

Perfeito, Rafael - bora deixar isso profissional de verdade.

E relaxa: vou te explicar cada parte de um jeito que você consiga visualizar a arquitetura inteira na cabeça antes de escrever uma linha de código.

A gente pode montar isso como se fosse um projeto financeiro real, com camadas bem separadas, testes cobrindo tudo e uma API robusta.

Antes de escolher por onde começar, deixa eu te mostrar o que cada parte faz, porque isso ajuda muito a decidir o próximo passo.

1. Camadas (Domain, Application, Infra, API)

Essa é a arquitetura mais usada em projetos sérios de C#.

Domain

Onde ficam as regras de negócio puras.

Exemplo: ContaBancaria, Transacao, validações, exceções de domínio.

Sem banco, sem API, sem nada externo.

Application

Onde ficam os casos de uso.

Exemplo: CriarConta, Depositar, Sacar, ConsultarSaldo.

Aqui você orquestra o domínio e chama repositórios.

Infra

Onde ficam as implementações técnicas:

- Entity Framework
- Repositórios

- Mapeamentos
- Acesso ao banco
- Serviços externos

API

Onde ficam os controllers, endpoints, DTOs, validações de entrada.

2. Repositórios

São interfaces no Domain e implementações no Infra.

Exemplo:

```
public interface IContaRepository  
{  
    Task<ContaBancaria> ObterPorId(int id);  
    Task Criar(ContaBancaria conta);  
    Task Atualizar(ContaBancaria conta);  
}
```

E no Infra você implementa usando EF Core.

3. Banco de dados (EF Core)

Aqui você cria:

- DbContext
- DbSet
- Mapeamentos
- Migrations

E conecta tudo ao SQL Server ou SQLite.

4. Testes de integração

Testam a API ou o repositório de verdade, com banco em memória ou SQLite.

Exemplo: Criar conta -> Depositar -> Consultar saldo -> Verificar resultado.

5. Testes com banco em memória

Usam o UseInMemoryDatabase do EF Core para testar repositórios sem precisar de SQL Server.

6. Autenticação JWT

Aqui você adiciona:

- Login
- Geração de token
- Autorização por roles
- Proteção de endpoints

7. Swagger completo

Documentação automática da API com:

- Descrição dos endpoints
- Exemplos
- Autenticação JWT integrada
- Versionamento

Agora a pergunta importante:

Para onde você quer ir agora?

Aqui estão três caminhos possíveis:

1. Começar pela arquitetura (Domain + Application + Infra + API)

2. Começar pelo banco (EF Core + Repositórios)
3. Começar pelos testes (TDD + testes de integração)

Rafael, qual desses caminhos você quer seguir primeiro - Arquitetura, Banco, ou Testes?