

1. Arquitetura em Camadas

Decisão:

A solução foi dividida em quatro camadas principais: API, Application, Domain e Infrastructure.

Justificativa:

- **Separação de Responsabilidades:** Cada camada concentra um conjunto específico de responsabilidades, facilitando a manutenção, escalabilidade e a possível realização de testes unitários, tudo seguindo o padrão Clean Code/DDD.
- **Isolamento de Mudanças:** Alterações em uma camada (por exemplo, na lógica de persistência ou na implementação dos serviços) não afetam diretamente as demais, tornando o sistema mais robusto e modular.

2. Uso do Padrão CQRS (Command Query Responsibility Segregation)

Decisão:

A camada de Application foi estruturada utilizando o padrão CQRS, separando operações de leitura (queries) e escrita (commands).

Justificativa:

- **Performance:** Em cenários de alto volume de requisições, a separação das operações de consulta e atualização permite otimizar cada fluxo de trabalho individualmente.
- **Escalabilidade:** Possibilita que a parte de leitura seja escalada separadamente da parte de escrita, atendendo ao requisito de alta performance.
- **Manutenibilidade:** Cada comando e query possui um handler específico, o que facilita a implementação e o isolamento de regras de negócio.

3. Persistência e Banco de Dados

Decisão:

- Utilização do SQL Server 2016 ou superior como banco de dados.
- Mescla de ORMs para operações de consulta e com stored procedures para operações de criação, atualização e remoção.

Justificativa:

- **ORM para Consultas Entity Framework:** O uso do ORM Entity Framework permite a escrita de queries de forma mais rápida e segura, aproveitando o mapeamento objeto-relacional e otimizando a leitura dos dados.
- **ORM Dapper com Stored Procedures para Operações:** Ao utilizar stored procedures para operações de escrita (CRUD) com Dapper, a solução se beneficia de performance aprimorada, já que a lógica de manipulação dos dados é executada no lado do banco de dados.
- **Flexibilidade:** Essa abordagem mista permite aproveitar o melhor de ambos os mundos, otimizando a performance e segurança de consultas e garantindo a robustez e rapidez das operações de escrita.