

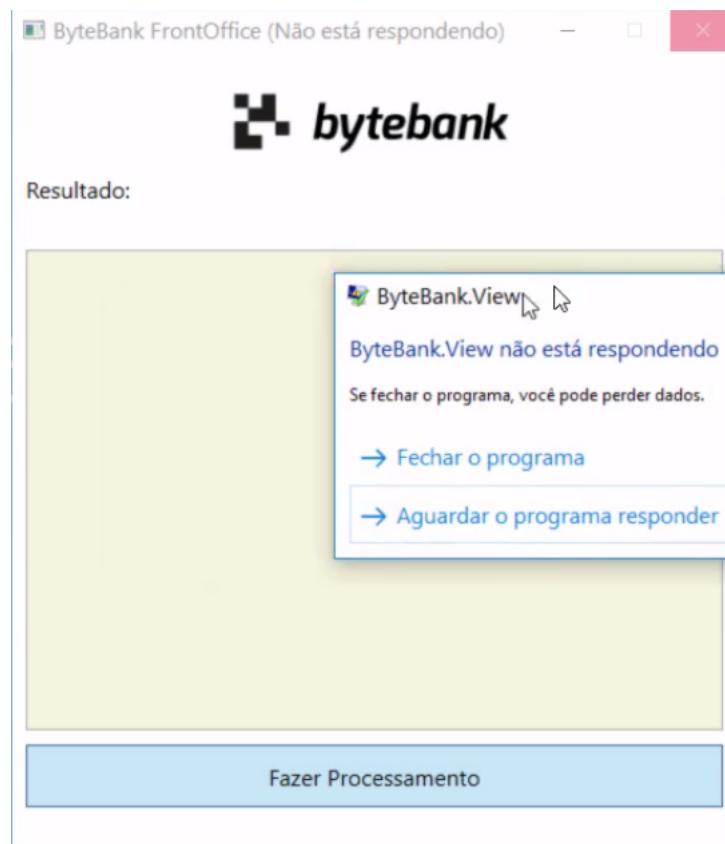
01

## Introdução

### Transcrição

Olá, meu nome é Guilherme, e neste curso iremos aprender **Paralelismo com C# e .NET**, técnicas e recursos do processador. Aprenderemos a identificar se ele possui dois *cores* (que em inglês significa “núcleo”), se é um processador *quad-core*, com quatro *cores* ou mais. Iremos aprender a utilizar todos os *cores* para atingir o objetivo da aplicação de forma mais rápida.

Além disto, aprenderemos formas de tornar a interface gráfica mais amigável ao usuário. Vocês se lembram daquela maneira mais tradicional de desenvolvimento de aplicações? Quando temos uma ação, um processamento muito intenso, e a janela deixa de responder:



Na verdade, o sistema operacional até chega a achar que a aplicação travou, sendo que não é o caso, ainda há algum processamento. Este é o problema encontrado por nosso cliente, ByteBank Front Office, uma startup voltada para transações bancárias. A nossa aplicação, desenvolvida de forma tradicional, faz a consolidação da movimentação diária dos clientes e cálculos financeiros. No entanto, ela atua muito lentamente.

Além disto, a app fica com essa "cara" de estar travada, deixando o usuário sem saber o que fazer e levando-o a clicar no ícone "x" para fechar o programa. A ação pode causar um estado inválido no banco de dados, corrompendo-o, o que resulta na perda de integridade das informações do cliente.

Com técnicas de paralelismo do .NET e do C#, vamos aprender a fazer uma aplicação muito mais amigável, moderna, a partir da qual o usuário fará o processamento, vendo o progresso do que está acontecendo. Obteremos, assim, um ganho na performance, pois utilizaremos vários *cores* do CPU.

Se clicarmos errado em "Fazer processamento", haverá o botão de "Cancelar" logo ao lado, de forma que o banco de dados, a própria aplicação e a integridade do projeto sejam mantidos.

No curso, veremos sobre *Threads*, *Tasks* do .NET, conceitos do *Task Scheduler*, *AsyncAwait* do C#, *Patterns* do .NET, juntamente com todas as classes e bibliotecas.

Esses conhecimentos não valem apenas para aplicações desktop ou com interface gráfica. Podemos utilizar o Paralelismo também ao lado do servidor, para ganho de performance, fazendo-se as operações de forma mais rápida e eficiente. Vamos lá?