

Teste Sênior

Objetivo

Criar uma aplicação de cadastro de veículos para revenda.

Funcionalidades

1. Cadastro de marcas
2. Cadastro de veículos
3. Cadastro de proprietário

Tecnologias

1. C#
2. Banco de dados a escolha do desenvolvedor
3. Projeto Web utilizando Asp.Net MVC 4 ou superior
4. EntityFramework 6 ou superior
5. Typescript
6. Razor
7. Docker
8. RabbitMQ
9. Testes

Informações das funcionalidades

1. Cadastro de marcas
 - a. Id
 - b. Nome
 - c. Status [Ativo, Cancelado]
2. Cadastro de proprietário
 - a. Id
 - b. Nome
 - c. Documento
 - d. E-mail
 - e. Endereço
 - f. Status [Ativo, Cancelado]
3. Cadastro de veículos
 - a. Id
 - b. Proprietário [somente ativos]
 - c. RENAVAM
 - d. Marca [somente ativos]

- e. Modelo
- f. Ano fabricação
- g. Ano modelo
- h. Quilometragem
- i. Valor
- j. Status [Disponível, Indisponível, Vendido]

Regras de negócio

1. Cadastro de marcas
 - a. Não permitir a duplicação, nome deve ser único
 - b. Não excluir
 - c. Não permitir alterar o nome
 - d. Permitir alterar o status para cancelado
 - e. Permitir alterar o status para ativo
 - f. Todos os campos são obrigatórios
2. Cadastro de proprietários
 - a. Não permitir a duplicação, documento deve ser único
 - b. Não excluir
 - c. Permitir alterar
 - d. Permitir alterar o status para cancelado
 - e. Permitir alterar o status para ativo
 - f. Todos os campos são obrigatórios
 - g. Buscar o endereço com base no CEP usando a API de cep
 - i. <https://brasilapi.com.br/docs#tag/CEP>
3. Cadastro de veículos
 - a. Não permitir a duplicação, RENAVAM deve ser único
 - b. Não excluir
 - c. Permitir alterar
 - d. Permitir alterar o status para vendido ou indisponível, nunca voltar para disponível
 - e. Todos os campos são obrigatórios
 - f. Ao cadastro o veículo deve ser enviado para um fila no RabbitMQ
 - g. Deve existir um consumer para ler a fila e enviar um e-mail para o proprietário informando o cadastro

Regras do projeto

- a. A aplicação, banco de dados e RabbitMQ deve estar em seus respectivos containers
- b. Utilizar docker-compose para levantar o ambiente

- c. Deve ser utilizada todas a tecnologias do item tecnologias
- d. Utilizar orientação objetos utilizando os princípios SOLID
- e. Implementar pelo menos um teste em cada funcionalidade
- f. Implementar pelo menos um teste unitário em cada funcionalidade
- g. Não utilizar geração automática de código (crud). Exemplo: criar controller e view usando o recurso de scaffold do Visual Studio

Avaliação

- a. Organização do código
- b. Boas práticas
- c. Atendimento aos requisitos
- d. Lógica e estrutura do projeto

Apresentação

Será apresentado o código para o time técnico onde será feito perguntas relacionadas ao projeto apresentado.

Conclusão

O projeto tem como objetivo a apresentação de conhecimento relacionados a C#, Docker, Asp.NET MVC, mensageria. Quaisquer outras técnicas ou tecnologias além dessas que possam contribuir com a solução é bem-vinda. Portanto caso queira incluir biblioteca e/ou tecnologias que podem contribuir com o projeto pode ficar à vontade. A biblioteca de testes fica a escolha do desenvolvedor.