



Relatório do 2º Trabalho Prático

Sistema Nacional de Vacinação

Sistemas Distribuídos

```
----- MENU -----  
Que operação deseja realizar?  
  
1: Consultar centros de vacinação  
2: Registar para vacinação  
3: Ver notificações  
4: Reagendar vacina  
Outros: sair
```

Realizado por:

Pedro Grilo (43012)
Diogo Castanho (42496)

Ano Letivo 2020/2021

1 Introdução

Neste trabalho foi-nos proposto o desenvolvimento de um sistema que interligasse as funcionalidades de três módulos distintos, de modo a termos um sistema de vacinação pronto a utilizar. Esses módulos são os seguintes:

- Módulo dos Utentes
- Módulo dos Centros de Vacinação
- Módulo da DGS

2 Arquitetura

A arquitetura do nosso trabalho consiste numa arquitetura REST, sendo que foram usadas as seguintes ferramentas:

- PostgreSQL - para o armazenamento persistente pedido
- Spring Boot- para o back end
- Aplicações Python - para as interfaces de utentes, centros e dgs

2.1 Execução da aplicação

- Spring Boot - estar na diretoria root do projeto e executar o comando: **gradle build** seguido de **./gradlew bootrun**
- Ter a base de dados criada com as configurações que estão no ficheiro `application.properties`, de modo a que seja possível fazer a conexão
- Correr as aplicações `cliente.py`, `centros.py` e `dgs.py` no terminal, para executar as operações de cada um deles

3 Base de Dados

A comunicação entre o back end(spring) e a nossa base de dados é feita através das queries fornecidas pelo JPA, usando algumas nativas, criando também algumas nossas de acordo com algumas definições que eram necessárias.

A nossa base de dados é constituída pelas seguintes tabelas:

- **centro vacinacao** - idCentro(pk), nomeCentro e vacinasDiarias
- **fila vacinacao** - idUtente (pk), idCentro e dataVacina
- **lista vacinados** - idUtente (pk), idCentro, dataVacinacao e nomeVacina
- **notificacao utente** - idUtente (pk) e notificacao
- **utentes registados** - idUtente (pk), nomeUtente, email e dataPreferida
- **vacinas disponiveis** - (idCentro, dataVacina)(pk) e dosesVacina

4 Backend

Para a realização do trabalho usamos apenas uma aplicação spring boot. Usamos o mesmo endereço para lidar com todos os requests de todos os módulos que temos.

Temos 3 controladores, 1 para cada módulo, para lidar com os pedidos que são feitos na sua aplicação.

Têm como principal objetivo receber as informações das aplicações no terminal, de maneira a que seja possível consultar, guardar ou eliminar valores da nossa base de dados, retornando os valores que especificamos de uma determinada operação.

Os endpoints que temos são os seguintes:

CentroController

- **/filasDisponiveis - GET**
- **/centroDisponiveis - GET**
- **/centro/id- GET**
- **/centro - POST**

DGSController

- /dgs/estatisticaVacinados - GET
- /dgs/doses - POST
- /dgs- GET
- /dgs/doses/idCentro/data - GET

UtentesControler

- /utente/filaVacinacao - POST
- /utenteRegistado/idUtente - GET
- /utente/filaVacinacao/idCentro/dataVacina - GET
- /utente/filaVacinacao/removeFila/idUtente - GET
- /utente/listaVacinados - POST
- /utente/notificacao/idUtente/email - PUT
- /utente/notificacao/remove/idUtente - DELETE
- /registraUtente - POST

Para os endpoint (*Get*) que são acessados pelas aplicações, o back end faz o acesso à query da operação que se pede, retornando esse valor em formato json para a aplicação. O backend pode também fazer algum trabalho em escolher alguns dos valores que vão ser passados, dependendo da operação.

Para os endpoint(*Post*), a aplicação envia para o backend os valores que deseja introduzir na base de dados, que após fazer a adição, retorna uma resposta à aplicação.

Quando é *delete*, funciona com o post mas em vez de adicionar, elimina um valor específico da base de dados. Pode retornar também informação.

Para *put* faz uma actualização de um tipo de dados específico que recebe. Pode apenas mudar alguns parâmetros, dependendo da operação de backend que tivermos.

5 Aplicações (“front end”)

As aplicações comunicam com o back end através de chamadas dos endpoints dados acima.

Cada uma delas tem apenas as suas funcionalidades, não interferindo umas com as outras.

Mediante a escolha que se fizer, será executado no back end a operação, que lhe retornará algum valor.

Nesta parte também se fazem algumas decisões e verificações de modo a funcionar tudo corretamente.

6 Conclusão

A nosso ver, a realização deste trabalho teve uma dificuldade elevada, tendo várias vezes algumas dificuldades em perceber o que fazer.

Apesar de termos conseguido implementar quase tudo, não conseguimos executar a redundância pedida pelo professor, nem a criação da web app, que dariam uma valorização superior. Posto isto, achamos que mesmo assim o trabalho entregue realiza a maioria das funcionalidades pedidas, sendo que com a sua realização conseguimos melhorar os nossos conhecimentos em sistemas distribuídos, com serviços REST e Spring, como também a melhor percepção das queries de uma base de dados.