

Relatório: Executor de Tarefas com Pool de Tamanho Fixo

1. Objetivo

Este exemplo demonstra o uso de um executor com pool de threads de tamanho fixo em Java para gerenciar a execução de tarefas. Utilizando `Executors.newFixedThreadPool`, o sistema garante que, no máximo, 5 threads sejam ativas simultaneamente, otimizando o uso de recursos e facilitando o controle do fluxo de execução das tarefas.

2. Estrutura do Código

- **Servidor:**
 - Gerencia a execução das tarefas utilizando um `ThreadPoolExecutor` com um pool fixo de 5 threads.
 - O método `executarTarefa` envia tarefas para o executor e exibe estatísticas do pool, como o tamanho do pool, o número de threads ativas, o total de tarefas e as tarefas concluídas.
 - O método `encerrarServidor` encerra o executor.
- **Tarefa:**
 - Representa uma tarefa a ser executada.
 - Cada tarefa registra sua data de criação, simula um processamento com duração aleatória e exibe mensagens indicando o início e a finalização.
- **Principal:**
 - Inicializa o Servidor e envia 100 tarefas para execução.
 - Após enviar todas as tarefas, o servidor é encerrado.

3. Fluxo de Execução

1. A classe `Principal` cria uma instância do `Servidor`, que configura um pool fixo com 5 threads.
2. Um loop envia 100 instâncias de `Tarefa` para o servidor.
3. Para cada tarefa, o servidor executa a tarefa e imprime informações sobre o estado do pool de threads.
4. Após enviar todas as tarefas, o método `encerrarServidor` é chamado para desligar o executor.

4. Exemplo de Execução

Saída do Console:

```
Servidor: Uma nova tarefa chegou  
Servidor: Tamanho do Pool: 5  
Servidor: Número de Threads Ativas: 5
```

```
Servidor: Total de Tarefas: 100
Servidor: Tarefas Concluídas: 0
Thread-1: Tarefa Tarefa 0: Criada em: Sat Feb 10 15:30:12 BRT 2025
Thread-1: Tarefa Tarefa 0: Iniciada em: Sat Feb 10 15:30:12 BRT 2025
Thread-1: Tarefa Tarefa 0: Executando por 3 segundos
...
Thread-3: Tarefa Tarefa 0: Finalizada em: Sat Feb 10 15:30:15 BRT 2025
...
```

5. Conclusão

O uso de um pool de threads de tamanho fixo permite gerenciar de forma eficiente a execução de múltiplas tarefas concorrentes, otimizando a utilização dos recursos do sistema e facilitando o monitoramento do processamento. Essa abordagem é ideal para aplicações que requerem uma execução controlada e escalável de tarefas, garantindo desempenho consistente mesmo sob alta carga.