

---

# Carregamento de Conexões e Fontes de Dados - Aguardando a Finalização de uma Thread



STATUS

DESENVOLVIDA

## Objetivo do Projeto



Este projeto implementa um sistema de carregamento de conexões de rede e fontes de dados utilizando Java e Threads. O código simula um ambiente onde diferentes processos são executados de forma concorrente, permitindo testar o comportamento das threads em diferentes cenários.

### Estrutura do Projeto

O projeto contém as seguintes classes: CarregadorConexoesRede: Implementa a interface Runnable e simula o carregamento de conexões de rede, demorando 4 segundos para ser concluído. CarregadorFontesDados: Implementa a interface Runnable e simula o carregamento de fontes de dados, demorando 6 segundos para ser concluído. Main: Inicia ambas as threads e aguarda a conclusão completa das execuções antes de exibir uma mensagem final. Main2: Similar à classe Main, mas aguarda apenas 1 segundo pela conclusão da thread de fontes de dados e indefinidamente pela thread de conexões de rede.



## Cenários teste



Foram executados três cenários principais para testar o comportamento das threads:

### Cenário 1: Execução Completa das Threads

Descrição: Ambas as threads (ThreadFonteDados e ThreadConexaoRede) são executadas até o fim.

### Resultado Esperado:

A thread de fontes de dados termina após 6 segundos. A thread de conexões de rede termina após 4 segundos. A mensagem final "Main: Configuração carregada" é exibida.

=====

## Cenário 2: Timeout na Thread de Fontes de Dados

Descrição: A thread de fontes de dados é interrompida após 1 segundo de espera (thread1.join(1000)), enquanto a thread de conexões de rede é executada normalmente.

Resultado Esperado:

A thread de fontes de dados é interrompida antes de terminar. A thread de conexões de rede termina após 4 segundos. A mensagem final "Main: Configuração carregada" é exibida.

=====

## Cenário 3: Interrupção de uma Thread

Descrição: Uma das threads é interrompida manualmente durante a execução.

Resultado Esperado:

A thread interrompida exibe o erro InterruptedException. A outra thread continua sua execução normalmente. A mensagem final "Main: Configuração carregada" é exibida.

## Conclusão:



O projeto permite entender a gestão de threads e concorrência em Java, simulando diferentes cenários de carregamento de dados e conexões de rede. As classes são bem documentadas e podem ser expandidas para casos mais complexos, como cancelamento dinâmico de threads ou implementação de Executors para melhor gestão do ciclo de vida das threads.

## Desenvolvedor



Marcio Fonseca



=====