
Documentação do software para Estoque

Autor: Robson Faria de Melo

Sumário

1. Introdução	3
2. História do usuário	3
3. Tecnologias Utilizadas	3
3.1. Backend:	4
3.1.1. SQLite	4
3.1.1.1. Estrutura do banco de dados:	4
3.1.2. Entity Framework.....	4
3.1.3. WCF (Windows Communication Foundation)	4
3.2. Frontend:	4
3.2.1. ASP.NET Web Application – MVC (Model / View / Controller)	4
3.3. Justificativa para utilização destas tecnologias:	4

1. Introdução

Este projeto foi criado com o intuito de realizar um teste prático para a empresa DTI digital onde nele foi desenvolvido uma aplicação web para realizar consultas a um estoque de produtos diversos.

2. História do usuário

Eu como estoquista desejo cadastrar/alterar e excluir produtos no meu sistema de estoque.

Desejo as seguintes informações:

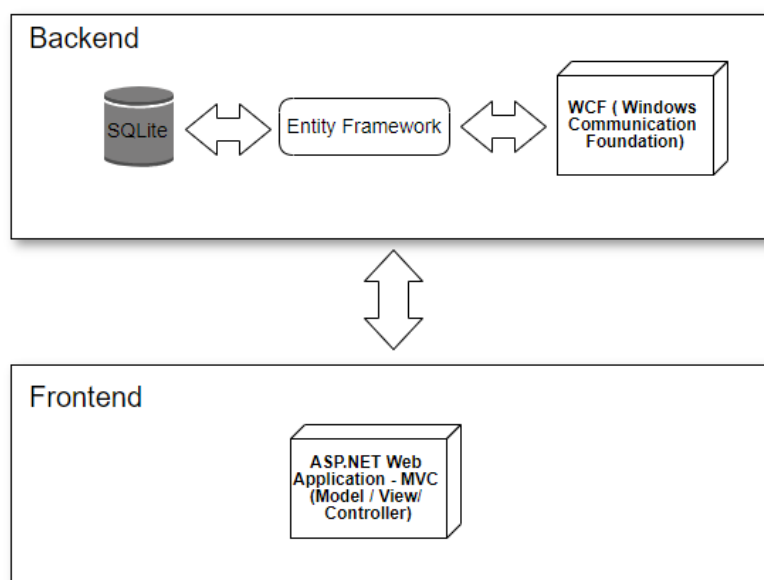
- * Nome do produto
- * Quantidade de itens do produto
- * Valor unitário do produto

Desejo que seja disponível uma tela de listagem onde eu possa selecionar um produto para exclusão ou edição (exibir uma nova tela para edição).

Nesta tela também deverá ser possível chamar uma tela para cadastrar um novo produto

3. Tecnologias Utilizadas

Para desenvolvimento deste projeto, foram utilizadas as tecnologias de acordo com o diagrama a seguir:



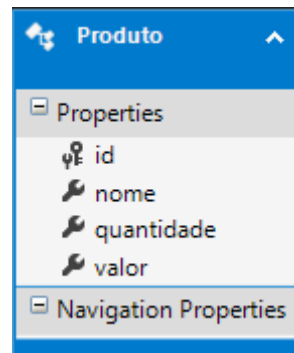
3.1. Backend:

3.1.1. SQLite

Para o banco de dados, foi utilizado o SQLite por não existir a necessidade de possuir um servidor pois seu sistema é baseado em um arquivo local.

3.1.1.1. Estrutura do banco de dados:

Inicialmente, para atender a necessidade do usuário, o banco de dados necessitava de uma única tabela com os dados Nome, Quantidade e Valor Unitário. Sendo assim, a estrutura do banco ficou da forma mostrada neste diagrama:



3.1.2. Entity Framework

O Entity Framework é uma ferramenta utilizada para auxiliar no mapeamento das classes que serão apresentadas no sistema e o banco de dados relacional.

Foi utilizado o modelo Database First, onde primeiro foi projetado o banco de dados relacional e, a partir deste, foi implementada a classe referente a sua tabela.

3.1.3. WCF (Windows Communication Foundation)

É uma tecnologia baseada na arquitetura orientada a serviços que facilitou bastante o desenvolvimento de aplicações distribuídas.

Neste projeto, foi desenvolvido um serviço que é consumido pelo cliente para realizar todo o acesso ao estoque de produtos.

3.2. Frontend:

3.2.1. ASP.NET Web Application – MVC (Model / View / Controller)

O MVC é um padrão de arquitetura que tenta separar as regras de lógica do negócio da apresentação, fazendo com que tenhamos maior controle sobre a aplicação facilitando a manutenção do projeto.

Nesta aplicação o controller realiza o controle entre o modelo dos dados e a visualização da página fazendo as requisições para o serviço criado pelo WCF.

3.3. Justificativa para utilização destas tecnologias:

As tecnologias utilizadas neste projeto foram escolhidas pensando no crescimento do mesmo.

O SQLite foi escolhido por sua fácil implementação em um projeto pequeno e por sua fácil migração para outro tipo de estrutura, como por exemplo, o MySQL.

O Entity Framework auxilia na persistência de dados, fazendo com que caso seja necessário realizar alterações no banco, como incluir campos nas tabelas ou criar novas tabelas, seja facilmente mapeado para a aplicação. Outro motivo para a utilização do Entity é por causa da forma como as consultas ao banco são realizadas, sendo que se mudarmos de SQLite para MySQL, não será necessário alterar o código referente as consultas pois os comandos serão os mesmos.

Em relação ao WCF, ele possui uma estrutura de serviço que não é complexa, onde utilizamos conceitos de programação .NET para a criação dos contratos e das classes que representam os serviços.

Já o MVC foi utilizado pois sua estrutura facilita bastante na manutenção do projeto e em seu crescimento, caso seja necessário criar outros tipos de interação com o banco, podemos analisar e criar outro controller para realizar a ligação entre o novo modelo e as novas views ou acrescentar mais operações no controller já criado.
