**Sobre a Barra de Progresso**

**Mário Leite**

...

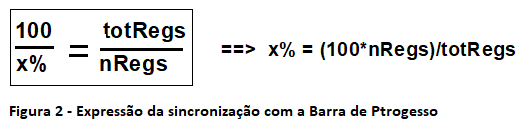
A “Barra de Progresso” (*ProgessBar*) é uma das classes de controles mais interessantes que são utilizados em aplicações Windows Forms. O objetivo deste controle é o de manter o usuário em “tempo de espera”, enquanto o programa continua a processar os dados; é a versão gráfica da tal mensagem “Aguarde..!” nos programas que rodam em sistemas operacionais não gráficos, como o DOS. Um exemplo disto é quanto o programa deve ler um base de dados muito grande, como está acontecendo no livro que estou escrevendo sobre a linguagem C#. No banco de dados relativo a um dos capítulos tem uma tabela (“Atores”) com mais de 22 mil registros, onde um dos métodos deve ler todos os atores. As propriedades mais importantes desta classe, depois, evidentemente, de *name*, são as seguintes: *maximum*, *minimum* e *step*. A primeira pode ser utilizada para controlar o progresso, fazendo a prefeita sincronização entre o tempo necessário para concluir o processo (dentro dos tais “cem por cento”) e o preenchimento do controle. A **figura 1** mostra a aplicação rodando e lendo uma tabela com uma quantidade muito grande de registros. Esta figuramostra uma tela de apresentação (tela Splash) do “Sistema de Controle Comercial”, com *maximum*=100, *minimum*=0 e *step*=10. A *ProgressBar* mostra o progresso de carregamento com o percentual e texto do título do sistema de maneira animada, mostrando letra por letra. Na codificação do progresso a sincronização com o percentual mostrado pode ser calculada como mostrado na **figura 2** (em uma proporção) que calcula o percentual atual (**x%)** em função de **nRegs** (número de registros lidos até o momento) e **totRegs** (total de registro da tabela). Ao final é mostrado o código do evento ***Tick()*** do temporizador (Timer), escrito em C#, para fazer a sincronização

----------------------------------------------------------------------------------------------------------------------------------------------------------

Para adquirir o *pdf/e-book* deste livro ou o *pdf* de outros livros sobre programação, entre em contado pelo *e-mail*: **marleite@gmail com**

----------------------------------------------------------------------------------------------------------------------------------------------------------





**private void** timer1\_Tick(**object** sender, **EventArgs** e)

{

nRegs = nRegs + 1; *//contador de registros lidos*

if (nRegs <= totRegs)

{

progressBar1.Value = progressBar1.Value + 1; *//incrementa progresso*

**double** x = (progressBar1.Value);

percx = (100 \* nRegs) / totRegs;

**string** percS = **String**.Format("{0:#.#,##}", percx);

lblPerc.Text = percS + "%";

}

**else**

{

tmrProgess.Enabled = **false**; *//desabilita o Timer*

**this**.Hide(); *//oculta o formulário*

**frmLogin** objFormLogin = **new** **frmLogin**(); *//cria objeto form de Login*

objFormLogin.ShowDialog();

}

}