Plataforma de apoio a centro de testes covid-19

Programação em Ambiente Web

Grupo inserir\_numero

Diogo Costa 8170455 | Luis Marques | Luis teixeira

Índice

[Histórico Alterações Documento 3](#_Toc40482927)

[1. Introdução 4](#_Toc40482928)

[1.1 Contextualização do Documento 4](#_Toc40482929)

[1.2 Apresentação do caso de estudo 4](#_Toc40482930)

[1.3 Objetivo 4](#_Toc40482931)

[1.3 Ferramentas e tecnologias utilizadas 4](#_Toc40482932)

[2. Definições, Acrónimos e Abreviaturas 5](#_Toc40482933)

[3. Visão geral do projeto 5](#_Toc40482934)

[3.1 Objetivo do Software 5](#_Toc40482935)

[4. Implementação da API REST 6](#_Toc40482936)

[4.1 Definição 6](#_Toc40482937)

[4.2 Definição de tabelas da base de dados 6](#_Toc40482938)

[4.3 Operações CRUD 7](#_Toc40482939)

[4.3.1 Users 8](#_Toc40482940)

[5. Conclusão 8](#_Toc40482941)

[6. GitLab 9](#_Toc40482942)

[7. Bibliografia 10](#_Toc40482943)

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 todo o projeto desenvolvidono âmbito da unidade curricular de Programação em Ambiente Web.

## 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.

## 1.3 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

3.1 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

## 4.1 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.

## 4.2 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 na aula, nomeadamente o MongoDB.

Possuímos duas tabelas na nossa base de dados.

A tabela “users” constituída pelos campos: xxxxxxxxx.

A tabela “testes” constituída pelos campos:

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

Prints

Prints

prints

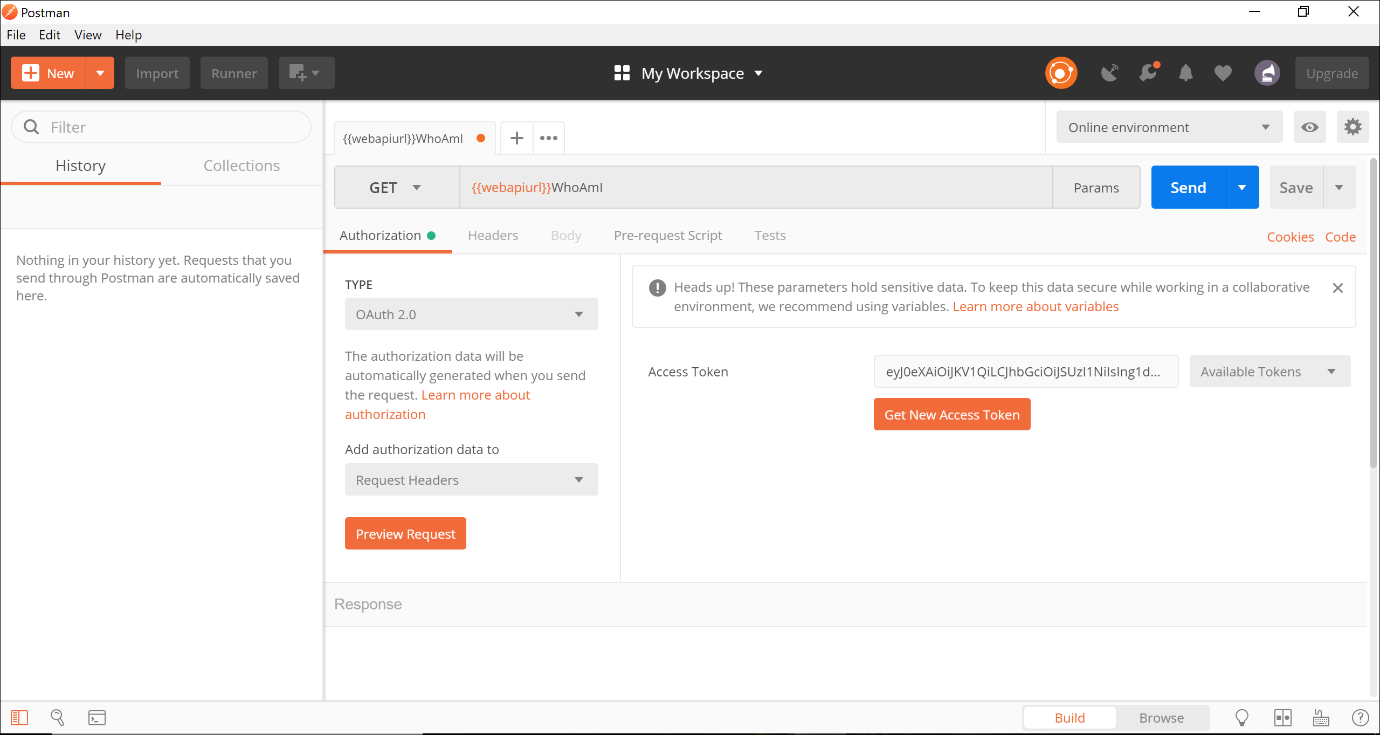
## 4.3 Operações CRUD

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.



### 4.3.1 Users

#### 4.3.1.1 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

#### 4.3.1.2 Listar Utilizador (GET)

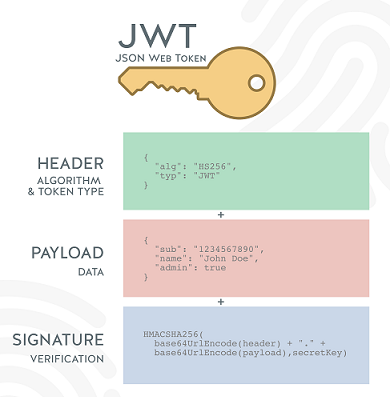
Após o pedido POST é possível listar os utilizadores já existentes na base de dados. Se não tiver qualquer utilizador adicionado, este responde com uma lista vazia.

Print

Print

#### 4.3.1.3 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

……………………………………………………………………………………………………………………

# 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>