

A young man with dark hair, wearing a red jacket over a striped shirt, has large black headphones around his neck and a backpack. He is holding a folder or book and looking towards the camera with a slight smile. The background is a blurred indoor setting with windows and plants.

PROMETEO

# PROGRAMACIÓN

Modelo Vista Controlador

# ¿Qué es MVC?

---

MVC = Model - View - Controller





Se define como un patrón arquitectónico que separa responsabilidades.

¿Qué problema resuelve MVC?

- ★ Aplicaciones desordenadas
  - Todo en `main`
- ★ Lógica mezclada
- ★ Dificultad para mantener el código

# Estructura

---

-  model
-  controller
-  service
-  dao
- App

# Estructura

---

## Modelo (Model)

- ★ Responsabilidad:
  - Representa los datos
  - Contiene reglas de negocio simples
  - No sabe nada de base de datos ni consola

# Estructura

---

## Controller

- ★ Responsabilidad:
  - Recoge datos
  - Controla el flujo
  - Llama al Service
  - Controla excepciones
  - Devuelve mensajes

*En aplicaciones reales, la recogida de datos estaría en la View, pero en consola lo simplificamos.*

# Estructura

---

## Service

- ★ Responsabilidad:
  - Contiene la lógica de negocio
  - Valida datos, no los pide ni imprime nada
  - Aplica reglas
  - Coordina operaciones

*Se puede decir que el Service es el cerebro de la aplicación.*

# Estructura

---

## DAO (Data Access Object)

★ Responsabilidad:

- Acceso a datos
- Leer / guardar
- JSON / fichero (.dat) / base de datos
- No contiene lógica de negocio
- Solo persistencia

# Estructura

---

## App (Vista)

★ Responsabilidad:

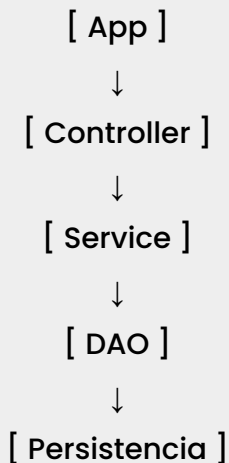
- Menú
- Punto de entrada
- Orquesta la acción de ejecutar

*No es exactamente la View (vista) completa, es el ejecutable.*

*En entorno profesional, la View estaría separada del Controller. En entorno formativo, simplificamos la interacción por consola.*



# Flujo real de la aplicación



Cada capa solo habla con la inmediatamente inferior



PROMETEO