

Roadmap de Desenvolvimento - Projeto Gestor Simplificado

Versão: 1.1

Data: 25 de setembro de 2025

Última atualização em: 08 de outubro de 2025

Autor: Felipe da Silva

Este é um plano passo a passo, dividido em fases para organizar o desenvolvimento.

Fase 1: Configuração do Ambiente e Fundação do Projeto (O Grande Desafio do Docker!)

1. **Versionamento:** Iniciar o projeto com `git init`.
2. **Estrutura de Pastas:** Criar a estrutura do monorepo: uma pasta `backend`, uma `frontend` e o `docker-compose.yml` na raiz.
3. **Docker Compose:**
 - Criar o arquivo `docker-compose.yml`.
 - Definir os serviços: `database` (usando a imagem oficial do PostgreSQL), `backend` (usando um Dockerfile para sua API Node) e `frontend` (usando um Dockerfile para seu app React).
 - Configurar volumes para o banco de dados e para o código, para que as alterações reflitam em tempo real nos containers.
4. **Backend - Conexão Inicial:**
 - Configurar o Express e o Prisma na pasta `backend`.
 - Conectar o Prisma ao banco de dados PostgreSQL que está rodando no container.
 - Criar o schema inicial do Prisma (pelo menos o model `User`).
 - Rodar a primeira migração do Prisma para criar a tabela de usuários.

Fase 2: Construção do Backend (API Core)

1. **Autenticação:** Implementar as rotas de registro e login (usando JWT para gerar tokens).
2. **Middleware de Proteção:** Criar um middleware que verifica o token JWT para proteger as rotas que exigem autenticação.
3. **CRUDs:** Desenvolver os endpoints (rotas, controllers, services) para:
 - Produtos/Serviços
 - Vendas
 - Despesas

Fase 3: Construção do Frontend (Interface e Interação)

1. **Setup Inicial:** Configurar o React com Vite e Tailwind CSS na pasta `frontend`.
2. **Roteamento:** Implementar o `react-router-dom` para criar as rotas públicas (Login, Registro) e privadas (Dashboard, Vendas, etc.).

3. **Autenticação no Front:** Criar as telas de Login e Registro e a lógica para salvar o token JWT.
4. **Layout Principal:** Construir o layout das páginas privadas (ex: um menu lateral de navegação e uma área de conteúdo principal).
5. **Telas de CRUD:** Desenvolver as páginas para listar, criar, editar e excluir produtos, vendas e despesas, integrando com a API do backend.

Fase 4: Relatórios e Dashboard (O "Gran Finale")

1. **Endpoints de Relatórios no Backend:** Criar rotas específicas na API que fazem as consultas mais complexas no banco (usando as funções de agregação do Prisma) para gerar os dados do dashboard e dos relatórios.
2. **Componentes do Dashboard:** No frontend, criar os componentes que vão consumir os dados dos relatórios e exibir os totais (vendas, despesas, saldo).
3. **Visualização de Dados (Bônus):** Integrar uma biblioteca de gráficos simples (como [Recharts](#) ou [Chart.js](#)) para exibir os dados de forma visual no dashboard e na página de relatórios.