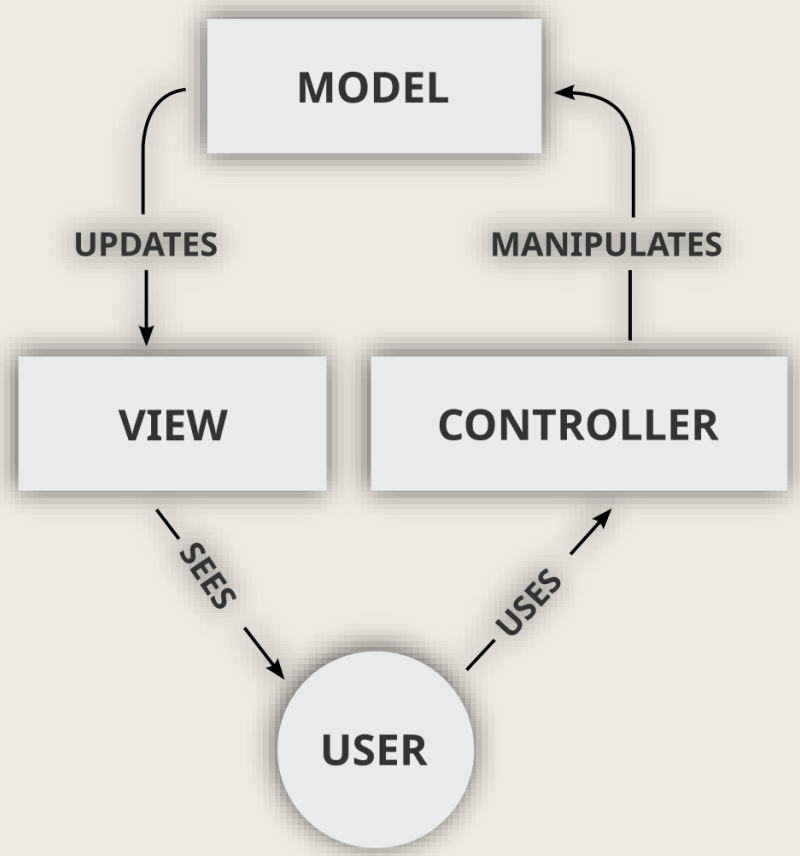
Tema 2. Arquitectura Hexagonal

¿De dónde partimos?

Arquitecturas basadas en MVC y capas tradicionales.

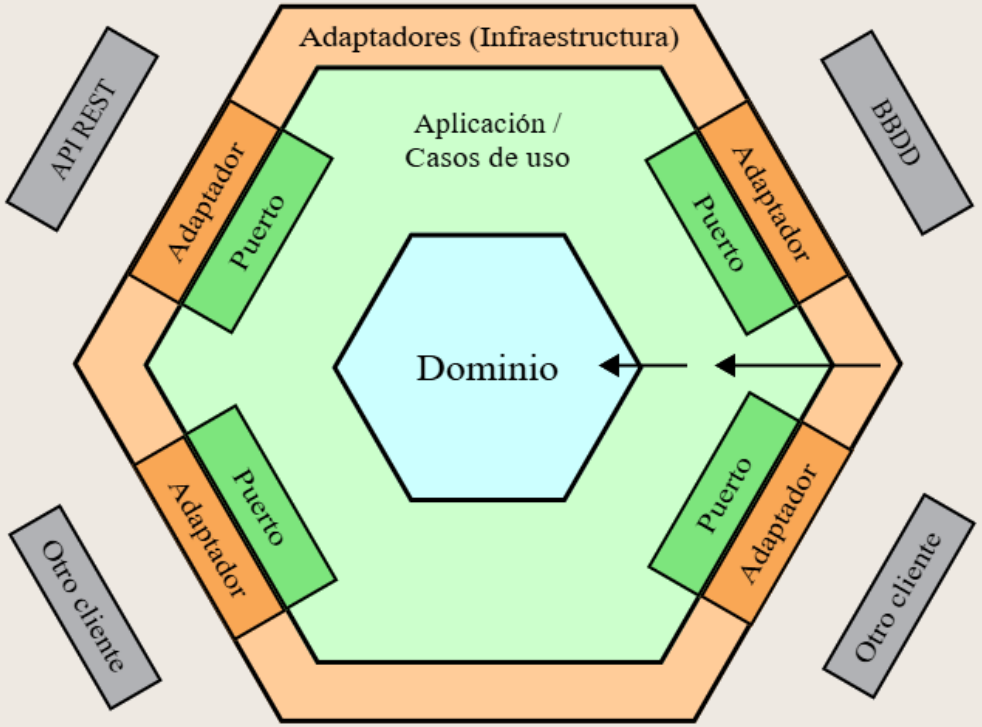
- Entidades de base de datos mezcladas con clases de dominio.
- Lógica de negocio repartida entre controladores, servicios y vistas.
- Difícil de testear.



¿Qué es la arquitectura hexagonal?

Arquitectura de puertos y adaptadores.

- El dominio se sitúa en el centro.
- Las dependencias siempre apuntan hacia el dominio

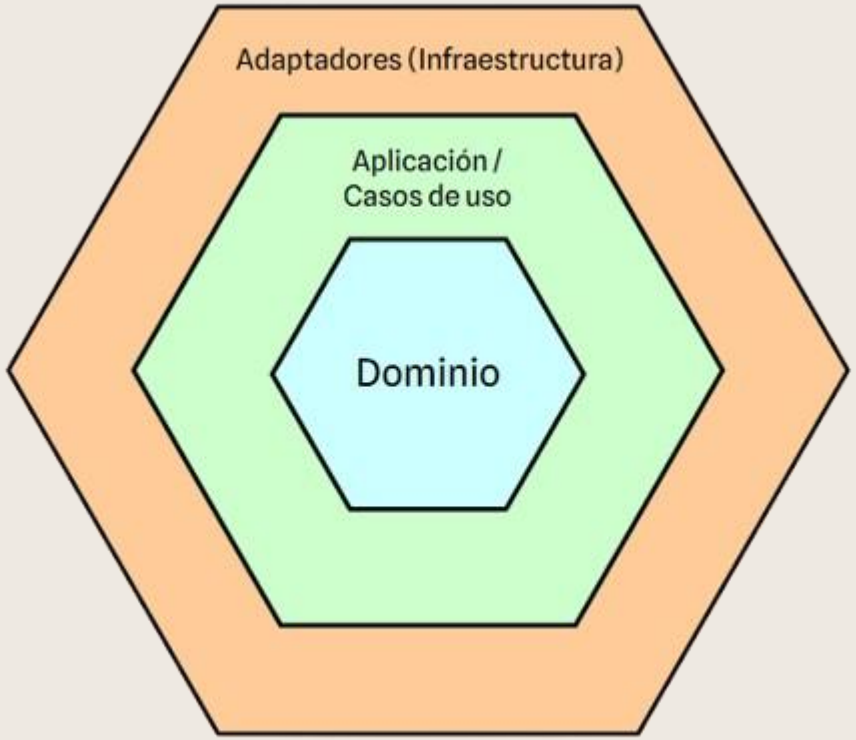


¿Qué ventajas me ofrece?

- La lógica de negocio queda aislada de frameworks y tecnología.
- Facilita los cambios en la infraestructura sin afectar al dominio.
- Encaja perfectamente con DDD.

Las capas principales

- Dominio
- Aplicación
- Infraestructura (Adapters)



Capa de infraestructura: Adapters

Capa que conecta nuestra aplicación con el exterior.

- **api**
  APIs que expone mi aplicación.
- **persistence**
  Conecta con B.D.
- **provider**
  Clientes a servicios externos para interactuar con ellos.
- **consumers**
  Consumidores de eventos proyecto.

Capa de aplicación: Application

Capa responsable de orquestar los casos de uso de la aplicación.

- **port**
  Define las interfaces que necesita la capa de aplicación.
- **service**
  Agrupa funcionalidad reutilizada por los casos de uso.
- **usecase**
  Contendrá los casos de uso.

Capa de dominio: Domain

Se encarga de expresar las reglas de negocio del sistema.

- **model**
  El modelo del dominio.
- **factory**
  Factorías para la creación de objetos del dominio.
- **event**
  Eventos que lanza el dominio cuando ocurre algo.
- **exception**
  Excepciones de dominio.

Arquitectura hexagonal y DDD

- DDD pone el foco en el dominio.
- Hexagonal protege ese dominio de la tecnología.
- Ambas buscan aislar la lógica de negocio.
- Se complementan de forma natural.

DDD nos dice qué es importante. La arquitectura hexagonal nos ayuda a protegerlo


UNIVERSIDAD DE MURCIA
1272