

DOCUMENTAÇÃO TÉCNICA DO PROJETO – API RESERVA DE CARROS

Título do Projeto: Sistema de Gerenciamento de Reservas de Carros

Disciplina: Tópicos Especiais de Sistemas

Data: 29/05/2025

Grupo: Matheus de Araujo Amaral, Luan Curtz, Caio Somacal, Arthur Flaminio

1. Objetivo do Projeto

O objetivo deste projeto é desenvolver uma API RESTful utilizando ASP.NET Core, Entity Framework Core e SQLite, voltada para o gerenciamento de reservas de veículos. A aplicação permite o controle do cadastro de carros, clientes e o processo completo de reserva. O sistema foi desenvolvido com foco em contextos operacionais onde é necessário controlar a disponibilidade de veículos, garantir integridade nos agendamentos e facilitar a consulta e manutenção de registros por parte de usuários e administradores.

2. Estrutura da Solução

2.1 Modelagem de Dados

A modelagem da aplicação baseia-se em três entidades principais: **Carro**, **Cliente** e **Reserva**. Cada uma dessas entidades foi estruturada com atributos essenciais ao processo de reserva:

- **Carro:** modelo, marca, ano, placa, status
- **Cliente:** nome, e-mail, telefone
- **Reserva:** relação entre carro e cliente, data de início, data de fim, status da reserva

As classes foram implementadas na camada de domínio e utilizadas como base para a criação da estrutura de banco de dados via Entity Framework Core.

2.2 Integração da API com a Solução

A API foi implementada como ponto central de acesso para operações CRUD (Create, Read, Update, Delete). Cada rota da aplicação representa uma funcionalidade específica como cadastro de cliente, consulta de carros, ou cancelamento de reservas. A comunicação entre o cliente (ex: frontend, Postman) e a base de dados se dá de maneira segura, performática e estruturada.

3. Endpoints da API

Entidade	Método	Rota	Descrição
Carro	GET	<code>/api/carros</code>	Lista todos os carros
Carro	GET	<code>/api/carros/{id}</code>	Retorna um carro por ID
Carro	POST	<code>/api/carros</code>	Cadastra um novo carro
Carro	PUT	<code>/api/carros/{id}</code>	Atualiza os dados de um carro
Carro	DELETE	<code>/api/carros/{id}</code>	Exclui um carro
Cliente	GET	<code>/api/clientes</code>	Lista todos os clientes
Cliente	GET	<code>/api/clientes/{id}</code>	Retorna um cliente por ID
Cliente	POST	<code>/api/clientes</code>	Cadastra um novo cliente
Cliente	DELETE	<code>/api/clientes/{id}</code>	Remove um cliente
Reserva	GET	<code>/api/reservas</code>	Lista todas as reservas
Reserva	GET	<code>/api/reservas/{id}</code>	Retorna uma reserva por ID
Reserva	POST	<code>/api/reservas</code>	Cria uma nova reserva
Reserva	DELETE	<code>/api/reservas/{id}</code>	Cancela uma reserva existente

Regras de negócio adicionais:

- Reservas só podem ser feitas para carros com status "Disponível".
- Cancelamento de reservas atualiza o status do carro para "Disponível".

4. Organização do Código

O projeto segue uma estrutura modular, com separação clara de responsabilidades:

- `Models/Carro.cs, Cliente.cs, Reserva.cs` – Modelos das entidades
- `Data/ApplicationDbContext.cs` – Contexto do banco de dados com EF Core
- `Data/DbInitializer.cs` – Inserção de dados exemplo
- `Routes/CarrosRoutes.cs,` – Rotas de operação com carros
- `Routes/ClientesRoutes.cs` – Rotas de operação com clientes
- `Routes/ReservasRoutes.cs` – Rotas de operação com reservas
- `Program.cs` – Configuração da aplicação, registro de serviços e endpoints

5. Execução da Aplicação

Pré-requisitos:

- [.NET 6 SDK](#)
- EF Core CLI (opcional):
`dotnet tool install --global dotnet-ef`

Passos para rodar:

`dotnet restore` # Restaurar pacotes

`dotnet ef database update` # Criar ou atualizar o banco de dados

`dotnet run` # Executar a aplicação

API disponível em `http://localhost:5000`

6. Conclusão

A API de Reserva de Carros foi desenvolvida com foco acadêmico e estrutural, utilizando boas práticas de desenvolvimento web com .NET. A organização modular, a lógica de reserva e a modelagem das entidades oferecem uma base sólida para evolução do projeto, como adição de autenticação, relatórios ou funcionalidades avançadas.