

INSTITUTO FEDERAL DE MINAS GERAIS

Bacharelado em Sistemas de Informação

Professor: Carlos Alexandre

Cálculo Numérico:   
Implementações da Linguagem C#

Turma 2016

Tares Liberato Orlande de Almeida

Sabará

2018

**Introdução**

Tem como objetivo a análise da performance da linguagem C# utilizando a biblioteca de diagnóstico (System.Diagnostics) que a IDE oferece. Com objetivo de medir o tempo das funções.

**Obs.: Código Implementado no Visual Studio.**

**O qual é gratuito para todo aluno matriculado no IFMG.**

**Problemas Propostos**

Dos 8 problemas propostos 4 foram implementados para a Linguagem

* Fibonacci recursivo
* Parse Int
* Quicksort
* Conjunto de Mandelbrot
* Geração do π pela fórmula de Euler
* Estatística em Matriz Randômica
* Método de Relaxação Sucessiva (SOR)
* Método de Newton

**O que é a Linguagem**

O **C# (C-Sharp)** é uma linguagem de programação orientada a objetos desenvolvida pela Microsoft, e está entre as mais utilizadas no mundo. Possui uma sintaxe bem parecida com outras linguagens populares como C, C++, Java e Object Pascal. É considerada por programadores como simples de aprender.

O C# é compilado para Common Intermediate Language (CIL) que é interpretado pela máquina virtual Common Language Runtime (CLR). É uma linguagem de programação multi-paradigma fortemente tipada.

Algumas outras características do C#:

* É uma linguagem visual.
* Dirigida por eventos.
* Possui paradigmas de programação funcional, genérica, declarativa, orientada a objetos e imperativa.
* Por utilizar o conceito de máquina virtual, é possível rodar a aplciação do C# sem envolvimento do sistema operacional. O Common Language Runtime (CLR) isola a aplicação do sistema operacional.
* Permite intercâmbio entre linguagens, através do uso de componentes.
* Possui funções que são consideradas diferenciais por alguns programadores, tais como:
  + permite implementar propriedades e sobrecarga de operadores.
  + possui a função de implementar um modo inseguro.
  + tem suporte à indexadores.
* O código em C# é armazenado em arquivos com extensão “.cs”.

**Implementação**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

//using static System.Console;

namespace Fibonacci

{

class Program

{

public static int Fibonatchi(int position)

{

if (position == 0)

{

return 1;

}

if (position == 1)

{

return 1;

}

else

{

return Fibonatchi(position - 2) + Fibonatchi(position - 1);

}

}

//----------------------------------------------

public static int ParseInt(string value)

{

try

{

return int.Parse(value);

}

catch

{

return -1;

}

}

//----------------------------------------------

static public void QuickSort\_Recursive(int[] vetor, int primeiro, int ultimo)

{

int baixo, alto, meio, pivo, repositorio;

baixo = primeiro;

alto = ultimo;

meio = (int)((baixo + alto) / 2);

pivo = vetor[meio];

while (baixo <= alto)

{

while (vetor[baixo] < pivo)

baixo++;

while (vetor[alto] > pivo)

alto--;

if (baixo < alto)

{

repositorio = vetor[baixo];

vetor[baixo++] = vetor[alto];

vetor[alto--] = repositorio;

}

else

{

if (baixo == alto)

{

baixo++;

alto--;

}

}

}

if (alto > primeiro)

QuickSort\_Recursive(vetor, primeiro, alto);

if (baixo < ultimo)

QuickSort\_Recursive(vetor, baixo, ultimo);

}

//----------------------------------------------

public static double Euler()

{

double pi = 0, sum = 0;

for (int n = 1; n < 1000000; n++) {

sum += 1/Math.Pow(n,2);

}

pi = (double)sum \* 6;

pi = Math.Sqrt(pi);

return pi;

}

//----------------------------------------------

static void Main(string[] args)

{

int value = 32;

int[] numeros = { 5, 8, 9, 6, 3, 2, 1, 5, 4 };

string valueStr = "150000";

var stopwatch = new Stopwatch();

//Fibonacci

Console.WriteLine("Fibonacci Recursivo");

stopwatch.Start();

Fibonatchi(value);

stopwatch.Stop();

Console.WriteLine($"Tempo passado: {stopwatch.Elapsed}");

Console.WriteLine("\n");

// Parse Int

Console.WriteLine("Parse Int");

stopwatch.Restart();

ParseInt(valueStr);

stopwatch.Stop();

Console.WriteLine($"Tempo passado: {stopwatch.Elapsed}");

Console.WriteLine("\n");

// QuickSort Recursivo

Console.Write("QuickSort Recursivo \nVetor Desordenado - ");

for (int i = 0; i < 9; i++)

Console.Write(numeros[i] + " ");

Console.WriteLine();

stopwatch.Restart();

QuickSort\_Recursive(numeros, 0, numeros.Length - 1);

stopwatch.Stop();

Console.Write($"Tempo passado: {stopwatch.Elapsed}");

Console.Write("\nVetor Ordenado - ");

for (int i = 0; i < 9; i++)

Console.Write(numeros[i]+" ");

Console.WriteLine("\n");

// Geração do Pi pela formula de Euler

Console.WriteLine("Geração do Pi pela formula de Euler");

stopwatch.Restart();

Euler();

stopwatch.Stop();

Console.Write($"Tempo passado: {stopwatch.Elapsed}");

Console.WriteLine("\n");

// Keep the console window open in debug mode.

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

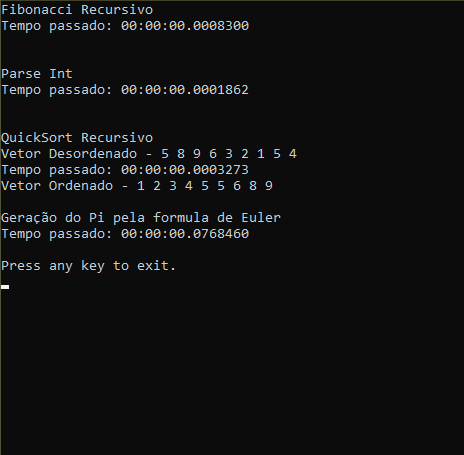
}

}

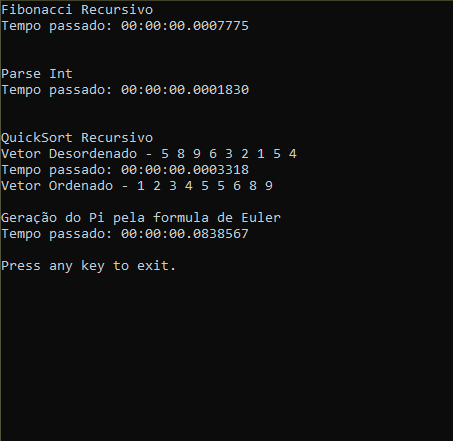
}

**Resultados**

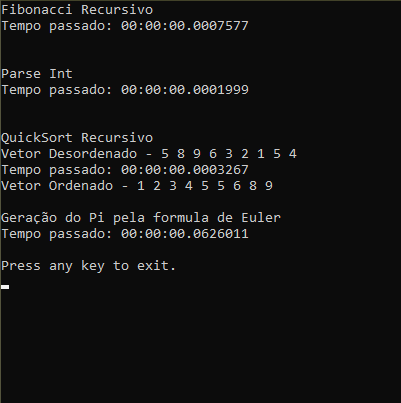
Saída 1



Saída 2



Saída 3



**Referências**

[**https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/inside-a-program/hello-world-your-first-program**](https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/inside-a-program/hello-world-your-first-program)

[**https://stackoverflow.com/questions/9828762/print-a-string-of-fibonacci-recursively-in-c-sharp**](https://stackoverflow.com/questions/9828762/print-a-string-of-fibonacci-recursively-in-c-sharp)

[**https://pt.stackoverflow.com/questions/16089/diferen%C3%A7as-entre-parse-vs-tryparse**](https://pt.stackoverflow.com/questions/16089/diferen%C3%A7as-entre-parse-vs-tryparse)

[**https://pt.stackoverflow.com/questions/89622/como-verificar-o-tempo-de-execu%C3%A7%C3%A3o-de-um-m%C3%A9todo**](https://pt.stackoverflow.com/questions/89622/como-verificar-o-tempo-de-execu%C3%A7%C3%A3o-de-um-m%C3%A9todo)

[**https://pt.stackoverflow.com/questions/86014/como-fazer-a-aplica%C3%A7%C3%A3o-esperar-por-um-tempo**](https://pt.stackoverflow.com/questions/86014/como-fazer-a-aplica%C3%A7%C3%A3o-esperar-por-um-tempo)

[**https://code.msdn.microsoft.com/windowsdesktop/Exemplo-de-mtodo-recursivo-1f51a7d8**](https://code.msdn.microsoft.com/windowsdesktop/Exemplo-de-mtodo-recursivo-1f51a7d8)

[**http://www.sbmac.org.br/cmacs/cmac-se/2013/trabalhos/PDF/4707.pdf**](http://www.sbmac.org.br/cmacs/cmac-se/2013/trabalhos/PDF/4707.pdf)

[**https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/arrays/multidimensional-arrays**](https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/arrays/multidimensional-arrays)

[**https://msdn.microsoft.com/pt-br/library/system.numerics.complex(v=vs.110).aspx**](https://msdn.microsoft.com/pt-br/library/system.numerics.complex(v=vs.110).aspx)

[**https://www.portalgsti.com.br/csharp/sobre/**](https://www.portalgsti.com.br/csharp/sobre/)