

Documentação de Instalação e Uso

Sumário

- [Documentação de Instalação e Uso](#)
 - [Sumário](#)
 - [1. Visão Geral do Projeto](#)
 - [2. Configuração do Ambiente](#)
 - [2.1 Requisitos](#)
 - [2.2 Instalação](#)
 - [2.3 Instalação para Desenvolvimento](#)
 - [Backend \(Django\)](#)
 - [Instalar dependências:](#)
 - [Configurar banco de dados PostgreSQL](#)
 - [Executar migrações](#)
 - [Criar superusuário \(opcional\)](#)
 - [Caso queira criar manualmente antes de entrar no site:](#)
 - [Iniciar servidor](#)
 - [Frontend \(React\)](#)
 - [Instalar dependências](#)
 - [Iniciar](#)
 - [3. Utilização das Funcionalidades](#)
 - [4. Testes](#)
-

1. Visão Geral do Projeto

Este projeto consiste o backend em Python (Django) e frontend em TypeScript (React). O objetivo é fornecer funcionalidades de autenticação e gerenciamento de livros.

2. Configuração do Ambiente

2.1 Requisitos

- [Python 3.9+](#)
- [Node.js 18+](#)
- [Yarn](#)
- [Docker](#)
- [Docker Compose](#)

2.2 Instalação

1. Clonar o repositório:

```
git clone https://github.com/leonzv/dpsc-atividade.git
cd dpsc-atividade
```

2. Criar o arquivo `.env`:

- Copie o arquivo `.env.example` para `.env` na pasta raiz e preencha com as suas configurações.

```
cp .env.example .env
```

- Copie o arquivo `.env.example` para `.env` para Frontend e Backend:

```
cd dpsc-atividade-frontend
cp .env.example dpsc-atividade-frontend/.env

cd dpsc-atividade-backend
cp .env.example dpsc-atividade-backend/.env
```

- **Obs:** As configurações de banco de dados no backend e no arquivo raiz `.env` devem ser iguais para garantir a conectividade correta.

3. Subir os contêineres:

```
docker-compose up --build
```

4. Verificar a execução:

- Backend disponível em `http://localhost:8000`
- Frontend disponível em `http://localhost:3000`

Deixe o PostgreSQL rodando em container para que não precise instalá-lo diretamente no seu sistema :D

- PostgreSQL: Rodando na porta 5432 no host

2.3 Instalação para Desenvolvimento

Backend (Django)

1. Criar e ativar o ambiente virtual:

◦ Windows:

```
python -m venv venv
venv\Scripts\activate
```

◦ Linux/MacOS:

```
python3 -m venv venv
source venv/bin/activate
```

Instalar dependências:

```
cd dpvc-atividade-backend
pip install -r requirements.txt
```

Configurar banco de dados PostgreSQL

Executar migrações

```
python manage.py migrate
```

Criar superusuário (opcional)

Por padrão, foi criado um usuário admin com as seguintes credenciais:

Email: admin@admin.com

Username: admin

Senha: admin123

Você também pode se registrar no próprio site :D

Caso queira criar manualmente antes de entrar no site:

```
python manage.py createsuperuser
```

Iniciar servidor

```
python manage.py runserver
```

Frontend (React)

Instalar dependências

```
cd dpvc-atividade-frontend
yarn install
```

Iniciar

```
yarn dev
```

Disponível em <http://localhost:3000>.

3. Utilização das Funcionalidades

- **Autenticação:**
 - Login, registro e acesso a páginas protegidas.
 - **Funcionalidades Adicionais:**
 1. Você só pode criar, editar e deletar se estiver autenticado.
 2. Se não estiver autenticado, o menu de criar não aparecerá e botões de deletar também não estarão visíveis.
 3. As rotas para criar e editar estão protegidas e não podem ser acessadas sem autenticação.
- **Gerenciamento dos livros:**
 - Cadastro, listagem, edição e remoção de livros.
 - **Funcionalidades Adicionais:**
 1. Na listagem de livros, você pode editar e excluir cada item.
 2. Para criar um novo livro, clique no menu "Adicionar Livro" na barra lateral.
 3. O formulário de criação possui menus de sugestões ao lado direito, digitando um autor ou título, as sugestões começam a aparecer.
 4. Clicar em uma sugestão preenche automaticamente os campos correspondentes.
 - **Admin Django:** Área de controle de administração, acesse <http://localhost:8000/admin>.

4. Testes

- **Backend:** `pytest` (se configurado o ambiente de desenvolvimento).
- **Frontend:** `yarn test` (se configurado o ambiente de desenvolvimento).