

Para saber mais: ThreadFactory padrão

No capítulo usamos a nossa própria implementação da interface `ThreadFactory` para personalizar a criação das threads no pool.

Quando *não* usamos uma `ThreadFactory` explicitamente é utilizada a implementação padrão que podemos acessar através:

```
ThreadFactory factory = Executors.defaultThreadFactory();
```

No [javadoc](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/Executors.html#defaultThreadFactory%28%29)

(<https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/Executors.html#defaultThreadFactory%28%29>) desse método consta:

Each new thread is created as a non-daemon thread with priority set to the smaller of Thread.NORM_PRIORITY and the maximum priority permitted in the thread group. New threads have names accessible via Thread.getName() of pool-N-thread-M, where N is the sequence number of this factory, and M is the sequence number of the thread created by this factory.

Ou seja, é criado uma thread que não é *daemon* com uma prioridade mais baixa do que `Thread.NORM_PRIORITY`, além de usar uma nomenclatura padrão.

Mas qual é a utilidade disso, já que temos a nossa própria fábrica que funciona corretamente?

Uma razão para conhecer a fábrica padrão seria aproveitar ela mesmo usando uma fábrica personalizada. Assim seguimos uma implementação correta e só fazemos ajustes nas threads criadas pela fábrica padrão!