

“Resumo” do vídeo <https://www.youtube.com/live/tetVY6jmlnM?feature=shared>

Para aplicar os princípios de SOLID ele utiliza algumas camadas de entities, use cases e i/o com dependências do express, jest, axios, nodemon, pg-promise e typescript.

Para abordar melhor o conteúdo farei um resumo das camadas utilizadas ao longo do vídeo.

O arquivo main do projeto foi usado para configurar as dependências de import e iniciar o servidor. Além dele, ao longo do vídeo foram criados as camadas de application, domain, infra e test.

Na camada de application, onde ele criou os arquivos CreateTransaction.ts e GetTransaction.ts para guardar a lógica da aplicação. Sendo assim, eles ficaram dependentes apenas da camada de domínio (domain).

Dentro da camada de domain, há outras subcamadas: Entity com os arquivos Transaction e Installment que serviram para encapsular o estado e o comportamento relacionado ao conceito de negócio e portanto ficaram independente de mecanismo de persistência. A outra subcamada usada foi Repository que guardou o arquivo TransactionRepository onde ele definiu a interface para operações relacionadas a transações.

Ele também utiliza uma camada infra com as subcamadas de api, database e repository. api contém os arquivos ExpressAdapters.ts e Router.ts que lidaram com a exposição dos endpoints HTTP. Database contém os arquivos de Connection.ts e PostgreSQLAdapter.ts que implementam as configurações da tabela feita em SQL no projeto. E a subcamada de repository do infra contém os arquivos >TransactionMemoryRepository.ts e >TransactionDatabaseRepository.ts que contém as implementações concretas do projeto.

E por último ele utiliza uma camada de test com os arquivos >CreateTransaction.test.ts, >main.test.ts, >Transaction.test.ts que serviram para testar as camadas de application, domain, infra e como a relação entre elas.