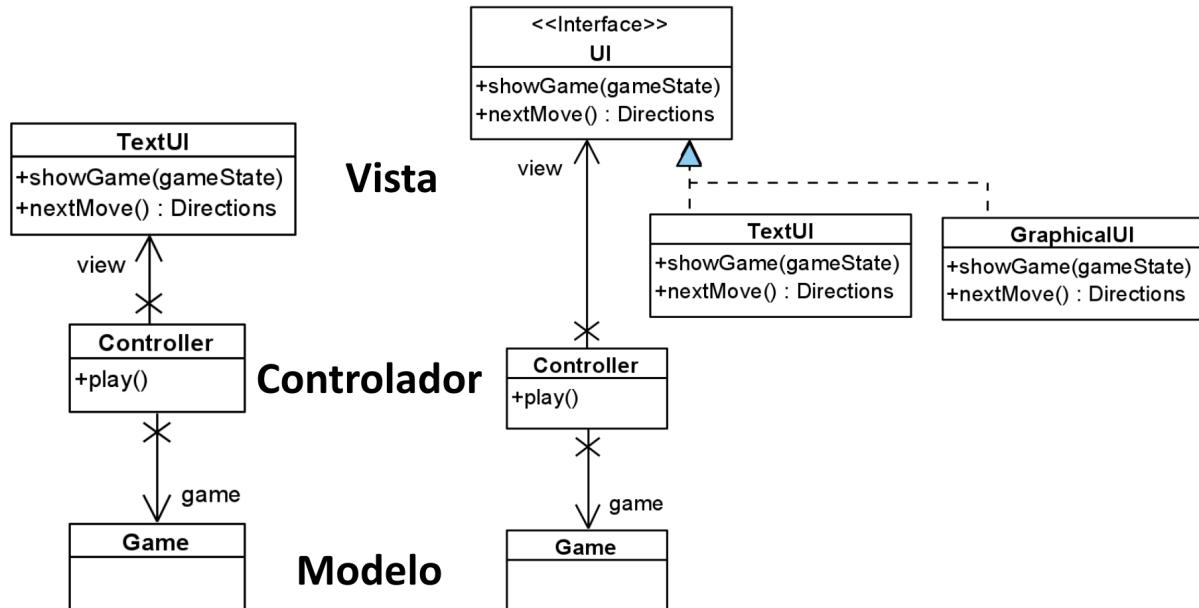


Material ayuda P5



Así que tu resultado en la práctica será tener en el paquete de Vista cuatro archivos:

```

<!-- irrgarten.UI -->
  <-- Cursors.java -->
  <-- GraphicalUI.java -->
  <-- TextUI.java -->
  <-- UI.java -->

```

CLASE UI

donde el código de **UI.java** será:

```

package irrgarten.UI;

import irrgarten.Directions;
import irrgarten.GameState;

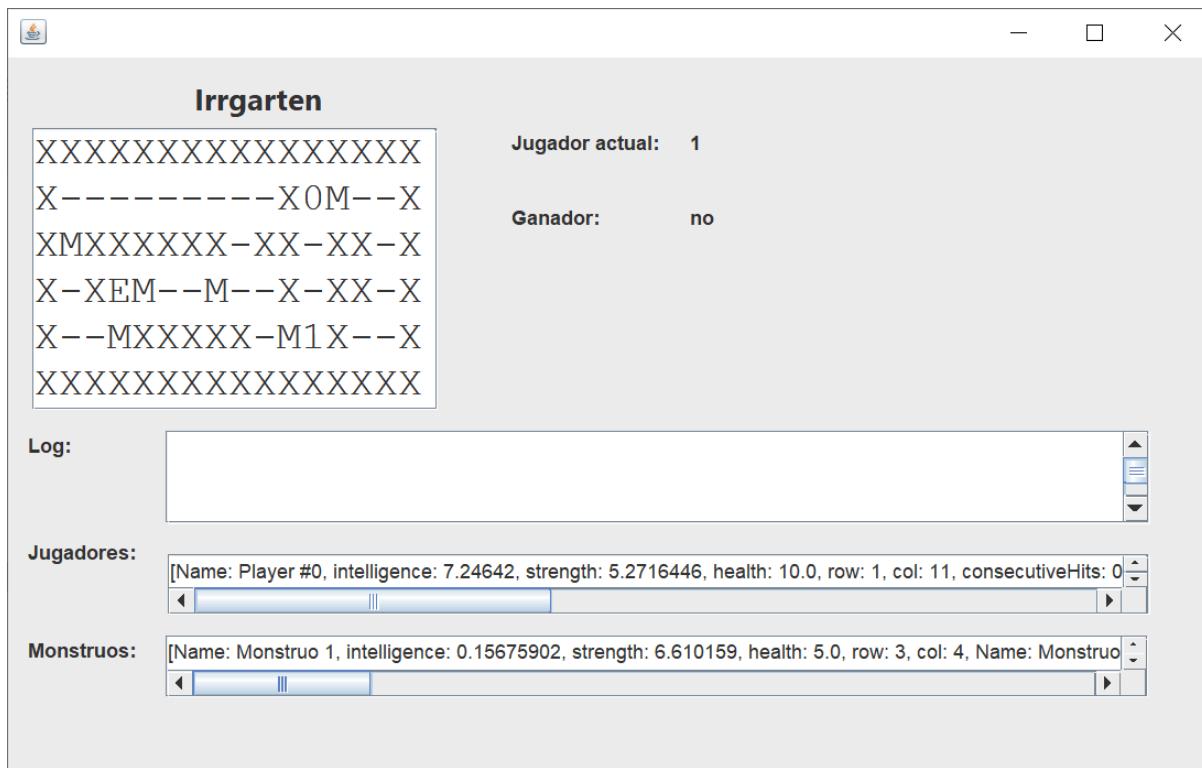
public interface UI {
    Directions nextMove();
    void showGame(GameState gameState);
}

```

Como se hace

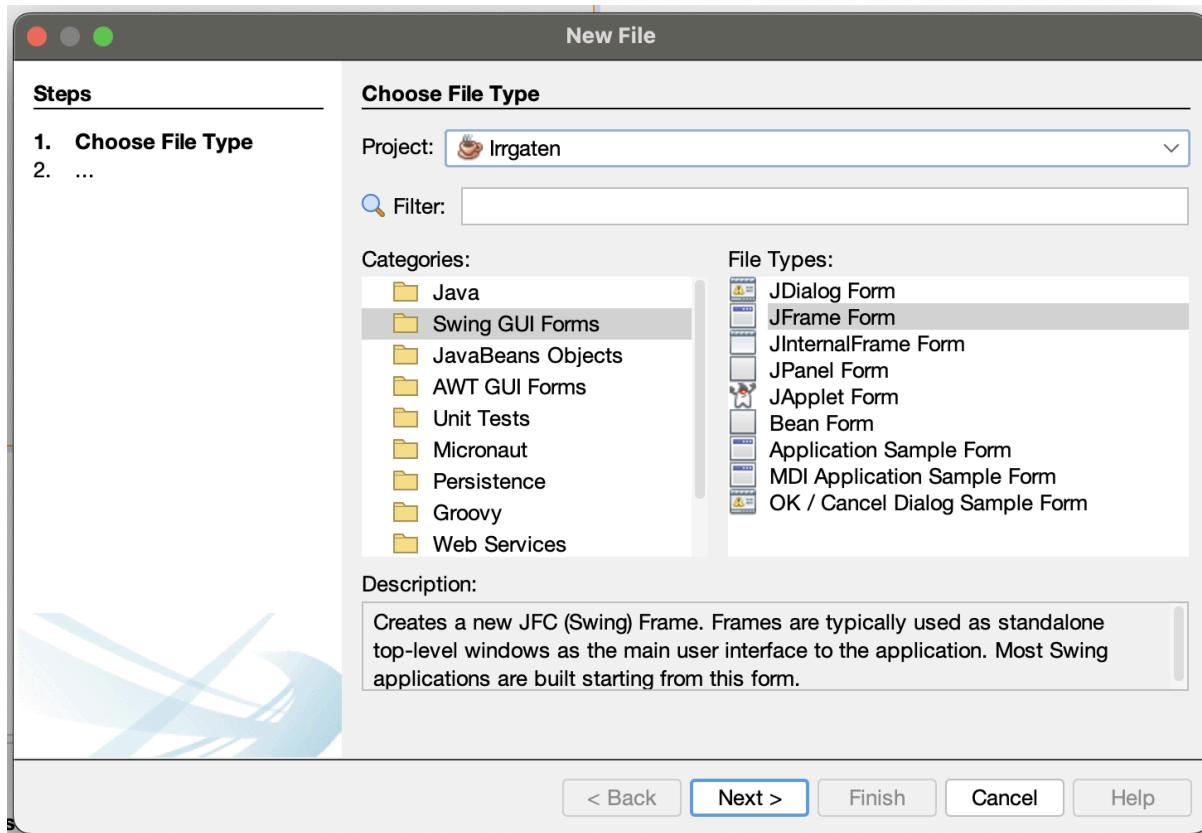
- Comenta la generación de la **TestUI** (en texto, para que esta ya no arranque)

- Creamos una vista gráfica, GraphicalUI como esta:

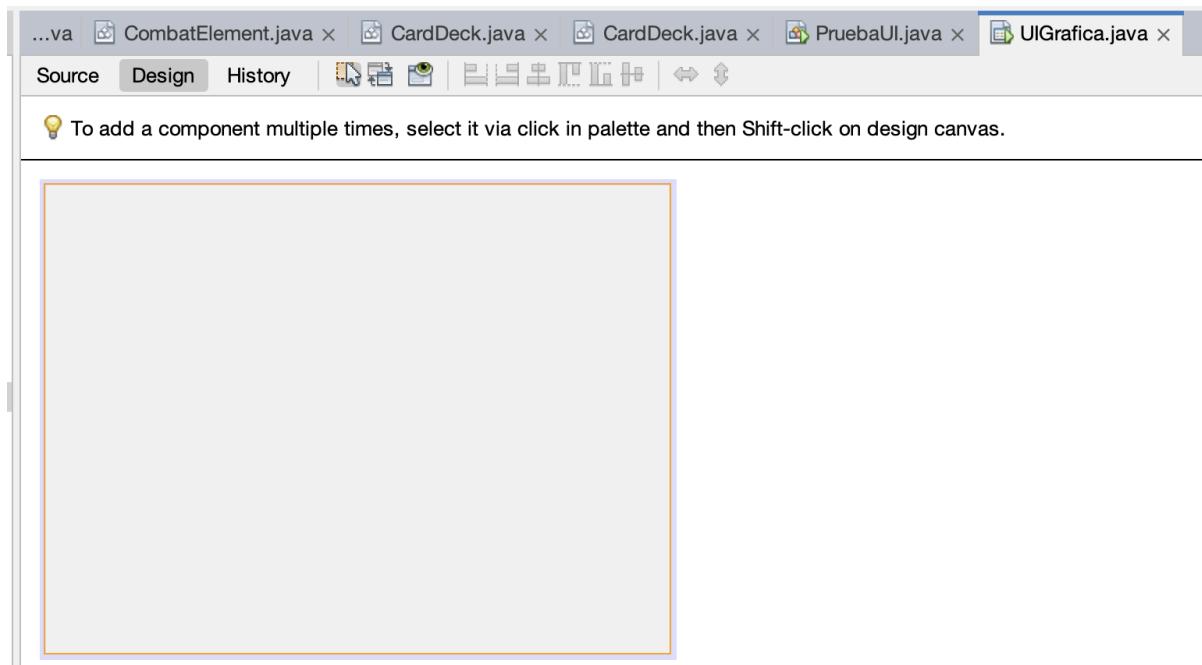


CLASE GraphicalUI

Para ello creamos un nuevo archivo en New → Swing GUI Forms → JFrame Form



Como ves, cuando pulsas a SOURCE cambia de la vista DISEÑO y podemos ver el código de tu clase GraphicalUI, que hereda de la clase javax.swing.JFrame e implementa el interfaz gráfico. Puedes comenzar a introducir elementos en vista diseño.



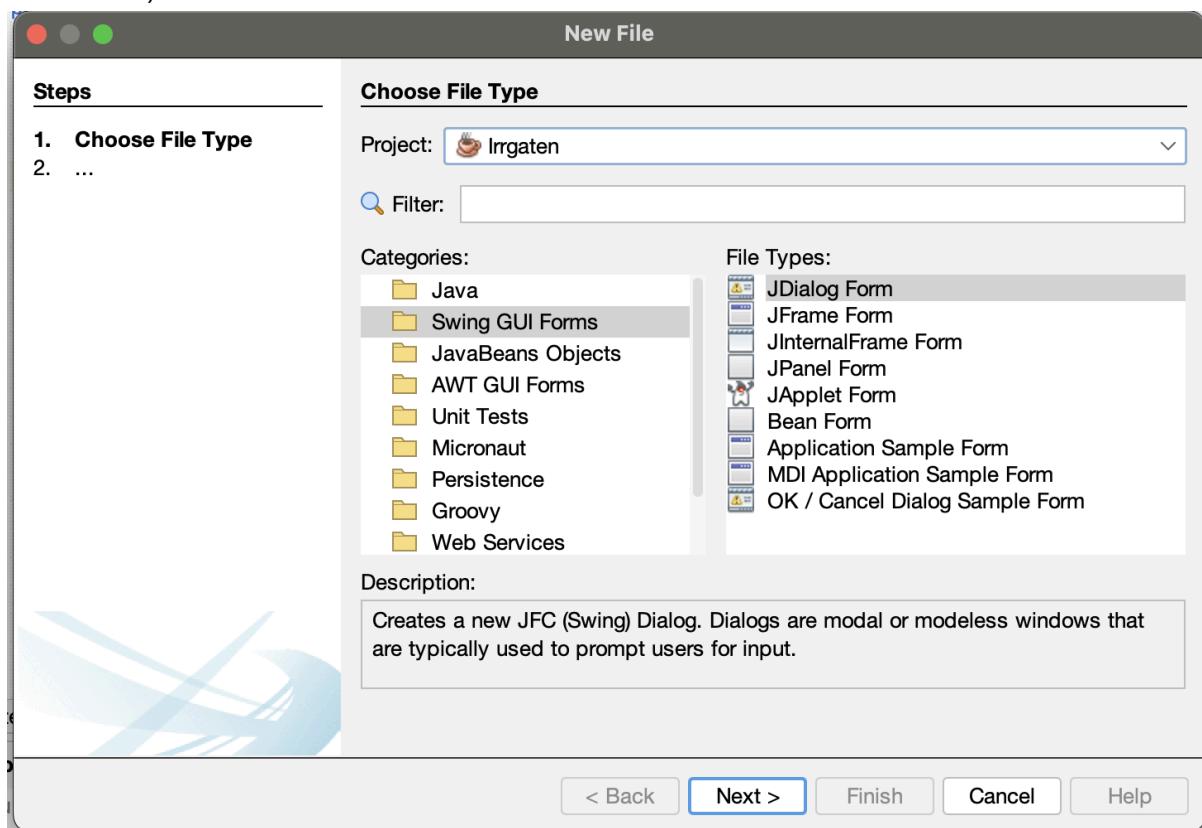
Recomendaciones:

- Asignar variables para referenciar los TextArea, Label y TextField: botón derecho → Change variable name
- Todos los elementos de visualización deben ser no editables (propiedades Editable → false y Focusable → false)
- Implementación del método showGame (GameState gameState):
 - método setText de un elemento: asigna el string a mostrar. Ejemplo:
`this.labyrinth.setText(gameState.getLabyrinthv());
 this.log.setText(gameState.getLog());`
 - Recuerda si lo necesitas convertir a string: Integer.toString(int)
 - Recuerda llamar al método `this.repaint()`, como última instrucción.

CLASE Cursors

Para ello creamos un nuevo archivo en New → Swing GUI Forms → JDialog Form

Si no aparece (botón derecho sobre el nombre del paquete → New... → Other... → Swing GUI Forms)



Añadimos los botones (buttons), y hacemos doble-click sobre el button para crear código fuente asociado a cada uno.



Recuerda que en esta clase deberás:

- Importar import irrgarten.Directions;
- Crear un atributo Directions.
- En cada pulsación de botón, hacer la siguiente operación: (ejemplo Izquierda)


```
this.direction = Directions.LEFT;
this.dispose();
```

(El método dispose cierra el foco de la ventana Cursors)
- implementar el método getDirections()


```
public Directions getDirection() {
    this.setVisible(true);
    return direction;
}
```

Además, en la clase GraphicalUI se

- añade un atributo nuevo:


```
(private Cursors cursors;)
```

y se inicializa en el constructor

```
this.cursors = new Cursors(this, true);
```
- el método nextMove usará a partir de ahora Cursors


```
public Directions nextMove() {
    return this.cursors.getDirection();
}
```

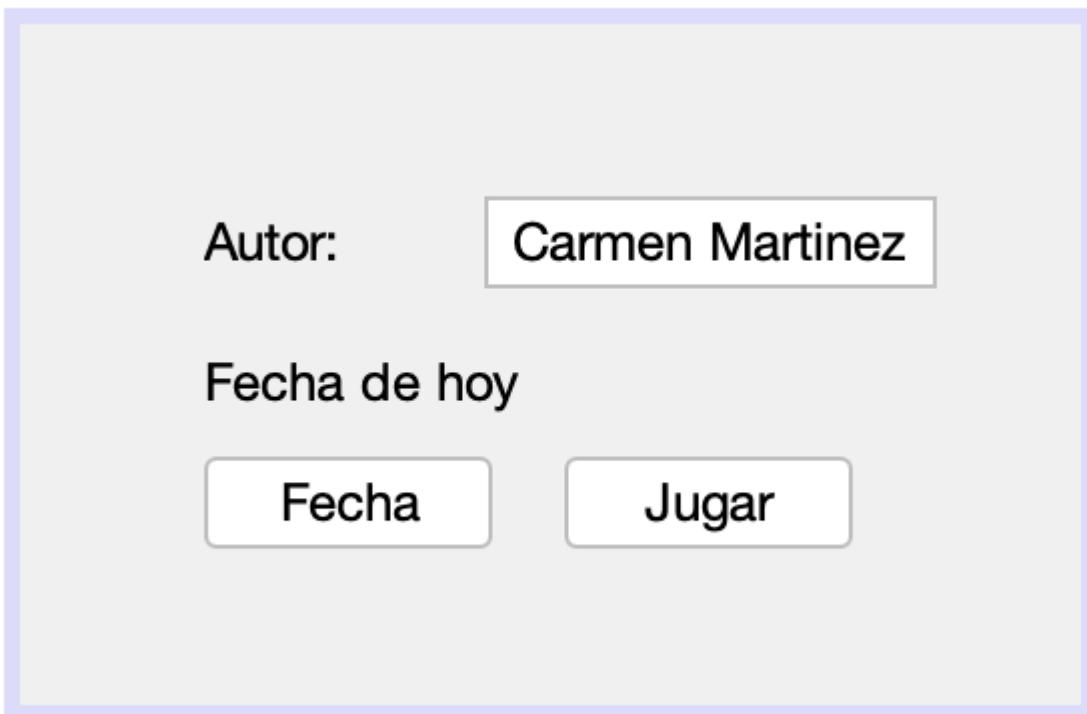
PEQUEÑO TUTORIAL DE SWING

1. Crea un nuevo proyecto, y crea un elemento JFRAME, que llamaremos PruebaUI



2. Crea el panel con los elementos de la imagen, y realiza las siguientes acciones con el botón derecho de ratón:

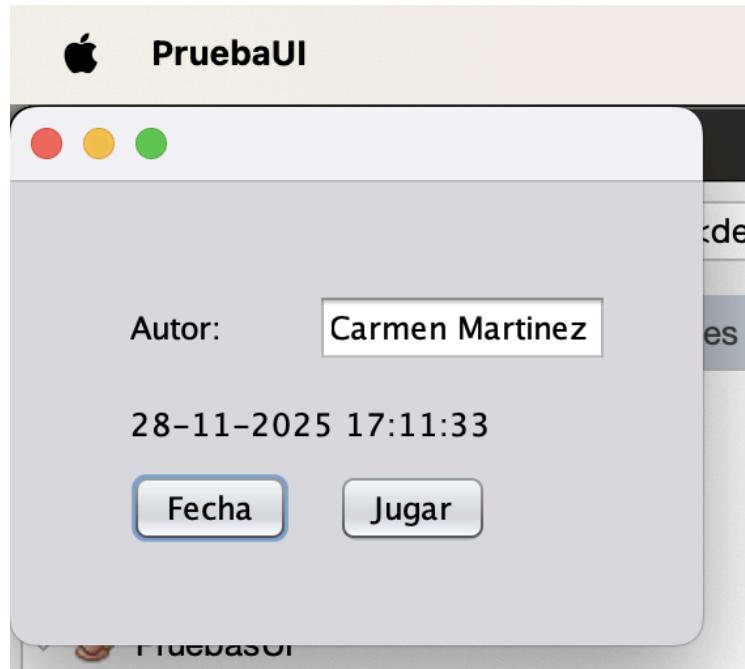
- a. JLabel, y escribe como contenido: Autor
- b. TextField
 - i. Cambia el nombre del objeto (Change Variable Name) a jTAutor
 - ii. Establece el contenido a [tu nombre] (Edit Text)
- c. JLabel que posteriormente incluirá la fecha.
 - i. Cambia el nombre del objeto a jLFecha
 - ii. Escribe como contenido: "Fecha de hoy"
- d. JButton, incluye un botón:
 - i. Cuyo texto ponga Fecha
 - ii. Cambia el nombre de objeto jTActualizar
- e. JButton, incluye un botón:
 - i. Cuyo texto ponga Jugar
 - ii. Cambia el nombre de objeto jTJugar



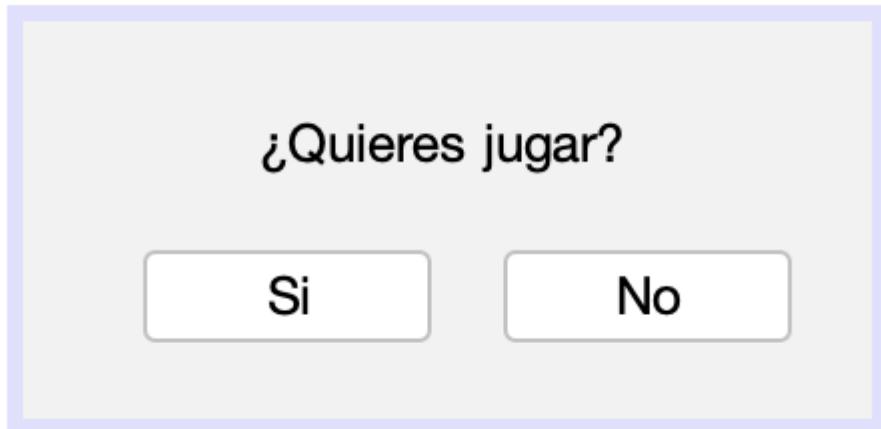
3. Haz doble click sobre fecha, para que se cree automáticamente el método

```
private void jBActualizarActionPerformed(....) {}
```
4. Incluye el siguiente código que obtiene la fecha de hoy en este método y modifica el contenido de jLFecha, con la función setText(texto) a esta fecha:

```
LocalDateTime ahora = LocalDateTime.now();  
DateTimeFormatter formato =  
DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");  
String fechaHoraFormateada = ahora.format(formato);
```
5. Haz en el constructor, no editable jTFAutor (setEditable(false))
6. Ejecuta el código.



7. Crea una nueva clase `JDialog`, con esta estructura (ya deberías saber). Llama a los botones `jBSi`, y `jBNo`
 8. Incluye en la clase `JDialog` creada, la variable de instancia `String respuesta`;
-



9. Cuando se pulse a Si, hay que mandar el mensaje siguiente "Yo Juego" a la ventana principal. Para ello, el el método `jBSiActionPerformed`, actualiza el valor de la variable respuesta a esta frase.
10. Haz lo mismo con `jBNoActionPerformed`, actualizando el valor de respuesta a "No juego"
11. Es importante, antes de finalizar los métodos `ActionPerformed` de `JDialog`, ejecutar la instrucción `dispose()`, para devolver el foco a la ventana de `PruebaUI`.

12. Modifica el código del constructor de la clase PruebaUI (la que hereda de JFrame), para que *siempre exista una instancia de la clase JDialo, llamada ventanaJugar*, para ello crea este atributo de clase.

Pregunta ventanaJugar = new Pregunta(this, true);
y estableces su visibilidad a false para que solo se vea cuando pulsemos al botón Jugar (setVisible(false)).

13. Crea un método en PruebaUI que obtenga de la clase Pregunta, el valor de su variable Respuesta y asigne este valor a jLFecha.

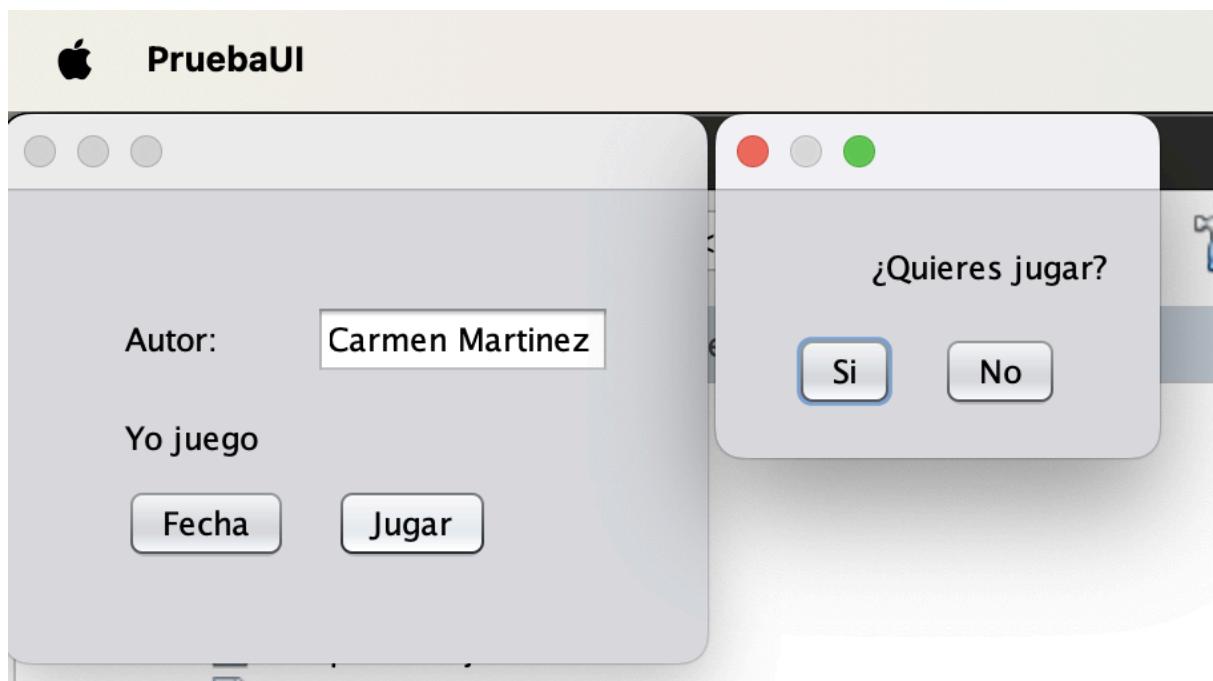
```
public void informacionJuego() {  
    this.jLFecha.setText(ventanaJugar.getRespuesta());  
    //this.repaint(); //esto no haría falta  
}
```

14. Para poder comunicarnos desde Respuesta con PruebaUI, deberemos modificar el constructor. Tendríamos que capturar la ventana padre que nos viene como parámetro de entrada en el constructor, y guardarla en una variable de instancia.

PruebaUI pu;

```
public Pregunta(PruebaUI parent, boolean modal) {.  
. . .  
    pu = parent;  
. . .}
```

15. Ahora en cada método que haga algo en Respuesta, habrá que llamar al método que se quiera ejecutar
pu.informacionJuego();



IMPORTANTE: Este es un tutorial, para que aprendais un poco cómo funciona Spring, de manera sencilla. Pero ojo!!, la práctica de IRRGARTEN, no necesita un main en el JFrame, ya que ya existe un main en la práctica. Tampoco necesita que hagamos los pasos indicados a partir del punto 14, es decir, que guardemos la clase JFrame padre en el JDialog, y que llamemos a un método desde los botones de izquierda, derecha, arriba o abajo, ya que estos valores los devuelve el método getDirection de Irrgarten.

CODIGO

PRUEBAUI.java

```
package irrergarten.UI;

import irrergarten.Directions;

public class Cursors extends javax.swing.JDialog {
    private Directions direction;

    public Cursors(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    public Directions getDirection() {
        this.setVisible(true);
        return direction;
    }

    /**
     * This method is called from within the constructor to
     * initialize the form.
     *
     * WARNING: Do NOT modify this code. The content of this
     * method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
    Code">
    private void initComponents() {

        up = new javax.swing.JButton();
        left = new javax.swing.JButton();
        right = new javax.swing.JButton();
        down = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_C
LOSE);

        up.setText("ARRIBA");
    }
}
```



```
nment.LEADING)
        .addComponent(down,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(up,
javax.swing.GroupLayout.PREFERRED_SIZE, 81,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(109, Short.MAX_VALUE))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(up)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(left)
        .addComponent(right))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(down)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);
}

pack();
}// </editor-fold>

private void leftActionPerformed(java.awt.event.ActionEvent evt) {
    this.direction = Directions.LEFT;
    this.dispose();
}

private void upActionPerformed(java.awt.event.ActionEvent evt) {
    this.direction = Directions.UP;
    this.dispose();
}

private void rightActionPerformed(java.awt.event.ActionEvent evt) {
    this.direction = Directions.RIGHT;
    this.dispose();
}
```

```
    private void downActionPerformed(java.awt.event.ActionEvent evt) {
        this.direction = Directions.DOWN;
        this.dispose();
    }

    // Variables declaration - do not modify
    private javax.swing.JButton down;
    private javax.swing.JButton left;
    private javax.swing.JButton right;
    private javax.swing.JButton up;
    // End of variables declaration
}
```

PREGUNTA.JAVA

```
/*
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default
.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JDialog.java to
edit this template
 */

/**
 *
 * @author cmcruz
 */
public class Pregunta extends javax.swing.JDialog {

    /**
     * Creates new form Pregunta
     */
    String respuesta;
    PruebaUI pu;

    public String getRespuesta() {
        return respuesta;
    }
    public Pregunta(PruebaUI parent, boolean modal) {
        respuesta = "No hay datos";
        pu = parent;
        super(parent, modal);
        initComponents();
    }
}
```

```
/**  
 * This method is called from within the constructor to  
 * initialize the form.  
 * WARNING: Do NOT modify this code. The content of this  
 * method is always  
 * regenerated by the Form Editor.  
 */  
@SuppressWarnings("unchecked")  
// <editor-fold defaultstate="collapsed" desc="Generated  
Code">  
private void initComponents() {  
  
    jLabel1 = new javax.swing.JLabel();  
    jBSi = new javax.swing.JButton();  
    jBNo = new javax.swing.JButton();  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_C  
LOSE);  
  
    jLabel1.setText("¿Quieres jugar?");  
  
    jBSi.setText("Si");  
    jBSi.addActionListener(new  
java.awt.event.ActionListener() {  
        public void  
actionPerformed(java.awt.event.ActionEvent evt) {  
            jBSiActionPerformed(evt);  
        }  
    });  
  
    jBNo.setText("No");  
    jBNo.addActionListener(new  
java.awt.event.ActionListener() {  
        public void  
actionPerformed(java.awt.event.ActionEvent evt) {  
            jBNoActionPerformed(evt);  
        }  
    });  
  
    javax.swing.GroupLayout layout = new  
javax.swing.GroupLayout(getContentPane());  
    getContentPane().setLayout(layout);  
    layout.setHorizontalGroup(  
  
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(layout.createSequentialGroup()  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment  
MENT.LEADING)  
    .addGroup(layout.createSequentialGroup()  
        .addGap(59, 59, 59)  
        .addComponent(jLabel1))
```

```
        .addGroup(layout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addComponent(jBSi)
            .addGap(18, 18, 18)
            .addComponent(jBNo)))
        .addContainerGap(20, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(22, 22, 22)
        .addComponent(jLabel1)
        .addGap(18, 18, 18)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jBSi)
        .addComponent(jBNo))
        .addContainerGap(19, Short.MAX_VALUE))
    );

    pack();
}// </editor-fold>

private void jBSiActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.respuesta="Yo juego";
    pu.informacionJuego();

    this.dispose();
}

private void jBNoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.respuesta="NO juego";
    pu.informacionJuego();
    this.dispose();
}

/**
 * @param args the command line arguments
 */

// Variables declaration - do not modify
private javax.swing.JButton jBNo;
private javax.swing.JButton jBSi;
private javax.swing.JLabel jLabel1;
// End of variables declaration
```

}