# Ficha Exercício 2: Processamento Concorrente com Sincronização de Dados Acadêmicos

## Objetivo

Desenvolver um sistema concorrente que processa informações acadêmicas e atualiza uma estrutura de dados compartilhada para registrar a média das notas dos alunos da Universidade do Mindelo. Neste cenário, várias threads acessam e atualizam um mapa compartilhado que armazena a média de notas dos alunos. Será necessário sincronizar o acesso ao mapa para evitar condições de corrida (Race Condition).

## Cenário

A Universidade do Mindelo possui um sistema de notas dos alunos, e periodicamente é necessário calcular e atualizar a média das notas de cada aluno em um registro central. Como diferentes arquivos de notas podem ser processados simultaneamente, as threads precisam de acesso coordenado ao mapa compartilhado que armazena a média de cada aluno

## Estrutura dos Arquivos

1. **Arquivo** `notas1.txt`: Contém notas de alunos em várias disciplinas.
2. **Arquivo** `notas2.txt`: Outro arquivo com notas de diferentes disciplinas para os mesmos alunos.
3. **Arquivo** `notas3.txt`: Mais um arquivo com notas adicionais para os alunos.

Cada linha desses arquivos tem o formato:

***ID:*** *<id do estudante>,* ***Nome:*** *<nome do estudante>,* ***Disciplina:*** *<nome da disciplina>,* ***Nota:*** *<nota>*

## Requisitos

1. **Classe** `RegistroDeNotas`
   - 1.1. Responsável por manter um mapa compartilhado de médias de notas dos alunos.
   - 1.2. Atributo `Map<Integer, Double> medias`, onde a chave é o ID do aluno e o valor é a média das notas.

1.3. Método `atualizarMedia(int id, double novaNota)`: Recebe uma nova nota e atualiza a média do aluno de forma sincronizada..

2. **Classe `ProcessadorDeNotas`**

2.1. Implementa `Runnable` e simula o processamento de um arquivo de notas.

2.2. Para cada linha, extrai a nota do aluno e usa `atualizarMedia` para atualizar a média no registro central.

2.3. Verifica periodicamente se foi interrompido, finalizando o processamento de forma controlada em caso de interrupção.

3. Classe Principal (**`Main`**)

3.1. Cria uma instância de `RegistroDeNotas`.

3.2. Inicia três threads de `ProcessadorDeNotas`, cada uma processando um arquivo de notas (`notas1.txt`, `notas2.txt`, `notas3.txt`).

3.3. Após 10 segundos, interrompe todas as threads para simular uma parada do sistema.

3.4. Usa `join()` para aguardar a finalização de todas as threads e, em seguida, exibe o conteúdo do mapa `medias`, que deve conter a média final de cada aluno.

**Resultados Esperados**

Ao executar o programa, ele deve exibir:

- As mensagens de processamento de cada linha.
- Mensagens de interrupção das threads após 10 segundos.
- A média final de cada aluno no mapa compartilhado.