

Conclusão

Transcrição

Parabéns, você terminou este curso! Aprendemos a utilizar *threads* no .NET, e que uma aplicação com interface gráfica tem a *thread* GUI (a principal), na qual não se deve realizar tarefas muito pesadas pois isto acaba dando a impressão de que a aplicação do nosso cliente, ByteBank, travou ou não funciona.

Resolvemos também a lentidão da aplicação por meio do uso de *threads*, paralelizando-se a execução das tarefas. Melhor ainda, aprendemos a utilizar tarefas, as *tasks*, que surgiram no .NET 4.0. Deixamos de nos preocupar com a quantidade de *cores* e processadores em uso, delegando estas responsabilidades ao ***Task Scheduler***, já implementado no .NET 4.0. Nosso código está ficando cada vez mais limpo pois agora sabemos usar recursos modernos do C#, como o ***AsyncAwait***. Não precisamos mais focar em recuperarmos o contexto de sincronização, já que se pode solicitar isto ao `TaskScheduler`.

Aprendemos a reportar progressos, a partir de uma barra que mostra visualmente o andamento do processamento, como em aplicações modernas, ou quando fazemos downloads ou esperamos uma ação. Vimos como se implementa o *pattern* `IProgress`, criado no .NET 4.0, e como oferecer ao usuário uma forma amigável de se cancelar uma operação, sem tornar a aplicação em um estado com risco de perda de dados dos clientes ou corrompimento de banco de dados.

Melhoramos a interface para o usuário, ganhamos em *performance*, temos um código mais limpo que dá mais prazer de escrever, pois está muito mais conciso, eficiente e elegante.

Foi um prazer fazer este curso para vocês, vamos continuar estudando e assistindo os vídeos sobre .NET, C# ou qualquer outra linguagem na Alura, para agregarmos conhecimento para nossas carreiras, projetos, e para nossas vidas. Um abraço!