

# Replicação de Quorum (Guião 2)

**Eduardo Silva, Ivo Baixo, José Magalhães e Pedro Oliveira**

{pg47167, pg47271, pg47355, pg47570}@alunos.uminho.pt

Grupo 3

*Tolerância a Faltas, MEI, Universidade do Minho*

18 de março de 2022

## 1. Justificação da necessidade de lock na operação CAS

A operação CAS (*Compare and Set*) para uma chave 'k' pode-se dividir em três partes:

1. Leitura dos valores para a chave 'k' nas diferentes réplicas do *quorum*;
2. Para os nodos com a versão atualizada, é feita uma comparação;
3. Por fim, se as comparações estiverem corretas, procede-se à escrita do novo valor nas réplicas.

Se não existir um *lock* na operação CAS é possível que após a leitura ser feita, outro processo escreva nas réplicas que foram lidas. Assim, o CAS escreveria por cima dessa escrita, ignorando-a por completo.

Com a utilização de *locks* garantimos que as operações do CAS são feitas atomicamente, isto é, sem processos no meio das mesmas e, por isso, a escrita e leitura dos valores é feita sempre com uma ordem correta.

Como expectável, quando retirados os *locks* nas operações de CAS, leitura e escrita, o Maelstrom dá erro na execução, enquanto que, com a presença dos *locks*, toda a execução corre bem, sem qualquer erro.

Acrescenta-se em forma de nota que, anexados a este documento se encontram dois ficheiros comprimidos, correspondendo estes à solução com e sem *locks*, representando assim o contra-exemplo obtido com o Maelstrom, conforme pretendido pelo enunciado.