

# Programación Orientada a Objetos

## Interfaz gráfica

CEIS

2025-1

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

Arquitectura

$P(V+C+I)$

¿Concurrencia?

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

Arquitectura

$P(V+C+I)$

¿Concurrencia?

## Mundial



Mundial de Fútbol Cupi2

**Equipos**

Equipo : Colombia

Director Técnico : Jose Pekerman

**Jugadores**



Nombre: RODRIGUEZ James

Edad: 23 años

Posición: Centrocampista

Altura: 1.8 mts

Peso: 77.0 Kgs

Salario: 6,000,000.00 euros anuales

**Opciones**

Agregar Equipo Agregar Jugador Calcular Nómina

Una interfaz gráfica:

Esta construída a partir de **componentes** gráficos

Los **componentes** se organizan en **componentes contenedores**.

A los **componentes contenedores** se les asocia un **estilo** para lograr el diseño deseado.

- ▶ Los componentes generan **eventos** avisando acciones de usuario

Los **eventos** que se quieren atender se programan asociando **oyentes** a los componentes que los generan

**Programación orientada por eventos**

## Mundial

Mundial de Fútbol CUPi2

**Mundial de Fútbol CUPi2**

**Equipos**

Equipo :

Director Técnico :

**Jugadores**



Nombre:

Edad: 23 años

Posición: Centrocampista

Altura: 1.8 mts

Peso: 77.0 Kgs

Salario: 6,000,000.00 euros anuales

**Opciones**

## Con respecto a la forma

- ▶ ¿Cuáles son componentes básicos (las hojas)?
- ▶ ¿Cuáles son componentes contenedores?
- ▶ ¿Cuál es el gran componente?

## Mundial



Mundial de Fútbol Cupi2

**Equipos**

Equipo :

Director Técnico :

**Jugadores**

 Nombre:

Edad: 23 años

Posición: Centrocampista

Altura: 1.8 mts

Peso: 77.0 Kgs

Salario: 6,000,000.00 euros anuales

**Opciones**

## Con respecto al comportamiento

- ▶ ¿Cuáles componentes deberían estar activos?
- ▶ ¿Cuáles acciones están pre-programadas?
- ▶ ¿Cuáles acciones debemos programar?

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

Arquitectura

$P(V+C+I)$

¿Concurrencia?

# Diseño arquitectónico

## Patrón: MVC

### 1. Modelo

Contiene los elementos que representa la información propia del dominio de la aplicación

### 2. Vista

Contiene los elementos que permiten mostrar la información del dominio al usuario

### 3. Controlador

Contiene los elementos que sirven de puente de comunicación entre el modelo y la vista

## Arquitectura tres niveles

### 1. Presentación

### 2. Dominio

### 3. Persistencia

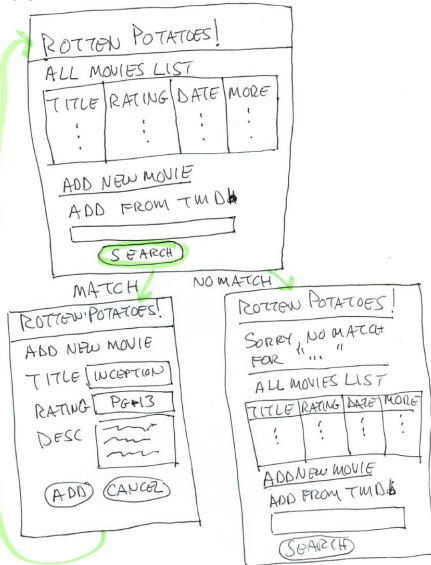
## Con respecto a los componentes

- ▶ ¿Cuál es la relación entre los componentes MVC y los de 3 niveles?
- ▶ ¿Cuáles hemos usado? ¿Cuáles hemos programado?



# Diseño gráfico

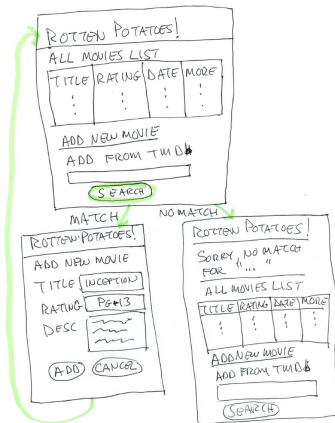
## Películas



## Máquina de estado

- **NODOS:** Componentes independientes
- **ARCOS:** Transiciones (condición/acción)

# Diseño gráfico. Mundial.



## Diseñando

- ▶ DISEÑEMOS. Mundial Completo.
- ▶ ¿Cuántos componentes independientes tendríamos?

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

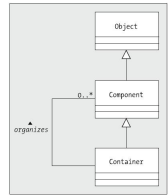
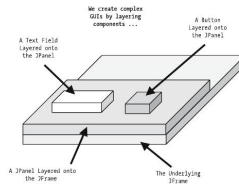
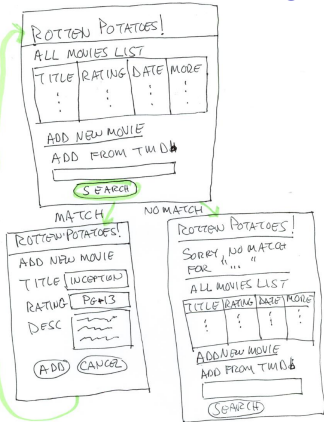
Arquitectura

P(V+C+I)

¿Concurrencia?

# MVC

## FORMA: Vista - Negro



- ▶ Esta construída a partir de **componentes** gráficos
- ▶ Los **componentes** se organizan en **componentes contenedores**.
- ▶ A los **componentes contenedores** se les asocia un **estilo** para lograr el diseño deseado.

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

**Componentes**

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

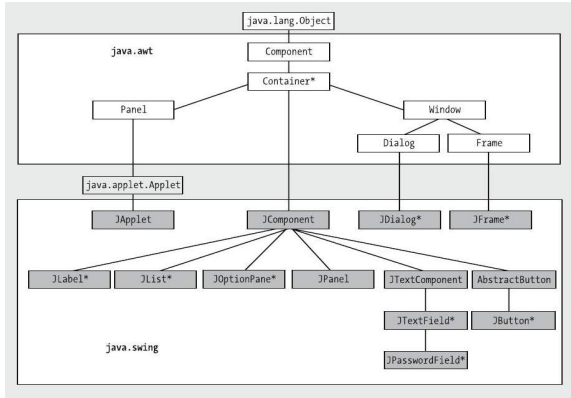
## Aspectos finales

Arquitectura

$P(V+C+I)$

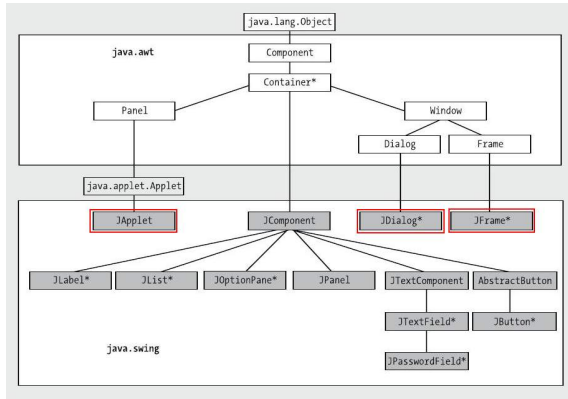
¿Concurrencia?

# SWING



- ▶ Superior  
JApplet, JFrame, JDialog, JOptionPane
- ▶ Intermedio  
JPanel, JScrollPane, JTabbedPane
- ▶ Atómico  
JLabel, JTextField, JButton, JList, JPasswordField

# SWING



- Superior  
JApplet, JFrame, JDialog, JOptionPane

# Componentes

## JFrame - JDialog

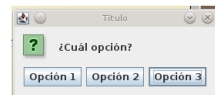
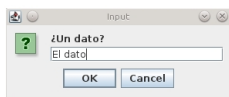
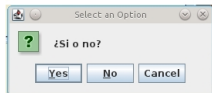
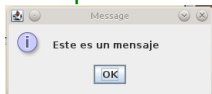


- ▶ Sólo son marcos, su contenido se debe diseñar y codificar.
- ▶ Lucen igual (Titulo, botones propios del sistema)
- ▶ **JFrame:** Componente de más alto nivel para aplicaciones independientes. Un JFrame por aplicación. Controla los JDialog.
- ▶ **JDialog:** Componente de alto nivel para aplicaciones (JFrame JApplet)  
Todos los JDialog que se necesiten por aplicación



# Componentes

## JOptionPane



Method Name	Description
showConfirmDialog	Asks a confirming question, like yes/no/cancel.
showInputDialog	Prompt for some input.
showMessageDialog	Tell the user about something that has happened.
showOptionDialog	The Grand Unification of the above three.

Están prefabricados, listos para su uso.

```
JOptionPane.showMessageDialog(this,"Este es un mensaje");

int siNo=JOptionPane.showConfirmDialog(this,"¿Si o no?");
/*JOptionPane.YES_OPTION, JOptionPane.NO_OPTION, JOptionPane.CANCEL_OPTION*/

String dato=JOptionPane.showInputDialog(this,"¿Un dato?");

String [] opciones={"Opción 1", "Opción 2", "Opción 3"};
int opcion= JOptionPane.showOptionDialog(this, "¿Cuál opción?", "Título",
JOptionPane.OK_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE,
null,
opciones,"Opción 3");
```

# Detalle diseño. Mundial.

¿Componentes superiores?

Mundial de Fútbol CUPi2



## Mundial de Fútbol Cupi2

**Equipos**

Equipo :

Director Técnico :



**Jugadores**



Nombre:

Edad: 23 años

Posición: Centrocampista

Altura: 1.8 mts

Peso: 77.0 Kgs

Salario: 6,000,000.00 euros anuales

**Opciones**

# Componentes

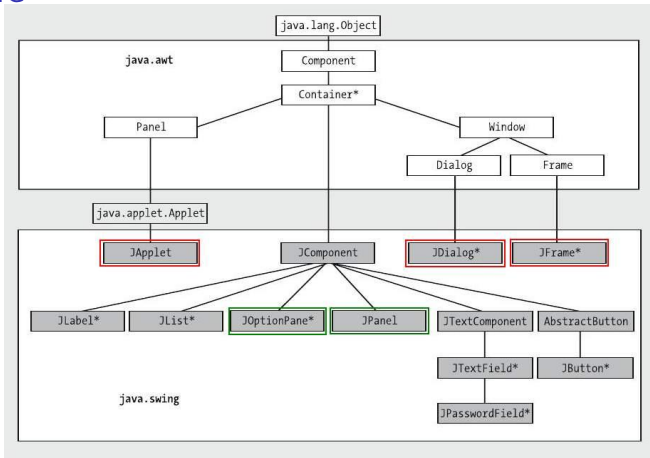


## JFrame

```
public class MundialGUI extends JFrame {  
  
    /*...*/  
  
    private MundialGUI(){  
        mundial=new Mundial();    /*MODELO*/  
        prepareElementos();        /*VISTA*/  
        prepareAcciones();         /*CONTROLADOR*/  
  
        inicie();  
    }  
  
    /*...*/  
  
    public static void main( String[] args ){  
        MundialGUI gui=new MundialGUI();  
        gui.setVisible(true);  
    }  
}
```

# Componentes. Intermedio.

## SWING



### ► Intermedios

**JPanel**, `JScrollPane`, `JTabbedPane`

Contenedor que debe estar en otro contenedor, permite organizar la información

# Detalle diseño. Mundial.

## ¿Componentes intermedios?

Mundial de Fútbol CUPi2



### Mundial de Fútbol Cupi2

**Equipos**

Equipo :

Director Técnico :



**Jugadores**



Nombre:

Edad: 23 años

Posición: Centrocampista

Altura: 1.8 mts

Peso: 77.0 Kgs

Salario: 6,000,000.00 euros anuales

**Opciones**

# Componentes. Intermedio.

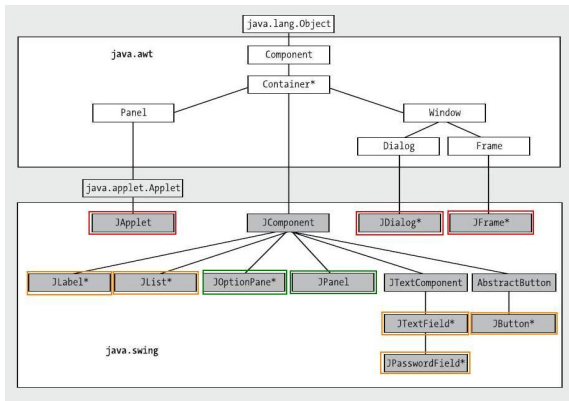


## JFrame

```
public class MundialGUI extends JFrame {  
  
    private JPanel panelImagen;  
    private JPanel panelEquipos;  
    private JPanel panelJugadores;  
    private JPanel panelBotones;
```

# Componentes. Básicos

## SWING



### ► Básicos

`JLabel`, `JTextField`, `JButton`, `JList`, `JPasswordField`

# Detalle diseño. Mundial.

## ¿Componentes básicos?

### Basic Controls

Simple components that are used primarily to get input from the user; they may also show simple state.



JButton



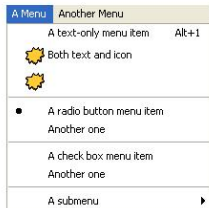
JCheckBox



JComboBox



JList



JMenu



JRadioButton



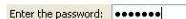
JSlider



JSpinner



JTextField



JPasswordField



# Detalle diseño. Mundial.

## ¿Componentes básicos?

**Equipos**

**Equipo :**  ▼

**Director Técnico :**




# Detalle diseño. Mundial.

## ¿Componentes básicos?

Equipos

Equipo :  ▼

Director Técnico :



```
/*Panel equipos*/  
private JLabel etiquetaEquipo;  
private JComboBox comboEquipos;  
private JLabel etiquetaDirector;  
private JTextField textoNombreDirector;  
private JLabel etiquetaBandera;
```

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

**Estilos**

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

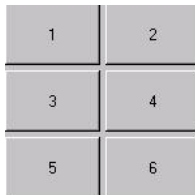
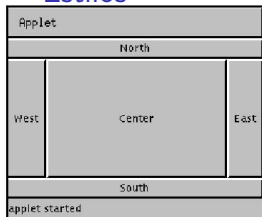
Arquitectura

P(V+C+I)

¿Concurrencia?

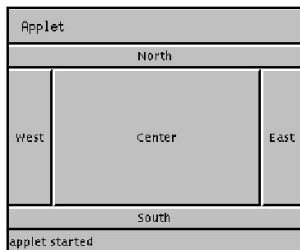
# Estilos

## Estilos



- ▶ BorderLayout
- ▶ GridLayout
- ▶ FlowLayout

# BorderLayout



```
setLayout(new BorderLayout());  
add(new Button("North"), BorderLayout.NORTH);  
add(new Button("South"), BorderLayout.SOUTH);  
add(new Button("East"), BorderLayout.EAST);  
add(new Button("West"), BorderLayout.WEST);  
add(new Button("Center"), BorderLayout.CENTER);
```

# GridLayout

1	2
3	4
5	6

```
setLayout(new GridLayout(3,2));  
add(new Button("1"));  
add(new Button("2"));  
add(new Button("3"));  
add(new Button("4"));  
add(new Button("5"));  
add(new Button("6"));
```

# FlowLayout



```
setLayout( new FlowLayout);

// Create several labels.
JLabel l1 = new JLabel("A short label");
JLabel l2 = new JLabel("A      long label");
JLabel l3 = new JLabel("Another label");

// Add them to the GUI.
add(l1);
add(l2);
add(l3);
```

# GUI

¿Estilo del panel botones?

Mundial de Fútbol CUPi2



## Mundial de Fútbol Cupi2

**Equipos**

Equipo :

Director Técnico :



**Jugadores**



Nombre:

Edad: 23 años

Posición: Centrocampista

Altura: 1.8 mts

Peso: 77.0 Kgs

Salario: 6,000,000.00 euros anuales

**Opciones**



## Estilo botones



```
private void elementosPanelBotones(){  
    panelBotones= new JPanel();  
    panelBotones.setBorder(new CompoundBorder(new EmptyBorder( 5, 5, 5, 5 ),  
                                              new TitledBorder( "Opciones" )));  
    panelBotones.setLayout( new GridLayout( 1, 3, 3, 3 ) );  
    botonAgregarEquipo = new JButton( "Agregar Equipo" );  
    botonAgregarJugador = new JButton( "Agregar Jugador" );  
    botonCalcularNomina = new JButton( "Calcular Nómina" );  
  
    panelBotones.add(botonAgregarEquipo);  
    panelBotones.add(botonAgregarJugador);  
    panelBotones.add(botonCalcularNomina);  
}
```

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

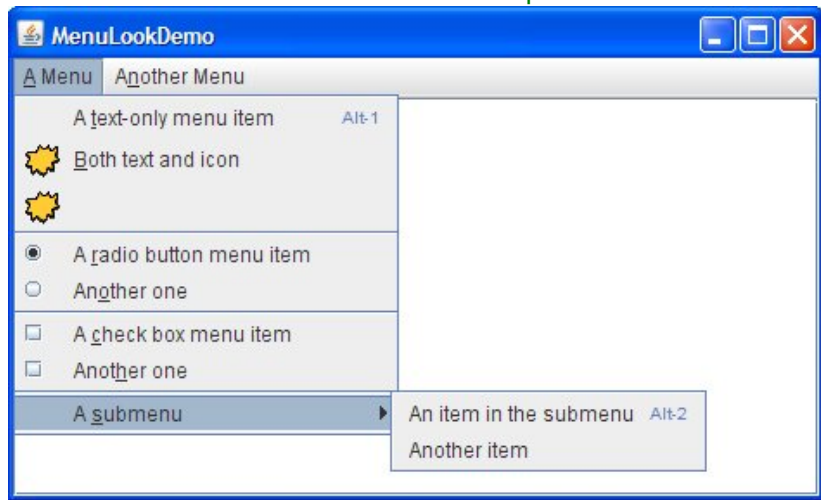
Arquitectura

P(V+C+I)

¿Concurrencia?

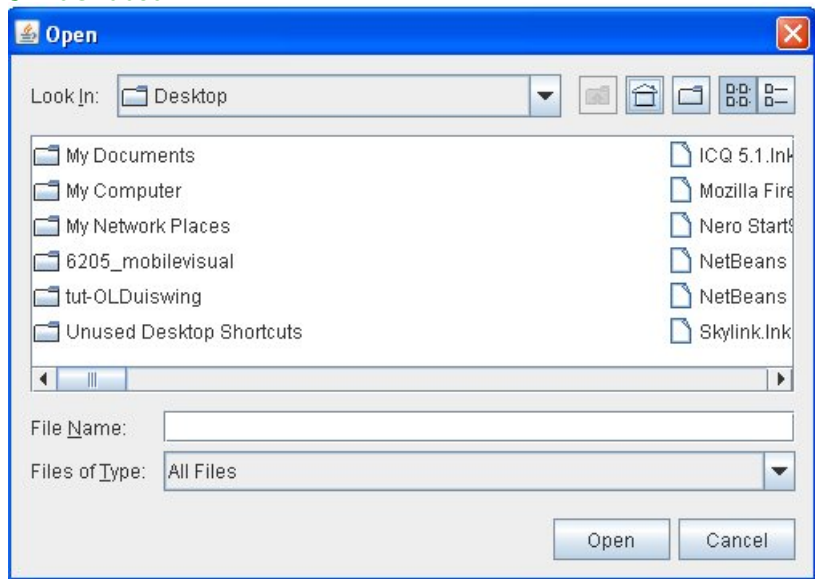
# Menu

JMenuBar JMenu JMenuItem JMenuSeparator



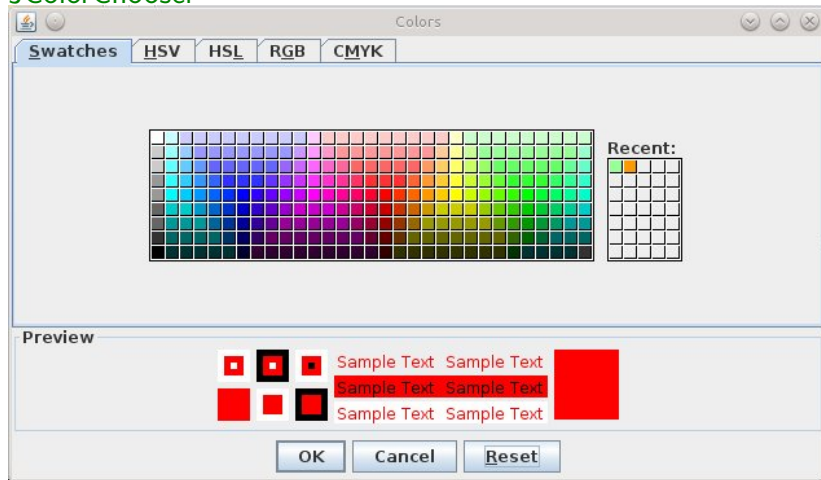
# Control Archivos

## JFileChooser



# Control Colores

## JColorChooser



# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

**Pantalla**

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

Arquitectura

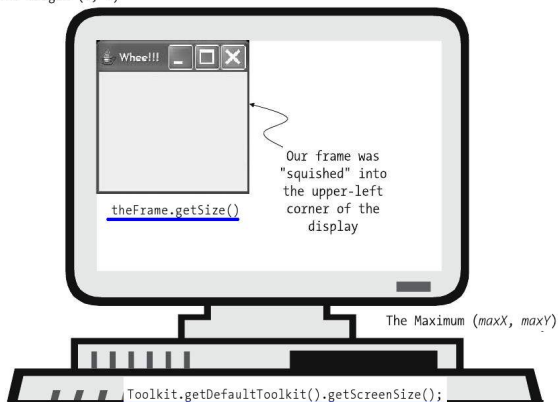
P(V+C+I)

¿Concurrencia?

# Componentes

## Pantalla

The Origin (0, 0)



# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

Arquitectura

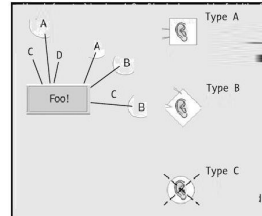
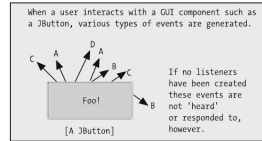
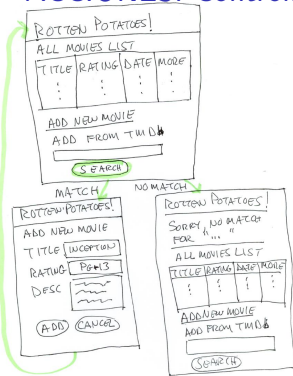
P(V+C+I)

¿Concurrencia?



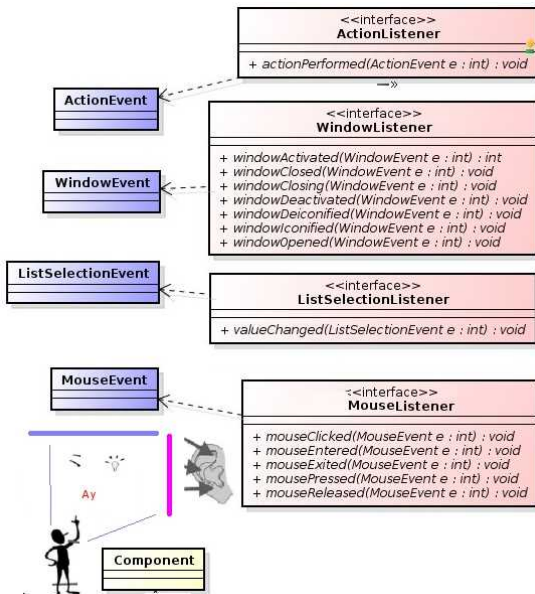
# MVC

## ACCIONES: Controlador - Verde

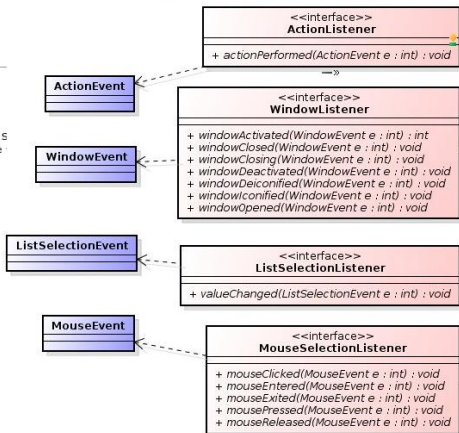


- Un componente puede generar un gran número de **eventos**.  
Son **objetos** `java.awt.event.AWTEvent`
- El programador decide cuáles eventos necesita atender: asigna un **oyente** al componente y programa la acción correspondiente  
Son **objetos** que implementan la *interface* necesaria para oír el evento `java.awt.event.Listener`

# Eventos



# Eventos



¿Qué oyentes?

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

**Esquema**

Adaptadores

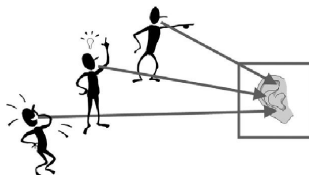
## Aspectos finales

Arquitectura

$P(V+C+I)$

¿Concurrencia?

# Manejo de eventos



## Receta

1. Definir un oyente de eventos
2. Registrarlo a la fuente de eventos
3. Implementar la acción correspondiente

# Manejo de eventos

## Definiendo

```
ActionListener l = new X();
```

where class X is in turn declared as follows:

```
class X implements ActionListener {  
    // There is one method that we must implement to satisfy  
    // the ActionListener interface:  
    public void actionPerformed(ActionEvent e) {  
        // Program the behind-the-scenes logic that we want to  
        // execute when an ActionEvent is detected;  
        // details omitted ...  
    }  
}
```

## Registrando

```
m Button.addActionListener(l);
```

## Programando acción

# Manejo de eventos

## Definiendo [ : ( ]

```
ActionListener l = new X();
```

where class X is in turn declared as follows:

```
class X implements ActionListener {  
    // There is one method that we must implement to satisfy  
    // the ActionListener interface:  
    public void actionPerformed(ActionEvent e) {  
        // Program the behind-the-scenes logic that we want to  
        // execute when an ActionEvent is detected;  
        // details omitted ...  
    }  
}
```

[ : ( ] Una clase por cada oyente

## Registrando

```
mButton.addActionListener(l);
```

## Programando acción [ : ( ]

[ : ( ] Programamos la acción en la clase del oyente, no en la actual

# Manejo de eventos

## Definiendo [ : ) ]

```
ActionListener l = new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        ...EventAction ( )  
    }  
}; // Don't forget the ending semicolon!!!
```

Clase anónima

\$1  
\$2

[ : ) ] Una clase **anónima** por cada oyente

## Registrando

```
mButton.addActionListener(l);
```

## Programando acción [ : ) ]

```
public void ...EventAction ( ActionEvent e ? ) {  
    // Program the behind-the-scenes logic that we want to  
    // execute when an ActionEvent is detected;  
    // details omitted ...  
}
```

[ : ) ] Programamos en la clase actual



# Mundial

## Mundial

Mundial de Fútbol CUPi2

 **Mundial de Fútbol CUPi2**

**Equipos**

Equipo :

Director Técnico :



**Jugadores**



Nombre:

Edad: 23 años

Posición: Centrocampista

Altura: 1.8 mts

Peso: 77.0 Kgs

Salario: 6,000,000.00 euros anuales

**Opciones**

Con respecto al comportamiento

- ▶ ¿Cuántos oyentes?
- ▶ ¿Cuáles son los pares oyente-acción?

## Mundial

## Acciones. Definiendo y registrando

```
public void prepareAcciones(){
    /*Marco*/
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    /*Equipos*/
    comboEquipos.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                comboEquiposAccion( );
            }
        });

    /*Jugadores*/
    comboJugadores.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                comboJugadoresAccion( );
            }
        });

    /*Panel botones*/
    botonAgregarEquipo.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                botonAgregarEquipoAccion( );
            }
        });
    botonAgregarJugador.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                botonAgregarJugadorAccion( );
            }
        });
    botonCalcularNomina.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                botonCalcularNominaAccion( );
            }
        });
}
```

## Acciones. Programando acción

```
public void comboEquiposAccion() {
    String nombreEquipo = (String) comboEquipos.getSelectedItemAt( );
    equipoSeleccionado = mundial.busqueEquipo(nombreEquipo);
    if (equipoSeleccionado != null){
        cambieEquipo(equipoSeleccionado);
    }
}

public void comboJugadoresAccion(){
    String nombreJugador = (String) comboJugadores.getSelectedItemAt( );
    if (nombreJugador != null) {
        Jugador jugador = equipoSeleccionado.busqueJugador(nombreJugador);
        if (jugador != null) {
            cambieJugador(jugador);
        }
    }
}

private void botonAgregarEquipoAccion(){
    DialogoCrearEquipo dialogo = new DialogoCrearEquipo(this);
    dialogo.setVisible( true );
}

private void botonAgregarJugadorAccion(){
    DialogoCrearJugador dialogo = new DialogoCrearJugador(this, equipoSeleccionado );
    dialogo.setVisible( true );
}

private void botonCalcularNominaAccion(){
    String resultado=formatearValor(mundial.calculerNomina(equipoSeleccionado.getPais( )));
    JOptionPane.showMessageDialog(this, resultado, "NOMINA", JOptionPane.INFORMATION_MESSAGE );
}
```

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

**Adaptadores**

## Aspectos finales

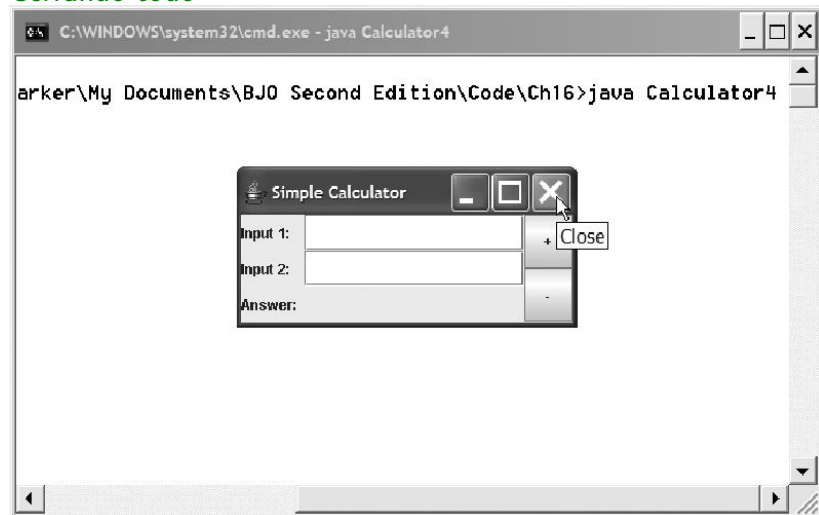
Arquitectura

$P(V+C+I)$

¿Concurrencia?

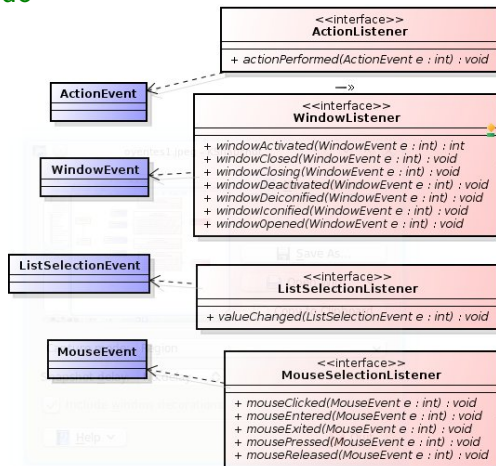
# Manejo de eventos

## Cerrando todo



# Manejo de eventos

## Cerrando todo

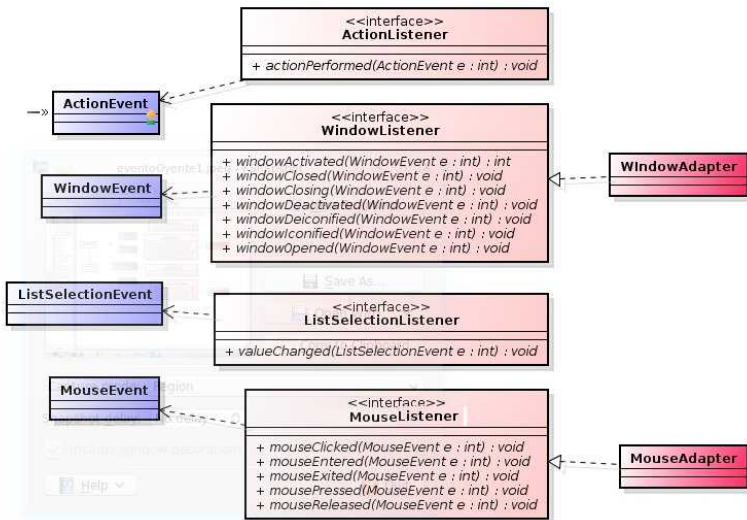


# Manejo de eventos

## Cerrando todo

```
public class Calculator5 extends JFrame {  
  
    public Calculator5() {  
  
        .....  
        WindowListener w = new WindowListener() {  
            public void windowClosing(WindowEvent e) {  
                // Note the need to preface "this." with  
                // the name of the outer class.  
                Calculator5.this.dispose(); // See notes after the code  
                System.exit(0);  
            }  
            public void windowOpened(WindowEvent e) { }  
            public void windowClosed(WindowEvent e) { }  
            public void windowIconified(WindowEvent e) { }  
            public void windowDeiconified(WindowEvent e) { }  
            public void windowActivated(WindowEvent e) { }  
            public void windowDeactivated(WindowEvent e) { }  
        };  
  
        this.addWindowListener(w);  
  
        this.setVisible(true);  
    }  
  
    // etc.  
}
```

# Adaptadores





# Manejo de eventos

## Cerrando todo

```
public class Calculator6 extends JFrame {  
    public Calculator6() {  
  
        ....  
        WindowListener w = new WindowAdapter() {  
            public void windowClosing(WindowEvent e) {  
                Calculator6.this.dispose();  
                System.exit(0);  
            }  
        };  
  
        this.addWindowListener(w);  
  
        this.setVisible(true);  
    }  
  
    // etc.  
}
```



# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

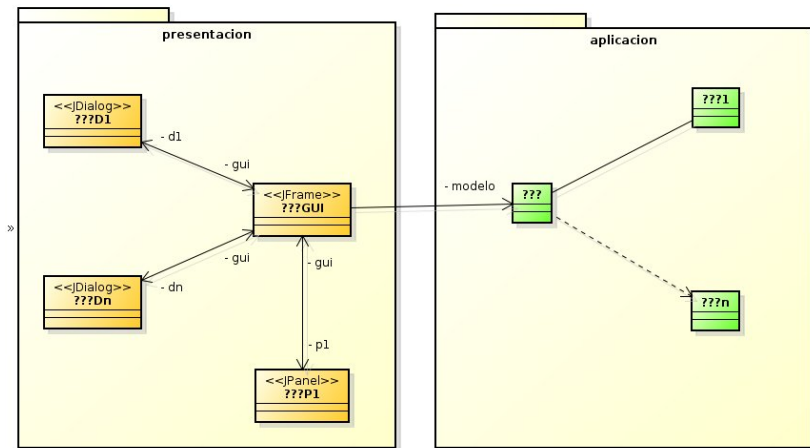
## Aspectos finales

Arquitectura

P(V+C+I)

¿Concurrencia?

# Mundial



# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

Arquitectura

$P(V+C+I)$

¿Concurrencia?

# Mundial

P(V+C+I)

```
public class MundialGUI extends JFrame {  
  
    /*...*/  
  
    private MundialGUI(){  
        mundial=new Mundial();    /*MODELO*/  
        prepareElementos();        /*VISTA*/  
        prepareAcciones();         /*CONTROLADOR*/  
  
        inicie();  
    }  
  
    /*...*/  
  
    public static void main( String[] args ){  
        MundialGUI gui=new MundialGUI();  
        gui.setVisible(true);  
    }  
}
```

# Mundial

## P(Vista+Cotrolador)

### Elementos

```
private void prepareElementos(){
    setTitle("Mundial de futbol CUP12" );
    elementosPanelImagen();
    elementosPanelEquipos();
    elementosPanelJugadores();
    elementosPanelBotones();
    setLayout(new FlowLayout());
    add(panelImagen);
    add(panelEquipos);
    add(panelJugadores);
    add(panelBotones);
    setSize( 560, 600 );
    centre();
}
```

### Acciones

```
public void prepareAcciones(){
    /*Marco*/
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    /*Equipos*/
    comboEquipos.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                comboEquiposAccion( );
            }
        });

    /*Jugadores*/
    comboJugadores.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                comboJugadoresAccion( );
            }
        });

    /*Panel botones*/
    botonAgregarEquipo.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                botonAgregarEquipoAccion( );
            }
        });
    botonAgregarJugador.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                botonAgregarJugadorAccion( );
            }
        });
    botonCalcularNomina.addActionListener(
        new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                botonCalcularNominaAccion( );
            }
        });
}
```

# Agenda

## Conceptos

Introducción

Diseño

## Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Pantalla

## Acciones

Contexto

Esquema

Adaptadores

## Aspectos finales

Arquitectura

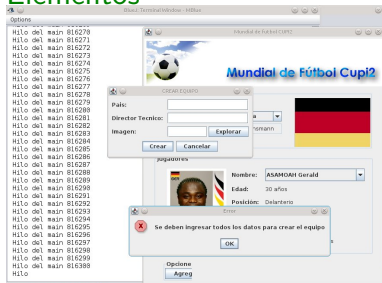
$P(V+C+I)$

¿Concurrencia?

# Mundial

Hilos)

Elementos



```
public static void main( String[] args ){  
    MundialGUI gui=new MundialGUI();  
    gui.setVisible(true);  
    int i=1;  
    while (true){  
        System.out.println("Hilo del main "+i);  
        i++;  
    }  
}
```



# Hilos

## Crear equipo

Mundial de futbol CUPi2

 **Mundial de Fútbol Cupi2**

**Equipos**

Equipo :  

Director

**Jugadores**

OLIVER KAHN

**Posición:** Arquero

**Altura:** 1.88 mts

**Peso:** 91.0 Kgs

**Salario:** \$ 18,000.50 millones anuales

**Opciones**

**CREAR EQUIPO**

**Pais:**

**Director Tecnico:**

**Imagen:**

# Manejo de eventos

## Funciones



# Manejo de eventos

## Crear equipo

```
public DialogoCrearEquipo( InterfazMundial ventana ) {  
    super( ventana, "CREAR EQUIPO", false );  
    principal = ventana;  
    prepareElementos();  
    prepareAcciones();  
}
```