

## SynchronizationContext

Escolha a alternativa que melhor define o comportamento do código abaixo:

```
01 | Task.Factory.StartNew(() =>  
02 | {  
03 |     var context = TaskScheduler.FromCurrentSynchronizationContext();  
04 |     Task.Factory.StartNew(() => CalculaRaiz(100))  
05 |         .ContinueWith(t => btnCalcular.IsEnabled = true, context);  
06 | });
```

Selecione uma alternativa

- A** O botão `btnCalcular` fica *enabled* com sucesso, mesmo que a alteração de sua propriedade seja feita em um contexto diferente da Thread principal
- B** Apesar de capturarmos o contexto na linha 3, o contexto capturado é diferente do contexto da thread principal, então uma exceção do tipo `InvalidOperationException` será lançada na linha 5
- C** O botão `btnCalcular` fica *enabled* com sucesso pois estamos capturando o contexto de execução na linha 3
- D** Uma exceção do tipo `InvalidOperationException` será lançada na linha 3 por estarmos tentando capturar o contexto da thread principal a partir de uma Task