# LABORATÓRIO 1: THREADS E MONITORES

INE5645 | Descrição | Implementação | Apresentação | Dúvidas

## Descrição

Nesta atividade de laboratório você deve implementar um sistema Escritor/Leitor.

---

## Implementação

Existem 240 threads, sendo 120 Escritor e 120 Leitores, e um objeto `Buffer` de tamanho 1 como dois métodos:

```
class Buffer {
   synchronized Escrever(int i) {...}
   synchronized int Ler() {...}
}
```

As threads Escritor devem ser escalonadas usando `ScheduledThreadPool`, ou seja, todas essas 120 threads devem ser controladas por esse escalonador. Esse escalonador deve liberar uma thread Escritor a cada 0,1 milissegundo (utilize `scheduleAtFixedRate`) que irá escrever um valor inteiro incremental no Buffer.

As 120 threads Leitor devem ser controladas por um `ExecutorService` usando `FixedThreadPool` que deve permitir até 4 threads Leitor em estado de pronto durante a execução do código. As threads Leitor em estado de pronto devem tentar ler o valor escrito no objeto Buffer. Verifique a quantidade de threads Leitor que leram o mesmo valor inteiro.

Na segunda parte do trabalho, a primeira thread Leitor que conseguir ler o valor escrito deve setar a variável do Buffer para zero. Para que as outras thread não leiam zero, você deve utilizar monitor juntamente com `wait/notify`. Isto é, quando uma thread Leitor verificar que o buffer está vazio, este invoca o método `wait()`. E quando uma thread Escritor tiver escrito um dado, este invoca o método `notify()`. Verifique o comportamento do programa em relação ao anterior.

Por fim, altere o programa para que a cada 4 thread Leitor leia o mesmo valor escrito por uma thread Escritor. Dica, implemente um barreira usando o método `wait()`. Quando 4 threads Leitor tiver invocado esse método uma thread Escritor deve invocar o método `notifyAll()` para que as 4 threads leiam o mesmo valor.

---

## Apresentação

A atividade deve ser desenvolvida **em duplas**. O programa deve ser apresentado ao professor no laboratório **até o dia 29/03**. Os dois componentes do grupo devem estar presentes. Será verificado o funcionamento do programa e em seguida os alunos devem responder a questões sobre a forma como foram utilizados *threads* e monitores no programa.

Trabalhos entregues com atraso terão desconto automático de 2 pontos por semana de atraso. Após a segunda semana (14 dias após o fim do prazo original), o trabalho não mais será aceito, ou seja, terá nota zero.

Podem ser atribuídas notas diferentes aos alunos de um grupo, dependendo das respostas às perguntas sobre o código do programa efetuadas pelo professor. Caso um dos alunos não esteja presente ou demonstre não conhecer o código do programa, será atribuída nota zero à atividade. Em caso de **cópia** do código de outro grupo, ambos terão nota igual a **zero**.