



08

O resultado futuro de uma ação!

Quando passamos um `Callable` para o método `submit` de um pool de Threads, não recebemos como resultado o retorno da operação, mas sim um objeto que possui o resultado futuro da operação. É por isso que recebemos como retorno um objeto do tipo `Future`.

Essa instância de `Future` pode ser passada como parâmetro para outros métodos:

```
ExecutorService pool = Executors.newCachedThreadPool();
Future<String> future = pool.submit(new MinhaClasseQueImplementaCallable());
```

Dada uma `Future`, nós obtemos seu resultado através de uma chamada ao método `get`; este sim, bloqueia a execução do nosso código até que o seja retornado o resultado da operação assíncrona:

```
ExecutorService pool = Executors.newCachedThreadPool();
Future<String> future = pool.submit(new MinhaClasseQueImplementaCallable());
String retorno = future.get(); //bloqueia a thread até receber o resultado
System.out.println(retorno);
```

O problema é que a chamada de `future.get` é bloqueante, isto é, se tivermos mais instâncias da classe `Future` teremos que esperar cada uma resolver antes de executar a próxima. Como podemos resolver isso?

Responda

INSERIR CÓDIGO	FORMATAÇÃO
<div></div>	