

Relatório

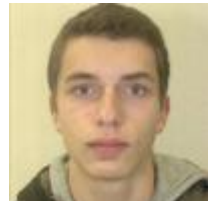
Tolerância a Faltas

Grupo 41

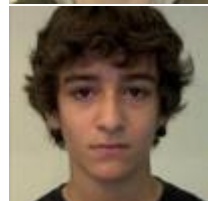


TÉCNICO
LISBOA

Francisco Sousa nº82037



José Canana nº82039



Inês Vilhena nº84593



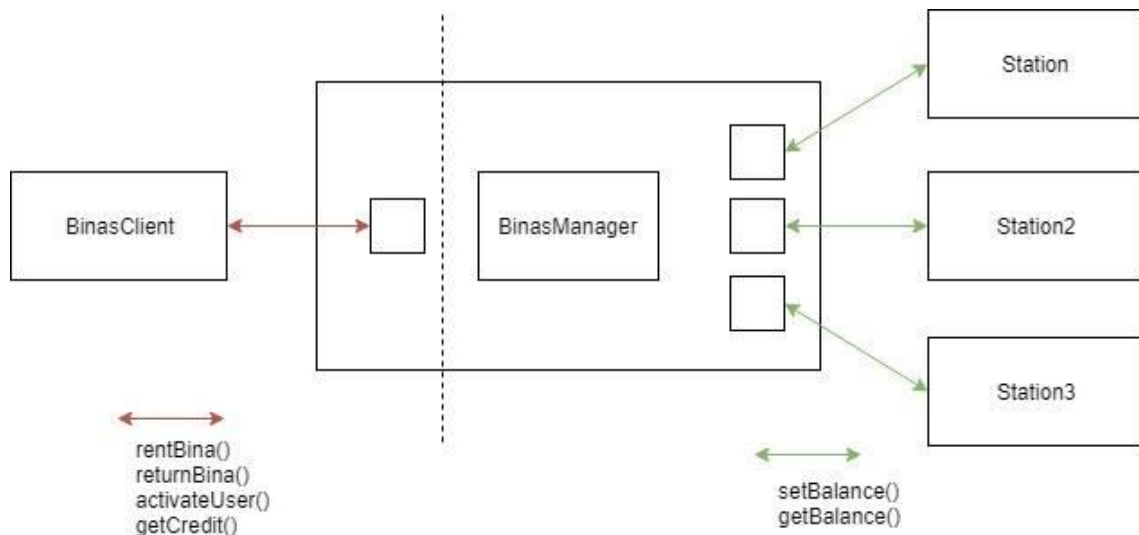
Replicação

A replicação do sistema BINAS é feita a partir da transmissão de informação sobre os utilizadores, nomeadamente o seu saldo, para cada uma das estações conectadas ao sistema. Dessa forma é garantida a informação tanto nas falhas de uma estação ou do próprio servidor Binas.

O protocolo Quorum Consensus permite duas operações:

- ➔ Read(), que utiliza o método `getBalanceAsync()` da Station para obter a informação;
- ➔ Write(), que utiliza o Read(), seguido do método `setBalanceAsync()` da Station para escrever a informação atualizada;

No âmbito do nosso projeto, os reads e os writes são realizados pelo Binas-cliente de forma assíncrona quando são feitos pedidos relacionados com o seu saldo.



Leitura de Registos (quorumRead)

- 1) Front-end (Binas_ws) envia a todos os gestores de replica (stations) um pedido de leitura (`getBalanceAsync()`);

Gestor de Replica → recebe pedido e devolve → BalanceView <saldo,tag>

- 2) Front-end (Binas_ws) espera por Q respostas, onde, Q é uma maioria das replicas;

```
Q = Math.round((sts/2)+1);
```

- 3) `quorumRead()` retorna uma BalanceView, que representa o registo mais atualizado de um determinado utilizador(email). O registo mais recente é determinado por a maior de todas as tags recebidas das stations (MAXTAG);
Nota: mesmo que a maioria devolva uma tag diferente da MAXTAG é o maior registo que é devolvido.

Escrita de Registos (quorumWrite)

- 1) Front-end (Binas_ws) começa por efetuar um quorumRead() donde retira a MAXTAG;
- 2) É calculada uma new_tag onde: **New_tag = maxtag + 1 ;**
- 3) Front-END (Binas_ws) envia setBalanceAsync(email,balanceView) a todas as replicas, atualizando todas as estações com os novos valores.
Replica: se new_tag for > que a tag guardada na replica esta altera o seu valor e a sua tag para o recebido do pedido e devolve um ack.
- 4) Front-END espera por Q acks.
- 5) Front-END (Binas_ws) retorna aos clients (Binas_client).

Polling

Esta implementação consiste no envio de pedidos a todos os gestores de réplica e na espera activa por uma resposta da maioria das réplicas, estando constantemente a verificar se o número mínimo de respostas foi recebido.

Callback

Analogamente ao polling, esta implementação envia o pedido a todos os gestores de réplica mas não fica em espera activa. Apos receber a primeira resposta cria uma nova thread designada a analisar as mesmas. Desta forma é possível continuar a enviar pedidos ao mesmo tempo que são analisadas as respostas já recebidas.

Polling vs Callback

No âmbito do nosso projeto decidimos implementar chamadas assíncronas do tipo polling visto apenas existir um único cliente (BinasManager) e portanto não existirem vários pedidos concorrentes que interfiram uns com os outros. A complexidade do Callback não se justifica para esta implementação.