

# Sistema de Autenticação com Node.js, Express e PostgreSQL

---

Este projeto é um sistema de autenticação de usuários que permite cadastro, login, visualização, edição, exclusão e exportação de dados em PDF. As senhas dos usuários são protegidas utilizando criptografia com **bcrypt**.

## ✦ Funcionalidades

- ☒ Cadastro de usuário com: nome, e-mail, senha e avatar
- ☒ Login com e-mail e senha
- ☒ Criptografia de senha com bcrypt
- ☒ Tela de sucesso exibindo todos os dados do usuário
- ☒ Edição de dados do usuário
- ☒ Exclusão de conta
- ☒ Download de informações do usuário em formato PDF

## 🔧 Tecnologias Utilizadas

- [Node.js](#)
- [Express](#)
- [PostgreSQL](#)
- [bcrypt](#)
- [pg \(node-postgres\)](#)
- [pdfkit](#) (para exportar PDF)
- [multer](#) (upload de avatar)
- [dotenv](#) (variáveis de ambiente)

## 📦 Instalação

1. Clone o repositório:

```
git clone https://github.com/Gbmonte9/hw13-login
cd hw13-login
```

2. Instale as dependências:

```
npm install
```

3. Configure seu **.env** com suas credenciais do PostgreSQL:

```
DB_HOST=localhost
DB_PORT=5432
DB_USER=seu_usuario
DB_PASSWORD=sua_senha
DB_DATABASE=nome_do_banco
PORT=3000
```

#### 4. Crie a tabela no PostgreSQL:

```
CREATE TABLE usuarios (
  id SERIAL PRIMARY KEY,
  nome VARCHAR(100),
  email VARCHAR(100) UNIQUE,
  senha VARCHAR(200),
  avatar TEXT
);
```

## Criptografia com bcrypt

As senhas são criptografadas ao cadastrar um novo usuário, garantindo mais segurança. O código básico usado:

```
const bcrypt = require('bcrypt');
const senhaCriptografada = await bcrypt.hash(req.body.senha, 10);
```

Durante o login:

```
const senhaValida = await bcrypt.compare(senhaDigitada, senhaDoBanco);
```

## Rotas Principais

Rota	Método	Descrição
/cadastro	POST	Cadastra novo usuário com avatar e senha criptografada
/index	POST	Autentica usuário com email e senha
/usuario/:id	GET	Retorna dados do usuário
/usuario/:id/atualizar	PUT	Edita dados do usuário
/usuario/:id/deleta	DELETE	Remove usuário
/usuario/:id/download	GET	Gera e baixa PDF com os dados do usuário

## Upload de Avatar

Utiliza **multer** para upload da imagem de perfil. As imagens são salvas localmente (ou podem ser adaptadas para salvar em um serviço como AWS S3).

## Exportação para PDF

Os dados do usuário podem ser baixados em PDF com a rota **/usuario/:id/download**.

---

## Como Rodar

```
npm start
```

Abra o navegador e acesse: <http://localhost:3000>

---

## Autor

Desenvolvido por **Gabriel Monte**

 gabrielmonte485@gmail.com

 [LinkedIn](#)

---

## Licença

Este projeto está sob a licença MIT.