

# Dossier de projeto

O presente documento serve para indicar como está organizado o dossier deste projeto.

## Repositório

O projeto encontra-se no repositório [https://github.com/BernardoPoiares/PS\\_1920v-Fowyv](https://github.com/BernardoPoiares/PS_1920v-Fowyv)

No repositório é possível encontrar a diretoria Fowyv, referente à implementação da aplicação móvel, assim como a diretoria Fowyv\_Backend, onde se encontra a implementação da aplicação servidora.

Seguidamente é descrito o conteúdo de cada diretoria.

## Fowyv

### Estrutura

A aplicação encontra-se estruturada de acordo com a seguinte estrutura:

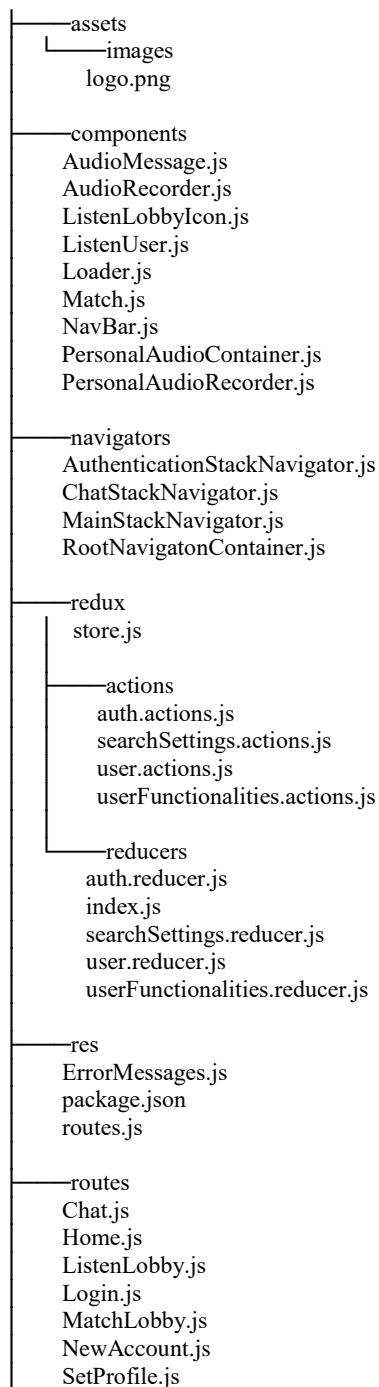
```
Fowyv
├── .buckconfig
├── .eslintrc.js
├── .flowconfig
├── .gitattributes
├── .gitignore
├── .prettierrc.js
├── .watchmanconfig
├── App.js
├── app.json
├── babel.config.js
├── index.js
├── metro.config.js
├── package-lock.json
├── package.json
├── .vscode
├── android
├── app
├── ios
└── __tests__
```

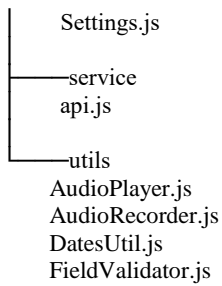
Esta estrutura foi criada maioritariamente através da interface de linha de comando exposta para o *React Native*, *React Native CLI*, que permite a criação de um novo projeto e que

serviu de base para o projeto da aplicação móvel, onde forem gerados os ficheiros e diretorias necessárias para a compilação e configuração da aplicação para os devidos ambientes nativos, assim como o *entry point* do projeto, o ficheiro *index.js*, responsável por registar o componente exposto no *app.js* como o componente principal da aplicação. A diretoria *app/* foi criada posteriormente com o intuito de albergar o código relativo aos componentes a serem desenvolvidos para implementar as funcionalidades desejadas.

A estrutura da diretoria *app* é a apresentada abaixo:

*app*





Na diretoria *assets/* encontra-se o conteúdo estático da aplicação, que por enquanto apenas contém a imagem do logo da aplicação utilizada no ecrã inicial.

Seguidamente podemos verificar a diretoria *components/*, onde se encontram os componentes que são reutilizados em toda a aplicação.

A diretoria *navigators/* contém os componentes responsáveis por definir a navegação entre os diferentes ecrãs.

Quanto à diretoria *res/*, serve para incluir ficheiros considerados como recursos que permitam referenciar constantes que sejam partilhadas pelo resto da aplicação, facilitando a definição das mesmas e alteração num único sítio se for necessário no futuro.

Relativamente à diretoria *routes/*, é nela que se encontram todos os componentes responsáveis pela *renderização* dos ecrãs disponíveis na aplicação.

A diretoria *utils/*, contém ficheiros que permitam partilhar funcionalidades comuns entre os diferentes componentes.

Relativamente à diretoria *service*, a mesma apenas contém o componente responsável pela comunicação com a *api* exposta pela aplicação servidora.

Por último encontramos a *redux*, onde se encontram os componentes responsáveis pela gestão do estado da aplicação e utilização da biblioteca *javascript redux*.

## Fowyv\_Backend

### **Estrutura**

Fowyv\_Backend

```
graph TD
    Fowyv_Backend --> .gitignore
    Fowyv_Backend --> app.js
    Fowyv_Backend --> package-lock.json
    Fowyv_Backend --> package.json
    Fowyv_Backend --> start.js
    Fowyv_Backend --> .vscode
    .vscode --> launch.json
    Fowyv_Backend --> config
    config --> auth.config.js
    Fowyv_Backend --> controllers
    controllers --> auth.controller.js
    controllers --> interactions.controller.js
    controllers --> search.controller.js
    controllers --> user.controller.js
    Fowyv_Backend --> dummy
    dummy --> AudioFiles.js
    dummy --> Users.js
    dummy --> UsersChoises.js
    dummy --> UsersDetails.js
    dummy --> UsersSearchSettings.js
    Fowyv_Backend --> middlewares
    middlewares --> authMiddleware.js
    middlewares --> index.js
    middlewares --> signUpMiddleware.js
    middlewares --> userDetailsMiddleware.js
    Fowyv_Backend --> routes
    routes --> auth.routes.js
    routes --> interactions.routes.js
    routes --> search.routes.js
    routes --> user.routes.js
    Fowyv_Backend --> utils
    utils --> dates.js
```

- .gitignore
- app.js
- package-lock.json
- package.json
- start.js
- .vscode
  - launch.json
- config
  - auth.config.js
- controllers
  - auth.controller.js
  - interactions.controller.js
  - search.controller.js
  - user.controller.js
- dummy
  - AudioFiles.js
  - Users.js
  - UsersChoises.js
  - UsersDetails.js
  - UsersSearchSettings.js
- middlewares
  - authMiddleware.js
  - index.js
  - signUpMiddleware.js
  - userDetailsMiddleware.js
- routes
  - auth.routes.js
  - interactions.routes.js
  - search.routes.js
  - user.routes.js
- utils
  - dates.js

A diretoria está estrutura de forma a agregar em cada subdiretoria os componentes que exponham funcionalidades semelhantes, como é o caso dos middlewares ou dos ficheiros com dados estáticos dummy. De salientar que o entry point da aplicação a nível funcional é o ficheiro app.js, mas de forma a poder utilizar ecmascript 6 no desenvolvimento da aplicação, foi definido o ficheiro start.js, onde está definido a utilização da toolchain babel.

## Software

NodeJS e NPM - Necessário para correr a aplicação servidora. <https://nodejs.org/en/>

Android Studio –Permite correr a aplicação móvel react native dentro de um emulador para testar a aplicação - <https://developer.android.com/studio>

## Passos a ter em conta para testar o projeto

1. Instalar os programas referidos acima.
2. Após instalado o Android Studio, criar um AVD (Android Virtual Device) através do Android Studio. <https://developer.android.com/studio/run/managing-avds#createavd>
3. Após se obter as sources do repositório, em cada uma das diretorias correr na linha de comando o comando “npm install” para instalar as dependências necessárias.
4. Na diretoria *Fowyv\_Backend*, executar o comando “node start.js” para correr a aplicação servidora.
5. Na diretoria *Fowyv*, executar o comando “npx react-native run-android” para correr aplicação móvel.