

# Simulação de Tarefas de Busca com Threads - Agrupando Threads em um Grupo



STATUS

DESENVOLVIDA

## Objetivo do Projeto



Este documento descreve os testes realizados no código que implementa uma simulação de tarefas de busca (SearchTask). As tarefas são executadas por múltiplas threads, utilizando ThreadGroup para gerenciar e monitorar seu estado.

### Visão Geral

O código implementa uma classe SearchTask que simula uma tarefa com tempo de execução aleatório. O resultado da execução é armazenado em um objeto Result. Um grupo de threads (ThreadGroup) gerencia as threads criadas e monitora seu estado.

### Objetivos dos Testes:

Avaliar o comportamento das threads sob diferentes condições. Analisar o impacto do tempo de espera entre a criação das threads. Verificar a capacidade do ThreadGroup de gerenciar e monitorar as threads.



## Cenários teste



### Cenário 1: Número de Threads e Tempo de Espera Padrão

Configuração: 5 threads, tempo de espera de 1 segundo entre a criação de cada thread.

Resultado: As threads são criadas e executadas com sucesso. Algumas threads podem ser interrompidas antes de concluir a tarefa devido ao tempo de execução aleatório.

=====

### Cenário 2: Aumento do Número de Threads

Configuração: 10 threads, tempo de espera de 1 segundo entre a criação de cada thread.

Resultado: Mais threads são criadas e o tempo total de execução aumenta. Algumas threads são interrompidas antes de finalizar.

=====

### Cenário 3: Redução do Tempo de Espera entre Threads

Configuração: 5 threads, tempo de espera de 0.5 segundos entre a criação de cada thread.

Resultado: As threads são criadas mais rapidamente, aumentando a concorrência. Mais threads são interrompidas antes de concluir a execução.

=====

### Cenário 4: Aumento do Tempo de Espera entre Threads

Configuração: 5 threads, tempo de espera de 2 segundos entre a criação de cada thread.

Resultado: As threads têm mais tempo para concluir suas tarefas antes de serem interrompidas. Mais threads conseguem finalizar a execução.

## Conclusão:



Os testes demonstraram que:

O tempo de espera entre a criação das threads afeta diretamente o grau de concorrência. O ThreadGroup permite monitorar e gerenciar eficientemente o estado das threads. O tempo aleatório de execução pode resultar na interrupção de algumas threads antes da conclusão.

Esses experimentos confirmam a importância do controle de concorrência e gerenciamento de threads para sistemas que dependem de execução paralela

## Desenvolvedor



Marcio Fonseca



=====