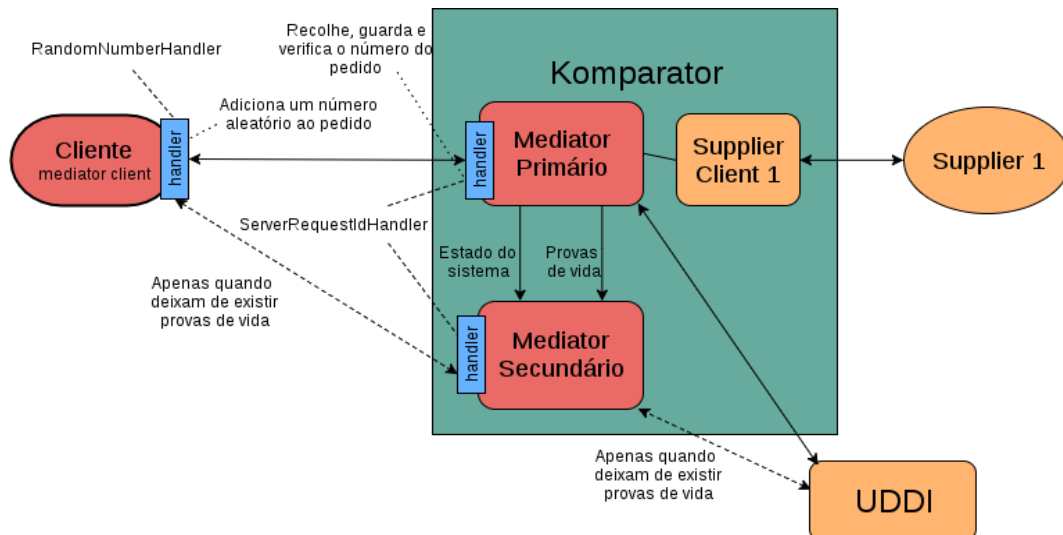


# Komparator

A52 - Tolerância a faltas

		
Daniel Correia 80967	João Crespo 81811	Pedro Caldeira 81888
<a href="https://github.com/tecnico-distsys/A52-Komparator">https://github.com/tecnico-distsys/A52-Komparator</a>		



Com esta entrega passamos a permitir que exista **uma** falha de um dos servidores Mediator sem que o cliente tenha qualquer tipo de constrangimento na utilização do serviço disponibilizado.

## Requisitos de tolerância a faltas

### 1) Réplica do servidor mediator

A nossa solução consiste na criação de um servidor secundário que mantém o seu estado atualizado com o primário e que está pronto a substituí-lo caso este demonstre uma falha na comunicação.

O servidor secundário só se regista no servidor UDDI se o primário falhar.

O **servidor primário, sempre que realiza uma operação não idempotente** (buyCart e addToCart), **envia ao secundário a informação necessária para se actualizar**, mantendo assim os estados de ambos os servidores consistentes.

### 2) Substituição do servidor primário pelo secundário

Para ser realizada a substituição do servidor primário, o secundário tem de saber quando o primário deixa de funcionar.

Por isso, criámos a classe LifeProof que é uma TimerTask responsável por: (1) **executar o método run a cada 5 segundos**; (2) **guardar o timestamp da última prova de vida** do primário recebida pelo servidor secundário.

O comportamento do método run varia em função do contexto de execução:

- No **servidor primário**, é chamado o método ImAlive e **atualiza-se o timestamp da última prova de vida recebida**.
- Se **correr no servidor secundário** e a última notificação do primário tiver ocorrido à **mais de 7 segundos**, então o **secundário substitui o primário no UDDI** e passa a responder aos pedidos do cliente.
- Se o secundário já estiver a substituir o primário, o método não faz nada.

Alterámos também propriedades do timer da biblioteca JaxWS que definimos para **10 segundos para que o cliente desista de esperar pela resposta**. Se o cliente não obtiver resposta do primário ao fim desse tempo (imaginemos, por falha do servidor) este **procura repetidamente pelo novo URL do serviço no UDDI para poder repetir o pedido**.

Se o primário deixar de comunicar, o servidor secundário regista-se no UDDI para o substituir (devido ao timeout no LifeProof) e o cliente passa a enviar pedidos para este.

### 3) Front-End Cliente com semântica no-max-1-vez.

Para garantir a semântica no-max-1-vez, fazemos com que **o cliente envie em todos os métodos não idempotentes um nonce que identifica o pedido** do cliente. As invocações idempotentes não necessitam de tratamento especial pois, se repetidas, não alteram o estado do servidor.

No servidor temos dois mapas, um para addToCart e outro para BuyCart que guardam o RequestID e a resposta ao pedido. **Se o cliente realizar duas vezes o mesmo pedido, o servidor envia a resposta que tem guardada no mapa.**

Os clientes apenas geram um novo nonce quando recebem uma resposta do servidor, ou seja, quando terminam uma operação com sucesso.

### Estrutura das mensagens SOAP com a solução de tolerância a faltas

Mensagens enviadas pelo mediator primário para o secundário para provar que está vivo

Timestamps das mensagens SOAP de prova de vida, devem ser recebidas com um intervalo máximo de 7 segundos

```
[2017-05-19T11:26:05.976] intercepted INbound SOAP message:
<S:Envelope xmlns:S="h...e/" xmlns:SOAP-ENV="h...e/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:imAlive xmlns:ns2="http://ws.mediator.komparator.org/"/>
  </S:Body>
</S:Envelope>
```

```
[2017-05-19T11:26:10.975] intercepted INbound SOAP message:
<S:Envelope xmlns:S="h...e/" xmlns:SOAP-ENV="h...e/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:imAlive xmlns:ns2="http://ws.mediator.komparator.org/"/>
  </S:Body>
</S:Envelope>
```

Mensagem enviada pelo mediator-client para o servidor primário com o número de pedido

```
[2017-05-19T12:05:12.086] intercepted OUTbound SOAP message:
<S:Envelope xmlns:S="http://sch...e/" xmlns:SOAP-ENV="http:...ope/">
  <SOAP-ENV:Header>
    <w:clientRequestId xmlns:w="htt...org/">
      17713452
    </w:clientRequestId>
  </SOAP-ENV:Header>
  <S:Body>...</S:Body>
</S:Envelope>
```

Número do pedido, gerado aleatoriamente, garante que os pedidos repetidos recebem a mesma resposta e não alteram o estado do sistema