Nesta entrega de projeto, assumimos que existem três réplicas Station, e que não pode haver mais que uma falha nas réplicas Station em simultâneo. Além disso, o servidor Binas pode falhar ocasionalmente, o que pode causar a perda ou corrupção na informação dos utilizadores. Para resolver os problemas que possam surgir utilizamos uma abordagem semelhante ao quorum consensos, com algumas alterações com objetivo de melhorar a eficiência no nosso projeto.

Uma das alterações realizadas foi na estrutura da tag. No nosso projeto, decidimos que a tag seria um número sequencial, que iria ser incrementado apenas quando é realizado a operação *setBalance,* ou seja, quando é realizada a operação de escrita. Além disso decidimos, também, guardar a tag mais recente no *Front End* do utilizador. Isso é possível no nosso projeto porque existe apenas um cliente, pelo que não é necessário guardar o clientId. O que por sua vez também torna possível o armazenamento da tag no *Front End* no utilizador. Logo, a leitura realizada antes de se fazer uma escrita com o objetivo de procurar qual a tag mais recente, também passa a ser desnecessária.

Outra otimização realizada consiste no armazenamento da tag respetiva à ultima escrita (*setBalance)* de cada utilizador.

Guardar tag mais recente – explicação e raciocínio. Não temos de fazer leitura, porque só há um cliente.

Considerámos a tag como uma sequence number, visto que só temos um cliente. Logo não é preciso .

Guardar tag da última escrita referente a cada utilizador no binas. Quando lê compara tag do utilizador com tag da útima escrita, se for tag de escrita for menor vai ler às stations. Senão assume que o saldo é o mais atual.

Quando faz por exemplo returnBina e GetBina tendo apagado todos os registos dos utilizadores, cria um novo utilizador no binas com os valores default.

~~Vamos assumir que existe 3 servidores sempre, portanto só pode haver 1 falha.~~