

Projeto Sistema de Obras de Arte - Romero Brique

Visão Geral

Este projeto tem como objetivo desenvolver uma aplicação web para gerenciar as **obras de arte da empresa Romero Brique**, permitindo:

- Login de usuários.
 - Cadastro, listagem, edição e exclusão (CRUD) de obras de arte.
 - Integração entre frontend e backend usando **Next.js**.
 - Persistência dos dados em um **banco PostgreSQL local**, com **Prisma** como ORM.
-

Design (UI/UX)

- Criar protótipo no **Figma** com as telas:
 - Tela de login
 - Dashboard de obras
 - Cadastro/edição de obra
 - Listagem de obras
 - Detalhes da obra
-

🔧 Tecnologias Utilizadas

Frontend

- **Next.js 15+**
- **React 19**
- **TypeScript**
- **Tailwind CSS**
- **shadcn/ui** (componentes)

Backend

- **Next.js (API Routes)**
- **Prisma ORM**
- **PostgreSQL (local)**
- **JWT** (para autenticação)

Funcionalidades de Autenticação

- Registro e login de usuários com senha criptografada (bcrypt).
- Sessão com **JWT** (Access Token + Refresh Token).
- Middleware para proteger rotas privadas.

CRUD de Obras de Arte

Cada obra conterá os seguintes campos:

- **ID** (UUID)
- **Título**
- **Artista**
- **Ano**
- **Descrição**
- **Imagem da obra** (upload/local ou S3 futuro)
- **Data de criação** (timestamp automático)

Estrutura do Banco de Dados (Prisma Schema)

```
model User {
  id          String    @id @default(uuid())
  name        String
  email       String    @unique
  password    String
  createdAt   DateTime  @default(now())
  updatedAt   DateTime  @updatedAt
}

model Artwork {
  id          String    @id @default(uuid())
  title       String
  artist      String
  year        Int
  description  String?
  imageUrl    String?
  createdAt   DateTime  @default(now())
  updatedAt   DateTime  @updatedAt
}
```

Estrutura de Pastas

```
romero-brique-app/
├─ prisma/
│   └─ schema.prisma
│   └─ app/
│       └─ login/
│       └─ dashboard/
│       └─ obras/
│           └─ page.tsx (listagem)
│           └─ [id]/page.tsx (detalhes/edição)
│           └─ new/page.tsx (cadastro)
│   └─ api/
│       └─ auth/
│           └─ login.ts
│           └─ register.ts
│       └─ artworks/
│           └─ route.ts (CRUD)
├─ components/
├─ lib/
│   └─ prisma.ts
│   └─ auth.ts
```

Passos do Desenvolvimento

1. Criar design no **Figma**.
2. Inicializar projeto com Next.js e Tailwind.
3. Configurar PostgreSQL local.
4. Configurar Prisma e gerar migrations.
5. Implementar autenticação (registro/login).
6. Criar CRUD de obras (API + frontend).
7. Integrar frontend com API.
8. Testar fluxo completo (login → cadastrar → editar → excluir obra).

Futuras Melhorias

- Upload de imagens em serviço externo (S3/Cloudinary).
- Controle de permissões (admin x usuário comum).
- Filtros e busca avançada de obras.
- Dashboard com métricas de artistas/obras.

Status Atual: Planejamento inicial.