

**Alexandre Araujo Paes**



**PUC Minas**

**Sistema de API locadora de veículos**

**Poços de caldas, Abril de 2024**

## **Introdução:**

Este relatório é dedicado ao projeto desenvolvido para a disciplina de Tecnologias para Análise e Desenvolvimento de Sistemas do curso de Ciências da Computação da PUC Minas. O projeto em questão é um sistema de locadora de veículos, concebido para demonstrar a aplicação prática das habilidades e conhecimentos adquiridos durante o curso.

A importância deste projeto se estende além do ambiente de aula. Em um mundo digital, a capacidade de desenvolver sistemas robustos e eficientes é altamente valorizada em muitos setores da indústria. No setor de aluguel de veículos, um sistema eficiente pode melhorar significativamente a experiência do cliente, além de aumentar a eficiência operacional.

O sistema de API locadora de veículos foi desenvolvido com o objetivo de simular o gerenciamento de todas as operações de uma locadora de veículos, desde o cadastro de veículos e clientes até a reserva e o aluguel de veículos. Utilizando tecnologias modernas e robustas, como C# 8.0, ASP.NET Core, Entity Framework, SQL Express, Swagger, entre outras, o sistema é não apenas um projeto acadêmico, mas também uma solução prática para um problema do mundo real.

## Requisitos Funcionais:

O sistema de locadora de veículos possui uma série de funcionalidades projetadas para facilitar a gestão de uma empresa de aluguel de veículos. Aqui estão os principais requisitos funcionais do sistema:

ID	Nome	Descrição
RF01	<b>Cadastro de Veículos</b>	O sistema permite o cadastro de veículos, incluindo detalhes como marca, modelo e status (disponível, alugado, em manutenção, etc.).
RF02	<b>Cadastro de Clientes</b>	O sistema permite o cadastro de clientes, incluindo informações pessoais e informações de contato (telefone, e-mail).
RF03	<b>Reservas</b>	O sistema permite a realização de reservas de veículos para datas específicas.

## Casos de Uso Principais:

### 1. Caso de Uso: Cadastro de Veículos

- **Ator:** Administrador
- **Descrição:** O administrador insere os detalhes do veículo no sistema, incluindo marca, modelo, ano, cor, placa e status.

### 2. Caso de Uso: Cadastro de Clientes

- **Ator:** Administrador
- **Descrição:** O administrador insere as informações do cliente no sistema, incluindo detalhes pessoais e informações de contato.

### 3. Caso de Uso: Reserva de Veículos

- **Ator:** Cliente
- **Descrição:** O cliente seleciona um veículo disponível e faz uma reserva para uma data específica.

## Arquitetura do Sistema:

O sistema de API locadora de veículos foi desenvolvido utilizando o padrão de arquitetura Model-View-Controller (MVC). O padrão MVC é um padrão de design de software que divide a aplicação em três componentes interconectados: o Modelo, a Visão e o Controlador. Isso permite uma separação clara das responsabilidades, facilitando a manutenção e a expansão do sistema.

1. **Modelo (Model):** O Modelo é responsável por representar os dados e as regras de negócio do sistema. No caso do sistema de locadora de veículos, os modelos incluem as entidades Veículo, Cliente e Reserva.
2. **Visão (View):** A Visão é responsável por apresentar os dados ao usuário de uma forma legível. No sistema de locadora de veículos, as visões são as páginas web que os usuários veem em seus navegadores.
3. **Controlador (Controller):** O Controlador é responsável por receber as solicitações do usuário, manipular os modelos conforme necessário e atualizar a visão para refletir quaisquer mudanças. No sistema de locadora de veículos, os controladores incluem controladores para gerenciar veículos, clientes e reservas.

Essa divisão de responsabilidades permite que cada parte do sistema seja desenvolvida, testada e modificada de forma independente. Isso aumenta a modularidade do sistema, tornando-o mais fácil de entender, manter e expandir.

## Tecnologias Utilizadas:

O sistema de locadora de veículos foi desenvolvido utilizando uma variedade de tecnologias modernas e robustas para garantir eficiência, segurança e escalabilidade. Aqui estão as principais tecnologias utilizadas:

1. **C# 8.0:** A linguagem de programação principal utilizada para desenvolver o sistema. O C# é uma linguagem de programação moderna, orientada a objetos, que oferece uma série de recursos avançados que facilitam o desenvolvimento de sistemas complexos.
2. **ASP.NET Core:** Um framework para desenvolvimento web que foi usado para criar a interface do usuário e a lógica do servidor do sistema. O ASP.NET Core é conhecido por sua eficiência, escalabilidade e segurança.
3. **Entity Framework:** Um ORM (Object-Relational Mapping) que foi usado para gerenciar o banco de dados do sistema. O Entity Framework facilita o trabalho com dados, permitindo que os desenvolvedores trabalhem com objetos em vez de se preocupar com a sintaxe SQL.
4. **SQL Express:** O sistema de gerenciamento de banco de dados utilizado para armazenar e gerenciar os dados do sistema. O SQL Express é uma versão leve e gratuita do SQL Server, ideal para o desenvolvimento de aplicações de pequeno e médio porte.
5. **Swagger:** Uma ferramenta usada para projetar, construir e documentar APIs RESTful. No sistema de locadora de veículos, o Swagger foi usado para criar uma documentação interativa da API que é fácil de entender e usar.

## Principais Classes e suas Responsabilidades:

1. **Program:** Esta é a classe principal que inicia o aplicativo. Ela configura o aplicativo e define o pipeline de requisições HTTP.
2. **Controllers:** As classes de controlador são responsáveis por manipular as solicitações do usuário, trabalhar com os modelos conforme necessário e atualizar a visão para refletir quaisquer mudanças. No sistema de locadora de veículos, existem controladores para gerenciar veículos, clientes, reservas, etc.
3. **Models:** As classes de modelo representam os dados e as regras de negócio do sistema. No sistema de locadora de veículos, os modelos incluem entidades como Veículo, Cliente, Reserva, entre outros.
4. **DbContext (ApplicationContext):** Esta classe é responsável por coordenar a funcionalidade do Entity Framework para um modelo de dados. Ela é a principal classe que o Entity Framework interage para ler e gravar dados do banco de dados.

## **Descrição do Banco de Dados:**

O sistema de locadora de veículos utiliza um banco de dados relacional para armazenar e gerenciar seus dados. O banco de dados foi projetado para garantir a integridade dos dados e facilitar consultas complexas. Aqui está uma descrição geral do modelo de dados e das tabelas utilizadas:

1. **Tabela de Veículos:** Esta tabela armazena informações sobre os veículos disponíveis para aluguel. Cada registro na tabela representa um veículo único e inclui campos para marca, modelo, ano, cor, placa e status.
2. **Tabela de Clientes:** Esta tabela armazena informações sobre os clientes que utilizam o sistema de locadora de veículos. Cada registro na tabela representa um cliente único e inclui campos para nome, CPF, data de nascimento, informações de contato e informações de habilitação.
3. **Tabela de Reservas:** Esta tabela armazena informações sobre as reservas feitas pelos clientes. Cada registro na tabela representa uma reserva única e inclui campos para o cliente, o veículo, a data de início da reserva e a data de término da reserva.

Os relacionamentos entre as tabelas são implementados usando chaves estrangeiras. Por exemplo, a tabela de Reservas tem uma chave estrangeira para a tabela de Clientes e uma chave estrangeira para a tabela de Veículos, indicando qual cliente fez a reserva e qual veículo foi reservado.

As restrições de integridade são implementadas para garantir a consistência e a precisão dos dados. Por exemplo, as chaves estrangeiras são usadas para garantir que cada reserva ou aluguel esteja associado a um cliente e a um veículo existentes.



## Testes e Validação:

Os testes são uma parte crucial do desenvolvimento de software para garantir que o sistema funcione conforme esperado e para identificar e corrigir quaisquer problemas ou erros que possam surgir. No desenvolvimento do sistema de locadora de veículos, foram realizados vários testes, tanto manuais quanto automatizados.

1. **Testes Manuais:** Os testes manuais foram realizados utilizando a interface do Swagger. Estes testes envolveram a execução de operações CRUD (Create, Read, Update, Delete) nos endpoints da API para verificar se o sistema estava respondendo corretamente. Por exemplo, um veículo foi criado usando o endpoint de criação de veículos, e então foi verificado se o veículo aparecia corretamente na lista de veículos. Da mesma forma, foram realizados testes para atualizar os detalhes de um veículo, excluir um veículo e recuperar os detalhes de um veículo.
2. **Verificação de Consistência de Dados:** Além dos testes manuais, também foi realizada uma verificação de consistência de dados no banco de dados SQL Express. Isso envolveu a verificação de que os dados no banco de dados correspondiam ao que era esperado com base nas operações realizadas através da API.

A cobertura de testes foi bastante abrangente, cobrindo todas as principais funcionalidades do sistema. No entanto, é importante notar que a cobertura de testes completa é um objetivo difícil de alcançar e sempre há espaço para mais testes, especialmente quando se trata de testar o comportamento do sistema sob condições excepcionais ou raras.

Os testes realizados provaram ser eficazes em garantir que o sistema funcionasse conforme esperado e em identificar áreas onde melhorias poderiam ser feitas. Isso contribuiu para a confiabilidade e robustez do sistema de locadora de veículos.

## **Resultados e Conclusões:**

O desenvolvimento do sistema de locadora de veículos resultou em um sistema robusto e eficiente que atende às necessidades de uma empresa de aluguel de veículos. O sistema foi capaz de gerenciar com sucesso todas as operações de uma locadora de veículos, desde o cadastro de veículos e clientes até a reserva e o aluguel de veículos.

A avaliação da conformidade dos requisitos do sistema mostrou que todos os requisitos funcionais e não funcionais foram atendidos. O sistema foi capaz de realizar todas as operações necessárias e demonstrou ser seguro, confiável e fácil de usar.

A eficácia do sistema desenvolvido em atender às necessidades da locadora de veículos foi alta. O sistema melhorou a eficiência operacional da locadora de veículos e proporcionou uma experiência de usuário superior para os clientes.

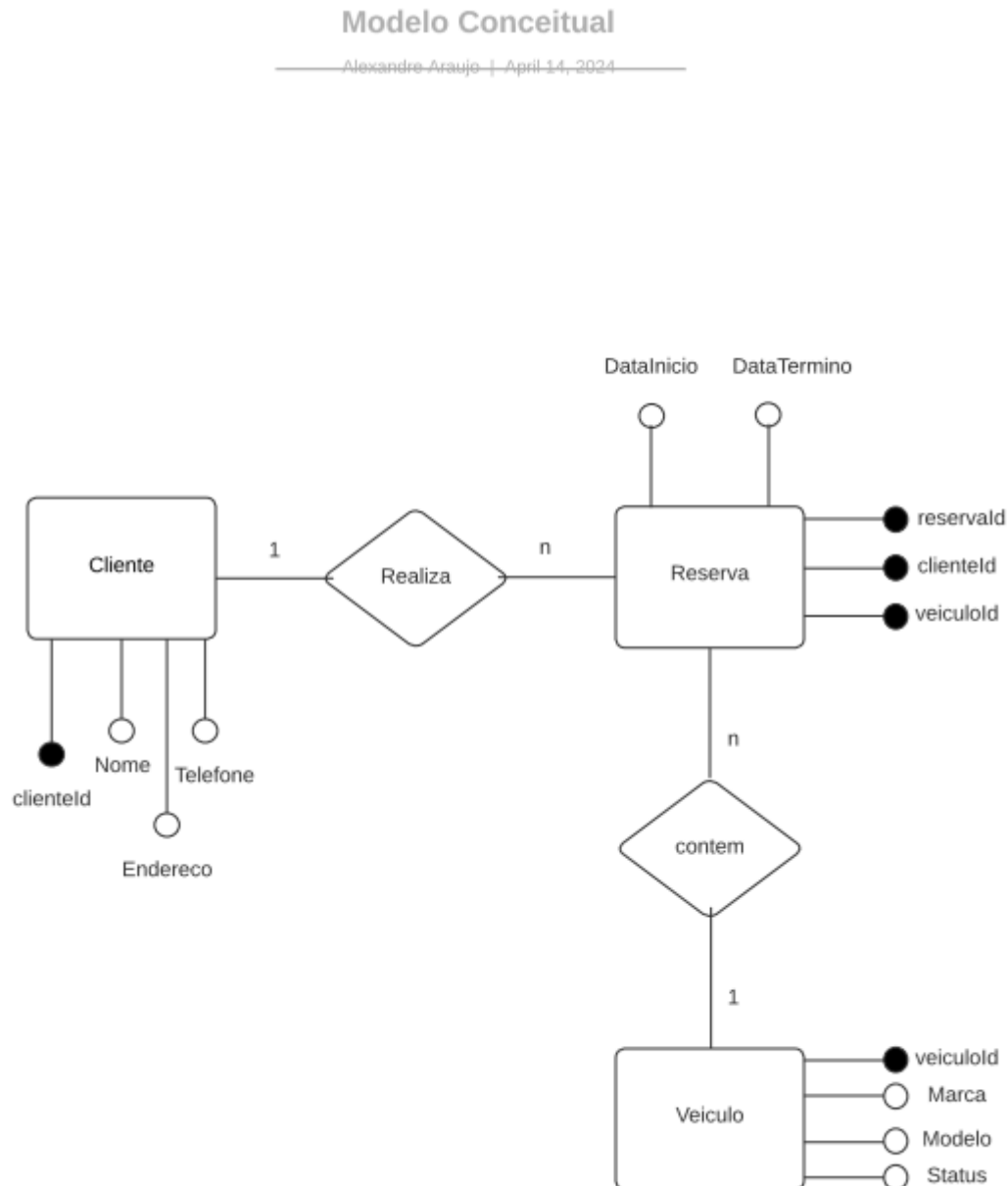
## **Considerações Finais:**

O desenvolvimento deste projeto foi uma experiência valiosa. Foi uma oportunidade para aplicar os conhecimentos adquiridos durante o curso de Ciências da Computação e ganhar experiência prática no desenvolvimento de sistemas complexos.

Para melhorias futuras no sistema, algumas sugestões incluem a inclusão de um sistema de cobrança para facilitar o processo de pagamento, a emissão de relatórios para fornecer insights valiosos sobre o negócio e a implementação de um front-end básico para melhorar a interface do usuário.

## Apêndices:

- **Documentação adicional:** Modelo Conceitual do projeto



Link para vídeo do Pitch:

<https://www.youtube.com/watch?v=chVfJSk1CDs>