

Reportando com Progress

Transcrição

Resolvemos um problema muito comum que tem a ver com o uso do contexto de uma *thread* para processamento do *report* de um progresso feito por outra *thread*. Trata-se de um padrão comum, porém é muito importante termos noção disto ao criarmos, no construtor, algo que recupera o `TaskScheduler`. Em `Report()` usaremos `Task.Factory.StartNew` a partir de `_taskScheduler`.

Na verdade, o que fizemos foi reescrever e entender como funciona uma classe já existente no .NET. Esta classe se chama `Progress`, com comportamento exatamente igual àquilo que acabamos de escrever. Então, no `MainWindow.xaml.cs` podemos utilizar o `ByteBankProgress` para termos mais opções dentro desta implementação. Se nosso objetivo é apenas atualizar o `PgsProgresso`, usaremos o `Progress` implementado no .NET. Além do nome, tem-se em comum o fato dele ser genérico, recebendo por parâmetro um manipulador que é uma `Action<string>`. É por este motivo que podemos deixá-lo comentado, como se vê abaixo:

```
var inicio = DateTime.Now;

var progress = new Progress<String>(str =>
    PgsProgresso.Value++);
// var byteBankProgress = new ByteBankProgress<String>(str =>
//     PgsProgresso.Value++);

var resultado = await ConsolidarContas(contas, progress);
```

Pressionaremos *"Start"* e faremos o processamento na aplicação, e seu comportamento permanecerá o mesmo. O que criamos em `ByteBankProgress` é o mesmo que foi implementado pela Microsoft em `Progress`. A diferença está em que o primeiro foi criado por nós, portanto, entendemos seu funcionamento - o que é muito legal. Para não reinventarmos a roda, vamos utilizar aquilo que foi implementado pelo .NET.

Ainda há o que melhorar em nossa aplicação: por ora, temos 10 clientes e o processamento acontece de forma rápida, em cerca de 10s. Esse tempo pode aumentar muito, e o usuário pode precisar sair, não podendo esperar ou lembrar que a consolidação não pode ser feita neste momento, pois o fechamento do mês só ocorrerá na semana seguinte... Como quando fazemos um download de um arquivo muito grande que não acaba nunca, ou quando clicamos nele por engano. Precisamos de algo que possibilite o cancelamento da ação.