Série Convergente 2

Mário Leite

...

Na postagem anterior foi comentado sobre séries convergentes e divergentes, apresentado um programa (codificado em Visualg) sobre a expressão da série **x(x+1)** onde foi calculada a soma dos seus termos quando esse número de termos tende ao infinito. O resultado obtido mostrou que esse limite é **1**, onde a **figura 1** mostra a expressão desta série, indicando o seu limite.

$$\lim_{x \to \infty} \left[x/(x+1) \right]$$

Figura 2 - Expressão a ser calculada

Nesta presente postagem é mostrado o programa **"Serie2"**, codificado em C#, agora considerando duas situações: somando um *milhão* e um *bilhão* de termos., respectivamente A **figura 1a** mostra a entrada do programa para um milhão de termos e a **figura 1b** a saída do programa e o tempo de processamento com este número de termos. A **figura 2a** mostra a entrada para um *bilhão* de termos e a **figura 2b** a saída com o tempo gasto no processamento para este número alto de termos. O programa foi rodado nas mesmas condições de *hardware* utilizado na postagem anterior para um *milhão* de termos, quando foi codificado em Visualg. Note a brutal diferença de tempo de processamento: **3995,3 segundos** em Visualg e de apenas **48,9 segundos** quando codificado em C#. Considerando um *bilhão* de termos, o tempo de processamento foi um pouco mais de dez horas, o que dá em torno de **37312,5** segundos, rodado nesta linguagem, no mesmo equipamento e nas mesas condições de memória e espaço útil em disco.

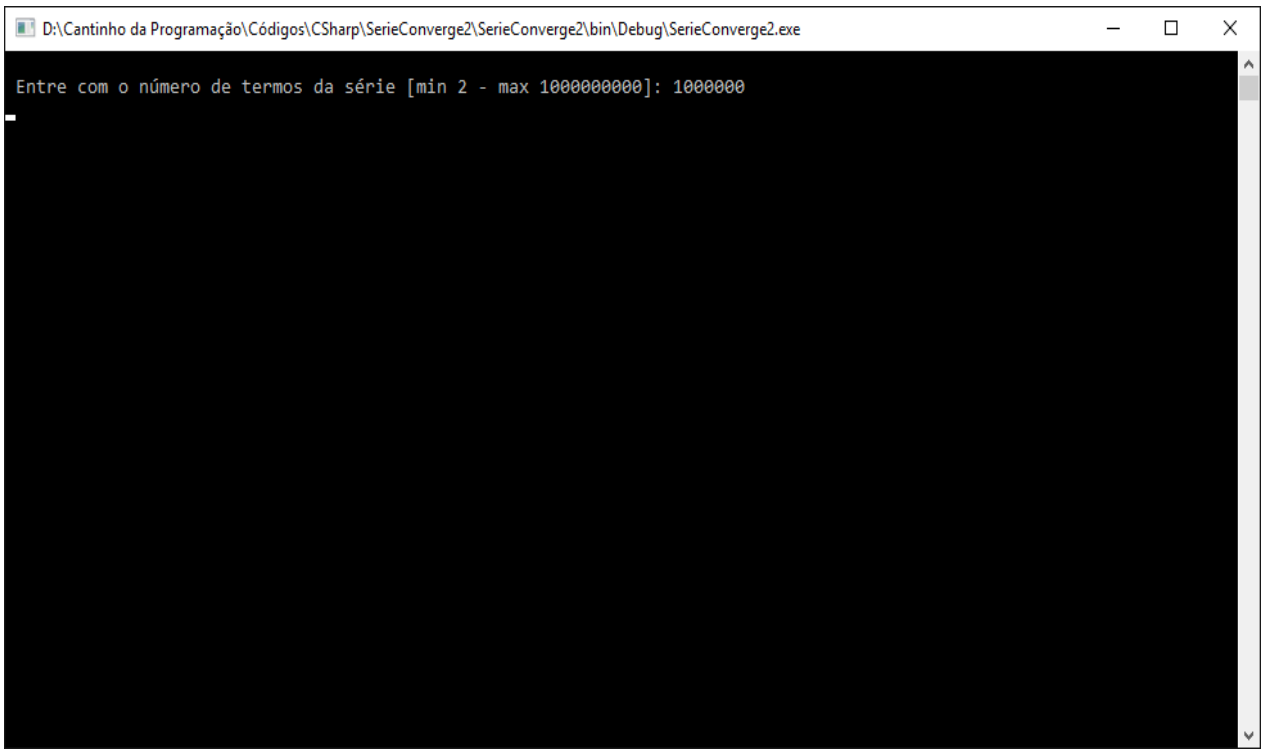

Figura 1a - Entrando com um milhão de termos da série

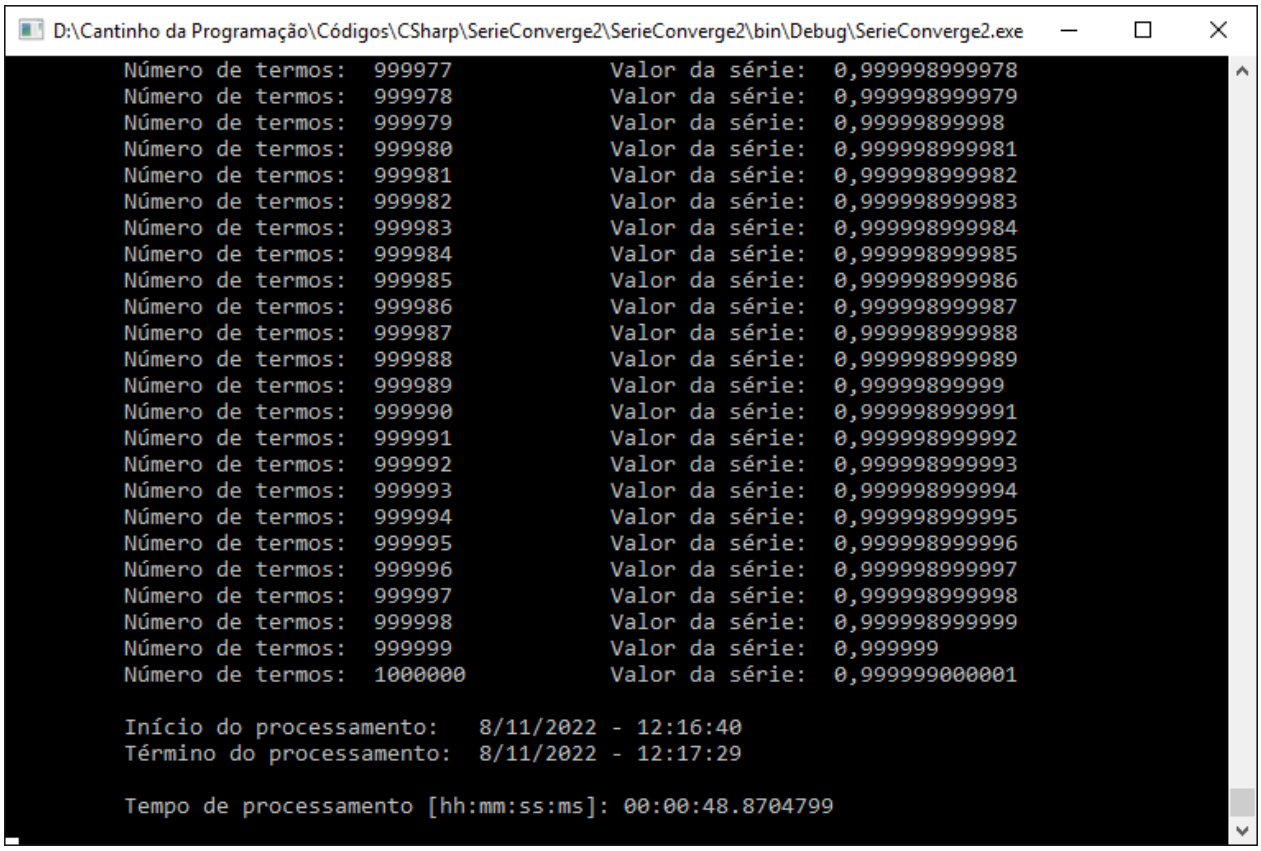

Figura 1b - Saída do programa para um milhão de termos da série

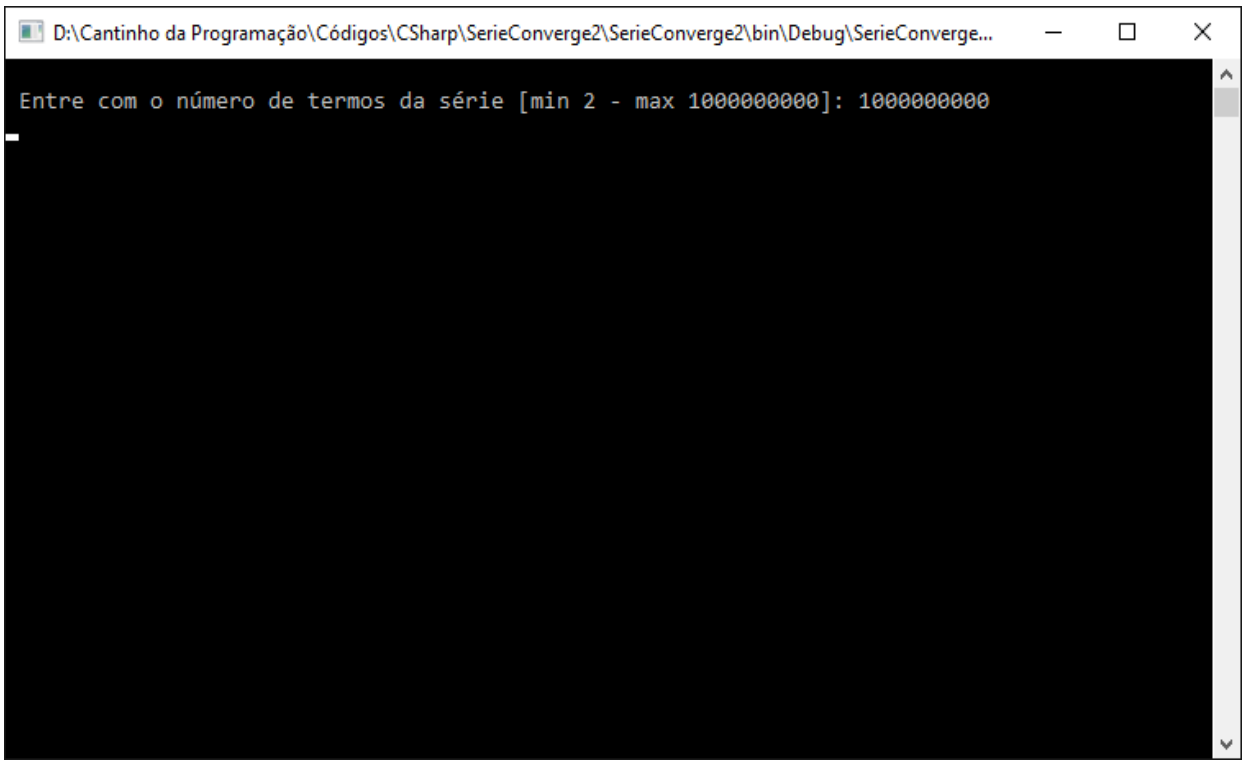

Figura 2a - Entrando com um bilhão de termos da série

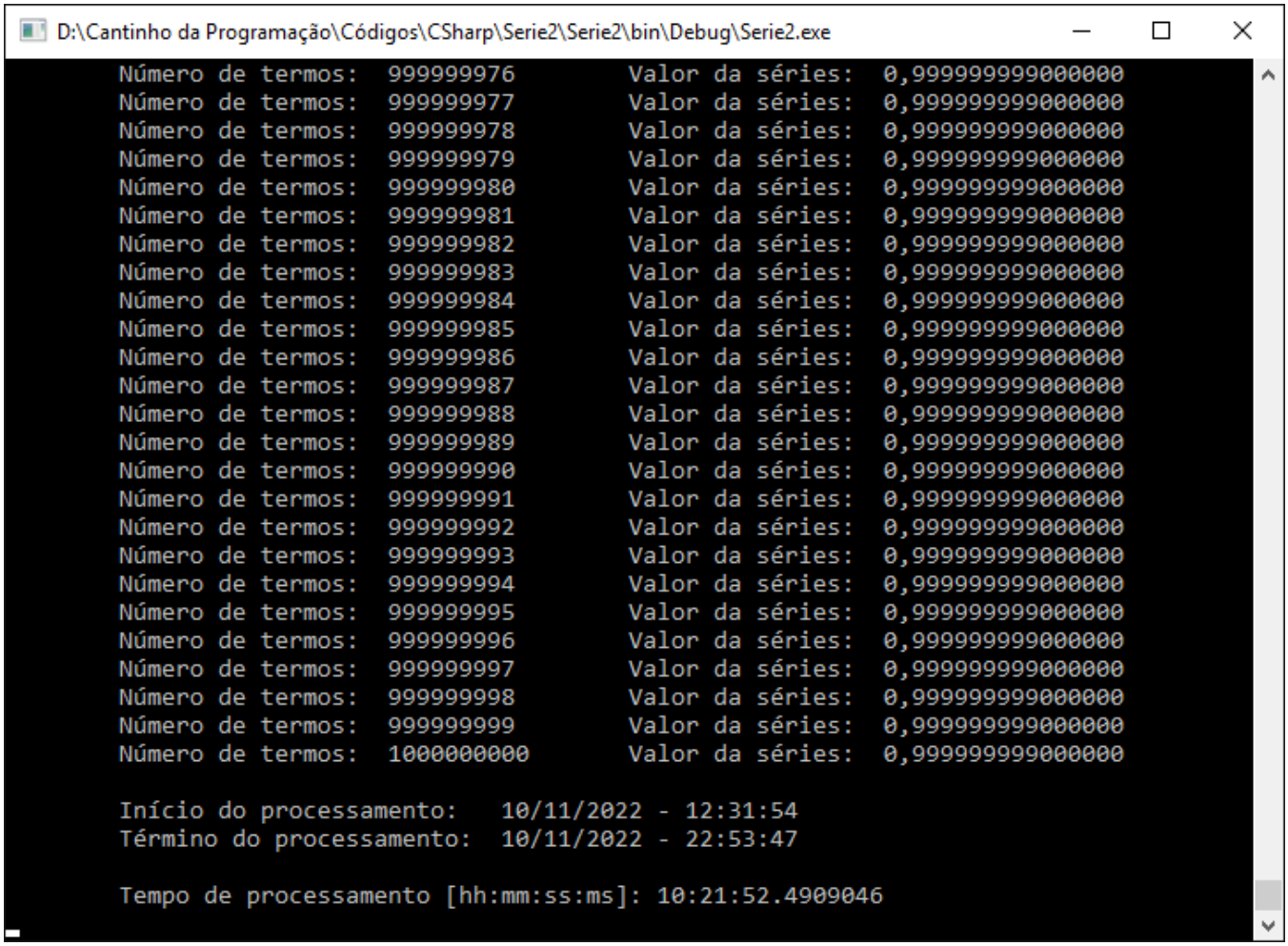

Figura 2b - Saída do programa para um bilhão de termos da série

Código da aplicação

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Diagnostics;

namespace Serie2
{
    //Calcula o limite de uma séria
    //Em C#
    //Autor: Mário Leite
    internal class Program
    {
        static void Main(string[] args)
        {
            const int MAX=1000000000;
            int n;
            double Sx;

            /* Define variáveis de espaços formatadas adequadamente */
            string strT = "        Número de termos:  ";
            string strS = "       Valor da série:  ";
            string strIni = "        Início do processamento:   ";
            string strFim = "        Término do processamento:  ";
```

```
            /* Define instâncias temporais iniciais */
            int diaHoje1 = int.Parse(DateTime.Now.Day.ToString());
            int mesHoje1 = int.Parse(DateTime.Now.Month.ToString());
            int anoHoje1 = int.Parse(DateTime.Now.Year.ToString());
            string hora1 = DateTime.Now.ToString("HH:mm:ss tt"); //formato 24 horas
            string data1 = diaHoje1 + "/" + mesHoje1 + "/" + anoHoje1;
            Stopwatch ObjSw = new Stopwatch(); //cria objeto de tempo
            ObjSw.Start(); //liga o cronômetro

            Console.WriteLine(); //apenas salta uma linha

            /* Valida entrada do número de termos da série */
            n = 1;
            while ((n < 2) || (n > MAX))
            {
                Console.Write(" Digite número de termos da série [min 2 - max " +MAX +"]: ");
                n = int.Parse(Console.ReadLine());
            } //fim da validação

            Console.WriteLine();

            /* loop para gerar os valores da Série */
            for (int x = 1; x <= n; x++)
            {
                Sx = (double)x / (x + 1); //expressão da Série
                string k = x.ToString();  //converte x para string
                double pot = Math.Pow(10, k.Length); //parcela de espaço ser adicionada
                int spc = (10 - k.Length) + k.Length; //espaço total ocupado pelo termo
                string termo = k.PadRight(spc,' '); //preenche o termo com espaços em branco
                var Serie = String.Format("{0:0.000000000000000}",Sx); //formata valor da Série
                Console.WriteLine(strT + termo + strS + Serie); //imprime linha do resultado
            } //fim do loop de geração dos valores da série

            Console.WriteLine();

            /* Define instâncias temporais finais e mostra o tempo de processamento */
            ObjSw.Stop(); //desliga o cronômetro
            int diaHoje2 = int.Parse(DateTime.Now.Day.ToString());
            int mesHoje2 = int.Parse(DateTime.Now.Month.ToString());
            int anoHoje2 = int.Parse(DateTime.Now.Year.ToString());
            string data2 = diaHoje2 + "/" + mesHoje2 + "/" + anoHoje2;
            string hora2 = DateTime.Now.ToString("HH:mm:ss tt"); //formato 24 horas
            Console.WriteLine(strIni + data1 + " - " + hora1);
            Console.WriteLine(strFim + data2 + " - " + hora2);
            Console.WriteLine();
            TimeSpan tempo = ObjSw.Elapsed; //calcula o tempo de processamento
            Console.WriteLine("        Tempo de processamento [hh:mm:ss:ms]: {0}", tempo);
            Console.ReadKey();
        } //fim do método principal
    } //fim da classe
} //fim do programa
```