

Relatório de etapas do IU Marketplace

Nomes: Natã Cezer Bordignon, Gabriel Felipe Martins

1. Introdução

O presente projeto, intitulado IU Marketplace, consiste em uma aplicação web voltada à compra e venda de produtos usados. O objetivo é proporcionar uma plataforma simples e funcional onde usuários possam se cadastrar, anunciar produtos e adquirir itens de terceiros. O sistema foi desenvolvido como parte do Projeto Integrador V, com foco em aplicações web modernas utilizando TypeScript e Node.js.

2. Planejamento do projeto

A escolha do tema baseou-se na necessidade comum de venda e compra de produtos usados com facilidade, de forma segura e acessível a qualquer pessoa.

2.2 Tecnologias Utilizadas

- Linguagem: TypeScript
- Backend: Node.js + Express
- Frontend: HTML + React + CSS
- Banco de Dados: PostgreSQL
- ORM: TypeORM
- Autenticação: JWT
- Ambiente: Docker e Docker Compose
- Gerenciamento de dependências: npm
- Ferramentas auxiliares: Adminer, dotenv

3. Arquitetura do sistema

3.1 Estrutura das pastas

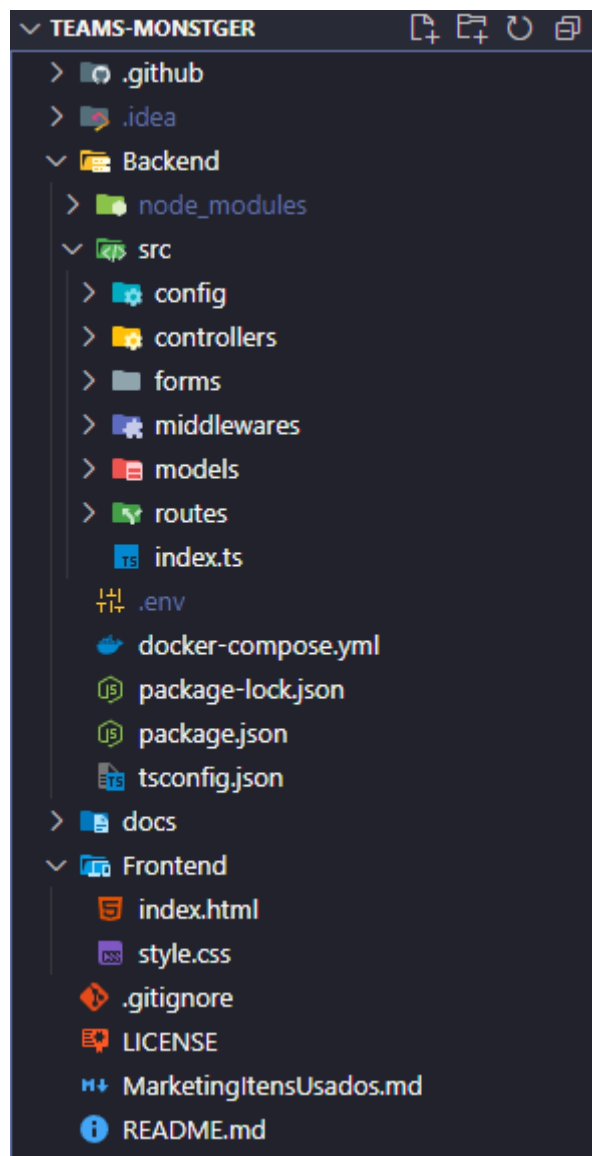


Imagem 1: estruturação das pastas

3.2 Modelagem de dados

- Usuário (User): id, username, email, password, produtos, carrinho.
- Produto (Product): id, nome, descrição, preço, imagem, categoria, disponível.
- Carrinho (Cart): id, userId, productId, quantity.

4. Rotas da API

4.1.2 Usuário

- POST /api/users/register: Cadastro de novos usuários.
- POST /api/users/login: Login e geração de token.
- GET /api/users/auth: Validação de token.

4.3.2 Produto

- POST /api/products/register: Cadastro de produtos.
- PUT /api/products/:id: Edição de produto.
- DELETE /api/products/:id: Remoção.
- GET /api/products/all: Listagem.
- GET /api/products/:id: Detalhes de um produto.

4.3.3 Carrinho

- POST /api/cart/add: Adição de produto ao carrinho.
- GET /api/cart/all: Itens do carrinho.
- PUT /api/cart/:id: Atualização de item.
- GET /api/cart/:id: Visualização individual.
- DELETE /api/cart/:id: Remoção de item.

4.4 Scripts npm

- npm run dev: executa o projeto com hot reload.
- npm run build: compila os arquivos TypeScript.
- npm start: executa versão compilada.
- npm run kill-port: encerra processos na porta 3000 (Windows).