

# Plano de Desenvolvimento - Projeto eSellers

## **Sprint 0 Preparacao do Ambiente**

### **- Tarefa 1 Criar repositorios no GitHub**

O que fazer:

- Criar repositorios separados para frontend e backend.
- Adicionar README com instrucoes basicas.

Referencia: <https://docs.github.com/en/get-started/quickstart/create-a-repo>

### **- Tarefa 2 Configurar Docker e PostgreSQL**

O que fazer:

- Criar docker-compose com servicos para o banco de dados e backend.

Referencia: <https://docs.docker.com/compose/gettingstarted/>

### **- Tarefa 3 Configurar ambiente base do frontend**

O que fazer:

- Iniciar projeto com Vite ou Create React App.
- Estrutura base com pastas: pages, components, styles, services

Referencia: <https://vitejs.dev/>

## **Sprint 1 Autenticacao e Layout Base**

### **- Tarefa 1 Implementar JWT no backend**

O que fazer:

- Implementar login, registro e autenticacao JWT.

Referencia: <https://jasonwatmore.com/post/2022/01/19/net-6-jwt-authentication-tutorial-with-example-api>

### **- Tarefa 2 Layout base no frontend**

O que fazer:

- Criar layout padrao com header, sidebar, content.
- Usar CSS Modules com SCSS.

Referencia: <https://reactjs.org/docs/components-and-props.html>

### **- Tarefa 3 Fluxo de login no frontend**

# Plano de Desenvolvimento - Projeto eSellers

O que fazer:

- Criar tela de login.
- Armazenar token JWT com contexto e localStorage.

Referencia: <https://reactrouter.com/en/main/start/tutorial>

## **Sprint 2 Gestao de Produtos e Estoque**

### **- Tarefa 1 Entidade Produto**

O que fazer:

- Criar entidade Produto com: nome, SKU, preco, estoque.

Referencia: <https://learn.microsoft.com/pt-br/ef/core/modeling/>

### **- Tarefa 2 CRUD de produtos no backend**

O que fazer:

- Endpoints: GET, POST, PUT, DELETE de produtos.

Referencia: <https://learn.microsoft.com/pt-br/aspnet/core/web-api/>

### **- Tarefa 3 Telas no frontend**

O que fazer:

- Criar telas: ProductList, ProductCreate, ProductEdit.
- Conectar a API com Axios ou Fetch.

Referencia: <https://axios-http.com/>

## **Sprint 3 Modulo de Pedidos**

### **- Tarefa 1 Criar entidades Pedido e PedidoItem**

O que fazer:

- Modelos no backend: Pedido e PedidoItem

Referencia: <https://learn.microsoft.com/pt-br/ef/core/modeling/relationships/>

### **- Tarefa 2 CRUD de Pedidos e Itens no backend**

O que fazer:

- Endpoints para criar, listar e atualizar pedidos.

## Plano de Desenvolvimento - Projeto eSellers

Referencia: <https://code-maze.com/net-core-web-api-with-ef-core-db-relationships/>

### - Tarefa 3 Paginas de pedidos no frontend

O que fazer:

- Criar telas: OrderList, OrderDetails, OrderCreate

Referencia: <https://react-select.com/home>

### - Tarefa 4 Conexao com API e calculo de totais

O que fazer:

- Atualizar valores dos pedidos conforme os itens.

Referencia: <https://reactjs.org/docs/hooks-effect.html>

## Sprint 4 Gestao de Clientes e Fornecedores

### - Tarefa 1 Criar entidades Cliente e Fornecedor

O que fazer:

- Nome, CNPJ/CPF, Endereco, etc.

Referencia: <https://docs.fluentvalidation.net/en/latest/>

### - Tarefa 2 CRUD completo no backend

O que fazer:

- Criar endpoints com validacoes e paginacao.

Referencia: <https://www.paginateanything.dev/articles/pagination-entity-framework-core>

### - Tarefa 3 Telas no frontend

O que fazer:

- Listagem e cadastro com mascaras e validacoes.

Referencia: <https://github.com/sanniassin/react-input-mask>

## Sprint 5 Modulo Financeiro (F&I)

### - Tarefa 1 Criar entidades Financeiro

O que fazer:

## **Plano de Desenvolvimento - Projeto eSellers**

- Lancamentos com tipo, valor, data, categoria.

Referencia: <https://www.devmedia.com.br/implementando-um-contas-a-receber-com-ddd-e-cqrs/34776>

### **- Tarefa 2 Endpoints financeiros**

O que fazer:

- GET e POST com filtros por cliente e data.

Referencia: <https://learn.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-7.0#controller-actions>

### **- Tarefa 3 Relatorios no frontend**

O que fazer:

- Dashboard com graficos de receitas/despesas.

Referencia: <https://www.react-chartjs-2.dev/>

## **Sprint 6 Nota Fiscal Eletronica (NF-e) + SEFAZ**

### **- Tarefa 1 Estudo tecnico da SEFAZ**

O que fazer:

- Compreender XMLs, certificados, Web Services.

Referencia: <https://www.nfe.fazenda.gov.br/portal/principal.aspx>

### **- Tarefa 2 Implementar geracao de XML e assinatura digital**

O que fazer:

- Gerar XMLs e assinar digitalmente com A1.

Referencia: <https://github.com/adeniltonbs/Zeus.Net.NFe.NFCe>

### **- Tarefa 3 Envio para SEFAZ e retorno**

O que fazer:

- Implementar SOAP, tratar rejeicoes.

Referencia: [https://www.youtube.com/watch?v=HbMz3nThX\\_8](https://www.youtube.com/watch?v=HbMz3nThX_8)

## **Sprint Futura Integracao com Canais de Revenda**

### **- Tarefa 1 Estudo de viabilidade e APIs publicas**

## Plano de Desenvolvimento - Projeto eSellers

O que fazer:

- Verificar APIs ou dados acessíveis nos portais.

Referencia: Portais Avon, Natura, etc.

### - Tarefa 2 Estrategia de scraping ou middleware

O que fazer:

- Automatizacao com Puppeteer/Playwright.

Referencia: <https://pptr.dev/>

### - Tarefa 3 Criar painel de revenda

O que fazer:

- Tela com pedidos, comissoes, credenciais.

Referencia: