

Projeto Rodolfo – Prova DTI Digital

01/04/2019

Detalhamento do projeto

Tecnologias Utilizadas

- 1 Entity Framework
- 2 Asp.Net MVC
- 3 Web Api
- 4 Bootstrap
- 5 SQL Server
- 6 JavaScript
- 7 AlertifyJS
- 8 C#

O projeto foi desenvolvido na linguagem C#.

O front-end foi desenvolvido utilizando o ASP.Net MVC, usando-se de recursos do bootstrap e o alertifyJS para gerar uma interface mais amigável.

O back-end foi feito como uma Web Api.

A tecnologia utilizada para criação, manutenção e controle de banco de dados foi o Entity Framework.

O SGBD utilizado nessa solução foi o SQL Server.

Ajustes a serem feitos (se necessário)

Caso necessário ajustar o sistema para que ele execute, é preciso alterar:

- Connection String dentro do arquivo ../WebAPI/Web.config
- Caminho que a API está rodando no servidor, dentro do arquivo ../MVC/variáveis.cs, no BaseAdress do WebClient.
- Instalar Entity Framework via Gerenciador de Pacotes Nugget.
- Instalar AlertifyJS via Gerenciador de Pacotes Nugget.
- Instalar WebAPI Client via Gerenciador de Pacotes Nugget.

Documentação API

Campos

Código: int, PK;
Nome: string, campo obrigatório;
Estoque: int;
Valor: string;

Métodos API

Dados de todos os registros: GET: api/produtos
Dados de um registro: GET: api/produtos/{código}
Incluir novo registro: POST: api/produtos/{produto}
Alterar um registro: PUT: api/produtos/{código}
Excluir um registro: DELETE: api/produtos/{código}

Decisões de Projeto

Escolha do banco de dados

Contexto: Necessário utilizar um sistema gerenciador de banco de dados que seja relacional e aceite a framework de persistência escolhida.

Decisão: Foi escolhido o SQL Server.

Consequências: O SQL Server é uma ferramenta de fácil integração com o Visual Studio e com o Entity Framework, que será utilizado para controlar a gravação/leitura dos dados em banco. O SQL Server exige maior capacidade de processamento da máquina servidora, em relação a alguns bancos mais leve, nesse caso, para rodar o projeto localmente, é necessário ter um computador melhor e com o sistema operacional atualizado.

Tecnologia de persistência de dados

Contexto: Necessário utilizar alguma tecnologia para controle da persistência dos dados no banco.

Decisão: Foi escolhido o Entity Framework.

Consequências: O Entity Framework é uma ferramenta bem simples de utilizar, que gerencia todas as questões relacionadas ao banco de dados sozinho, podendo inclusive criá-lo automaticamente, baseado nas classes do sistema. É possível também, ter um sistema multi banco de dados, sem que seja desenvolvida nenhuma questão sobre isso, sendo necessário apenas ajustar a connection string do projeto. Porém, só será possível utilizar bancos de dados que a ferramenta aceite, há uma ligeira lentidão em consultas mais complexas e não é possível utilizar comandos SQL, sendo necessário o uso de expressões lambda.

Escolha da tecnologia para o front-end

Contexto: Necessário escolher algum modelo para a criação do front-end.

Decisão: Foi escolhido o ASP.Net MVC.

Consequências: O ASP.Net MVC, é uma ferramenta nativa no Visual Studio que possui seu processo de criação e utilização bastante automatizado, pois a própria IDE já gera as páginas formatadas. Além disso, ele é estruturado no formato MVC, que atende as boas práticas para programação. As páginas também são geradas de forma responsiva. Porém, o sistema fica um pouco preso na forma de layout e funcionamento do ASP.Net MVC, onde modificá-lo é um pouco trabalhoso.

Layout do sistema

Contexto: Necessário escolher um layout para o sistema.

Decisão: Foi escolhido um layout clean, apenas com uma navbar na parte superior, onde as novas telas serão chamadas.

Consequências: O sistema fica bastante leve visualmente e a nível de performance. É fácil encontrar todas as opções de acesso no sistema, pois só existe um lugar onde procurar. Porém, o sistema não fica muito rico visualmente.

Emitir alertas ao usuário

Contexto: Necessário emitir alertas e notificar o usuário sobre ações realizadas.

Decisão: Foi escolhido o AlertifyJS.

Consequências: O AlertifyJS é uma biblioteca Javascript, que simplifica bastante a criação de alertas, mensagens de confirmação, etc... Agilizando assim o desenvolvimento dessas funções. Porém o desenvolvedor fica preso ao padrão estipulado pela biblioteca e precisa estudar seu funcionamento.

Escolha da tecnologia do back-end

Contexto: Necessário escolher uma tecnologia de back-end para o sistema.

Decisão: Foi utilizada a Web API.

Consequências: Com a utilização do back-end em uma Web Api, Rest, facilita a utilização daquelas informações em um outro sistema qualquer, inclusive de fora da organização. É uma tecnologia nativa no Visual Studio, que possui seu processo de criação, utilização e documentação bastante automatizado e simplificado, pois a própria IDE já gera as páginas. Por ser uma API, exige maiores cuidados em seu desenvolvimento, inclusive na parte de segurança. Também é uma rotina um pouco mais complexa de ser desenvolvida.