



DOMAIN DRIVEN DESIGN (DDD)

Arquitectura de microservicios basada en DDD

Axel Jordano Morales Utrera
axeljordanomu@gmail.com

Introducción:

Dentro de la arquitectura de microservicios basada en DDD, la capa del modelo de dominio juega un papel fundamental. En esta capa, se representan los conceptos y reglas de negocio de la aplicación. Aquí es donde se definen las entidades del dominio, los objetos de valor y los agregados. Las entidades del dominio encapsulan tanto los datos como el comportamiento relacionado con un concepto de negocio específico, mientras que los objetos de valor representan conceptos inmutables y sin identidad propia. Los agregados son conjuntos coherentes de entidades y objetos de valor que se tratan como una unidad en el modelo de dominio.

Una de las premisas importantes en DDD es la independencia del modelo de dominio respecto a la capa de infraestructura. Esto significa que las entidades del dominio no deben tener dependencias directas con frameworks de acceso a datos o tecnologías de persistencia. Idealmente, estas entidades deben ser objetos simples y sin acoplamiento a la infraestructura, lo que permite una mayor flexibilidad y reutilización.

La capa de aplicación, por otro lado, se encarga de definir las tareas y responsabilidades que el software debe realizar. Es responsable de coordinar las interacciones entre los objetos de dominio y de manejar las solicitudes y comandos recibidos desde el exterior. Esta capa se mantiene delgada y no contiene reglas de negocio ni conocimiento del dominio. Su objetivo principal es orquestar y dirigir la ejecución de las tareas utilizando los objetos de dominio y delegando en ellos la aplicación de las reglas de negocio. También puede manejar la comunicación con otros sistemas mediante el uso de APIs web u otros mecanismos de integración.

La capa de infraestructura es la encargada de persistir los datos del modelo de dominio y proporcionar los mecanismos de acceso a datos necesarios. Aquí es donde se implementan patrones como el Repositorio, que se encarga de la comunicación con el almacenamiento de datos subyacente, como una base de datos relacional o un servicio de almacenamiento en la nube. Es importante tener en cuenta que la capa de infraestructura debe mantenerse separada del modelo de dominio para evitar la contaminación de la lógica de negocio con detalles de implementación.

En cuanto a las comunicaciones entre las capas, es esencial establecer una clara delimitación de dependencias. Cada capa debe depender únicamente de las capas inferiores y no tener conocimiento de las capas superiores. Esto permite mantener una estructura modular y facilita el mantenimiento y la evolución del sistema. Al implementar las capas como bibliotecas de clases independientes, se pueden controlar y gestionar mejor las dependencias entre ellas.

En resumen, el diseño de microservicios orientados a DDD implica mantener límites de contexto pequeños y cohesivos, y definir claramente las responsabilidades de cada capa en el sistema. El modelo de dominio se enfoca en representar los conceptos y reglas de negocio, mientras que la capa de aplicación se encarga de coordinar y orquestar las tareas utilizando los objetos de dominio. La capa de infraestructura se ocupa de la persistencia de datos y el acceso a los recursos externos. Mantener una buena separación y coherencia entre estas capas es fundamental para lograr un diseño robusto y escalable en un entorno de microservicios basado en DDD.

Fuente:

Microsoft. Domain-Driven Design (DDD) Oriented Microservice Architecture Guide. Recuperado de <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>