

# README para rodar o Projeto de BD1

Github do projeto: [kenjiThiago/ProjetoBD1](https://github.com/kenjiThiago/ProjetoBD1)

## Requisitos

### Backend:

- **Python:** 3.12.7
  - **Nota:** No Windows, pode ser necessário instalar o *Microsoft C++ Build Tools*, pois o Python depende de suporte para compilar alguns componentes em C.

### Frontend:

- **Node.js:** 22.11.0
- **NPM:** 10.9.0

### Banco de Dados:

- **PostgreSQL:** 16.6

### Opcional:

- **Git:** Qualquer versão recente, caso precise clonar o repositório do projeto no GitHub.

**Observação:** Estas são as versões utilizadas durante o desenvolvimento e teste do projeto. Outras versões podem funcionar, mas não foram verificadas.

## Etapas para rodar o projeto

### Windows

#### Importante:

Para evitar problemas ao utilizar o ambiente virtual do Python, prefira usar o **Prompt de Comando (cmd)** em vez do **PowerShell**. No PowerShell, configurações adicionais são necessárias para executar o ambiente virtual do Python, enquanto no cmd isso não é necessário.

### Linux

Para rodar no Linux, basta substituir o comando da etapa 3 do backend. Ao invés de usar `.\venv\Scripts\activate`, utilize `source .venv/bin/activate` para ativar o ambiente virtual.

## SQL

O script SQL usado para gerar o banco de dados do projeto está localizado no diretório "Códigos SQL".

```
C:\Users\thiag\ProjetoBD1>cd "Códigos SQL"
```

```
29/11/2024  17:46    <DIR>          .
29/11/2024  17:14    <DIR>          ..
29/11/2024  17:14             127.392 projetoBD_Script.sql
```

### 1. Criar o Banco de Dados

- Acesse o **pgAdmin** (ou outra interface gráfica do PostgreSQL) e crie um novo banco de dados.
- Abra o diretório "**Códigos SQL**" no projeto e execute o script **projetoBD\_Script.sql** para criar o banco de dados com todas as tabelas e dados necessários.

## Importante

Antes de seguir com as instruções para configurar o backend, edite o arquivo:

**ProjetoBD1\Backend\Database\conector.py**

Este arquivo contém as configurações de conexão com o banco de dados e precisa ser ajustado para refletir o nome, usuário e senha do banco criado.

```
(venv) C:\Users\thiag\ProjetoBD1\Backend\Database>dir
```

```
29/11/2024  17:14                1.839 classe_aluno.py
29/11/2024  17:14                2.135 classe_curso.py
29/11/2024  17:14                1.311 classe_empresa.py
29/11/2024  17:14                1.036 classe_professor.py
29/11/2024  17:14                5.309 classe_vaga.py
29/11/2024  17:14                1.810 conector.py
29/11/2024  17:14                 0 __init__.py
29/11/2024  17:41          <DIR>          __pycache__
```

```
8     def __init__(self) -> None:
9         self.conn = psycopg2.connect(
10             dbname="trabalhodb",
11             user="postgres",
12             password="1234",
13             host="127.0.0.1",
14             port=5432,
15         )
```

No arquivo **ProjetoBD1\Backend\Database\conector.py**, atualize os seguintes parâmetros:

- **password**: Substitua pela senha que você utiliza no PostgreSQL.
- **dbname**: Substitua pelo nome do banco de dados que você criou.

## Backend

### 1. Acessar o Diretório do Backend

- Navegue até o diretório **ProjetoBD1\Backend**.

```
C:\Users\thiag\ProjetoBD1>cd Backend  
C:\Users\thiag\ProjetoBD1\Backend>
```

### 2. Criando o Ambiente Virtual em python (venv)

- No diretório **ProjetoBD1\Backend**, rode o comando **python -m venv venv**.  
Esse comando irá criar um subdiretório no diretório Backend chamado venv.

```
C:\Users\thiag\ProjetoBD1\Backend>python -m venv venv
```

29/11/2024	17:19	<DIR>	.
29/11/2024	17:14	<DIR>	..
29/11/2024	17:14	<DIR>	Database
29/11/2024	17:14		1.394 main.py
29/11/2024	17:14		56 requirements.txt
29/11/2024	17:14	<DIR>	Rotas
29/11/2024	17:19	<DIR>	<b>venv</b>
29/11/2024	17:14		0 __init__.py

### 3. Iniciando o venv

- Execute o comando `.\venv\Scripts\activate` no terminal.
- Após executar o comando, um indicador `(venv)` aparecerá no início da linha do terminal, confirmando que o ambiente virtual do Python foi ativado.

```
C:\Users\thiag\ProjetoBD1\Backend>.\venv\Scripts\activate  
(venv) C:\Users\thiag\ProjetoBD1\Backend>
```

### 4. Instalando as Dependências para o Backend

- No diretório `ProjetoBD1\Backend`, execute o comando `pip install -r requirements.txt` para instalar todas as dependências listadas no arquivo `requirements.txt`, necessárias para rodar o backend.

```
(venv) C:\Users\thiag\ProjetoBD1\Backend>pip install -r requirements.txt
```

### 5. Rodando o Servidor Backend

- Execute o comando `python main.py` para rodar o servidor backend.

```
(venv) C:\Users\thiag\ProjetoBD1\Backend>python main.py  
* Serving Flask app 'main'  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:8000  
* Running on http://192.168.0.10:8000  
Press CTRL+C to quit
```

## Frontend

### 1. Primeiros Passos

- Abra um novo terminal e navegue até o diretório **frontend**.  
**Importante:** Não feche o terminal onde o servidor backend está rodando.

```
C:\Users\thiag\ProjetoBD1>cd frontend  
C:\Users\thiag\ProjetoBD1\frontend>
```

### 2. Instalando Dependências para Rodar o Frontend

- No diretório **frontend**, execute o comando **npm install** para instalar todas as dependências necessárias para rodar o frontend.

```
C:\Users\thiag\ProjetoBD1\frontend>npm install  
  
added 13 packages, and audited 14 packages in 2s  
  
3 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
  
C:\Users\thiag\ProjetoBD1\frontend>
```

### 3. Rodando o Frontend

- Execute o comando `npm run build` para realizar o build dos arquivos do frontend.

```
C:\Users\thiag\ProjetoBD1\frontend>npm run build

> frontend@0.0.0 build
> vite build

vite v5.4.10 building for production...
✓ 78 modules transformed.
dist/index.html 1.68 kB | gzip: 0.80 kB
dist/empresas/vagas/index.html 2.83 kB | gzip: 1.07 kB
dist/empresas/index.html 2.84 kB | gzip: 1.06 kB
dist/vagas/index.html 2.96 kB | gzip: 1.08 kB
dist/assets/logo-DZxWEcFq.png 2.96 kB
dist/alunos/index.html 3.05 kB | gzip: 1.10 kB
dist/vagas/alunosInscritos/index.html 3.15 kB | gzip: 1.16 kB
dist/cursos/index.html 3.15 kB | gzip: 1.14 kB
dist/cursos/alunos/index.html 3.25 kB | gzip: 1.18 kB
dist/alunos/vagas/index.html 3.25 kB | gzip: 1.17 kB
dist/alunos/vagas/cursos/index.html 3.36 kB | gzip: 1.21 kB
dist/dashboardAluno/index.html 4.79 kB | gzip: 1.57 kB
dist/assets/dashboardAluno-B6TxAvxP.css 0.97 kB | gzip: 0.45 kB
dist/assets/graph-DbTihgxG.css 1.15 kB | gzip: 0.52 kB
dist/assets/header-X7LvJPqD.css 2.10 kB | gzip: 0.76 kB
dist/assets/tableButtons-Cbb7BwAr.css 2.87 kB | gzip: 0.99 kB
dist/assets/tableButtons-C4qSsxMR.js 0.59 kB | gzip: 0.28 kB
dist/assets/main-orZkmy0b.js 1.17 kB | gzip: 0.47 kB
dist/assets/orderingButton-DRTmq63E.js 1.22 kB | gzip: 0.49 kB
dist/assets/empresas-CmK58Es9.js 1.74 kB | gzip: 0.83 kB
dist/assets/alunos-CpH0wi-w.js 1.76 kB | gzip: 0.86 kB
dist/assets/cursos-C5zTWVz5.js 1.79 kB | gzip: 0.87 kB
dist/assets/vagasEmpresa-CnS1ueCK.js 1.90 kB | gzip: 0.91 kB
dist/assets/vagas-CSD9l0X3.js 1.91 kB | gzip: 0.89 kB
dist/assets/alunosInscritos-BwMypFR8.js 2.13 kB | gzip: 0.99 kB
dist/assets/alunosCurso-CYTo5x_q.js 2.35 kB | gzip: 1.06 kB
dist/assets/cursosNecessarios-BP9ZWtu6.js 2.40 kB | gzip: 1.06 kB
dist/assets/vagasAluno-BFPGr_6d.js 2.48 kB | gzip: 1.09 kB
dist/assets/dashboardAluno-g-DoGHkA.js 4.00 kB | gzip: 1.84 kB
dist/assets/header-pUL_vQ3w.js 9.95 kB | gzip: 5.46 kB
dist/assets/graph-CI106sX1.js 218.47 kB | gzip: 75.49 kB
✓ built in 862ms

C:\Users\thiag\ProjetoBD1\frontend>
```

- Após o build, execute o comando `npm run preview` para iniciar o servidor do frontend.

```
C:\Users\thiag\ProjetoBD1\frontend>npm run preview
```

```
> frontend@0.0.0 preview
```

```
> vite preview
```

```
→ Local:    http://localhost:4173/
```

```
→ Network: use --host to expose
```

```
→ press h + enter to show help
```

- Para entrar no frontend acesse o <http://localhost:4173/>

