

Precisamos desenvolver um sistema web que nos permita controlar a utilização dos automóveis de uma empresa. Para isso precisaremos construir uma WebAPI com as funcionalidades abaixo:

- Cadastro de automóvel:
 - Cadastrar um novo automóvel
 - Atualizar um automóvel cadastrado
 - Excluir um automóvel cadastrado
 - Recuperar um automóvel cadastrado pelo seu identificador único
 - Listar os automóveis cadastrados. Deve ser possível filtrar a listagem dos automóveis por cor e marca.
- Cadastro de motoristas
 - Cadastrar um novo motorista
 - Atualizar um motorista cadastrado
 - Excluir um motorista cadastrado
 - Recuperar um motorista cadastrado pelo seu identificador único
 - Listar os motoristas cadastrados. Deve ser possível filtrar a listagem dos motoristas por nome.
- Utilização de um automóvel
 - Criar um registro que represente a utilização de um automóvel por um motorista, com uma data de início e um texto do motivo de utilização.
 - Finalizar a utilização de um automóvel por um motorista guardando a data de finalização
 - Listar os registros de utilização cadastrados no sistema com o nome do motorista e as informações do automóvel utilizado

O que devemos controlar de cada recurso:

- Automóvel
 - Placa
 - Cor
 - Marca
- Motorista
 - Nome
- Utilização do automóvel
 - Data de início da utilização
 - Data de término da utilização
 - Motorista que utilizou
 - Automóvel utilizado
 - Motivo de utilização

Regras de negócio: Um automóvel só pode ser utilizado por um motorista por vez. Um motorista que já esteja utilizando um automóvel não pode utilizar outro automóvel ao mesmo tempo.

O que você deve entregar?

- O código fonte da sua aplicação, as instruções de como executar e testar a sua aplicação. Deve ser entregue um link para um repositório git público (ex: Github, Bitbucket, Gitlab, etc).
- Testes de unidade que garantam que seu código está funcionando corretamente.

O que será avaliado?

- Funcionalidades: sua aplicação cobre as funcionalidades que foram solicitadas?
- Estrutura do código: Se o código foi bem escrito, se está organizado, se é fácil de ler (nomes de variáveis, funções, etc..) e quando/se for necessário, se existe um bom comentário que explique um determinado ponto do código.

Observações:

- É necessário que o teste seja desenvolvido utilizando Node.js.
- É recomendado que utilize o ExpressJs como framework para apoio, mas o teste pode ser desenvolvido com qualquer outro framework.
- Não se faz necessário a utilização de persistência forte (banco de dados). Pode ser utilizado a persistência em memória.
- O prazo para resolução do teste é de 3 dias

É importante que o projeto esteja funcional, mas não avaliaremos somente esse ponto no teste prático então mesmo que não consiga terminar a solução, envie-nos seu trabalho.

Muito obrigado pelo seu tempo, participação e boa sorte!