C# para iniciantes – Udemi

1. O que é .NET e o que é possível fazer com C#?

Bibliotecas são módulos que estendem a linguagem de programação.

Plataforma é um conjunto de bibliotecas. .NET é uma plataforma onde se encontram as bibliotecas. Múltiplas linguagens de programação podem utilizá-la.

C# é uma linguagem de programação da Microsoft.

Em 2020 existiam 3 ambientes: .NET framework, .NET Core e Xamarin – a Microsoft unificou?

Ambiente MONO (não é oficial da Microsoft) – roda em qualquer sistema e serve para criar qualquer coisa.

1. Instalação do .NET Framework

Ok

<https://dotnet.microsoft.com/en-us/learn/dotnet/hello-world-tutorial/create>

<https://tecnoblog.net/responde/7-maneiras-de-abrir-o-prompt-de-comando-no-windows-10-e-11/>

1. –
2. Montando ambiente de desenvolvimento com o Visual Studio

O que é o Visual Studio? – Ambiente de desenvolvimento integrado – ferramenta utilizada para criar aplicações com C#.

2. Entendendo a estrutura de um programa C# - arquivos da linguagem C# tem extensão .cs.   
   Estrutura inicial de um programa c#: using – módulos; pacote; biblioteca – cada conjunto de funcionalidades está dentro de um módulo. Só inserir o que será usado do C#. Serve para diminuir o tamanho do programa para que não seja muito grande mesmo sendo um código simples – economiza memória, fica mais simples.

Módulo System – C# seja capaz de manipular coisas do sistema operacional do usuário. Text – permite que manipule textos. Threading Tasks   
Ver vídeo: o que é um processador e para que serve?

1. Cada comando que coloca no programa C# é necessário finalizar com ;.

Para aparecer no console, iniciar o código com Console.  
\n = caracter de escape – diz para o computador para quebrar uma linha = escrever WriteLine.

Alinhar código = ctrl KD

Navegar nas pastar do Windows via CMD

Cd nomeDaPasta=> abre uma pasta c:/

Cd .. => volta uma pasta

Explorer (nome do arquivo pra abrir)

GIT – Estudar

Git status => identifica o que há modificado

Git add . (OU) git add nome do arquivo

Git commit -m “aqui coloca msg da modificaçao”

Git push

Cd hh

**Exemplo:**

cd c:/

ls

cd git

ls

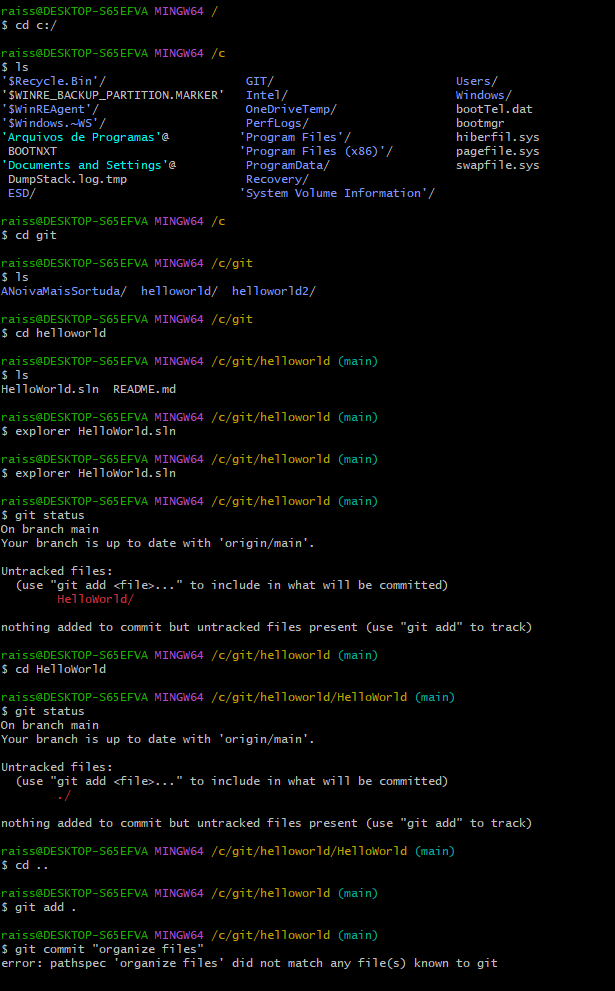
cd helloworld

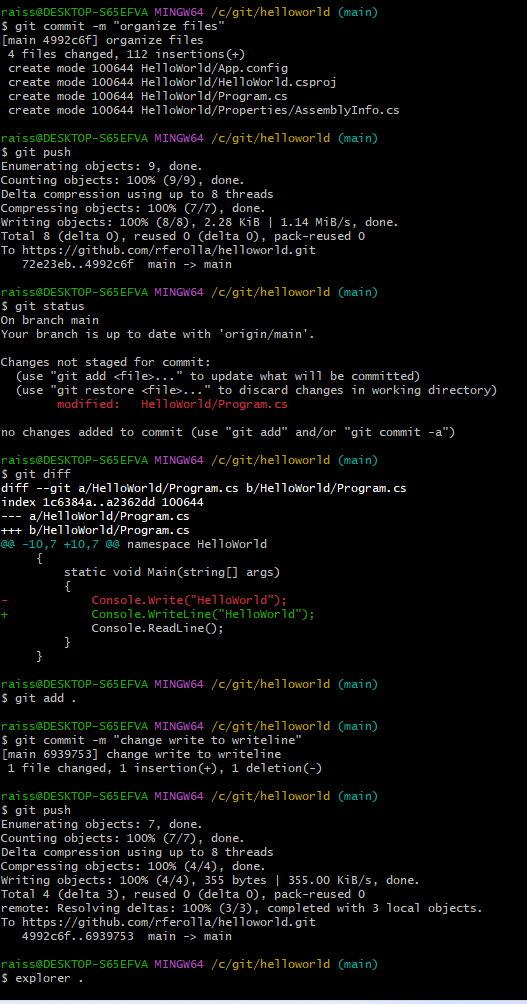
git status

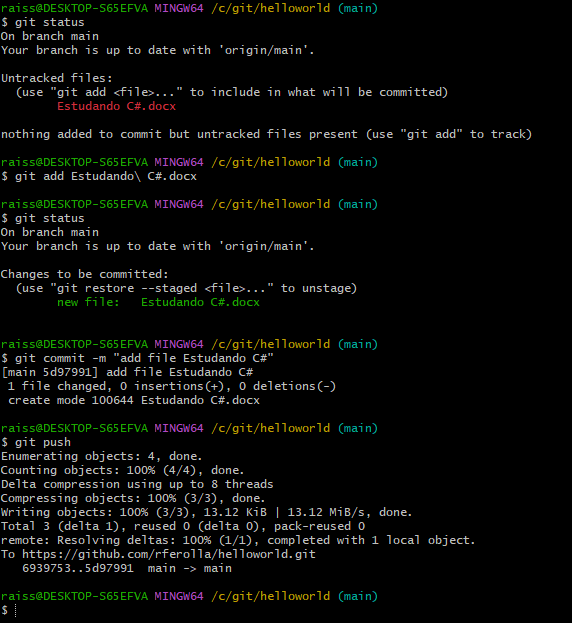
git add

git commit -m (add name)

git push







1. Conceito de variável

- Variáveis são locais na memoria RAM do pc onde conseguimos guardar informações. E podemos recuperar informações em qualquer momento do ciclo de vida do programa.

Guardar dado na variável e, enquanto o programa não fechar, o dado fica salvo na variável. –

***Dados do tipo texto começam com aspas duplas e terminam com aspas duplas.***

Toda vez que se está fazendo algo extremamente repetitivo na programação, provavelmente há uma forma melhor de se fazer. Utilizando variáveis.

Código: var (nome que quer dar ao arquivo) = “palavras a serem repetidas quando colado”

Ao utilizar: Console.WriteLine(meuTexto);

***Utilizar quando for utilizar a variável em várias partes do código, pois é só alterar o var para alterar nos outros lugares.***

Ex: var meuTexto = “Hello World, Raissa Ferolla”

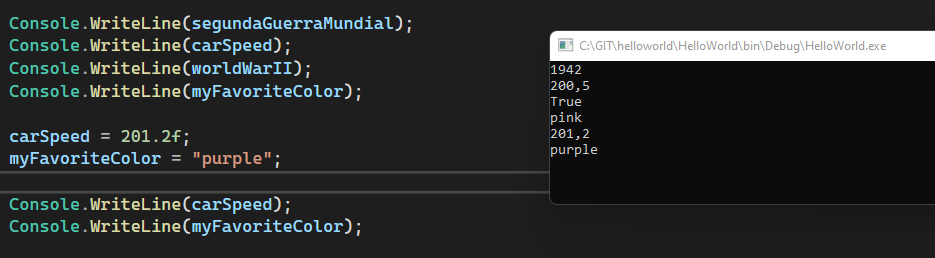
1. Declaração de variáveis e tipos

Como declarar variáveis no C#?

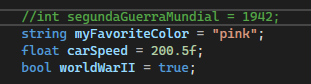
* Declaração de variável por **tipo**: // **Int** – (serve para armazenar dados; são números inteiros) // int um56 como 32334 ; // **Float** – (flutuante – número que pode ser negativo ou positivo, mas as casas decimais podem mudar.) // float – um5.6 como 232.132 ; // **Bool** – true or false; // **String** – “” (identificado por aspas duplas – qualquer dado identificado com aspas duplas é uma string, um texto).; // Char – ‘’ (identificado por aspas simples – character). ‘a’
* Como declarar variáveis: a maneira mais fácil é falar para o c# qual o tipo da variável que vai ser criada. ex: **int** SegundaGuerraMundial (não pode ter espaço, caracteres especiais – apenas \_ - e o nome da variável não pode começar com número; também não pode dar um nome já reservado ao C# - ex: class ou using ou static).
* Quando armazenamos um número puro (int ou float) é possível fazer operações matemáticas (soma, subtração, divisão..)
* Quando utilizar número decimal do tipo float é preciso colocar um f no f inal – ex: float carSpeed = 200.5f
* Não podemos ter variáveis com o mesmo nome no mesmo escopo/lugar. Ex: não podemos usar: int SegundaGuerraMundial = 1942 e bool SegundaGuerraMundial = true

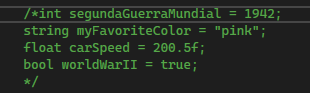


* Caso queira alterar uma variável, não é necessário colocar o tipo em linhas abaixo, apenas o nome da mesma. Ex: carSpeed = 201.2f



1. Se, antes do código, colocar // o C# identifica aquela linha como comentário e ignora. É bom para explicar. Para comentário com múltiplas linhas utilizar /\* e, no final, colocar \*/





Outras formas de declarar variáveis:

* Omitindo o tipo:

var (nome da variável escolhido) = (variável)

Exemplos:

var corFavorita = “rosa”

var modeloDoProduto = 2323

O comportamento dos tipos é o mesmo. Mesmo que se omita, não podemos dar um nome como texto “ “ e depois utilizar a variável de outra forma.

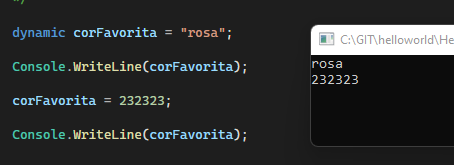
Ex: var corFavorita = “rosa”; var corFavorita = “roxo” -> certo

var corFavorita = “rosa”; var corFavorita = 2212 -> errado

* Tipagem dinâmica:

É possível utilizar o tipo dynamic caso haja necessidade de mudar o tipo da variável ao longo do código.

Ex: dynamic corFavorita = “rosa” ; corFavorita = 2323232



Esta variável tem menor desempenho. Não é recomendável de ser utilizada, apenas em situações muito específicas.

1. Lista de exercícios.
2. String
3. Float, Double
4. Char
5. Bool
6. C

Link para estudo:

<https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/builtin-types/floating-point-numeric-types>

1. Constantes

Tem a mesma função das variáveis, mas uma característica diferente. Quando se cria uma constante, se define um valor para ela uma vez e nunca mais pode ser mudado durante toda a execução do programa.

Serve para segurança – pois certos dados não devem ser mudados – evita acidentes.

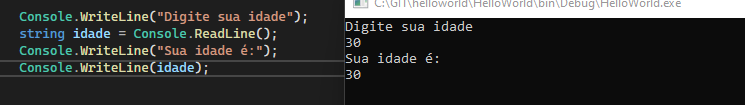
const (tipo) (nome dado) = valor;

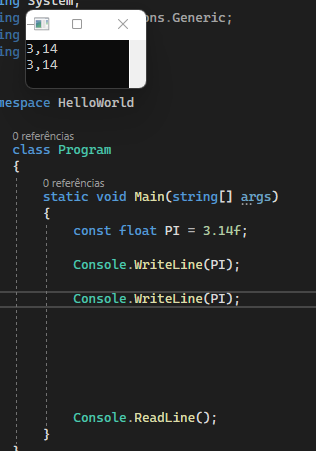
ex: const float PI = 3.14f;

1. Inserindo usuário – como fazer o visual studio ler o que se escreve na caixa de texto.

Código:

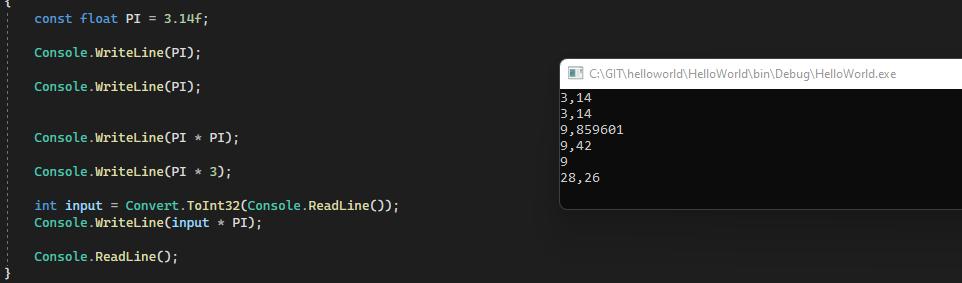
Console.WriteLine(“Escreva seu nome: “) – aparece na caixa de texto  
string nome = Console.ReadLine (); - o programa vai ler aquilo que for digitado e guardar como variável (neste caso, como ‘nome’)  
Console.WriteLine(“Seu nome é: “); – aparece na caixa de texto  
Console.WriteLine(nome); - repete o nome que foi digitado acima.





Console.ReadLine ()

Console.WriteLine (input)



**Dar uma olhada:** [**https://sites.google.com/site/csharpgpsi/home**](https://sites.google.com/site/csharpgpsi/home)

**Ler sobre: lógica de programação**

1. Operadores aritméticos

É possível fazer operações aritméticas pelo C# e ele te dá os resultados.

Ex: int numeroQualquer = 20 + 20 – 80;

int numeroMulti = 10 \* 9;

Int numeroDivi = 90 / 10;

Console.WriteLine(numeroQualquer);

Console.WriteLine(numeroMulti);

Console.WriteLine(numeroDivi);

Console.ReadLine();

* **Atenção!** Ao executar uma divisão, é preciso tomar cuidado com qual tipagem escolhemos utilizar. Isso porque, o C# permite que executemos contas que darão resultados com vírgula, mesmo com a variável int (int numeroDivi = 5/2); mas, ao exibir o resultado, ficará como número inteiro (Console.WriteLine(numeroDivi); - vai aparecer 2 na caixa de texto e não 2.5. O C# não arredonda – simplesmente pega a casa decimal e joga no lixo.

**Ao dividir dois números inteiros, o resultado vai ser sempre inteiro. Se não tem “.” alguma coisa (5.5) e nem f (5.5f), significa que o número é inteiro – mesmo que esteja com outra tipagem escrita.**

**Como resolver?** Dentro da divisão, é obrigatório que pelo menos um dos números seja decimal, seja float.

float numeroDiv = 5f/2; (ou)

float numeroDiv = 5.0f/2

**Tenha em mente: No C# quando dividimos um número do tipo inteiro, o resultado vai ser sempre inteiro, sem casa decimal – independente da divisão. E quando dividimos um número float com um número inteiro ou dois números float, o resultado vai ser sempre completo.**

**Atenção:** o C# segue as regras da matemática. Logo, int teste = 2 + 2 \* 10; Console.WriteLine(teste); = 22; mas int teste = (2+2) \* 10; Console.WriteLine(teste2); = 40.