



## 07. Entendendo Deadlock

📅 Date	@29/08/2022
▼ Categoria	Java
▼ Curso	Curso Threads em Java 1: programação paralela

### Tópicos

1. Entendendo Deadlock
2. Condição para deadlock
3. Quando um deadlock acontece
4. O que aprendemos?
5. Como continuar?

### Condição para deadlock

Qual das condições abaixo são necessárias para ocorrer um deadlock: Uma thread segura um recurso e aguarda por outro.

### Quando um deadlock acontece

O que acontece em um software quando ocorre um deadlock?

Somente as threads envolvidas no deadlock ficam paradas.

### O que aprendemos?

- Podemos usar um bloco `synchronized` dentro do outro.
- Ao obter os recursos, pode acontecer um impasse (*interbloqueio*).
- O impasse também é chamado **deadlock**.

- Podemos evitar o impasse obtendo as chaves dos objetos na mesma ordem, em todos os threads.

## Como continuar?

Neste treinamento falamos sobre os conceitos principais das threads, como ordem de execução, sincronização, `wait` e `notify`, `daemon`, `prioridade` e `deadlock`. Já temos uma boa base mas podemos aprender ainda mais.

Com o lançamento do Java 5, entrou um pacote específico na API Java para melhorar e simplificar a programação concorrente. No segundo curso sobre threads, focamos justamente neste pacote e apresentando os recursos dentro de um projeto *Cliente-Servidor*. É um *curso avançado*, no qual é criado uma aplicação cliente-servidor. Se quiser aprender mais, convido você a participar do curso abaixo:

Curso Online Threads em Java 2: programação concorrente avançada | Alura

Aprenda conceitos avançados da programação concorrente em Java e aprenda programar com Futures, ExecutorService, Callable e estruturas de dados como BlockingQueue.

 <https://cursos.alura.com.br/course/threads-java-2>



Curso Online

**Threads 2: Programação  
concorrente avançada**