

Sistema de Gerenciamento de Reservas de Carros

Alunos: Matheus de Araujo Amaral e Caio Somacal

Data: 12/06/2025

Turma: Quinta-Feira manhã

Tópico 1 – Resumo do projeto e do problema abordado

O projeto consiste no desenvolvimento de uma API RESTful para gerenciar reservas de veículos em uma locadora. A solução visa organizar o cadastro de carros, clientes e o controle completo do processo de reservas, garantindo que apenas veículos disponíveis possam ser reservados e facilitando o cancelamento, atualização e consulta desses dados. O sistema busca resolver problemas comuns em operações de locação de veículos, como evitar conflitos de agendamento, manter o controle de status dos carros e permitir análise de dados para suporte à tomada de decisão.

Tópico 2 – Lista das funcionalidades implementadas

Cadastro, listagem, atualização e exclusão de carros

Cadastro, listagem e exclusão de clientes

Criação, listagem e cancelamento de reservas

Validação para reservar somente carros disponíveis

Atualização automática do status do carro após reserva ou cancelamento

Endpoints RESTful organizados para operações completas

Estrutura modular para facilitar manutenção e escalabilidade

Integração front-end consumindo os endpoints da API

Relatório e análise de reservas

Tópico 3 – Descrição técnica das funcionalidades e da integração com a API

A aplicação foi implementada utilizando ASP.NET Core, com a arquitetura Minimal API e Entity Framework Core para persistência com banco SQLite. A modelagem inclui três entidades principais: Carro, Cliente e Reserva, refletindo os dados essenciais para o funcionamento do sistema.

Cada entidade possui rotas específicas para manipulação via HTTP, utilizando verbos RESTful:

GET para consulta de dados (lista completa e por ID)

POST para criação de registros

PUT para atualização (carros)

DELETE para remoção (carros, clientes, reservas)

A API assegura regras de negócio importantes, como permitir reservas apenas para carros com status “Disponível” e atualizar o status do veículo ao cancelar reservas. O projeto conta com uma estrutura organizada em pastas, separando modelos, rotas e contexto do banco de dados, facilitando a escalabilidade e manutenção.

O front-end consome essa API por meio de requisições HTTP e operações CRUD atualizando a interface dinamicamente.

Tópico 4 – Endpoints

| 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".