

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/PS1920V_FOWYV

No repositório é possível encontrar as diretorias *Entregas*, referente às entregas feitas no decorrer do projeto no âmbito da cadeira de Projeto e Seminário, *Código Fonte*, onde se encontra o código fonte do projeto e *Ficheiros de instalação*, onde se encontra o ficheiro *APK* para instalar a aplicação móvel, para além de na raiz da diretoria se encontrar o dossier de projeto e o relatório final.

Seguidamente é descrito o conteúdo de cada diretoria.

Entregas

Nesta diretoria é possível encontrar os documentos entregues referentes às diversas fases do projeto: proposta do projeto, relatório de progresso, relatório beta, cartaz e relatório final. Na estrutura abaixo pode-se verificar os ficheiros que se encontram dentro da mesma.

Entregas

- |—Cartaz
 - | Cartaz.pdf
- |
- |—Proposta de projeto
 - | Proposta de projeto Fowyv.pdf
- |
- |—Relatório de progresso
 - | Relatório de progresso.pdf
- |
- |—Relatório e dossier beta
 - Dossier de projeto.pdf
 - Relatorio Beta.pdf

Código fonte

Na diretoria *Código fonte* encontra-se todo o código em que assenta o projeto *Fowyv*. Esta diretoria contém 3 subdiretorias: *Fowyv*, *Fowyv_Backend* e *DB*. A diretoria *Fowyv* refere-se ao

código fonte da aplicação móvel. A diretoria *Fowyv_Backend* contem o código em que assenta a aplicação servidora e por fim a diretoria *DB* contem apenas o script de criação das coleções da base de dados. Também é possível observar o historial de todo o desenvolvimento do projeto a nível de código fonte através do repositório https://github.com/BernardoPoiares/PS_1920v-Fowyv.

Seguidamente é descrito o conteúdo de cada diretoria.

Fowyv

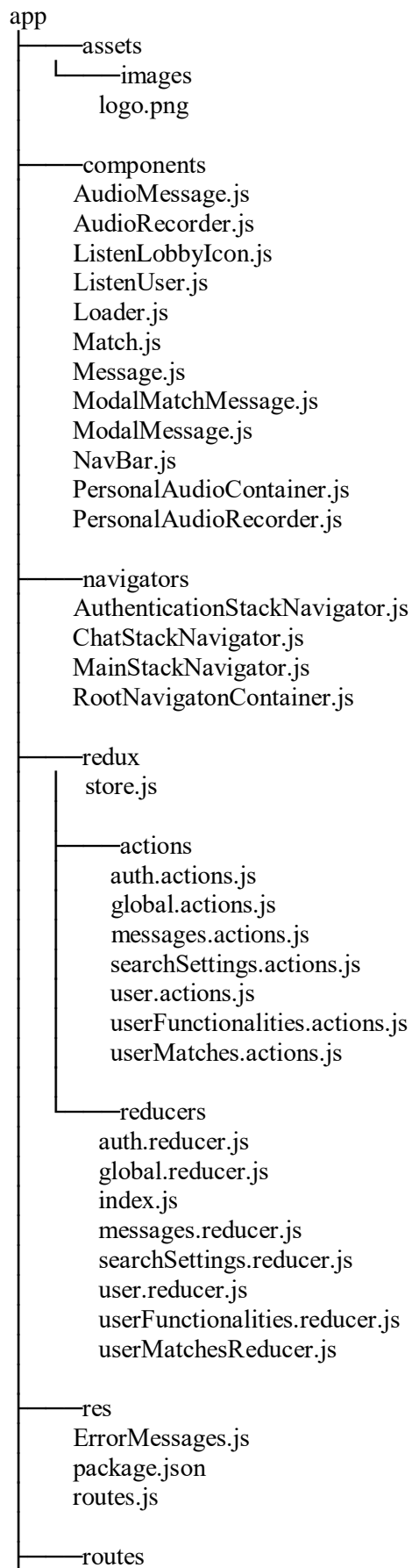
Estrutura

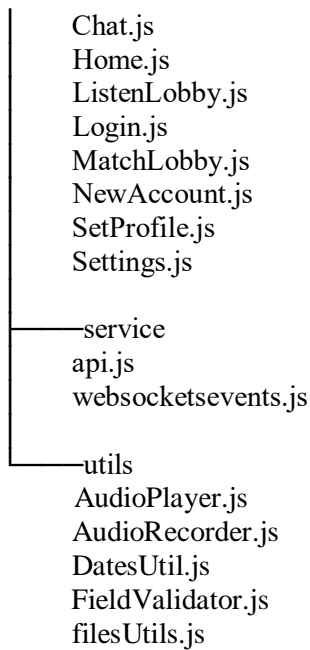
A aplicação encontrasse estruturada de acordo com a seguinte estrutura:

```
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:





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 sitio 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 comum entre os diferentes componentes.

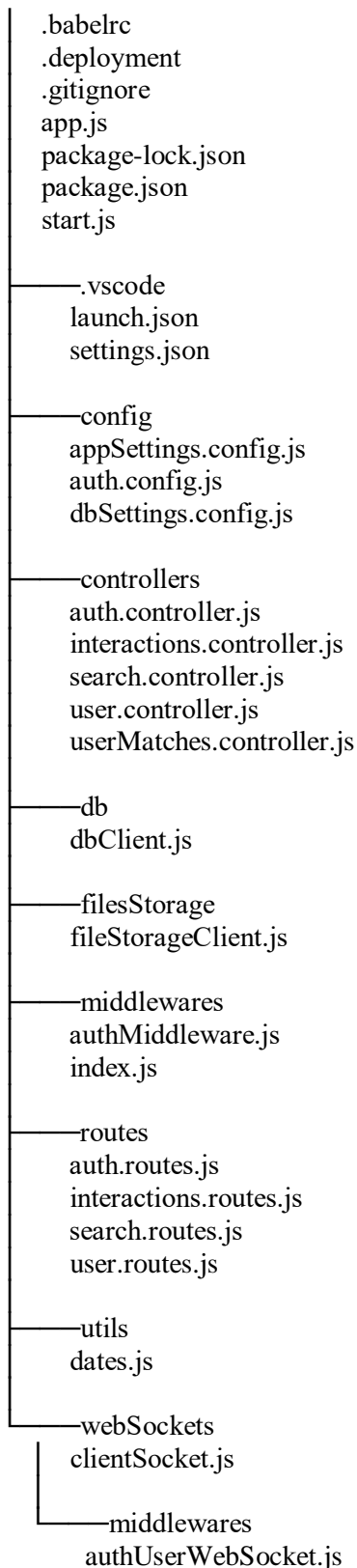
Relativamente à diretoria *service*, a mesma contém os componentes responsáveis pela comunicação com a *api* e com websockets expostos pela aplicação servidora.

Por último encontra-se 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



A diretoria contem na raiz o *entry point* da aplicação a nível funcional que 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*, assim como outros ficheiros

necessários para o desenvolvimento da aplicação, como o *package.json* que indica as dependências da aplicação.

Relativamente às subdiretórias que contêm código fonte da aplicação servidora, a diretoria *config/* contêm as configurações da aplicação, onde se encontra por exemplo a chave utilizada para assinar os *tokens*

A diretoria *controllers/* contém os ficheiros responsáveis por processar os pedidos recebidos pela *Web API*.

Quanto à diretoria *db/*, a mesma contém o ficheiro que permite estabelecer conexões com a base de dados e executar queries.

Seguidamente, a diretoria *filesStorage/* contém o ficheiro onde é feita toda a comunicação com o serviço de alojamento de dados.

Posteriormente, a diretoria *middlewares/* contém os ficheiros que expõem os *middlewares* que pelos quais os pedidos da *Web API* recebidos necessitam de passar, validando os mesmos antes de serem processados pelos *controllers*.

Relativamente à diretoria *routes/* a mesma contém os ficheiros responsáveis por definir os endpoints que são possíveis de comunicar com a *Web API*.

Por fim, é na diretoria *websockets/* que se encontram os ficheiros que contêm a lógica que permite a criar e gerir as conexões websocket.

Software

NodeJS e NPM - Necessário para correr a aplicação servidora localmente. <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 a aplicação móvel através de um simulador localmente

1. Instalar Android Studio.
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, na diretoria *Fowyv* correr na linha de comando o comando “npm install” para instalar as dependências necessárias.
4. Na diretoria *Fowyv*, executar o comando “npx react-native run-android” para correr aplicação móvel.

Passos a ter em conta para testar o projeto num dispositivo móvel Android

1. Fazer download do ficheiro APK disponibilizado no repositório Github na diretoria *Ficheiros de instalação*, ou através do link abaixo.
https://github.com/BernardoPoiars/PS1920V_FOWYV/raw/master/Ficheiros%20de%20instala%C3%A7%C3%A3o/app-release.apk
2. Permitir que seja instalada aplicações de fontes desconhecidas no dispositivo móvel. Para mais informações consultar
<https://developer.android.com/distribute/marketing-tools/alternative-distribution#unknown-sources>

Nota: Se não for possível testar o envio de mensagens ou a aplicação ficar num estado pendente sem resposta, deve-se fechar e voltar a abrir a aplicação. Se mesmo assim não funcionar, deve-se limpar a cache da aplicação através das definições do dispositivo móvel.

Passos a ter em conta para testar a aplicação servidora localmente

1. Instalar o NodeJs acima referido.
2. Após se obter as sources do repositório, na diretoria *Fowyv_Backend* correr na linha de comando o comando “npm install” para instalar as dependências necessárias.
3. Na diretoria *Fowyv*, executar o comando “node start.js” para correr aplicação servidora.

Nota: Para permitir a comunicação entre a aplicação móvel e a aplicação servidora, correndo ambas localmente, é necessário alterar as constantes definidas nos dois ficheiros dentro da diretoria *Fowyv/service/* as constantes *BASE_URL* para o ip local da maquina a testar, com o porto 4000.