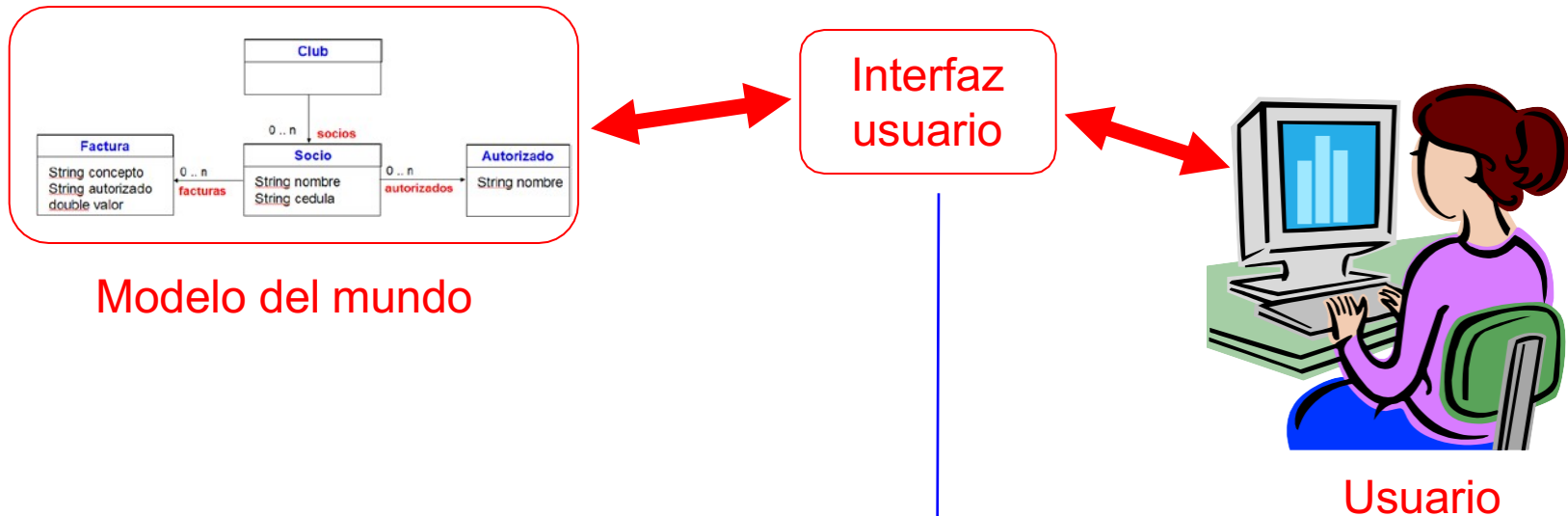


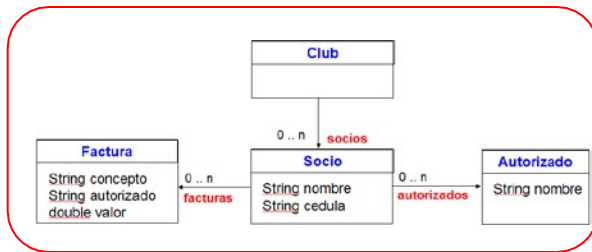
Semana 5: Modelo Vista Controlador (MVC)

Motivación



Único medio de comunicación entre el usuario
y el modelo del mundo

Motivación



Modelo del mundo

Interfaz
usuario



Usuario

A través de la interfaz:

- Ejecuta operaciones sobre el modelo del mundo.
- Ve el resultado de sus acciones.

Aspectos importantes



Diseño funcional y gráfico

- Colores
- Distribución de elementos gráficos (menús, botones, ...)

Arquitectura

- Estructura clara
- Fácil de mantener

Qué vamos a aprender en este nivel:

- ❶ Cómo proponer una **arquitectura** para un programa simple:
 - Repartir responsabilidades entre interfaz, mundo y pruebas.
- ❷ Cómo construir las **clases** que implementan una **interfaz** usuario
- ❸ Cómo **integrar** las clases de la interfaz con las clases del mundo

Caso de estudio: Calculador de Impuestos

El Calculador de Impuestos

- Funcionalidad
- Interfaz usuario
- Requerimientos Funcionales
- Arquitectura
 - Mundo
 - Interfaz

El Calculador de Impuestos - Funcionalidad

- Calcula el valor de los impuestos que una persona debe pagar por su carro
- Vehículo se caracteriza por:
 - Marca, línea, modelo y precio
- Cálculo de impuestos como porcentaje del precio del vehículo (% determinado por un rango)
- Hay tres tipos de descuento
 - Pronto pago, servicio público, traslado de cuenta

El Calculador de Impuestos – Interfaz usuario

Ventana Principal

3 Zonas de trabajo:

- ❶ Información del vehículo →
- ❷ Descuentos →
- ❸ Cálculo de impuestos + inicializar la aplicación →

The screenshot shows a Windows-style application window titled "Cálculo impuestos". It is divided into three main sections, each with a blue header:

- Datos del vehículo:** Contains four text input fields: "Marca" (filled with "Renault"), "Línea" (filled with "307"), "Modelo" (filled with "2005"), and "Valor" (filled with "\$ 71,800,000"). Below these fields is a "Buscar" button.
- Descuentos:** Contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All checkboxes are currently unchecked.
- Resultados:** Contains a "Total a pagar" label next to a text input field filled with "\$ 0". To the right of this field are two buttons: "Limpiar" and "Calcular". At the bottom of this section are two more buttons: "Opcion 1" and "Opcion 2".

El Calculador de Impuestos – Requerimientos Funcionales

| | |
|-----|--|
| R1 | Buscar el avalúo de un vehículo |
| R2 | Calcular el pago de impuesto de un vehículo |
| RNF | Requerimiento NO FUNCIONAL (inicializar la aplicación sin tener que volver a ejecutarla) |

The screenshot shows a Windows-style application window titled "Cálculo impuestos". It contains three main sections: "Datos del vehículo", "Descuentos", and "Resultados".

- Datos del vehículo:** Includes text boxes for "Marca" (Renault), "Línea" (307), "Modelo" (2005), and "Valor" (\$ 71,800,000). Below these is a "Buscar" button. A blue circle labeled "R1" has an arrow pointing to this button.
- Descuentos:** Includes three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público", all of which are currently unchecked.
- Resultados:** Includes a "Total a pagar" text box showing "\$ 0". To its right are two buttons: "Limpiar" and "Calcular". A blue circle labeled "R2" has an arrow pointing to the "Calcular" button. A red circle labeled "RNF" has an arrow pointing to the "Limpiar" button.
- At the bottom of the window are two buttons labeled "Opcion 1" and "Opcion 2".

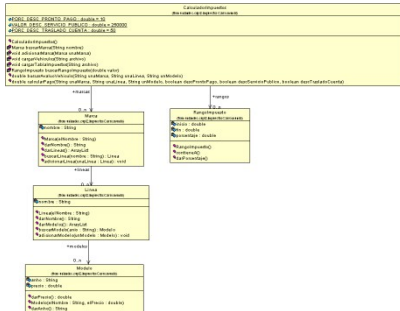
Tarea No. 1

- RF1: Buscar el avalúo de un vehículo
 - Resumen:
 - Dada la información del vehículo, presentar el valor de avalúo del mismo
 - Entradas:
 - Resultado:

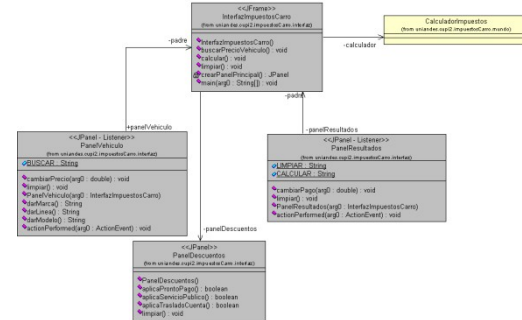
Tarea No. 1

- RF2:Calcular el pago de impuesto de un vehículo
 - **Resumen:**
 - Dada la información de un vehículo, y el conjunto de descuentos que le aplican, dar el valor a pagar por los impuestos del vehículo
 - **Entradas:**
 - **Resultado:**

Arquitectura

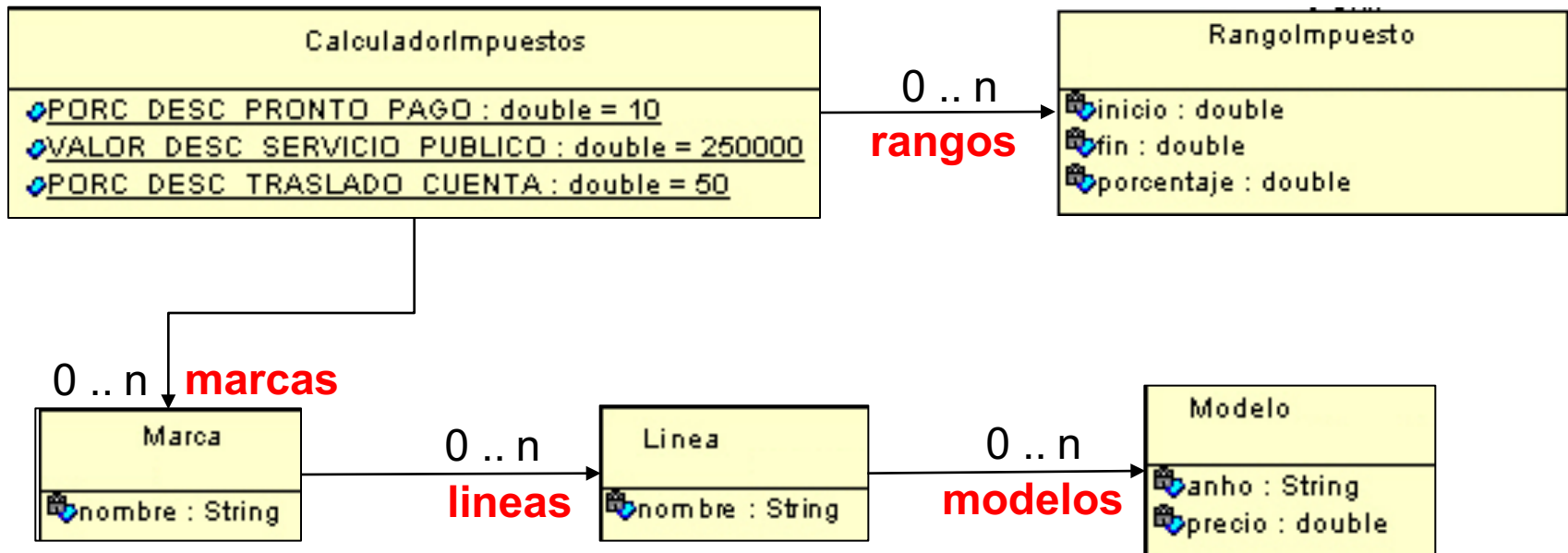


Mundo



Interfaz

El Calculador de Impuestos (Mundo)



Tarea No. 2

- **CalculadorImpuestos**
 - Constantes
 - Asociaciones
- **Marca**
 - Atributos
 - Asociaciones
- **Linea**
 - Atributos
 - Asociaciones
- **Modelo**
 - Atributos
- **RangoImpuesto**
 - Atributos

Métodos de la clase CalculadorImpuestos (pag. 3)

buscarAvaluoVehiculo

```
/**
```

- * Retorna el valor de avalúo de un vehiculo de la marca, línea y modelo dado.
 - * @param unaMarca - marca del vehiculo
 - * @param unaLinea - linea del vehiculo
 - * @param unModelo - modelo del vehiculo
 - * @return precio de avalúo del vehiculo
 - * @throws Exception si no encuentra la marca o la linea o el modelo registrados
- ```
*/
```

```
public double buscarAvaluoVehiculo(String unaMarca, String unaLinea, String unModelo)
throws Exception
```

# Métodos de la clase

## CalculadorImpuestos (pag. 3)

### calcularPago

```
/**
 * Calcular el pago de impuesto que debe hacer un vehículo de un modelo dado. Si no
 * encuentra un rango para el modelo devuelve 0
 * @param unaMarca - marca del vehiculo
 * @param unaLinea - linea del vehiculo
 * @param unModelo - modelo del vehiculo
 * @param descProntoPago - indica si aplica el descuento por pronto pago
 * @param descServicioPublico - indica si aplica el descuento por servicio público
 * @param descTrasladoCuenta - indica si aplica el descuento por traslado de cuenta
 * @return valor a pagar
 * @throws Exception si no encuentra el vehiculo dado por la marca, la linea y el modelo
 */
public double calcularPago(String unaMarca, String unaLinea, String unModelo,
boolean descProntoPago, boolean descServicioPublico, boolean descTrasladoCuenta
) throws Exception
```

# Construcción de interfaces gráficas

# El Calculador de Impuestos – Interfaz usuario

## Ventana Principal

### 3 Zonas de trabajo:

- ❶ Información del vehículo →
- ❷ Descuentos →
- ❸ Cálculo de impuestos + inicializar la aplicación →

The screenshot shows a Windows-style application window titled "Cálculo impuestos". It is divided into three main sections, each with a blue header:

- Datos del vehículo:** Contains four text input fields: "Marca" (filled with "Renault"), "Línea" (filled with "307"), "Modelo" (filled with "2005"), and "Valor" (filled with "\$ 71,800,000"). Below these fields is a "Buscar" button.
- Descuentos:** Contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All checkboxes are currently unchecked.
- Resultados:** Contains a "Total a pagar" label next to a text field showing "\$ 0". To the right of this field are two buttons: "Limpiar" and "Calcular". At the bottom of this section are two more buttons: "Opcion 1" and "Opcion 2".

# El Calculador de Impuestos – Interfaz usuario

## InterfazImpuestosCarro (JFrame)

3 Zonas de trabajo:

❶ PanelVehiculo (JPanel)

❷ PanelDescuentos (JPanel)

❸ PanelResultados (JPanel)

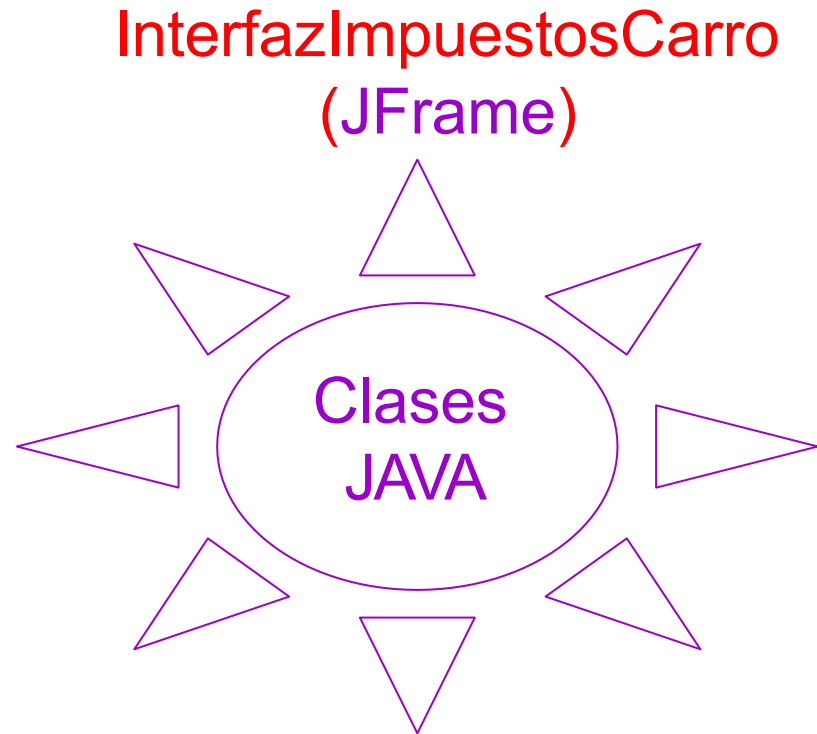
The screenshot shows a Java Swing window titled "Cálculo impuestos". It is divided into three main sections:

- Datos del vehículo:** Contains four text input fields for "Marca" (filled with "Renault"), "Línea" (filled with "307"), "Modelo" (filled with "2005"), and "Valor" (filled with "\$ 71,800,000"). Below these fields is a "Buscar" button.
- Descuentos:** Contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All are currently unchecked.
- Resultados:** Contains a "Total a pagar" label next to a text field showing "\$ 0". To the right of this field are two buttons: "Limpiar" and "Calcular". At the bottom of the section are two more buttons: "Opcion 1" and "Opcion 2".

Blue arrows from the text on the left point to the corresponding sections in the window: one arrow points to the "Datos del vehículo" section, another points to the "Descuentos" section, and a third points to the "Resultados" section.

# El Calculador de Impuestos – Interfaz usuario

- ❶ PanelVehiculo (JPanel)
- ❷ PanelDescuentos (JPanel)
- ❸ PanelResultados (JPanel)



# Entidades



## Mundo del Problema

- Estudiante
- Tienda
- Banco
- Recurso
- Avion
- ...



## Mundo Gráfico y de Interacción

- Ventana (JFrame)
- Panel (JPanel)
- Botón (JButton)
- Zona de texto (JTextField)
- Etiqueta (JLabel)
- Caja de chequeo (JCheckBox)

# Entidades del mundo gráfico y de interacción

Ventana

**Cálculo impuestos**

**Datos del vehículo**

Marca: Renault

Línea: 307

Modelo: 2005

Valor: \$ 71,800,000

Buscar

**Descuentos**

☐ Pronto pago ☐ Traslado de cuenta

☐ Servicio público

**Resultados**

Total a pagar: \$ 0

Limpiar

Calcular

Opcion 1 Opcion 2



# Entidades del mundo gráfico y de interacción

Panel

**Cálculo impuestos**

**Datos del vehículo**

|        |               |
|--------|---------------|
| Marca  | Renault       |
| Línea  | 307           |
| Modelo | 2005          |
| Valor  | \$ 71,800,000 |

Buscar

**Descuentos**

☐ Pronto pago ☐ Traslado de cuenta

☐ Servicio público

**Resultados**

|               |      |          |
|---------------|------|----------|
| Total a pagar | \$ 0 | Limpiar  |
|               |      | Calcular |

Opcion 1 Opcion 2

# Entidades del mundo gráfico y de interacción

Zona de  
texto

The image shows a software window titled "Cálculo impuestos" (Tax Calculation). It contains several sections for data entry and calculation. Red arrows originate from the text "Zona de texto" (Text Zone) and point to the input fields for "Marca", "Línea", "Modelo", "Valor", and "Total a pagar".

| Datos del vehículo                    |               |
|---------------------------------------|---------------|
| Marca                                 | Renault       |
| Línea                                 | 307           |
| Modelo                                | 2005          |
| Valor                                 | \$ 71,800,000 |
| <input type="button" value="Buscar"/> |               |

| Descuentos                                |                                             |
|-------------------------------------------|---------------------------------------------|
| <input type="checkbox"/> Pronto pago      | <input type="checkbox"/> Traslado de cuenta |
| <input type="checkbox"/> Servicio público |                                             |

| Resultados                              |                                         |
|-----------------------------------------|-----------------------------------------|
| Total a pagar                           | \$ 0                                    |
| <input type="button" value="Limpiar"/>  |                                         |
| <input type="button" value="Calcular"/> |                                         |
| <input type="button" value="Opcion 1"/> | <input type="button" value="Opcion 2"/> |

# Entidades del mundo gráfico y de interacción

Caja de  
chequeo

The image shows a software window titled "Cálculo impuestos" (Tax Calculation). It contains three main sections: "Datos del vehículo" (Vehicle Data), "Descuentos" (Deductions), and "Resultados" (Results). In the "Descuentos" section, the "Servicio público" (Public service) checkbox is checked, and a red arrow points from the "Caja de chequeo" (Checklist) text to this checkbox. The "Total a pagar" (Total to pay) field shows "\$ 0".

| Datos del vehículo |               |
|--------------------|---------------|
| Marca              | Renault       |
| Línea              | 307           |
| Modelo             | 2005          |
| Valor              | \$ 71,800,000 |

Buscar

| Descuentos                                           |                                             |
|------------------------------------------------------|---------------------------------------------|
| <input type="checkbox"/> Pronto pago                 | <input type="checkbox"/> Traslado de cuenta |
| <input checked="" type="checkbox"/> Servicio público |                                             |

| Resultados    |      |
|---------------|------|
| Total a pagar | \$ 0 |

Limpiar Calcular

Opcion 1 Opcion 2

# Entidades del mundo gráfico y de interacción

Etiqueta

The image shows a screenshot of a software window titled "Cálculo impuestos" (Tax Calculation). The window is divided into several sections: "Datos del vehículo" (Vehicle Data), "Descuentos" (Discounts), and "Resultados" (Results). Red arrows originate from the word "Etiqueta" (Label) and point to the following elements: the "Marca" (Brand) label, the "Línea" (Line) label, the "Modelo" (Model) label, the "Valor" (Value) label, and the "Total a pagar" (Total to pay) label. The "Datos del vehículo" section contains input fields for "Marca" (Renault), "Línea" (307), "Modelo" (2005), and "Valor" (\$ 71,800,000), along with a "Buscar" (Search) button. The "Descuentos" section has three checkboxes: "Pronto pago" (Paid in advance), "Traslado de cuenta" (Account transfer), and "Servicio público" (Public service). The "Resultados" section displays "Total a pagar" as \$ 0, with "Limpiar" (Clear) and "Calcular" (Calculate) buttons. At the bottom, there are "Opcion 1" and "Opcion 2" buttons.

| Datos del vehículo                    |               |
|---------------------------------------|---------------|
| Marca                                 | Renault       |
| Línea                                 | 307           |
| Modelo                                | 2005          |
| Valor                                 | \$ 71,800,000 |
| <input type="button" value="Buscar"/> |               |

| Descuentos                                |                                             |
|-------------------------------------------|---------------------------------------------|
| <input type="checkbox"/> Pronto pago      | <input type="checkbox"/> Traslado de cuenta |
| <input type="checkbox"/> Servicio público |                                             |

| Resultados                              |                                         |
|-----------------------------------------|-----------------------------------------|
| Total a pagar                           | \$ 0                                    |
| <input type="button" value="Limpiar"/>  |                                         |
| <input type="button" value="Calcular"/> |                                         |
| <input type="button" value="Opcion 1"/> | <input type="button" value="Opcion 2"/> |

# Entidades del mundo gráfico y de interacción

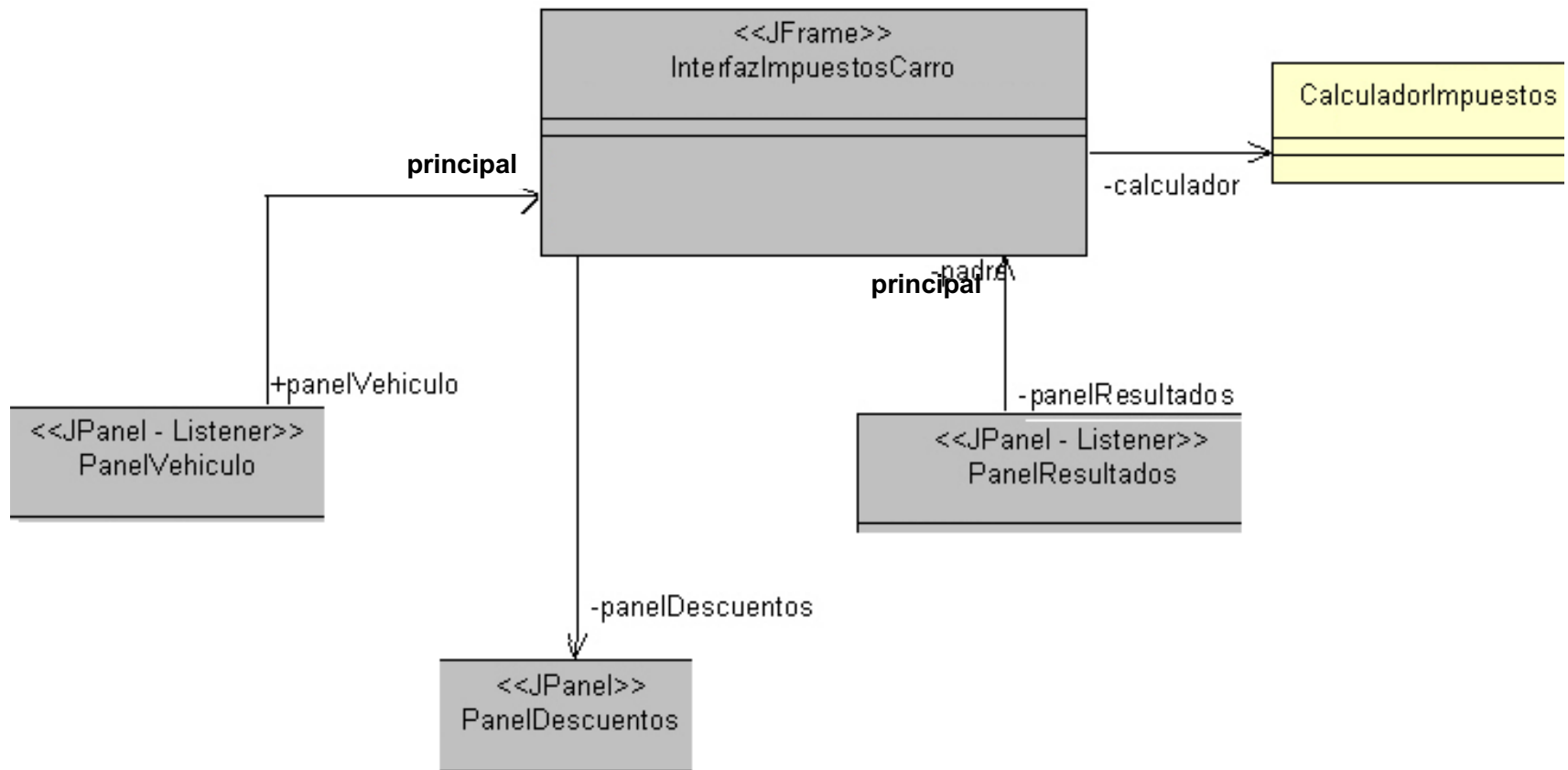
Boton

The image shows a screenshot of a software window titled "Cálculo impuestos". The window is divided into three main sections: "Datos del vehículo", "Descuentos", and "Resultados".

- Datos del vehículo:** Contains four text input fields: "Marca" (filled with "Renault"), "Línea" (filled with "307"), "Modelo" (filled with "2005"), and "Valor" (filled with "\$ 71,800,000"). Below these fields is a "Buscar" button.
- Descuentos:** Contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All are currently unchecked.
- Resultados:** Contains a "Total a pagar" label next to a text field showing "\$ 0". Below this are three buttons: "Limpiar", "Calcular", and "Opcion 1". To the right of "Opcion 1" is another button labeled "Opcion 2".

Four red arrows originate from a single point on the left, labeled "Boton", and point to the "Buscar" button, the "Calcular" button, and the "Opcion 1" and "Opcion 2" buttons.

# El Calculador de Impuestos (Interfaz)



# El Calculador de Impuestos – Interfaz usuario

## InterfazImpuestosCarro (JFrame)

3 Zonas de trabajo:

① PanelVehiculo (JPanel)

② PanelDescuentos (JPanel)

③ PanelResultados (JPanel)

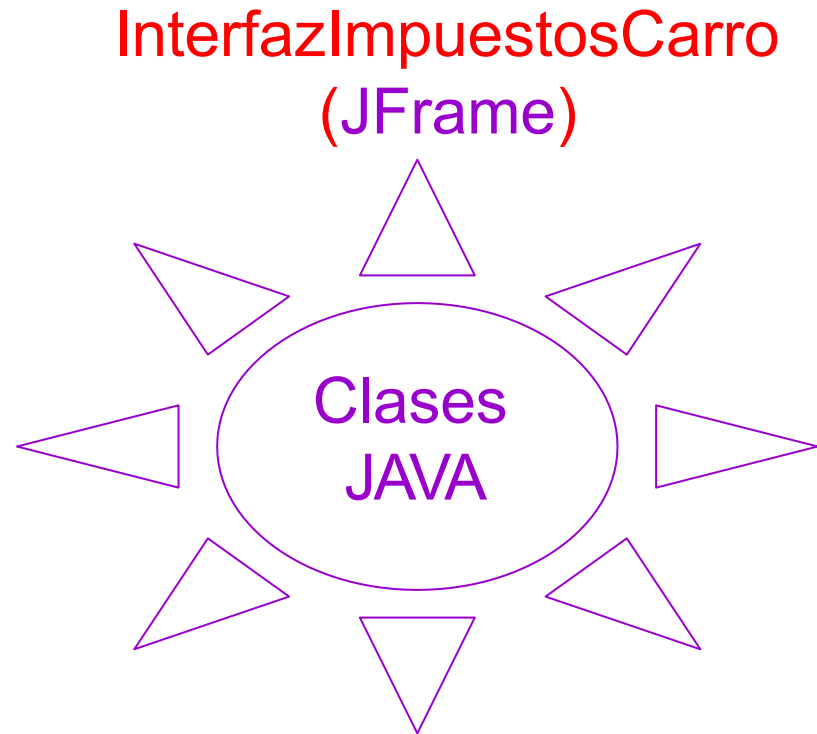
The screenshot shows a Java Swing window titled "Cálculo impuestos". It is divided into three main sections, each with a blue header:

- Datos del vehículo:** Contains four text input fields for "Marca" (filled with "Renault"), "Línea" (filled with "307"), "Modelo" (filled with "2005"), and "Valor" (filled with "\$ 71,800,000"). Below these fields is a "Buscar" button.
- Descuentos:** Contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All are currently unchecked.
- Resultados:** Contains a "Total a pagar" label next to a text field showing "\$ 0". To the right of this field are two buttons: "Limpiar" and "Calcular". At the bottom of this section are two more buttons: "Opcion 1" and "Opcion 2".

Blue arrows from the text on the left point to each of these three sections in the window.

# El Calculador de Impuestos – Interfaz usuario

- ❶ PanelVehiculo (JPanel)
- ❷ PanelDescuentos (JPanel)
- ❸ PanelResultados (JPanel)





# El Calculador de Impuestos – Interfaz usuario

## InterfazImpuestosCarro (JFrame)

- ❶ PanelVehiculo (JPanel)
- ❷ PanelDescuentos (JPanel)
- ❸ PanelResultados (JPanel)

- Se encuentran en una biblioteca gráfica (swing)
- Paquete: java.swing
- Deber ser importado

# Elementos gráficos estructurales

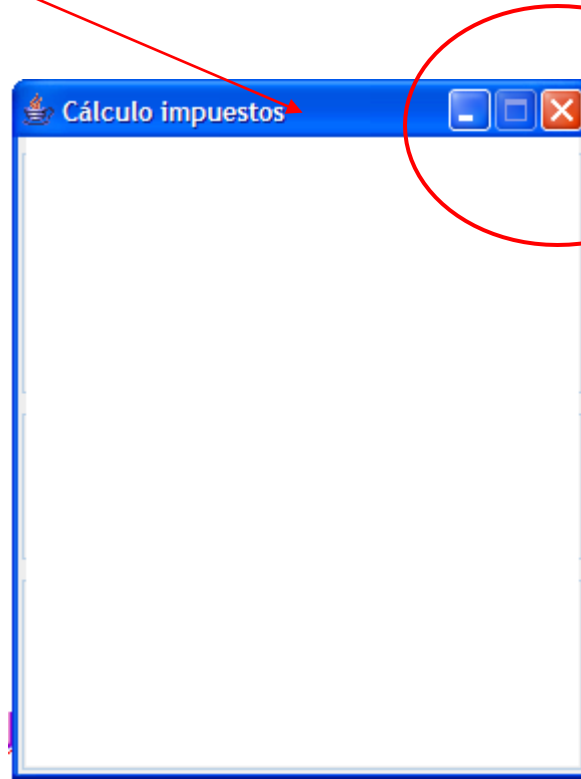
La Ventana Principal

# La Ventana Principal

- Contiene TODOS los elementos de visualización e interacción (con los que el usuario va a utilizar el programa)
- UNICA FUNCION:
  - Servir como marco para los demás elementos de la interfaz (AGRUPA)
- Es un contenedor gráfico

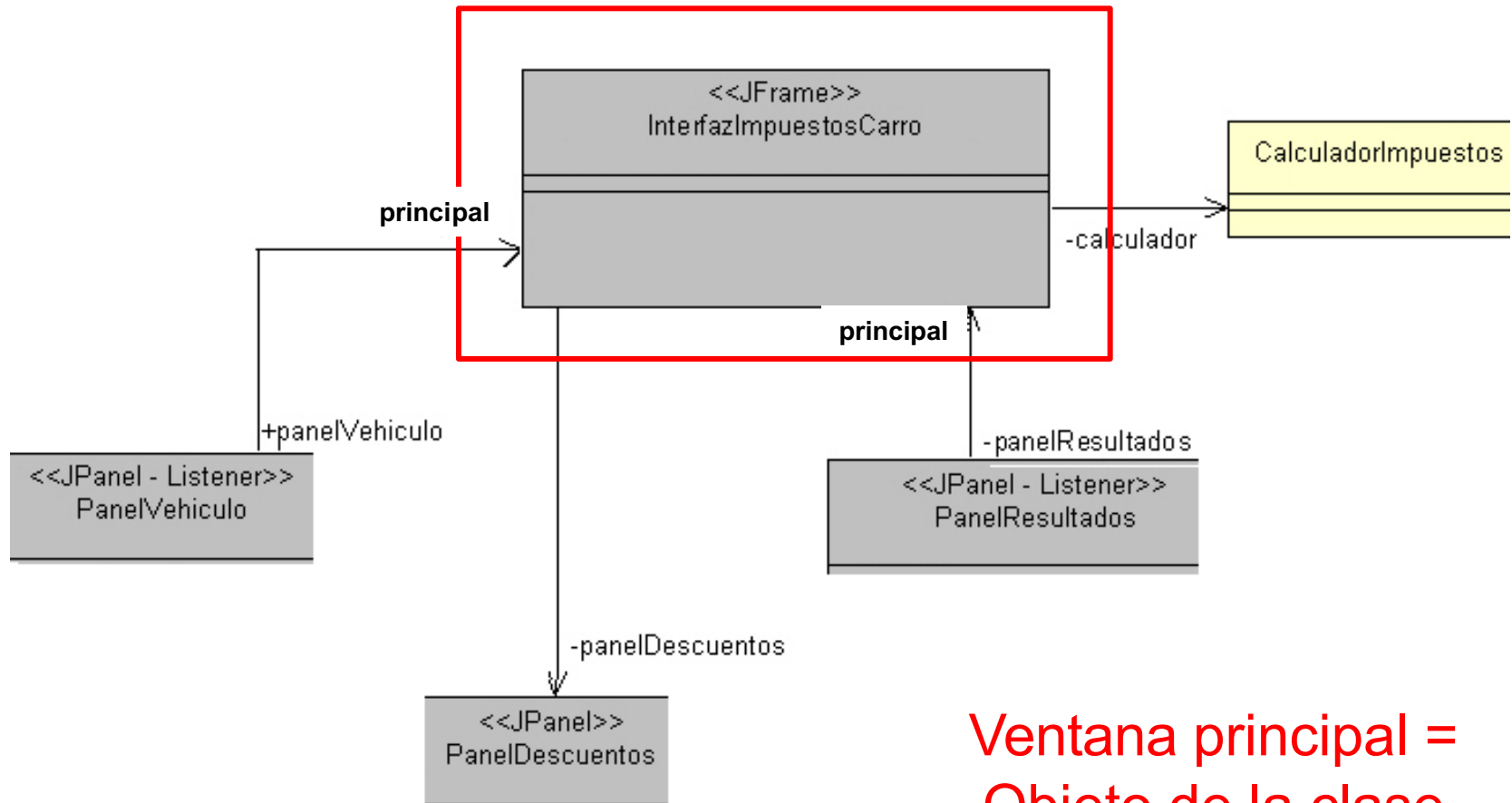
# La Ventana Principal

Título



Controles para  
cerrar el  
programa

# La Ventana Principal



**Ventana principal =  
Objeto de la clase  
InterfazImpuestosCarro**

# Clase InterfazImpuestosCarro

- Como cualquier clase:
  - Está declarada en su propio archivo:  
InterfazImpuestosCarro.java (pag. 8)
  - Sigue las mismas reglas que cualquier clase del mundo
- Diferencia:
  - Está declarada en otro paquete:  
uniandes.cupi2.impuestosCarro.interfaz (ver en Eclipse)

## Clase InterfazImpuestosCarro

```
package uniandes.cupi2.impuestosCarro.interfaz;
import java.awt.*;
import javax.swing.*;
import uniandes.cupi2.impuestosCarro.mundo.*;
/**
 * Interfaz de cálculo de impuestos de carros
 */
public class InterfazImpuestosCarro extends
 JFrame
{
}
```



# Clase InterfazImpuestosCarro

```
package uniandes.cupi2.impuestosCarro.interfaz;
```

La clase se declara dentro del paquete de las clases de la interfaz usuario

# Clase InterfazImpuestosCarro

```
package uniandes.cupi2.impuestosCarro.interfaz;
import java.awt.*;
import javax.swing.*;
```

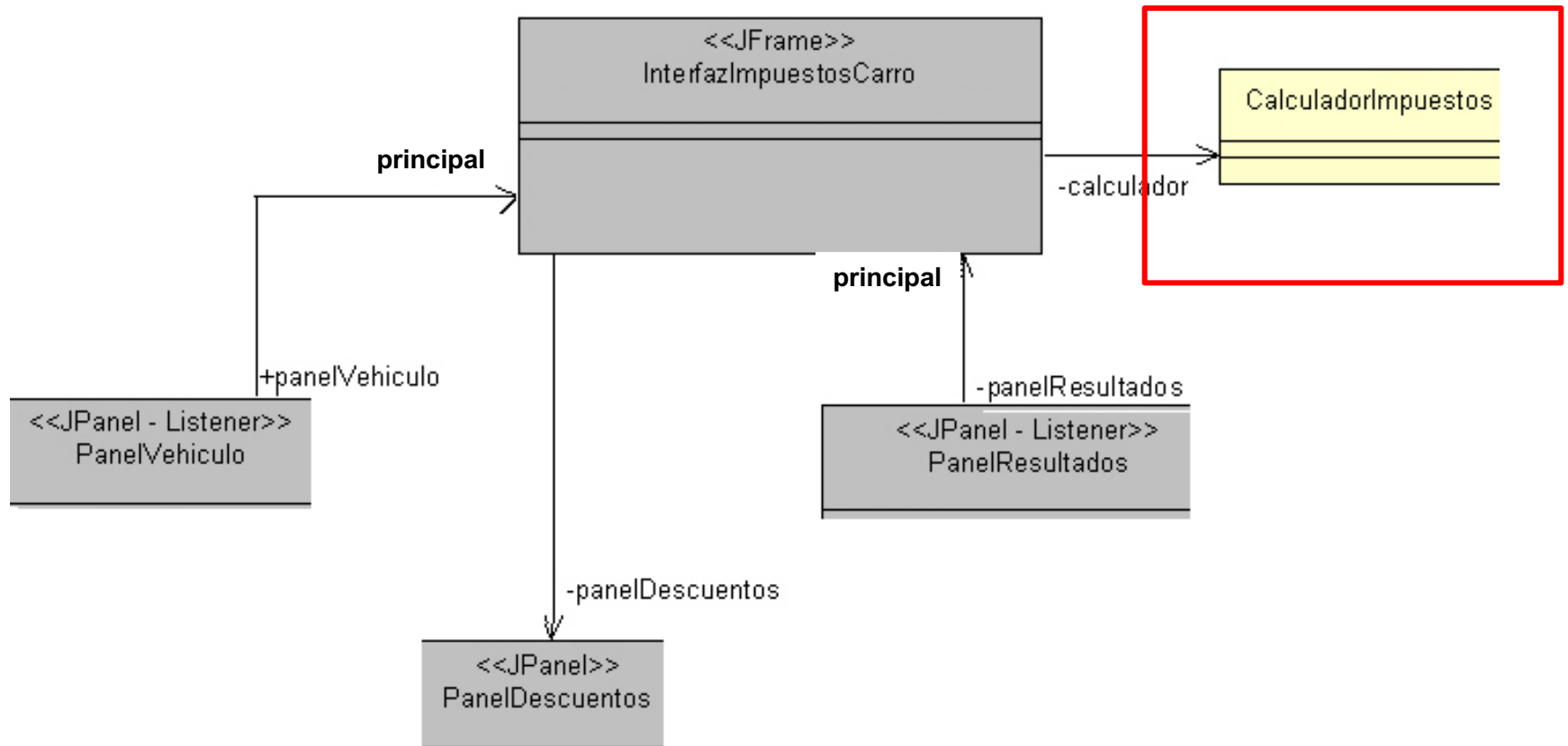
Se importan las clases de los dos paquetes mostrados (swing y awt)

# Clase InterfazImpuestosCarro

```
package uniandes.cupi2.impuestosCarro.interfaz;
import java.awt.*;
import javax.swing.*;
import uniandes.cupi2.impuestosCarro.mundo.*;
```

Se importan las clases del modelo del mundo

# Clase InterfazImpuestosCarro



# Clase InterfazImpuestosCarro

```
package uniandes.cupi2.impuestosCarro.interfaz;
import java.awt.*;
```

```
import javax.swing.*;
```

```
import uniandes.cupi2.impuestosCarro.mundo.*;
```

```
/**
```

```
 * Interfaz de cálculo de impuestos de carros
```

```
 */
```

```
public class InterfazImpuestosCarro extends JFrame
```

```
{
 La clase se declara con la misma sintáxis de las clases del
 modelo del mundo
}
```

# Clase InterfazImpuestosCarro

```
package uniandes.cupi2.impuestosCarro.interfaz;
import java.awt.*;
```

```
import javax.swing.*;
```

```
import uniandes.cupi2.impuestosCarro.mundo.*;
```

```
/**
```

```
 * Interfaz de cálculo de impuestos de carros
```

```
 */
```

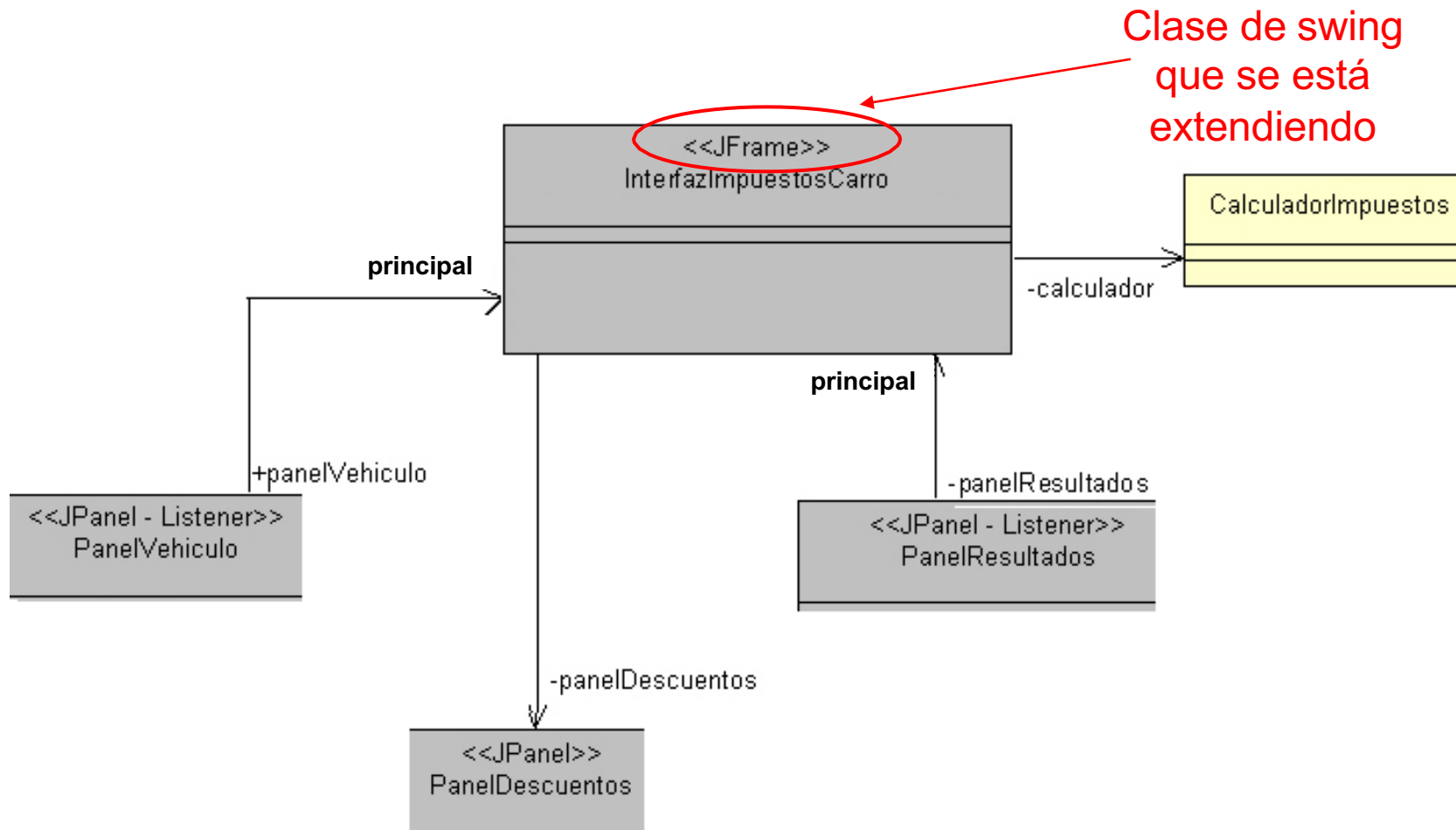
```
public class InterfazImpuestosCarro extends JFrame
```

```
{
```

Se agrega extends JFrame para indicar que es una ventana

```
}
```

# Clase InterfazImpuestosCarro



# Clase InterfazImpuestosCarro

Cómo cambiar el  
estado (apariencia)  
de la ventana?



# Clase InterfazImpuestosCarro

- Cómo cambiar el estado (apariencia) de la ventana?

- R// Con los métodos de la clase JFrame



Cambiar el  
título de la  
ventana

Cambiar el  
tamaño de la  
ventana

Agregar  
componentes  
gráficos

## Métodos de JFrame

- setSize ( ancho, alto )
- setResizable ( true/false )
- setTitle ( titulo )
- setDefaultCloseOperation ( EXIT\_ON\_CLOSE )
- setVisible ( true/false )
- setLayout( layout )
- add ( componente )

Configuración de la ventana en el método  
Constructor

```
public InterfazImpuestosCarro()
{
 setTitle("Cálculo impuestos");
 setSize(290, 350);
 setResizable(false);
 setDefaultCloseOperation(EXIT_ON_CLOSE);
 setLayout(new BorderLayout());
}
```

## Distribuidor gráfico de elementos (layout)

- Se encarga de distribuir los elementos (NO tenemos que hacerlo)
- Sólo tenemos que asociar a la ventana un **objeto** de este tipo (**clase**) que se encargue de hacerlo
- Java (swing) tiene varios distribuidores gráficos (**varias clases**). Vamos a ver dos:
  - BorderLayout
  - GridLayout

Configuración de la ventana en el método  
Constructor

```
public InterfazImpuestosCarro()
{
 setTitle("Cálculo impuestos");
 setSize(290, 350);
 setResizable(false);
 setDefaultCloseOperation(EXIT_ON_CLOSE);
 setLayout(new BorderLayout());
}
```

# Divisiones y Paneles

# El Calculador de Impuestos – Interfaz usuario

## InterfazImpuestosCarro (JFrame)

3 Zonas de trabajo:

❶ PanelVehiculo (JPanel)

❷ PanelDescuentos (JPanel)

❸ PanelResultados (JPanel)

The screenshot shows a Java Swing window titled "Cálculo impuestos". It is divided into three main sections:

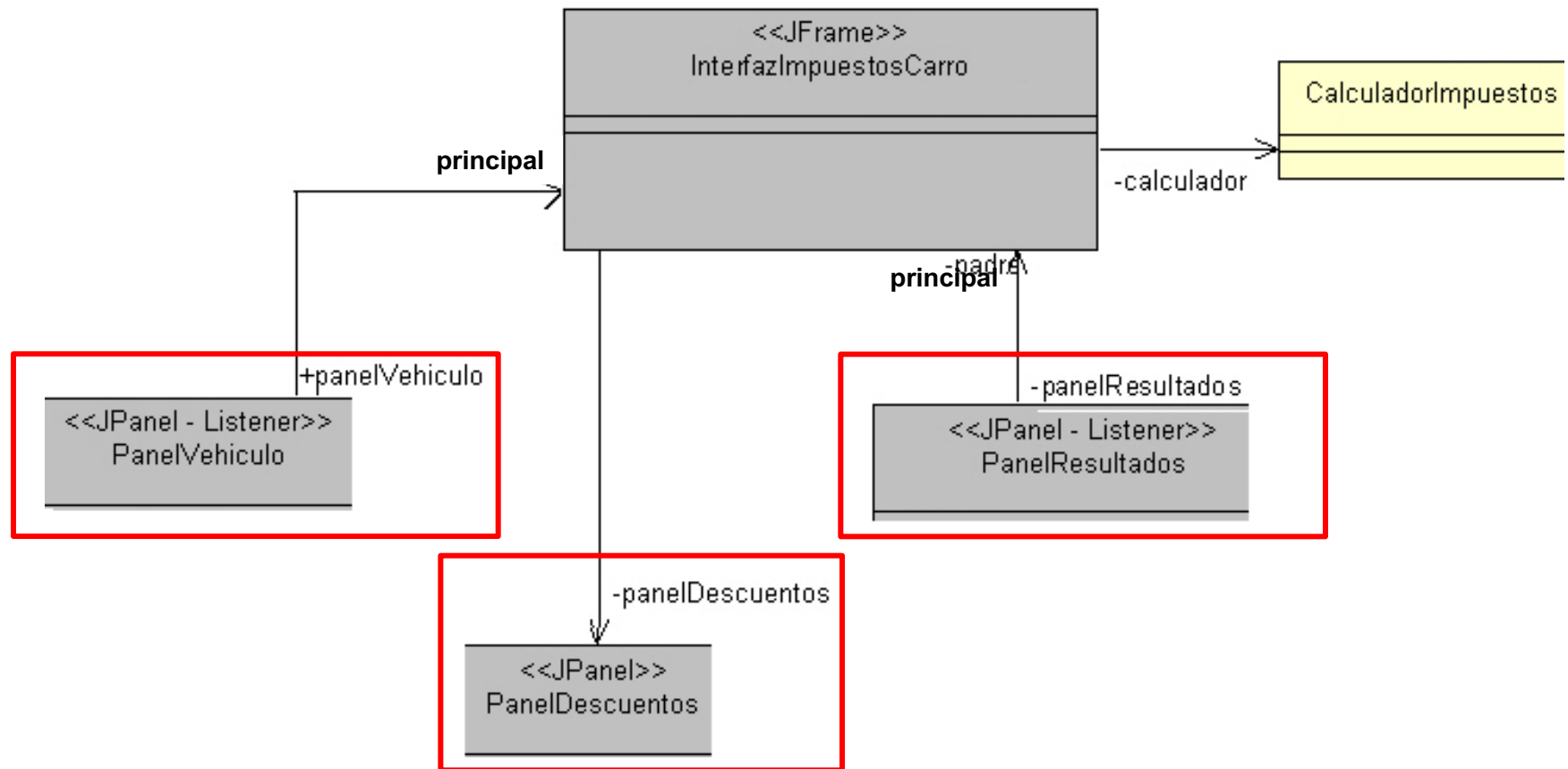
- Datos del vehículo:** Contains four text input fields for "Marca" (filled with "Renault"), "Línea" (filled with "307"), "Modelo" (filled with "2005"), and "Valor" (filled with "\$ 71,800,000"). Below these fields is a "Buscar" button.
- Descuentos:** Contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All are currently unchecked.
- Resultados:** Contains a "Total a pagar" label next to a text field showing "\$ 0". To the right of this field are two buttons: "Limpiar" and "Calcular". At the bottom of this section are two more buttons: "Opcion 1" and "Opcion 2".

# Un Panel

- Se encarga de **agrupar** elementos gráficos por contenido y uso
- **Facilita** al usuario su localización y su uso
- Cada **panel** se implementa como una **clase aparte** en el modelo
  - PanelVehiculo
  - PanelDescuentos
  - PanelResultados



# El Calculador de Impuestos (Interfaz)



# Clase InterfazImpuestosCarro

## InterfazImpuestos

```
/**
 * Interfaz de cálculo de impuestos de carros
 */

public class InterfazImpuestosCarro extends JFrame

//-----
// Atributos
//-----

/** Calculador de impuestos */
private CalculadorImpuestos calculador;

//-----
// Atributos de la interfaz
//-----

/** Panel de vehiculos */
private PanelVehiculo panelVehiculo;

/** Panel de descuentos */
private PanelDescuentos panelDescuentos;

/** Panel de resultados */
private PanelResultados panelResultados;
```

The screenshot shows a Java Swing window titled "Cálculo impuestos". It is divided into three main sections:

- Datos del vehículo:** Contains four text input fields for "Marca" (filled with "Renault"), "Línea" (filled with "307"), "Modelo" (filled with "2005"), and "Valor" (filled with "\$ 71,800,000"). A "Buscar" button is located below these fields.
- Descuentos:** Contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All are currently unchecked.
- Resultados:** Contains a "Total a pagar" label next to a text field showing "\$ 0". To the right of this field are two buttons: "Limpiar" and "Calcular". At the bottom of the section are two more buttons: "Opcion 1" and "Opcion 2".

Arrows from the code on the left point to the corresponding UI elements: from `panelVehiculo` to the vehicle data section, from `panelDescuentos` to the discounts section, and from `panelResultados` to the results section.

# Creación de los paneles en el método Constructor

```
public class InterfazImpuestosCarro extends JFrame
{
```

```
 private PanelVehiculo panelVehiculo;
 private PanelDescuentos panelDescuentos;
 private PanelResultados panelResultados;
```



Se declara un atributo por cada una de las divisiones

```
 public InterfazImpuestosCarro() throws Exception
 {
 setLayout(new BorderLayout());
```

```
}
```

# Creación de los paneles en el método Constructor

```
public class InterfazImpuestosCarro extends JFrame
{
 private PanelVehiculo panelVehiculo;
 private PanelDescuentos panelDescuentos;
 private PanelResultados panelResultados;
```

```
 public InterfazImpuestosCarro() throws Exception
 {
```

```
 setLayout(new BorderLayout());
```



Se asocia a la  
ventana un  
distribuidor gráfico

```
}
```

# Creación de los paneles en el método Constructor

```
public class InterfazImpuestosCarro extends JFrame
{
 private PanelVehiculo panelVehiculo;
 private PanelDescuentos panelDescuentos;
 private PanelResultados panelResultados;

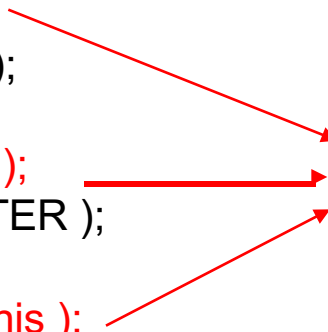
 public InterfazImpuestosCarro() throws Exception
 {
 setLayout(new BorderLayout());

 panelVehiculo = new PanelVehiculo(this);
 add(panelVehiculo, BorderLayout.NORTH);

 panelDescuentos = new PanelDescuentos();
 add(panelDescuentos, BorderLayout.CENTER);

 panelResultados = new PanelResultados(this);
 add(panelResultados, BorderLayout.SOUTH);
 }
}
```

Se crea una instancia  
de cada uno de los  
paneles



# Creación de los paneles en el método Constructor

```
public class InterfazImpuestosCarro extends JFrame
{
 private PanelVehiculo panelVehiculo;
 private PanelDescuentos panelDescuentos;
 private PanelResultados panelResultados;

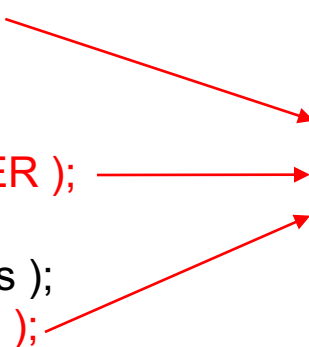
 public InterfazImpuestosCarro() throws Exception
 {
 setLayout(new BorderLayout());

 panelVehiculo = new PanelVehiculo(this);
 add(panelVehiculo, BorderLayout.NORTH);

 panelDescuentos = new PanelDescuentos();
 add(panelDescuentos, BorderLayout.CENTER);

 panelResultados = new PanelResultados(this);
 add(panelResultados, BorderLayout.SOUTH);
 }
}
```

Se agrega cada  
panel en una posición  
de las definidas en el  
distribuidor gráfico



# Creación de los paneles en el método Constructor

```
public class InterfazImpuestosCarro extends JFrame
{
 private PanelVehiculo panelVehiculo;
 private PanelDescuentos panelDescuentos;
 private PanelResultados panelResultados;

 public InterfazImpuestosCarro() throws Exception
 {
 setLayout(new BorderLayout());

 panelVehiculo = new PanelVehiculo(this);
 add(panelVehiculo, BorderLayout.NORTH);

 panelDescuentos = new PanelDescuentos();
 add(panelDescuentos, BorderLayout.CENTER);

 panelResultados = new PanelResultados(this);
 add(panelResultados, BorderLayout.SOUTH);
 }
}
```

The screenshot shows a Java Swing window titled "Cálculo impuestos". It contains three distinct panels arranged vertically, which correspond to the code in the constructor:

- Datos del vehículo:** This panel contains four text input fields labeled "Marca" (with "Renault"), "Línea" (with "307"), "Modelo" (with "2005"), and "Valor" (with "\$ 71,800,000"). A "Buscar" button is located at the bottom right of this section.
- Descuentos:** This panel contains three checkboxes: "Pronto pago", "Traslado de cuenta", and "Servicio público". All are currently unchecked.
- Resultados:** This panel displays the "Total a pagar" as "\$ 0". It includes a "Limpiar" button, a "Calcular" button, and two buttons at the bottom labeled "Opcion 1" and "Opcion 2".

Colored arrows and brackets on the left side of the image link the code to the UI elements: a red arrow points from the `panelVehiculo` initialization to the vehicle data fields; a blue arrow points from the `panelDescuentos` initialization to the discount checkboxes; and a purple arrow points from the `panelResultados` initialization to the results section.

# Analicemos la creación del panelVehiculo

Atributo panelVehiculo de la  
clase InterfazImpuestosCarro



```
panelVehiculo = new PanelVehiculo(this);
add(panelVehiculo, BorderLayout.NORTH);
```



# Analicemos la creación del panelVehiculo

- Llamado al constructor de la clase PanelVehiculo



- `panelVehiculo = new PanelVehiculo( this ); add( panelVehiculo, BorderLayout.NORTH );`

# Analicemos la creación del panelVehiculo



Parámetro del método  
constructor de la  
clase PanelVehiculo



```
panelVehiculo = new PanelVehiculo(this);
add(panelVehiculo, BorderLayout.NORTH);
```

Analicemos la creación del `panelVehiculo`

```
panelVehiculo = new PanelVehiculo(this);
add(panelVehiculo, BorderLayout.NORTH);
```



Método de la clase `JFrame` para  
adicionar un componente

Analicemos la creación del panelVehiculo

```
panelVehiculo = new PanelVehiculo(this);
add(panelVehiculo, BorderLayout.NORTH);
```



Objeto que se va a adicionar

Analicemos la creación del panelVehiculo

```
panelVehiculo = new PanelVehiculo(this);
add(panelVehiculo, BorderLayout.NORTH);
```



↑  
Zona (posición relativa dentro de la ventana) donde se va a adicionar el objeto. Es una constante de la clase BorderLayout.

## Dos conceptos nuevos



- BorderLayout



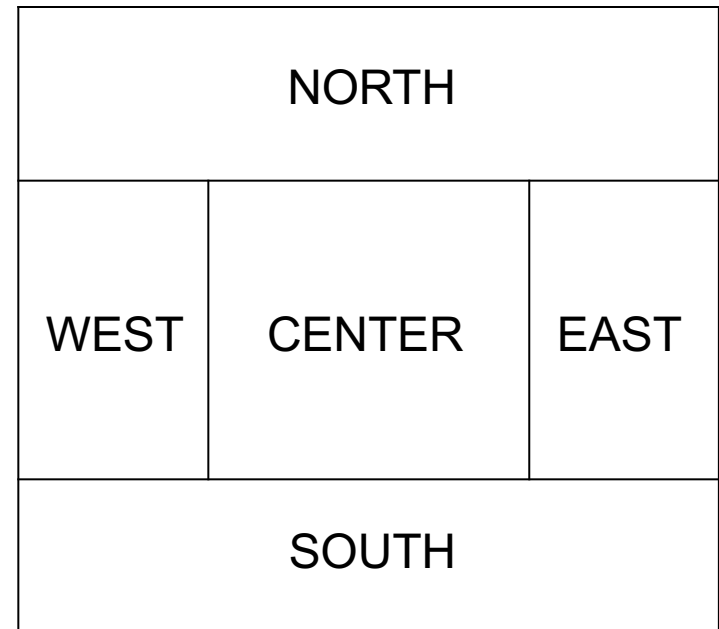
- this

# Dos conceptos nuevos

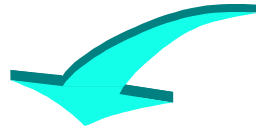


## • BorderLayout

- **Distribuidor gráfico “en los Bordes”**
- Divide el espacio de la ventana en **5 zonas**: NORTH, CENTER, SOUTH, WEST, EAST.
- Al agregar un componente a la ventana, SE DEBE pasar como parámetro la zona donde se va a ubicar. Ejemplo:
  - `add( panelVehiculo, BorderLayout.NORTH );`
- Utiliza el tamaño definido para cada uno de los componentes y asigna TODO el espacio sobrante al componente que se encuentre en la zona del centro



# Dos conceptos nuevos



## • Otro distribuidor: GridLayout

- Distribuidor gráfico “en Malla”
- Divide el espacio de la ventana en filas y columnas.
- La cantidad de filas y columnas se establecen en el método constructor del GridLayout. Ejemplo:
  - `setLayout( new GridLayout ( 4, 3 ) );`
- Al agregar un componente a la ventana, NO SE DEBE especificar la posición. Esta es asignada en el orden de llegada (fila 1, fila 2, ...)
- Ignora el tamaño definido para cada componente. Hace una distribución uniforme del espacio.

|        |    |    |    |
|--------|----|----|----|
| Fila 1 | 1  | 2  | 3  |
| Fila 2 | 4  | 5  | 6  |
| Fila 3 | 7  | 8  | 9  |
| Fila 4 | 10 | 11 | 12 |



## Dos conceptos nuevos



- BorderLayout



- this

# Dos conceptos nuevos



- **this**

- Es una variable de JAVA
- Hace referencia al objeto que está ejecutando un método

# Ejemplo de uso de this

```
public class InterfazImpuestosCarro extends JFrame
{
 private PanelVehiculo panelVehiculo;
 private PanelDescuentos panelDescuentos;
 private PanelResultados panelResultados;

 public InterfazImpuestosCarro() throws Exception
 {
 setLayout(new BorderLayout());

 panelVehiculo = new PanelVehiculo(this);
 add(panelVehiculo, BorderLayout.NORTH);

 panelDescuentos = new PanelDescuentos();
 add(panelDescuentos, BorderLayout.CENTER);

 panelResultados = new PanelResultados(this);
 add(panelResultados, BorderLayout.SOUTH);
 }
}
```

**Cálculo impuestos**

**Datos del vehículo**

|        |               |
|--------|---------------|
| Marca  | Renault       |
| Línea  | 307           |
| Modelo | 2005          |
| Valor  | \$ 71,800,000 |

Buscar

**Descuentos**

☐ Pronto pago ☐ Traslado de cuenta

☐ Servicio público

**Resultados**

|               |          |          |
|---------------|----------|----------|
| Total a pagar | \$ 0     | Limpiar  |
|               |          | Calcular |
|               | Opcion 1 | Opcion 2 |

## Ejemplo de uso de this

Es la ventana principal, (objeto de la clase InterfazImpuestosCarro, que es el “padre” del panel)



```
panelVehiculo = new PanelVehiculo(this);
add(panelVehiculo, BorderLayout.NORTH);
```

Para qué sirve? ... VER MAS ADELANTE

Hasta aquí hemos visto ...

- Cómo construir la clase de la ventana principal (**InterfazImpuestosCarro**)

**Ahora vamos a ver ...**

- Cómo construir las clases de los paneles (**PanelVehiculo, PanelDescuentos, PanelResultados**)

# Construcción de las clases de los paneles

- Proceso similar a la construcción de la clase de la ventana principal
- Al igual que la ventana principal, los paneles también son contenedores gráficos

Ventana Principal



JFrame

# Construcción de las clases de los paneles

- Proceso similar a la construcción de la clase de la ventana principal
- Al igual que la ventana principal, los paneles también son contenedores gráficos

Ventana Principal



JFrame

Panel

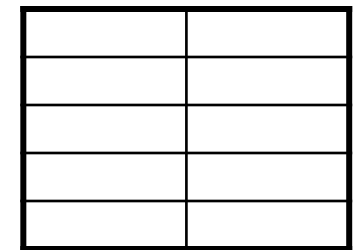


JPanel

# PanelVehiculo

```
public class PanelVehiculo extends JPanel
{
 public PanelVehiculo()
 {
 ...
 setLayout(new GridLayout(5, 2));
 setPreferredSize(new Dimension(0, 130));
 ...
 }
}
```

Creación y asociación de  
un GridLayout de 5 filas y  
2 columnas



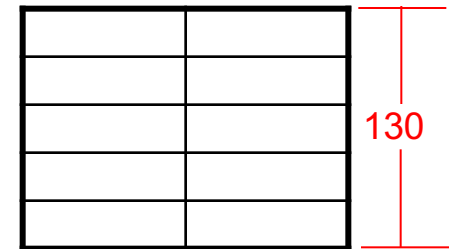


# PanelVehiculo

```
public class PanelVehiculo extends JPanel
{
 public PanelVehiculo()
 {
 ...
 setLayout(new GridLayout(5, 2));
 setPreferredSize(new Dimension(0, 130));
 ...
 }
}
```

Clase de Java que  
permite definir un ancho  
y un alto usando un  
objeto

Definir la altura del panel.  
No se define el ancho  
porque va a ser igual al  
de la ventana

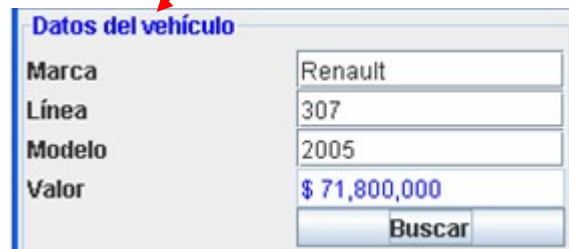


| Datos del vehículo                    |               |
|---------------------------------------|---------------|
| Marca                                 | Renault       |
| Línea                                 | 307           |
| Modelo                                | 2005          |
| Valor                                 | \$ 71,800,000 |
| <input type="button" value="Buscar"/> |               |

# PanelVehiculo

```
public class PanelVehiculo extends JPanel
{
 public PanelVehiculo()
 {
 ...
 setLayout(new GridLayout(5, 2));
 setPreferredSize(new Dimension(0, 130));
 TitledBorder border = BorderFactory.createTitledBorder
 ("Datos del vehículo");
 border.SetTitleColor(Color.BLUE);
 setBorder(border);
 ...
 }
}
```

Se crea y se asocia un borde al panel para facilitar la identificación de las divisiones dentro de la ventana



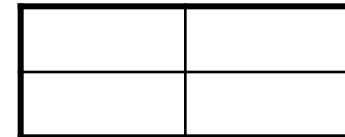
The screenshot shows a Java Swing window titled "Datos del vehículo". The window has a blue title bar and a blue border. Inside, there is a form with four labels and text fields: "Marca" (Renault), "Línea" (307), "Modelo" (2005), and "Valor" (\$ 71,800,000). The text color for the labels and the "Valor" field is blue. At the bottom right, there is a "Buscar" button.

|                                       |               |
|---------------------------------------|---------------|
| Marca                                 | Renault       |
| Línea                                 | 307           |
| Modelo                                | 2005          |
| Valor                                 | \$ 71,800,000 |
| <input type="button" value="Buscar"/> |               |

# PanelDescuentos

```
public class PanelDescuentos extends JPanel
{
 public PanelDescuentos()
 {
 ...
 setLayout(new GridLayout(2, 2));
 ...
 }
}
```

Creación y asociación de  
un GridLayout de 2 filas y  
2 columnas



# PanelDescuentos

```
public class PanelDescuentos extends JPanel
{
 public PanelDescuentos()
 {
 ...
 setLayout(new GridLayout(2, 2));
 TitledBorder border = BorderFactory.createTitledBorder
 ("Descuentos");
 border.setTitleColor(Color.BLUE);
 setBorder(border);
 ...
 }
}
```

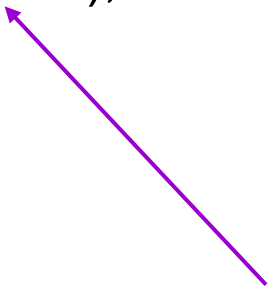
Se crea y se asocia un  
borde



# Etiquetas y Zonas de Texto


# Etiquetas

- Permiten agregar un **texto** corto en la interfaz
- Son objetos de la clase **JLabel** de Java
- Algunos **métodos** de la clase JLabel:
  - setText (texto);
  - setForeground ( **color** );




Cualquier constante de la clase Color de Java (BLACK, GREEN, RED, BLUE, ...) o un nuevo color creado con 3 índices RGB

# Zonas de texto


- Cumplen dos **funciones**:
  - Permiten el **ingreso** de información por parte del usuario (**ENTRADAS** de los requerimientos funcionales)
  - Permiten **mostrar** las respuestas calculadas por el programa
- Son objetos de la clase **JTextField** de Java
- Algunos **métodos** de la clase JText:
  - **getText ( );**  Retorna la cadena de caracteres tecleada por el usuario. SIEMPRE es String.

# Zonas de texto

- Cumplen dos **funciones**:
  - Permiten el **ingreso** de información por parte del usuario (**ENTRADAS** de los requerimientos funcionales)
  - Permiten **mostrar** las respuestas calculadas por el programa
- Son objetos de la clase **JTextField** de Java
- Algunos **métodos** de la clase JLabel:
  - `getText ( );`
  - `setText ( texto );`  Despliega el texto que se pasa como parámetro. Se usa para mostrar los resultados del programa.



# Zonas de texto

- Cumplen dos **funciones**:
  - Permiten el **ingreso** de información por parte del usuario (**ENTRADAS** de los requerimientos funcionales)
  - Permiten **mostrar** las respuestas calculadas por el programa
- Son objetos de la clase **JTextField** de Java
- Algunos **métodos** de la clase JLabel:
  - `getText ( );`
  - `setText ( texto );`
  - `setEditable ( true / false )`  **Indica si el usuario puede o no modificar el texto (escribir encima)**

# Zonas de texto

- Cumplen dos **funciones**:
    - Permiten el **ingreso** de información por parte del usuario (**ENTRADAS** de los requerimientos funcionales)
    - Permiten **mostrar** las respuestas calculadas por el programa
  - Son objetos de la clase **JTextField** de Java
  - Algunos **métodos** de la clase JLabel:
    - `getText ( );`
    - `setText ( texto );`
    - `setEditable ( true / false)`
    - **`setForeground (color )`**
    - **`setBackground ( color )`**
- ← Definen respectivamente el color del texto y del fondo

Cómo agregar una etiqueta (o una zona de texto) a un panel

- ❶ Declarar en el panel un **atributo** de la clase **JLabel** (o **TextField**)
- ❷ Crear la etiqueta (o la zona de texto) (**new**) en el método constructor del panel
- ❸ Configurar las **características** de la etiqueta (o de la zona de texto) con los métodos de la clase **JLabel** (o **TextField**)
- ❹ **Agregar** la etiqueta (o la zona de texto) al panel (**add**)

# Ejemplo en PanelVehiculo

```
public class PanelVehiculo extends JPanel
{
```

```
...
```

```
private JLabel labMarca;
private JLabel labLinea;
private JLabel labModelo;
private JLabel labValor;
```

```
private JTextField txtMarca;
private JTextField txtLinea;
private JTextField txtModelo;
private JTextField txtValor;
```

```
...
```


```
}
```

❶ Declarar en el panel los atributos de clases **JLabel** y **JTextField**

# Ejemplo en PanelVehiculo

```
public PanelVehiculo ()
{
 ...
 labMarca = new JLabel ("Marca");
 labLinea = new JLabel ("Línea");
 labModelo = new JLabel ("Modelo");
 labValor = new JLabel ("Valor");

 txtMarca = new JTextField();
 txtLinea = new JTextField();
 txtModelo = new JTextField();
 txtValor = new JTextField("$ 0");
 ...
}
```



② Crear las etiquetas y las zonas de texto (**new**) en el método constructor del panel

# Ejemplo en PanelVehiculo

```
public PanelVehiculo ()
{
 ...
 labMarca = new JLabel ("Marca");
 labLinea = new JLabel ("Línea");
 labModelo = new JLabel ("Modelo");
 labValor = new JLabel ("Valor");

 txtMarca = new JTextField();
 txtLinea = new JTextField();
 txtModelo = new JTextField();
 txtValor = new JTextField("$ 0");

 txtValor.SetEditable(false);
 txtValor.SetForeground(Color.BLUE);
 txtValor.SetBackground(Color.WHITE);
 ...
}
```

③ Configurar las **características** de las **etiquetas** o de las **zonas de texto** con los métodos de las clases **JLabel** o **JTextField**

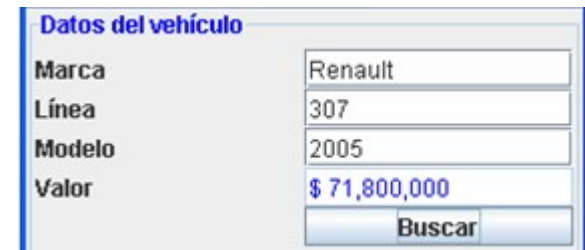
# Ejemplo en PanelVehiculo

```
public PanelVehiculo ()
{
 ...
 labMarca = new JLabel ("Marca");
 labLinea = new JLabel ("Línea");
 labModelo = new JLabel ("Modelo");
 labValor = new JLabel ("Valor");
```

```
 txtMarca = new JTextField();
 txtLinea = new JTextField();
 txtModelo = new JTextField();
 txtValor = new JTextField("$ 0");
```

```
 txtValor.SetEditable(false);
 txtValor.SetForeground(Color.BLUE);
 txtValor.SetBackground(Color.WHITE);
```

```
 add(labMarca);
 add(txtMarca);
 add(labLinea);
 add(txtLinea);
 add(labModelo);
 add(txtModelo);
 add(labValor);
 add(txtValor);
```



| Datos del vehículo                    |               |
|---------------------------------------|---------------|
| Marca                                 | Renault       |
| Línea                                 | 307           |
| Modelo                                | 2005          |
| Valor                                 | \$ 71,800,000 |
| <input type="button" value="Buscar"/> |               |

**4** Agregar las etiquetas  
y las zonas de texto  
al panel (add)

Selección de Opciones por  
medio de Cajas de Chequeo



# Cajas de chequeo


- Permiten al usuario seleccionar o deseleccionar una opción.
- Son objetos de la clase **JCheckBox** de Java
- Algunos **métodos** de la clase JCheckBox:

– `isSelected ( );`



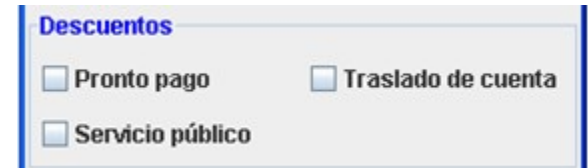
Indica (true/false) si el usuario seleccionó la opción

# Cajas de chequeo

- Permiten al usuario seleccionar o deseleccionar una opción.
- Son objetos de la clase **JCheckBox** de Java
- Algunos **métodos** de la clase JCheckBox:
  - `isSelected ( );`
  - `setSelected ( true/false );`  **Marca como seleccionado o no, la caja de chequeo**

# Ejemplo en PanelDescuentos

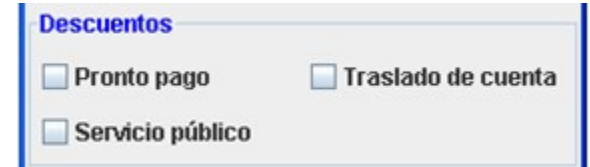
```
public class PanelDescuentos extends JPanel
{
 ...
 private JCheckBox cbPago;
 private JCheckBox cbSPublico;
 private JCheckBox cbTCuenta;
 ...
}
```



❶ Declarar en el panel los atributos de clase JCheckBox

# Ejemplo en PanelDescuentos

```
public PanelDescuentos ()
{
 ...
 cbPago = new JCheckBox ("Pronto pago");
 cbSPublico = new JCheckBox ("Servicio público");
 cbTCuenta = new JCheckBox ("Traslado de cuenta");
 ...
}
```

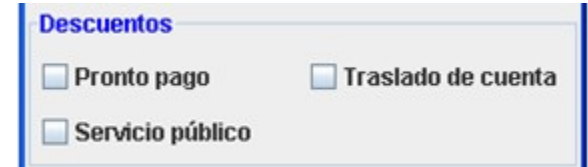


② Crear las **cajas de chequeo (new)** en el método constructor del panel

# Ejemplo en PanelDescuentos

```
public PanelDescuentos ()
{
 ...
 cbPago = new JCheckBox ("Pronto pago");
 cbSPublico = new JCheckBox ("Servicio público");
 cbTCuenta = new JCheckBox ("Traslado de cuenta");

 add(cbPago);
 add(cbTCuenta);
 add(cbSPublico);
 ...
}
```



③ Agregar las cajas de chequeo al panel (add)

# Interacción con la aplicación mediante Botones

# Botones

- Permiten al usuario expresar sus órdenes al programa (es el mecanismo más simple de interacción).
- Son objetos de la clase **JButton** de Java
- La clase JButton tiene **DOS METODOS** especiales:
  - `setActionCommand ( evento );`
  - `addActionListener ( panel );`

# Ejemplo en PanelResultados

```
public class PanelResultados extends JPanel
{
 ...
 private JLabel labTotal;
 private JTextField txtTotal;

 private JButton butLimpiar;
 private JButton butCalcular;
 ...
}
```



❶ Declarar en el panel los atributos de clase JButton



# Ejemplo en PanelResultados

```
public PanelResultados ()
{
 ...
 labTotal = new JLabel ("Total a pagar");
 txtTotal = new JTextField ("$ 0");

 butLimpiar = new JButton ("Limpiar");
 butCalcular = new JButton ("Calcular");
 ...
}
```



② Crear los botones (new) en el método constructor del panel

# Ejemplo en PanelResultados

```
public PanelResultados ()
{
 ...
 labTotal = new JLabel ("Total a pagar"
);
 txtTotal = new JTextField ("$ 0");

 butLimpiar = new JButton ("Limpiar");
 butCalcular = new JButton ("Calcular");

 txtTotal.SetEditable(false);
 txtTotal.SetForeground(Color.BLUE);
 txtTotal.SetBackground(Color.WHITE);

 add(new JLabel(""));
 add(new JLabel(""));
 add(butLimpiar);
 add(labTotal);
 add(txtTotal);
 add(butCalcular);
 ...
}
```



|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |

③ Agregar los botones  
al panel (add)