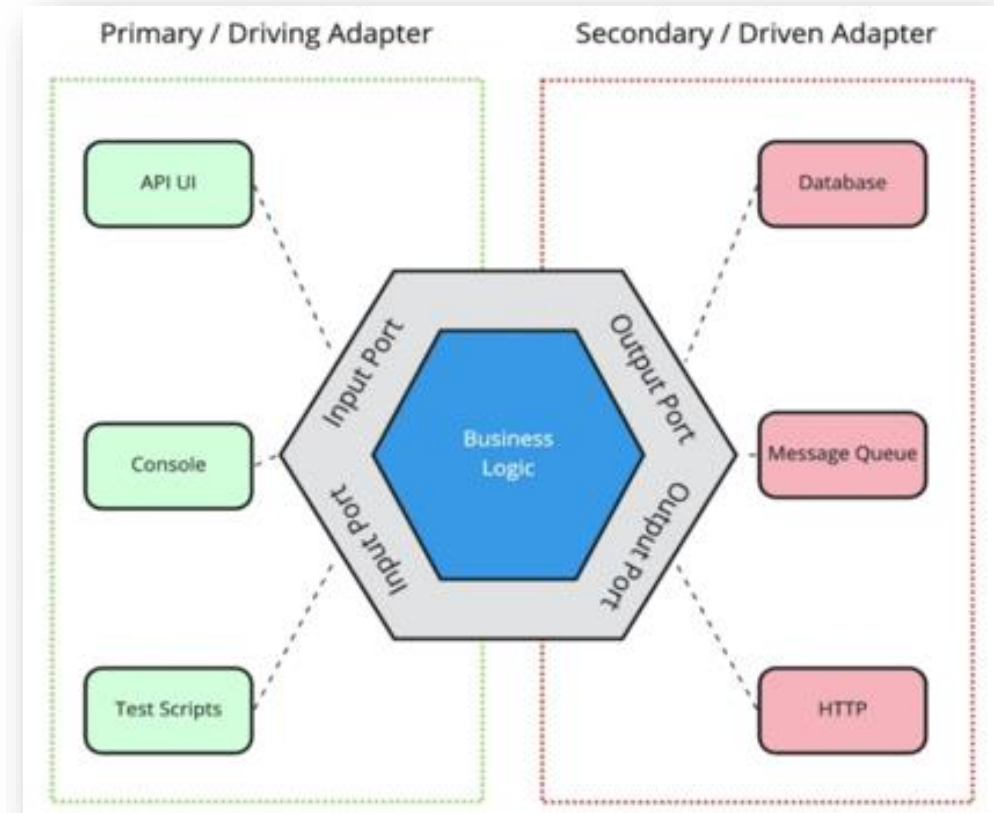
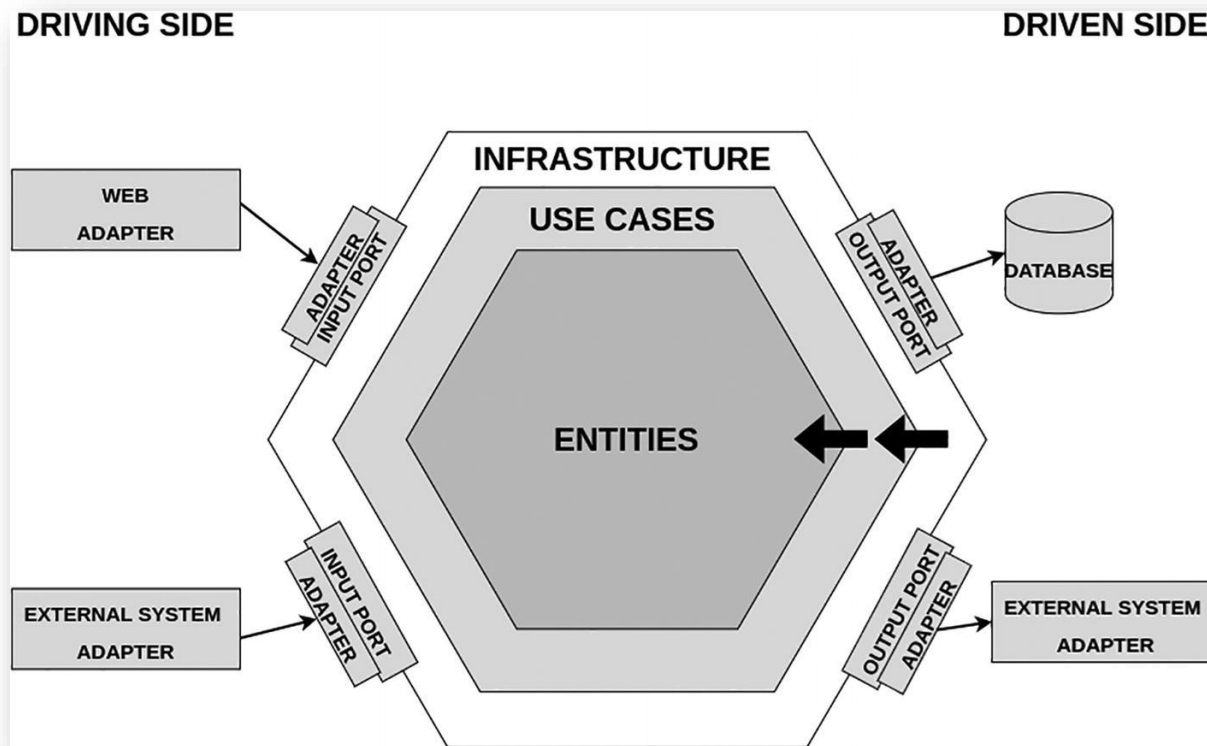




Arquitectura de Aplicaciones Web

Arquitectura Hexagonal

La arquitectura en capas ofrece a los desarrolladores una forma sencilla de crear una aplicación y comprender la separación de roles o responsabilidades. En 2005, Alistar Cockburn propuso una nueva alternativa que encapsula o mantiene agnóstica toda comunicación externa con bases de datos o sistemas externos



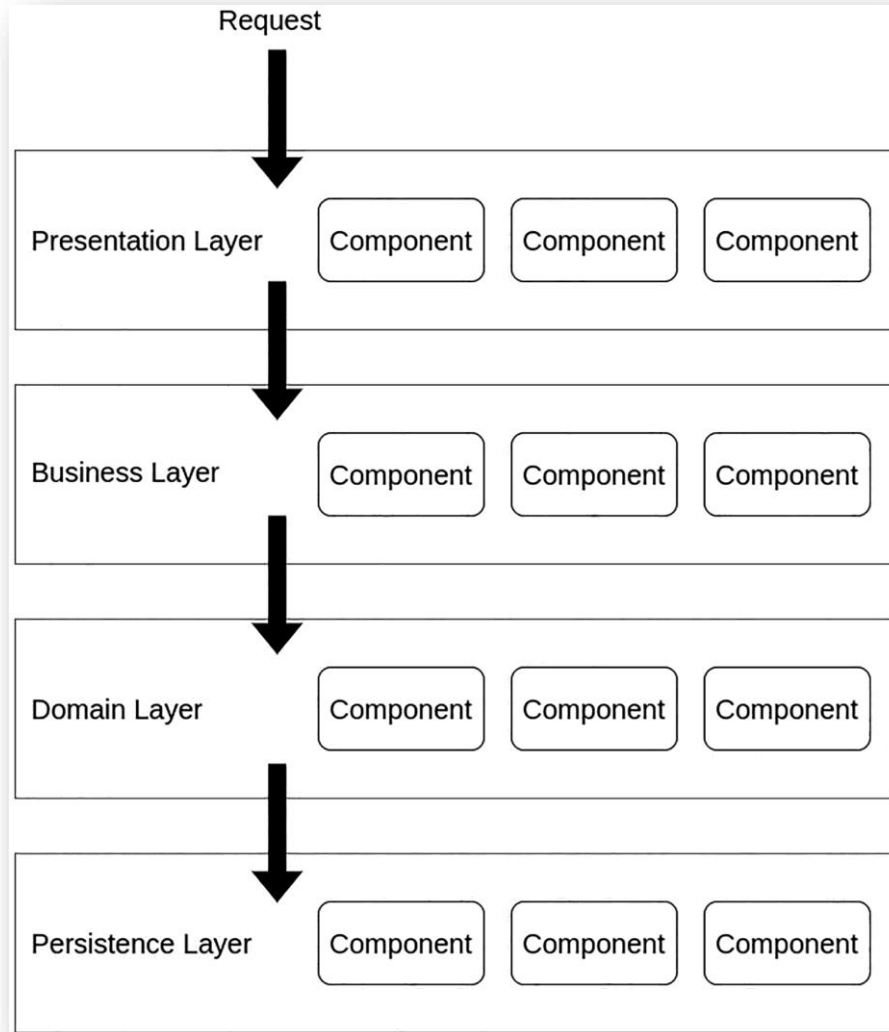
Beneficio

- Usado mayormente en aplicaciones grandes y complejas.
- Puedes cambiar o intercambiar adaptadores sin problemas.
- El dominio y los casos de uso son puros, por lo que puede comprender la lógica comercial.
- Puede cambiar la implementación de los puertos con fines de prueba.

Inconvenientes

- Implementar algunos frameworks es difícil porque puede cometer errores con lo que considera adaptadores/puertos .
- Introducir indirección para encontrar la implementación de cada puerto.
- Ninguna directriz explica todas las cosas sobre la organización de los proyectos, por lo que cualquiera puede introducir algo mal.

Arquitectura de Capas - Layers



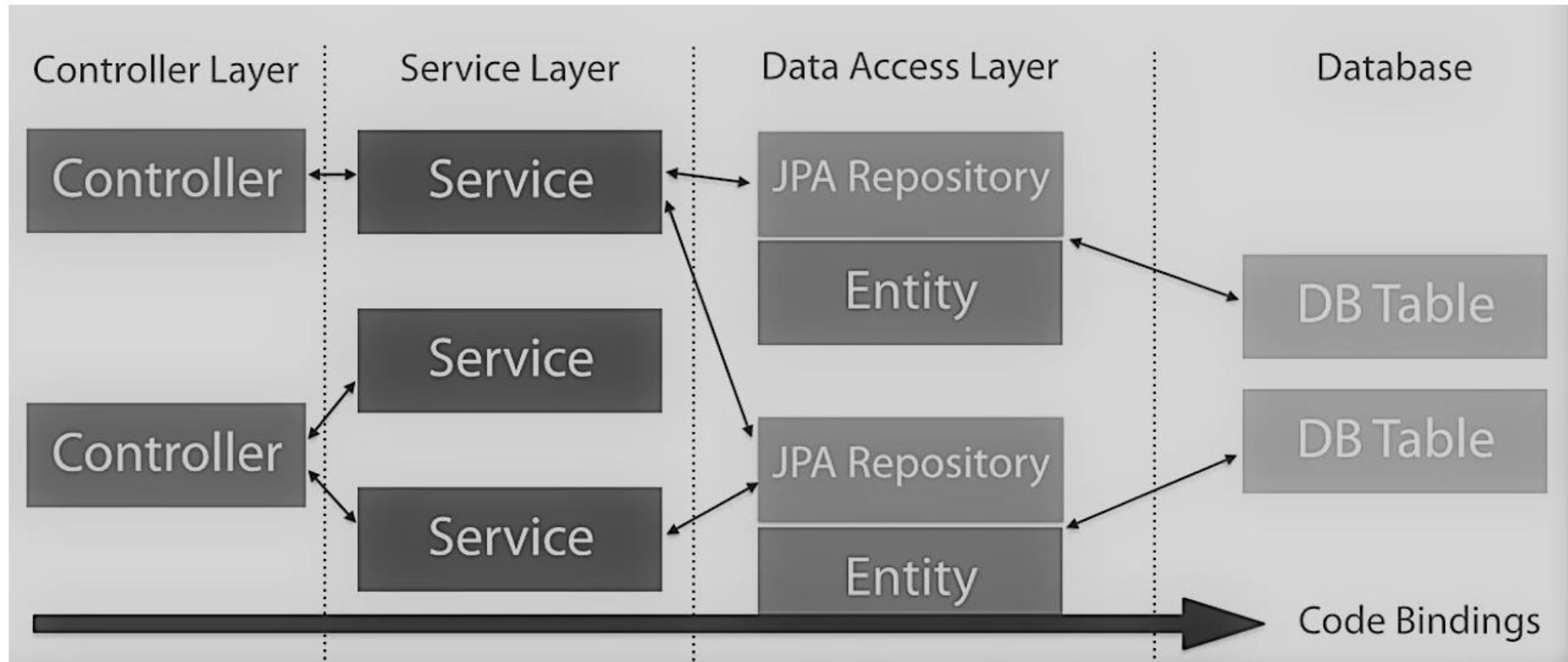
Beneficios

- Es simple de implementar y mantener porque solo tiene una jerarquía de capas.
- Cada capa tiene una sola responsabilidad y no sabe cómo implementar la lógica de las otras capas, por lo que puede introducir modificaciones en la lógica de las capas que podrían no afectar directamente a otras.
- La estructura es más o menos la misma en todos los proyectos.

Inconvenientes

- Al ocultar los casos de uso, es difícil saber dónde están conectadas todas las clases u objetos para representar un caso de uso. *
- Dependiendo del tamaño y la cantidad de capas, puede tener problemas con la escalabilidad porque puede tener varias clases/interfaces en el mismo paquete o capa .

Backend Layers of Spring



Persistencia

- La persistencia debe lograr que los datos que maneja o genera una aplicación **se almacenen fuera de esta para su uso posterior.**