

# Laboratório 11

## Pool de threads e computação assíncrona em Java

Programação Concorrente (ICP-361)  
Profa. Silvana Rossetto

<sup>1</sup>Instituto de Computação/CCMN/UFRJ

### Introdução

O objetivo deste Laboratório é praticar a implementação e uso de um **pool de threads** e de outros recursos de concorrência disponíveis na linguagem Java (veja o pacote `java.util.concurrent`).

### Atividade 1

**Objetivo:** Compreender como pode ser feita uma implementação de um *pool de threads* em Java.

#### Roteiro:

1. Edite o arquivo **MyPool.java** e compreenda o que faz e como funciona (*acompanhe explicação da professora*).
2. Documente a classe `FilaTarefas`.
3. Execute o programa (**várias vezes**) alterando o tamanho do pool de threads e o número de tarefas executadas. Avalie se o programa funciona como esperado.
4. **Acrescente um novo tipo de tarefa para ser executada no mesmo pool de threads:** dado um número inteiro positivo verifica se ele é primo, como no código mostrado abaixo. Complete a implementação da classe `Primo` e descomente as linhas 94 e 95 do programa (comente as duas linhas anteriores).

```
//funcao para determinar se um numero é primo
int ehPrimo(long unsigned int n) {
    int i;
    if(n<=1) return 0;
    if(n==2) return 1;
    if(n%2==0) return 0;
    for(i=3; i< sqrt(n)+1; i+=2) {
        if(n%i==0) return 0;
    }
    return 1;
}
```

### Atividade 2

**Objetivo:** Usar uma implementação de *pool de threads* oferecida por Java.

#### Roteiro:

1. Edite o arquivo **HelloPool.java** e compreenda o que faz e como funciona (*acompanhe explicação da professora*).
2. Experimente a aplicação.

3. Agora edite o arquivo **AnotherHelloPool.java** e o compare com o programa anterior.
4. Altere o programa **AnotherHelloPool.java** para que as threads incrementem uma variável compartilhada e exiba o valor final dessa variável no final do programa.

### Atividade 3

**Objetivo:** Avaliar e usar uma implementação de computação assíncrona em Java.

**Roteiro:**

1. Edite o arquivo **FuturePool.java** e compreenda o que faz e como funciona (*acompanhe explicação da professora*).
2. **Acrescente um novo tipo de tarefa para ser executada:** dado um número inteiro positivo verifica se ele é primo, como na Atividade 1.
3. Complete o programa fazendo com que ele retorne a quantidade de números primos no intervalo de 1 a N (podendo N ser um valor bastante grande) usando computação assíncrona com futuros.

**Entrega do laboratório (Exercício 4)** Disponibilize o código implementado na **Atividade 1, item 4** e na **Atividade 3, itens 2 e 3** em um ambiente de acesso remoto (GitHub ou GitLab). Use o formulário de entrega desse laboratório para enviar o link do repositório do código implementado e acrescentar comentários ou dúvidas.