

MVC

Modelo Vista Controller

MVC: ¿Qué es?

- Es un **patrón de arquitectura** de software.
- Establece una separación entre:
 - ♦ La interfaz de usuario (Vista)
 - ♦ La lógica de negocio (Controlador)
 - ♦ Los datos (Modelo)

MVC: ¿Para qué?

- Permite:
 - ◆ Modularización.
 - ◆ Reutilización de código.

- Facilita:
 - ◆ El desarrollo.
 - ◆ El mantenimiento.
 - ◆ La legibilidad del código.

MVC: Componentes

Vista

Interfaz de usuario.
Presenta nuestra aplicación.

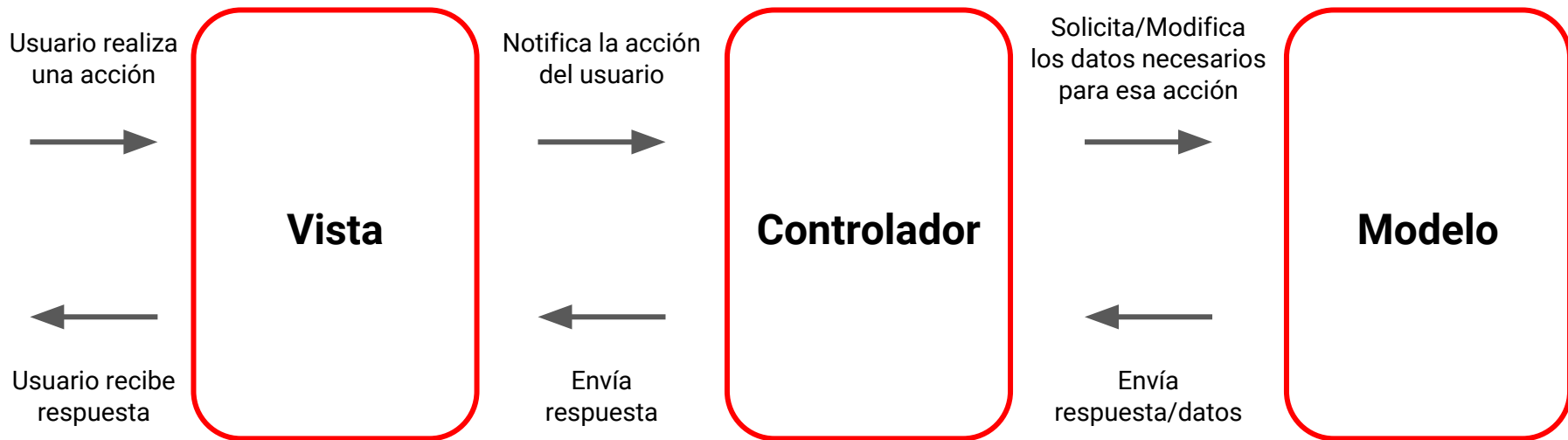
Controlador

Responde a eventos del usuario y realiza peticiones al modelo.
Intermediario entre la vista y el modelo.

Modelo

Representa la información que utiliza nuestra aplicación.

MVC: Interacción



MVC: Modelo

→ Dentro del modelo, existen dos componentes:

- ◆ **DAO: (Data Access Object)**

Es el encargado de gestionar y administrar el acceso a datos en bases de datos, archivos e internet.

- ◆ **Model POJO: (Plain Old Java Object)**

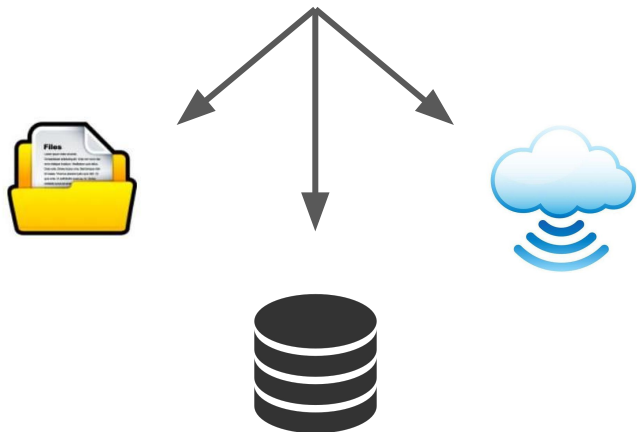
Es el modelo de datos que utiliza nuestra aplicación.

Modelo



DAO
POJO

MVC: Modelo



```
public class Estudiante {  
    private Integer numeroDeLegajo;  
    private String nombre;  
  
    public Integer getNumeroDeLegajo() {  
        return numeroDeLegajo;  
    }  
  
    public void setNumeroDeLegajo(Integer numeroDeLegajo) {  
        this.numeroDeLegajo = numeroDeLegajo;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
}
```

MVC: Aplicación

De ahora en adelante, todos nuestros proyectos, estarán divididos en cuatro packages:

→ **View:**

- ◆ En este package incluiremos todas las clases relacionadas con la Vista.
- ◆ *Ejemplo: ActivityRecyclerViewProductos - AdapterProductos*

→ **Controller:**

- ◆ En este package incluiremos un controller por cada modelo que debamos controlar.
- ◆ *Ejemplo: ProductosController*

→ **DAO:**

- ◆ En este package incluiremos todas las clases que permitan y administren el acceso a datos en bases de datos e internet. *Ejemplo: ProductosDAO*

→ **Model:**

- ◆ En este package incluiremos todas las clases que modelen nuestros datos.
- ◆ *Ejemplo: Producto*