



Paradigmas de Sistemas Distribuídos

2020/2021

Grupo 7

Mestrado Integrado em Engenharia Informática



José Pinto

A84590

Eduardo Costa

A85735

José Rodrigues

A85501

Ricardo Carvalho

A84261

Pedro Gomes

A84220

Introdução

Serve o presente relatório para explicitar as várias fases de desenvolvimento e diferentes componentes do trabalho realizado no âmbito da Unidade Curricular de Paradigmas de Sistemas Distribuídos, cujo objetivo é construir um protótipo de uma plataforma para suporte de rastreio de contactos e deteção de concentração de pessoas relativo à pandemia de COVID-19, utilizando *Java*, *Erlang*, *ZeroMQ* e *Dropwizard*.

Para tal, desenvolvemos uma aplicação cliente, um servidor *frontend*, diversos servidores distritais e um diretório, tomando partido da programação por atores, *sockets* ZeroMQ e da *framework* *Dropwizard*, de forma a fornecer aos clientes várias formas de interação com o sistema.

Conceção da Solução

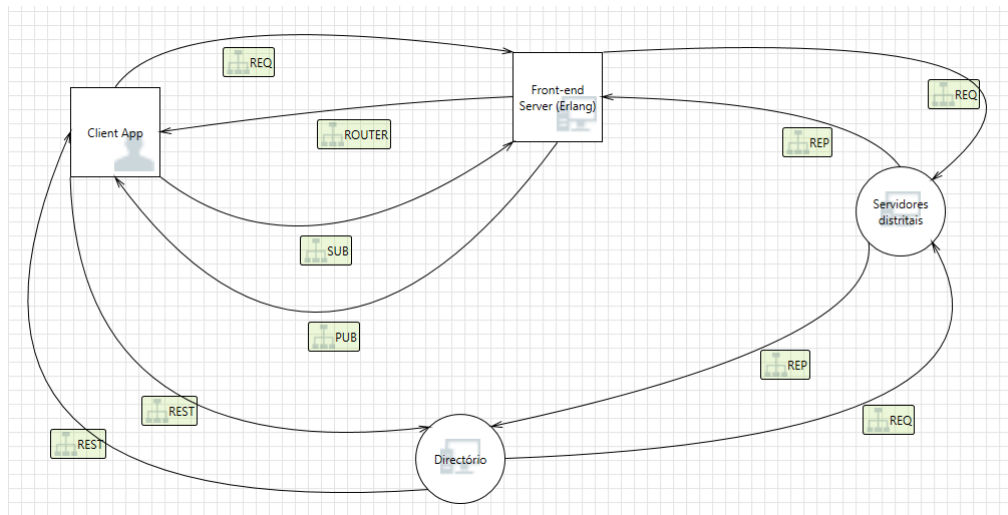


Figura 1 - Modelo da Solução

Aplicação cliente

Aplicação com uma interface de texto que permite ao cliente explorar todas as funcionalidades do sistema. Esta inicia dois *sockets* ZeroMQ, um do tipo REQ e o outro SUB. É através do *socket* REQ que todos os pedidos são feitos ao *frontend*, enquanto o *socket* SUB permite receber todas as notificações públicas existentes. Esta função é composta por uma *main* onde são inicializados os *sockets* previamente referidos e são mostradas as opções iniciais de *login* e registo. Conforme a opção selecionada, são invocadas as respetivas funções, fazendo os pedidos correspondentes ao *frontend* e, de seguida, bloqueiam à espera da sua resposta.

Caso haja uma autenticação bem-sucedida, o utilizador passa a receber as notificações no caso de contacto com um infetado e a função menu é invocada. Esta, por sua vez, é onde todos os pedidos de interação com o servidor são executados e respostas aos mesmos processadas. É também possível, através da interface, interagir com o diretório fazendo *HTTP requests*, de forma a obter os resultados de cada uma das *queries* suportadas.

Servidor *Frontend*

O servidor *frontend* é composto por dois ficheiros, *zmqServer* e *login_manager*. Este funciona como elo de ligação entre clientes e respetivos servidores distritais, tendo um *socket* ROUTER que lhe permite lidar com múltiplos clientes e um *socket* REQ para cada um dos distritos existentes.

Após a inicialização dos *sockets* necessários, o servidor entra na função *loop*, que é recursivamente chamada e é responsável pela receção das mensagens provenientes do cliente. Sempre que chega uma mensagem, é invocada a função *responde_usr*, onde o pedido do cliente é devidamente processado. Caso se trate de um pedido relacionado com registo ou autenticação de contas, recorremos às funções desenvolvidas no ficheiro *login_manager*, onde se encontra toda a lógica de criação, manutenção e utilização das contas de utilizador. Caso o pedido tenha sido efetuado por um cliente que já se encontre devidamente autenticado, é então invocada a função *menu*. Por sua vez, a função *menu* funciona como um espelho do seu homónimo na aplicação cliente. Enquanto de um lado os pedidos são realizados, aqui são processados e respondidos.

O processamento destes pedidos envolve, no entanto, efetuar pedidos aos servidores distritais onde se encontram armazenadas todas as informações, tirando as credenciais dos clientes (que se encontram no servidor *frontend*). Ou seja, nesta função é feita a receção e interpretação de pedidos de clientes, o reencaminhamento desses pedidos aos respetivos servidores distritais, a espera pela resposta dos mesmos e devolução da resposta obtida aos clientes.

É importante ainda referir que é o servidor *frontend* o responsável por publicar eventos como notificações públicas ou notificações de contacto com um infetado. A lógica da publicação das notificações parte do princípio que, para existir qualquer notificação, tem de haver primeiro um determinado pedido de um qualquer utilizador. Ou seja, quando, por exemplo, um cliente avisa que está infetado, o servidor *frontend* sabe que, para além de processar o seu pedido, terá de notificar os utilizadores que contactaram com ele. O mesmo acontece no caso das notificações públicas e quando é enviado um pedido de mudança de localização.

No servidor *frontend* é também monitorizado o número de subscrições feitas por um cliente (limitadas a três no enunciado) e a permissão, ou não, de um cliente usar a aplicação por se encontrar infetado.

Servidor Distrital

O servidor distrital é composto por três classes: *ServidoresInitializer*, *LigacaoDistrito* e *ServidorDistrital*. A *ServidoresInitializer* inicia os servidores de todos os distritos. A classe *LigacaoDistrito* representa a ligação do servidor *frontend* a um servidor distrital. Nesta classe encontra-se um ciclo para a receção de pedidos, com um *switch case* com todos os tipos de pedidos diferentes que podem surgir. Finalmente, *ServidorDistrital* é a classe com toda a lógica de funcionamento dos servidores distritais.

Cada servidor distrital, como se demonstra na figura abaixo, possui uma matriz representativa do seu mapa de localizações onde, em cada posição, contem os IDs dos utilizadores que lá se encontram num determinado momento. Tem, também, um *Map* onde, para cada utilizador, mantém a sua lista de contactos.

```

public class ServidorDistrital {
    private final String nome;
    private final int aresta;
    private List<String>[][] mapa; //matriz de ArrayLists de IDs dos users, para cada localização do distrito
    private Map<String,ArrayList<String>> contactos; //utilizador -> lista contactos que teve
    private List<String> notificacoes; //pseudo lista de users->sockets a notificar
    private int numUtilizadores;
    private int numInfetados;
}

```

Figura 2 - Servidor distrital

Esta classe é composta por métodos para todo o tipo de chamadas ao servidor distrital, sendo alguns dos mais importantes o moveTo, que atualiza a posição de um utilizador na matriz e que invoca o método *atualizarContactos*, que, dado um utilizador e um par de coordenadas, atualiza a sua lista de contactos e a de todos os utilizadores na nova posição das coordenadas.

Temos também o método *notifInfetado*, que é utilizado sempre que um utilizador reporta estar infetado e devolve uma *String* com todos os utilizadores que estiveram em contacto com ele. Por fim, temos ainda o método *top5posicao*, que é chamado por ordem do diretório e que percorre o mapa do distrito, devolvendo, tal como o nome indica, as 5 localizações com mais pessoas nesse momento.

Diretório

O diretório está implementado com base na arquitetura REST-*Dropwizard*, enviando pedidos ao servidor distrital do tipo REQ e recebendo a informação que este lhe envia, devolvendo a mesma no formato JSON. Aquando do primeiro pedido, são estabelecidas as ligações do diretório a todos os servidores distritais para futuros pedidos aos mesmos. Deste modo, o diretório recebe pedidos HTTP do tipo GET, enviando, depois, o mesmo para o servidor distrital. Depois de esperar pela resposta, devolve a informação pedida no formato JSON.

A informação dos pedidos feitos ao diretório é representada através das classes que são apresentadas na imagem abaixo, onde a criação do formato JSON é feita de forma automática, através da atribuição dos valores pretendidos à classe respetiva do pedido efetuado.

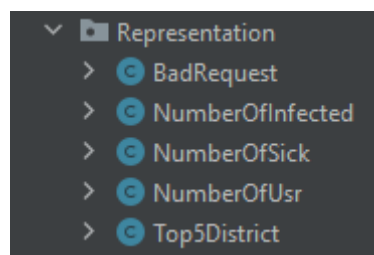


Figura 3 - Estrutura do Diretório

Exemplo de Utilização

Foram feitos vários testes ilustrativos do funcionamento do sistema. É apresentado, de seguida, um exemplo onde existem 4 utilizadores. Os primeiros três a pertencerem ao distrito de Braga e outro ao distrito do Porto. Neste exemplo, podemos ver o funcionamento das notificações públicas no utilizador 3 e notificações de contacto com um infetado em todos os utilizadores.

O que acontece neste exemplo é:

1. Utilizador 1 e 2 encontram-se na mesma posição;
2. Utilizador 4 reporta estar infetado;
3. Utilizador 1 reporta estar infetado;
4. Utilizador 2 vai para a posição do utilizador 3;
5. Utilizador 2 reporta estar infetado;

Utilizador 1:

```
Login feito com sucesso
@-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Escreva as suas coordenadas. apenas serão consideradas validas se o valor se encontrar entre 0 e 20.
Inserir coordenada x:

Inserir coordenada y:

A sua posição foi atualizada
@-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Escreva as suas coordenadas. apenas serão consideradas validas se o valor se encontrar entre 0 e 20.
Inserir coordenada x:

Inserir coordenada y:

2
@-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer confirmar que está infetado?
Pressione 'Y' se sim

◆ Obrigatorio realizar isolamento completo por um periodo minimo de 6 anos.
```

Figura 4 - Exemplo de utilização do utilizador 1

Utilizador 2:

```
Login feito com sucesso
@-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Escreva as suas coordenadas. apenas serão consideradas validas se o valor se encontrar entre 0 e 20.
Inserir coordenada x:

Inserir coordenada y:

A sua posição foi atualizada
@-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações
?2?,ATENCAO! Esteve em contacto com um infetado.

Escreva as suas coordenadas. apenas serão consideradas validas se o valor se encontrar entre 0 e 20.
Inserir coordenada x:

Inserir coordenada y:

A sua posição foi atualizada
@-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer confirmar que está infetado?
Pressione 'Y' se sim

◆ Obrigatorio realizar isolamento completo por um periodo minimo de 6 anos.
```

Figura 5 - Exemplo de utilização do utilizador 2

Utilizador 3:

```

Login feito com sucesso
0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Escreva as suas coordenadas. apenas serão consideradas validas se o valor se encontrar entre 0 e 20.
Inserir coordenada x:
Inserir coordenada y:
A sua posição foi atualizada
0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer ativar/desativar as notificações?
Pressione '1' para ativar, Pressione '2' se sim desativar

Qual distrito quer subscrever?
Porto
Notificações públicas para o distrito Porto ativadas
0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer ativar/desativar as notificações?
Pressione '1' para ativar, Pressione '2' se sim desativar

Qual distrito quer subscrever?
Braga
Notificações públicas para o distrito Braga ativadas

0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer ativar/desativar as notificações?
Pressione '1' para ativar, Pressione '2' se sim desativar

Qual distrito quer subscrever?
Lisboa
Notificações públicas para o distrito Lisboa ativadas

0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer ativar/desativar as notificações?
Pressione '1' para ativar, Pressione '2' se sim desativar

Qual distrito quer subscrever?
Vila Real
Notificações públicas para o distrito Vila Real ativadas
0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer ativar/desativar as notificações?
Pressione '1' para ativar, Pressione '2' se sim desativar

Qual distrito quer subscrever?
Lisboa
ERROR: Não foi possível ativar notificações públicas

```

Figura 6 - Exemplo de utilização do utilizador 3

Utilizador 4:

```

Login feito com sucesso
0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Escreva as suas coordenadas. apenas serão consideradas validas se o valor se encontrar entre 0 e 20.
Inserir coordenada x:
Inserir coordenada y:
A sua posição foi atualizada
0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Escreva as suas coordenadas. apenas serão consideradas validas se o valor se encontrar entre 0 e 20.
Inserir coordenada x:
Inserir coordenada y:
1
0-quit 1-Nova localização 2-Nr pessoas por localização 3-Estou infetado! 4-Subscrição de Notificações

Quer confirmar que está infetado?
Pressione 'Y' se sim

Obrigatorio realizar isolamento completo por um periodo minimo de 6 anos.

```

Figura 7 - Exemplo de utilização do utilizador 4

Como exemplo de utilização do Diretório, adicionamos 2 utilizadores ao distrito de Braga na posição (1,1). De seguida um dos utilizadores informou que estava infetado.

Para a query “http://localhost:8080/User?distrito=Braga” que corresponde ao número de utilizadores num dado distrito, neste caso Braga, a resposta foi a seguinte.

```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

1 {
2   "Status": 200,
3   "Distrito": "Braga",
4   "Número de utilizadores": 2
5 }
```

Figura 8 - Exemplo de utilização do diretório 1

Para a query “http://localhost:8080/Infected?distrito=Braga” que corresponde ao número de infectados num dado distrito, neste caso Braga, a resposta foi a seguinte.



```
1 {
2   "Status": 200,
3   "Distrito": "Braga",
4   "Número de infectados": 1
5 }
```

Figura 9 - Exemplo de utilização do diretório 2

Para a query “http://localhost:8080/Racio” que corresponde ao top 5 dos distritos com maior rácio de infectados/ utilizadores, a resposta foi a seguinte.



```
1 {
2   "Status": 200,
3   "Top 1": "Braga - 0.5",
4   "Top 2": "Lisboa - 0.0",
5   "Top 3": "Porto - 0.0",
6   "Top 4": "Setubal - 0.0",
7   "Top 5": "Aveiro - 0.0"
8 }
```

Figura 10 - Exemplo de utilização do diretório 3

Para a query “http://localhost:8080/Locations” que corresponde ao top 5 das localizações que tiveram o maior número de pessoas em simultâneo, a resposta foi a seguinte.



```
1 {
2   "Status": 200,
3   "Top 1": "Braga - 1,1 - 2",
4   "Top 2": "Lisboa - 16,14 - 0",
5   "Top 3": "Lisboa - 16,13 - 0",
6   "Top 4": "Lisboa - 16,12 - 0",
7   "Top 5": "Lisboa - 16,11 - 0"
8 }
```

Figura 11 - Exemplo de utilização do diretório 4

Para a query “http://localhost:8080/Disease” que corresponde ao número medio de utilizadores que se cruzaram com utilizadores declarados doentes, a resposta foi a seguinte.



```
1 {
2   "Status": 200,
3   "Número médio de utilizadores que se cruzaram com utilizadores declarados doentes": 0.05
4 }
```

Figura 12 - Exemplo de utilização do diretório 5

Conclusão

Concluímos, então, que o objetivo do projeto foi alcançado, considerando ter realizado um bom trabalho e ter correspondido ao enunciado com um bom desempenho. Foram encontrados vários obstáculos ao longo do desenvolvimento do mesmo, mas, de uma forma geral, sentimos tê-los ultrapassado todos com sucesso, culminando este esforço no trabalho descrito neste relatório.

Na nossa maneira de ver, o trabalho cumpre todos requisitos básicos pretendidos.

À medida que o trabalho foi sendo concebido, sentimos que fomos também aprofundando os conceitos lecionados nas aulas teóricas e práticas da Unidade Curricular, nomeadamente no que diz respeito a programação por atores, *sockets* ZeroMQ e arquitetura REST-*Dropwizard*. Sendo assim, fazemos um balanço positivo do projeto, do trabalho realizado e do seu resultado final.