

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

Sistemas Distribuídos

Ano Letivo 2021/22

Ficha 6

Elaborado em: 2022/04/19

Daniel Santos | Nº2019133865

Índice

List of Figures	ii
1. Exercício 1	1
2. Exercício 2	2

List of Figures

FIGURA 1 - SERVER E NOME DO CONTADOR.....	1
FIGURA 2 - CLIENTE E SUA INCREMENTAÇÃO NO CONTADOR	1
FIGURA 3 - ERRO CLIENT SOCKETEXCEPTION	1
FIGURA 4 ERRO CLIENT CONNECTIONEXCEPTION	1
FIGURA 5 - COMPUTADOR DO COLEGA A INICIAR O SERVER.....	2
FIGURA 6 - IP DO SERVER INTRODUZIDO NO CLIENT	2
FIGURA 7 - CLIENT A COMUNICAR COM O SERVER.....	2
FIGURA 8 - SERVER METHODS	3

1. Exercício 1

Execução do programa:

```
run:  
Nome do objeto remoto: test  
servidor RMI iniciado
```

Figura 1 - Server e nome do contador

```
Contadores disponíveis:  
test  
Nome do contador: test  
Valor inicial: 0  
Valor final: 6
```

Figura 2 - Cliente e sua incrementação no contador

Pergunta: O que ocorre com o registo de um objeto remoto junto ao RMI Registry quando o programa servidor relacionado termina?

```
java.rmi.UnmarshalException: Error unmarshaling return header; nested exception is:  
    java.net.SocketException: Connection reset
```

Figura 3 - Erro Client SocketException

Resposta: A ligação termina, pois, o servidor fechou a socket a que o cliente estava ligado.

Pergunta: O que ocorre com um programa cliente que tente aceder a este objeto?

```
java.rmi.ConnectException: Connection refused to host: localhost; nested exception is:  
    java.net.ConnectException: Connection refused: connect
```

Figura 4 Erro Client ConnectionException

Resposta: Não dá para conectar pois o servidor não está a manter o objeto remoto.

2. Exercício 2

Execução do programa:

```
run:  
Nome do objeto remoto: teste  
servidor RMI iniciado
```

Figura 5 - Computador do colega a iniciar o Server

```
Introduza o ip: 192.168.2.49
```

Figura 6 - Ip do Server introduzido no Client

```
Conectado...  
1.Ver cotação  2.Dolar para Euro  3.Euro para Dolar  4.Alterar cotação  0.Sair  
Introduza a opção que deseja: 1  
1USD = 1.0EUR , N/A  
1.Ver cotação  2.Dolar para Euro  3.Euro para Dolar  4.Alterar cotação  0.Sair  
Introduza a opção que deseja: 4  
Cotação: 0,5  
1.Ver cotação  2.Dolar para Euro  3.Euro para Dolar  4.Alterar cotação  0.Sair  
Introduza a opção que deseja: 1  
1USD = 0.5EUR , 19-04-2022 15:11:46  
1.Ver cotação  2.Dolar para Euro  3.Euro para Dolar  4.Alterar cotação  0.Sair  
Introduza a opção que deseja: 2  
Quantia dolar: 1  
0.5  
1.Ver cotação  2.Dolar para Euro  3.Euro para Dolar  4.Alterar cotação  0.Sair  
Introduza a opção que deseja: 3  
Quantia euro: 1  
2.0  
1.Ver cotação  2.Dolar para Euro  3.Euro para Dolar  4.Alterar cotação  0.Sair  
Introduza a opção que deseja: 0  
Adeus...
```

Figura 7 - Client a comunicar com o Server

Como desenvolvi o código:

```
public String obterCotacao() throws RemoteException {
    String output;
    output = "1USD = " + cotacao + "EUR , " + time;
    return output;
}

public float UsdToEur(float num) throws RemoteException {
    EUR = cotacao * num;
    return EUR;
}

public float EurToUsd(float num) throws RemoteException {
    USD = (1/cotacao) * num;
    return USD;
}

public void alterarCotacao(float num) throws RemoteException {
    cotacao = num;
    LocalDateTime currentDateTime = LocalDateTime.now();
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");
    time = currentDateTime.format(formatter);
}
```

Figura 8 - Server Methods

Para o Client conectar-se à ficha este introduz o ip do servidor.

Depois aparece as várias funções ao cliente como apresentado na página anterior (fig. 7).

A primeira função cria um string com o valor da cotação e o tempo quando foi alterado e retorna.

A segunda função converte de dólar para euro.

A terceira função converte de euro para dólar.

A quarta função pede ao utilizador a cotação que deseja e atualiza a data da última alteração.