

# Instruções para Executar o Código

## Pré-requisitos

- **Node.js** instalado em sua máquina (versão 12 ou superior)
- **npm** (gerenciador de pacotes do Node.js)
- **PostgreSQL** instalado e em execução
- Um editor de código ou IDE de sua preferência

## Passos para Configuração e Execução

### 1. Clonar o Repositório ou Copiar os Arquivos

Se você estiver usando um repositório Git, clone-o:

```
git clone <URL do repositório>
```

### 2. Instalar as Dependências

Instale as dependências necessárias:

```
npm install express body-parser jsonwebtoken bcrypt pg
```

### 3. Configurar o Banco de Dados PostgreSQL

- Certifique-se de que o PostgreSQL está instalado e em execução em sua máquina.
- Abra o terminal ou prompt de comando e acesse o cliente PostgreSQL:

```
psql -U postgres
```

- Crie um novo banco de dados chamado `estacio`:

```
CREATE DATABASE estacio;
```

- *(Opcional)* Crie um usuário dedicado:

```
CREATE USER seu_usuario WITH PASSWORD 'sua_senha'; GRANT ALL PRIVILEGES ON DATABASE  
estacio TO seu_usuario;
```

- Atualize as configurações de conexão no código, caso tenha criado um novo usuário. Certifique-se de que as credenciais de acesso ao banco de dados (usuário, senha, host, nome do banco e porta) estão corretas no seu arquivo `app.js`.

### 4. Executar o Script para Criar as Tabelas

O código já inclui funções para criar as tabelas `users` e `contracts` se elas não existirem. Ao iniciar a aplicação, as tabelas serão criadas automaticamente.

## 5. Configurar a Chave Secreta do JWT

No código, há uma variável `SECRET_KEY` usada para assinar os tokens JWT. Certifique-se de que a chave secreta está definida corretamente no código. Por razões de segurança, em um ambiente de produção, é recomendável usar variáveis de ambiente para armazenar informações sensíveis, mas como você não está usando um arquivo `.env`, mantenha a chave secreta definida diretamente no código com atenção à segurança.

## 6. Iniciar a Aplicação

Execute o seguinte comando no terminal:

```
node pratica.js
```

Você deverá ver a mensagem:

```
Servidor está executando na porta 3000 Tabelas criadas ou já existentes.
```

## 7. Testar a API

Use um cliente HTTP como **Postman** para testar os endpoints da API.

Importe o arquivo `postman_import.json`.

### Endpoints Disponíveis:

- `POST /api/auth/register` - Registrar um novo usuário
- `POST /api/auth/login` - Autenticar um usuário e obter um token JWT
- `GET /api/profile` - Obter os dados do usuário autenticado
- `GET /api/users` - Obter todos os usuários (apenas para administradores)
- `POST /api/contracts` - Criar um novo contrato (apenas para administradores)
- `GET /api/contracts` - Obter contratos existentes (apenas para administradores)

### Passos para Testar:

#### 1. Registrar um Usuário

Envie uma requisição `POST` para `http://localhost:3000/api/auth/register` com as informações necessárias do usuário (username, password, email, perfil).

#### 2. Autenticar o Usuário

Envie uma requisição `POST` para `http://localhost:3000/api/auth/login` com o nome de usuário e senha.

Você receberá um token JWT na resposta.

#### 3. Acessar Endpoints Protegidos

Use o token JWT obtido no passo anterior no cabeçalho `Authorization` no formato `Bearer token`.

#### 4. Criar um Contrato

Envie uma requisição `POST` para `http://localhost:3000/api/contracts` com os detalhes do contrato (empresa, data\_inicio, detalhes).

Certifique-se de incluir o cabeçalho `Authorization` com o token JWT de um usuário administrador.

#### 5. Obter Contratos

Envie uma requisição `GET` para `http://localhost:3000/api/contracts`.

Você pode adicionar parâmetros de consulta para filtrar os resultados, por exemplo:

`http://localhost:3000/api/contracts?empresa=EmpresaXYZ&inicio=2023-09-15`

## Possíveis Erros e Soluções

- **Erro de Conexão com o Banco de Dados:**

- Verifique se o PostgreSQL está em execução.
- Confirme se as credenciais de acesso estão corretas.
- Verifique se as configurações de conexão no código estão corretas.

- **Erro ao Instalar Dependências:**

- Certifique-se de que o Node.js e o npm estão instalados corretamente.
- Execute `npm cache clean --force` e tente instalar novamente.

- **Problemas com o JWT:**

- Assegure-se de que a `SECRET_KEY` é a mesma para geração e verificação do token.
- Verifique se o token está sendo enviado corretamente no cabeçalho `Authorization`.