

Sistema Gerenciamento de Estoque

Evolutiva do projeto Marvel/DC Store

Diego Oliveira Froes



Objetivo

O objetivo principal do projeto foi desenvolver um sistema que forneça, de forma intuitiva, funcionalidades para o gerenciamento de estoques de produtos.



MVP - Produto Mínimo Viável

Para a entrega do MVP do sistema, foi solicitado que o mesmo deveria apresentar algumas características, como:

- ▶ Uma tela que exiba todos os produtos do estoque;
- ▶ Nessa tela, contemple:
 - ▶ Nome, Quantidade e Valor Unitário do produto;
 - ▶ Funcionalidades de Adicionar, Editar e Excluir produto;
 - ▶ As funcionalidades de adicionar e editar sejam feitas em uma tela distinta.



Desenvolvimento

O aplicativo web 'Marvel/DC Store' foi desenvolvido mediante os recursos:

- ▶ ASP.NET Core 2;
- ▶ Linguagem de programação C#;
- ▶ React para o Front-end;
- ▶ Entity Framework Core com Code First Migrations;
- ▶ Banco de dados SQL Server.



ASP.NET Core

O ASP.NET Core 2 é um framework completamente novo baseado em .NET, a qual apresenta-se bem arquitetada sendo reescrita do zero.

- ▶ Manteve a arquitetura MVC implementando algumas variantes;
- ▶ Por ter mantido o MVC, a Controller pode ser isolada, sendo assim, facilmente testável por framework como xUnit.
- ▶ Construída através de componentes e interfaces independentes, o que é possível estender seu comportamento por outros da sua própria implementação.



ASP.NET Core

- ▶ Sendo multiplataforma, é possível rodar em ambientes Windows, MacOS e distribuições Linux;
- ▶ Open Source;
- ▶ Pipeline de solicitação HTTP leve, modular e de alto desempenho;
- ▶ Boa documentação além de diversos cursos e tutoriais;
- ▶ Segundo estudos com desenvolvedores mostram que o .NET Core apresenta futuro promissor na categoria frameworks, o C# ocupa a quarta posição na categoria linguagens de programação e o Visual Studio lidera como IDE mais popular.



Linguagem C#

Utilizamos a linguagem de programação C# por apresentar sintaxe simples, curva de aprendizagem e redução de complexidades quando comparado com outras linguagens (Java, C, C++).

Ela segue os pilares da POO e apresenta recursos de delegações, expressões lambdas, acesso direto à memória, fortemente tipada, suporte a DLL's, COM e COM+.

Vale ressaltar o AsyncFeature que, em linhas gerais, fornece métodos assíncronos, permite que usuários não sejam bloqueados enquanto alguns processos estão sendo executados em segundo plano.



Utilizamos o React por apresentar uma biblioteca JavaScript declarativa, eficiente e flexível para a criação de interfaces de usuário. Em geral, bibliotecas são mais flexíveis e menos complexas do que frameworks, como exemplo Angular.

- ▶ Utilização de componente, o que facilita a reutilização de código;
- ▶ O virtual DOM fornece maior dinâmica e eficiência por armazenar todos os nós do DOM, promovendo alterações somente quando necessário.
- ▶ Compatível com plugins: npm, Webpack, commonJs, AMD, Gulp e etc.



Referência para quem trabalha com desenvolvimento front-end, JavaScript é a linguagem de programação que mais evoluiu nos últimos tempos.

Baseamos o desenvolvimento do Front na linguagem TypeScript a qual é definida como um superconjunto de JavaScript. A forma que fora construída, permite desenvolver projetos mais bem estruturados e mais fáceis de manter.

Diante disso, conclusão rápida e precisa, recursos de refatoração e feedback imediato faz do TypeScript a linguagem ideal para JavaScript em grande escala.



Entity Framework Core

O Entity Framework Core é uma versão leve, extensível, de software livre e multiplataforma da tecnologia de acesso a dados do Entity Framework.

O EF Core pode servir como um ORM (Mapeador de Objeto Relacional), permite trabalhar com banco de dados usando objetos .NET, dessa forma, elimina a necessidade de grande parte de código referente a gerenciamento de dados.

Utilizamos o recurso Migration como Code First para automatizar a geração do modelo de dados. Isso permite atualizar de forma incremental o esquema do banco de dados em sincronia com modelo de dados do aplicativo.



SQL Server

Utilizamos o SQL Server como nosso banco relacional por apresentar recursos avançados (robustez) que facilitam a atualização dos dados e garantem a confiabilidade das informações armazenadas.



A solução do sistema está dividida em:

- ▶ Projeto Web (MarvelOrDCStore);
- ▶ Projeto para camada de Negócio (BusinessLibrary);
- ▶ Projeto para camada acesso a dados (DataAccessLibrary).

ConnectionString

Encontra-se em DataAccessLibrary > Models > MarvelDCStoreDBContext.cs para ser alterada conforme necessário.



Considerações Finais

Este trabalho desenvolveu um aplicativo web para sistema de gerenciamento de estoque. Diante do contexto, tempo de entrega e conhecimento, foram tomadas as decisões supracitadas referente a arquitetura do projeto com intuito de prover maior performance, organização e expansão do sistema.



Trabalhos Futuros

Contudo, para trabalho futuro e visando maior escalabilidade do sistema, vale ressaltar seguintes pontos:

- ▶ Migrar o aplicativo para ASP.NET Core 2.2 ou 3.0;
- ▶ Aplicar React com Redux, a qual, em linhas gerais, soluciona o problema de compartilhamento de estados entre componentes, tornando-o unidirecional.
- ▶ Utilizar TDD - Desenvolvimento Orientado por Testes.



Referências



<https://medium.com/fabridamazio-pt/porque-voc>



<https://www.brunobrito.net.br/asp-net-core/>



<https://medium.com/@wiliambuzatto/asp-net-core-mvc-principais-benef>



<https://revistapegn.globo.com/Como-comecar/noticia/2016/06/5-passos-para-fazer-um-produto-minimo-viavel-mvp.html>



<https://www.devmedia.com.br/introducao-a-linguagem-csharp/27711>



<https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>



Referências



<https://www.impacta.com.br/blog/2019/02/27/saiba-o-que-e-react-e-entenda-como-usa-lo-no-desenvolvimento/>



<https://medium.com/by-vinicius-reis/o-que-e-react-ng2-auleria-vue-e34b0c77b5a1>



<https://medium.com/@oieduardorabelo/typescript-o-guia-definitivo-1a63b04259cc>



<https://tableless.com.br/react-o-que-e-e-como-funciona-essa-ferramenta/>



<https://www.devmedia.com.br/top-10-linguagens-de-programacao-mais-usadas-no-mercado/39635>



http://www.macoratti.net/17/05/efcore_ini1.htm



Referências



<https://docs.microsoft.com/pt-br/ef/core/>



<https://docs.microsoft.com/pt-br/aspnet/core/data/ef-mvc/intro?view=aspnetcore-2.2>



<https://docs.microsoft.com/pt-br/ef/core/managing-schemas/migrations/>



http://www.macoratti.net/12/09/ef4_mig1.htm



<https://www.opservices.com.br/banco-de-dados/>



<https://www.impacta.com.br/blog/2017/12/22/entenda-de-uma-vez-por-todas-o-banco-de-dados-sql-server/>

