

---

# Uso de ThreadFactory vs. Criação Manual de Threads - Criando Threads através de uma Fábrica



STATUS

DESENVOLVIDA

## Objetivo do Projeto



Este relatório documenta os testes realizados para comparar dois cenários de criação de threads em Java: um sem o uso de ThreadFactory e outro utilizando uma fábrica de threads personalizada. O objetivo é analisar a diferença entre as abordagens em termos de estrutura, controle e monitoramento das threads criadas.



## Cenários teste



### Cenário 1: Sem Uso de ThreadFactory

#### Descrição

Neste cenário, as threads são criadas manualmente sem o uso de ThreadFactory. Isso permite uma abordagem mais simples e direta, sem o registro de estatísticas ou controle sobre a criação das threads.

#### Resultado Esperado

As threads são criadas e executam a tarefa normalmente. Nenhuma estatística de criação é registrada.

#### Análise

Apropriado para cenários simples onde não há necessidade de rastreamento ou controle adicional das threads. Não permite coleta de estatísticas sobre a criação das threads.

=====

### Cenário 2: Com Uso de ThreadFactory

## Descrição

Neste cenário, uma fábrica de threads (MyThreadFactory) é utilizada para gerenciar a criação das threads. Isso permite armazenar estatísticas sobre cada thread criada, incluindo ID, nome e horário de criação.

## Resultado Esperado

As threads são criadas e iniciadas com a fábrica de threads. Estatísticas sobre cada thread são registradas e exibidas.

## Análise

Permite monitoramento da criação das threads, fornecendo informações úteis para depuração. Ideal para cenários que exigem controle sobre a gestão de threads.

## Conclusão:



### Conclusão

Cenário 1 (Sem ThreadFactory): Recomendado para aplicações simples onde não há necessidade de rastreamento ou controle sobre a criação das threads. Cenário 2 (Com ThreadFactory): Indicado para cenários mais complexos, onde é essencial manter um controle detalhado sobre a criação e execução das threads.

O uso de ThreadFactory é uma abordagem recomendada para sistemas que necessitam de um gerenciamento mais eficiente e estruturado de threads, permitindo um melhor monitoramento e otimização da execução das tarefas em paralelo.

## Desenvolvedor



Marcio Fonseca



=====