



Universidad
de Alcalá

Departamento Ciencias
de la Computación



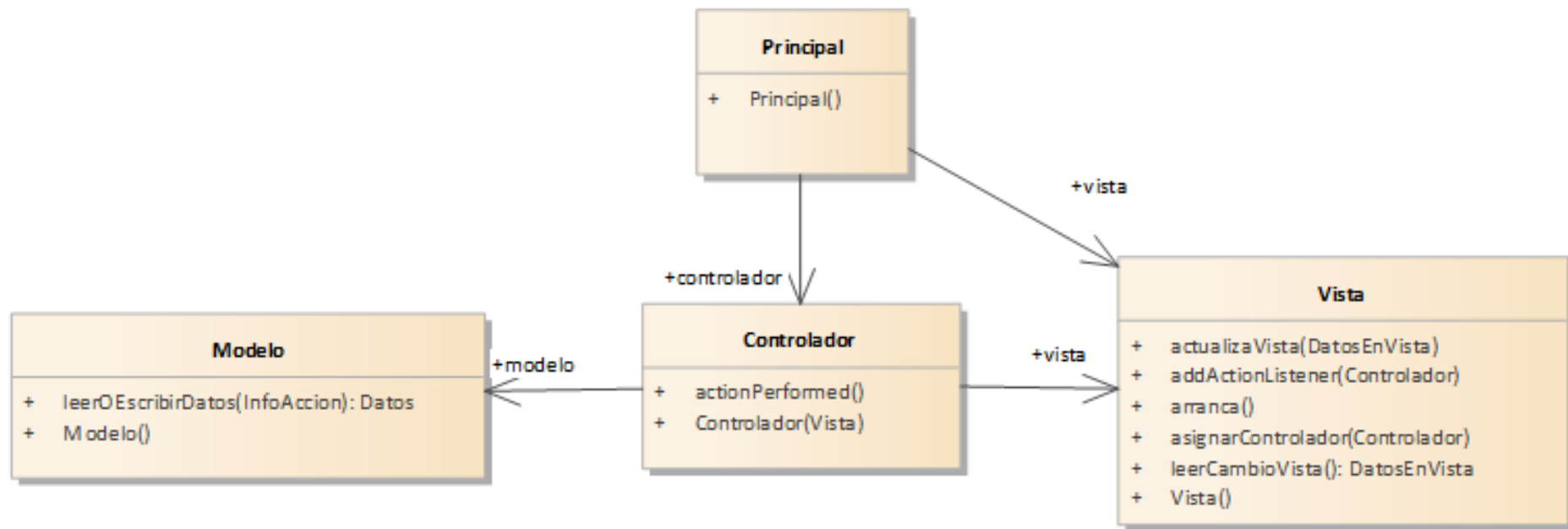
Patrón de diseño

Modelo-Vista-Controlador (MVC)

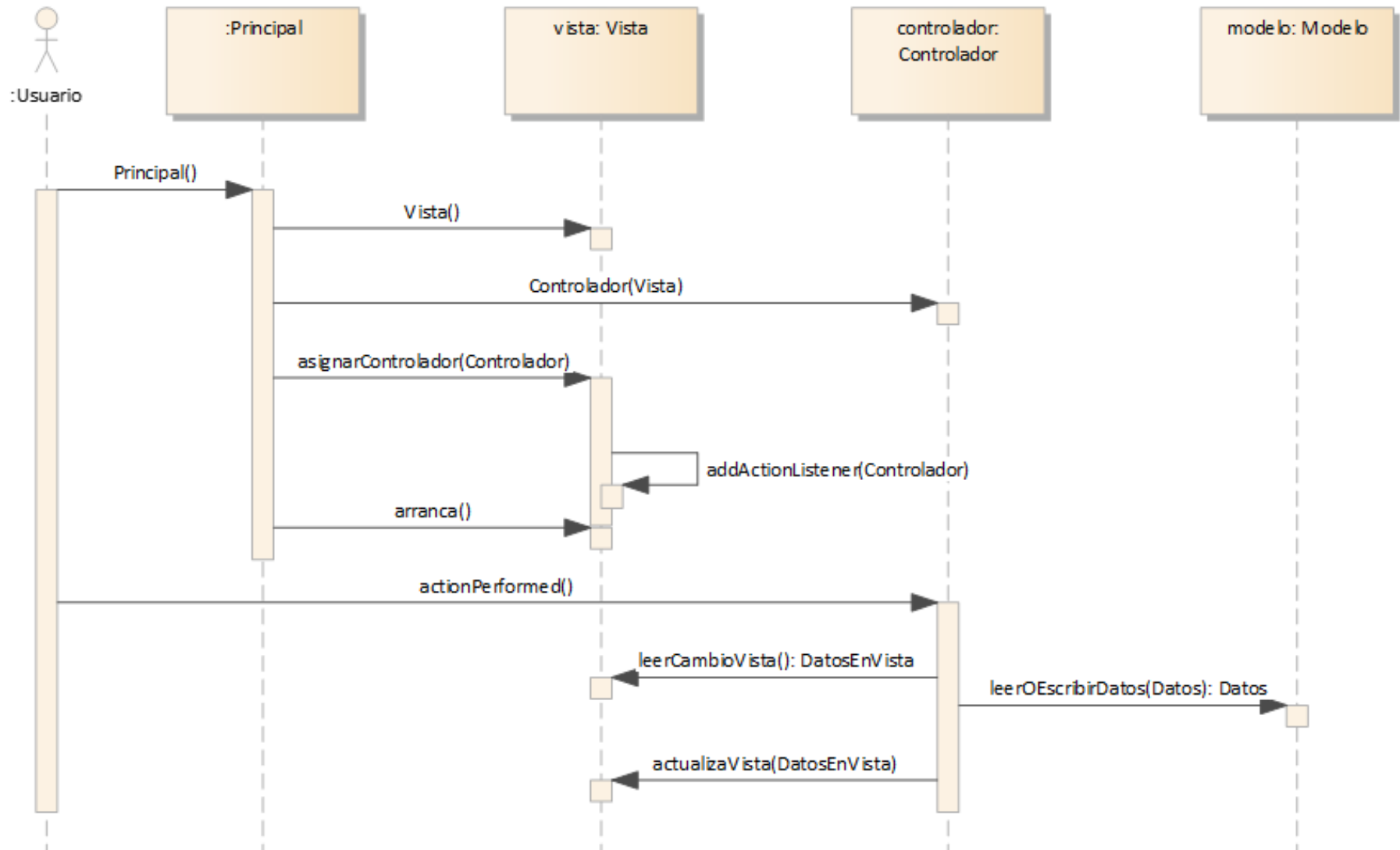
Objetivos del patrón MVC

- El objetivo es independizar las interfaces de usuario de los datos
- Para ello propone organizar la arquitectura del software en clases de tres tipos
 - Modelo: Clases que representan datos y la lógica de negocio.
 - Vista: Clases que representan interfaces de usuario.
 - Controlador: Clases que relacionan la vista con el modelo.

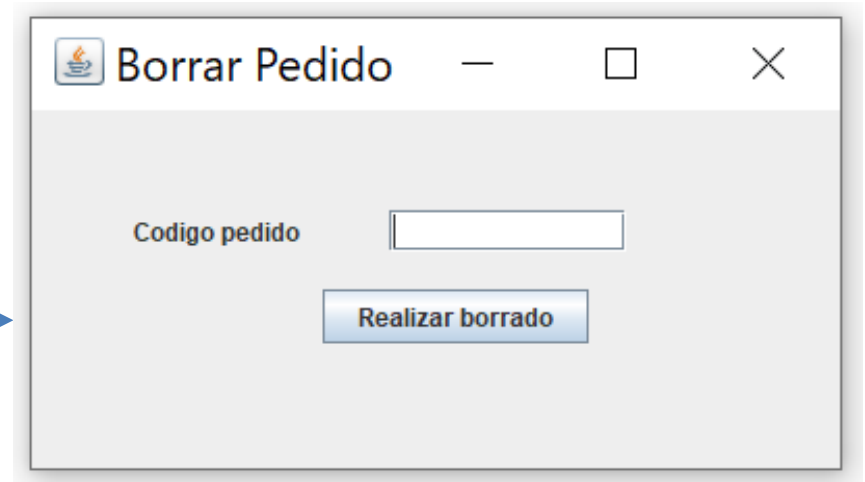
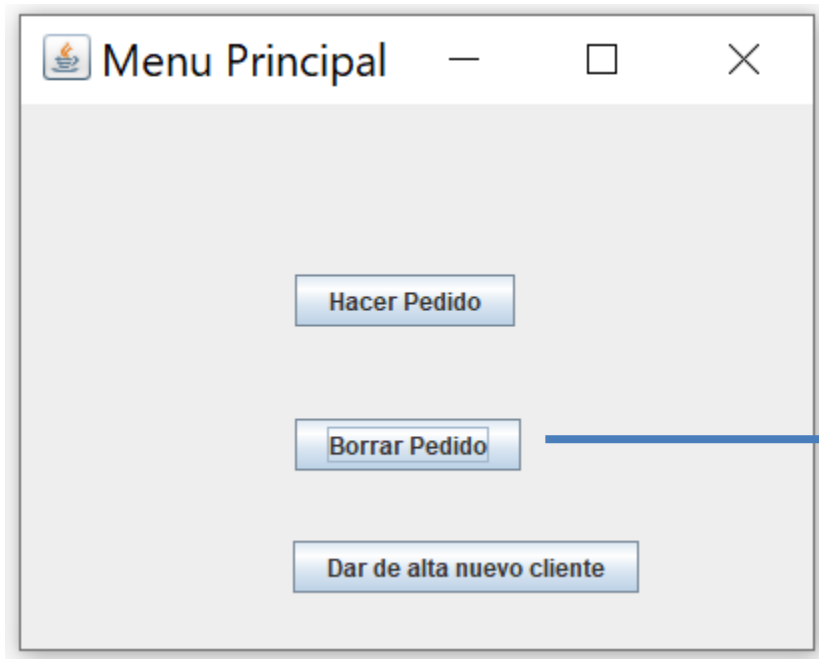
Clases en MVC



Funcionamiento MVC

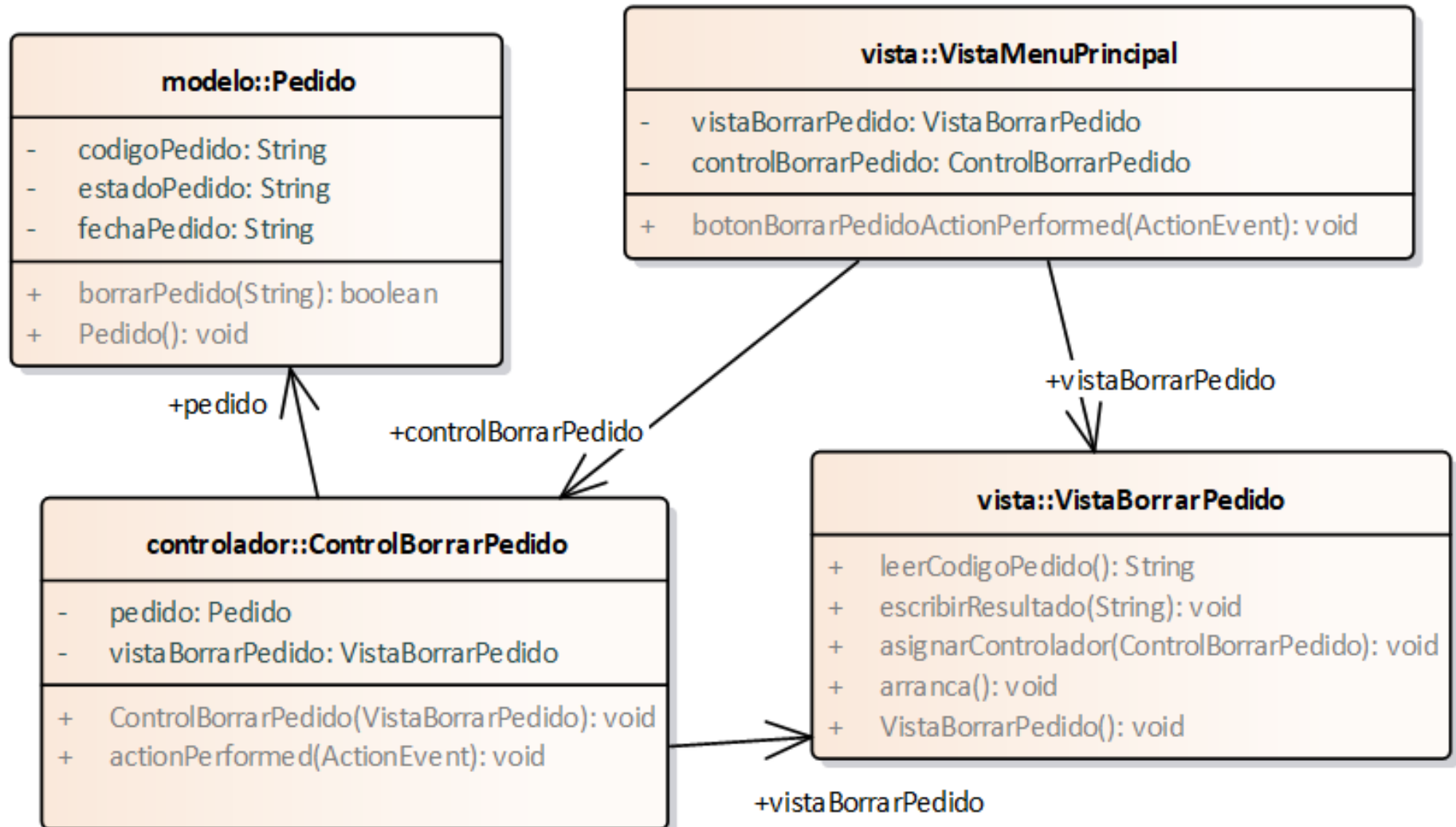


Ejemplo en Java



Ejemplo en Java

Diagrama de clases



Ejemplo en Java

Pedido.java (modelo)

```
public class Pedido {  
  
    private String codigoPedido;  
    private String estadoPedido;  
    private String fechaPedido;  
  
    public Pedido() {  
    }  
  
    public boolean borrarPedido (String codigoPedido) {  
        ...  
    }  
  
    ...  
}
```

Ejemplo en Java

VistaBorrarPedido.java (vista)

```
public class VistaBorrarPedido extends JFrame {

    private JButton botonRealizarBorrado;
    private JLabel labelCodigoPedido;
    private JLabel labelResultado;
    private JTextField textoCodigoPedido;

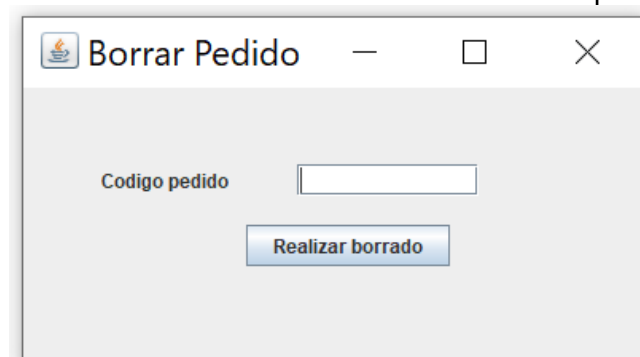
    public VistaBorrarPedido() { ...
    }

    public String leerCodigoPedido (){
        return textoCodigoPedido.getText();
    }

    public void escribirResultado (String resultado){
        labelResultado.setText(resultado);
    }

    void asignarControlador (ControlBorrarPedido c) {
        botonRealizarBorrado.addActionListener(c);
    }

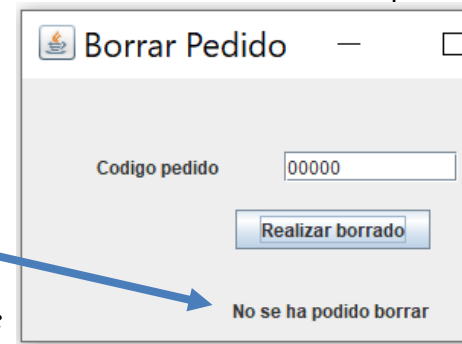
    void arranca () {
        setVisible(true);
    }
}
```



Ejemplo en Java

ControlBorrarPedido.java (controlador)

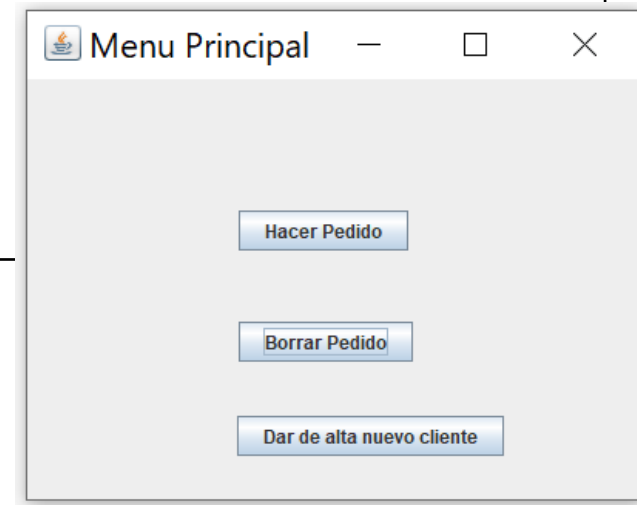
```
public class ControlBorrarPedido implements ActionListener {  
  
    Pedido pedido;                // Modelo  
    VistaBorrarPedido vistaBorrarPedido;    // Vista  
  
    public ControlBorrarPedido(VistaBorrarPedido v) {  
        vistaBorrarPedido = v;  
    }  
  
    public void actionPerformed(ActionEvent evento) {  
        pedido = new Pedido();  
        String codigoPedido = vistaBorrarPedido.leerCodigoPedido();  
        String resultado;  
        if (pedido.borrarPedido(codigoPedido)) {  
            resultado = "Borrado con éxito";  
        } else {  
            resultado = "No se ha podido borrar";  
        };  
        vistaBorrarPedido.escribirResultado(resultado);  
    }  
}
```



Ejemplo en Java

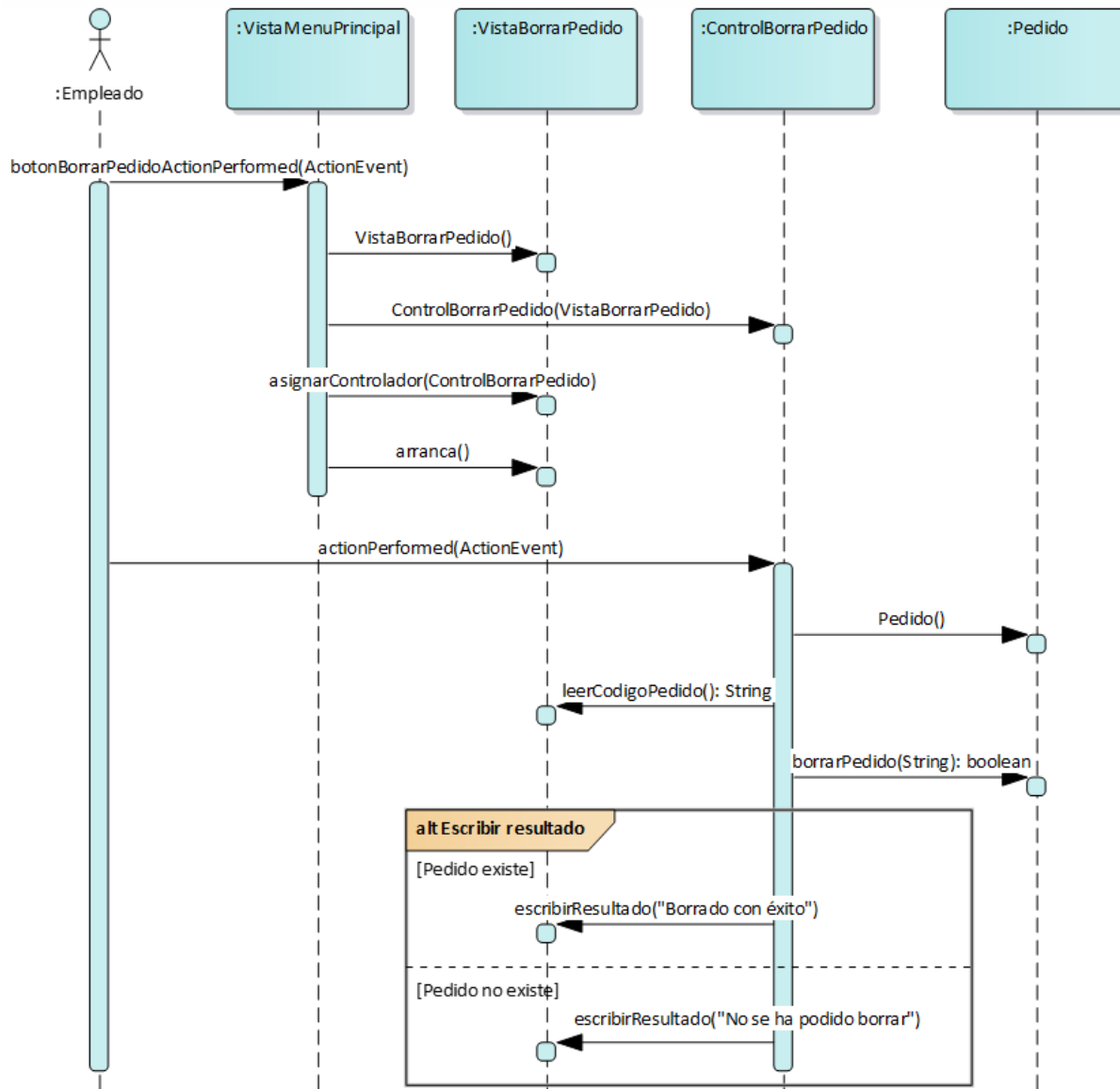
VistaMenuPrincipal.java (principal)

```
public class VistaMenuPrincipal extends JFrame {  
  
    VistaBorrarPedido vistaBorrarPedido;  
    ControlBorrarPedido controlBorrarPedido;  
  
    private void botonBorrarPedidoActionPerformed(ActionEvent evt) {  
        vistaBorrarPedido = new VistaBorrarPedido();  
        controlBorrarPedido = new ControlBorrarPedido(vistaBorrarPedido);  
        vistaBorrarPedido.asignarControlador(controlBorrarPedido);  
        vistaBorrarPedido.arranca();  
    }  
    ...  
}
```



Ejemplo en Java

Diagrama de secuencia



Ejemplo en Java

Diagrama de comunicación

