

Centro Universitário de Brasília – CEUB

Curso: Ciência da Computação • **Disciplina:** Desenvolvimento de Sistemas

Documento de Especificação de Requisitos de Software (ERS)

Projeto: HOT WHEELS – Sistema de Vendas e Gestão

Versão: 1.0 • **Data:** 12/09/2025

Autores: Francisco Beleza Neto; Arthur Gabriel da Silva Barbosa

1. Introdução

1.1 Objetivo do Documento

Este documento especifica os **requisitos funcionais e não funcionais** do sistema HOT WHEELS, além de regras de negócio e critérios de aceitação, servindo como guia para o desenvolvimento e validação da solução.

1.2 Escopo do Sistema

O sistema permitirá:

- **Login seguro** com autenticação e armazenamento protegido de credenciais.
- **CRUD completo** de clientes, produtos (pneus), e vendas.
- **Relatórios gerenciais** de vendas e estoque.
- **Integração com Swagger** para documentação da API.
- **Execução em ambiente Docker** com banco de dados relacional.

1.3 Definições e Siglas

- **ERS:** Especificação de Requisitos de Software.
- **CRUD:** Create, Read, Update, Delete.

- **API REST:** Interface de programação que segue o padrão REST.
 - **Swagger:** Ferramenta de documentação e teste de APIs.
 - **Docker:** Plataforma de containers para padronização do ambiente.
 - **ORM:** Object–Relational Mapping.
-

2. Descrição Geral

2.1 Perspectiva do Produto

O sistema será uma aplicação web, com backend em **C#/.NET Core**, banco de dados em container Docker e APIs REST documentadas em Swagger.

2.2 Funções do Sistema

- Login e autenticação segura.
- Cadastro e gerenciamento de clientes.
- Cadastro e gerenciamento de produtos (estoque).
- Registro e acompanhamento de vendas.
- Relatórios de vendas e estoque.

2.3 Usuários e Stakeholders

- **Administradores:** acesso total ao sistema.
- **Equipe de vendas:** cadastro de clientes e pedidos.
- **Logística:** atualização e expedição de pedidos.
- **Financeiro:** aprovação de pedidos e relatórios.
- **Clientes:** consulta de status de pedidos.

2.4 Restrições e Premissas

- O sistema deve rodar em ambiente **Windows/Linux**.
 - O banco de dados deve rodar em container **Docker**.
 - Uso de **ORM (Entity Framework Core)** para persistência.
 - **Premissa:** Conexão com internet necessária para uso da API/documentação Swagger.
-

3. Requisitos Funcionais (RF)

Módulo de Autenticação

- **RF01.** Permitir login com autenticação segura (JWT ou similar).
- **RF02.** Armazenar credenciais de forma criptografada.

Módulo de Clientes

- **RF03.** Cadastrar clientes (nome, CPF, telefone, email).
- **RF04.** Consultar, atualizar e excluir clientes.

Módulo de Produtos

- **RF05.** Cadastrar produtos (marca, modelo, medida, preço, quantidade).
- **RF06.** Atualizar estoque automaticamente após venda.

Módulo de Vendas

- **RF07.** Registrar venda (cliente, produto, quantidade, data, vendedor).
- **RF08.** Emitir comprovante digital da venda.

Módulo de Relatórios

- **RF09.** Gerar relatório de vendas por período.

- **RF10.** Gerar relatório de estoque.
-

4. Requisitos Não Funcionais (RNF)

- **RNF01.** O sistema deve ser simples e fácil de usar.
 - **RNF02.** O sistema deve responder em até 3 segundos por requisição.
 - **RNF03.** O sistema deve utilizar autenticação JWT e criptografia de senhas.
 - **RNF04.** O banco deve rodar em container Docker.
 - **RNF05.** O código deve seguir princípios de **Clean Code** e arquitetura limpa.
 - **RNF06.** O sistema deve suportar múltiplos usuários simultâneos.
 - **RNF07.** A API deve ser documentada no **Swagger**.
-

5. Regras de Negócio (RN)

- **RN01.** A venda só pode ocorrer se houver estoque disponível.
 - **RN02.** O cliente deve estar cadastrado para efetuar a compra.
 - **RN03.** Pedidos com pendências financeiras não podem ser aprovados.
 - **RN04.** Cancelamentos só são permitidos antes da expedição.
-

6. Critérios de Aceitação (CA)

- **CA01.** O login deve autenticar corretamente com credenciais válidas.
- **CA02.** O estoque deve ser atualizado automaticamente após cada venda.
- **CA03.** Relatórios de vendas devem exibir corretamente os dados por período.

- **CA04.** O sistema deve impedir vendas sem estoque.
 - **CA05.** O Swagger deve listar e permitir teste de todos os endpoints CRUD.
-