
Sistemas Distribuídos 2016/2017

Guiao de Demonstração de Segurança (Parte 4 do Projeto)

Grupo 35

Projeto no website do GitHub: <https://github.com/tecnico-distsys/T35-Komparator>

1. Para obter o projeto:

```
git clone -b SD_P4 https://github.com/tecnico-distsys/T35-Komparator/
```

2. Para instalar todos os módulos:

```
cd T35-Komparator/  
mvn clean install -DskipITs
```

A demonstração das várias funcionalidades encontra-se numa só execução do cliente, sendo cada parte desta devidamente explicitada. Explicaremos de seguida as várias componentes da demonstração, alertando em particular para a ocorrência dos seguintes casos:

Caso R1: passos para demonstrar funcionamento normal da replicação

Caso R2: passos para demonstrar tolerância a falta

Serão necessários 5 terminais. 2 para os mediadores, 2 para os suppliers e 1 para o cliente.

Na classe MediatorPortImpl do mediator-ws, no método buyCart, encontra-se em comentário um Thread.sleep(). Descomente-o.

Comece por correr os 2 suppliers e os 2 mediadores. Em cada terminal, e assumindo que se encontra na pasta base do projecto:

```
cd supplier-ws
mvn exec:java
```

```
cd supplier-ws
mvn exec:java -Dws.i=2
```

```
cd mediator-ws
mvn exec:java
```

```
cd mediator-ws
mvn exec:java -Dws.i=2
```

Espere até que os vários servidores fiquem à espera de conexões. De seguida, corra, no ultimo terminal, o cliente:

```
cd mediator-ws-cli
mvn exec:java
```

A partir daqui, terá instruções no terminal para o que deverá fazer, e o que cada operação revela. Encontra-se aqui uma descrição das várias operações:

1. Os suppliers serão populados com alguns produtos. Ao mediador será adicionado um carrinho de compras.
2. O cliente procederá a correr o método buyCart(). Note-se aqui a presença de **[R2]**: o cliente terá de suportar um timeout, repetir o pedido e o mediador terá de reconhecer que o pedido já foi feito anteriormente e agir de acordo. Ocorre também **[R1]** no terminal do mediador secundário – são feitos prints sempre que recebe uma alteração ao carrinho compras ou ao histórico de compras.

3. O cliente apresentar-lhe-á o resultado da compra, para comprovar o funcionamento esperado do buyCart().
4. De seguida, o cliente adicionará novos produtos ao carrinho de compras, e imprimirá os conteúdos actuais da lista de carrinhos de compras e de histórico de compras do mediador.
5. Pedir-lhe-á então para terminar forçadamente (Ctrl-C) o mediador primário. De seguida, pedirá novamente os valores actuais da lista de carrinhos de compras e de histórico de compras ao mediador (previamente secundário), até que este responda. Após a resposta, imprimirá estes valores. Note-se que aqui serão demonstrados **[R1]** e **[R2]**:
 - a. O mediador secundário terá que tomar o lugar do primário.
 - b. O cliente terá de repetir pedidos enquanto receber Connection Refused.
 - c. Os resultados da lista de carrinhos de compras e de histórico de compras terão que ser iguais aos que foram obtidos antes de ser terminado o mediador primário, comprovando a manutenção do estado.

Termina assim a demonstração.