**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**EliteCar**Sistema de gestão de vendas de carros

**Sertãozinho  
2025**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

Sumário

[1. REQUISITOS DO SISTEMA 5](#_Toc210893951)

[1.1 Requisitos funcionais 5](#_Toc210893952)

[1.2 Requisitos não funcionais 6](#_Toc210893953)

[1.3.1 RF001 – Cadastrar Cliente 6](#_Toc210893954)

[1.3.2 RF002 – Editar Cliente 7](#_Toc210893955)

[1.3.3 RF003 – Excluir Cliente 7](#_Toc210893956)

[1.3.4 RF004 – Cadastrar Carro 8](#_Toc210893957)

[1.3.5 RF005 – Editar Carro 8](#_Toc210893958)

[1.3.6 RF006 - Excluir Carro 9](#_Toc210893959)

[1.3.7 RF007 – Cadastro de pedido de venda 9](#_Toc210893960)

[1.3.8 RF008 – Editar Pedido de Venda 10](#_Toc210893961)

[1.3.9 RF003 – Consultar Pedido de Venda 10](#_Toc210893962)

[2. MODELAGEM 11](#_Toc210893963)

[2.1 DIAGRAMA DE CLASSE 12](#_Toc210893964)

[2.2 DIAGRAMA DE ENTIDADE-RELACIONAMENTO 13](#_Toc210893965)

# REQUISITOS DO SISTEMA

No contexto do sistema EliteCar, os requisitos suprem as necessidades, capacidades ou condições do usuário de atingir suas metas ou seus objetivos, os requisitos são essenciais para o programador, pois eles servem para orientar o desenvolvimento e assegurar que o sistema do site funcione sem erros conforme o esperado.

## 1.1 Requisitos funcionais

Para garantir o funcionamento eficiente do sistema, suas funcionalidades devem ser claramente definidas. Os requisitos funcionais descrevem as operações essenciais, como controle de estoque, geração de relatórios e gestão de vendas, determinando como o sistema deve responder a diferentes entradas. Um bom detalhamento desses requisitos assegura que o sistema atenda às necessidades dos usuários e funcione corretamente em diversos cenários.

A seguir, a Tabela 1 apresenta a descrição dos requisitos funcionais do sistema de gerenciamento financeiro.

|  |  |
| --- | --- |
| Requisito Funcional | |
| RF001 | Cadastrar Cliente |
| RF002 | Editar Cliente |
| RF003 | Excluir Cliente |
| RF004 | Cadastrar Carro |
| RF005 | Editar Carro |
| RF006 | Excluir Carro |
| RF007 | Cadastrar Pedido de Venda |
| RF008 | Editar Pedido de Venda |
| RF009 | Consultar Pedido de Venda |

Tabela 1 - Requisitos Funcionais

­­

## 1.2 Requisitos não funcionais

A transformação precisa acontecer quase que de imediato, de maneira ágil e eficaz. Para garantir que os números mostrados sejam exatos, a ferramenta deve buscar dados em plataformas de câmbio sempre revisadas e de reputação sólida, prevenindo erros nos cálculos. Adicionalmente, a página precisa ser simples de usar, facilitando a escolha das moedas que serão convertidas, sem gerar confusão. O uso deve ser leve e direto, sem passos extras, fazendo com que qualquer pessoa consiga usar sem problemas. Para finalizar, o valor final da conversão deve ser exibido de forma organizada, nítida e fácil de entender, assegurando que o usuário compreenda o resultado na hora, sem precisar pensar muito.

|  |  |
| --- | --- |
| Requisito Não Funcional | |
| RF0010 | Relatório de venda por período |

Tabela 2 – Requisito Não Funcional

### 1.3.1 RF001 – Cadastrar Cliente

Descrição: O sistema deve permitir que o usuário cadastre novos clientes no acervo do banco.

Prioridade: Alta

Entrada:

* Nome do cliente (obrigatório).
* CPF (obrigatório).
* Telefone (obrigatório).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao cadastrar o cliente.
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro.

Pós-condição:

* O cliente cadastrado deve ser exibida na lista de clientes cadastrados.

### 1.3.2 RF002 – Editar Cliente

Descrição: O sistema deve permitir que o usuário edite as informações de um cliente previamente cadastrado

Prioridade: Alta

Entrada:

* Identificação do cliente a ser cadastrado (ID).
* Nome do cliente (Editável).
* CPF (Editável).
* Telefone (Editável).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvas as alterações.
* Mensagem de erro caso haja duplicidade ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas no cliente e nos seus registros.

### 1.3.3 RF003 – Excluir Cliente

Descrição: O sistema deve permitir que o usuário exclua seu próprio cadastro.

Prioridade: Média.

Entrada:

* Identificação do cliente/usuário (ID)

Processamento:

* O sistema deve verificar se há vendas ativas ou pendentes associados ao cliente.

Se houver vendas associadas, o sistema deve informar por meio de um alerta que possuí metas associadas ao cliente, e perguntar se ele tem certeza de sua exclusão.

Saída:

* Mensagem de sucesso ao excluir a conta.
* Aviso de alerta ao tentar excluir a conta se possuir metas associadas ao cliente.

### 1.3.4 RF004 – Cadastrar Carro

Descrição: O sistema deve permitir que o usuário cadastre novos carros no cadastro do banco.

Prioridade: Alta.

Entrada:

* Marca do carro (obrigatório).
* Modelo do carro (obrigatório).
* Ano de fabricação (obrigatório).
* Cor do carro (obrigatório).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao cadastrar o carro.
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro.

Pós-condição:

* O carro cadastrado deve ser exibida na lista de carros cadastrados.

### 1.3.5 RF005 – Editar Carro

Descrição: O sistema deve permitir que o usuário edite as informações de um carro previamente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do carro a ser editado (ID)
* Marca do carro (Editável).
* Modelo do carro (Editável).
* Ano de fabricação (Editável).
* Cor do carro (Editável).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja duplicidade ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas na lista de carros cadastrados.

### 1.3.6 RF006 - Excluir Carro

Descrição: O sistema deve permitir que o usuário exclua um carro do banco de dados.

Prioridade: Média

Entrada:

* Identificação do carro a ser excluído (ID).

Processamento:

* O sistema deve verificar se há vendas ativas associadas ao carro.
* Se houver metas ativas, o sistema deve impedir a exclusão e informar o usuário.

Saída:

* Mensagem de sucesso ao excluir o carro.
* Mensagem de erro caso exista vendas associadas impedindo a excussão.

Pós-condição:

* O carro deve ser removido da lista de carros e não deve aparecer nas buscas e consultas.

### 1.3.7 RF007 – Cadastro de pedido de venda

Descrição: O sistema deve permitir que o usuário registre o banco de uma venda para um cliente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do cliente (ID).
* Identificação do carro (ID).
* Data da venda (obrigatório).
* Valor total da venda (obrigatório).

Processamento:

* O sistema deve verificar a disponibilidade do carro para a venda.
* O sistema deve associar a venda ao cliente.

Saída:

* Mensagem de sucesso ao registrar o banco.
* Mensagem de erro caso alguma meta não esteja disponível.

### 1.3.8 RF008 – Editar Pedido de Venda

Descrição: O sistema deve permitir que o usuário edite as informações de um pedido de venda previamente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do pedido de venda (ID).
* Identificação do cliente (ID).
* Identificação do carro (ID)
* Data de venda (editável).
* Valor total da venda (editável).

Processamento:

* O sistema deve verificar a disponibilidade dos carros no estoque.
* O sistema deve associar a venda ao cliente.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja duplicidade ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas nos registros do banco.

### 1.3.9 RF003 – Consultar Pedido de Venda

Descrição: O sistema deve permitir que o usuário consulte os pedidos de venda já realizados.

Prioridade: Alta

Entrada:

* Identificação do pedido de venda (ID).

Processamento:

* O sistema deve verificar se há pedido de venda no banco de dados.
* Se houver o sistema deve mostrar suas informações ao usuário.

Saída:

* Informações sobre o pedido de venda.
* Mensagem de erro ao gerar o relatório.

Pós-condição:

* As informações devem ser exibidas em forma de tabela.

# 2. MODELAGEM

Modelagem do de software é uma etapa fundamental no desenvolvimento de sistemas, permitindo a criação de representações estruturadas dos componentes, como Diagramas de classes (define a estrutura do back-end, organizando as classes, atributos, métodos e relações antes da implementação), Diagramas de entidade e relacionamento (DER) (modela a estrutura do banco de dados, garantindo a integridade e organização das informações) e o Mockup (representa a estrutura do front-end, ajudando na definição da interface do usuário antes da implementação), além das funcionalidades e interações do sistema implantado, auxiliando na organização do código. No caso do sistema EliteCar garante a eficiência do software, bem estruturadas antes da aplicação, assegurando que os requisitos funcionais sejam atendidos corretamente.

## 2.1 DIAGRAMA DE CLASSE

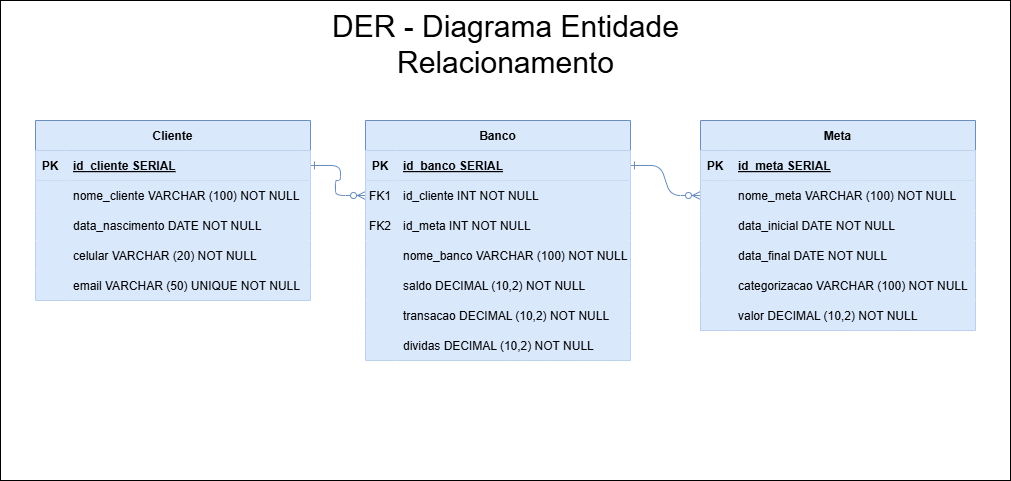
O diagrama de classes organiza as partes do sistema, como "Cliente", "Carro", "PedidoVenda" e as interfaces “ClienteDTO”, “CarroDTO” e “PedidoVendaDTO”, mostrando suas relações. Auxiliando o programador na hora do back-end, na criação dos atributos e métodos, mostrando claramente o que devemos fazer, como os parâmetros das funções serão e como se relacionarão. O diagrama de classes da Finança Fácil tem como por exemplo esses atributos e métodos mostrados na figura 1.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

Figura – Diagrama de Classes

## 2.2 DIAGRAMA DE ENTIDADE-RELACIONAMENTO

O Diagrama Entidade-Relacionamento (DER) é crucial para planejar e estruturar a estrutura de um banco de dados, particularmente em sistemas de administração de finanças, assegurando eficácia, integridade e clareza nas conexões entre os dados.******Parte superior do formulário**

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Figura - Diagrama Entidade-Relacionamento