

# **Trio Bit Garage**

## Documento de Especificação de Requisitos

### **Integrantes do Grupo:**

Fabio Damas Valim  
Caio Finnochio Martins  
Guilherme Lirio Miranda

18 de novembro de 2025

## Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Propósito . . . . .	2
1.2	Escopo do Produto . . . . .	2
1.3	Definições, Acrônimos e Abreviações . . . . .	2
1.4	Visão Geral do Documento . . . . .	2
<b>2</b>	<b>Descrição Geral</b>	<b>3</b>
2.1	Perspectiva do Produto . . . . .	3
2.2	Funções do Produto . . . . .	3
2.3	Características dos Usuários . . . . .	3
2.4	Restrições . . . . .	3
<b>3</b>	<b>Requisitos Funcionais</b>	<b>4</b>
<b>4</b>	<b>Requisitos Não Funcionais</b>	<b>6</b>
<b>5</b>	<b>Diagrama de Casos de Uso</b>	<b>7</b>

# 1 Introdução

## 1.1 Propósito

Este documento tem como objetivo descrever as especificações de requisitos de software para o projeto **Trio Bit Garage**...

## 1.2 Escopo do Produto

O sistema Trio Bit Garage é uma solução voltada para o gerenciamento de locação de veículos. O sistema visa automatizar processos de cadastro de veículos e clientes, controle de locações e registro de pagamentos.

## 1.3 Definições, Acrônimos e Abreviações

- **RF:** Requisito Funcional
- **RNF:** Requisito Não Funcional
- **MVC:** Model-View-Controller (Padrão de Arquitetura)
- **LTS:** Long Term Support (Suporte de Longo Prazo)

## 1.4 Visão Geral do Documento

O restante deste documento está organizado da seguinte forma: A Seção 2 descreve a visão geral do sistema e seus usuários. A Seção 3 lista os Requisitos Funcionais detalhados. A Seção 4 especifica os Requisitos Não Funcionais. A Seção 5 apresenta o Diagrama de Casos de Uso.

## 2 Descrição Geral

### 2.1 Perspectiva do Produto

O sistema atua como uma plataforma centralizada que substitui controles manuais, oferecendo uma interface web para administração e controle das operações da locadora.

### 2.2 Funções do Produto

As principais funções do sistema incluem:

- Gestão completa de veículos e clientes (CRUD).
- Controle de locações (registro, alteração, cancelamento).
- Registro e consulta de pagamentos.
- Geração de relatórios gerenciais.

### 2.3 Características dos Usuários

- **Administrador/Funcionário:** Responsável por operar o sistema, cadastrar dados e gerenciar as locações.

### 2.4 Restrições

- O sistema requer conexão com a internet ou rede local.
- O backend deve seguir estritamente a arquitetura MVC conforme definido nos requisitos não funcionais.

### 3 Requisitos Funcionais

<b>RF001</b>	<b>Realizar Login no Sistema</b>
<b>Tabelas:</b>	Usuário
<b>Descrição:</b>	Permitir que usuários autorizados acessem o sistema através de credenciais válidas.

<b>RF002</b>	<b>Cadastrar Veículo</b>
<b>Tabelas:</b>	Veículo
<b>Descrição:</b>	Permitir o registro de novos veículos no sistema.

<b>RF003</b>	<b>Consultar Veículo</b>
<b>Tabelas:</b>	Veículo
<b>Descrição:</b>	Permitir a busca e visualização de informações dos veículos cadastrados.

<b>RF004</b>	<b>Alterar Veículo</b>
<b>Tabelas:</b>	Veículo
<b>Descrição:</b>	Permitir a atualização de dados de veículos existentes.

<b>RF005</b>	<b>Excluir Veículo</b>
<b>Tabelas:</b>	Veículo
<b>Descrição:</b>	Permitir a remoção lógica/física de veículos do sistema.

<b>RF006</b>	<b>Cadastrar Cliente</b>
<b>Tabelas:</b>	Cliente
<b>Descrição:</b>	Permitir o registro de novos clientes no sistema.

<b>RF007</b>	<b>Consultar Cliente</b>
<b>Tabelas:</b>	Cliente
<b>Descrição:</b>	Permitir a busca e visualização de dados dos clientes cadastrados.

<b>RF008</b>	<b>Alterar Cliente</b>
<b>Tabelas:</b>	Cliente
<b>Descrição:</b>	Permitir a atualização de informações de clientes existentes.

<b>RF009</b>	<b>Excluir Cliente</b>
<b>Tabelas:</b>	Cliente
<b>Descrição:</b>	Permitir a remoção lógica/física de clientes do sistema.

<b>RF010</b>	<b>Registrar Locação</b>
<b>Tabelas:</b>	Locação, Cliente, Veículo
<b>Descrição:</b>	Permitir o registro de uma nova locação vinculando cliente e veículo.

<b>RF011</b>	<b>Consultar Locação</b>
<b>Tabelas:</b>	Locação, Cliente, Veículo
<b>Descrição:</b>	Permitir a busca e visualização de locações com dados do cliente e veículo.

<b>RF012</b>	<b>Alterar Locação</b>
<b>Tabelas:</b>	Locação, Cliente, Veículo
<b>Descrição:</b>	Permitir a modificação de dados de locações existentes.

<b>RF013</b>	<b>Cancelar Locação</b>
<b>Tabelas:</b>	Locação, Cliente, Veículo
<b>Descrição:</b>	Permitir o cancelamento de locações registradas.

<b>RF014</b>	<b>Registrar Pagamento</b>
<b>Tabelas:</b>	Pagamento, Locação, Cliente
<b>Descrição:</b>	Permitir o registro de pagamentos vinculados a locações.

<b>RF015</b>	<b>Consultar Pagamento</b>
<b>Tabelas:</b>	Pagamento, Locação, Cliente
<b>Descrição:</b>	Permitir a visualização de histórico de pagamentos.

<b>RF016</b>	<b>Gerar Relatório de Locações</b>
<b>Tabelas:</b>	Locação, Cliente, Veículo, Pagamento
<b>Descrição:</b>	Permitir a geração de relatórios gerenciais sobre locações realizadas.

## 4 Requisitos Não Funcionais

<b>RNF001</b>	<b>Compatibilidade com Navegadores</b>
<b>Categoria:</b>	Portabilidade
<b>Descrição:</b>	O sistema deve ser compatível com Google Chrome versão 100 ou superior.
<b>Métrica:</b>	Teste de funcionalidade completa no navegador especificado.

<b>RNF002</b>	<b>Criptografia de Dados Sensíveis</b>
<b>Categoria:</b>	Segurança
<b>Descrição:</b>	O sistema deve armazenar senhas de usuários utilizando o algoritmo de hash bcrypt com fator de custo mínimo de 10 rounds.
<b>Métrica:</b>	Validação através de auditoria de código e testes de penetração verificando ausência de senhas em texto plano.

<b>RNF003</b>	<b>Disponibilidade do Sistema</b>
<b>Categoria:</b>	Confiabilidade
<b>Descrição:</b>	O sistema deve manter disponibilidade de 99% durante o horário comercial (08:00 às 18:00, dias úteis), permitindo no máximo 7,2 horas de indisponibilidade não planejada por mês.
<b>Métrica:</b>	Monitoramento através de ferramentas de uptime (ex: UptimeRobot) com registro de logs de disponibilidade.

<b>RNF004</b>	<b>Padrão de Desenvolvimento Backend</b>
<b>Categoria:</b>	Manutenibilidade
<b>Descrição:</b>	O backend do sistema deve ser desenvolvido utilizando Node.js versão 18 LTS ou superior com framework Express.js versão 4.18 ou superior, seguindo o padrão de arquitetura MVC (Model-View-Controller).
<b>Métrica:</b>	Análise de estrutura de pastas e arquivos conforme padrão MVC documentado.

## 5 Diagrama de Casos de Uso

Este diagrama representa a fronteira do sistema e como o Ator principal (Funcionário/Administrador) interage com as funcionalidades (Requisitos Funcionais - RFs). As linhas tracejadas indicam dependências entre as funcionalidades.

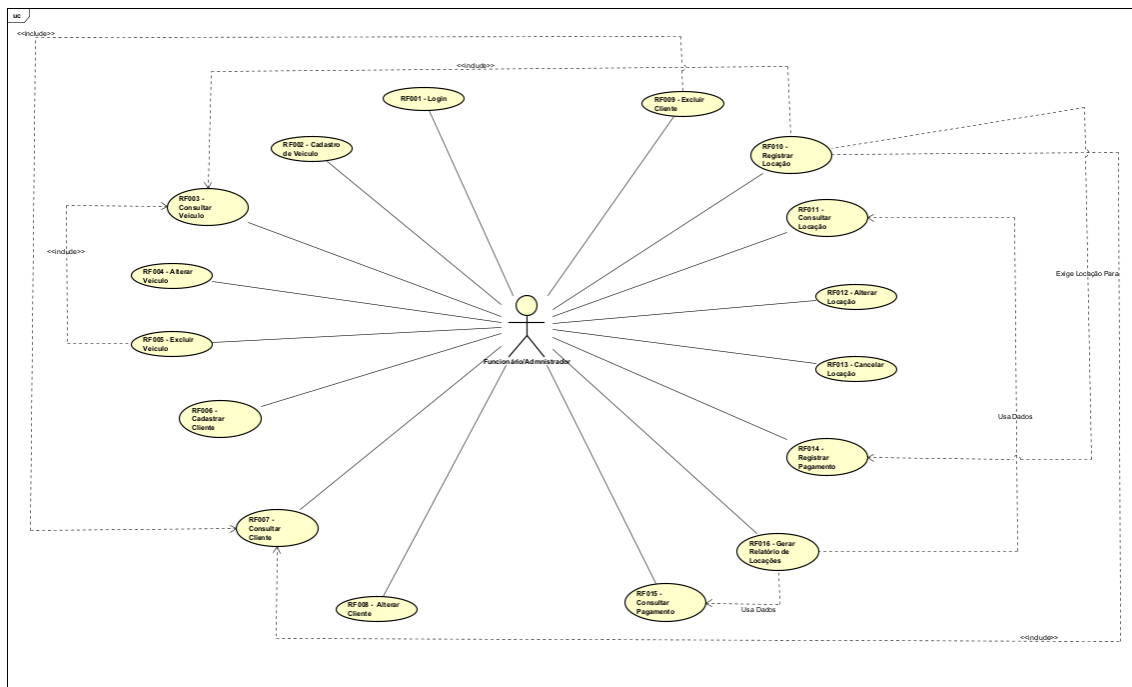


Figura 1: Diagrama de Casos de Uso do Sistema Trio Bit Garage (UML)