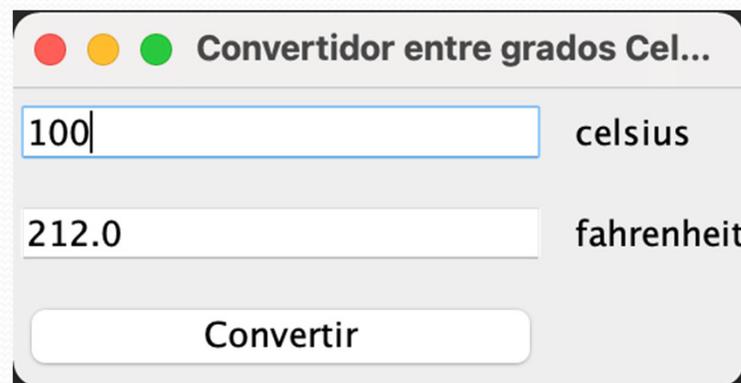


Interfaces gráficas de usuario

Ejemplo con el diseñador de formas de IntelliJ

Creando GUIs en IntelliJ IDEA

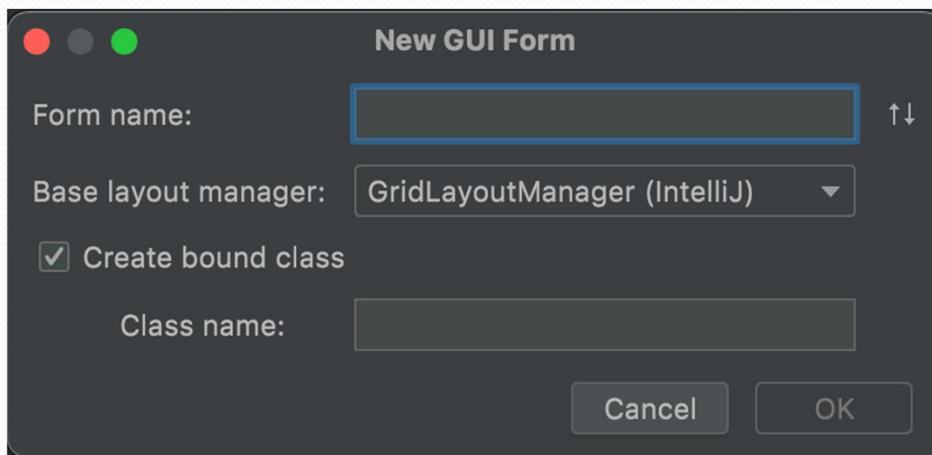
- Cómo ejemplo se muestra la programación de un programa para convertir entre grados Celsius y Fahrenheit.



- Se usará el diseñador de formas.

Formas GUI

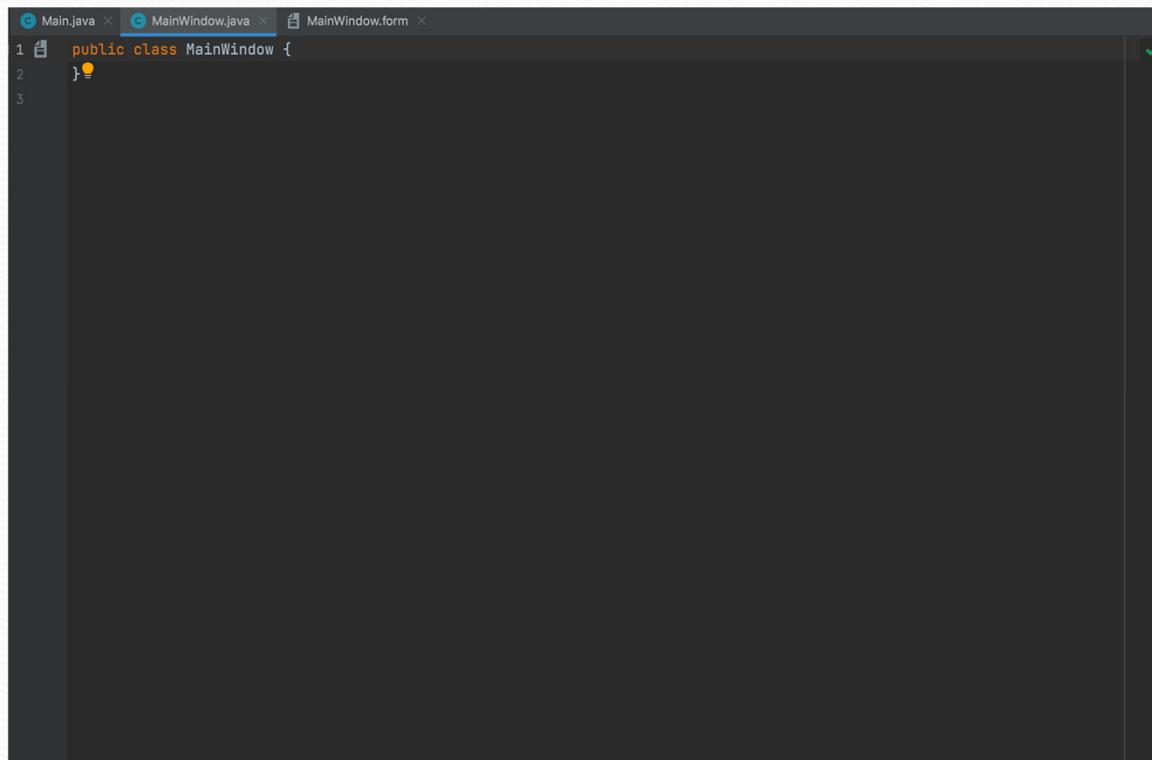
- Seleccionar en el menú la opción File▷New▷Swing UI Designer▷GUI Form.
- Se abre el siguiente diálogo:



Formas GUI

- Se les da nombres a la forma y a la clase (por default es el mismo nombre) y se oprime el botón OK.
- En este proyecto se usa el nombre `MainWindow` para la forma y la clase.
- Se generan dos archivos:
 - `MainWindow.form`: en este archivo se puede programar la interface gráfica de manera visual, arrastrando los componentes Swing que se muestran en la paleta de la derecha.
 - `MainWindow.java`: en este archivo se puede programar la interface gráfica usando instrucciones de Java.

MainWindow.java



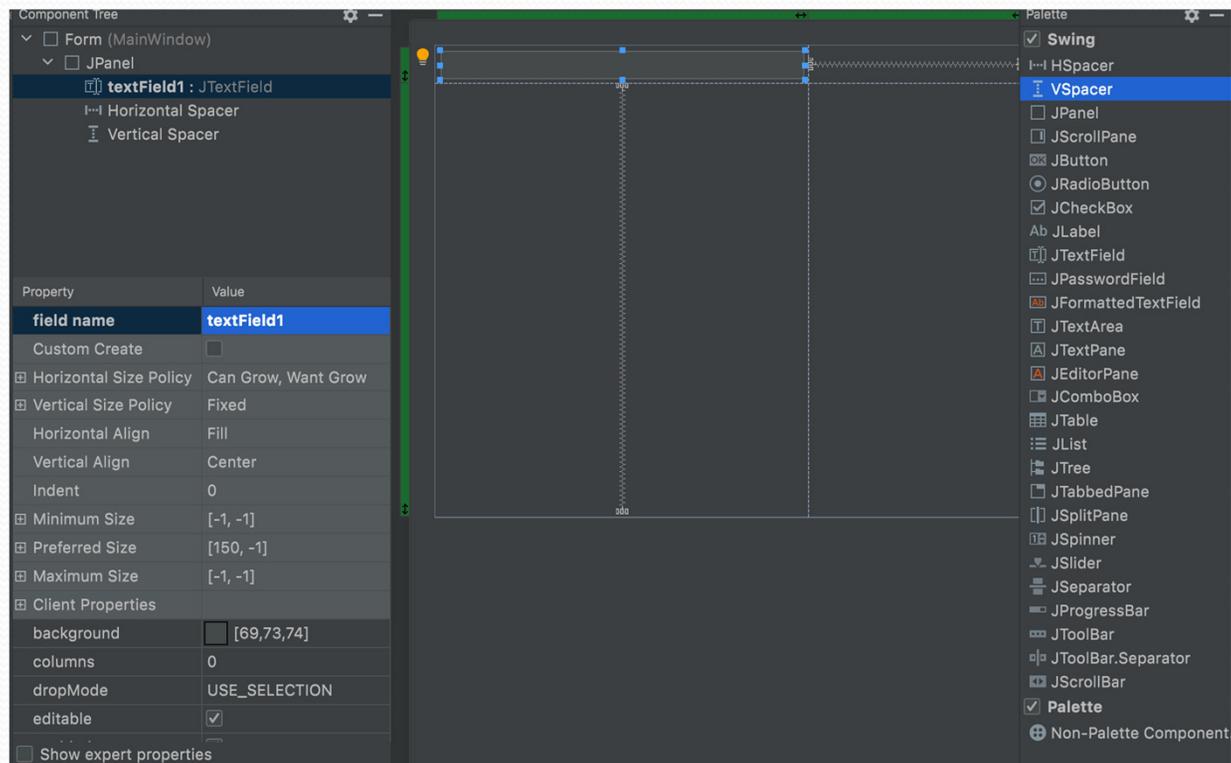
The screenshot shows an IDE window with three tabs: 'Main.java', 'MainWindow.java', and 'MainWindow.form'. The 'MainWindow.java' tab is active and displays the following code:

```
1 public class MainWindow {  
2     }  
3
```

Agregando componentes

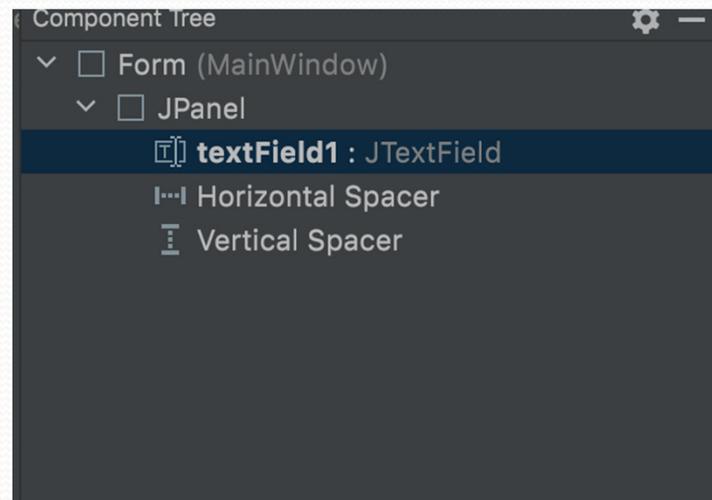
- La interface gráfica consta de los siguientes componentes: dos campos de texto, dos etiquetas y un botón.
- Los componentes se arrastran a `MainWindow.form` utilizando el ratón.
- Después de arrastrar un campo de texto a la parte izquierda superior de la forma el editor se ve así.

Editor



Árbol de componentes

- En la parte superior izquierda está el árbol de componentes.
- Hasta el momento hay una forma, un panel y un campo de texto.



Editor de propiedades

- En la parte inferior izquierda está el editor de propiedades.
- Conviene cambiarle el nombre tanto al panel como al campo de texto.
- Esto con el objetivo de que sea más fácil acceder las variables en el archivo `MainWindow.java`.

| Property | Value |
|---|-------------------------------------|
| field name | textField1 |
| Custom Create | <input type="checkbox"/> |
| ⊕ Horizontal Size Policy | Can Grow, Want Grow |
| ⊕ Vertical Size Policy | Fixed |
| Horizontal Align | Fill |
| Vertical Align | Center |
| Indent | 0 |
| ⊕ Minimum Size | [-1, -1] |
| ⊕ Preferred Size | [150, -1] |
| ⊕ Maximum Size | [-1, -1] |
| ⊕ Client Properties | |
| background | <input type="checkbox"/> [69,73,74] |
| columns | 0 |
| dropMode | USE_SELECTION |
| editable | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> Show expert properties | |

Editor de propiedades

- Los cambios se reflejan en el editor de propiedades, en el árbol de componentes y en el archivo `MainWindow.java`.
- Para cambiar una propiedad a un componente, primero se marca el componente en el árbol de componentes y luego se busca la propiedad deseada en el editor de propiedades.

Editor de propiedades

- Notar que se usa la *notación húngara*, en la cual el nombre de la variable se precede con un indicador de su tipo, en este caso “tf” por TextField (campo de texto).

| Property | Value |
|---|-------------------------------------|
| field name | tfCelsius |
| Custom Create | <input type="checkbox"/> |
| ⊕ Horizontal Size Policy | Can Grow, Want Grow |
| ⊕ Vertical Size Policy | Fixed |
| Horizontal Align | Fill |
| Vertical Align | Center |
| Indent | 0 |
| ⊕ Minimum Size | [-1, -1] |
| ⊕ Preferred Size | [150, -1] |
| ⊕ Maximum Size | [-1, -1] |
| ⊕ Client Properties | |
| background | <input type="checkbox"/> [69,73,74] |
| columns | 0 |
| dropMode | USE_SELECTION |
| editable | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> Show expert properties | |

Árbol de componentes

- Los nombres asignados a los campos se ven en el árbol de componentes.
- Los campos también aparecen en en MainWindow.java.



MainWindow.java

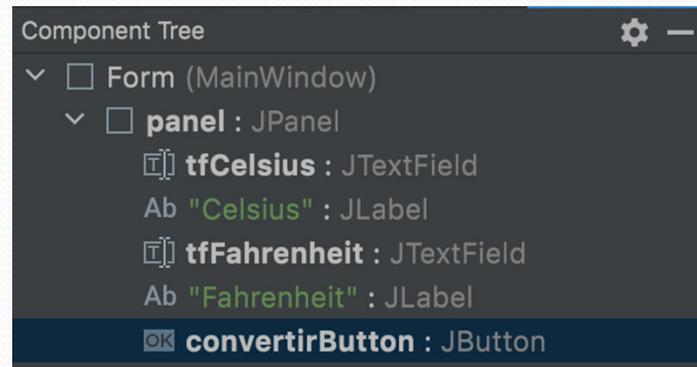
```
1  import javax.swing.*;  
2  
3  public class MainWindow {  
4      private JTextField tfNombre;  
5      private JPanel panel;  
6  }  
7
```

Resto de la interface

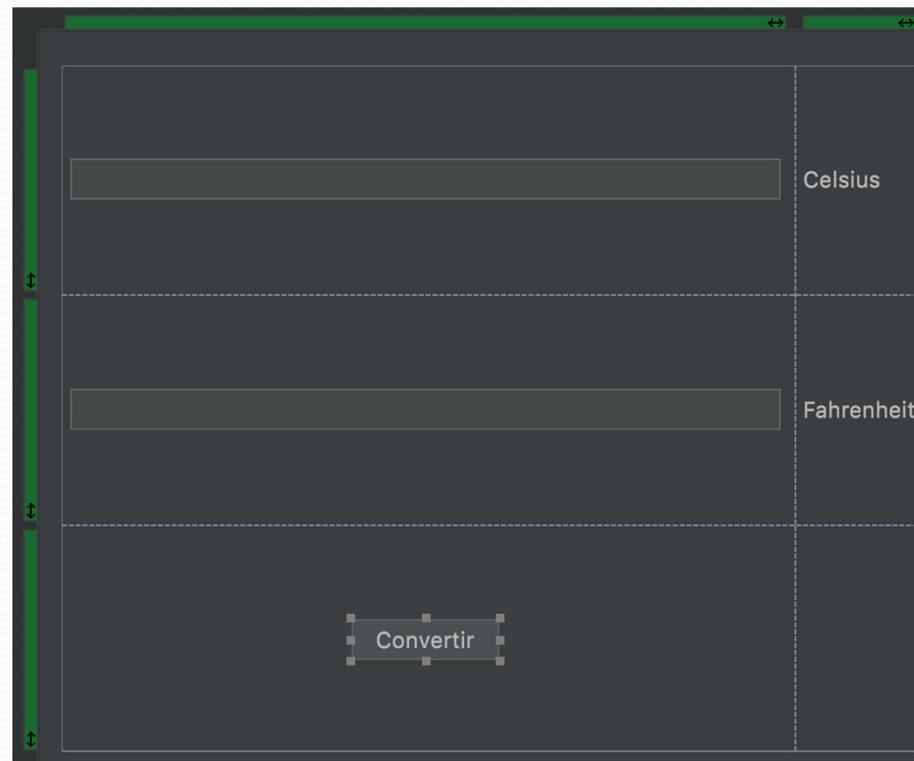
- De la misma forma se agrega el otro campo de texto, las dos etiquetas y el botón.
- En las etiquetas no es necesario dar un nombre al campo.
- Excepto si a una etiqueta se le quisiera cambiar alguna propiedad (por ejemplo el texto) desde el programa.
- Para asignarle un texto a una etiqueta se busca la propiedad `text` en el editor de preferencias.
- Al botón se le cambia sus propiedades `Horizontal Align` a `Center` y `text` a “Convertir”.

Resto de la interface

- Al finalizar, el árbol de componentes se ve así.
- Y la interface se debe ver de esta forma.



Interface



Eventos

- El programa hace las conversiones al generarse un evento.
- Si el usuario escribe un número en el campo de texto de grados Celsius y oprime la tecla de `Enter`, el programa convierte el número a grados Fahrenheit y lo escribe en el segundo campo de texto.
- Si el usuario escribe un número en el campo de texto de grados Fahrenheit y oprime la tecla de `Enter`, el programa convierte el número a grados Celsius y lo escribe en el primer campo de texto.

Eventos

- Si el usuario oprime el botón “Convertir”
 - Si el campo de texto de grados Celsius tiene un número, se convierte el número a grados Fahrenheit y lo escribe en el segundo campo de texto.
 - Si el campo de texto de grados Celsius está vacío y el de grados Fahrenheit tiene un número, se convierte el número a grados Celsius y lo escribe en el primer campo de texto.
 - Si los dos campos de texto están vacíos, se abre un diálogo de error.
- En cualquier caso, si el campo de texto que se lee contiene datos no numéricos se abre una ventana de error y se guarda un cero en el campo de texto.

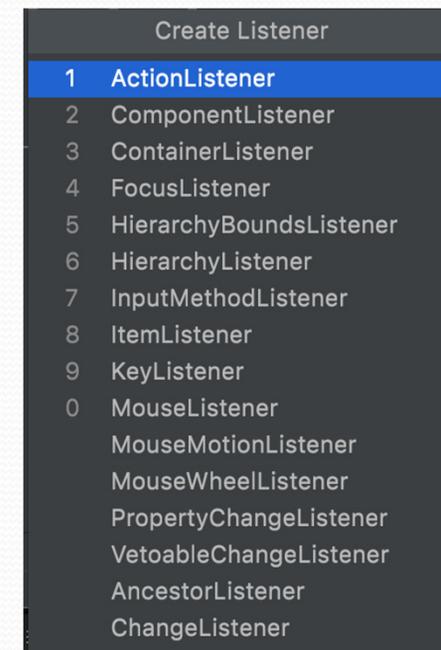


Eventos

- Para este ejemplo, se necesita registrar objetos `ActionListener` en el botón y en los campos de texto.

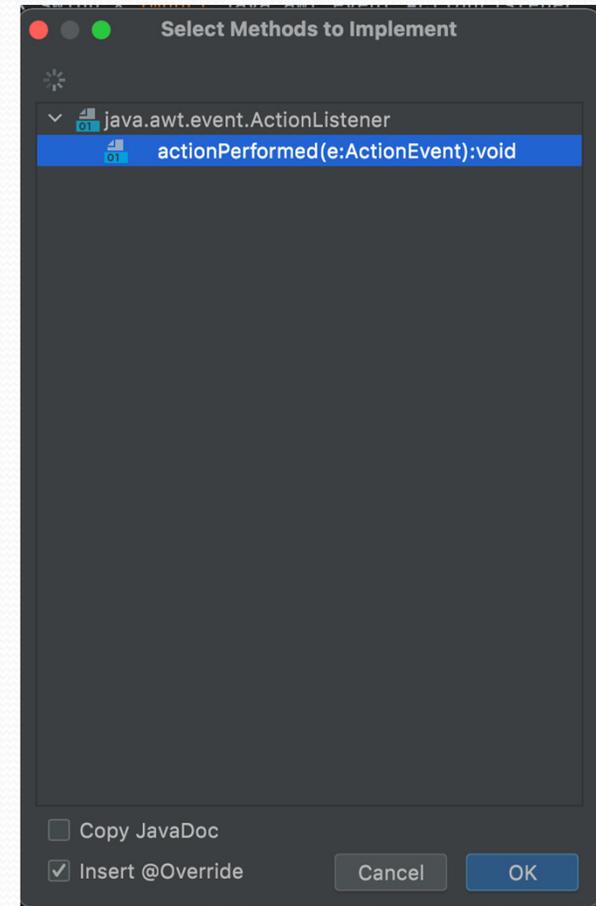
Agregar un listener

- Para agregar un listener al campo de texto Celsius, se marca el campo de texto en el árbol de componentes, se oprime el botón derecho del ratón y se selecciona la opción Create Listener.
- Se abre el menú con las distintas opciones de listeners.
- Se selecciona ActionListener.



Agregar un listener

- Se abre el siguiente menú.
- Se oprime el botón OK.
- En el archivo MainWindows.java se genera el siguiente código.



Evento de acción

```
Main.java x MainWindow.java x MainWindow.form x
1  import javax.swing.*;
2  import java.awt.event.ActionEvent;
3  import java.awt.event.ActionListener;
4
5  public class MainWindow {
6      private JTextField tfCelsius;
7      private JPanel panel;
8      private JTextField tfFahrenheit;
9      private JButton convertirButton;
10
11     public MainWindow()
12     {
13         tfCelsius.addActionListener(new ActionListener() {
14             @Override
15             public void actionPerformed(ActionEvent e)
16             {
17
18             }
19         });
20     }
21 }
22
```

Evento de acción

- Dentro del método `actionPerformed` se introduce el código para leer el campo de texto Celsius, hacer la conversión a grados Fahrenheit y escribir el resultado en el segundo campo de texto.
- Para poder correr el programa y revisar si la conversión de grados Celsius a Fahrenheit es correcta se necesitan los siguientes pasos:
 - Indicar que `MainWindow` extiende a `JFrame`.
 - Configurar el frame (i.e. título, tamaño, localización).
 - Indicar que el contenido del frame es el panel que se diseñó visualmente.
 - Hacer visible el frame.
 - En la clase `Main` crear un objeto de tipo `MainWindow`.

MainWindow.java

```
public class MainWindow extends JFrame {  
    private JTextField tfCelsius;  
    private JPanel panel;  
    private JTextField tfFahrenheit;  
    private JButton convertirButton;  
    NumberFormat formatter = new DecimalFormat(".00");  
  
    public MainWindow()  
    {  
        this.setTitle("Convertidor C – F");  
        this.setSize(300, 200);  
    }  
}
```

MainWindow.java

```
this.setLocation(400, 20);
this.setContentPane(panel);
this.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
this.setVisible(true);
tfCelsius.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        celsiusToFahrenheit();
    }
});
}
```

MainWindow.java

```
private void celsiusToFahrenheit()
{
    String strCelsius = tfCelsius.getText();
    double celsius = 0;
    try {
        celsius = Double.parseDouble(strCelsius);
    }
}
```

MainWindow.java

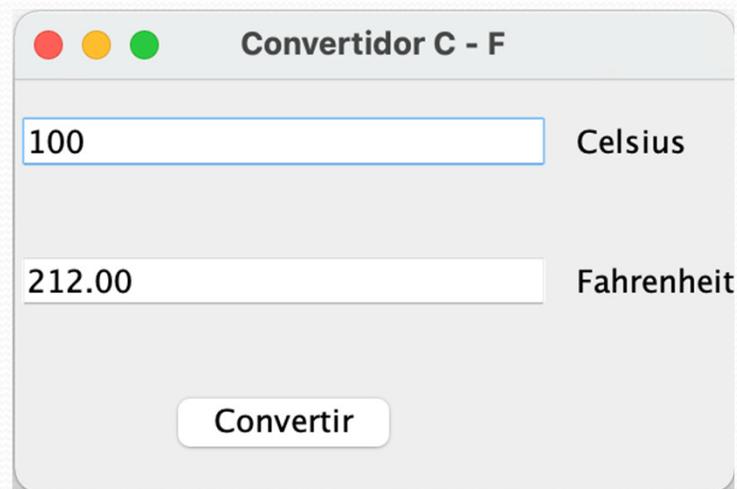
```
catch (Exception x) {  
    JOptionPane.showMessageDialog(null,  
        "El valor Celsius no es numérico", "Error",  
        JOptionPane.ERROR_MESSAGE);  
    tfCelsius.setText("0");  
}  
double fahrenheit = celsius * 1.8 + 32;  
tfFahrenheit.setText(formatter.format(fahrenheit));  
}  
}
```

Main.java

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

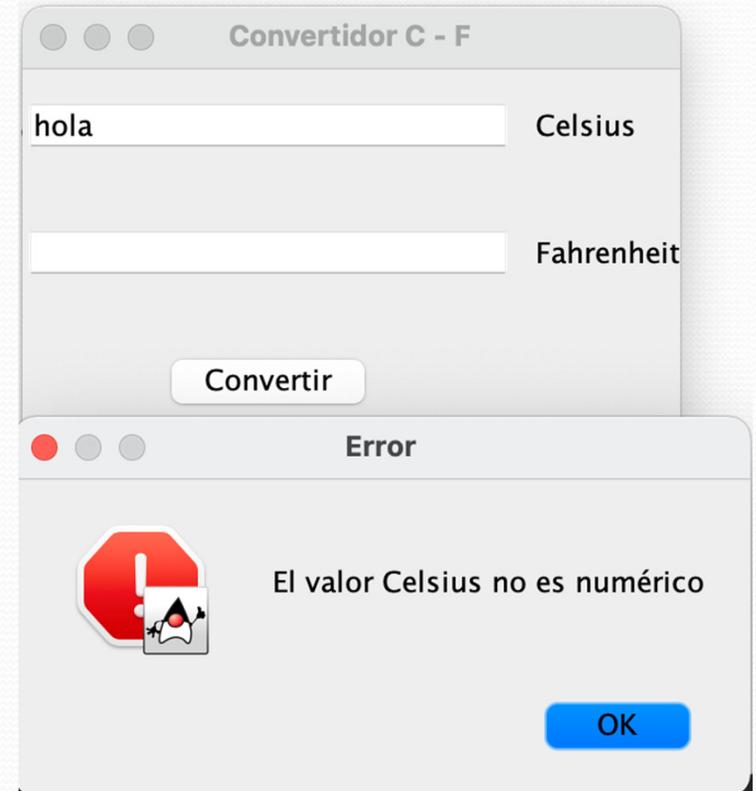
Resultado

- Al correr el programa se abre el frame.
- Se escribe un valor en el primer campo y se oprime Enter.



Resultado

- Un dato no numérico abre el diálogo de error.



Resto del programa

- Del mismo modo que se le agregó el listener al campo de texto Celsius se le agrega un listener al campo de texto Fahrenheit y otro al botón “Convertir”.
- El código que se le agrega a cada listener es el siguiente.
- Notar que todo el código va adentro de la clase MainWindow.

Listener del campo Fahrenheit

```
tfFahrenheit.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e)  
    {  
        fahrenheitToCelsius();  
    }  
});
```

Método fahrenheitToCelsius

```
private void fahrenheitToCelsius()
{
    String strFahrenheit = tfFahrenheit.getText();
    double fahrenheit = 0;
    try {
        fahrenheit = Double.parseDouble(strFahrenheit);
    }
}
```

Método fahrenheitToCelsius

```
catch (Exception x) {  
    JOptionPane.showMessageDialog(null,  
        "El valor Fahrenheit no es numérico",  
        "Error", JOptionPane.ERROR_MESSAGE);  
    tfFahrenheit.setText("0");  
}  
double celsius = (fahrenheit - 32) * (5.0 / 9);  
tfCelsius.setText(formatter.format(celsius));  
}
```

Listener del botón “Convertir”

```
convertirButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e)  
    {  
        String strCelsius = tfCelsius.getText().trim();  
        String strFahrenheit = tfFahrenheit.getText().trim();  
        if (strCelsius.length() != 0) {  
            celsiusToFahrenheit();  
        }  
        else if (strFahrenheit.length() != 0) {  
            fahrenheitToCelsius();  
        }  
    }  
});
```

Listener del botón “Convertir”

```
else {  
    JOptionPane.showMessageDialog(null,  
        “No hay nada que convertir”,  
        “Error”, JOptionPane.ERROR_MESSAGE);  
}  
}  
});  
}
```