

## Instruções de Instalação e Configuração

Obtenção do código no repositório Git :

Para obter o código deve-se ir primeiro para a directoria onde se quer colocar a raiz do projecto , de seguida deve se abrir a uma bash do git e correr o seguinte comando:

```
$ git clone https://github.com/tecnico-distsys/A53-Komparator.git
```

Depois de correr o comando a cima , ficará então com uma cópia local do código.

Compilação do código:

Antes de compilar o código deve-se certificar que tem as seguintes bibliotecas instaladas na sua máquina:

- uddi-naming
- cert-util
- junit

O projecto contém um master-pom que vai buscar todos os serviços do qual do projecto.

Antes de instalarmos as dependências do projecto temos de gerar os recursos para cada serviço.

```
mvn generate-resources
```

Podemos então instalar todas as dependências do código e compilar, para isso abra a pasta raiz do projecto ../A53-Komparator e abra uma bash nessa directoria e corra o seguinte comando:

```
mvn clean install -DskipITs
```

Agora que temos as dependências instaladas e o código compilado passemos à execução do mesmo.

Para isso precisamos correr os serviços por uma ordem específica, deve então proceder da seguinte forma:

Na directoria ../A53-Komparator/supplier-ws/ abrir uma bash e correr o seguinte comando:

```
mvn exec:java
```

Ficando então com o serviço "A53-Supplier1" a correr nessa bash.

Abrir uma bash na mesma directoria e correr o seguinte comando:  
`mvn exec:java -Dws.i=2`

Ficando então com o serviço "A53\_Supplier2" a correr nessa bash.

Abrir uma bash na mesma directoria e correr o seguinte comando:  
`mvn exec:java -Dws.i=3`

Ficando então como serviço "A53\_Supplier3" a correr nessa bash.

Agora que tem os Suppliers a correr pode correr o serviço A53\_Mediator, para isso na directoria ../A53-Komparator/mediator-ws/ deve correr o seguinte comando:

```
mvn exec:java -Dws.i=2
```

Ficando então com o servidor secundário a correr nessa bash.

```
mvn exec:java
```

Ficando então com o servidor primário "A53\_Mediator" a correr nessa bash.

A razão de se lançar primeiro o servidor secundário é por este ser um backup do primário, ou seja, queremos que mesmo antes de o primário ser lançado já esteja pronto para qualquer eventualidade.

## Demonstração de Segurança e Tolerância de Faltas

Agora que tem os serviços "A53-Supplier1", "A53\_Supplier2", "A53\_Supplier3" e "A53\_Mediator" a correr podemos passar à demonstração de casos de segurança e tolerância de faltas.

### Segurança :

Duração inferior a 5 minutos

Caso S1:

Passos para demonstrar funcionamento normal:

- Abrir uma bash na directoria ../A53-Komparator/mediator-ws-cli/
- Correr o comando :  
mvn verify -Dtest=BuyCartIT

Irá correr o teste em que se compra um carrinho, deve observar especificamente o elemento da mensagem soap <creditCardNr> e poderá observar que o seu valor está encriptado e apenas o serviço "A53-Mediator" será capaz de saber a informação que se encontra lá dentro.

Caso S2::

Duração inferior a 5 minutos

Passos para demonstrar funcionamento normal:

- Abrir uma bash na directoria ../A53-Komparator/supplier-ws-cli/
- Correr o comando :  
mvn verify

Verifica-se que enquanto os testes estão a ser corridos e existe troca de mensagens podemos observar que os valores do header das mensagens soap estão sempre a ser actualizados e imprimidos com a data em que foi enviada a mensagem.

Caso S3:

Passos para demonstrar ataque e como se lida com ele:

- No ficheiro DateHeaderServerHandler que se encontra na directoria ../A53-Komparator/security/src/main/java/org/komparator/security/handler/ trocar o código onde se encontra a seguinte linha de código:  
if(Math.abs(sdf.parse(headerValue).getSeconds() - (new Date()).getSeconds())) > 3)

para :

if(Math.abs(sdf.parse(headerValue).getSeconds() - (new Date()).getSeconds())) -20 > 3)

- Abrir uma bash na directoria ../A53-Komparator/supplier-ws-cli/
- Correr o comando :  
mvn verify

Podemos observar que os valores das mensagens soap foram actualizados e imprimidos com uma soapfault.

**Replicação:**

Duração inferior a 5 minutos

**Caso R1:**

Passos para demonstrar funcionamento normal da replicação.

Assumindo que todos os passos de configuração apresentados neste guião foram seguidos, podemos então:

- Abrir uma bash na directoria ../A53-Komparator/mediator-ws-cli/
- Correr o comando :  
mvn verify -Dtest=BuyCartIT

Tendo as consolas do servidor primário , servidor secundário e do cliente abertas, podemos então verificar que as alterações que são feitas no servidor primário estão a ser propagadas para o servidor secundário e também que de 5 em 5 segundos o servidor primário envia uma mensagem de imAlive ao servidor secundário.

**Caso R2:**

Passos para demonstrar tolerância a falta.

Assumindo que todos os passos de configuração apresentados neste guião foram seguidos, podemos então:

- Causar a paragem súbita do servidor primário através de sigkill (CTRL-C)

Tendo as consolas do servidor primário e do servidor secundário abertas, podemos então verificar que o secundário ao fim de 5 segundos sem ter recebido a mensagem de imAlive por parte do servidor primário, vai então publicar-se para o uddi e assumir-se como servidor primário.

**Caso R3:**

Passos para demonstrar substituição do servidor.

Assumindo que todos os passos de configuração apresentados neste guião foram seguidos, podemos então:

- Causar a paragem súbita do servidor primário através de sigkill (CTRL-C)

Tendo as consolas do servidor primário e do servidor secundário abertas, podemos então verificar que o secundário ao fim de 5 segundos sem ter recebido a mensagem de imAlive por parte do servidor primário, vai então publicar-se para o uddi e assumir-se como servidor primário.

- Abrir uma bash na directoria ../A53-Komparator/mediator-ws-cli/
- mvn exec:java (Vai correr o comando Ping)

Verifica-se na consola do servidor que anteriormente era servidor secundário que a comunicação está ser feita e como servidor primário.

#### Caso R4:

Passos para demonstrar funcionamento do front end, pedidos repetidos.

Assumindo que todos os passos de configuração apresentados neste guião foram seguidos, podemos então:

- No ficheiro  
../A53-Komparator/mediator-ws-cli/src/main/java/org/komparator/mediator/ws/cli/MediatorApp.java descomentar a linha  
String res2 = client.ping("pinga 2");
- Abrir uma bash na directoria ../A53-Komparator/mediator-ws-cli/
- mvn exec:java (Vai correr o comando Ping)

Tendo as consolas do servidor primário e do cliente podemos então verificar que o cliente recebe a mesma resposta para os pedidos de ping da parte do servidor primário, pois seriam o mesmo pedido.