

---

# Processamento Paralelo com CyclicBarrier - Sincronizando Tarefas em um Ponto Comum



STATUS

DESENVOLVIDA

## Objetivo do Projeto



Este projeto implementa um sistema de busca paralela em uma matriz de inteiros utilizando a classe CyclicBarrier do Java. O objetivo é contar a ocorrência de um número específico dentro da matriz e agregar os resultados de forma sincronizada.

### Estrutura do Projeto

O código é composto pelas seguintes classes:

MatrixMock: Gera uma matriz de números aleatórios e contabiliza a ocorrência de um número alvo. Results: Armazena os resultados parciais de cada thread. Searcher: Implementa Runnable para buscar um número na matriz e armazenar os resultados. Grouper: Implementa Runnable para processar e exibir o resultado final após a sincronização das threads. Main: Configura e executa as threads, gerenciando a sincronização com CyclicBarrier.



## Cenários teste



### Configuração do Cenário de Teste

Matriz com 10.000 linhas e 1.000 colunas.

O número a ser buscado é 5. 5 threads processando 2.000 linhas cada.

CyclicBarrier sincroniza as threads e executa o Grouper após a finalização.

Fluxo de Execução A matriz é gerada aleatoriamente. As threads Searcher iniciam e processam segmentos da matriz. Cada Searcher armazena o resultado parcial e chama barrier.await(). Quando todas as threads atingem o barrier, Grouper soma os resultados. O resultado final é exibido.

## Conclusão:



O uso de CyclicBarrier garante sincronização eficiente entre as threads. Cada thread processa uma parte independente da matriz, evitando condições de corrida. O Grouper é executado apenas quando todas as threads terminam, garantindo um resultado preciso.

## Desenvolvedor



Marcio Fonseca



=====