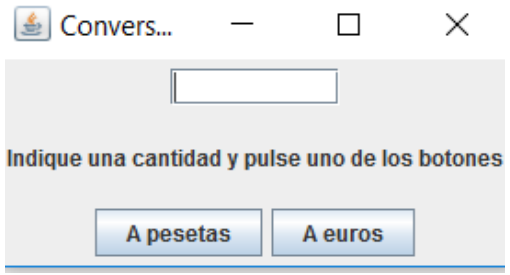
MODELO VISTA CONTROLADOR

Javier García-Retamero Redondo

UN PEQUEÑO EJEMPLO

- Piensa en cómo sería el desarrollo sin arquitectura de una calculadora que convierta de euros a pesetas y de pesetas a euros.



UN PEQUEÑO EJEMPLO

- ¿Qué modificaciones tendríamos que hacer si quisiéramos que nuestra aplicación estuviese preparada para utilizar otro tipo de interfaz gráfica?

```
Indica la operación que quiere realizar:
1: convertir euros a euros
2: convertir pesetas a pesetas
0: salir
```

- Si algún día tenemos que modificar la lógica de negocio, ¿tendríamos que modificar las dos aplicaciones?¿habría que volver a distribuirlas a los clientes?

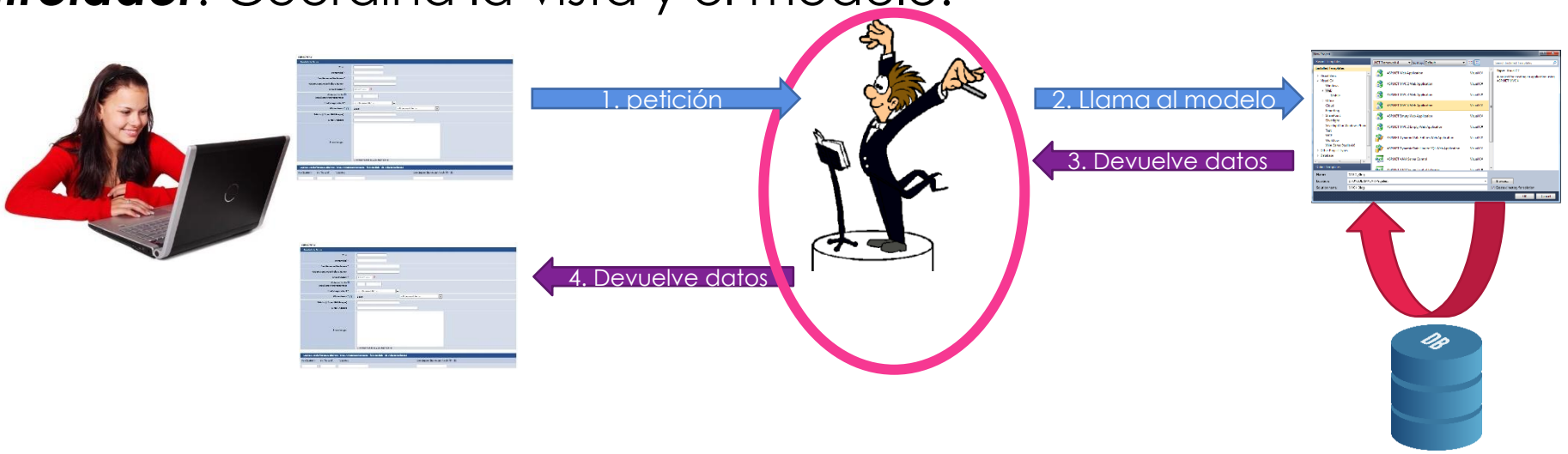
EL MVC LLEGA A NUESTRAS VIDAS

¿QUÉ ES EL MVC?

- Es un patrón de diseño de arquitecturas para el desarrollo de aplicaciones software.
- Separa la lógica de negocio de la interfaz de usuario.
  - *Ventajas*:
    - Separa los datos de la representación visual de los mismos.
    - Diseño de aplicaciones modulares.
    - Reutilización de código.
    - Facilidad para probar las unidades por separado.
    - Facilita el mantenimiento y la detección de errores.
  - *Inconvenientes*:
    - Gran cantidad de archivos
    - Complejidad aumentada al separar conceptos en capas

COMPONENTES

- ***La vista***: Parte gráfica que interactúa con el usuario.
- ***El modelo***: Datos de la aplicación y reglas de negocio.
- ***El controlador***: Coordina la vista y el modelo.



VISTAS: Recogen las peticiones del usuario y muestra los resultados

CONTROLADOR: Selecciona el modelo a ejecutar y la vista en la que mostrar el resultado

MODELOS: Procesan los datos y los devuelven

DESARROLLO UTILIZANDO EL MVC

APLICAMOS EL MVC

- En nuestra pequeña aplicación debemos desarrollar lo siguientes componentes:
  - Modelo:
    - Conversor
    - ConversorEurosPesetas
  - Vista:
    - InterfazVista
    - VentanaConversor
  - El controlador:
    - ControlConversor
  - El programa
    - ProgramaDeConversion

EL MODELO

- Conversor

```
package com.break4learning.mvc;

public class Conversor {
    private final double cambio;

    public Conversor (double valorCambio) {
        cambio = valorCambio;
    }
    public double eurosAmoneda (double cantidad) {
        return cantidad * cambio;
    }
    public double monedaAeuros (double cantidad) {
        return cantidad / cambio;
    }
}
```

EL MODELO

- ConversorEurosPesetas

```
package com.break4learning.mvc;

public class ConversorEurosPesetas extends Conversor {

    public ConversorEurosPesetas () {
        super(166.386D);
    }
    public double eurosApesetas(double cantidad) {
        return eurosAmoneda(cantidad);
    }
    public double pesetasAeuros(double cantidad) {
        return monedaAeuros(cantidad);
    }
}
```

LA VISTA

- InterfazVista

Definimos los métodos que deberá implementar cualquier interfaz gráfica para que puedan interactuar con ella el controlador y el programa

```
package com.break4learning.mvc;

public interface InterfazVista {
    // especifica el controlador que se va a encargar de procesar
    //las acciones realizadas en la vista
    void setControlador(ControlConversor c);

    // comienza la visualización
    void arranca();

    // obtiene de pantalla la cantidad a convertir
    double getCantidad();

    //setea el valor convertido
    void escribeCambio(String s);

    // Constantes para las operaciones:
    static final String AEUROS="Pesetas a Euros";
    static final String APESETAS="Euros a Pesetas";
}
```

LA VISTA

- VentanaConversor

```
public class VentanaConversor extends JFrame implements InterfazVista {
    private JButton convertirApesetas;
    private JButton convertirAeuros;
    private JTextField cantidad;
    private JLabel resultado;

    public VentanaConversor () {
        super("Conversor de Euros y Pesetas");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel panelPrincipal = new JPanel();
        panelPrincipal.setLayout(new BorderLayout(10,10));
        cantidad = new JTextField(8);
        JPanel panelaux = new JPanel();
        panelaux.add(cantidad);
        panelPrincipal.add(panelaux, BorderLayout.NORTH);
        resultado = new JLabel("Indique una cantidad y pulse uno de los botones");
        JPanel panelaux2 = new JPanel();
        panelaux2.add(resultado);
        panelPrincipal.add(panelaux2, BorderLayout.CENTER);
        convertirApesetas = new JButton("A pesetas");
        convertirApesetas.setActionCommand(APESETAS);
        convertirAeuros = new JButton("A euros");
        convertirAeuros.setActionCommand(AEUROS);
        JPanel botonera = new JPanel();
        botonera.add(convertirApesetas);
        botonera.add(convertirAeuros);
        panelPrincipal.add(botonera, BorderLayout.SOUTH);
        getContentPane().add(panelPrincipal);
    }
```

Dibujamos los componentes

Nombre para detectar el botón que lanza el evento

LA VISTA

- VentanaConversor

Implementamos los métodos de la interface a los que llamará el controlador

```
@Override
public void escribeCambio(String s) {
     resultado.setText(s);
}
@Override
public double getCantidad() {
     try {
          return Double.parseDouble(cantidad.getText());
     } catch (NumberFormatException e) {
          return 0.0D;
     }
}
@Override
public void arranca() {
     pack();// coloca los componentes
     setLocationRelativeTo(null);// centra la ventana en la pantalla
     setVisible(true);// visualiza la ventana
}
@Override
public void setControlador(ControlConversor c) {
     convertirApesetas.addActionListener(c);
     convertirAeuros.addActionListener(c);
}
}
```

Quien realiza la acción de los Listener es el Controlador

EL CONTROLADOR

- ControlConversor

Para el controlador necesitamos la vista y el modelo

```
public class ControlConversor implements ActionListener {
     private final InterfazVista vista;
     private final ConversorEurosPesetas modelo;

     public ControlConversor(InterfazVista vista, ConversorEurosPesetas modelo) {
          this.vista = vista;
          this.modelo = modelo;
          this.vista.setControlador(this);
          this.vista.arranca();
     }
```

Método que se lanza al pulsar un botón

```
     public void actionPerformed(ActionEvent evento) {
          double cantidad = vista.getCantidad();
           switch (evento.getActionCommand()) {
                case InterfazVista.AEUROS -> vista.escribeCambio(cantidad +" pesetas son: "+
                                                 modelo.pesetasAeuros(cantidad) + " euros");
                case InterfazVista.APESETAS -> vista.escribeCambio(cantidad + " euros son: "+
                                                 modelo.eurosApesetas(cantidad) + " pesetas");
                default -> vista.escribeCambio("ERROR");
             }}
```

EL PROGRAMA

- ProgramaDeConversion

```
public class ProgramaDeConversion {
    public static void main(String[] args) {

        // la vista:
        InterfazVista vista = new VentanaConversor();

        // el modelo:
        ConversorEurosPesetas modelo = new ConversorEurosPesetas();

        // Creamos el control pasándole la vista y el modelo:
        ControlConversor control = new ControlConversor (vista,modelo);

    }
}
```

MODIFICANDO LA INTERFAZ GRÁFICA

LA VISTA

- VentanaConversor

Gestión de la entrada por teclado

```
public class VentanaConversor implements InterfazVista {

	private ControlConversor controlador;
	private BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

	private int leeOpción() {
		String s = null;
		try {
			s = in.readLine();
			return Integer.parseInt(s);
		} catch(Exception e){
			operaciónIncorrecta();
			return 0;
		}
	}

	private double leeCantidad() {
		String s = null;
		try{
			s = in.readLine();
			return Double.parseDouble(s);
		} catch(IOException | NumberFormatException e){
			System.out.println("Error en formato del número, tiene que ser 99.99: ");
			return leeCantidad();
		}
	}
```

LA VISTA

- VentanaConversor

Gestión de la entrada por teclado

```
private void solicitaOperación() {
      System.out.println("Indica la operación que quiere realizar:");
      System.out.println("1: convertir euros a euros");
      System.out.println("2: convertir pesetas a pesetas");
      System.out.println("0: salir");
}

private void procesaNuevaOperacion() {
      int operacion;
      solicitaOperación();
      operacion = leeOpción();

      switch (operacion) {
            case 0 -> {
                System.out.println("Adiós.");
                System.exit(0);
            }
            case 1 -> controlador.actionPerformed(new ActionEvent(this, operacion, AEUROS));
            case 2 -> controlador.actionPerformed(new ActionEvent(this, operacion, APESETAS));
      }
      operaciónIncorrecta();
}

private void operaciónIncorrecta() {
      System.out.print("Operación incorrecta. ");
      procesaNuevaOperacion();
}
```

Es el controlador el encargado de llamar a los modelos para realizar las operaciones asociadas a los botones.

LA VISTA

- VentanaConversor

Implementamos los métodos de la interface a los que llamará el controlador

```
    @Override
    public void escribeCambio(String s) {
        System.out.println(s);
        procesaNuevaOperacion();
    }

    @Override
    public double getCantidad() {
        System.out.print("Cantidad a convertir (formato 99.99): ");
        return leeCantidad();
    }

    @Override
    public void arranca() {
        procesaNuevaOperacion();
    }

    @Override
    public void setControlador(ControlConversor c) {
        controlador = c;
    }
}
```

Quien realiza la acción de las opciones es el Controlador

EL MODELO

- NO ES NECESARIO MODIFICARLO

EL CONTROLADOR

- NO ES NECESARIO MODIFICARLO

EL PROGRAMA

- NO ES NECESARIO MODIFICARLO