

Página de Login e Registro (FormTrilhas)

1. Visão Geral do Projeto

Descrição

O projeto tem como objetivo fornecer aos candidatos a opção de cadastro e login, permitindo a coleta de informações essenciais para um processo seletivo. Na página de registro, o candidato pode criar uma conta informando nome, e-mail e senha. Além disso, há a possibilidade de adicionar dados complementares, como CPF, CEP, estado (UF), logradouro e, se desejar, anexar um currículo. Dessa forma, o sistema oferece flexibilidade para que cada candidato forneça as informações que considerar relevantes para sua candidatura.

2. Tecnologias Utilizadas

2.1 Front-end

- Linguagem: [Javascript]
- Framework/Biblioteca: [React.js, TailwindCSS]
- Axios
- Gerenciador de Pacotes: [npm]

2.2 Back-end

- Linguagem: [Node.js]
- Framework/Biblioteca: [Express.js]
- JWT (Json Web Token)
- Gerenciador de Pacotes: [npm]

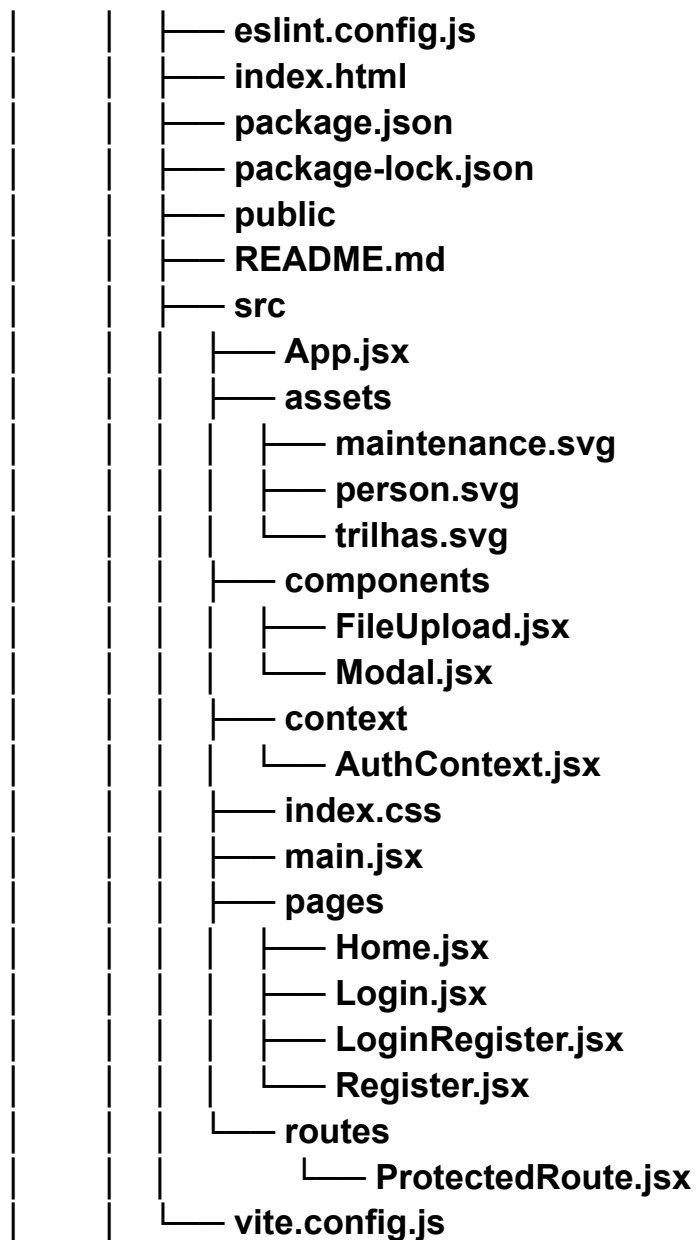
2.3 Banco de Dados

- SGBD: [MongoDB]
- ORM: [Prisma]
- Tipo: NoSQL

3. Estrutura do Projeto

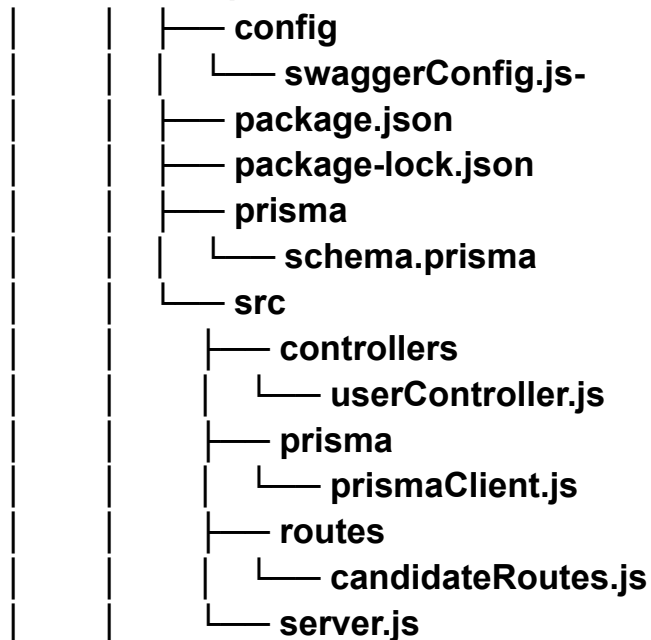
3.1. Front-end

form-trilhas-front



3.2. Back-end

form-trilhas-api



3.3. Banco de Dados (NoSQL JSON)

```
{
  "_id": "67da251879ad3abe3b82",
  "externalId": "3f272b9a-8a2e-4bc1",
  "name": "José",
  "email": "jose@email.com",
  "password": "$2b$10$US4uMIXldgSYMCLI",
  "cpf": "12345678909",
  "cep": "6578729",
  "uf": "MA",
  "logradouro": "Av 1",
  "role": "USER",
  "createdAt": "2025-03-19T01:59:52.716+00:00",
  "updatedAt": "2025-03-19T01:59:52.716+00:00"
}
```

4. Configuração do Projeto

4.1. Configuração do Front-end

1. Acesse a pasta **form-trilhas-front** e instale as dependências:

```
cd form-trilhas-front  
npm install
```

2. Inicie o front-end:

```
npm run dev
```

4.2. Configuração do Back-end

3. Acesse a pasta **form-trilhas-api** e instale as dependências:

```
cd form-trilhas-api  
npm install
```

4. Crie um arquivo **.env** e configure as variáveis de ambiente:

```
DATABASE_URL = <URL_DO_BANCO>  
JWT_SECRET   = <CHAVE_JWT>
```

5. Inicie o servidor:

```
npm run dev
```

5. Autenticação com JWT

O sistema utiliza **JWT** para autenticação dos usuários. O fluxo é o seguinte:

1. O usuário faz login enviando email e senha.
2. O servidor verifica as credenciais e retorna um **token JWT**.
3. O frontend armazena esse token no **localStorage**.
4. Para acessar rotas protegidas, o token deve ser enviado no **header Authorization**.

6. Executando a Aplicação

1. Inicie o Back-end (`cd form-trilhas-api && npm run dev`)
2. Inicie o Front-end (`cd form-trilhas-front && npm run dev`)
3. Acesse a aplicação via navegador em `http://localhost:5173`

7. Funcionalidades Principais

7.1. Registro

1. O sistema permite que novos usuários se cadastrem preenchendo as seguintes informações:

- **Obrigatórias:**

- Nome Completo
- E-mail
- Senha de acesso (necessária para login posterior)

- **Opcionais**

- CPF
- CEP
- UF (Estado)
- Logradouro
- Upload de Currículo (caso for necessário)

7.2. Regras e Validações

- O sistema valida todos os campos e impede o cadastro caso algum esteja em branco, exibindo uma mensagem de aviso na tela.
- Se todos os dados forem preenchidos corretamente, o sistema confirma o registro com um aviso e permite que o usuário faça login.
- A senha do usuário passa por um processo de hash (criptografia) para ser salva no banco de dados.
- Autenticação segura via JWT para manter a sessão do usuário ativa.

7.3. Fluxo de Registro

- Acessar a página de registro
- Preencher o formulário com os dados obrigatórios e se preferir, preencher as informações adicionais e fazer upload de um currículo
- O sistema valida as informações
- Se houver erros, o sistema exibe mensagens para correção
- Se os dados estiverem corretos, o sistema confirma o cadastro com uma mensagem de sucesso
- O usuário pode fazer login com as credenciais cadastradas

7.4. Login

- Os usuários podem acessar o sistema informando:
 - E-mail
 - Senha
- Após preencher esses campos, basta clicar no botão **"ENTRAR"** para acessar a conta.

7.5. Regras e Validações

- Caso o usuário digite sua senha errada o sistema retorna uma notificação com o aviso (Invalid Password).
- Caso email não seja encontrado no banco de dados o sistema retorna uma notificação com o aviso (User not found)
- Se tudo estiver correto, o sistema gera um **token JWT** para o usuário poder acessar paginas protegidas.

8. Considerações Finais

- Este projeto oferece um sistema simples e flexível para cadastro e login de candidatos. Sua flexibilidade garante que possa evoluir continuamente, facilitando melhorias e inovações ao longo do tempo. O objetivo é assegurar um processo simples, eficiente e intuitivo para os usuários, promovendo uma experiência otimizada e alinhada às necessidades futuras.

8. Links

1. API: <https://form-trilhas-api.onrender.com>
2. Swagger Docs: <https://form-trilhas-api.onrender.com/api-docs/>
3. Aplicação: <https://form-trilhas-front.vercel.app/>
4. Repositórios:
 - <https://github.com/KawannSouza/form-trilhas-front>
 - <https://github.com/KawannSouza/form-trilhas-api>