

Watch: [Flujo de una petición en Arquitectura Hexagonal](#)

06:31

Arquitectura Hexagonal

Como hemos comentado en videos anteriores, DDD se sustenta sobre **Arquitectura Hexagonal**, por lo que vamos a repasar brevemente de qué se trata y cómo se estructura. Recordad que podéis profundizar mucho más en el curso de [Arquitectura Hexagonal](#) que tenemos publicado

Capas y Regla de Dependencia

Partiendo de la premisa de que las capas se conocen entre sí desde fuera hacia dentro, en esta arquitectura se identifican las siguientes capas:

- **Infraestructura:** Aquí se encuentra el código que o bien toca **Entrada/Salida** (BD, Ficheros...), o bien que esté **acoplado** a clases de un vendor externo.

- **Aplicación:** Representa los **casos de uso**, empapándose de toda la semántica de negocio. Actuará como una barrera transaccional en términos (tanto en términos de BD como de publicación de eventos)
- **Dominio:** Representa el **modelado** de nuestro negocio (Principalmente *Value Objects* y *Entidades*), en esta capa encontraremos también los *servicios de dominio* (Profundizamos más sobre ellos en el curso de Arquitectura Hexagonal)

La regla de dependencia que existe entre estas capas implica como veíamos que la capa de Dominio es conocida por la de Aplicación, y ésta es conocida por la capa de Infraestructura, de tal modo que si tenemos la necesidad de realizar cambios en Infraestructura (Distinto framework, ORM, etc) no tendrá afectación en el resto (no tenemos ningún acoplamiento)

Flujo de una Petición

El flujo parte de una petición recibida por el Controller (Infraestructura) desde el que se enviaría al Application Service.

En este punto plantearíamos dos alternativas, una de ellas es que el Application Service sea quien montara los Value Objects a partir de los datos primitivos recibidos desde el Controller. Por otro lado, tal como vimos en el curso de [CQRS](#) también podríamos hacer que ese mapeo se llevara a cabo a través un Bus (Command Bus o Query Bus) que los enviaría al Application Service

Desde el Application Service se llamaría a los distintos Services, Models y Repositories.

Es importante señalar aquí que la capa de aplicación se comunicaría con una Interfaz del repo de la capa de dominio, mientras que la implementación de dicha Interfaz estará dentro de la capa de Infraestructura siguiendo el **Patrón Adaptador**

¿Alguna duda?

En el siguiente video de la lección veremos Nuestro primer caso de uso: Student Sign Up Si tienes dudas o sugerencias sobre esta lección recuerda que puedes abrir una nueva discusión ¡Justo aquí abajo!

