

# DETALHES DO DESAFIO ESTACIONAMENTO

Solicitante: **Empresa VSOFTE**.  
Analista responsável pelo desenvolvimento: **Luiz Navarro Silva Junior**

## SUMÁRIO

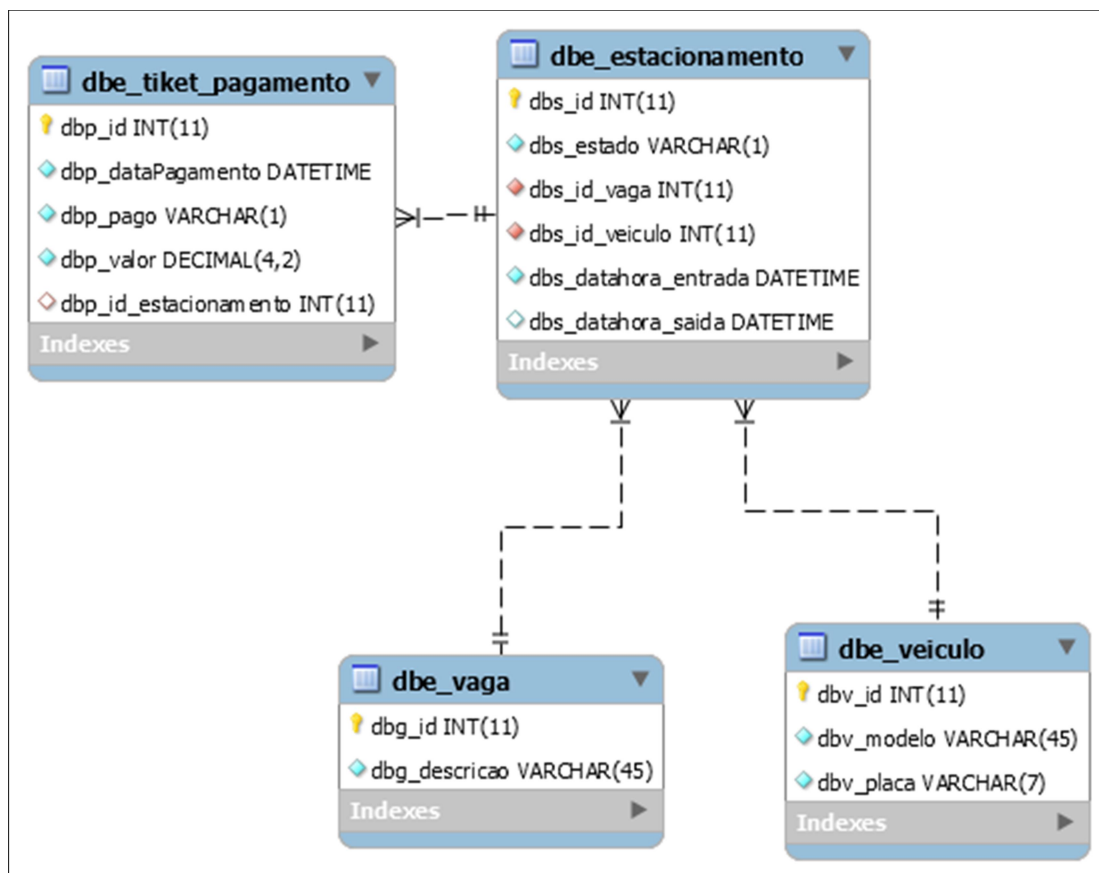
1.	Arquitetura e tecnologias utilizadas.....	2
2.	MODELO DO BANCO DE DADOS.....	2
3.	FRAMEWORK FRONT-END .....	2
4.	RELATÓRIOS .....	2
5.	TESTES.....	3
6.	AUTENTICAÇÃO .....	3
7.	DOCUMENTAÇÃO.....	4
8.	INSTRUÇÕES PARA RODAR O PROJETO.....	4
9.	MAPEAMENTO DAS FUNCIONALIDADES .....	4
10.	CONSIDERAÇÕES FINAIS.....	6

## 1. Arquitetura e tecnologias utilizadas

Para elaboração do produto foi utilizado o .Net Core na versão 2.0, A solução foi composta por três projetos sendo o primeiro projeto a API contendo os métodos Rest , o segundo projeto foi um aplicação Web pra consumo da API e por último um projeto de testes dos controles da API, sendo utilizada a arquitetura MVC nos dois primeiros citados.

## 2. Modelo do banco de dados

Foi adotado o banco de dados MySQL para ser o responsável pelo armazenamento de dados do projeto sendo composto de quatro tabelas, suficiente para demonstrar a lógica necessária. Detalhes do modelo abaixo:



## 3. Framework Front-End

Foi adotado o Bootstrap padrão do .Net Core para ser o front-end da aplicação, sendo mais que suficiente para suportar as funcionalidades requeridas.

## 4. Relatórios

Para exportação dos dados em relatórios PDFs foi utilizada a biblioteca Rotativa disponível via nuget (<https://www.nuget.org/packages/Rotativa>) , este plugin é capaz de gerar relatórios a partir de Views.

## 5. Testes

Além do PostMan (<https://www.getpostman.com/>) e o próprio swagger para realizar os testes na própria API, foi utilizado alguns plug-ins para construção dos testes nos controles da API:

- Microsoft.AspNetCore.All
- Microsoft.AspNetCore.TestHost
- FluentAssertions
- Moq

O primeiro contém as dependências do .Net Core no meu projeto, o segundo para testes de integração para poder construir um host de testes, o FluentAssertions um modo mais elegante para fazer manipulações e o Moq para criar objetos fake.

Um detalhe importante para rodar os testes a autenticação não é realizada por não ter acesso ao HttpContext, em modo debug esta verificação não é realizada.

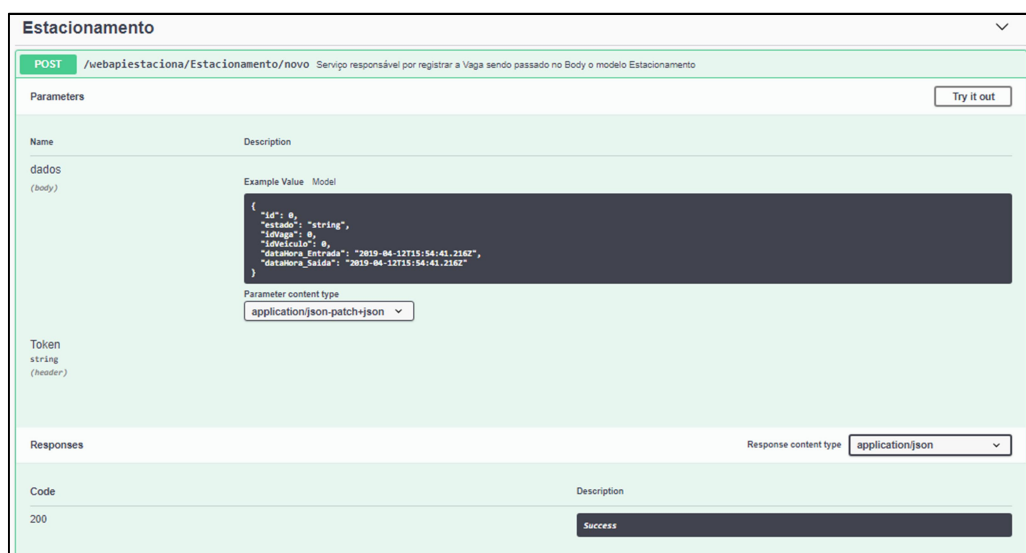
Caso seja feito testes via PostMan ou Swagger deve-se passar o parâmetro do token cujo valor é "Token" = 12234JHJKHKHDdududjddksksawxcb34568.

Mais uma observação alguns métodos de teste necessitam da correta informação dos dados como é o caso do pagamento e da ação de estacionar.

## 6. Autenticação

Foi incluído no projeto um mecanismo de autenticação simples que utiliza o HttpContext para armazenar os dados de conexão/transação. A autenticação é realizada através de um token, para realizar as transações é necessário passar por parâmetro a variável token no header da requisição. Para nosso projeto de consumo o token já está configurado.

No Swagger foi adicionado o parâmetro no header para que seja informado o token de acesso.



## 7. Documentação

Como sugerido a documentação da api é realizada através do swagger.

## 8. Instruções para rodar o projeto

Para poder rodar o projeto deve-se seguir os seguintes passos:

1. Criação do banco de dados, o script de criação da estrutura está disponível nos fontes. (Há uma fonte já com os dados e outra apenas com a estrutura).
2. Instanciação do projeto API (ao instanciar a aplicação pela primeira vez as vagas (15) são criadas)
3. Instanciação do projeto aplicação web

## 9. Mapeamento das funcionalidades

Abaixo serão descritas as telas e funcionalidades do projeto.

**Menu Principal > Veículos:** Exibe, cadastra e estaciona os veículos. Para estacionar um veículo é necessário selecionar uma vaga disponível e confirmar.

**Cadastro de Veículos**

Modelo	Placa	
corolla 2018	MNC8989	Estacionar
Ford Ka	MNC8999	Estacionar
Voyage	MPP8976	Estacionar
Voyage	MPP8970	Estacionar
Caminhão	MPP8930	Estacionar
Civic	AAA8989	Estacionar
Pallo Único	QQQ6565	Estacionar
FORD Ranger	WWW8889	Estacionar
RENAULT SANDERO	WER1233	Estacionar

**Estacionar Veículo**

Dados:

Vaga:

Entrada:

© 2019 - WebAplicacaoEstaciona

**Menu Principal > Estacionamento > Vagas disponíveis:** Exibe todas as vagas disponíveis

**Vagas Disponíveis**

Vaga
A1
A2
A3
A4
A5
A6
A8
A9
A10
A11

**Menu Principal > Estacionamento> Vagas Ocupadas:** Posição geral do estacionamento. No botão "documento pdf" é exibido o relatório em PDF.

Controle de Estacionamento   Veiculos   Estacionamento ▾					
Vagas Ocupadas					
Vaga	Registro no Estacionamento	Veiculo Modelo	Veiculo Placa	Entrada	
A7	13	Caminhão	MPP8930	12/04/2019 09:20:10	<a href="#">Pagar Estacionamento</a>
A9	14	Voyage	MPP8976	12/04/2019 10:49:55	<a href="#">Pagar Estacionamento</a>
A4	15	Voyage	MPP8970	12/04/2019 11:06:33	<a href="#">Pagar Estacionamento</a>
<a href="#">Documento (PDF)</a>					
© 2019 - WebAplicacaoEstaciona					

**Menu Principal > Estacionamento> Relatório Geral:** Relatório geral e financeiro do estacionamento por período. Detalhe para a possibilidade de exportar para pdf.

### Relatório Geral

Dados:

Data Inicio

Data Final

[Sair](#) [Exibir](#)

© 2019 - WebAplicacaoEstaciona

Controle de Estacionamento

Veiculos

Estacionamento ▾

Relatório Gerencial

Vaga Ocupada	Veiculo Modelo	Veiculo Placa	Entrada	Saida	Valor Pago
A1	corolla 2018	MNO8989	10/04/2019 06:15:09	10/04/2019 11:40:32	13,00
A2	Ford Ka	MNO8999	10/04/2019 14:59:33	11/04/2019 11:14:59	58,00
A8	Caminhão	MPP8930	11/04/2019 11:30:55	11/04/2019 11:34:25	7,00
A4	Voyage	MPP8970	11/04/2019 00:57:44	11/04/2019 14:30:09	37,00
A3	corolla 2018	MNO8989	11/04/2019 11:24:25	11/04/2019 21:19:03	25,00
A9	Pallo Único	QQQ6565	11/04/2019 21:30:19	11/04/2019 21:30:55	7,00
A7	Ford Ka	MNO8999	11/04/2019 11:29:29	11/04/2019 21:43:21	28,00
A5	RENAULT SANDERO	WER1233	12/04/2019 07:18:51	12/04/2019 07:28:56	7,00
A4	Voyage	MPP8970	11/04/2019 21:43:07	12/04/2019 07:31:49	25,00
A3	Voyage	MPP8976	12/04/2019 07:32:01	12/04/2019 07:32:46	7,00
A1	corolla 2018	MNO8989	12/04/2019 09:08:29	12/04/2019 09:18:37	7,00

Valor Total no Período

R\$221,00

Documento (PDF)

## **10. Considerações finais**

Um problema em meu computador impediu de ser utilizado o Entity Framework no projeto que seria minha escolha para acesso aos dados, devido a este fato foi utilizado as manipulações de dados através de uma classe DAO.