



Instituto Superior Técnico - Universidade de Lisboa  
Sistemas Distribuídos

Alameda Campus  
Grupo A09

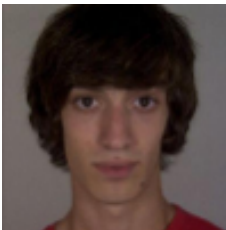
## Relatório - Tolerância a falhas

---

### Sistema Binas

<https://github.com/tecnico-distsys/A09-SD18Proj>

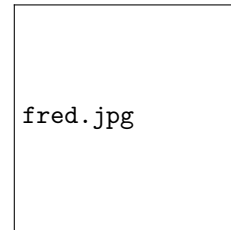
---



Pedro Mela  
78876



Moustafa  
ist number



Frederico Delaere  
79674



# 1 Descrição e Explicação da Solução

A solução de tolerância de faltas, passou por implementar o protocolo Quorum Consensus como forma de replicação activa da informação sobre o user, os métodos `getCredit` (agora `getCreditReplic`), `rentBina`, `returnBina` do gestor de réplicas Binas-ws foram modificados para consultar um quorum de respostas, esta implementação permite tolerar até  $2/N+1$  falhas de réplicas, uma replica que entre em falha temporária e recupere, está pronta para participar, encontrando a sua informação sobre user actualizada no próximo pedido de escrita (neste caso informação correcta sobre o utilizador).

Ao WSDL foi acrescentado um novo tipo complexo que representa a informação do utilizador replicada, com mais informação necessária para que o binas consiga implementar o quorum e para que seja possível respeitar consistência sequencial: `cid` (identificador da estação), `seq` (número de sequência), `request_id` (id do pedido de leitura, ou escrita)

O binas funciona como um gestor de réplicas de  $N$  `stationClients` e implementa o protocolo Quorum Consensus, extendemos o contracto WSDL `station_1.0` para suportar chamadas assíncronas por parte de um cliente nos métodos de `station-ws`, em particular para que o gestor de réplicas binas possa ler ou escrever sem bloquear, a troca de mensagens assíncrona entre o gestor de replicas binas-ws e as `stations-ws` permitirá ao cliente esperar apenas por um quórum de respostas (para cada pedido ou de escrita ou leitura que use o Quórum, existe um timeout para detecção de falha caso a maioria não seja atingida) em vez de ter que esperar pela resposta de todas as estações. Com as chamadas Assincronas `getBalanceAsync` e `setBalanceAsync`, o binas consegue usar o quórum e realizar pedidos que envolvam leitura ou escrita.

## 2 Troca de Mensagens no protocolo de tolerância a faltas

### Troca de Mensagens do protocolo em pedidos de Leitura (operação `getUserReplic`):

Nos pedidos que envolvem leitura, binas prepara um pedido identificado unicamente por `request_id`, e vai obter respostas do tipo `UserReplic` vindas de cada uma das  $N$  réplicas `station-ws` que replicam a informação do utilizador, estas respostas são guardadas com chave `request_id` correspondente ao pedido a que dizem respeito no gestor de replicas binas, que através das chamadas assíncronas com callback `getBalanceAsync`, não bloqueia, e entra num ciclo com a condição de esperar por quórum de  $N/2+1$  respostas (Por exemplo, se  $N=3$ , binas só espera por 2 respostas, se uma maioria não for atingida, é dado um timeout passado 5s), binas não tem que esperar pela resposta de todas as replicas, se uma operação `getBalanceAsync` assincrona é executada com sucesso (`station` executa `getBalance`), o handler de `getBalanceAsync` é chamado para tratar a resposta obtida, esta handler trata o processamento da resposta, aumentando assim o número de respostas recebidas por binas, o `cid` permite identificar qual a estação de onde veio a réplica na resposta, e o `request_id` identificador do pedido da resposta. De notar que a thread onde corre o nosso gestor de réplicas binas dorme enquanto não for obtido um quórum, ou seja receber  $N/2+1$  respostas, mas poderia estar a realizar trabalho, como as chamadas foram assincronas, esperamos apenas pela maioria do quorum para avançar. Ao ser atingida essa maioria avançamos no algoritmo quórum consensus, iteramos pelas respostas obtidas e escolhemos aquela que tem o maior número de sequência (De notar que apesar de podermos avançar para a escolha da resposta com maior número de sequência sem ter obtido todas as respostas das stations, estas podem chegar a qualquer altura e serão tratadas pela função callback que definimos de forma a respeitar a consistência sequencial das mensagens)

## Troca de mensagens do protocolo em pedidos de Escrita (operações incCredit, returnBina)

Todos os pedidos de escrita no protocolo (no binas são as operações incCredit / returnBina) envolvem um pedido de leitura, para descobrir qual a informação replicada do user que queremos aceder, com maior número de sequência, depois da leitura ser efectuada é passado à chamada assíncrona setBalanceAsync: o response\_number (identificador de um pedido de escrita), informação replicada do utilizador (com sequência incrementada), email identificador. A função do servidor réplica station setBalance actualiza o \_user identificado pelo email. A chamada com callback assíncrona permite ao cliente não bloquear enquanto as stations realizam o pedido de escrita, o cliente entra num ciclo, e espera sobre um timeout de 5 segundos por um  $N/2+1$  respostas de reconhecimento que a escrita foi executada com sucesso, os handlers olham para dentro da mensagem de resposta após o seu return pelas réplicas station-ws tratando do processamento da resposta, aumentando o número de respostas recebidas, se a maioria for atingida, a escrita foi um sucesso, caso contrário não foi atingindo nenhum consenso.