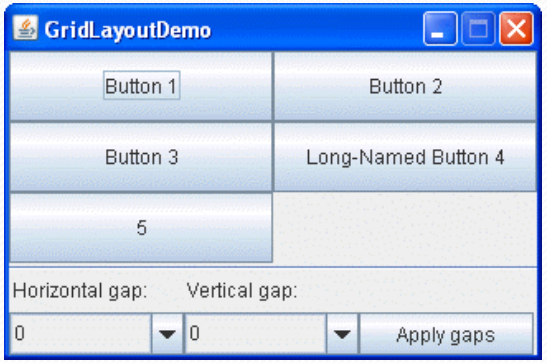


Layouts

GridBagLayout

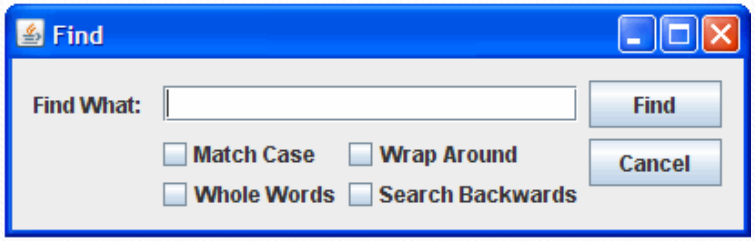


GridLayout

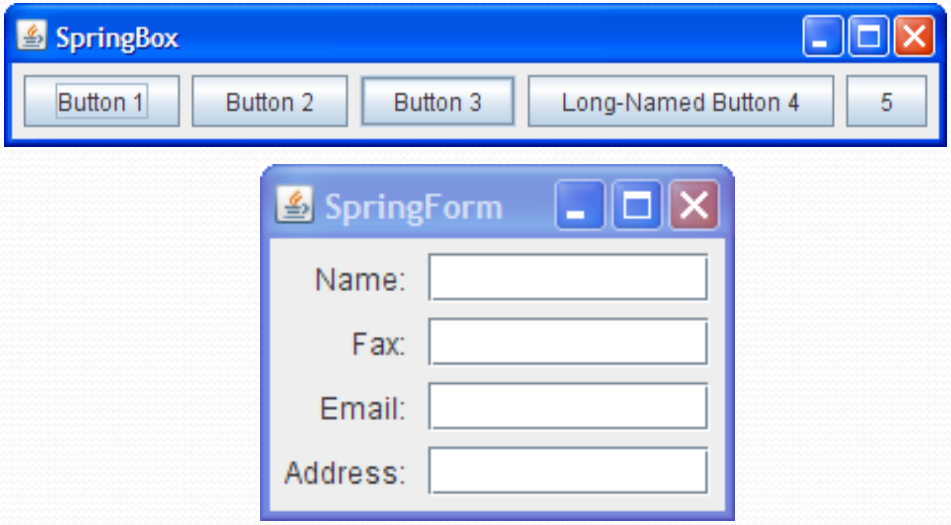


Layouts

GroupLayout



SpringLayout

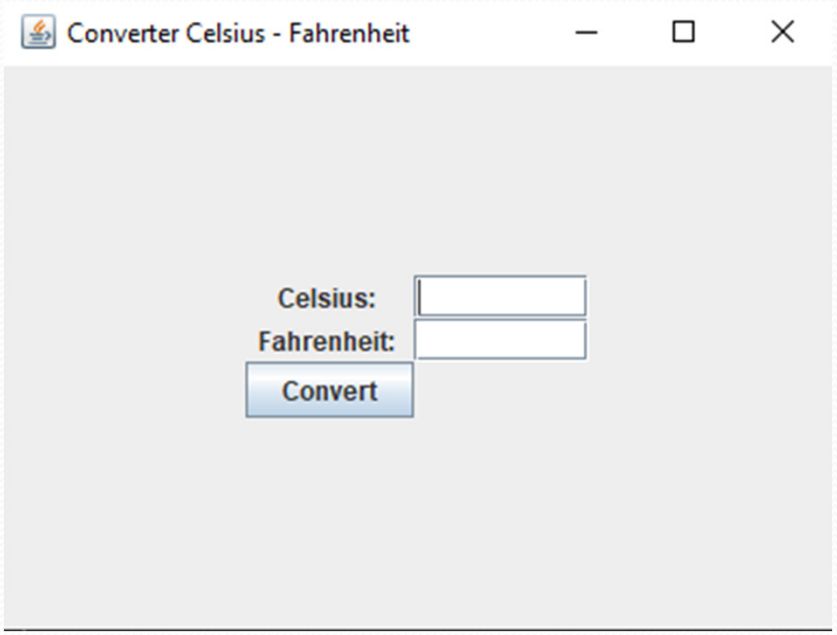


A Visual Guide to Layout Managers

- https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

Ejemplo

- Hacer un programa que convierta entre grados Celsius y grados Fahrenheit.



Ejemplo

- Clases que se utilizan: JFrame, JTextField, JLabel, JButton.
- Layout: GridBagLayout.
- Listeners: ActionEvent.

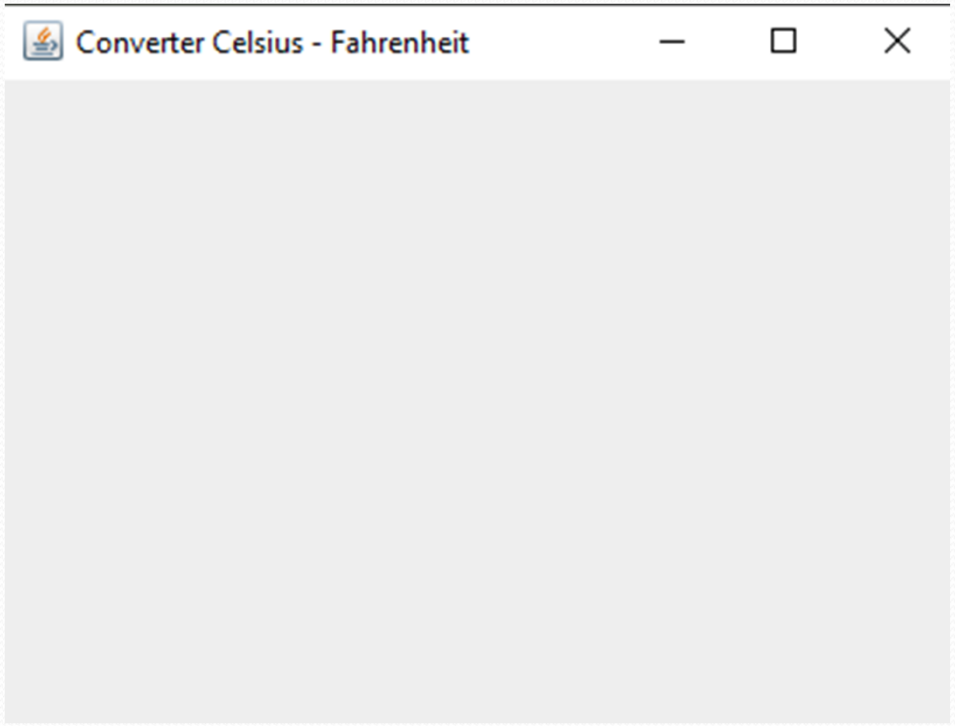
Ventana principal inicial

```
public class MainWindow extends JFrame {
    public MainWindow()
    {
        super("Converter Celsius – Fahrenheit");
        this.setLayout(new GridBagLayout());
        this.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        this.setSize(400, 300);
        this.setLocation(200, 20);
    }
}
```

Clase principal

```
public class Main {
    public static void main(String[] args)
    {
        JFrame window = new MainWindow();
        window.setVisible(true);
    }
}
```

Resultado



Agregar componentes a la ventana

- En la clase MainWindow, declarar los siguientes campos:

```
private JTextField tfCelsius, tfFahrenheit;
DecimalFormat decimalFormat = new DecimalFormat("###,###.00");
```

- En el constructor agregar la siguiente llamada:

```
addComponents();
```

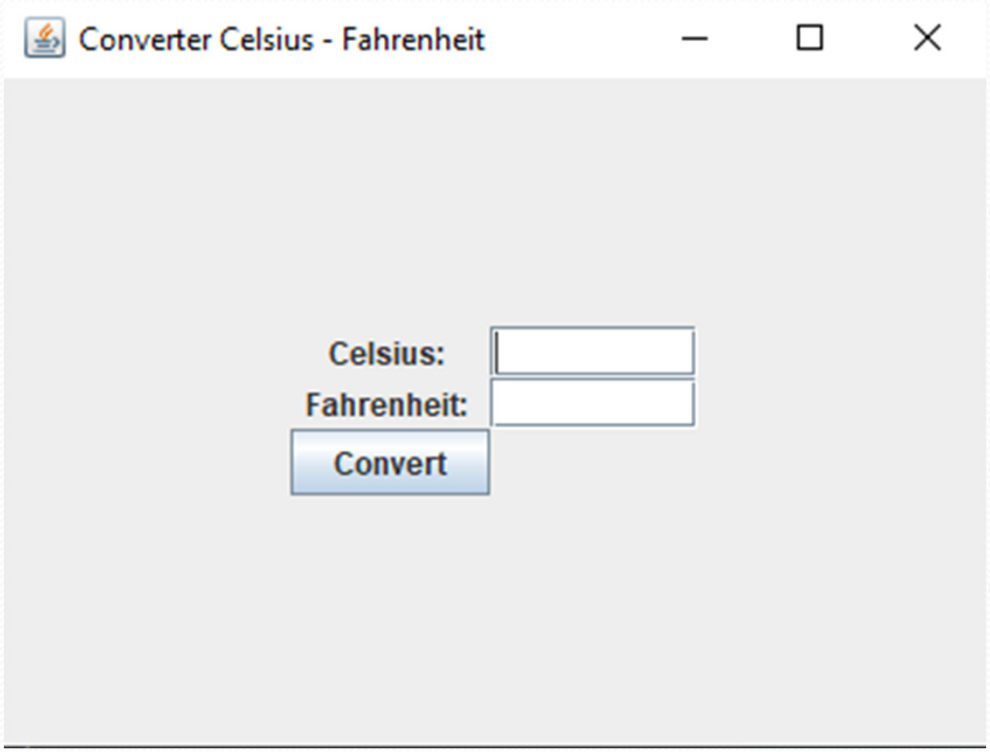
Método addComponents

```
private void addComponents() {
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.gridx = 0; gbc.gridy = 0;
    this.add(new JLabel("Celsius: "), gbc);

    gbc.gridx = 1; gbc.gridy = 0;
    tfCelsius = new JTextField(7);
    this.add(tfCelsius, gbc);

    gbc.gridx = 0; gbc.gridy = 1;
    this.add(new JLabel("Fahrenheit: "), gbc);
```

Método addComponents

```
    gbc.gridx = 1; gbc.gridy = 1;
    tfFahrenheit = new JTextField(7);
    this.add(tfFahrenheit, gbc);

    gbc.gridx = 0; gbc.gridy = 2;
    JButton btnConvert = new JButton("Convert");
    this.add(btnConvert, gbc);
}
```

Resultado


Converter Celsius - Fahrenheit
Celsius:
Fahrenheit:
Convert

Mejorando la interface

- La interface se puede mejorar usando las variables del objeto GridBagConstraints.
- El botón se puede centrar indicando que ocupe dos columnas.

```
gbc.gridwidth = 2;
```

- A los componentes se les puede dar pesos para ajustarlos en la ventana usando weightx, weighty y anchor.

Método addComponents

```
private void addComponents() {
    GridBagConstraints gbc = new GridBagConstraints();

    gbc.gridx = 0; gbc.gridy = 0;
    gbc.weightx = 20; gbc.weighty = 15;
    gbc.anchor = GridBagConstraints.EAST;
    this.add(new JLabel("Celsius: "), gbc);

    gbc.gridx = 1; gbc.gridy = 0;
    gbc.weightx = 80; gbc.weighty = 0;
    gbc.anchor = GridBagConstraints.WEST;
    tfCelsius = new JTextField(7);
    this.add(tfCelsius, gbc);
```

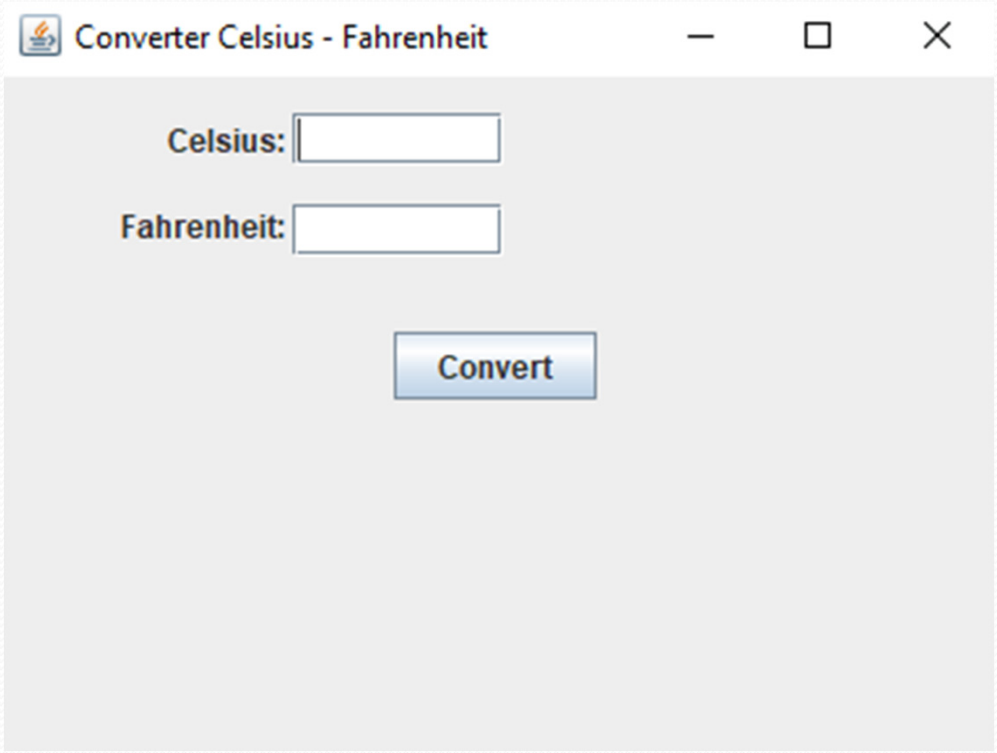
Método addComponents

```
gbc.gridx = 0; gbc.gridy = 1;
gbc.weightx = 0; gbc.weighty = 15;
gbc.anchor = GridBagConstraints.NORTHEAST;
this.add(new JLabel("Fahrenheit: "), gbc);

gbc.gridx = 1; gbc.gridy = 1;
gbc.weightx = 0; gbc.weighty = 0;
gbc.anchor = GridBagConstraints.NORTHWEST;
tfFahrenheit = new JTextField(7);
this.add(tfFahrenheit, gbc);
```

Método addComponents

```
    gbc.gridx = 0; gbc.gridy = 2;
    gbc.gridwidth = 2;
    gbc.weightx = 0; gbc.weighty = 70;
    gbc.anchor = GridBagConstraints.NORTH;
    JButton btnConvert = new JButton("Convert");
    this.add(btnConvert, gbc);
}
```

Resultado


Converter Celsius - Fahrenheit
Celsius:
Fahrenheit:
Convert

Agregar eventos

- El programa tiene el siguiente comportamiento.
- Al oprimir el botón, se revisa el campo de texto Celsius.
- Si tiene un número, se convierte a grados Fahrenheit y se escribe en el campo correspondiente.
- Si está vacío, se lee el campo de grados Fahrenheit, se convierte a grados Celsius y se escribe en el campo correspondiente.
- El primer paso es agregar un listener al botón de convertir.

```
btnConvert.addActionListener(e -> convert());
```

Método convert

```
private void convert()
{
    if (!tfCelsius.getText().trim().isEmpty()) {
        celsius2Fahrenheit(tfCelsius.getText());
    }
    else {
        fahrenheit2Celsius(tfFahrenheit.getText().trim());
    }
}
```

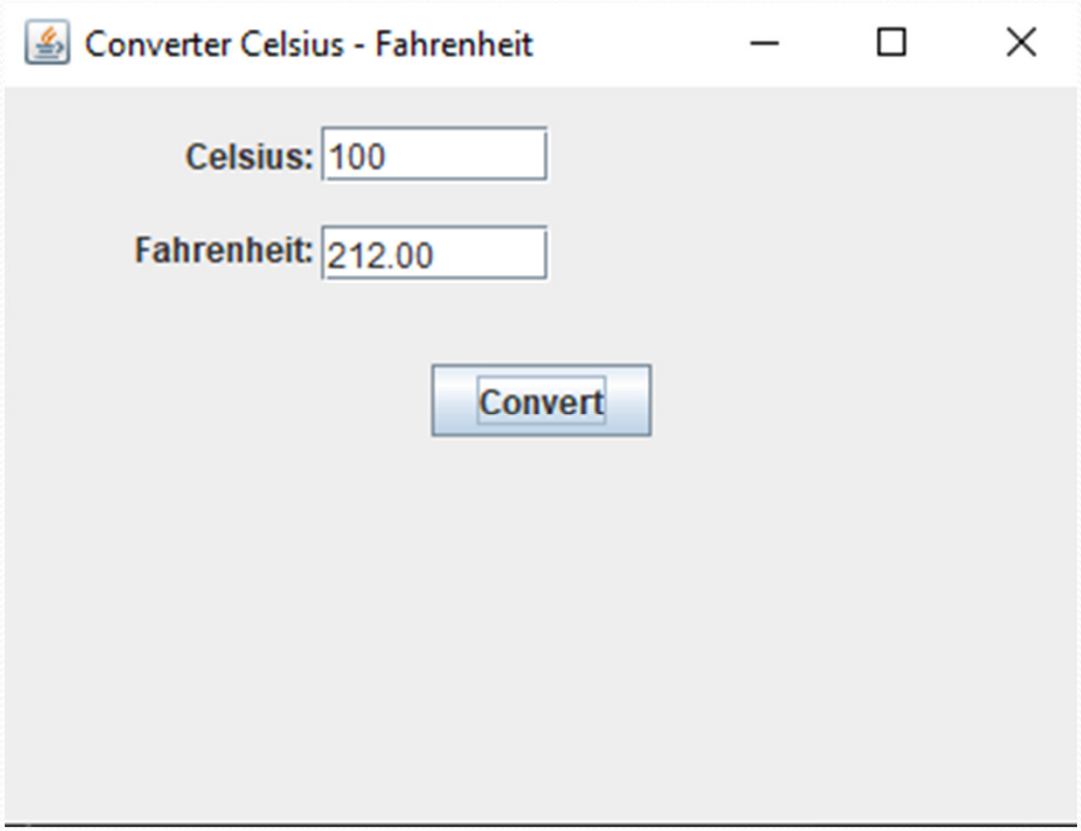
Método celsius2Fahrenheit

```
private void celsius2Fahrenheit(String strCelsius) {
    double celsius = 0;
    try {
        celsius = Double.parseDouble(strCelsius);
    }
    catch (NumberFormatException e) {
        tfCelsius.setText("0");
        JOptionPane.showMessageDialog(null, "Not numeric Celsius", "Error",
                JOptionPane.ERROR_MESSAGE);
    }
    double fahrenheit = celsius * (9.0 / 5) + 32;
    tfFahrenheit.setText(decimalFormat.format(fahrenheit));
}
```

Método fahrenheit2Celsius

```
private void fahrenheit2Celsius(String strFahrenheit) {
    double fahrenheit = 0;
    try {
        fahrenheit = Double.parseDouble(strFahrenheit);
    }
    catch (NumberFormatException e) {
        tfFahrenheit.setText("0");
        JOptionPane.showMessageDialog(null, "Not numeric Fahrenheit", "Error",
                JOptionPane.ERROR_MESSAGE);
    }
    double celsius = (fahrenheit – 32) * (5.0 / 9);
    tfCelsius.setText(decimalFormat.format(celsius));
}
```

Resultado


Converter Celsius - Fahrenheit
Celsius: 100
Fahrenheit: 212.00
Convert

Agregar eventos

- Se desea que al escribir un número y dar enter en el campo de grados Celsius, se llame al convertidor de Celsius a Fahrenheit.
- La solución es agregar un listener de acción a tfCelsius.

```
tfCelsius.addActionListener(e ->
celsius2Fahrenheit(tfCelsius.getText().trim()));
```

- Se puede hacer lo mismo con tfFahrenheit.

```
tfFahrenheit.addActionListener(e ->
fahrenheit2Celsius(tfFahrenheit.getText().trim()));
```