



## 03. Melhorando o cliente

📅 Date	@30/08/2022
▼ Categoria	Java
▼ Curso	Threads em Java 2: programação concorrente avançada

### Tópicos

1. Melhorando o cliente
2. Devolvendo informação
3. Melhorando o cliente
4. E agora? Como resolver
5. Juntando threads
6. Neste contexto, pool faz sentido?
7. Mãos à obra: Enviando e recebendo comandos do servidor
8. Mãos à obra: Confirmando o recebimento dos comandos
9. Para saber mais: Esperando um tempo limitado
10. Para saber mais: Threads e Java 8
11. Um pouco sobre o projeto
12. Download do código fonte
13. (Opcional) Interface Gráfica

### Melhorando o cliente

## O que aprendemos?

- Usar o `Runnable` através de classes anônimas.
- O método `thread.join()` faz com que a thread que executa espere até o outro acabar.
- só faz sentido usar um pool de threads quando realmente queremos reaproveitar uma thread

## E agora? Como resolver

Podemos deixar a Thread `main` esperar que só deve voltar a ser executada quando a `threadEnviaComando` terminar. Fazemos isso através do método `join()`:

```
threadRecebeResposta.start();
threadEnviaComando.start();
// faz que a thread que executa essa linha fica esperando (no nosso caso main).// só volta
a ser executada quando `threadEnviaComando` terminar.
threadEnviaComando.join();
socket.close();
```

## Juntando threads

O que podemos afirmar sobre o método `join()` da classe `Thread`?

A) É um método da classe `Thread` com acesso estático ( `static` ).

A) **Errado**: É um método da classe `Thread` com acesso estático ( `static` ).

*O método é da instância.*

B) É para coordenar a execução de duas threads.

B) **Correta**: É para coordenar a execução de duas threads.

*Podemos deixar uma thread esperar a finalização da outra.*

C) Quando uma thread `t2` chama `t1.join()`, significa que `t1` vai esperar `t2` finalizar.

C) **Errado:** Quando uma thread `t2` chama `t1.join()`, significa que `t1` vai esperar `t2` finalizar.

*É justamente contrário.*

D) Quando uma thread `t2` chama `t1.join()`, significa que `t2` vai esperar `t1` finalizar.

D) **Correto:** Quando uma thread `t2` chama `t1.join()`, significa que `t2` vai esperar `t1` finalizar.

`t2` vai se "juntar" ao `t1`, isso é esperar a finalização do `t1`.