

# Relatório: Sincronização de Múltiplos Eventos com `CountDownLatch`

## 1. Objetivo

Este exemplo demonstra o uso de `CountDownLatch` em Java para coordenar a execução de uma videoconferência. O sistema aguarda a chegada de um número específico de participantes antes de iniciar a reunião.

## 2. Estrutura do Código

- **Videoconferencia:**  
Representa a reunião e gerencia a chegada dos participantes. Utiliza `CountDownLatch` para bloquear o início da conferência até que todos tenham chegado.
- **Participante:**  
Representa um usuário que entra na conferência após um tempo aleatório. Quando chega, reduz o contador do `CountDownLatch`.
- **Principal:**  
Inicializa a videoconferência e cria os participantes como threads independentes, simulando o comportamento real de uma chamada de vídeo.

## 3. Fluxo de Execução

1. A classe `Principal` cria uma instância de `Videoconferencia`, esperando 10 participantes.
2. Inicia uma thread para executar a videoconferência, que aguarda até que todos cheguem (`await()`).
3. São criadas e iniciadas 10 threads de `Participante`, cada uma chegando após um tempo aleatório.
4. Quando um participante chega, chama `chegar()`, que reduz o contador do `CountDownLatch`.
5. Quando o contador chega a zero, a conferência inicia.

## 4. Exemplo de Execução

Saída do Console:

```
Videoconferencia: Inicialização - 10 participantes esperados.
```

```
Participante 3 chegou.
```

```
Videoconferencia: Aguardando 9 participantes.
```

```
Participante 7 chegou.
```

```
Videoconferencia: Aguardando 8 participantes.
```

```
...
```

Participante 0 chegou.

Videoconferencia: Aguardando 0 participantes.

Videoconferencia: Todos os participantes chegaram.

Videoconferencia: Vamos começar...

## 5. Conclusão

Este exemplo mostra como `CountDownLatch` pode ser usado para sincronizar múltiplas threads, garantindo que um evento (início da conferência) só ocorra após a ocorrência de outros eventos (chegada dos participantes). Essa técnica é útil em cenários que exigem coordenação entre tarefas concorrentes.