

03

Tasks vs. Threads

Dica: começando agora? [Aqui \(https://github.com/alura-cursos/Paralelismo-com-CSharp-e-.NET/archive/d7075d6faf94f0791bc1cfcb435e359696a0aa7.zip\)](https://github.com/alura-cursos/Paralelismo-com-CSharp-e-.NET/archive/d7075d6faf94f0791bc1cfcb435e359696a0aa7.zip) tem o link do projeto completo até a aula anterior.

Do ponto de vista de performance, em um processador quad-core, qual a diferença entre os códigos abaixo?

1.

```
for(int i = 0; i < 100; i++) {  
    var msg = "Thread número " + i;  
    var thread = new Thread(() => Console.WriteLine(msg));  
    thread.Start();  
}
```

2.

```
for(int i = 0; i < 100; i++) {  
    var msg = "Task número " + i;  
    Task.Factory.StartNew(() => Console.WriteLine(msg));  
}
```

Selezione uma alternativa

A

A solução em 2. é mais performática, porque passamos a responsabilidade de provisionamento e gerenciamento das Threads disponíveis para o `TaskScheduler default` usado pela `Task.Factory`, que possui uma inteligência

B

A solução em 2. é mais performática, pois não é usado Threads de forma alguma

C

Nenhuma, uma vez que as tasks criadas em 2. serão executadas em threads separadas, como em 1.

D

A solução em 1. é mais performática, por usarmos diretamente a classe `Thread` em vez de um intermediário