

Multimédia e Tecnologia Web

Relatório Trabalho Prático

Trabalho realizado por:

José Gomes 14829

Pedro Carvalho 15708

José Carreira 15709

Índice

Índice de figuras	2
Estado da Arte - State of the art.....	3
Implementação.....	5
Arquitetura	11
Serviços.....	12
Tecnologias Utilizadas.....	12

Índice de figuras

Figura 1 –Gráfico temporal da evolução do vírus	4
Figura 2 - Gráfico circular indicando numero de registos por país	4
Figura 3 - Gráfico de barras com os números atuais registados	4
Figura 4 - Exemplo de registos inseridos na Firebase	5
Figura 5 - Formulário de criação de um novo utilizador	6
Figura 6 - Formulário de Login	7
Figura 7 - Opção de editar uma submissão	8
Figura 8 - Formulário de atualização de uma submissão	9

Estado da Arte - State of the art

O produto final desenvolvido pelo grupo permite a um utilizador consultar um conjunto de gráficos detalhados que fornecem informação sobre o COVID 19.

Estes gráficos fornecem informação como:

- Gráfico temporal da evolução de casos.
- Gráfico de barras com os números atuais de casos, divididos pelo seu estado
- Gráfico circular indicando o numero de casos por país.
- Gráfico de barras com o número de casos atuais, divididos por sexo
- Numero de óbitos por faixa etária.
- Informação adicional relativa ao numero atual de submissões registadas na base de dados, ultimo pais a submeter informação e taxa de mortalidade.

Com estes gráficos é possível verificar a evolução do vírus através de submissões realizadas por utilizadores registados. Os gráficos também possuem interatividade para ser possível uma análise mais aprofundada.

Para submeter um registo é necessário fazer Login na aplicação, caso o utilizador não tenha conta, existe opção de criar uma.

Cada utilizador pode realizar uma submissão particular, ou pode submeter uma organização, que corresponde a um conjunto de utilizadores. Com os dados submetidos, o utilizador tem a opção de os atualizar a qualquer altura, tendo ao seu dispor uma lista com todos as submissões que este realizou.

Evolução do Virus

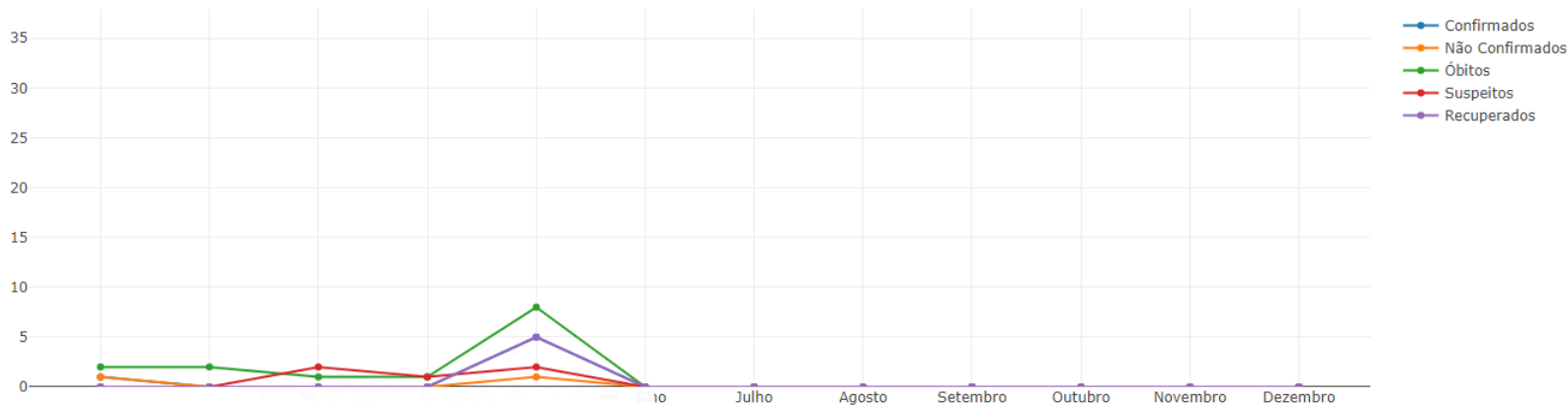


Figura 1 –Gráfico temporal da evolução do vírus

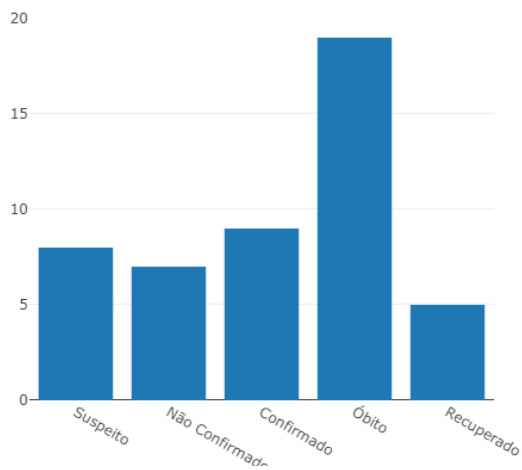


Figura 3 - Gráfico de barras com os números atuais registados

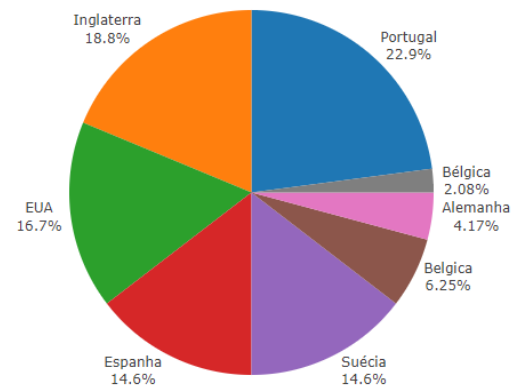


Figura 2 - Gráfico circular indicando numero de registos por país

As imagens acima representadas são exemplos de gráficos utilizados no nosso trabalho.

Implementação

Base de dados - A base de dados escolhida foi Firebase. Uma vantagem que constatamos com o uso de Firebase foi a sua rapidez de implementação, permitindo ao grupo focar no desenvolvimento do restante backend sem perder tempo “desnecessário”, o facto da mesma ser não relacional significou que mudanças no conteúdo da base de dados não causaram dificuldades, no entanto, uma desvantagem enorme da Firebase é o facto da mesma não deixar realizar ‘queries’ com dois valores de ordenação, isto provou ser um problema na gestão de dados, o que nos levou a fazer o processamento de informação manualmente.

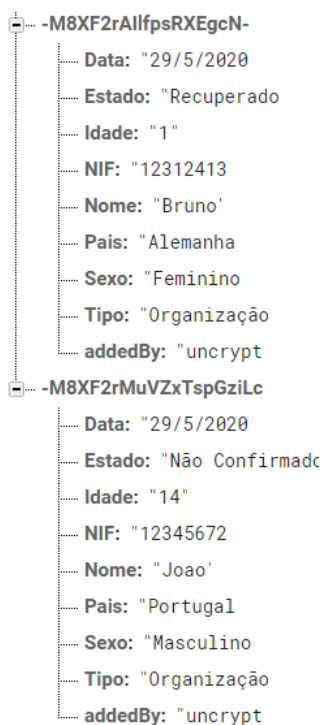
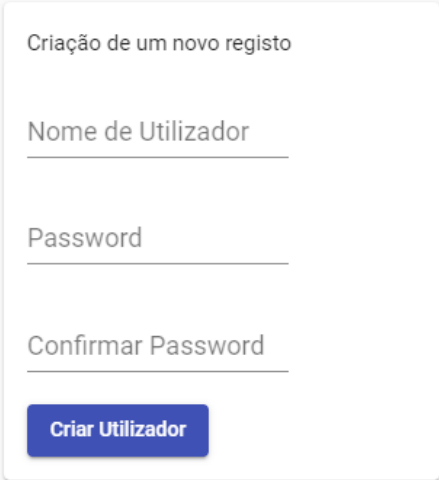


Figura 4 - Exemplo de registos inseridos na Firebase

Signup – O processo de signup é auxiliado por um serviço, este serviço é chamado quando o utilizador selecciona o botão ‘criar utilizador’, o serviço utiliza o nome colocado no form para fazer um pedido POST a uma API externa (desenvolvida pelo grupo) que faz uma ligação á Firebase, o primeiro passo é verificar se o username já existe na base de dados, caso exista vai ser devolvido uma mensagem de erro informando que esse username já está em uso , caso o username inserido pelo o utilizador seja “novo” é feita a inserção na base de dados, sendo depois possível realizar login.



Criação de um novo registo

Nome de Utilizador

Password

Confirmar Password

Criar Utilizador

Figura 5 - Formulário de criação de um novo utilizador

Login - O processo de login é auxiliado por um serviço, este serviço é chamado quando o utilizador selecciona o botão de login, e utilizando o username e a password submetidos no formulários, faz um pedido GET a uma API externa (desenvolvida pelo grupo) que faz uma ligação á Firebase, de seguida utilizando a função *compare* da biblioteca *bcrypt* faz a comparação entre a palavra-passe inserida pelo utilizador, e a *hash* guardada na base de dados, caso a função determine que o hash é correspondente à palavra passe, é devolvido ao serviço um valor de *true*, caso a palavra passe não corresponda á hash é devolvido o valor *false*. O serviço devolve este valor ao componente de Login, onde é de seguida informado se o Login teve sucesso ou não. Caso o Login seja bem-sucedido, o username e a palavra passe são guardados em *localStorage* para uso futuro.

Obs: Quando um user realiza o login é disponibilizado uma nova opção na sidebar que permite fazer logout, apagando a *localStorage*.

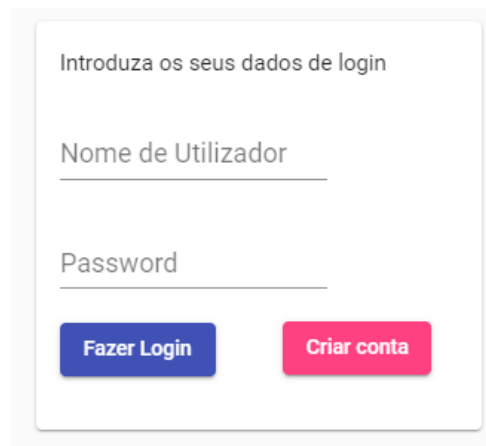
A imagem mostra um formulário de login com o título "Introduza os seus dados de login". Há dois campos de entrada: "Nome de Utilizador" e "Password". Abaixo dos campos, há dois botões: "Fazer Login" (azul) e "Criar conta" (rosa).

Figura 6 - Formulário de Login

Atualização de registos – Com os logins implementados, achamos relevante registar todas as submissões feitas por um utilizador, para que seja possível a

atualização de registos. A cada submissão de dados que realizamos é acrescentado o campo *addedby* na base de dados, este campo corresponde ao utilizador que adicionou o registo. O componente *submissions.component* vai buscar todas as submissões do utilizador e coloca-as num array 'usersarray' e o ID de cada uma no array 'id', garantindo que a cada ID corresponde uma submissão do 'userarray', de seguida, utilizando um *NgFor* é feita a renderização de toda a informação correspondente a uma submissão e a opção de a editar.

A opção de editar está ligada a um novo componente *edit-user.component* que recebe o ID da submissão a ser alterada através do routing, e utilizando a Firebase obtemos os dados respetivos a esse ID. Com estes dados preenchemos os campos de um novo form, permitindo ao utilizador alterar os mesmos. A submissão deste form, selecionando o botão 'Alterar Registo' faz o *update* dos campos na base de dados.

Bruno

Estado

Óbito

Idade

86

País

Espanha

Sexo

Masculino

[Editar](#)



Figura 7 - Opção de editar uma submissão

Introduza o seu NIF

1235123

Máximo 10 caracteres

☐

Suspeito

☐

Não Confirmado

☐

Confirmado

☒

Óbito

☐

Recuperado

País de Residencia:

Espanha

Idade:

86

☒ Masculino

☐ Feminino

Alterar Registo

Figura 8 - Formulário de atualização de uma submissão

Submissão de um formulário - Na submissão de um formulário o utilizador deve preencher os campos apresentados no formulário, o campo NIF deve ter 8 caracteres para ser considerado válido. Quando o utilizador submete os dados é chamado um serviço que utiliza o NIF preenchido no formulário e faz um pedido GET para a nossa API, este pedido tem como finalidade determinar se o NIF se encontra já inserido num registo da base de dados, caso esteja, vamos despoletar um alerta na página indicando para o utilizador rever os dados inseridos. Caso o NIF seja único vamos realizar a inserção de uma nova submissão.

Obs: As informações mencionadas acima para a submissão de um formulário também se aplicam para a submissão de uma lista de formulários (organização).

Dashboard - A dashboard é o componente que faz o display dos gráficos e informação relevante. O código relevante ao processamento de dados é executado dentro do construtor, a Firebase possui um método 'valuechanges' que verifica regularmente alterações na base de dados, todo o código que faz o processamento de valores encontra-se dentro deste método.

Arquitetura

SideNav – Em todos os momentos da aplicação existe uma sidenav lateral que permite trocar entre os diferentes componentes, esta sidenav é um componente em si e está responsável pelo routing da aplicação.

Routing list – Foram criadas diversas routes para tornar o uso da plataforma o mais fluido possível:

- {path: 'form', component: UserformComponent} – Responsável por apresentar o formulário de submissão de um caso.
- {path: 'login', component: LoginComponent} – Responsável por apresentar o menu de login.
- {path: 'dashboard', component: DashboardComponent} – Responsável por apresentar o menu com todos os gráficos e dados processados.
- {path: 'submissions', component: SubmissionsComponent} – Route responsável por apresentar todas as submissões enviadas pelo utilizador.
- {path: 'edituser/:id', component: EditUserComponent } – Route responsável por apresentar o formulário para alteração de dados respetivos a uma submissão, recebe como parâmetro o ID do user.

Serviços

Dataservice- O serviço DataService é constituído por funções que servem como meio intermédio entre os componentes e a API. Este serviço tem o objetivo de receber a resposta da API e fazer o encaminhamento da resposta para o componente.

API- O API é utilizado para fazer a comunicação com a base de dados com o objetivo de validar dados e devolver informação. A API implementada faz a verificação do login, validação no caso da criação de um novo utilizador e validação do numero de NIF.

Tecnologias Utilizadas

Firestore- Utilizado para fazer a disponibilização e armazenamento de dados

Bcrypt – Utilizado para fazer a encriptação/Desencriptação da palavra-passe do utilizador.

Node/Express- Utilizado para hospedar a nossa API.

PlotlyJs – Utilizado para gerar diversos gráficos com a informação fornecida.