Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática



**Relatório de:**

**Trabalho Prático AI2**

Alexandre Maio - 16516

Rafaela Silva - 18527

Martinho Malhão - 18727

Hugo Marques – 18736

Diogo Tojal - 17172

Viseu, 2020/21

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório de Trabalho Prático AI2

Curso de Licenciatura em

Engenharia Informática

Trabalho Prático AI2

Ano Letivo 2020/2021

Alexandre Maio - 16516

Rafaela Silva - 18527

Martinho Malhão - 18727

Hugo Marques – 18736

Diogo Tojal - 17172

Viseu, 2020/21

**Índice**

[1. Introdução 3](#_Toc76677910)

[2. Frontend 4](#_Toc76677911)

[2.1. Login 4](#_Toc76677912)

[2.2. Menu de Navegação 4](#_Toc76677913)

[2.3. Páginas 4](#_Toc76677914)

[2.3.1. Home 4](#_Toc76677915)

[2.3.2. Mapa 4](#_Toc76677916)

[2.3.3. Lotação 4](#_Toc76677917)

[2.3.4. Alertas 4](#_Toc76677918)

[2.3.5. Reports 4](#_Toc76677919)

[2.3.6. Opiniões 4](#_Toc76677920)

[2.3.7. Utilizadores 4](#_Toc76677921)

[2.3.8. Definições 4](#_Toc76677922)

[2.4. React 4](#_Toc76677923)

[3. Backend 5](#_Toc76677924)

[3.1. Sequelize 5](#_Toc76677925)

[3.2. Module 5](#_Toc76677926)

[3.3. Controllers 5](#_Toc76677927)

[4. Conclusão 6](#_Toc76677928)

# Introdução

Neste trabalho temos como objetivo implementar um frontend para os administradores referidos na cadeira de PINT assim como um backend que sustente ambas as partes mobile e administrador do projeto integrado. Para tal usamos o NodeJS para servir o frontend e o backend da aplicação, sendo que no frontend usamos o ReactJS para criar o website e fazer comunicação com o backend. Já o backend está responsável de comunicar com a base de dados criada em PostgreSQL através da ORM SequelizeJS.

# Frontend

## Login e Registo

O nosso frontend começa pela página de login/registo, nesta página é permitido fazer login com uma conta atribuída ao utilizador pela instituição ou criada pelo mesmo, para dar login, figura 1, basta preencher os campos email e password corretamente e clicar no botão “Iniciar sessão”, caso não tenha uma conta é possível “Pedir acesso” e criar uma conta na página de registo, figura 2, aqui é possível introduzir todos os dados necessários para uma conta na aplicação.

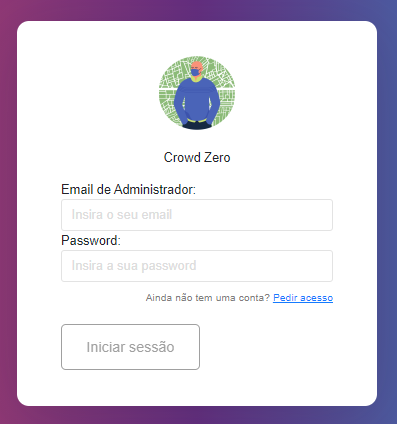


Figura 1 - Login

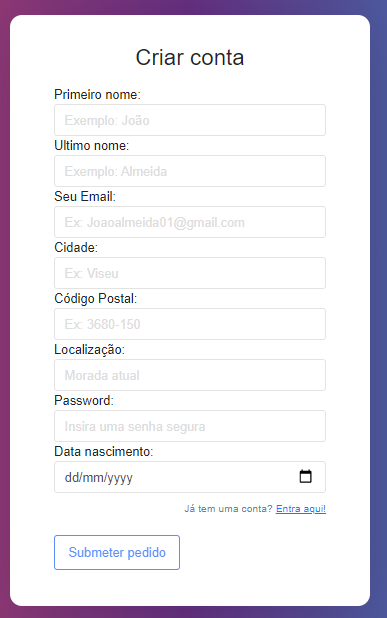


Figura 2 - Registo

De modo a validar os logins usamos uma função, figura 3, para verificar as contas na base de dados, se as credenciais estiverem corretas é carregada a página “admin/home”, caso contrário um pop-up aparece com o devido erro.

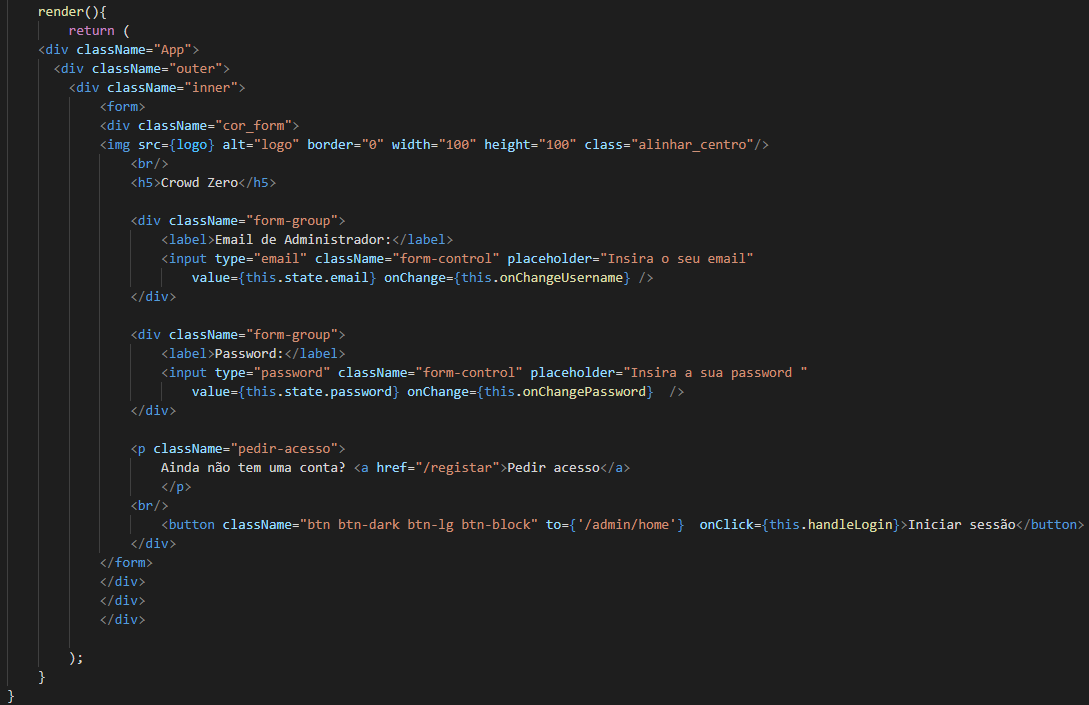


3 - Verificação das Credenciais

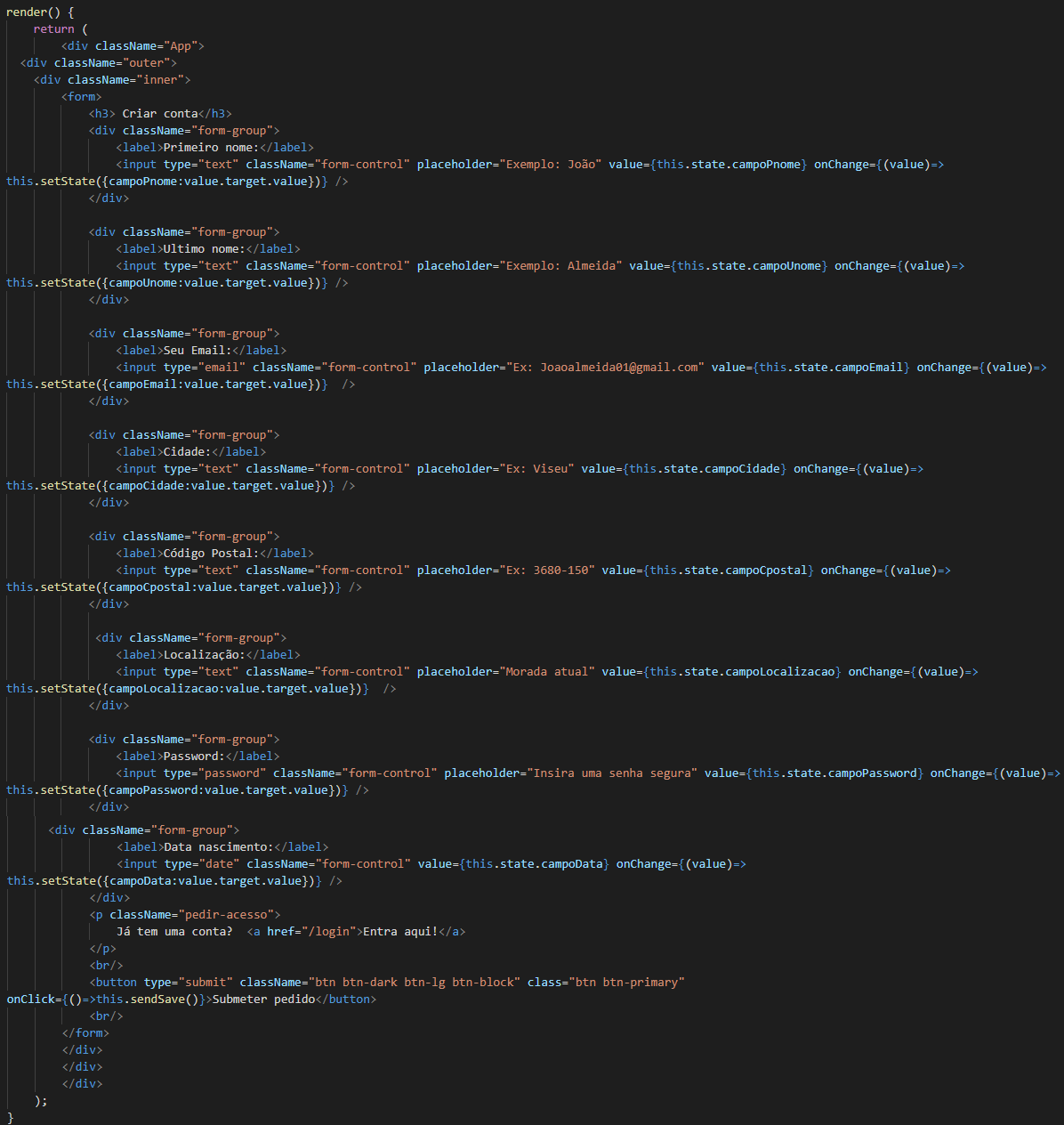
Para registar uma nova conta é necessário preencher todos os campos, ou um erro vai ser emitido através de um pop-up avisando que campo estava em branco, clicando no botão “Submeter pedido” os dados são submetidos e uma conta nova é adicionada na base de dados, figura 4.



4 - Novo Registo



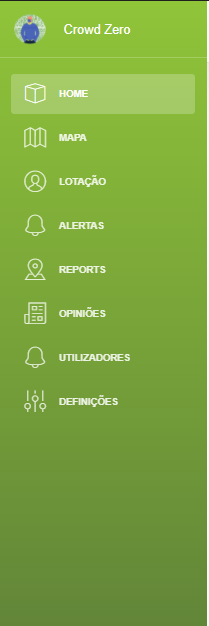
5 - Render “Login”



6 - Render “Registar”

## Menu de Navegação

Para facilitar a navegação no website criamos uma barra de navegação, figura 7, do lado esquerdo que estará sempre presente, esta permite um rápido acesso a maioria das funcionalidades implementadas. Este menu é feito de forma dinâmico, ou seja, se for adicionada uma nova rota esta vai aparecer automaticamente.



7 - Menu de Navegação



8 - Criação dinâmica do Menu de Navegação



9 – Rotas

## Páginas

### Home

A homepage permite um acesso rápido a diversas informações e estatísticas pertinentes aos utilizadores, sendo estas, o número de reportes feitos naquele dia até ao momento, um gráfico com os reportes, e outro com a lotação dos últimos sete dias, uma tabela com novos registos para aceitar ou rejeitar novos utilizadores que se inscreveram na instituição e outro gráfico que mostra a percentagem de utilizadores da instituição em relação a todas as instituições.

### Mapa

Esta página permite, com um mapa e uma lista de locais existentes, procurar de uma forma interativa os locais existentes na aplicação e analisar o estado em que se encontra. O mapa apresenta os locais com um marcador rodeado por um círculo que pode ser de cor verde, amarela ou vermelha representando o número de reportes, as cores correspondem a local pouco, muito ou extremamente populado, respetivamente.

Na tabela que lista os locais é possível procurar pelo nome e ordenar pelo tipo de reportes, novamente pouco, muito ou extremamente populado.

É possível também adicionar um novo local ou editar um local existente diretamente nesta página ou nas definições.

### Lotação

Nesta página é dada a informação necessária para determinar em que locais, indoor e outdoor, é que se deteta o maior número de aglomerados. Temos dois gráficos que mostram a lotação nos locais outdoor e indoor, respetivamente, assim como outra tabela com todos os locais que pode ser filtrada pelo tipo ou total de reportes, e pode ser ordenada de forma ascendente ou descendente.

### Alertas

Na página Alertas é possível editar, criar e eliminar tipos de alertas, assim como, emitir e consultar alertas.

O primeiro gráfico nesta página mostra as desinfeções feitas num período de tempo, por exemplo nos últimos 3 meses ou na última semana, nos locais indoor.

A próxima tabela mostra o tipo de alertas que podem ser emitidos, estes podem ser editados, eliminados ou pode-se criar novos tipos.

Na tabela seguinte é mostrado os últimos alertas e desinfeções feitos com a data(dia/mês/ano), local e tipo de desinfeção, é possível filtrar o tipo de alerta feito assim como a ordem, do mais recente ao mais antigo, ou vice-versa.

### Reports

A página Reports permite consultar diferentes dados referentes aos reportes, alterar os limites de lotação da instituição, assim como, pesquisar e remover os reportes existentes através de uma lista.

O primeiro gráfico mostra as zonas com o maior número de reportes seguido por um número total de reportes na instituição e a instituição com o maior número de reportes. Aqui é também possível alterar os limites de lotação na instituição.

O gráfico seguinte mostra a quantidade de reportes totais por dia da semana.

Por fim temos uma lista de reportes feitos na instituição, estes podem ser removidos e pode-se organizar a lista pela data do reporte de forma ascendente ou descendente.

### Opiniões

Para gerir os comentários basta ir a página Opiniões, onde também existe a classificação da instituição assim como uma classificação média da na última semana.

A primeira tabela lista os comentários feitos quanto aos locais, é também possível excluir comentários a partir desta tabela.

Há outros dois gráficos nesta página, o primeiro mostra as classificações dadas á instituição e o segundo mostra as opiniões médias da semana.

### Utilizadores

Nesta página é possível gerir os utilizadores pertencentes a instituição. No início temos um grupo de estatísticas referentes ao número de utilizadores existentes, ativos, etc., seguido por uma lista de utilizadores da instituição onde é possível pesquisar por um utilizador em especifico ou ordenar pelo data.

### Definições

Por fim, nas definições é possível atualizar os dados da instituição, nome, email, limites de lotação, imagem etc., assim como, gerir os locais indoor, outdoor e a lista de alertas.

Para atualizar os dados da instituição basta preencher os dados a alterar e clicar em “Atualizar Dados”.

Nas listas de locais indoor e outdoor é possível editar a informação dos locais, adicionar um novo, ou remover existentes.

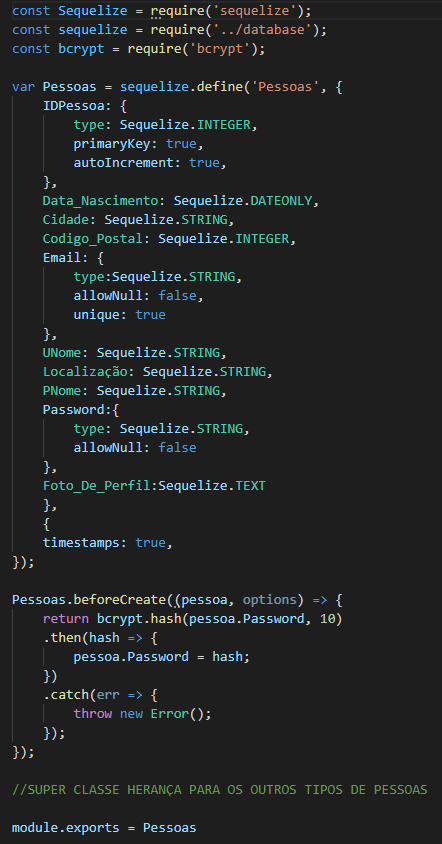
Na lista de alertas é possível editar os detalhes do alerta, removê-lo ou criar um novo diretamente.

# Backend

## Models

### Pessoas:

Neste ficheiro para além da declaração do modelo das Pessoas temos também o que seria equivalente a um trigger em SQL puro. Neste trigger é feito um hash da password inserida de modo a evitar guardar a password em pleno texto na BD.



### Administrador



Possuí as respetivas Foreign Keys associadas: (Id\_Instituição & Id\_Pessoa), sendo que as Instituições podem estar a cargo de vários Admins e cada Admin possuí somente uma instituição. Também, cada Id\_Pessoa pode corresponder a um só Admin, ou seja, um Admin tem somente um Id\_Pessoa.

### Uma imagem com texto Descrição gerada automaticamenteUtilizador da Instituição

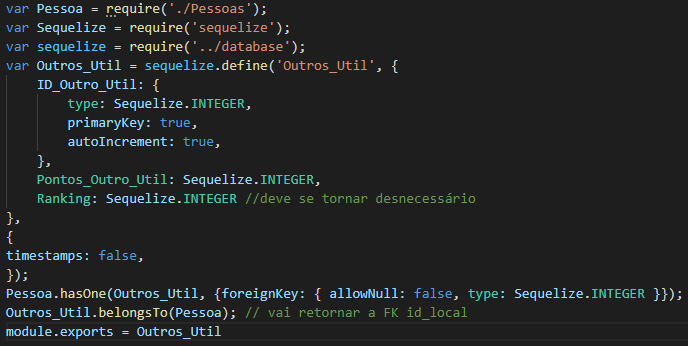
Uma imagem com texto

Descrição gerada automaticamente

Possuí a respetiva Foreign Key associada: (Id\_Pessoa), sendo que cada Id\_Pessoa pode corresponder a um só Utilizador, e um Utilizador tem somente um Id\_Pessoa.

Da relação do Utilizador com a(s) sua(s) Instituição(ões) resulta uma terceira tabela (Util\_pertence\_Inst) pelo tipo de ligação (M-N) sendo que uma Instituição pode possuir muitos utilizadores, e o Utilizador pode ter várias Instituições associadas.

### Outros Utilizadores



Possuí a respetiva Foreign Key associada: (Id\_Pessoa), sendo que cada Id\_Pessoa pode corresponder a um só Outro\_Utilizador, e um Outro\_Utilizador tem somente um Id\_Pessoa.

### Instituição

### Local



Possuí a respetiva Foreign Key associada: (Id\_Instituição), sendo que cada Id\_Local corresponde a uma só Instituição, e uma Instituição pode corresponder a vários Locais.

### local indoor

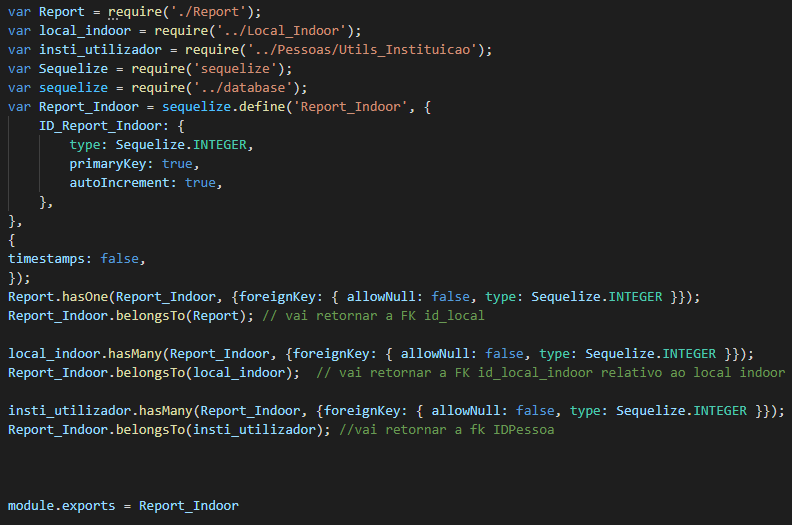


Possuí a respetiva Foreign Key associada: (Id\_Local), sendo que cada Id\_Local\_Indoor corresponde a um só local, e um local possuí vários locais Indoor.

### Reports

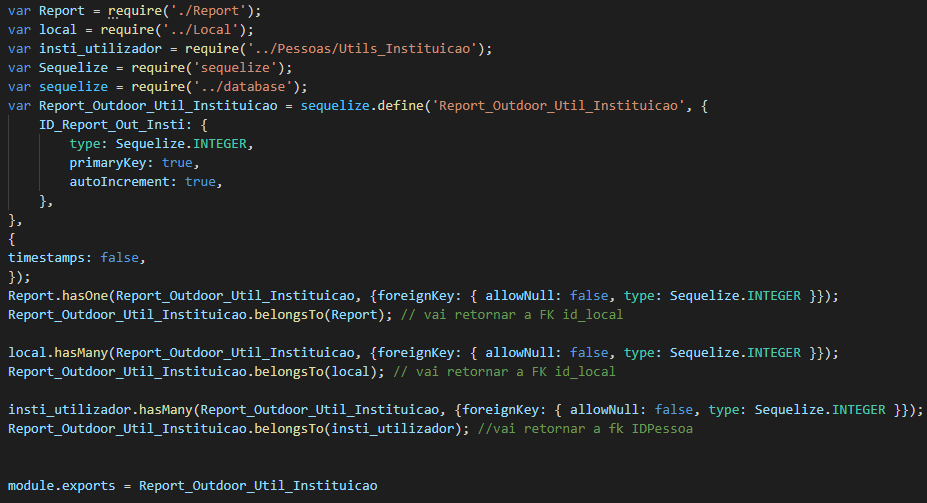
Uma imagem com texto

Descrição gerada automaticamenteReport

Report Indoor

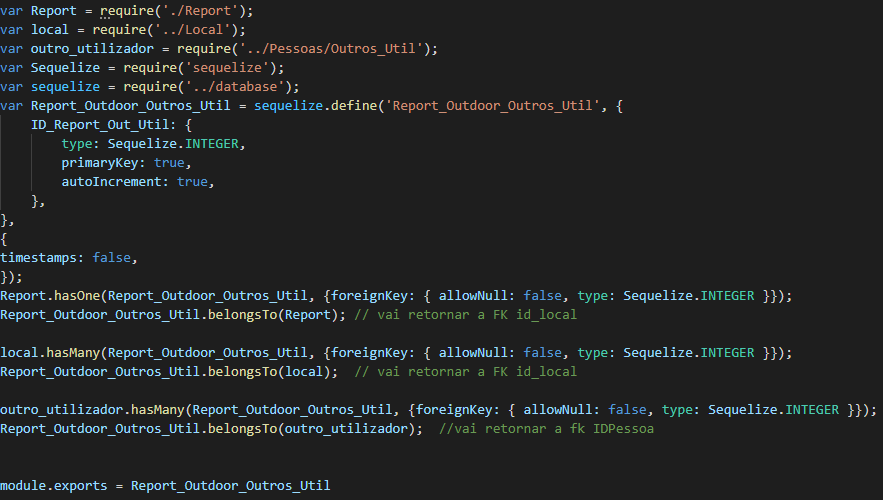
Possuí as respetivas Foreign Keys associadas: (Id\_Util, Id\_localIndoor, Report).

Report outdoor utilizador instituição



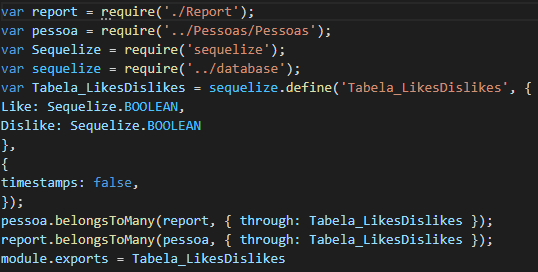
Possuí as respetivas Foreign Keys associadas: (Id\_Util, Id\_local, Report).

Report outdoor outro utilizador



Possuí as respetivas Foreign Keys associadas: (Id\_outroUtil, Id\_local, Report).

### Tabela like-dislike

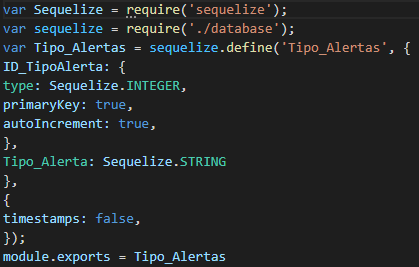


Possuí as respetivas Foreign Keys associadas: (Id\_Pessoa, Id\_Report), sendo que muitas pessoas podem dar muitos like/dislikes, mas cada pessoa só pode dar um like/dislike por parte de report.

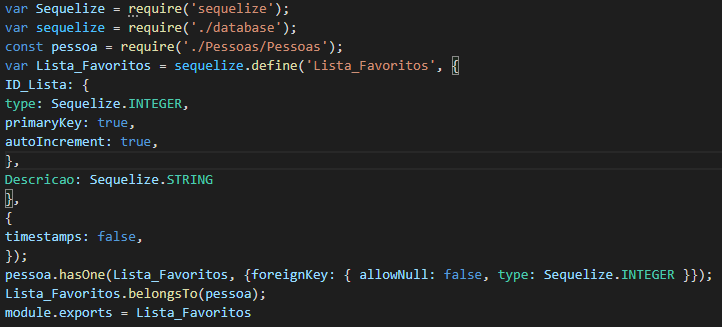
### Alertas



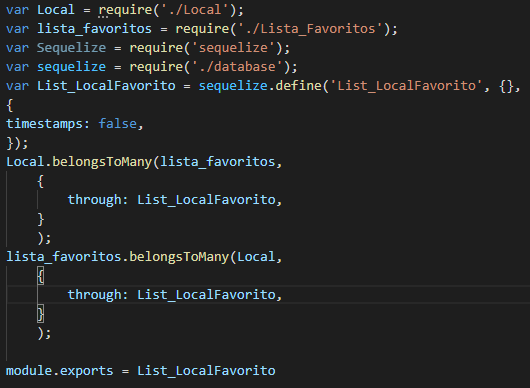
### Tipo alerta



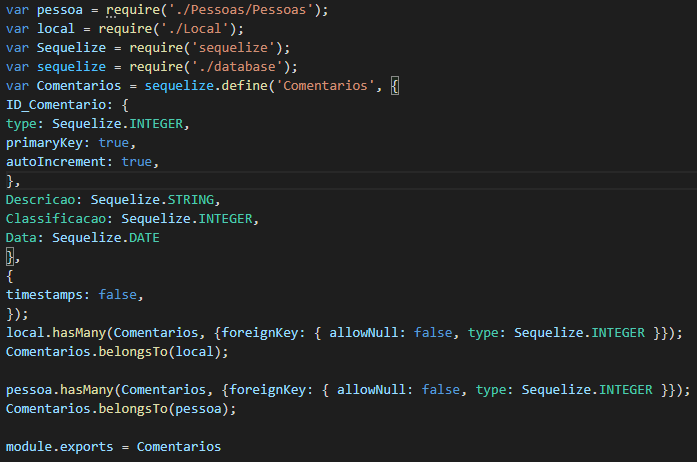
### Lista favoritos



### Lista local favoritos



### Comentários



### Base de dados



## Controllers

Para cada um dos principais models (Utilizadores, Reports, Instituições,etc.) temos um ficheiro com as funções que vão fazer a conexão com a base de dados.

### Controllers relacionados aos utilizadores

Todos os creates feitos neste ficheiro criam uma Pessoa nova primeiro e só depois criam o tipo específico de utilizador. Por isso, quando houver um erro na criação de um utilizador, irá ficar uma pessoa “fantasma” registada na BD. Para resolver isso simplesmente, visto que recorremos ao uso do try/catch, verificamos se a pessoa foi criada e caso tenha sido é apagada.

Uma imagem com texto

Descrição gerada automaticamente

Figura 10 Apagar pessoa "fantasma" em caso de erro

Uma das funções mais usadas nos controladores relacionados às pessoas/utilizadores normais da aplicação é a função que atualiza o ranking de toda a gente. Para tal, utilizamos a função do sequelize que permite fazer uma raw query diretamente em PostgreSQL:

Uma imagem com texto

Descrição gerada automaticamente

Figura 11 Função Atualizar Ranking Utilizadores

### Controllers relacionados aos reportes

Este é o ficheiro com as funções mais importantes para o funcionamento da aplicação.

Uma destas é a função que devolve a densidade média do local. Esta recebe pelos parâmetros do pedido o id do local e no corpo recebe o intervalo de tempo para calcular

Na pesquisa dos reports fazemos a inclusão do modelo dos Reports geral de modo a podermos selecionar consoante a data, enquanto que o id do local é guardado na tabela do report especifico. Esta densidade não tem em conta os reportes feitos nos locais indoor.

Muitas das funções que fazem a seleção de reports usam esta mesma maneira de obter os reports num intervalo de tempo.

Uma imagem com texto

Descrição gerada automaticamente

Figura 12 Densidade média de local

Uma imagem com texto

Descrição gerada automaticamente

Figura 13 Cálculo da densidade média do local

Outra função que seleciona os reports fá-lo também num intervalo de tempo, mas desta vez devolvendo os dados separados por dias. Esta função apenas devolve o numero de reports feitos, respondendo com um array.

Uma imagem com texto

Descrição gerada automaticamente

Figura 14 Número de reports feitos nos últimos X dias

### Controllers relacionados aos locais

# Conclusão

Com este projeto, ficamos a saber que o Javascript oferece uma versatilidade enorme, podendo ser usado como backend em servidores para comunicar com bases de dados e fazer operações CRUD, usando ORMs tipo o SequelizeJS assim como servir websites usandos bibliotecas como o ReactJS e os vários componentes existentes na plataforma NPM.