

# M E D I C

Tecnologias de Programação Web

102491

Raquel Paradinha

103234

Paulo Pinto

104142

Tiago Carvalho

## Índice

|                                  |   |
|----------------------------------|---|
| Links e Utilizadores Disponíveis | 3 |
| Introdução Ao Conceito           | 4 |
| Estrutura do Trabalho            | 5 |
| Funcionalidades da Aplicação     | 6 |
| Base de Dados                    | 8 |
| Deploy dos Ambientes             | 8 |
| Referências                      | 9 |

## Links e Utilizadores Disponíveis

O trabalho desenvolvido encontra-se totalmente disponível e funcional nos seguintes links:

1. Client-Side - <https://medictpw.web.app>
2. Server-Side - <http://tiagosora.pythonanywhere.com>

Basta aceder à aplicação no client-side e efetuar o login para começar a explorar. Para fins de utilização preparámos alguns utilizadores com diferentes tipos de permissões:

1. Paciente            Utilizador Comum
  - a. email            -        mandogalo@gmail.com
  - b. password       -        patientpass
2. Doutor             Utilizador Autorizado
  - a. email            -        cristina-santos@medic.com
  - b. password       -        doctorpass
3. Diretor             Utilizador Autorizado
  - a. email            -        fmendes@medic.com
  - b. password       -        deanpass

As informações sobre os utilizadores e o deploy serão explicadas de seguida.

## Introdução Ao Conceito

A nossa aplicação baseia-se num site de suporte a uma clínica de saúde. Para tal, desenhamos algumas funcionalidades que se adequam ao tema e que vão de encontro aos objetivos propostos no guião do projeto. Desenvolvemos 3 tipos diferentes de utilizador, com permissões igualmente diferentes que correspondem aos use cases que planeamos para a aplicação.

Para a criação deste projeto, usamos maioritariamente as tecnologias aprofundadas em contexto de aula, criando dois ambientes totalmente separados, intuitivos e de fácil acesso.

Um destes é focado no armazenamento, tratamento e pesquisa de dados, server-side; e outro no utilizador e na apresentação dos dados armazenados, client-side. Para o primeiro, utilizamos Django REST Framework (DRF), com o qual criámos elementos que enriquecem o back-end da nossa aplicação, por exemplo métodos de autenticação, validação, pesquisa, routing, entre outros. Para a view da nossa aplicação web, como lecionado na cadeira e solicitado pelo professor, utilizamos Angular Framework, tecnologia baseada em TypeScript (TS), com o qual desenvolvemos todas as interfaces de utilizador que podem ser exploradas.

Além disso, aproveitamos o nosso conhecimento em HTML, CSS e Django para criar a camada de apresentação da nossa aplicação. Para melhor tratar das migrações, existe um ficheiro Bash, de nome bashrun.sh, na root do projeto, criado com esse objetivo.

## Estrutura do Trabalho

Como dito anteriormente, a aplicação é separada em dois ambientes que comunicam entre si através de métodos de HTTP Request e Response.

Para o back-end da nossa aplicação utilizamos várias tecnologias que surgem do uso de Django REST Framework (DRF). Indo mais ao pormenor, utilizámos a estrutura base de um projeto Django e acrescentamos mecanismos que satisfizeram as nossas necessidades. Para o tratamento dos requests e especificação das routes utilizámos ViewSets, como permissões específicas desenvolvidas da mesma forma que as Django Base Permissions. Criamos 3 tipos diferentes de permissões que caracterizam os nossos 3 tipos de utilizador (Paciente, Doutor, Diretor). Criámos 7 tipos diferentes de modelos e para qualquer interação com qualquer objecto do sistema (pesquisar, criar, alterar, eliminar) é primeiro feita uma validação dos dados da interação através do uso de Serializers, que para além disso ainda facilitam o acesso aos objetos armazenados. Uma lista contendo os endpoints da API pode ser vista mais abaixo.

Para o front-end da nossa aplicação usamos Angular Framework, criando um total de 18 componentes, muitos deles encapsulados dentro de outros componentes, que são invocados a partir de chamadas para qualquer das routes (RouterModule). De forma a salvar o estado atual da utilização recorremos a EventEmitters, com os quais guardamos a autenticação e autorização do utilizador atual. Para facilitar a troca de dados com a API, desenvolvemos web services que invocam qualquer um dos base models e tratam de efetuar o HTTP Request.

Em termos de componentes Angular, em todas as páginas existem 3 grandes componentes, a barra de navegação, o componente principal da página e o rodapé. A barra de navegação faculta o acesso aos routes disponíveis para cada utilizador. As páginas da aplicação serão explicadas nas funcionalidades do sistema.

## Funcionalidades da Aplicação

As funcionalidades podem ser facilmente explicadas se forem tidos em causa os nossos tipos de utilizador.

### Utilizador Não Autenticado

Quando o utilizador visita a aplicação pela primeira vez, e até se registar, encontra-se não autenticado. O utilizador tem acesso a navegar pela página principal (Home), assim como pelas outras páginas de teor mais informativo (About e Contact). O utilizador poderá ainda, ver os botões de Login e SignUp na barra de navegação de cada uma das páginas.

O botão de Login dá a possibilidade de o utilizador se autenticar com uma conta existente no sistema. No entanto, caso o utilizador não possua nenhuma conta na aplicação, poderá ainda recorrer ao SignUp para se registar no sistema. Para um registo adequado, o utilizador deverá fornecer um nome, uma conta de email (que não coincida com nenhuma outra no sistema) e uma palavra-passe. Assim que fizer signup com sucesso, o utilizador é redirecionado para a página de Login, onde se poderá autenticar

### Paciente

Caso o utilizador se autentique com uma conta correspondente a um paciente no sistema, passará a ter permissões de paciente. Este tem acesso às mesmas funcionalidades do utilizador não autenticado e a outras exclusivas dos pacientes.

Duas dessas funcionalidades estão relacionadas com as marcações (appointments) que todos os pacientes podem fazer. Para realizar uma marcação, o paciente deve, primeiramente, utilizar o botão "New Appointment" visível no canto superior esquerdo do ecrã. De seguida, é direcionado para uma nova página onde poderá escolher um dos departamentos médicos disponíveis no sistema, uma data para a marcação e ainda poderá submeter uma pequena mensagem descritiva da marcação. Para finalizar esta interação, o paciente deve apenas submeter a marcação, através do botão "Add Appointment".

O paciente pode ainda aceder, através da barra de navegação, à página "My Appointments", onde poderá ver todas as suas marcações. Caso o utilizador não tenha feito até à data qualquer marcação, a tabela desta página não terá qualquer linha. Caso já tenham sido feitas marcações, o paciente poderá ver essas marcações, assim como modificá-las ou removê-las.

O paciente pode também consultar as suas prescrições. Prescrições estas que só podem ser criadas por um utilizador com permissões de doutor.

Relativamente à sua conta, o paciente pode ainda aceder ao seu Profile e também fazer Logout. Na página de perfil, o paciente pode verificar todos os dados que submeteu durante o registo (SignUp), à exceção da palavra-passe (para fins de privacidade). O utilizador pode personalizar essas informações.

## **Doutor**

Caso o utilizador se autentique com uma conta correspondente a um doutor no sistema, passará a ter permissões de doutor. Este mais uma vez também tem acesso a todas as funcionalidades do utilizador não autenticado.

Para além dessas funcionalidades, também pode aceder à página My Appointments, onde consegue ver as suas marcações para cada um dos departamentos a que pertence. Para isso deve primeiro escolher um dos departamentos na select box. Caso o departamento do doutor não tenha até à data qualquer marcação, a tabela não terá qualquer linha.

O doutor pode ainda aceder à página de prescrições em My Prescriptions, onde pode ver as prescrições que já realizou, assim como fazer uma nova prescrição, clicando em Add Prescription. Para tal basta submeter um id de um paciente (exemplo: 1) e uma mensagem relativa à prescrição.

Relativamente à sua conta, o doutor pode ainda aceder ao seu Profile e também fazer Logout. Na página de perfil, o doutor pode verificar todos os dados que submeteu durante o registo (SignUp), à exceção da palavra-passe (para fins de privacidade). Ao contrário do paciente, o doutor só tem acesso a alterar a sua password.

## **Diretor**

O diretor (dean) é uma entidade criada para fins administrativos. Este pode gerir a clínica e as entidades/objetos funcionais nele (departamentos e doutores).

O diretor tem acesso a basicamente todas as funcionalidades do sistema, à exceção da secção de administração da API. Em relação aos restantes utilizadores, este tipo de usuário tem acesso a novos componentes, acedidos através da barra de navegação.

Um deles, Departments, apresenta a lista de departamentos da clínica e outro, Doctors, apresenta a lista de doutores de cada departamento. O diretor tem acesso a gerir cada um destes elementos como bem lhe apetecer.

## Base de Dados

Para o armazenamento dos dados, foi feito o recurso a uma base de dados SQLite, diretamente ligada ao Django, que facilita a recolha e o fornecimento de dados.

Para a criação da base de dados, recorreu-se a migrações de tabelas/modelos desenvolvidos em Django e executadas pelo terminal com recurso a um ficheiro Bash, criado para esse mesmo efeito.

## Deploy dos Ambientes

Ambos os ambientes (server e client) encontram-se atualmente online. Para o server-side utilizamos PythonAnywhere, assim como aconselhado pelo professor e para o client-side utilizamos Firebase, pois trata-se de um sistema de hospedagem que oferece inúmeras vantagens/funcionalidades para quem pretende dar deploy da sua Angular Application.



## Referências

Recorremos aos materiais disponibilizados pelo professor na página da cadeira, assim como sites com a documentação das tecnologias usadas:

- <https://www.django-rest-framework.org/>
- <https://angular.io/docs>
- <https://material.angular.io/>