

Projeto de Gerenciamento de Estoque

O projeto está contido dentro da pasta "GerenciamentoEstoque". Dentro dela podemos encontrar as pastas "Estoque.API", referente ao projeto do backend, e "WebApp", referente ao projeto do frontend.

- **Back end:**

O backend foi desenvolvido em C# utilizando o framework .NET CORE 3.0. A estrutura escolhida para o projeto foi o ASP.NET Core MVC. O framework de persistência utilizado neste projeto é o Entity Framework Core. O banco de dados foi feito no SQL Server e criado a partir do Migration.

O framework .NET CORE utiliza uma arquitetura de microsserviços que possibilitam uma combinação de tecnologias, possui alto desempenho e sistemas escalonáveis devido ao tempo de execução do servidor. Estes são alguns dos motivos por ter sido utilizado nesse projeto.

A escolha da arquitetura ASP.NET Core MVC foi feita devido a sua separação de responsabilidades adequados para os aplicativos Web. Ela também é multi-plataforma, ou seja, roda em qualquer ambiente, facilita as publicações e tem ótimo suporte para containers.

Entity Framework Core é leve, extensível e permite ao desenvolvedor trabalhar com banco de dados usando objetos .NET, o que elimina a necessidade de uma grande parte de código de acesso a dados.

O migrations é uma forma de versionar as alterações do banco de dados juntamente com as alterações da aplicação. Algumas de suas vantagens incluem a manipulação de banco de dados simplificada e estruturada, além da fácil identificação do estado atual de dados. Ao rodar a aplicação qualquer alteração ou criação de banco de dados necessária é feita sem grandes esforços.

Organização do código:

O backend está localizado na pasta "GerenciamentoEstoque\Estoque.API". Dentro dela o projeto foi subdividido em 3 projetos do tipo Class Library que se diferenciam de acordo com suas responsabilidades:

- Estoque.API: nela está localizada as controllers, os arquivos "appsettings.json", "Program.cs" e "Startup.cs";
- Estoque.Domain: nela encontram-se as configurações do Entity, do Migration para a criação de banco de dados, os models usados no projeto e também as classes de validações;
- Estoque.Service: responsável pela camada de negócio do sistema.

Passos para rodar o backend:

Para a execução do código e criação do banco de dados é necessária a criação de uma instância de banco em SQL Server e a substituição de seus dados no “DefaultConnection” da “ConnectionStrings” do arquivo “GerenciamentoEstoque\Estoque.API\Estoque.API.appsettings.json”. Ao rodar o projeto o banco de dados será criado automaticamente no local indicado da ConnectionStrings.

● Frontend:

O frontend foi desenvolvido em JavaScript - Angular 8 com a utilização da biblioteca do Angular Material.

A escolha do Angular 8 foi feita devido a sua aplicação mais performática, permitindo diversos modelos de arquitetura, inclusive microsserviços. A integração com serviços Resfull é bem fácil deixando a aplicação totalmente assíncrona e rápida e com ele é possível utilizar SPA (Single Page Application), uma vez que sua configuração de rotas é muito simples e dinâmica.

O projeto contempla o serviço PWA, embora eu não tenha conseguido testá-lo.

Organização do código:

O frontend está localizado na pasta “GerenciamentoEstoque\WebApp\gerenciamento-estoque”. Dentro dela foram criadas as seguintes pastas e arquivos:

- app\app-root-component: componente root que será renderizado ao abrir a aplicação;
- app\estoque-add-edit: componente de adição e edição de estoque. Ambas funcionalidades foram desenvolvidas no mesmo componente;
- app\estoque-list: componente de listagem de estoque de produtos;
- app\models: models utilizados no projeto;
- app\services: serviços utilizados no projeto. Por enquanto só tem o de estoque;
- app\shared\components: componentes compartilhados em todo o projeto;
- app\shared\directives: diretivas compartilhadas em todo o projeto;
- app\shared\services: serviços compartilhados em todo o projeto;
- material.module.ts: módulo separado para os imports e exports dos módulos do angular material;
- assets\mocks: mock dos objetos de retorno para testar o frontend sem a comunicação com o backend.

Passos para rodar o frontend:

Para sua execução é necessário dar o comando “npm install” e em seguida “ng server”.
O projeto será rodado na url <http://localhost:4200/>