Memorial Descritivo - E-Terapias

Discentes: Isaac Allef Santos Cruz e Maria Luíza Teixeira Santos.

BACK-END

Para o desenvolvimento back-end do projeto, utilizamos a biblioteca Node, da linguagem JavaScript, assim como outras dependências importantes, como o Express, e para o banco de dados, o Postgres.

A estrutura do projeto consiste basicamente em uma pasta src, que usamos para o desenvolvimento da aplicação, e arquivo knexfile.js com configurações para conexão entre knex e do banco de dados. Essa pasta contém outras três pastas: config, database, handdles, middlewares e models, além de dois arquivos, index.js e routes.js.

- Na pasta config, temos a chave secreta para codificação das senhas dos moderadores;
- Na pasta apiGoogleSheets, temos conexões com Google Sheets;
- Na pasta controllers, temos os arquivos com regras de negócio para gerenciamento de dados do banco;
- Na pasta database, temos uma pasta contendo as migrations utilizadas para o banco de dados, pasta contendo os arquivos seeds, para povoamento do mesmo, e um arquivo connection.js para conexão do knex, query builder utilizado no projeto;
- Na pasta handdles, temos arquivos que facilitam tanto a codificação das senhas dos moderadores, manipulação de objetos e de strings;
- Na pasta Middlewares, temos o middleware utilizado para autenticação do moderador;
- Na pasta Models, temos os models de algumas tabelas criadas no banco de dados, como os encontros das e-terapias, os moderadores e os participantes;
- No arquivo index.js, temos as configurações para criação do servidor;

- No arquivo routes.js, temos as rotas do projeto.

FRONT-END

Para o desenvolvimento front-end do projeto, utilizamos a biblioteca React, da linguagem JavaScript, assim como outras dependências importantes para a utilização da mesma, como a react-dom e a react-router-dom. Dentro das demais utilizadas, temos a styled-components e o pacote typescript.

A estrutura do projeto consiste em uma pasta public, com arquivos já criados ao iniciarmos um projeto em React, com logos, arquivo json e html, e uma pasta src, que usamos para o desenvolvimento da aplicação. Essa pasta contém outras três pastas: assets, components e pages, além de dois arquivos, index.tsx e App.tsx.

- Na pasta assets, podemos encontrar as imagens utilizadas como logos no projeto, o brasão da universidade e a logo oficial do projeto e-terapias;
- Na pasta components, temos os componentes utilizados na aplicação, que seriam basicamente os que utilizamos para formar as páginas. No React, temos a possibilidade de separar cada item que vai compor uma página como um componente. Dessa forma, podemos estilizar separadamente e dividir certas regiões de páginas em blocos, que posteriormente podem ser reutilizados em outras páginas;
- Na pasta pages, temos as páginas que vão compor nossa aplicação. Essas páginas são as utilizadas para criarmos as rotas de acesso;
- No arquivo index.tsx, informamos ao React DOM o que desejamos renderizar;
- No arquivo App.tsx, basicamente informamos as rotas que irão compor a aplicação, assim como possíveis estruturas comuns entre todas elas, como é o caso do Header, e do arquivo de estilização em comum, o GlobalStyles.ts.

Rotas:

- http://localhost:3000/ - Página de login do moderador;
- http://localhost:3000/profile - Página de login do moderador;

- http://localhost:3000/project - Página com informações sobre as e-terapias participadas pelo moderador;
- http://localhost:3000/frequency - Página com informações sobre o a lista de presença dos participantes da e-terapia escolhida pelo moderador;
- http://localhost:3000/change-password- Página para alteração de senha do moderador;

INSTALAÇÃO

Para instalação das dependências do projeto, basta extrair o arquivo eterapias-master.zip, acessar a pasta, abrir o terminal e:

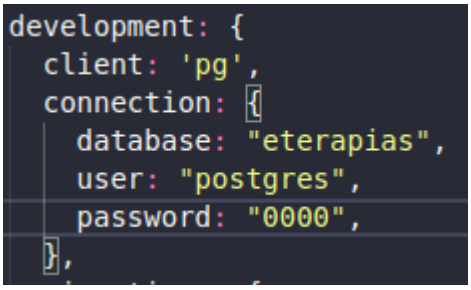
- Na pasta backend
  - Executar comandos
    - npm/yarn install
    - npm/yarn install nodemon -D
    - npx knex migrate:rollback caso esteja repetindo esse processo
    - npx/yarn knex migrate:latest
    - npx/yarn knex seed:run
    - npm/yarn start
  - URL
    - http://localhost:3333/

- Na pasta fronted
  - Executar comandos
    - npm/yarn install
    - npm/yarn start
  - URL
    - http://localhost:3000/

Com esses comandos, todas as libs necessárias serão instaladas automaticamente e o projeto está pronto para ser executado.

Obs.:

- É necessário ter o Postgres no sistema operacional, seja por meio da instalação padrão do próprio quanto com a utilização de um docker, por exemplo.
  - Criar database para o banco de dados
  - Modificar o usuário, senha e database para os dados do seu Postgres no objeto development arquivo knexfile.js



- Caso ocorra algum problema com os pacotes, utilizar comando npm/yarn upgrade

DADOS PARA TESTE

- Login 1
  - Usuário: daniela
  - Senha: 111
- Login 2
  - Usuário: ana
  - Senha: 222
- Login 3
  - Usuário: rosane
  - Senha: 333

- Login 4
  - Usuário: jessica
  - Senha: 444