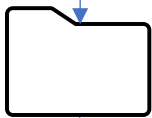


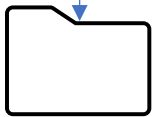
1 Aplicação



2 Infra



3 Domain



Levando em consideração que o sistema evoluirá para um sistema de estoque completo, defini DDD com Repository Pattern pela sua facilidade de manutenção, reuso e escalabilidade.

Criei a camada de domínio como uma class library para manter a organização do projeto e para e ser possível que outra biblioteca à importe.

Criei uma entidade base somente com o id pensando em futuramente alguns atributos como data de criação, alteração ou algum outro comum para todas as entidades.

Criei a entidade de produto.

Criei a Camada de Infra como class library o mesmo caso da do domínio.

Criei a base config para a entidade base com as configurações de tipo do entity framework.

Criei a classe de configuração do produto passando a tabela e propriedade de cada coluna.

Criei um contexto base para comunicação com o banco de atribuindo uma conexão, utilizando o migration e na criação do modelo usei o entity framework com a configuração do produto.

Criei um mapeamento do produto para utilizar o AutoMap do DapperExtensions.

Utilizei um repositório base que instancia o contexto no seu construtor e inicializa a configuração do AutoMapper, além disso foi criado os métodos base do CRUD usando tipo

genérico para ser usado por qualquer entidade que realize essas mesmas configurações, os métodos do CRUD utilizam o Dapper que foi especificado no escopo da task.

Criei uma interface do repositório base para ser encapsulado e usado nos Controllers.

Ajustei o Controller do produto para buscar um produto por id, inserir um produto, alterar, deletar e também listar todos, usei a interface base repository passando o produto para usar os métodos.

Executei o migration no console para gerar o banco e criar a tabela e colunas.

O arquivo do banco se encontra em \Api\Database.db.

Fiz ajustes no startup para assegurar que o banco existe no contexto base adicionado algumas dependências para o serviço comunicar com o SQLite, além de injetar a dependência do repositório base e adicionar configurações do dapper para o sqlite além do CORS para ser possível aceitar requisições de outra origem.

Testei com o Postman os métodos e iniciei o projeto do React.

Utilizei o Typescript por ter a possibilidade de tipagem dos models.

Criei um arquivo api para instanciar o axios e gravar o endpoint padrão, escolhi pela facilidade de uso, pelas features e compatibilidade nos navegadores mais utilizados.

Fiz as rotas e criei as páginas de listar os produtos com os botões remover, editar e incluir um novo, e página de adicionar/editar o produto.