

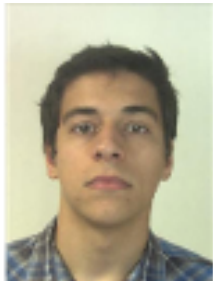
# Projeto Komparator

## Sistemas Distribuídos

2016/2017

GRUPO T50

<https://github.com/tecnico-distsys/T50-Komparator>



77917

Daniel Madruga

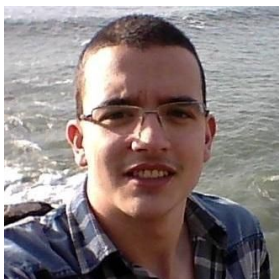
[daniel.madruga@tecnico.ulisboa.pt](mailto:daniel.madruga@tecnico.ulisboa.pt)



78013

Bruno Henriques

[bruno.s.henriques@tecnico.ulisboa.pt](mailto:bruno.s.henriques@tecnico.ulisboa.pt)

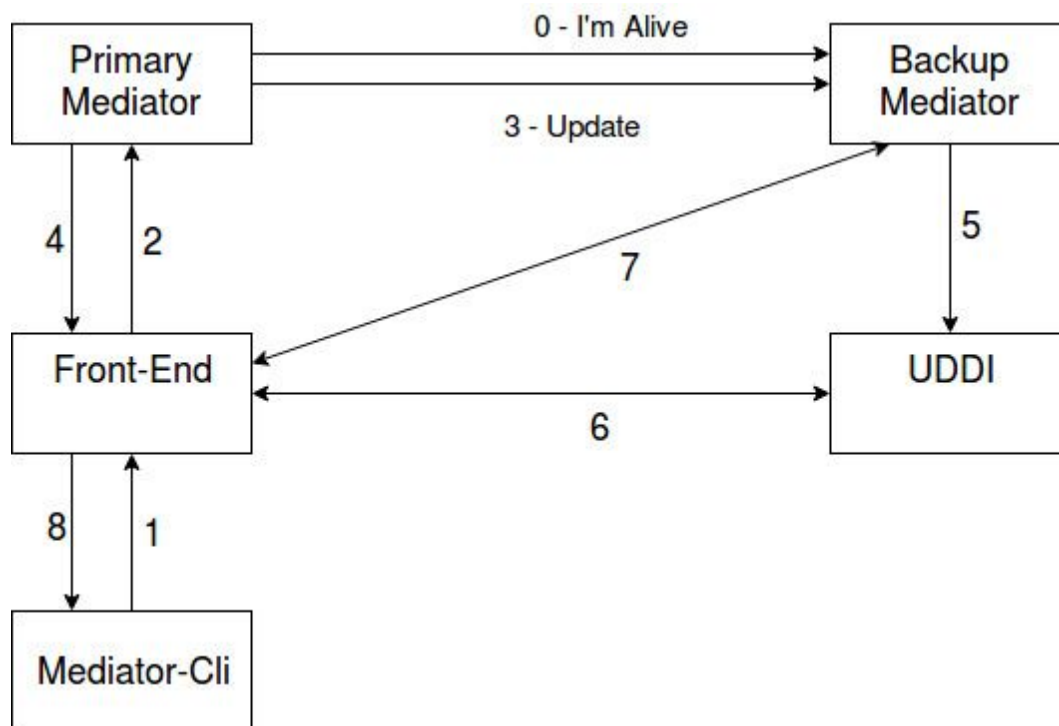


78042

Bruno Carola

[bruno.carola@tecnico.ulisboa.pt](mailto:bruno.carola@tecnico.ulisboa.pt)

## Esquema da Replicação



## Normal Funcionamento

Ligação 1 : O módulo Mediator-Cli utiliza a classe Front End que serve de interface para comunicar com o servidor Mediator disponível.

Ligação 2 : O Front-End comunica o pedido com o Primary Mediator.

Ligação 3 : O Mediator Primário envia informação para atualizar o estado do servidor de Backup que foi alterado.

Ligação 4 : Resposta do Mediator Primário para o Front End.

Ligação 8 : O Front End passa essa resposta ao Mediator-Cli.

## Se a comunicação com o Primary Mediator não for bem sucedida

Se o Mediator Primário não enviar uma prova de vida quando era suposto, o Backup Mediator ao fim de dois segundos assume que este morreu e, neste caso, publica-se no UDDI no lugar do Primário.

Ligação 4 : O Front End apanha uma exceção, e espera o tempo suficiente para o Backup Mediator se publicar no UDDI.

Ligação 5 : O Backup publica-se no UDDI.

Ligação 6 : O Front End vai ao UDDI buscar o URL do Backup Mediator.

Ligação 7 : Faz o Pedido e recebe a resposta do Backup Mediator.

Ligação 8 : O Front End passa essa resposta ao Mediator-Cli.

## Mecanismo de Substituição

O Mediator Primário de X em X segundos (valor configurável), com o valor inicial de 5 segundos, envia uma mensagem ao Mediator Secundário. O Mediator Secundário guarda a data da hora da chegada dessa prova de vida (*timestamp*).

Se ao fim de um intervalo de tempo esperado (com 2 segundos adicionais para suportar o atraso que os pacotes podem sofrer na rede) ainda não tiver recebido a prova de vida do servidor Primário, o Backup Mediator publica-se no UDDI para o substituir e torna-se o Primário, passando ele a executar as operações.

## Garantir Semântica No-Max-Uma-Vez

```
Primary: Sending imAlive signal to BackupMediator...
LoggingHandler: Handling INbound message. at 2017-05-19T00:06:33.147
<S:Envelope>
  <SOAP-ENV:Header>
    <token>yvBai2Q4npKoECy2wWZ2fQ==</token>
  </SOAP-ENV:Header>
  <S:Body>
    <ns2:ping xmlns:ns2="http://ws.mediator.komparator.org/">
      <arg0>clientXPT0</arg0>
    </ns2:ping>
  </S:Body>
</S:Envelope>
```

De modo a garantir a semântica no-máximo-uma-vez, utiliza-se um *handler* FreshnessHandler na comunicação Mediator <-> Mediator-Cli, que adiciona um identificador(*token*, como a figura ilustra) único a cada operação.

Os pedidos são registados num ficheiro de tokens. No caso dos Mediators (primário e secundário), este ficheiro é partilhado. Isto garante que os identificadores se encontram sincronizados entre Mediators.

Este identificador é comparado com os identificadores existentes. Caso exista, a operação é rejeitada, evitando que seja processada mais que uma vez.