Plataforma de apoio a centro de testes covid-19

Programação em Ambiente Web

Grupo inserir\_numero

Diogo Costa 8170455 | Luis Marques | Luis Teixeira

Índice

[1 Introdução 4](#_Toc42887840)

[1.1 Contextualização do Documento 4](#_Toc42887841)

[1.2 Apresentação do caso de estudo 4](#_Toc42887842)

[2 Objetivo 5](#_Toc42887843)

[2.1 Ferramentas e tecnologias utilizadas 5](#_Toc42887844)

[3 Definições, Acrónimos e Abreviaturas 6](#_Toc42887845)

[4 Visão geral do projeto 7](#_Toc42887846)

[4.1 Objetivo do Software 7](#_Toc42887847)

[5 Implementação da API REST 8](#_Toc42887848)

[5.1 Definição 8](#_Toc42887849)

[5.2 Definição de tabelas da base de dados 8](#_Toc42887850)

[5.3 Operações CRUD 8](#_Toc42887851)

[5.3.1 Users 9](#_Toc42887852)

[6 FrontEnd (Angular) 11](#_Toc42887853)

[6.1 Introdução 11](#_Toc42887854)

[6.2 Estrutura 11](#_Toc42887855)

[6.2.1 Services 11](#_Toc42887856)

[6.2.2 Models 12](#_Toc42887857)

[6.2.3 Components 12](#_Toc42887858)

[7 Conclusão 13](#_Toc42887859)

[8 GitLab 14](#_Toc42887860)

[9 Bibliografia 15](#_Toc42887861)

Índice de Figuras

|  |  |  |
| --- | --- | --- |
| Date | Description | Version |
| 02/05/2020 | * Contextualização do Documento * Apresentação do Caso de Estudo * Definições, Acrónimos e Abreviaturas | 1.0 |
|  |  | 1.1 |
|  |  | 1.2 |
|  |  | 1.3 |

Histórico Alterações Documento

# Introdução

## Contextualização do Documento

Este documento descreve o processo de implementação do projeto desenvolvido no âmbito da unidade curricular ‘Programação em Ambiente Web’, e fornece uma visão geral sobre o produto final.

## Apresentação do caso de estudo

Devido ao panorama atual e à crescente necessidade de dar resposta a um elevado número de pedidos de testes de despiste à COVID.19, é necessário montar um novo centro de análises regional para a realização dos testes. Este documento descreve todo trabalho realizado, durante o desenvolvimento de uma plataforma web, que dará suporte ao novo centro de análises.

# Objetivo

O trabalho descrito neste documento, tem como objetivo o desenvolvimento de uma plataforma web, para processamento de pedidos de teste de diagnóstico, agendamento de testes e registo de resultados de um centro de testes despiste e imunização à Covid-19. Para isso, será necessário ir de encontro a uma série de requisitos, que serão descritos numa secção posterior deste documento.

## Ferramentas e tecnologias utilizadas

No processo de desenvolvimento do projeto, recorremos às seguintes ferramentas:

* NodeJS e a framework ExpressJS
* Postman
* MongoDB
* Angular
* VSCode
* Git e Github/Gitlab

# Definições, Acrónimos e Abreviaturas

REST - Representational State Transfer

API - Application Programming Interface

# Visão geral do projeto

## Objetivo do Software

A aplicação web desenvolvida neste trabalho, foi idealizada para agilizar o processo de pedidos de teste de diagnóstico, agendamento de testes e registo do histórico de cada paciente testado no centro de análises. No pedido de teste de diagnóstico, o utilizador deverá indicar se foi encaminhado pela linha Saúde24, se pertence a um grupo de risco ou se trabalha em locais de risco. Após a realização do teste, a ficha do utilizador será alterada para ‘teste realizado’. Quando obtidos os resultados, será possível registar o resultado clínico na ficha do pedido do utilizador anexando um ficheiro (pdf) com os resultados clínicos e adicionando o resultado ao pedido. Por defeito, todos os pacientes estarão classificados como ‘suspeito’, e consoante os resultados de testes forem inseridos no histórico do paciente, a classificação será atualizada.

# Implementação da API REST

## Definição

API (Application Program Interface): é um conjunto de funções e procedimentos que permitem a criação de aplicações que acedem a recursos ou dados de um sistema, aplicação ou outro serviço REST API: definição de apis baseadas no protocolo HTTP universal.

Os métodos HTTP mais usados em serviços REST, são GET, POST, PUT, DELETE e que predefinem o CRUD em HTTP.

Partindo do que foi descrito anteriormente elaboramos então o CRUD da nossa API para responder às nossas necessidades.

## Definição de tabelas da base de dados

A primeira abordagem foi delinear as tabelas que necessitávamos para responder ao nosso problema.

Para tal, utilizamos a ferramenta de base de dados abordada nas aulas, nomeadamente o MongoDB.

Possuímos duas tabelas na nossa base de dados. A tabela “users” constituída pelos campos: xxxxxxxxx.

A tabela “testes” constituída pelos campos:

Prints

Prints

prints

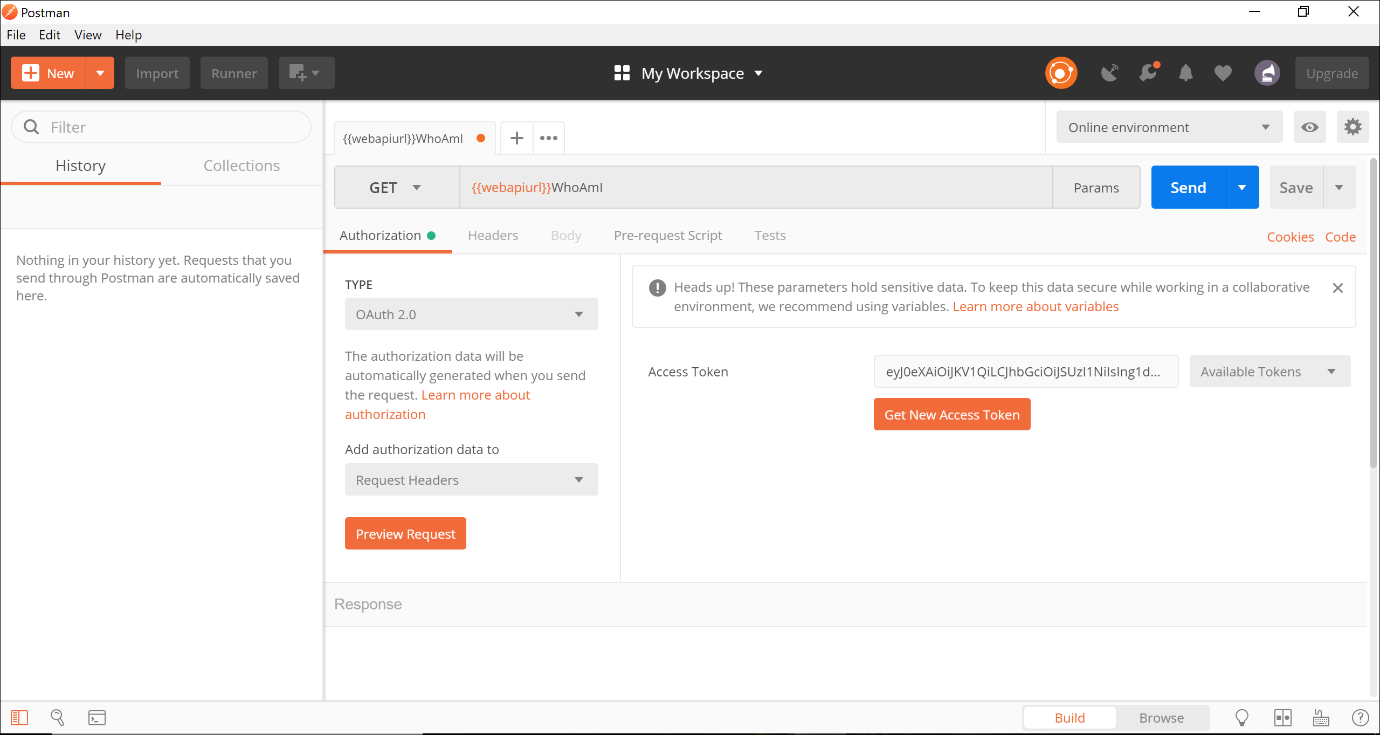
Estas tabelas foram implementadas na pasta “models” sendo essa pasta por padrão onde ficam implementadas as tabelas.

## Operações CRUD

Depois de já definidos os “models” elaboramos as operações CRUD ( Create, Remove ,Update and Delete) dos “Users” e “Tests”.

Todas estas operações foram inseridas na pasta “controllers” onde se encontram todas as ações do controlador.

Posteriormente, para testar os nossos pedidos REST utilizamos o Postaman sendo uma ferramenta bastante vantajosa, que permite:

* Definir parâmetros na área Hearder e Body dos pedidos HTTP;
* Verificar as respostas do servidor ;
* Criar projetos para teste e validação de APIs.

### Users

#### Registar Utilizador (POST)

Neste pedido é feito o registo do utilizador, ou seja, após este pedido POST o utilizador é inserido na base de dados.

Print

Print

Print

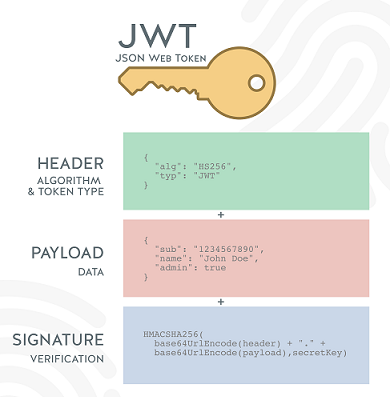
#### Listar Utilizador (GET)

Após o pedido POST é possível listar os utilizadores já existentes na base de dados. Se não tiver qualquer utilizador adicionado, este responde com uma lista vazia.

Print

Print

#### Authenticate (POST)

Para este pedido utilizamos uma ferramenta lecionada nas sessões desta unidade curricular nomeadamente o JWT.

JSON Web Token (JWT) é um open standard (RFC 7519) que define um método compacto e autocontido para transmitir com segurança informações entre as partes num objeto JSON.

Quando os tokens são assinados usando pares de chaves pública/privada, a assinatura também certifica que a parte que é proprietária da chave privada é aquela que a assinou.

# FrontEnd (Angular)

## Introdução

Para a parte do cliente utilizamos a Framework Angular.Angular é uma plataforma e framework para construção da interface de aplicações usando HTML, CSS e TypeScrit, criada pelos desenvolvedores da Google.

Dentre os principais, podemos destacar os componentes, templates, pastas, roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas, como a execução de testes unitários de uma aplicação.

## Estrutura

### Services

Para dar resposta aos nossos pedidos HTTP feitos ao servidor criamos serviços em Angular nomeadamente:

* **Serviço para os Users (users.service.ts)**

Serviço para responder aos pedidos READ, UPDATE e DELETE dos utilizadores

* **Serviço para os Testes (tests.service.ts)**

Serviço para responder aos pedidos CREATE, READ, UPDATE e DELETE dos testes

* **Serviço para Autenticação (auth.service.ts)**

Serviço para responder aos pedidos CREATE, READ dos utilizadores nomeadamente o registar e fazer login do utilizador.

* **Serviço para o Token (token-interceptor.service.ts)**

Serviço para o token pois precisamos de colocar o esquema de autenticação HTTP: Bearer Authentication.

### Models

* **Model para o User**

Classe com todos os campos do utilizador e seus respetivos tipos.

* **Model para o Teste**

Classe com todos os campos do teste e seus respetivos tipos.

### Components

Componentes que contêm um ficheiro TypeScript responsável por toda.

# Conclusão

Ao realizar este trabalho foi nos permitido consolidar toda a matéria lecionada nesta unidade curricular.

Cumprimos todos os objetivos inicialmente delineados sendo que implementamos todos os requisitos obrigatórios e requisitos de bonificação, nomeadamente, o uso de Live Data e View Models, a implementação de testes à aplicação (ad hoc), o uso webservices adicionais e os serviços do firebase para autenticação.

# GitLab

<https://gitlab.com/8170262/SeniorApp>

# Bibliografia

* <https://developer.android.com/training/basics/fragments/communicating>
* <https://developer.android.com/training/data-storage/room>
* <https://codelabs.developers.google.com/codelabs/android-persistence/#3>
* <https://square.github.io/retrofit/>
* <https://developer.android.com/guide/topics/sensors/sensors_overview>
* <https://developer.android.com/guide/components/services>
* <https://developer.android.com/guide/components/processes-and-threads.html#Threads>
* <https://firebase.google.com/docs?authuser=0>
* <https://firebase.google.com/docs/auth?authuser=0>