

Documentação do Projeto de Gerenciamento de Usuários

Tecnologias:

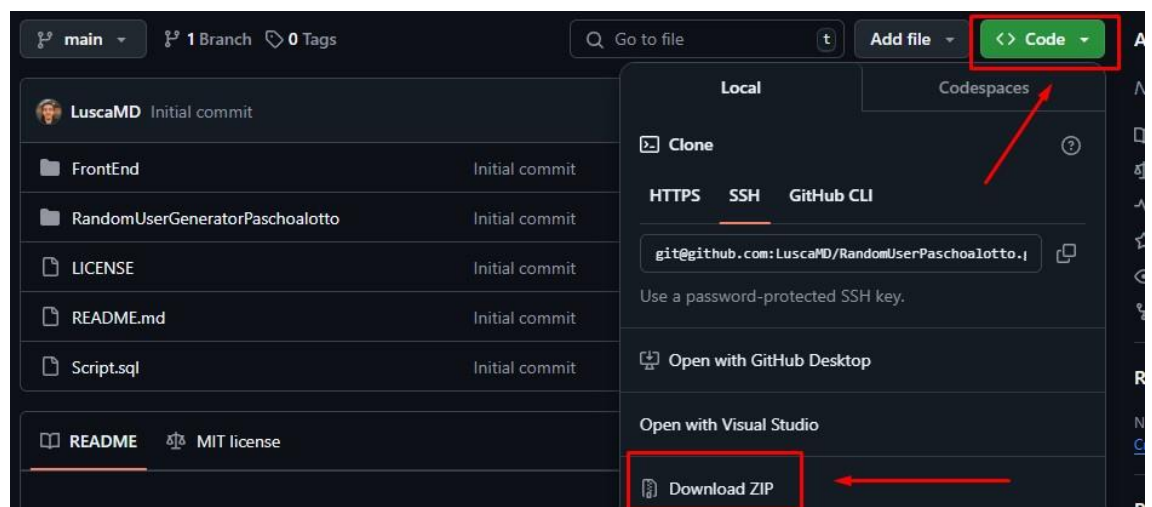
- C#
- ASP.NET Core Web API
- EntityFramework Core
- Swagger
- PostgreSQL
- HTML
- CSS
- JavaScript

Instruções de uso:

Clonagem do Projeto:

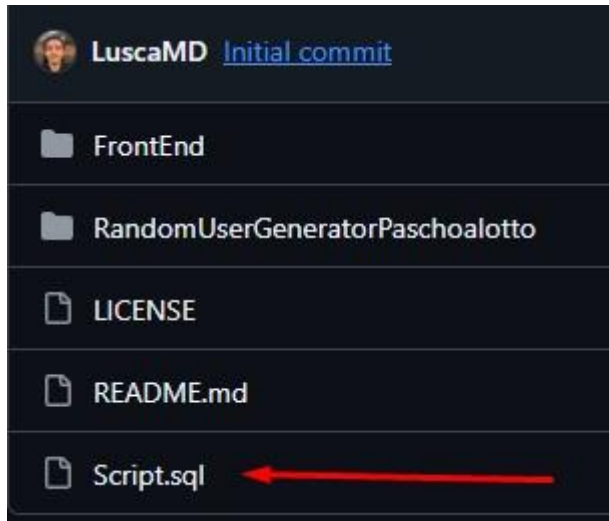
Para obter uma cópia do projeto, siga os passos abaixo:

1. Acesse o link do repositório: [RandomUserPaschoalotto](https://github.com/LuscaMD/RandomUserPaschoalotto) (<https://github.com/LuscaMD/RandomUserPaschoalotto>)
2. Clique no botão Code e selecione Download ZIP.
3. Extraia o conteúdo do arquivo ZIP para uma pasta de sua escolha.

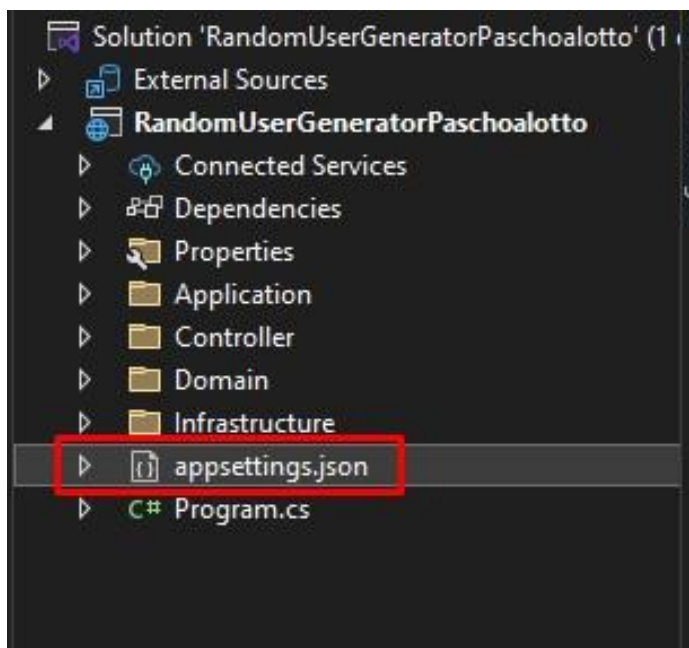


Configuração do Banco de dados:

1. No diretório do projeto, encontre o arquivo Script.sql.
2. Execute o script Script.sql no seu banco de dados PostgreSQL para criar o banco e a tabela necessária para o projeto.

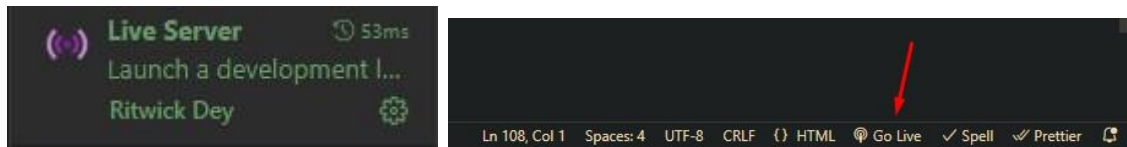


- 3.
4. A connection string do projeto está localizada no arquivo appsettings.json. Você pode:
 - Criar um usuário “postgres” com a senha “lucas123” no seu banco de dados PostgreSQL, ou
 - Alterar a connection string para utilizar um usuário e senha já existentes no seu ambiente.



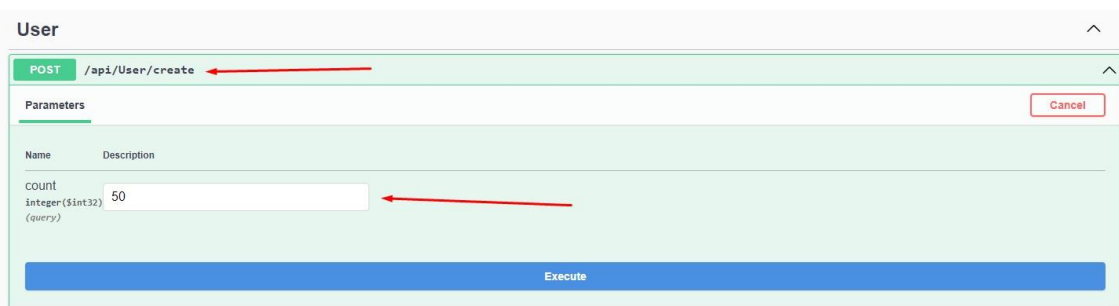
Inicialização do Front-end:

1. Abra o **Visual Studio Code**.
2. Instale a extensão Live Server (caso ainda não tenha).
3. Navegue até a pasta FrontEnd do projeto.
4. No canto inferior direito do Visual Studio Code, clique em “Go Live” para iniciar o servidor local.
5. Acesse o front-end do projeto no navegador pelo endereço fornecido pelo Live Server.



Inicialização do Back-end (API):

1. Abra o projeto no **Visual Studio**.
2. Execute o projeto. O Swagger será iniciado automaticamente no navegador, exibindo todas as rotas disponíveis para a API.
3. Para inserir dados iniciais, utilize a rota POST `/api/User/add` no Swagger:
4. No campo count, insira o número de registros que deseja criar no banco de dados.



Testando as Rotas da API:

GET	/api/User/{id}	▼
GET	/api/User/all	▼
PUT	/api/User/update/{id}	▼
DELETE	/api/User/delete/{id}	▼

- **GET /api/User/{id}**: Retorna os dados do usuário correspondente ao id informado.
- **GET /api/User/all**: Retorna todos os usuários cadastrados.
- **PUT /api/User/update/{id}**: Atualiza as informações do usuário com o id especificado.
- **DELETE /api/User/delete/{id}**: Remove o usuário com o id informado.