Introdução

Este trabalho é um projeto acadêmico para uma disciplina da faculdade, que visa desenvolver um sistema de locadora de veículos. A importância deste projeto reside na simulação de um ambiente real de desenvolvimento de software, permitindo que os alunos pratiquem a modelagem de banco de dados, a implementação de APIs e o uso de tecnologias modernas.

Os principais objetivos do sistema incluem a gestão de veículos, clientes e locações, oferecendo funcionalidades como cadastro, consulta e atualização de dados. A implementação utiliza tecnologias como C# 8.0, ASP.NET Core, Entity Framework e SQL Express, proporcionando uma base sólida para a construção do sistema.

Contextualização

O trabalho consiste em um sistema de locação de veículos realizado para a matéria de Tecnologias para Análise e Desenvolvimento de Sistema. Para a criação do sistema foram utilizados C#, VisualStudio, Entity Framework, Asp.Net Core, Swagger e SQLServer.

Requisitos

 RF-01: Deve ser criado um modelo conceitual para o banco de dados da locadora de veículos, incluindo entidades como Veículo, Cliente, Locação, Manutenção e Funcionário.

RF-02: O sistema deve utilizar o Entity Framework para traduzir o modelo conceitual em um esquema de banco de dados relacional.

 RF-03: Deve ser desenvolvido um backend em C# 6.0 (ou superior) utilizando o ASP.NET Core para criar APIs RESTful que interajam com o banco de dados.

 RF-04: O sistema deve implementar endpoints para operações CRUD (Criar, Ler, Atualizar, Deletar) para as entidades do sistema, incluindo veículos e clientes.

 RF-05: O sistema deve integrar o Swagger para documentar e testar as APIs desenvolvidas.

Arquitetura do sistema:

Model: Define as entidades da aplicação, como Veiculo, Cliente, Locacao e Manutencao, incluindo suas propriedades e a lógica de negócios.

View: Responsável pela interface do usuário, garantindo uma experiência intuitiva e responsiva para consultas e alterações de dados.

Controller: Atua como intermediário entre o modelo e a visão, processando as solicitações do usuário e gerenciando operações como a criação de locações

Foi utilizado Entity Framework para a interação com o banco de dados e Swagger para documentar e testar nossas APIs.

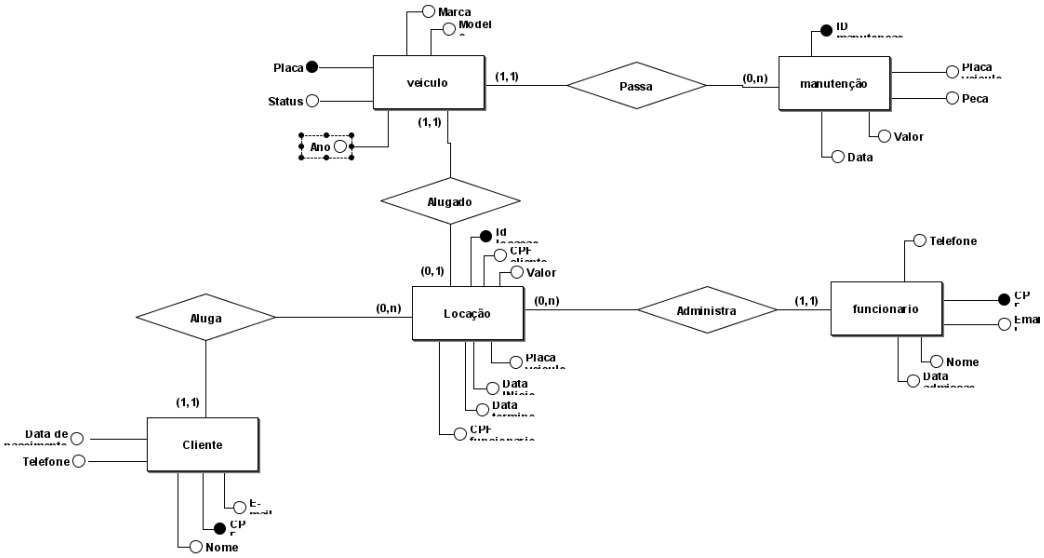
Tecnologias Utilizadas

- **C# 8.0**: Linguagem de programação moderna e orientada a objetos, utilizada para desenvolver a lógica do backend e as APIs do sistema.
- **ASP.NET Core**: Framework web da Microsoft, que permite criar aplicações robustas e escaláveis, especialmente voltadas para APIs RESTful.
- **Entity Framework**: ORM (Object-Relational Mapper) que facilita a interação com o banco de dados, permitindo trabalhar com dados usando objetos C# e evitando a escrita de SQL diretamente.
- **SQL Express**: Sistema de gerenciamento de banco de dados relacional leve da Microsoft, utilizado para armazenar e gerenciar os dados da aplicação.
- **Swagger**: Ferramenta de documentação de APIs que permite gerar uma interface interativa para testar e documentar as APIs desenvolvidas.

Classes e Responsabilidades

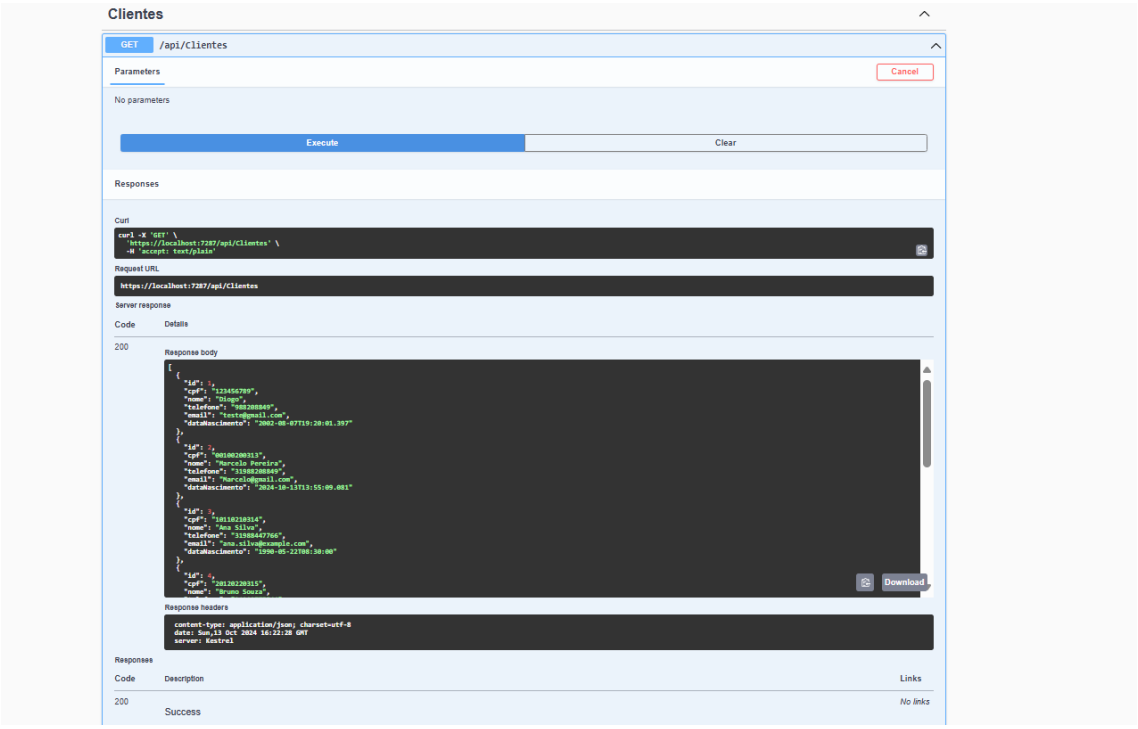
1. **Locacao**: Representa as locações de veículos, contendo informações como LocacaoId, VeiculoId, ClienteId, FuncionarioId, Data e ValorTotal.
2. **Veiculo**: Representa os veículos disponíveis para locação, com atributos como ID, Placa, Marca, Modelo, Ano e Status.
3. **Cliente**: Representa os clientes da locadora, incluindo ID, CPF, Nome, Telefone, Email e DataNascimento.
4. **Funcionario**: Representa os funcionários, com atributos como ID, CPF, Nome, DataAdmissao, Telefone e Email.
5. **Manutencao**: Representa as manutenções dos veículos, contendo ManutencaoId, VeiculoId, Data, DescricaoServico e Custo.
6. **LocadoraContext**: Gerencia a conexão com o banco de dados e as operações de CRUD sobre as entidades.
7. **Controladores**: Gerenciam requisições HTTP para as entidades, implementando operações como GET, POST, PUT e DELETE.

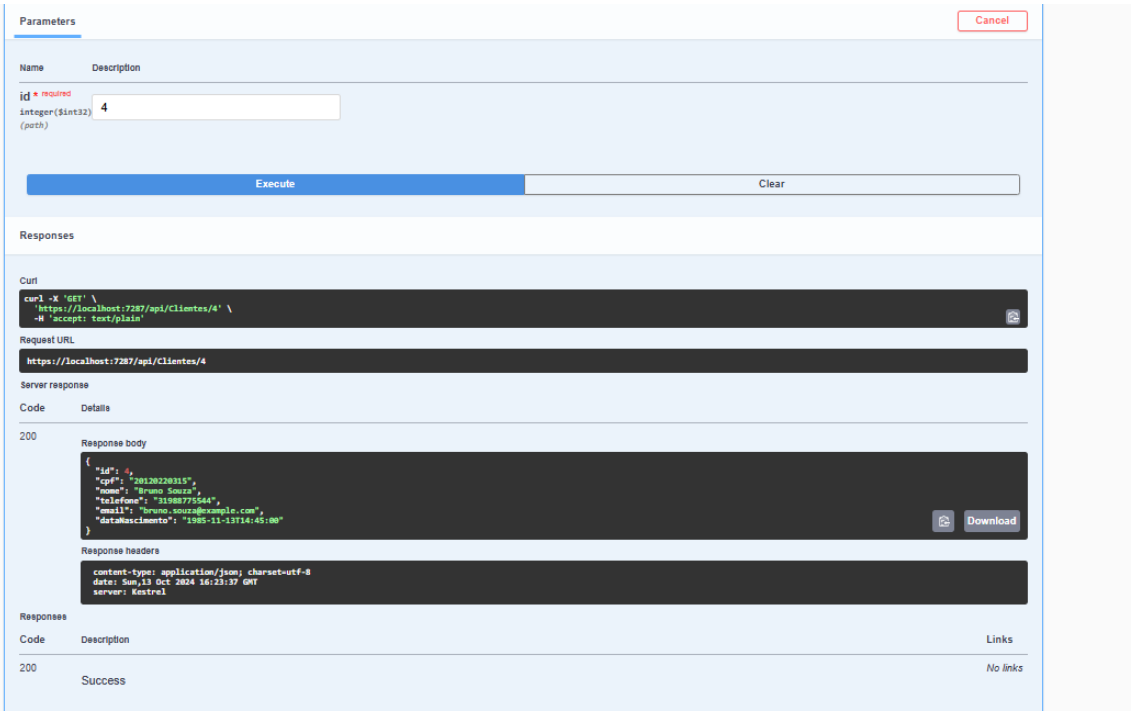
Descrição do Banco de Dados:



Testes:

Exemplo de alguns teste realizados:


Clientes
GET
/api/Clientes
Parameters
Cancel
No parameters
Execute
Clear
Responses
Curl
curl -X 'GET' \
'https://localhost:7287/api/Clientes' \
-H 'accept: text/plain'
Request URL
https://localhost:7287/api/Clientes
Server response
Code
Details
200
Response body
Download
Response headers
content-type: application/json; charset=utf-8
date: Sun,13 Oct 2024 16:22:28 GMT
server: Kestrel
Responses
Code
Description
Links
200
Success
No links


Parameters
Cancel
Name
Description
id
* required
integer($int32)
(path)
4
Execute
Clear
Responses
Curl
curl -X 'GET' \
'https://localhost:7287/api/Clientes/4' \
-H 'accept: text/plain'
Request URL
https://localhost:7287/api/Clientes/4
Server response
Code
Details
200
Response body
Download
Response headers
content-type: application/json; charset=utf-8
date: Sun,13 Oct 2024 16:23:37 GMT
server: Kestrel
Responses
Code
Description
Links
200
Success
No links

POST /api/Funcionarios

Parameters

Cancel Reset

No parameters

Request body application/json

```
{
  "cpf": "00000000000",
  "nome": "Mauricio",
  "dataAdmissao": "2022-10-13T16:24:07.967Z",
  "telefone": "13456345",
  "email": "Mauricio@yahoo.com"
}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7287/api/Funcionarios' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "cpf": "00000000000",
  "nome": "Mauricio",
  "dataAdmissao": "2022-10-13T16:24:07.967Z",
  "telefone": "13456345",
  "email": "Mauricio@yahoo.com"
}'
```

Request URL

```
https://localhost:7287/api/Funcionarios
```

Server response

Code	Details
201 *Undocumented*	Response body

```
{
  "id": 4,
  "cpf": "00000000000",
  "nome": "Mauricio",
  "dataAdmissao": "2022-10-13T16:24:07.967Z",
  "telefone": "13456345",
  "email": "Mauricio@yahoo.com"
}
```

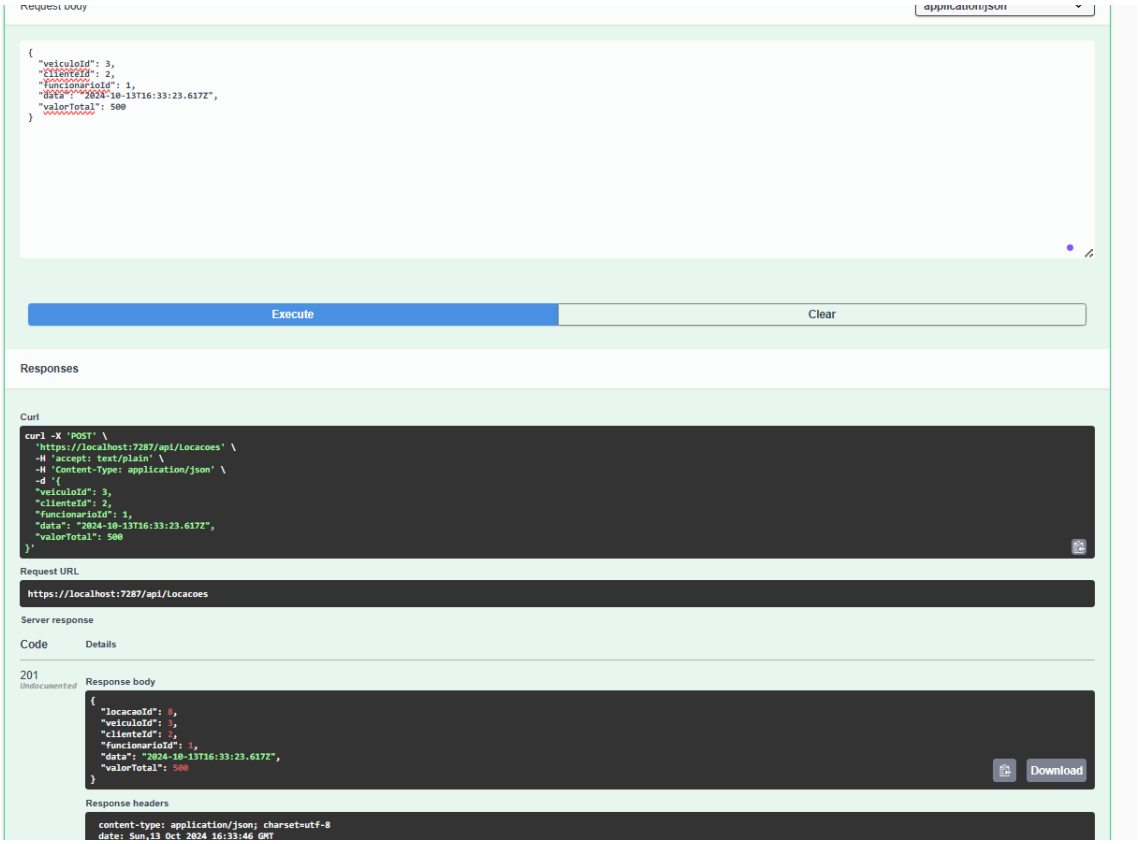
Download

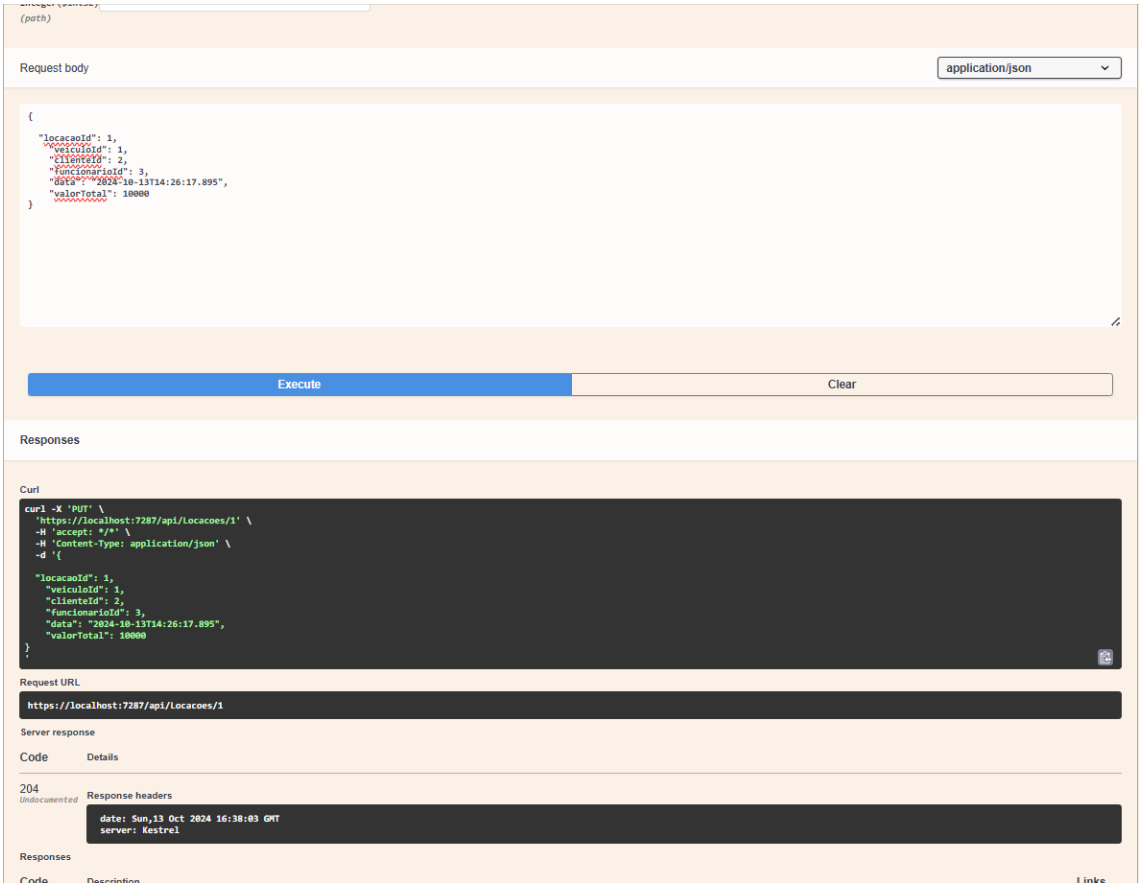
Response headers

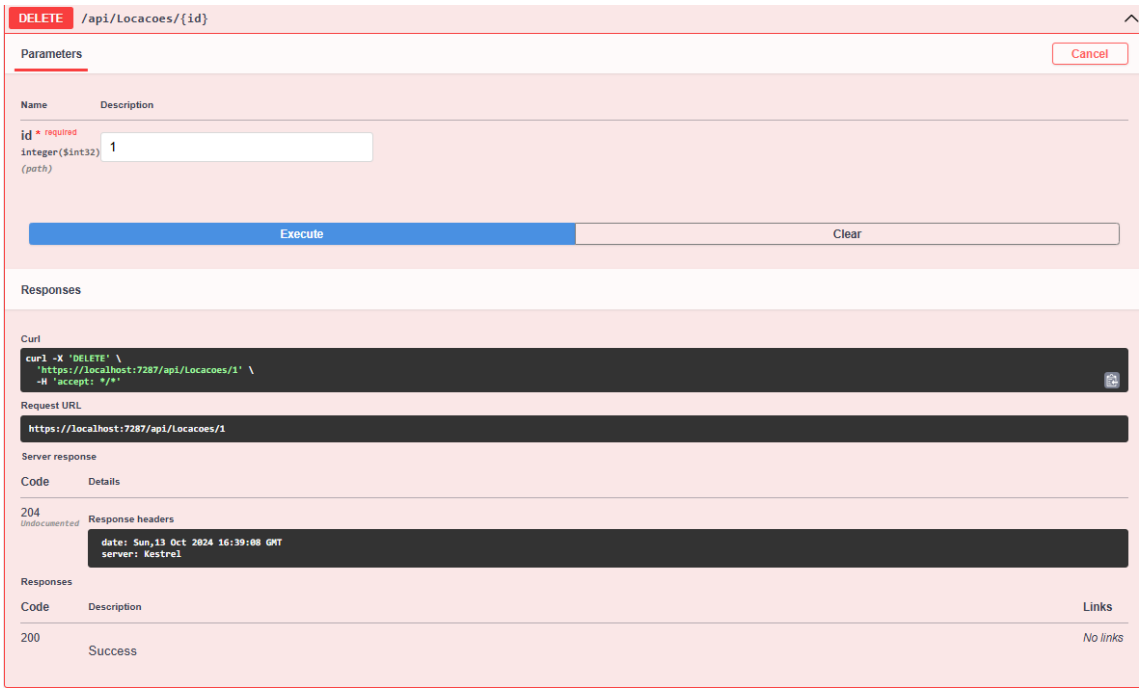
```
content-type: application/json; charset=utf-8
date: Sun,13 Oct 2024 16:24:56 GMT
location: https://localhost:7287/api/Funcionarios/4
server: Kestrel
```

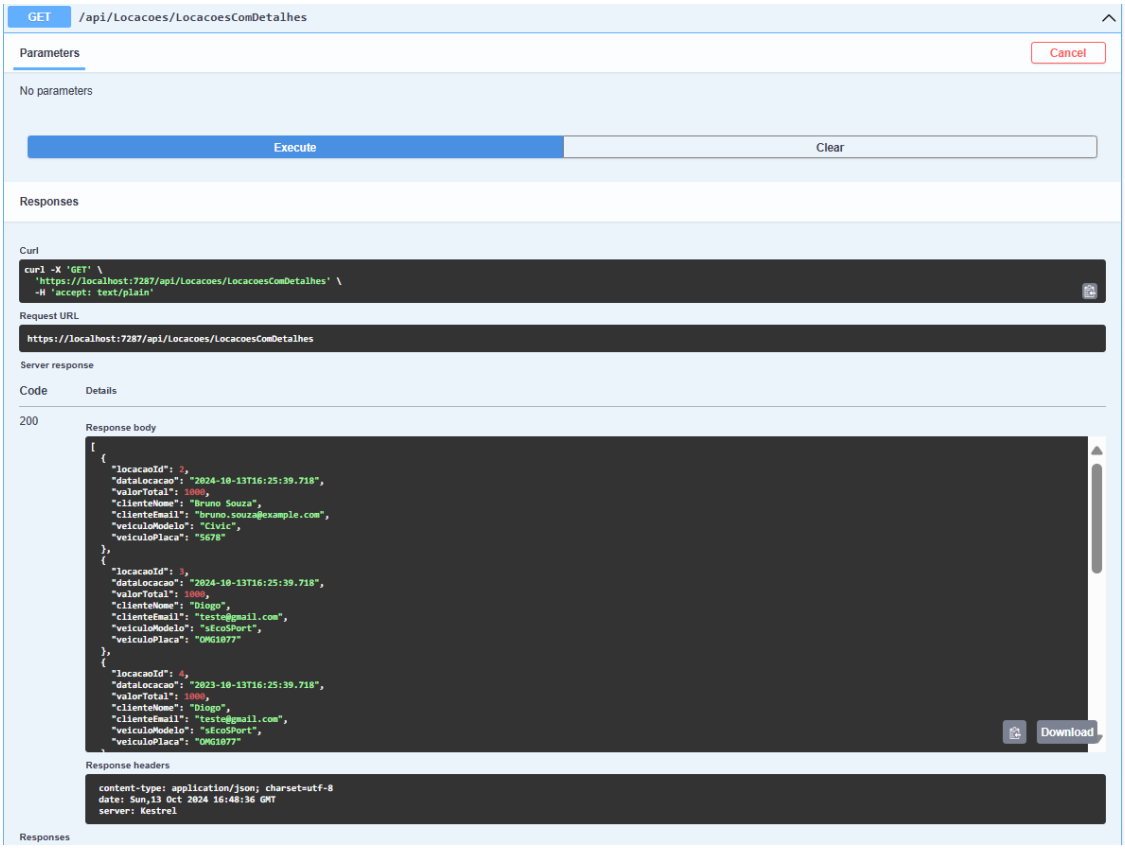
Responses

Code	Description	Links
200		*No links*


Request body
application/json
{
"veiculoId": 3,
"clienteId": 2,
"funcionarioId": 1,
"data": "2024-10-13T16:33:23.617Z",
"valorTotal": 500
}
Execute
Clear
Responses
Curl
curl -X 'POST' \
'https://localhost:7287/api/Locacoes' \
-H 'accept: text/plain' \
-H 'Content-Type: application/json' \
-d '{
"veiculoId": 3,
"clienteId": 2,
"funcionarioId": 1,
"data": "2024-10-13T16:33:23.617Z",
"valorTotal": 500
}'
Request URL
https://localhost:7287/api/Locacoes
Server response
Code
Details
201
Undocumented
Response body
{
"locacaoId": 8,
"veiculoId": 3,
"clienteId": 2,
"funcionarioId": 1,
"data": "2024-10-13T16:33:23.617Z",
"valorTotal": 500
}
Download
Response headers
content-type: application/json; charset=utf-8
date: Sun,13 Oct 2024 16:33:46 GMT


(path)
Request body
application/json
{
"locacaoId": 1,
"veiculoId": 1,
"clienteId": 2,
"funcionarioId": 3,
"data": "2024-10-13T14:26:17.895",
"valorTotal": 10000
}
Execute
Clear
Responses
Curl
curl -X 'PUT' \
'https://localhost:7287/api/Locacoes/1' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
"locacaoId": 1,
"veiculoId": 1,
"clienteId": 2,
"funcionarioId": 3,
"data": "2024-10-13T14:26:17.895",
"valorTotal": 10000
}
'
Request URL
https://localhost:7287/api/Locacoes/1
Server response
Code
Details
204
Undocumented
Response headers
date: Sun,13 Oct 2024 16:38:03 GMT
server: Kestrel
Responses
Code
Description
Links





Relato de problemas encontrados durante os testes

O desenvolvimento do projeto passou por diversas dificuldades, principalmente por causa da falta de conhecimento. Os métodos de get foram os que mais deram erros, so sendo resolvidos após muitos erros e pesquisas.

Conclusões finais

O desenvolvimento do sistema de locadora de veículos proporcionou valiosas experiências, como a prática na modelagem de banco de dados com Entity Framework e a construção de APIs RESTful com ASP.NET Core. A integração do Swagger foi um diferencial importante para a documentação e testes das APIs, aumentando a eficiência do desenvolvimento.

Para melhorias futuras, sugiro a implementação de um sistema de autenticação e autorização para proteger as APIs, além de recursos de relatórios que permitam uma análise mais profunda dos dados. A inclusão de um painel administrativo poderia facilitar o gerenciamento das locações, veículos e clientes, melhorando ainda mais a experiência do usuário.

Referencias:

- MICROSOFT. ASP.NET Core Documentation. Disponível em: https://docs.microsoft.com/aspnet/core/?view=aspnetcore-8.0. Acesso em: 13 out. 2024.
- MICROSOFT. Entity Framework Core Documentation. Disponível em: https://docs.microsoft.com/ef/core/. Acesso em: 13 out. 2024.
- SWAGGER. Swagger Documentation. Disponível em: https://swagger.io/docs/. Acesso em: 13 out. 2024.
- MICROSOFT. Testing in ASP.NET Core. Disponível em: https://docs.microsoft.com/aspnet/core/test/?view=aspnetcore-8.0. Acesso em: 13 out. 2024.
- MICROSOFT. C# Coding Guidelines. Disponível em: https://docs.microsoft.com/dotnet/csharp/programming-guide/inside-a-program/coding-conventions. Acesso em: 13 out. 2024.