

Documento de Escopo, Requisitos e Regras de Negócio - Projeto Gestor Simplificado

Versão: 1.1

Data: 25 de setembro de 2025

Última atualização em: 08 de outubro de 2025

Autor: Felipe da Silva

Objetivo: Criar um sistema web para que usuários possam cadastrar suas vendas e despesas, permitindo a visualização de relatórios simples sobre a saúde financeira do seu negócio.

Tecnologias Principais:

- **Backend:** Node.js, Express, Prisma ORM
 - **Frontend:** React, Tailwind CSS
 - **Banco de Dados:** PostgreSQL (É um banco de dados relacional robusto, perfeito para dados financeiros e relatórios, e uma ótima escolha para usar com Docker e Prisma)
 - **Ambiente:** Docker e Docker Compose
-

Requisitos Funcionais (O que o sistema deve fazer):

- **RF01 - Autenticação de Usuário:**
 - O usuário deve poder criar uma conta (registro).
 - O usuário deve poder entrar no sistema (login).
 - O sistema deve manter o usuário logado enquanto ele navega.
 - O usuário deve poder sair do sistema (logout).
- **RF02 - Dashboard Principal:**
 - Ao logar, o usuário verá um painel com um resumo rápido: total de vendas do mês, total de despesas do mês e o saldo (vendas - despesas).
- **RF03 - Gestão de Produtos/Serviços:**
 - O usuário deve poder cadastrar, editar, visualizar e excluir seus produtos ou serviços (ex: Nome, Descrição, Preço Padrão).
- **RF04 - Registro de Vendas:**
 - O usuário deve poder registrar uma venda, associando um ou mais produtos/serviços, a data e o valor total.
- **RF05 - Registro de Despesas:**
 - O usuário deve poder registrar uma despesa (ex: Descrição, Valor, Categoria - "Aluguel", "Marketing", "Material", etc.).
- **RF06 - Geração de Relatórios:**
 - O sistema deve gerar um relatório de vendas por período (ex: últimos 30 dias).
 - O sistema deve gerar um relatório de despesas por categoria em um determinado período.

Requisitos Não-Funcionais (Como o sistema deve ser):

- **RNF01 - Segurança:** As senhas dos usuários devem ser armazenadas de forma segura no banco de dados (usando hash).
- **RNF02 - Usabilidade:** A interface deve ser limpa, intuitiva e responsiva para funcionar bem em desktops e celulares.
- **RNF03 - Containerização:** Todo o ambiente de desenvolvimento (backend, frontend, banco de dados) deve ser configurado para rodar com um único comando (`docker-compose up`).
- **RNF04 - Performance:** As consultas ao banco de dados e as respostas da API devem ser otimizadas para serem rápidas.

Regras de Negócio (As lógicas do sistema):

- **RN01:** Um usuário só pode visualizar e gerenciar os dados (vendas, despesas, produtos) que ele mesmo cadastrou.
- **RN02:** Uma venda deve estar obrigatoriamente associada a um usuário.
- **RN03:** Uma despesa deve estar obrigatoriamente associada a um usuário.
- **RN04:** Todos os valores monetários (preços, despesas, vendas) devem ser maiores que zero.
- **RN05:** Não é possível excluir um produto que já esteja associado a um registro de venda (para manter a integridade do histórico).