Universidade de Évora

Relatório do Trabalho Prático
Sistemas Distribuídos

Isabel Nunes (nº43091) Ludgero Teixeira (nº41348)

Docente: Professor José Saias

Abril de 2021

1 Introdução

O 2º trabalho desta disciplina consiste na implementação de um Sistema de nacional de vacinação cujo sistema distribuído inclui os módulos:

- **Aplicação do cidadão:**

  O utente pode escolher um centro de vacinação e proceder ao auto-agendamento.

  - Consulta de centros de vacinação.
  - Auto-agendamento para vacinação num dos centros, indicando o nome, idade, e-mail e data preferida do utente.
  - Confirmação do agendamento, ou da impossibilidade para aquele dia/centro.

- **Centro de vacinação:**

  Módulo executado por cada centro municipal.

  - Registar o seu funcionamento junto da DGS, com indicação do nº máximo de vacinas que pode administrar por dia.
  - Registar a realização de uma vacinação já agendada, com determinado ćodigo X, ficando registada a data e tipo de vacina.
  - Comunicar uma lista de vacinações concretizadas à DGS.
  - Obter da DGS informação sobre o número de vacinas que receberá para uma data em particular e em função desse número, confirmar ou cancelar agendamentos previstos para esse dia.

- **Módulo central:**

  Referente á DGS, para operações de âmbito nacional e coordenação dos centros.

  - Manter lista de centros e respectiva capacidade diária.

  - Inserindo um nº nacional de vacinas para uma data, verificar os agendamentos para esse dia e respectivas idades, e distribuir as vacinas pelos centros para abranger as pessoas mais velhas.
  - Listar nº total de vacinados por tipo de vacina, e por dia, de acordo com dados recebidos dos centros.

2 RESTful Web Service com Spring

Representational State Transfer (REST) é uma arquitetura que define um conjunto de restrições que são usadas para a criação de web services.

Os Web services que estão em conformidade com a arquitetura REST, denominados por Web services RESTful, fornecem a capacidade de um sistema agir e comunicar com outros, sendo que permitem que os sistemas solicitantes tenham acesso e manipulem representações textuais de recursos Web, usando um conjunto uniforme e predefinido de operações.

Ao usar a framework Spring a nossa aplicação ficou alojada num servidor Tomcat embutido nesta.

No frontend da aplicação utlizamos a template engine Thymeleaf bem como a framework Bootstrap.

Para a implementação das aplicações utlizamos a framework Spring Boot.

Para a compilação das aplicações é utlizado Gradle.

2.1 Spring MVC

As nossas aplicações web estão divididas em 3 partes:

- **Model:** onde estão incluídos os objetos java que guardam os dados provenientes das bases de dados.
- **View:** o contéudo com qual o utlizador interage, sendo estas renderizadas no servidor.
- **Controller:** responsável pela comunicação entre a view e o model, recebem os pedidos e reencaminha-os para o endpoint correcto e respondendo aos mesmos.

3 Procedimento das Operações

3.1 Cidadão

A aplicação do cidadão comunica principalmente com o serviço DGS para as suas operações.

3.1.1 Consulta de centros de vacinação

Em cada pedido de consulta é feito um pedido GET ao serviço da DGS. Esse pedido é realizado serviço da DGS que envia um objeto com a lista dos centros de vacinação que são depois mostrados.

3.1.2 Auto-agendamento para vacinação

Após o preenchimento do formulário para o agendamento, é realizado um pedido PUT ao serviço DGS. Este último procura por algum agendamento já realizado (com o mesmo ID do utente), atualiza-o ou cria um novo num dos seus repositório e envia novamente ao utente. Por último é mostrado o código X, que corresponde ao ID gerado ao guardar no repositório.

3.1.3 Notificações

A Web App do Cidadão começa com um formulário para introduzir o ID de utente, este é o que vai distinguir os utentes e as suas notificações. As notificações estão guardadas num repositório da aplicação e são encontradas (se houver), a seguir de guardar o ID do utente autenticado. As notificações são mostradas no final da página seguinte.

3.2 Centro de Vacinação

3.2.1 Registar o seu funcionamento junto da DGS

Para o bom funcionamento da aplicacão é primeiro preciso fazer esta operação. Após o preenchimento do formulário, é guardado o nome do centro autenticado e mandado o objeto representante dos centros à DGS de forma de um pedido PUT. O serviço DGS guarda ou atualiza por último o centro.

3.2.2 Registar a realização de uma vacinação

Todos os registo de vacina estão guardados na aplicação até este os mandar à DGS. Desta forma, esta operação apenas guarda um registo no seu repositório com os dados recolhidos.

3.2.3 Informar a DGS em relação ás vacinações realizadas

Nesta operação é apenas enviado um pedido POST à DGS com um objeto que contém uma lista de registos realizados. A DGS irá percorrer os registos e separá-los por tipo de vacina, para mais tarde os poder listar.

3.2.4 Obter informação da DGS

A seguir da execução da operação da DGS de distribuir vacinas, em cada centro é executado esta operação que corresponde a obter da DGS todos os agendamentos e o número de pessoas a serem vacinadas para o seu centro destinados para o dia definido. Obtém estes dados a partir de dois pedidos GET. De seguida, ordena dos agendamentos por idade e em função do número

obtido pela DGS, envia um email e notificação via POST ao utente a confirmar ou a cancelar o agendamento.

3.3 DGS

3.3.1 Manter lista de centros e respectiva capacidade diária

A DGS contém um repositório com todos os centros de vacinação e a sua capacidade diária.

3.3.2 Distribuição de vacinas

Para a distribuição de vacinas foi preciso ordenar os agendamentos para o dia definido por ordem maior a menor idade, de seguida até ao final da lista e a até o número definido para as vacinas é marcado o agendamento como confirmado caso não tenha chegado à capacidade máxima do centro. Sempre que confirmado é guardado o novo número de vacinas a serem administradas no centro.

3.3.3 Listar nºtotal de vacinados

Nesta operação é listada o número de todos os tipos de vacina por data.

4 Persistência dos dados

Para a persistência de dados, as 3 aplicações utilizam bases de dados Postgres especificas, de maneira a facilitar as interacções com as bases de dados, sendo que todas as operações são feitas através do CrudRepository (interface Spring Data), que fornece vários métodos para usar e interagir com a base de dados.

5 Comandos para execução

Para a execução deste trabalho é necessário ter uma instância de uma base de dados Postgres para cada uma das aplicações.

As configurações da base de dados, como o nome da mesma, utlizador e palavra-passe, podem ser encontradas no ficheiro *application.properties*.

Foi criado um ficheiro *make* no intuito de facilitar a compilação e execução das aplicações .

As aplicações são compiladas e executadas utlizando os seguintes comandos:

- make dgs
- make utente
- make center

Estes 3 comando vão realizar os comandos *gradle clean*, *gradle build* e *gradle bootRun* em cada uma das respetivas aplicações.

Cada um destes comandos deve ser executado num terminal diferente.

Todas as operações deste programa foram testadas num ambiente *UNIX Linux Ubuntu* e em *MacOs*.

6 Conclusão

Em suma sentimos que este trabalho foi bastante importante, na medida que podemos meter em prática os conceitos leccionados nas componentes teóricas e práticas desta unidade curricular, mas também pelo facto de nos ter permitido trabalhar com várias tecnologias web.
Este trabalho foi bastante desafiante pois foi dada uma grande liberdade de implementação e onde tivemos a oportunidade de trabalhar com várias tecnologias novas.
As componentes base como as componentes adicionais do trabalho foram todas totalmente implementadas com sucesso.

Apesar de terem existido algumas dúvidas e "road blocks"ao longo do desenvolvimento do trabalho, facilmente foram esclarecidas e ultrapassados com alguma pesquisa e com o apoio do conteúdo disponibilizado na plataforma moodle da unidade curricular, sendo que sentimos que o mesmo foi concluído com sucesso.