

## Guião de demonstração

### Instruções de instalação e configuração

1. Obtenção do código entregue através do git executando o comando “git clone -b SD\_P4 <https://github.com/tecnico-distsys/A52-Komparator/>”
2. Compilar o projeto através do master pom.xml na diretoria base do projeto utilizando o comando “mvn clean install -DskipTs”
3. Abrir 1 terminal onde irá ser executado o servidor secundário Mediator: entrar na pasta mediator-ws e utilizar o comando “mvn exec:java -Dws.i=2”
4. Abrir 3 terminais para os servidores Supplier: entrar na pasta supplier-ws e utilizar o comando “mvn clean compile exec:java -Dws.i={Numeros}” sendo os valores de Numeros 1,2 e 3.
5. Se o servidor secundário já estiver a correr (passo 3), abrir 1 terminal onde irá ser executado o servidor primário: entrar na pasta mediator-ws e utilizar o comando “mvn exec:java -Dws.i=1”
6. Neste ponto, com os 2 servidores Mediator a correr, já podemos observar a receção das mensagens ImAlive no terminal do mediador secundário a cada 5 segundos.

### Caso 1: funcionamento normal

<b>Descrição</b>	Executar uma operação addToCart para observar que o cliente adiciona um nonce ao header para identificar o seu pedido e que o estado do servidor secundário é atualizado através da operação updateCart.
<b>Estado inicial</b>	Os 3 terminais de servidores Supplier e os terminais dos servidores primário e secundário de Mediator estão em execução.
<b>Instruções de execução</b>	<ol style="list-style-type: none"> <li>1. Abrir 1 terminal na pasta mediator-ws-cli (MediatorClient)</li> <li>2. Executar um caso de teste de AddToCart através do comando “mvn verify -Dit.test=AddToCartIT#successWithMaxQuantity”</li> <li>3. Analisar mensagens impressas nos terminais de cada servidor</li> </ol>
<b>Resultados esperados</b>	<p>O servidor secundário recebe mensagens ImAlive do primário a cada 5 segundos.</p> <p>A mensagem SOAP enviada pelo MediatorClient para o servidor Mediator tem o nonce identificador do pedido no header.</p> <p>O servidor secundário recebe um pedido de updateCart do primário e o número de pedidos guardados no mapa passa de 0 para 1.</p>

**Caso 2: tolerância a uma falta**

<b>Descrição</b>	Executar uma operação addToCart e provocar falha do servidor primário antes do cliente receber uma resposta. Pretendemos observar que o cliente volta a repetir o pedido, que o secundário substitui o primário no UDDI e que o secundário responde ao pedido repetido do cliente com a resposta guardada no mapa.
<b>Estado inicial</b>	Os 3 terminais de servidores Supplier e os terminais dos servidores primário e secundário de Mediator estão em execução.
<b>Instruções de execução</b>	<ol style="list-style-type: none"> <li>1. Abrir 1 terminal na pasta mediator-ws-cli (MediatorClient)</li> <li>2. Executar um caso de teste de AddToCart através do comando "mvn verify - Dit.test=AddToCartIT#successWithMaxQuantity"</li> <li>3. Quando o terminal do servidor primário, apresentar a mensagem "5 segundos de espera, caso queiras parar o servidor", enviar sigkill para o terminal através do ctrl+c.</li> <li>4. Analisar mensagens impressas nos terminais de cada servidor e do cliente</li> </ol>
<b>Resultados esperados</b>	<p>No terminal do cliente, podemos observar mensagens de tentativa de renovar a ligação a um servidor mediator através do UDDI.</p> <p>No terminal do servidor secundário, podemos observar que o secundário deteta a falha do primário e se publica no UDDI para substituir o primário.</p> <p>Ainda no terminal do servidor secundário, podemos observar que o secundário deteta que já tem a resposta ao pedido repetido pelo cliente ao imprimir a mensagem "(addCart)ja tenho este token".</p>