# Tolerância a Faltas

Trabalho Prático

2020/2021

## Resumo

O trabalho consiste na implementação em Java, usando o protocolo de comunicação em grupo Spread, de um serviço tolerante a faltas. O código fonte deverá ser entregue no *e-Learning* acompanhado de um relatório. Cada grupo de trabalho é constituído no máximo por cinco elementos. A data limite é 11 de junho de 2021.

## Descrição

Pretende-se concretizar o sistema informático de um banco que gere contas individuais e oferece as seguintes operações:

- Depósito ou levantamento numa conta existente, fornecendo uma descrição e uma quantia;
- Transferência entre duas contas, fornecendo também uma descrição e uma quantia.
- Extrato de uma conta, que deve incluir para cada um dos últimos $N$ movimentos:
  - descritivo;
  - data e hora de lançamento no servidor;
  - valor do movimento;
  - saldo após o movimento.
- Crédito de juros de $X\%$ dos saldos atuais das contas.

Um levantamento ou transferência só é aceite se a conta origem tem um saldo suficiente. Não é necessário considerar operações para criar ou fechar contas, devendo assumir-se um conjunto de $C$ contas previamente existentes. $N$, $X$ e $C$ são constantes definidas no código.

## Requisitos

A resolução do trabalho deve incluir:

- Par cliente/servidor da interface descrita, replicado para tolerância a faltas usando o protocolo de replicação passiva. Exceto na valorização 5(b), pode assumir-se um modelo VS não particionável.

- Transferência de estado para permitir a reposição em funcionamento de servidores sem interrupção do serviço.
- Interface do utilizador mínima para teste do serviço. Não se consideram questões de segurança, pelo que não é preciso validar a identidade dos utilizadores.
- Um relatório sucinto que justifique as decisões principais.

# Valorização

O trabalho é valorizado se:

1. Utilizar corretamente técnicas genéricas de sistemas distribuídos, tais como serialização, programação concorrente e arquitetura cliente/servidor.
2. Separar claramente o código entre *middleware* genérico de replicação e aplicação.
3. Permitir o tratamento de varias operações concorrentemente.
4. Fizer uma avaliação experimental de desempenho.
5. Suportar um dos seguintes cenários adicionais:
   (a) Armazenamento persistente dos dados numa base de dados *embedded* (e.g., Derby ou o HSQLDB).
   (b) Modelo de grupos particionáveis (EVS) com o Spread.
6. Permitir transferência de estado incremental.

Identifique claramente no relatório quais as valorizações que completou.