



Universidade De Évora
Curso: Engenharia Informática
UC: Sistemas Distribuídos
Docente: José Saias
Entregue Janeiro 2023

Room Rent

Relatório Do Trabalho Prático



Alunos:

Joana Carrasqueira, nº 48566

João Condeço, nº 48976

Índice

Introdução.....	2
Server	2
Client	3
Admin	4
Extras	5
Conclusão	5

Introdução

Este trabalho prático tem com o objetivo conciliar os conhecimentos adquiridos na cadeira de Sistemas Distribuídos para aprimorar sistema desenvolvido no primeiro trabalho, de modo a garantir que os métodos de comunicação utilizados não seriam dependentes da tecnologia. Assim, recorreremos ao REST Jersey na implementação do servidor (Server). Criámos as seguintes aplicações:

- ⇒ Client
- ⇒ Admin
- ⇒ Server

Este projeto foi desenvolvido em *Java* com o auxílio das ferramentas de automação de compilação *Maven*, no caso do Server, e Gradle, no caso dos clientes (Client, Admin).

Server

A aplicação server tem como objetivo fornecer um servidor REST implementado em Java, com o auxílio da framework Jersey. Verificadas as condições o servidor fica online até o utilizador o encerrar. A conexão à base de dados, do qual este depende, é feita com base nas credenciais indicadas no ficheiro *database.properties*. Foi ainda incluído na pasta do servidor um ficheiro com os dados utilizados para testar o trabalho. Desta forma o professor pode gerar a base de dados e testar o trabalho.

⇒ Compilação:

```
mvn clean  
mvn compile
```

⇒ Execução:

```
mvn exec:java -Dexec.args="port"
```

Client

A aplicação client tem como objetivo permitir ao utilizador comum interagir com o programa. Fornecendo a este as seguintes funcionalidades:

- ⇒ sign in – Inicia a sessão no sistema;
- ⇒ sign up - Regista utilizador no sistema;
- ⇒ add - Permite adicionar anúncios;
- ⇒ show ads - Mostra os anúncios atualmente ativos;
- ⇒ show search - Permite procurar com filtros;
- ⇒ search by name - Permite procurar anúncios pelo nome do anunciante;
- ⇒ search by id - Mostra os anúncios através do id inserido;
- ⇒ send msg ou send message - Envia uma mensagem para um anúncio ou para um utilizador;
- ⇒ check msg ou check message - Permite ver as mensagens de um anúncio ou de um utilizador;
- ⇒ check users - Permite ver os utilizadores do sistema;
- ⇒ logout - Termina a sessão do utilizador no sistema;
- ⇒ help ou ? - Mostra este menu;
- ⇒ quit - Termina o programa.

Existem certas funcionalidades que exigem que o sign in seja feito previamente.

⇒ **Compilação:**

```
gradle build
```

⇒ **Execução:**

```
gradle run -args="address port"
```

Admin

A aplicação admin tem como objetivo permitir ao utilizador de gestão interagir com o programa e administrar as informações. Fornecendo a este as seguintes funcionalidades:

- ⇒ sign in - Inicia a sessão no sistema;
- ⇒ show ads - Mostra os anúncios atualmente ativos;
- ⇒ show search - Permite procurar com filtros;
- ⇒ search by name - Permite procurar anúncios pelo nome do anunciante;
- ⇒ search by id - Mostra os anúncios através do id inserido;
- ⇒ activate ad - Muda o estado do anúncio para ativo do id inserido;
- ⇒ deactivate ad - Muda o estado do anúncio para inativo do id inserido;
- ⇒ upgrade user - Muda um utilizador cliente para gestor do id inserido;
- ⇒ check users - Permite ver os utilizadores do sistema;
- ⇒ remove ad - Remove um anúncio de acordo com o id inserido;
- ⇒ remove user - Remove um utilizador de acordo com o username inserido;
- ⇒ logout - Termina a sessão do utilizador no sistema;
- ⇒ help ou ? - Mostra este menu;
- ⇒ quit - Termina o programa.

O sign in é exigido logo de início para garantir que o utilizador comum não tem acesso aos dados e funcionalidades de administrador.

⇒ **Compilação:**

```
gradle build
```

⇒ **Execução:**

```
gradle run -args="address port"
```

Extras

Dos extras propostos implementamos os seguintes:

⇒ redundância

⇒ pub sub

Conclusão

Na elaboração deste projeto baseámo-nos no trabalho anterior, apresentando as mesmas funcionalidades, mas com as correções propostas e uma implementação diferente.

A nível dos extras foi implementado a funcionalidade pub/sub. A ideia passa por cada vez que é inserido um novo anúncio ou uma mensagem é enviada para um anúncio já existente os clients serem notificados de tal ocorrência, no entanto apenas é notificado um client ativo sendo este um defeito do trabalho em causa.