

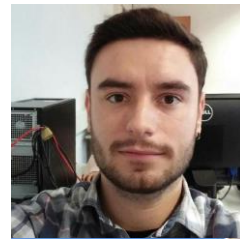
# Sistemas Distribuídos 2016/2017

## Relatório de Tolerância a Faltas (Parte 4 do Projeto)

### Grupo 35

76479

Miguel Marques



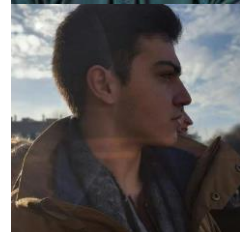
77915

Goncalo Santos



78029

Vasco Pombo



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

The diagram illustrates a Mediator-based system architecture. It includes the following components and interactions:

- Primary Mediator** (oval) and **Secondary Mediator** (oval): Both are connected to the **LifeProof** component.
- LifeProof** (rectangle): There are two instances. The bottom one receives **imAlive** messages from the Primary Mediator and sends **updateCart** and **updateShoppingHistory** messages to the top one.
- UDDI** (oval): Receives messages from the Secondary Mediator and the bottom LifeProof component.
- Mediator Client** (oval): Interacts with the Primary Mediator and UDDI.

Failure points are indicated by large 'X' marks:

- A red 'X' marks a failed message from the top LifeProof to the Secondary Mediator.
- A yellow 'X' marks a failed message from the Mediator Client to the Primary Mediator.
- A green 'X' marks a failed message from the Mediator Client to the UDDI.

Existem dois mediadores. O Mediador Primário regista-se no UDDI e recebe pedidos do cliente. O Mediador Secundário não se regista no UDDI.

Além disto, existe um método `imAlive`, que envia provas de vida num intervalo de tempo arbitrário ao mediador secundário. Na eventualidade de o intervalo de tempo ultrapassar o definido (cruz vermelha), o mediador secundário regista-se no UDDI com o mesmo nome do primário, sobrepondo-se a este e recebendo dali em diante pedidos do cliente.

Do ponto de vista do cliente, este comunica com o mediador primário até que algum pedido demore mais do que um tempo pré-definido a ser obtido ou que a ligação seja recusada (cruzes amarela e verde). Ocorrendo algum destes eventos, o cliente acede ao UDDI, obtém novamente o endereço do mediador e tenta enviar o seu pedido de novo. Desta forma, poderá comunicar com o mediador secundário após este detectar a falha do mediador primário.

O último ponto refere-se ao tratamento do envio repetido de mensagens iguais. Para garantir a semântica no-máximo-uma-vez, o cliente envia, com o seu pedido, um identificador único. O servidor verifica se um pedido com este identificador alguma vez foi recebido. Se não, procede normalmente. Se sim, devolve o mesmo resultado que devolveu da primeira vez que leu esse identificador, sem proceder a alterações do seu estado. Esta lógica foi implementada modificando o WSDL para permitir o envio do identificador.