

Programación Orientada a Objetos

Interfaz gráfica

CEIS

2019-01

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

$P(V+C+I)$

¿Concurrencia?

Aspectos finales

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

P(V+C+I)

¿Concurrencia?

Aspectos finales

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

P(V+C+I)

¿Concurrencia?

Aspectos finales

GUI

Mundial

Mundial de Fútbol CUPi2



Mundial de Fútbol Cupi2

Equipos

Equipo :

Director Técnico :



Jugadores



Nombre:

Edad:

Posición:

Altura:

Peso:

Salario:

Opciones

Mundial



Una interfaz gráfica:

- ▶ Esta construída a partir de **componentes** gráficos
- ▶ Los componentes generan **eventos** avisando acciones de usuario

Mundial



Una interfaz gráfica:

- ▶ Esta construída a partir de **componentes** gráficos

Los **componentes** se organizan en **componentes contenedores** a los que se les asocia un **estilos** para lograr el diseño deseado.

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

Los **eventos** pueden ser atendidos si se asocian **oyentes** a los mismos

Mundial

Mundial de Fútbol CUPi2

Mundial de Fútbol CUPi2

Equipos

Equipo :

Director Técnico :

Jugadores



Nombre:

Edad:

Posición:

Altura:

Peso:

Salario:

Opciones

Una interfaz gráfica:

- ▶ Esta construída a partir de **componentes** gráficos

Los **componentes** se organizan en **componentes contenedores** a los que se les asocia un **estilos** para lograr el diseño deseado.

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

Los **eventos** pueden ser atendidos si se asocian **oyentes** a los mismos

Programación orientada por eventos

Mundial



Mundial de Fútbol CUPi2

Equipos

Equipo :

Director Técnico :

Jugadores

Nombre:

Edad:

Posición:

Altura:

Peso:

Salario:

Opciones

Con respecto a la forma

- ¿Cuáles son componentes básicos (las hojas)?

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 a la forma

- ▶ ¿Cuáles son componentes básicos (las hojas)?
- ▶ ¿Cuáles son componentes contenedores?

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?

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

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

Mundial



Mundial de Fútbol CUPi2

Equipos

Equipo :

Director Técnico :

Jugadores



Nombre:

Edad:

Posición:

Altura:

Peso:

Salario:

Opciones

Con respecto al comportamiento

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

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

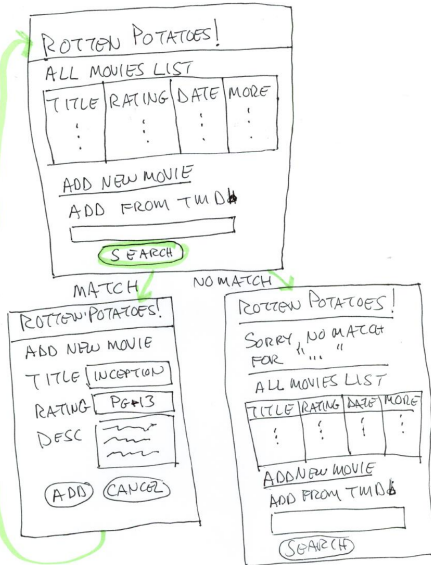
$P(V+C+I)$

¿Concurrencia?

Aspectos finales

Diseño gráfico

Películas



Máquina de estado

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

Diseño gráfico. 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

ROTTEN POTATOES!

ALL MOVIES LIST

TITLE	RATING	DATE	MORE
...

ADD NEW MOVIE

ADD FROM TMDb

SEARCH

MATCH

NO MATCH

ROTTEN POTATOES!

ADD NEW MOVIE

TITLE INCEPTION

RATING PG-13

DESC

ADD CANCEL

ROTTEN POTATOES!

SORRY NO MATCH FOR ...

ALL MOVIES LIST

TITLE	RATING	DATE	MORE
...

ADD NEW MOVIE

ADD FROM TMDb

SEARCH

Diseñando

- ▶ ¿Cuántos componentes independientes tenemos?
- ▶ DISEÑEMOS.

Una persona por componente independiente. Negro verde.

Diseño arquitectónico

Patrón: MVC

1. Modelo
2. Vista
3. Controlador

Arquitectura tres niveles

1. Presentación
2. Aplicación
3. Persistencia

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. Aplicación

3. Persistencia

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. Aplicación

3. Persistencia

Con respecto a los componentes

- ▶ ¿Cuál es la relación entre los paquetes y los elementos MVC?
- ▶ ¿Cuáles hemos trabajado? ¿Cuáles no hemos trabajado?

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

P(V+C+I)

¿Concurrencia?

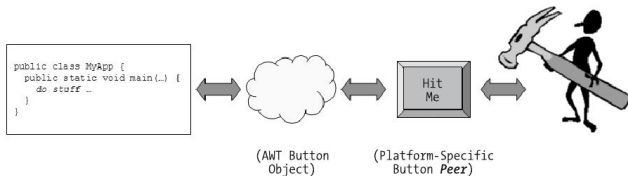
Aspectos finales

AWT-SWING

AWT. Abstract Window Toolkit API

Permite que una aplicación java automaticamente tome el *look* propio de la plataforma en la que se está ejecutando.

:) Más natural a los usuarios acostumbrados a la plataforma



Invoca realmente a elementos nativos de la plataforma

- :(El API debe escribirse para cada plataforma
- :(Los programas debían probarse en cada plataforma

heavyweight components

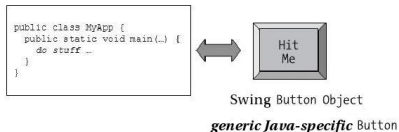
- ▶ `java.awt`
- ▶ `java.awt.event`

AWT-SWING

SWING

Permite que una aplicación java ofrezca un *look* generico propio de todas las aplicaciones java

:) Más natural a los usuarios acostumbrados a las aplicaciones java



Trata de no invocar a elementos nativos de la plataforma

- :) El API se escribe sólo una vez
- :) Los programas se pueden probar sólo en una plataforma

lightweight components

- ▶ `java.swing`
- ▶ `java.awt.event`

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

P(V+C+I)

¿Concurrencia?

Aspectos finales

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

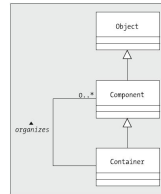
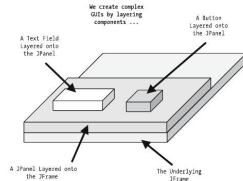
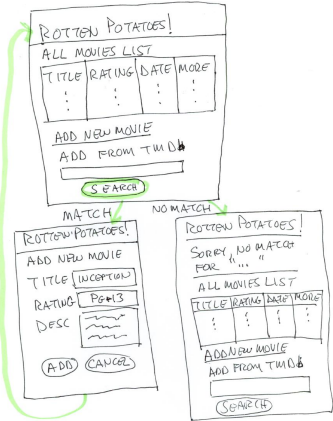
P(V+C+I)

¿Concurrencia?

Aspectos finales

MVC

FORMA: Vista - Negro



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

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

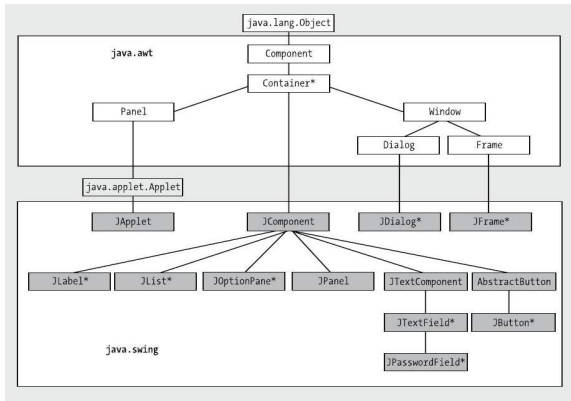
Arquitectura

P(V+C+I)

¿Concurrencia?

Aspectos finales

SWING



► Superior

`JApplet`, `JFrame`, `JDialog`, `JOptionPane`

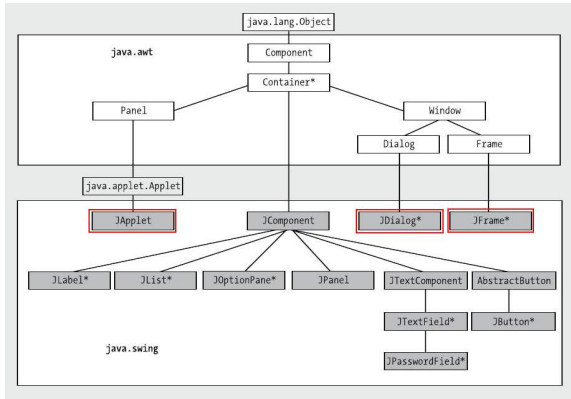
► Intermedio

`JPanel`, `JScrollPane`, `JTabbedPane`

► Atómico

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

SWING



- Superior
 - JApplet, JFrame, JDialog, JOptionPane

Componentes

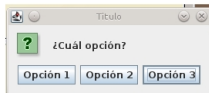
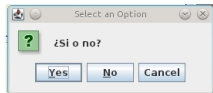
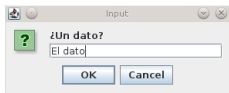
JFrame - JDialog



- ▶ 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
- ▶ **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, listo 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?", "Titulo",
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:

Posición:

Altura:

Peso:

Salario:

Opciones

Componentes

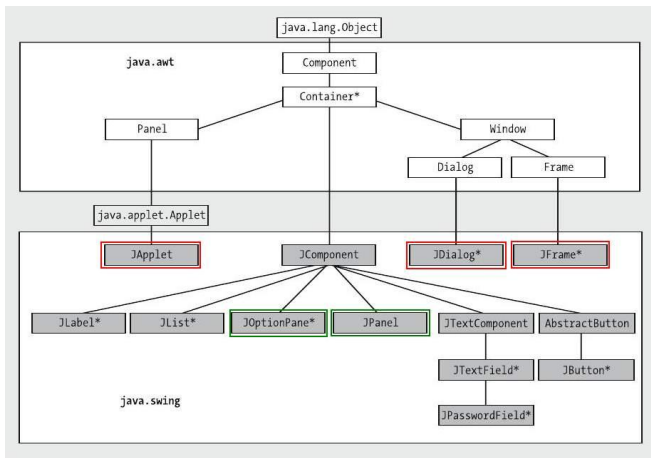
JFrame



```
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



► Intermedio

`JPanel`, `JScrollPane`, `JTabbedPane`

Componentes. Intermedio.

JPanel



The screenshot shows a Java Swing window titled "Mundial de Fútbol CUPi2". The window contains a header with a soccer ball icon and the title "Mundial de Fútbol CUPi2". Below the header, there are two main sections: "Equipos" and "Jugadores".

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

- 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.

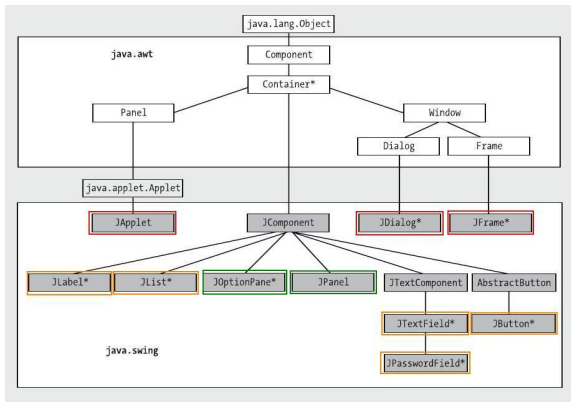


JFrame

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

Componentes. Básicos

SWING



► Atómicos

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

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

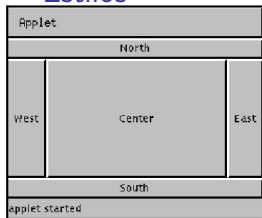
P(V+C+I)

¿Concurrencia?

Aspectos finales

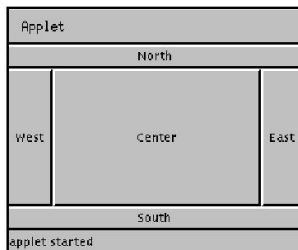
Estilos

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 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

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

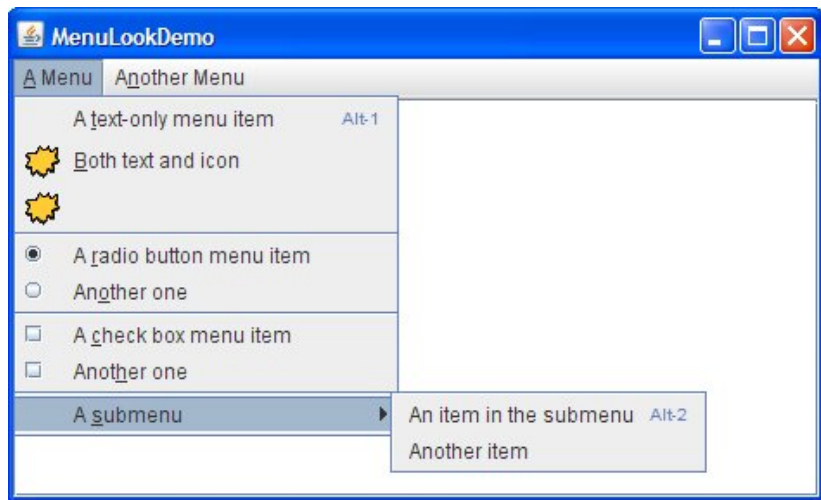
P(V+C+I)

¿Concurrencia?

Aspectos finales

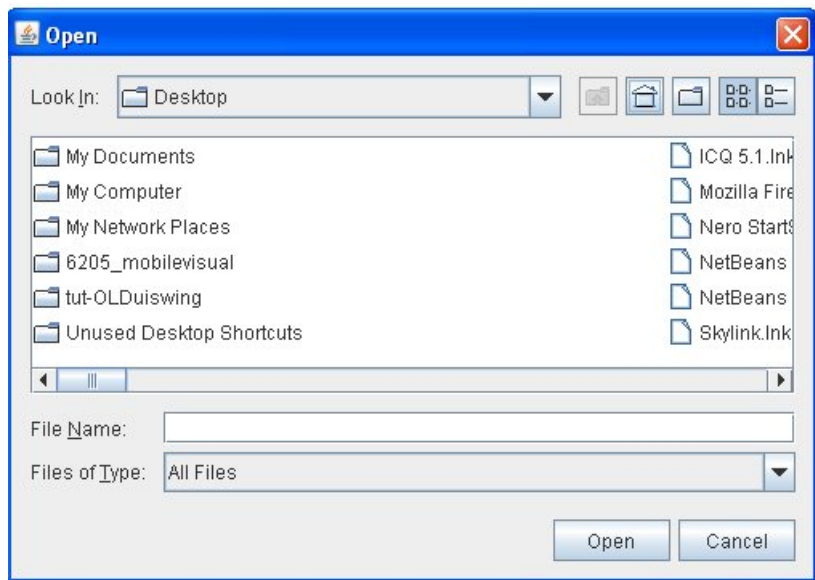
Menu

JMenuBar JMenu JMenuItem JMenuSeparator



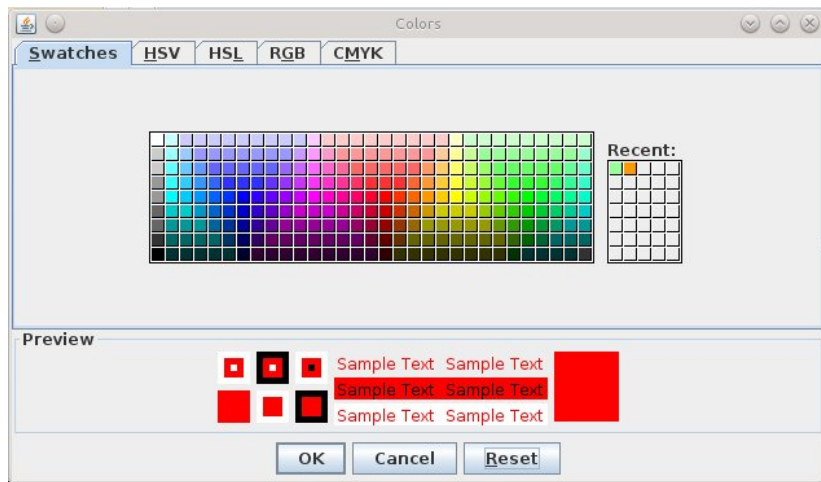
Control Archivos

JFileChooser



Control Colores

JColorChooser



Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

P(V+C+I)

¿Concurrencia?

Aspectos finales

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

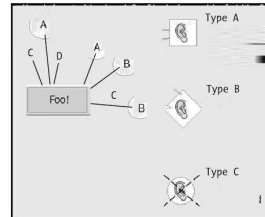
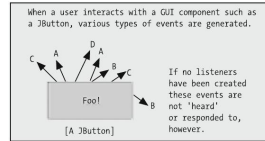
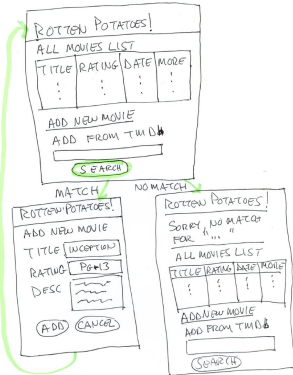
P(V+C+I)

¿Concurrencia?

Aspectos finales

MVC

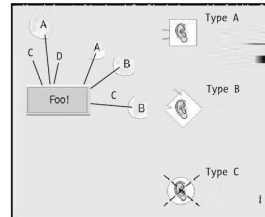
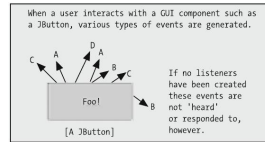
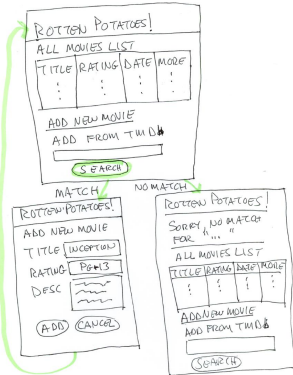
ACCIONES: Controlador - Verde



- Un componente puede generar un gran número de **eventos**.
- El programador decide cuáles eventos necesita atender: asigna un oyente y se programa una acción correspondiente

MVC

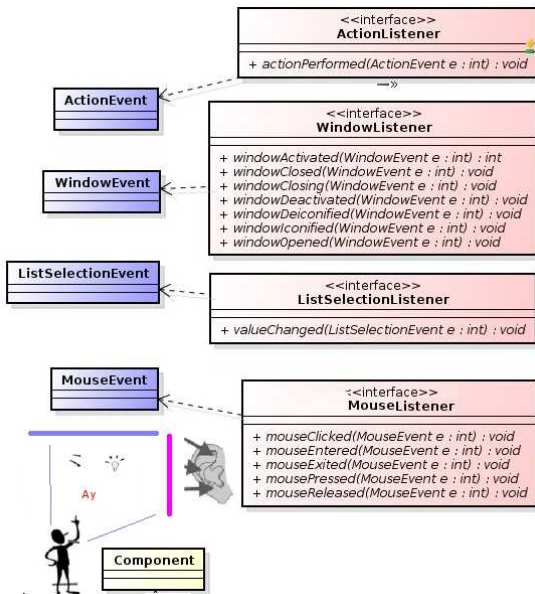
ACCIONES: Controlador - Verde



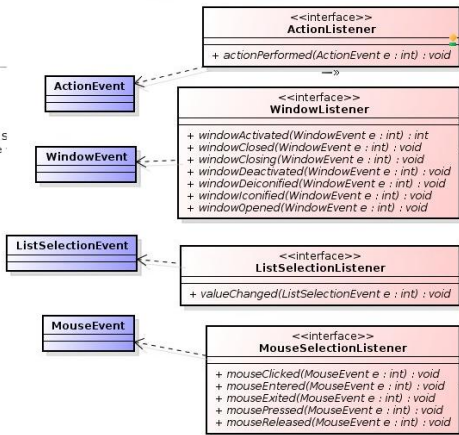
- Un componente puede generar un gran número de **eventos**.
Son objetos `java.awt.event.AWTEvent`
- El programador decide cuáles eventos necesita atender: asigna un oyente y se programa una acción correspondiente

Son objetos que implementan la *interface* necesaria para oír `java.awt.event.Listener`

Eventos



Eventos



¿Qué oyentes?

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

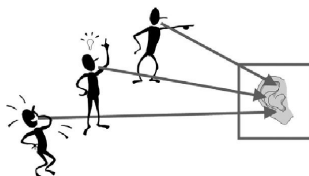
Arquitectura

P(V+C+I)

¿Concurrencia?

Aspectos finales

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) {  
  
    }  
}
```

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) {  
  
    }  
}
```

Registrando

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

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

```
mButton.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 ...  
    }  
}
```

Registrando

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

Programando acción (: ())

Manejo de eventos

Definiendo (:))

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

Registrando

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

Programando acción (:))

Manejo de eventos

Definiendo (:))

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

Clase anónima

\$1
\$2

Registrando

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

Programando acción (:))

Manejo de eventos

Definiendo (:))

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

Clase anónima

:\$1
:\$2

Registrando

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

Programando acción (:))

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

Mundial

Vista



Mundial de Fútbol Cupi2

Equipos

Equipo :

Director Técnico :



Jugadores



Nombre:

Edad:

Posición:

Altura:

Peso:

Salario:

Opciones

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( );
            }
        });
}
```

Mundial

Acciones. Programando acción

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

public void comboJugadoresAccion(){
    String nombreJugador = (String) comboJugadores.getSelectedItem();
    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.calculeValorNomina(equipoSeleccionado.getPais()));
    JOptionPane.showMessageDialog(this, resultado, "NOMINA", JOptionPane.INFORMATION_MESSAGE);
}
```

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

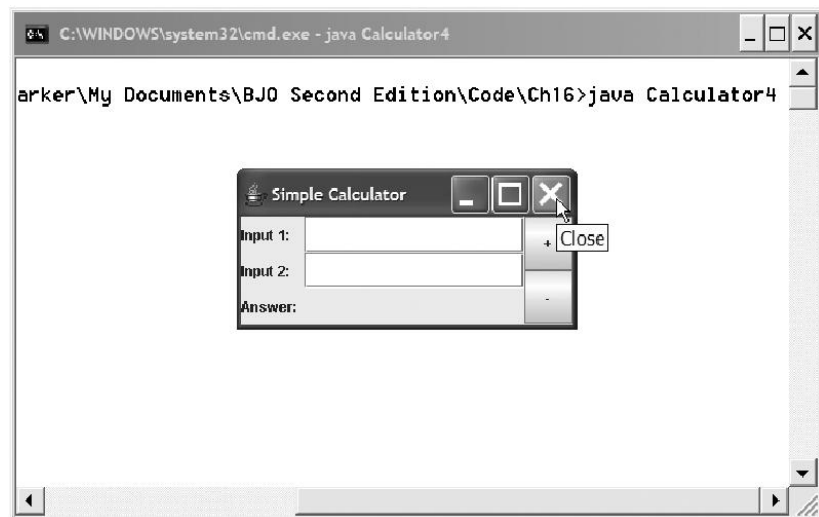
P(V+C+I)

¿Concurrencia?

Aspectos finales

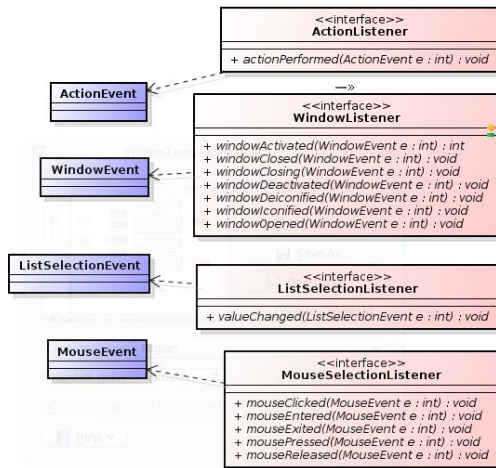
Manejo de eventos

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

Manejo de eventos

Un adaptador

```
public class WindowAdapter implements WindowListener {  
    // All SEVEN required methods are stubbed out!  
    public void windowClosing(WindowEvent e) { }  
    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) { }  
}
```

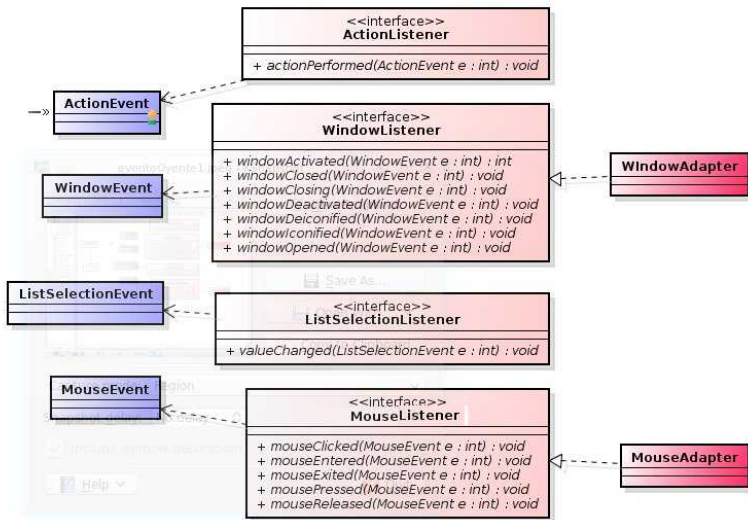
Manejo de eventos

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.  
}
```



Adaptadores



Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

$P(V+C+I)$

¿Concurrencia?

Aspectos finales

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

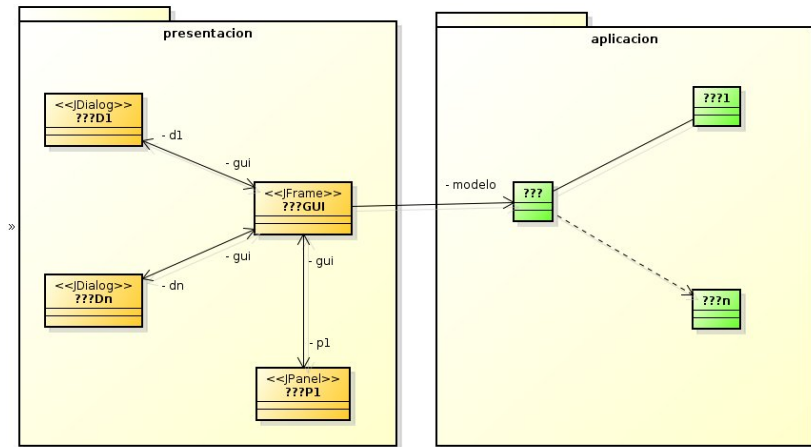
Arquitectura

$P(V+C+I)$

¿Concurrencia?

Aspectos finales

Mundial



Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

P(V+C+I)

¿Concurrencia?

Aspectos finales

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

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

$P(V+C+I)$

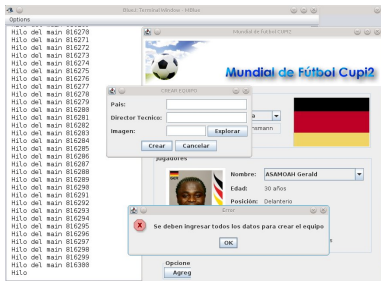
¿Concurrencia?

Aspectos finales

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 fútbol CUPi2

 **Mundial de Fútbol Cupi2**

Equipos

Equipo : 

Director :

Jugadores

  GER 

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();  
}
```

Agenda

Conceptos

Introducción

Diseño

AWT-SWING

Forma

Contexto

Componentes

Estilos

Otros componentes preprogramados

Acciones

Contexto

Esquema

Adaptadores

Aspectos finales

Arquitectura

P(V+C+I)

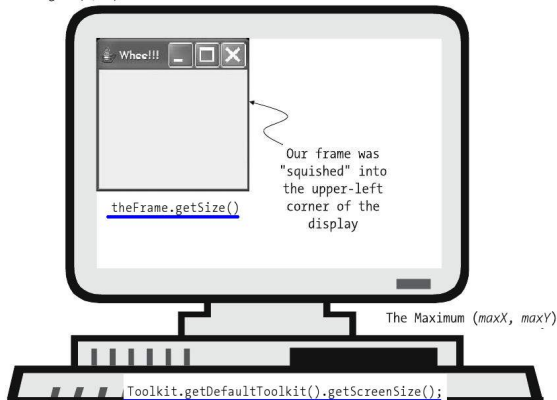
¿Concurrencia?

Aspectos finales

Componentes

Pantalla

The Origin (0, 0)



GUI

¿Estilos equipos?

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

Mundial



```
private void elementosPanelEquipos(){

    panelEquipos= new JPanel();
    panelEquipos.setBorder( new CompoundBorder(new EmptyBorder( 5, 5, 5, 5 ),
                                                new TitledBorder( "Equipos" )));
    panelEquipos.setLayout( new BorderLayout() );

    JPanel panelDatos = new JPanel();
    panelDatos.setBorder(new EmptyBorder(30,30,30,30));
    panelDatos.setLayout( new GridLayout(2,2,5,5) );
    etiquetaEquipo = new JLabel( "Equipo : " );
    comboEquipos = new JComboBox();
    comboEquipos.setEditable( false );
    etiquetaDirector = new JLabel( "Director Técnico : " );
    textoNombreDirector = new JTextField( "" );
    textoNombreDirector.setEditable(false);
    panelDatos.add(etiquetaEquipo );
    panelDatos.add(comboEquipos);
    panelDatos.add(etiquetaDirector);
    panelDatos.add(textoNombreDirector);

    etiquetaBandera = new JLabel();
    etiquetaBandera.setPreferredSize(new Dimension(175,100));

    panelEquipos.add( panelDatos,BorderLayout.CENTER);
    panelEquipos.add( etiquetaBandera,BorderLayout.EAST);
}
```