

Programação Microinformática

Professor: Gilmar Aquino

gilmar.aquino@fatec.sp.gov.br

www.aquino.pro.br

Apresentação do Professor



NOME: GILMAR AQUINO

FORMAÇÃO: BACHAREL EM SISTEMA DE INFORMAÇÃO, ENGENHEIRO DE PRODUÇÃO, PÓS EM TECNOLOGIAS EM EDUCAÇÃO A DISTÂNCIA, MESTRADO EM ENSINO DE CIÊNCIAS E TECNOLOGIA, DOUTORADO EM ENSINO DE CIÊNCIAS E MATEMÁTICA (TECNOLOGIAS).



Ementa

Fatec
São Paulo

CPS
Centro
Paula Souza



GOVERNO DO ESTADO
DE SÃO PAULO

Objetivo

Fatec
São Paulo

CPS
Centro
Paula Souza



GOVERNO DO ESTADO
DE SÃO PAULO

Bibliografia

Fatec
São Paulo

CPS
Centro
Paula Souza



GOVERNO DO ESTADO
DE SÃO PAULO

Sistema de Avaliação

- Fórmula: $(TP*0.25) + (AP*0.25) + (P*0.5)$
- Legenda:

TP - Trabalho Prático

AP - Atividade Presencial/Remoto (Comunicação e Presença)

P - Prova



Formulário de Justificativa
aquino.pro.br/justificafatec

08/06/2021	Apresentação do Professor, apresentação disciplina, introdução ao conteúdo.	02/07/2021	Estrutura While como podemos aplicar em nossos programas, comparação entre While x For, qual momento utilizar While e FOR. Qual a diferença para o do/while.
12/06/2021	Funcionamento de um microcomputador, entrada de dados, processamento e saída. O que é Programação Microinformática? Funcionamento de IDE, Compilação, criação de arquivo, tipos de arquivos extensões. Evolução da programação.	03/07/2021	Criando sua própria função de maneira simples e funcional, enviando e recebendo dados, atividades de exemplos e funcionamento prático.
15/06/2021	Tipos de Dados, Principais tipos de dados, funcionamento de uma memória, como é uma variável, quais são as regras na programação. Primeira Programação de computador. Exemplo analisando um diagrama.	05/07/2021	ATIVIDADE TP
19/06/2021	Funcionamento de funções numéricas / Alfanuméricas, como as funções podem facilitar a vida de um programador, exemplos de execuções de funções.	06/07/2021	Introdução a Internet das coisas, funcionamentos, equipamentos, investimento em IoT, desktop x IoT.
22/06/2021	Trabalhando com cor no console, funções que facilitam a demonstração de resultado no console do usuário. Como funciona uma condição, operador mod, aplicação e funcionamento. Operadores Lógicos, exemplos e funcionamento.	07/07/2021	Estrutura física do equipamento, programação em microcomputador para IoT, qual a diferença, primeiros programas. O que devemos conhecer para programar.
26/06/2021	ATIVIDADE TP	08/07/2021	IDE Simples, simuladores, "Olá Mundo IoT", comparando as linguagens. As primeiras funções para interagir com o ambiente. Um pouco de eletrônica para programador.
28/06/2021	ATIVIDADE TP	09/07/2021	ATIVIDADE TP
30/06/2021	Estrutura de condição Switch/case, aplicações em programas de console, funcionamento e aplicação prática.	10/07/2021	O que são sensores e atuadores, como podemos receber um sinal processador e acionar um atuador. Quais as possibilidades de funcionamento da plataforma.
01/07/2021	Estrutura de repetição FOR, funcionamento aplicação e exercícios práticos.	12/07/2021	Outras funções específicas do equipamento, protocolos de comunicações para IoT, o que o futuro nos reserva.
		13/07/2021	AVALIAÇÃO FINAL
		14/07/2021	EXAME



www.menti.com

Conceitos

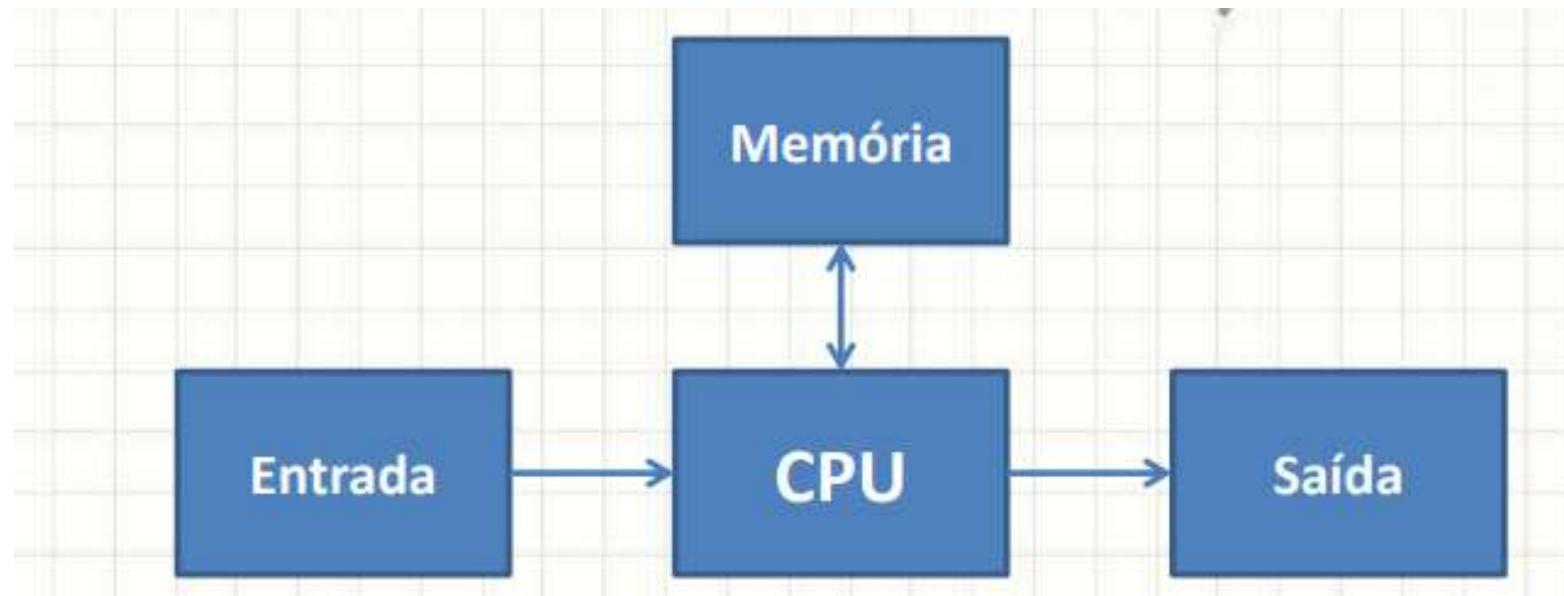
- Você está recebendo uma lista de objetos, conceitos, elementos, que fazem parte do estudo de Programação Microinformática. Escreva o que você sabe sobre cada um deles (podem ser explicações curtas de 1 ou 2):
- Programação
- Programa
- Microinformática
- Fluxograma
- Algoritmo
- Compilação
- Hardware
- Memória Volátil
- Memória não Volátil
- Software
- Sistema Aplicativo
- Sistema Operacional
- Arquivos
- IDE
- Microprocessador
- CPU
- Computador
- Dados
- Informação
- Conhecimento

Computador?

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO



Níveis de Máquina



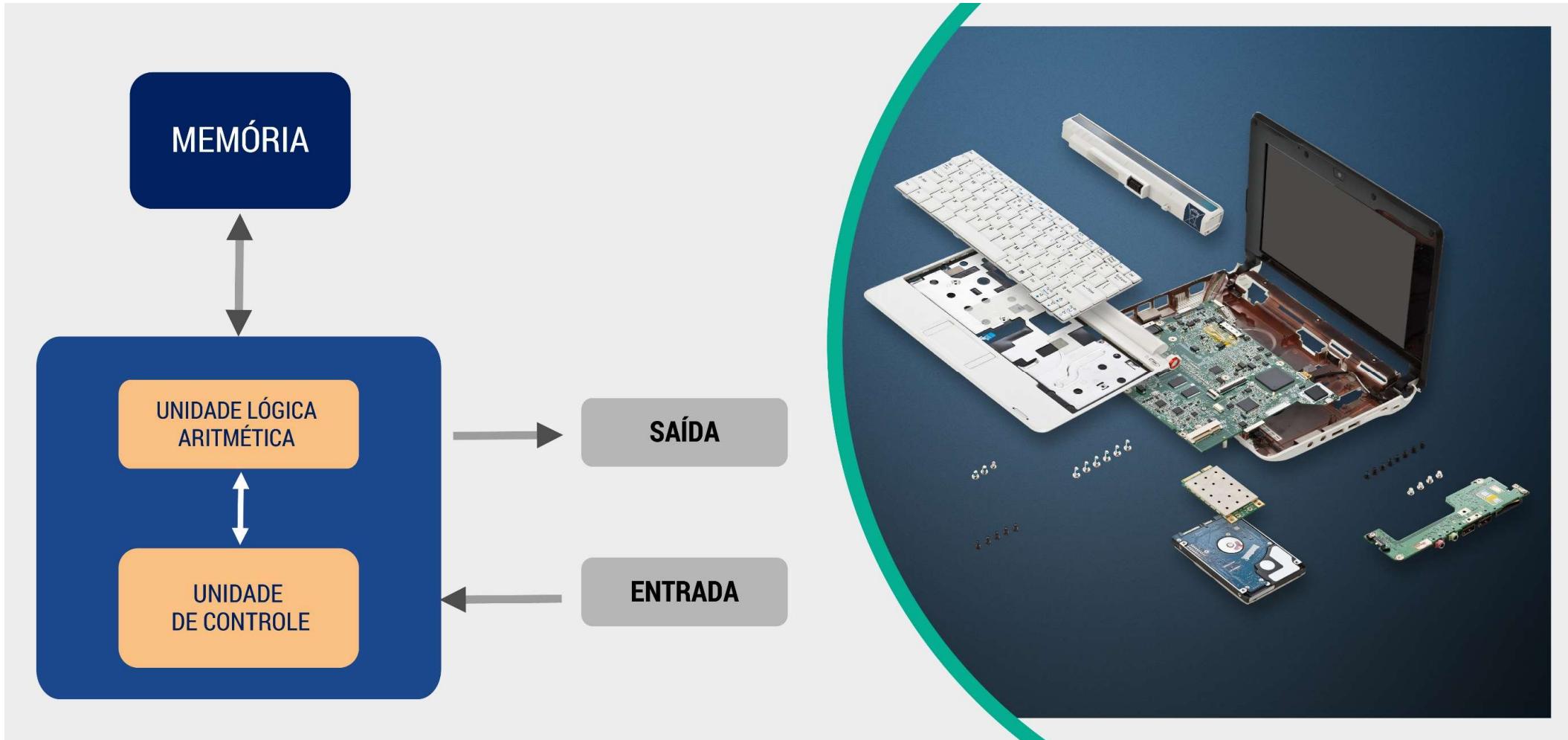
Armazenando Dados Eletricamente

No nível mais baixo, tudo vira sinal elétrico

- 0V = desligado = valor 0
- 3.3V (ou 5V) = ligado = valor 1

Existe só uma base numérica?

Decimal	Binario	Octal	Hexa
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F





smartphone
é ou não um
computador?



POR QUÊ?

DADOS TÉCNICOS DE UM *SMARTPHONE* ENCONTRADO NO MERCADO

Processador	A6 1.3 GHz Dual Core
Memória	64 GB
RAM	1 GB
Dispositivos de entrada e saída	<ul style="list-style-type: none">• Tela Touch Screen• Capacitiva• WiFi• Bluetooth



O que é um programa?

Programa?

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

**É um conjunto de instruções que
descrevem uma tarefa a ser realizada por
um computador.**

Programação é o processo de escrita, teste
e manutenção de um programa de
computador

Eu preciso de Lógica?
Para você o que é Lógica?



A lógica é o estudo do raciocínio analítico e coerente, fortalecido pelos fatos e pela argumentação sólida.

A lógica é a capacidade de pensar de forma racional

Na Tecnologia?

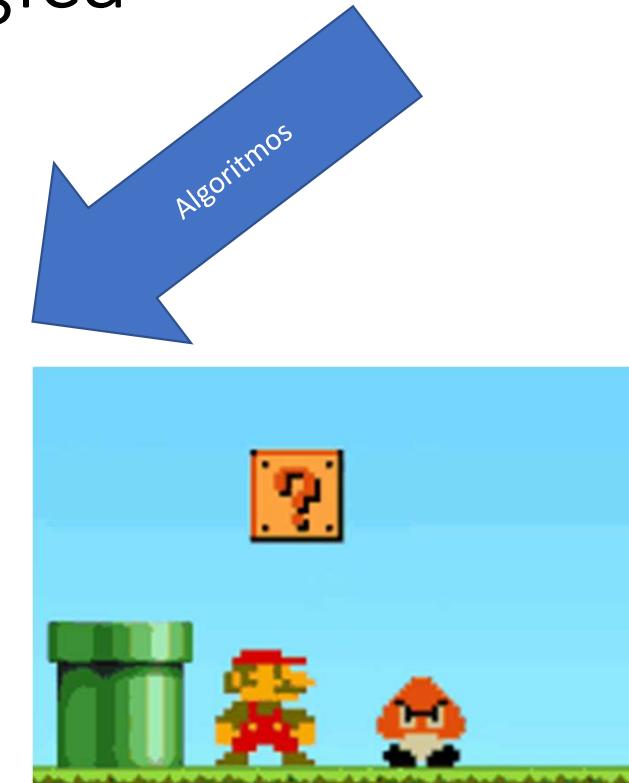
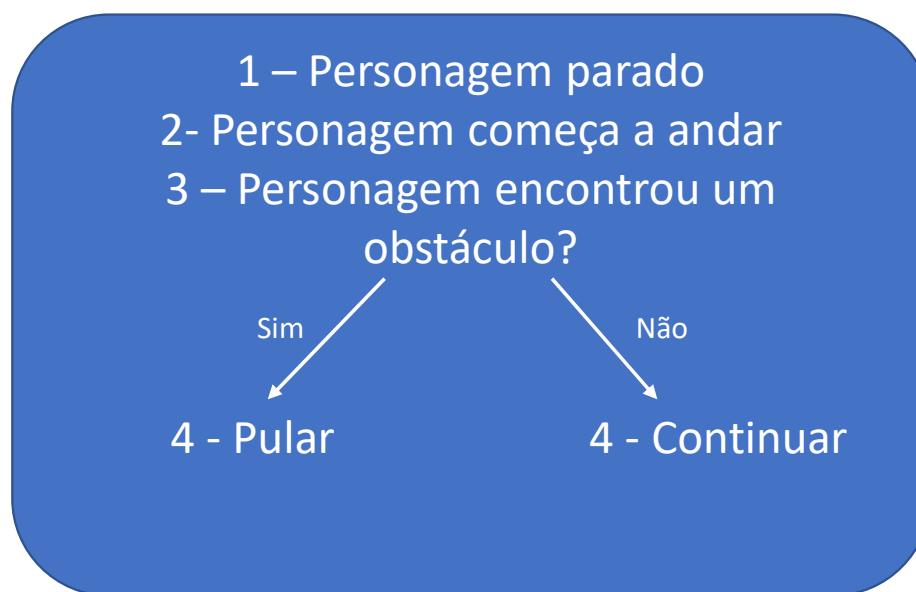
Na Tecnologia é fundamental ter um raciocínio lógico bem desenvolvido.

Lógica de Programação é a técnica Capaz de encadear os pensamentos para se atingir um objetivo.

Programas ou **softwares** nada mais são do que as instruções que o computador deve seguir em determinadas situações. Estas instruções vem de onde? De um **programador** é claro.

Instruções / Sequência Lógica

- Podemos descrever até mesmo situações simples.



Algoritmos

Fatec
São Paulo

CPS
Centro
Paula Souza



Antes de programar precisamos analisar o que desejamos desenvolver e em seguida escrever uma possível solução. Chamamos esta solução de **Algoritmos**.

Sequência Lógica de instruções que propõe resolver uma necessidade

Receita

Destino

Agenda

Média

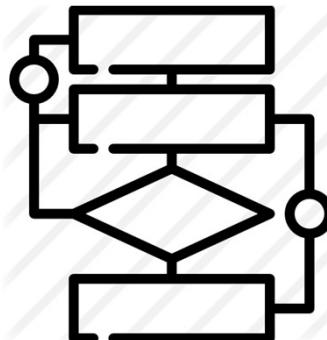
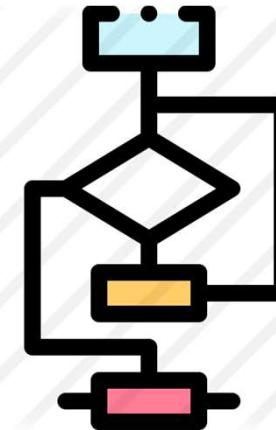
Algoritmos

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

Em ciência da computação, um algoritmo é uma sequência finita de ações executáveis que visam obter uma solução para um determinado tipo de problema. Ziviani, 2011.



"algoritmos são procedimentos precisos, não ambíguos, mecânicos, eficientes e corretos" Dasgupta, Papadimitriou e Vazirani, 2010,

A ordem das etapas é importante para se atingir o resultado desejado, todo algoritmo deve conter um início e fim.

Funcionamento de um computador



Como calcular uma média aritmética?

Entrada	Processamento	Saída
P1,P2,P3	$Média = \frac{(P1 + P2 + P3)}{3}$	Média Final

Funcionamento de um computador

- A **entrada** diz respeito as informações que o programa vai coletar, de varias formas, estas informações.
- O **processamento** diz respeito ao que o programa vai fazer com estas informações, é a transformação dos dados coletados na fase de entrada para que o objetivo do programa se cumpra.
- E por fim tempos a **saída**, que claramente nos indica o momento em que estas informações novas, geradas na fase de processamento, serão mandadas para fora do programa.

Sistemas de Arquivo

O que é um Arquivo?



Sistema de Arquivo

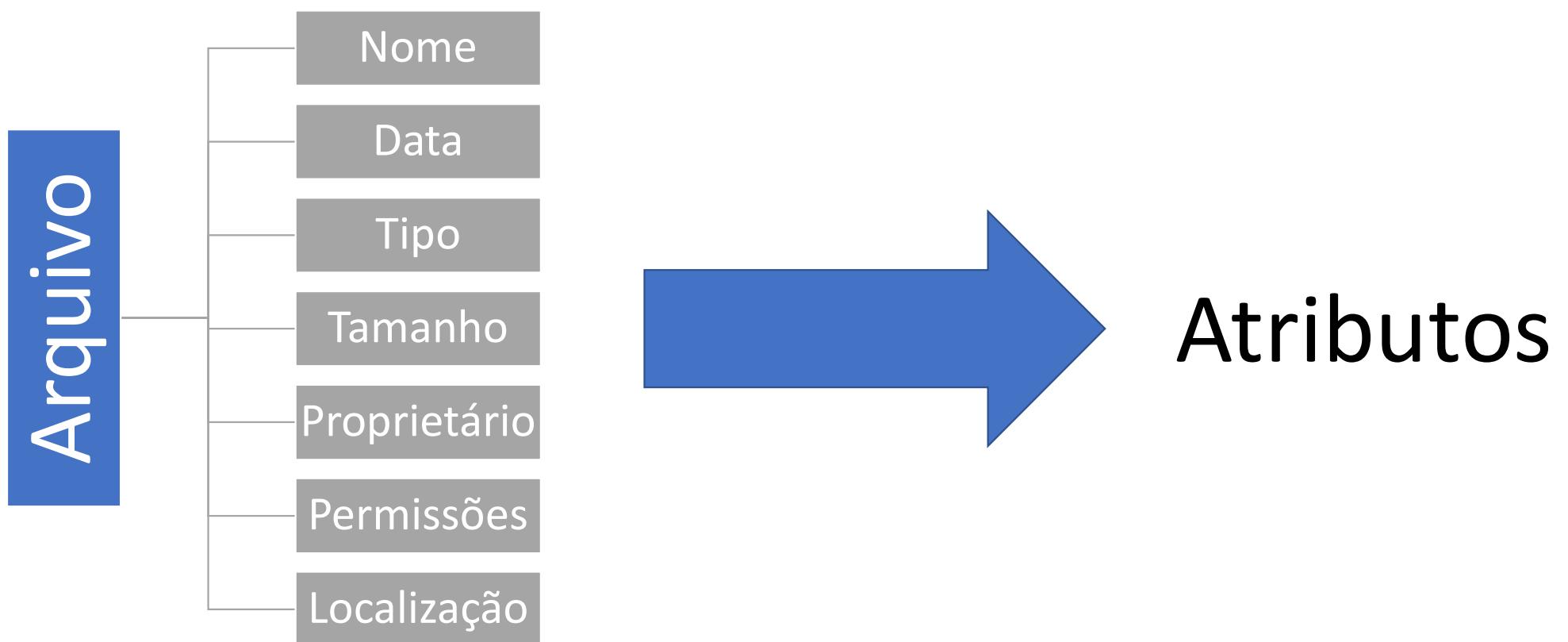
Arquivos

É um recurso para armazenamento de informação

Um conjunto de dados armazenados em um dispositivo
físico não volátil

Arquivos podem ter diversos formatos, desde um simples arquivo
texto ASCII com alguns poucos bytes , até mesmo vídeos com vários
megabytes ou gibabytes em diversos formatos

Sistemas de Arquivo



Sistemas de Arquivos

Fatec
São Paulo

CPS
Centro
Paula Souza



Atributos

- NOME:
 - Uma sequência de caracteres para identificar de tal forma que um ser humano.
- DATA:
 - Quando um arquivo foi criado, pode também controlar a data e a hora do último acesso, da última modificação do conteúdo e até mesmo da última alteração de outro atributo.
- TIPO:
 - Indica se o formato do arquivo é áudio, vídeo, imagem, texto ou outro. O sistema de arquivos do Windows e outros usam a extensão.

Sistemas de Arquivos

Atributos

- TAMANHO:
 - Esse atributo determina quanto de espaço em bytes (1 byte = 8 bits) .
- PROPRIETÁRIO:
 - Atualmente, com os sistemas em rede e multiusuários, cada arquivo tem um proprietário, que deve estar corretamente identificado, em alguns casos o próprio sistema é o proprietário do arquivo e por isso nenhum usuário deveria alterá-lo.
- PERMISSÃO:
 - Determina qual usuário têm acesso ao arquivo(leitura, escrita, remoção etc.).
- LOCALIZAÇÃO:
 - Esse atributo indica o dispositivo físico onde o arquivo se encontra e da posição do arquivo dentro do mesmo.

Sistemas de Arquivos

Fatec
São Paulo

CPS
Centro
Paula Souza

SP
GOVERNO DO ESTADO
DE SÃO PAULO

Operações

- As operações básicas envolvendo arquivos são:
 - Criar
 - Ler
 - Abrir
 - Escrever
 - Fechar
 - Mudar atributos
 - Remover

Sistemas de Arquivo

- Quando criados, nomes são dados
- Referenciados por meio de seu nome
- Tamanho: até 255 caracteres
 - Restrição MS-DOS aceita de 1 – 8 caracteres
- Letras, números, caracteres especiais podem compor nomes de arquivos:
 - Caracteres permitido: A-Z, a-z, 0-9, ^ ,&,',@,{},[],\$,=,!,-,#,(,),%,.,+,~,_
 - Caracteres não permitido: \/,|,?,<,>,*,;,"

Sistemas de Arquivos

- Alguns S.O são sensíveis (Case Sensitive)
- Unix é sensível:
 - Teste.c é diferente de teste.c
- DOS não é sensível:
 - Teste.c é = teste.c
- Windows 95~
 - Herdaram características do MS-DOS
- Windows NT/2000~
 - Utilizam sistemas de arquivos próprio **NTFS**

Sistemas de Arquivos

- Alguns Sistemas Operacionais suportam extensão:

- MS-DOS
 - 1 a 3 Caracteres
 - 8 Caracteres para nome + 3 extensão.
- Unix
 - Mais de 3 caracteres
 - Mais de 1 extensão (teste.c.Z)
 - Sem extensão



Linguagens de programação

A linguagem de programação é uma língua escrita onde o programador escreve as tais instruções e, após passar por um processo chamado **compilação**, as instruções se tornam um programa que pode ser **executado** pelo computador.

Este processo de compilação consiste em converter os comandos escritos na linguagem de programação em código de máquina, um tipo de código que a máquina entende e executa.

Existe somente um

Rank	Language	Type	Score
1	Python	🌐💻🖱️	100.0
2	C	📱💻🖱️	94.4
3	C++	📱💻🖱️	87.5
4	C#	🌐📱💻🖱️	74.5
5	Arduino	🖱️	67.2
6	Assembly	🖱️	63.7
7	Rust	🌐💻🖱️	55.5
8	LabView	💻🖱️	38.5
9	VHDL	🖱️	36.0
10	Verilog	🖱️	33.4

Evolução das linguagens

- <https://www.youtube.com/watch?v=8anOprjALhs>

IDE de Desenvolvimento

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

Vamos conhecer algumas →

eclipse-workspace - zend-expressive/src/Container/NotFoundDelegateFactory.php - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- bin
- docs
- src
 - Container
 - Exception
 - ApplicationFactory.php
 - ErrorHandlerFactory.php
 - ZendExpressive\Container
 - ErrorResponseGeneratorFactory.php
 - NotFoundDelegateFactory.php
 - NotFoundHandlerFactory.php
 - WhoopsErrorResponseGeneratorFactory.php
 - WhoopsFactory.php
 - WhoopsPageHandlerFactory.php
 - Delegate
 - Emitter
 - Exception
 - Middleware
 - AppFactory.php
 - Application.php

Quick Access

NotFoundDelegateFactory.php

```
1 <?php
2 /**
3  * @see      https://github.com/zendframework/zend-expressive for the canonical source re
4  * @copyright Copyright (c) 2017 Zend Technologies USA Inc. (http://www.zend.com)
5  * @license  https://github.com/zendframework/zend-expressive/blob/master/LICENSE.md New
6 */
7
8 namespace Zend\Expressive\Container;
9
10 use Psr\Container\ContainerInterface;
11
12 class NotFoundDelegateFactory
13 {
14     /**
15      * @param ContainerInterface $container
16      * @return NotFoundDelegate
17     */
18     public function __invoke(ContainerInterface $container)
19     {
20         $config = $container->has('config') ? $container->get('config') : [];
21         $renderer = $container->has(TemplateRendererInterface::class)
22             ? $container->get(TemplateRendererInterface::class)
23             : null;
24         $template = isset($config['zend-expressive']['error_handler']['template_404'])
25             ? $config['zend-expressive']['error_handler']['template_404']
26             : NotFoundDelegate::TEMPLATE_DEFAULT;
27         $layout = isset($config['zend-expressive']['error_handler']['layout'])
28             ? $config['zend-expressive']['error_handler']['layout']
29             : NotFoundDelegate::LAYOUT_DEFAULT;
30
31         return new NotFoundDelegate(new Response(), $renderer, $template, $layout);
32     }
33
34 }
```

PHP Writable Smart Insert 14:1 40

sketch_dec07a | Arduino 1.8.3

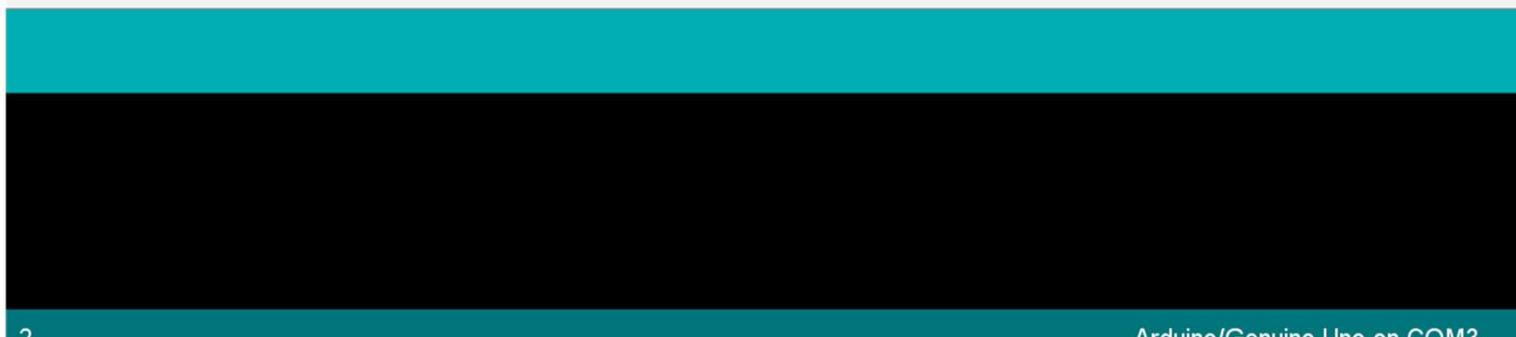
File Edit Sketch Tools Help



sketch_dec07a



```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```





GOVERNO DO ESTADO
DE SÃO PAULO

The screenshot shows the Notepad++ interface with the title bar reading "*D:\source\notepad4ever.cpp - Notepad++". The toolbar contains standard file operations like Open, Save, Print, and Find. Below the toolbar, two tabs are visible: "Notepad_plus.cpp" and "notepad4ever.cpp", with "notepad4ever.cpp" being the active tab. The code editor displays the following C++ code:

```
1 #include <GPL.h>
2 #include <free_software.h>
3
4 void notepad4ever()
5 {
6     while (true)
7     {
8         NotePad++;
9     }
10 }
```

A red bracket on the left margin highlights the entire function body from line 5 to line 10. The word "NotePad" is highlighted in purple, indicating it is a user-defined identifier.

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search Visual Studio... ProductivityFeatures KH

Debug Any CPU ProductivityFeatures ProductivityFeatures Live Share PREVIEW

Server Explorer Toolbox

Update160P1.cs Update160P2.cs

ProductivityFeatures

ProductivityFeatures.Update160P1 ForeachLoopToLinqMethod()

```
54
55 // Place cursor in foreach keyword and type (Ctrl + .)
56 // Select 'Convert to Linq(call form)' to convert to a method
57 0 references | kendrahavens, 20 days ago | 1 author, 1 change
58 public void ForeachLoopToLinqMethod()
59 {
60     var greetings = new List<string>()
61     {
62         "hi", "yo", "hello", "howdy"
63     };
64     foreach(var greet in greetings)
65     {
66         if(greet.Length < 3)
67         {
68             Console.WriteLine(greet);
69         }
70     }
71
72     // A codefix for adding the needed "using" will appear on copy/pasted code
73     // Uncomment the text into the method below: string json = JsonConvert.Serialize
74     // With your cursor in "JsonConvert" type (Ctrl + .)
75     // Select to add the Using Newtonsoft.Json
76 0 references | kendrahavens, 20 days ago | 1 author, 1 change
77 public void UsingRefAddedCopyPaste()
78 {
79     //string json = JsonConvert.SerializeObject("some json");
}
```

Solution Explorer

Search Solution Explorer (Ctrl + F)

Solution 'ProductivityFeatures' (2 projects)

- ProductivityFeatures
 - Dependencies
 - Program.cs
 - Update158.cs
 - Update160P1.cs
 - Update160P2.cs
- + XUnitTestProject1

Properties

Ready Ln 62 Col 16 Ch 16 INS ↑ 0 ⌂ 5 Productivity-15.8 ↕ 1ES ▲ 43

Ferramentas

- Instalação da IDE VisualStudio;
- <https://visualstudio.microsoft.com>



Instalação

Instalando – Visual Studio Community 2019 – 16.7.1

Cargas de trabalho **Componentes individuais** **Pacotes de idiomas** **Locais de instalação**

Web e Nuvem (4)

- ASP.NET e desenvolvimento Web
Crie aplicativos Web usando ASP.NET Core, ASP.NET, HTML/JavaScript e Contêineres, incluindo suporte a Docker.
- Desenvolvimento para Azure
SDKs do Azure, ferramentas e projetos para desenvolver aplicativos na nuvem e criar recursos usando o .NET Core ...
- Desenvolvimento Python
Edição, depuração, desenvolvimento interativo e controle do código-fonte para Python.
- Desenvolvimento em node.js
Crie aplicativos de rede escalonáveis usando Node.js, um tempo de execução do JavaScript conduzido por evento...

Área de trabalho e Dispositivos móveis (5)

- Desenvolvimento para desktop com .NET
Crie aplicativos do WPF, do Windows Forms e de console usando C#, Visual Basic e F# com o .NET Core e o .NET...
- Desenvolvimento para desktop com C++
Crie aplicativos C++ modernos para Windows usando as ferramentas de sua escolha, incluindo MSVC, Clang, CMak...
- Desenvolvimento com a Plataforma Universal do Windows
Crie aplicativos para a Plataforma Universal do Windows...
- Desenvolvimento mobile com .NET
Crie aplicativos multiplataforma para iOS, Android ou Windows usando Xamarin.

Local
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community [Alterar...](#)

Ao prosseguir, você concorda com a [licença](#) da edição do Visual Studio selecionada. Também oferecemos a possibilidade de baixar outros softwares com o Visual Studio. Este software é licenciado separadamente, como consta nos [Avisos de Terceiros](#) ou na licença de acompanhamento do produto. Ao prosseguir, você também concorda com essas licenças.

Espaço total necessário 5,25 GB

Instalar durante o download **Instalar**

Introdução ao C#

- **O que é linguagem de programação?**

Esta linguagem é uma língua escrita onde o programador escreve instruções e, após passar por um processo chamado compilação, as instruções se tornam um programa que pode ser executado pelo computador.

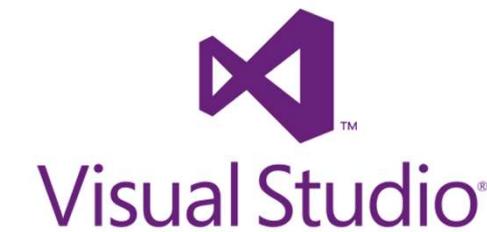
- **Compilação?**

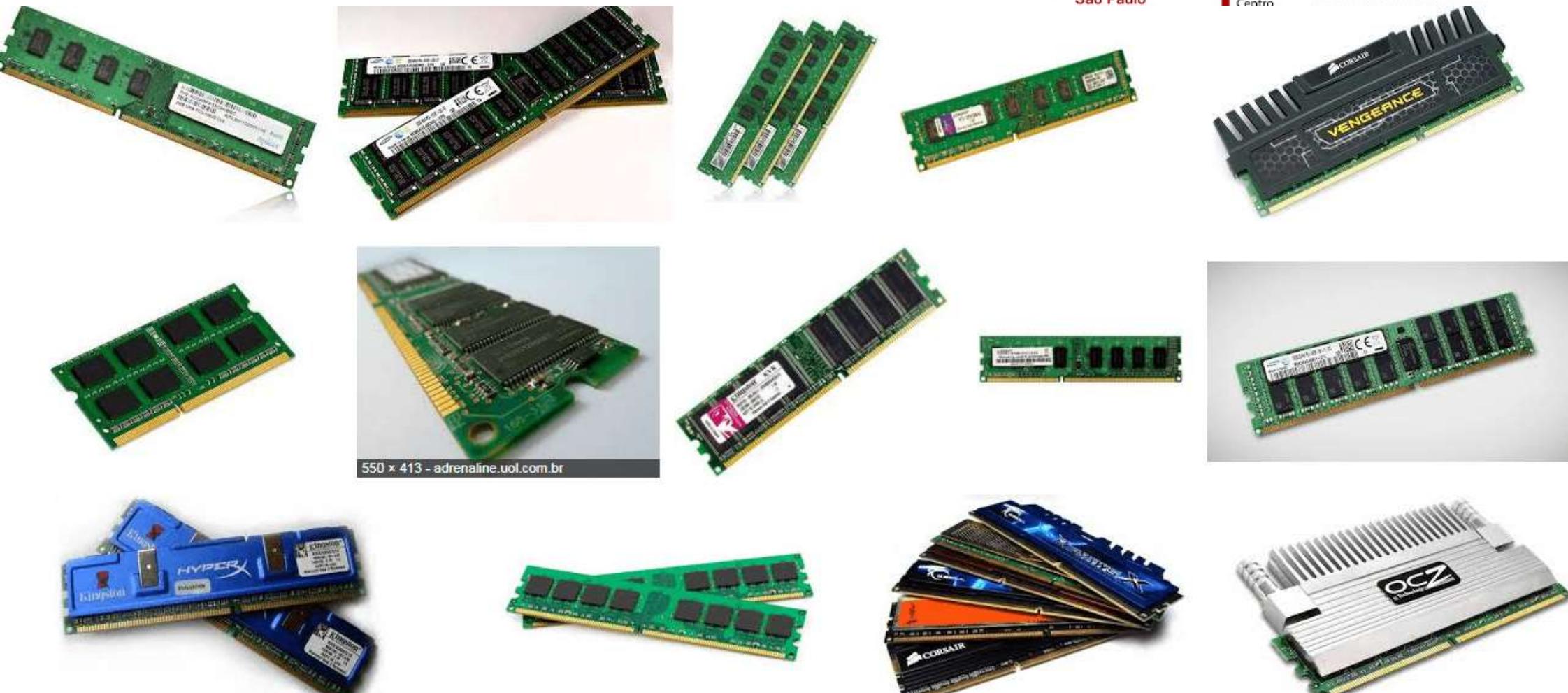
Este processo de compilação consiste em converter os comandos escritos na linguagem de programação em código de máquina

Introdução ao C#

- Microsoft Visual C#

O Microsoft Visual C# é uma IDE (Integrated Development Environment) ou Ambiente de Desenvolvimento Integrado ou seja, uma plataforma de programação completa com o compilador

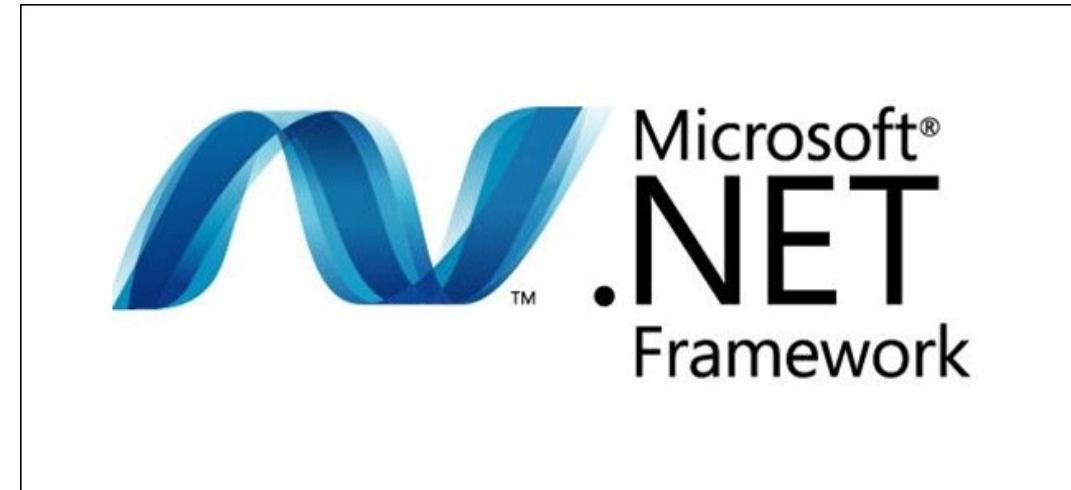


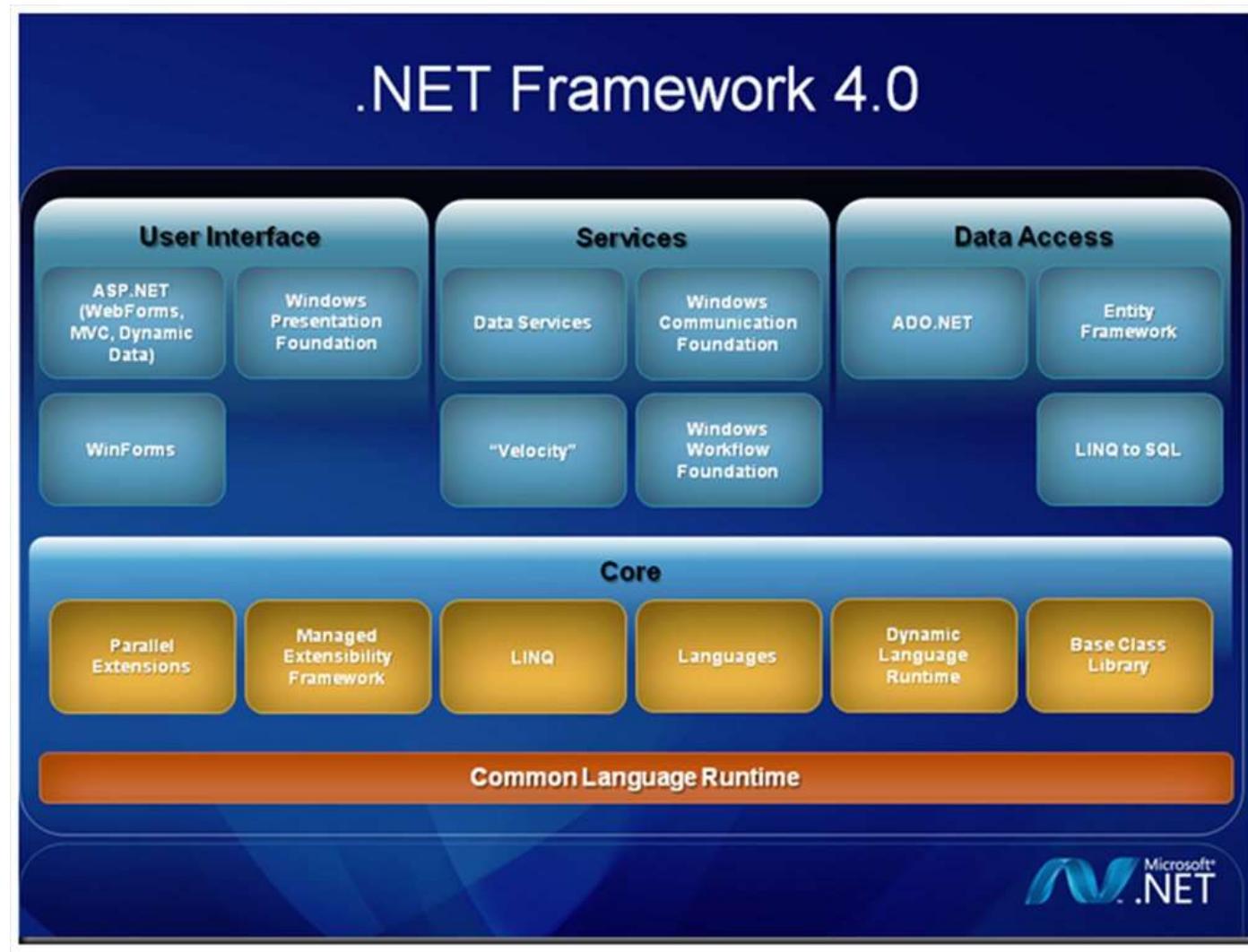


Introdução ao .NET Framework

.NET Framework é um ambiente de execução gerenciado para Windows que fornece uma variedade de serviços para seus aplicativos em execução.

- Gerenciamento de memória.
- Uma biblioteca de classes abrangente.
- Estruturas e tecnologias de desenvolvimento.
- Multiplataforma.





Tipos de dados

Dados é a menor informação fornecida pelo usuário, ou processo computacional.

Informação e a matéria-prima que faz com que seja necessária a existência dos computadores, pois eles são capazes de manipular e armazenar um grande volume de dados

Principais tipos de dados

- Inteiro: toda e qualquer informação numérica que pertença ao conjunto dos números inteiros relativos (negativa, nula ou positiva).
- Real: toda e qualquer informação numérica que pertença ao conjunto dos números reais (negativa, nula ou positiva).
- Caractere: toda e qualquer informação composta por um conjunto de caracteres alfanuméricos .
- Lógico: toda e qualquer informação que pode apenas assumir duas situações, verdadeiro ou falso.

Variáveis

Fatec
São Paulo

CPS
Centro
Paula Souza

SP
GOVERNO DO ESTADO
DE SÃO PAULO

Variáveis são locais onde armazenamos valores na memória. Toda variável é caracterizada por um nome, que a identifica em um programa, e por um tipo, que determina o que pode ser armazenado naquela variável.

Os tipos mais comuns são, inteiro, caractere, real e lógico.

Regras

- Não ter o nome de algo já conhecido pela linguagem (palavra reservada).
- Só pode ser iniciada com uma letra ou com o *underline* (_).
- Não são permitidos espaços, pontos ou caracteres de acentuação.
- Não podemos ter mais de uma variável com o mesmo nome.

Tipo	Valores possíveis de se armazenar
bool	Verdadeiro ou Falso (Valores booleandos)
byte	0 a 255 (8 bits)
sbyte	-128 a 127 (8 bits)
char	Um caractere (16 bits)
decimal	$\pm 1.0 \times 10^{-28}$ a $\pm 7.9 \times 1028$ (128 bits)
double	$\pm 5.0 \times 10^{-324}$ a $\pm 1.7 \times 10308$ (64 bits)
float	$\pm 1.5 \times 10^{-45}$ a $\pm 3.4 \times 1038$ (32 bits)
int	-2,147,483,648 a 2,147,483,647 (32 bits)
uint	0 a 4,294,967,295 (32 bits)
long	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 (64 bits)
ulong	0 a 18,446,744,073,709,551,615 (64 bits)
object	Qualquer tipo.
short	-32,768 a 32,767 (16 bits)
ushort	0 a 65,535 (16 bits)
string	Sequência de caracteres (16 bits por caractere)

Declaração de Variáveis

- double num,r;
- int idade,b,c;
- string nome;

Atribuição de valores

nome = FATEC;

Idade = 40;

num = 34,5;

Exemplo

```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         double a,b,c;
8         a = 10;
9         b = 20;
10        c = a + b;
11        Console.WriteLine(c);
12    }
13 }
```

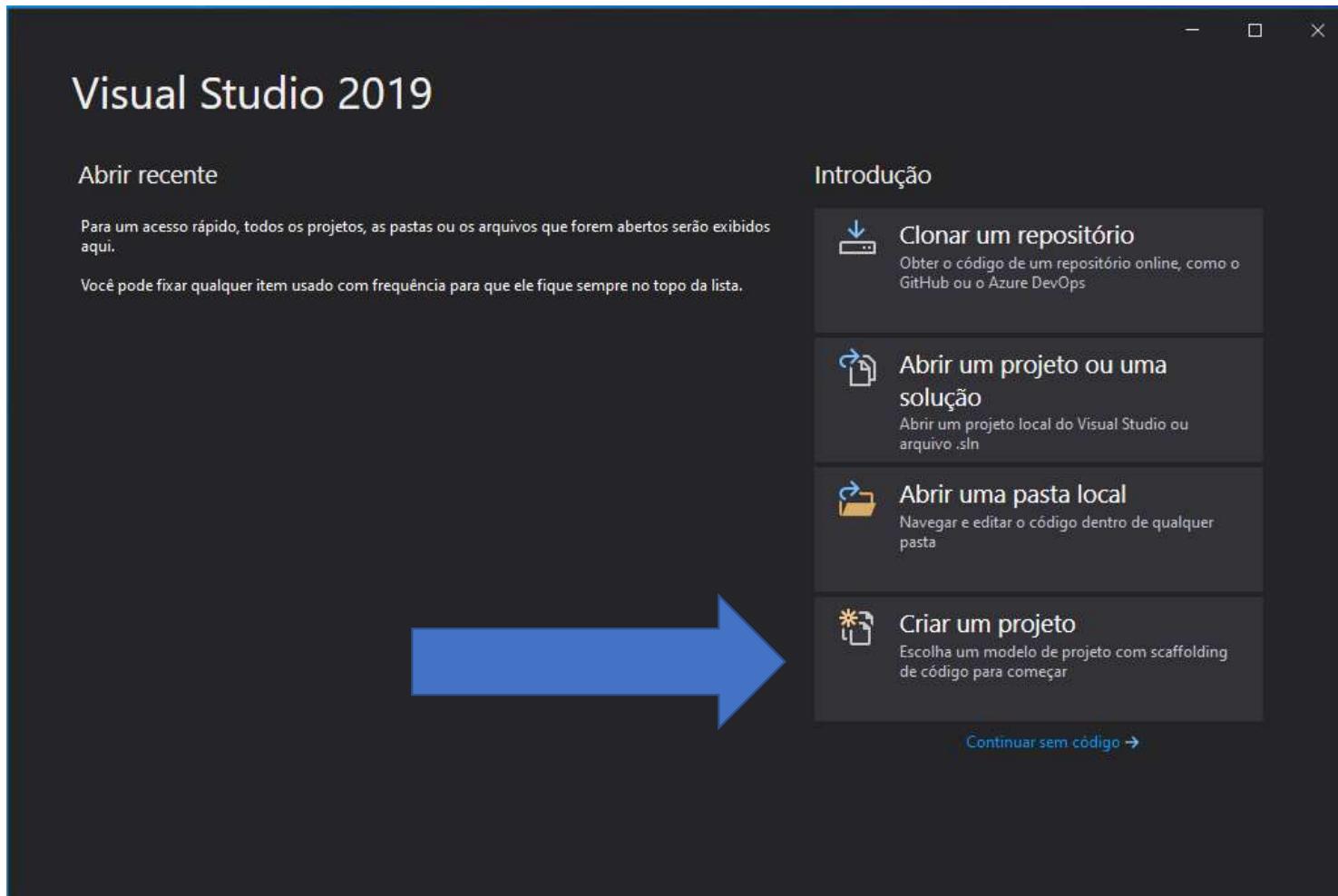
- Console.WriteLine("AQUI"); Tem a função de imprimir uma mensagem na tela de seu programa.

IDE

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

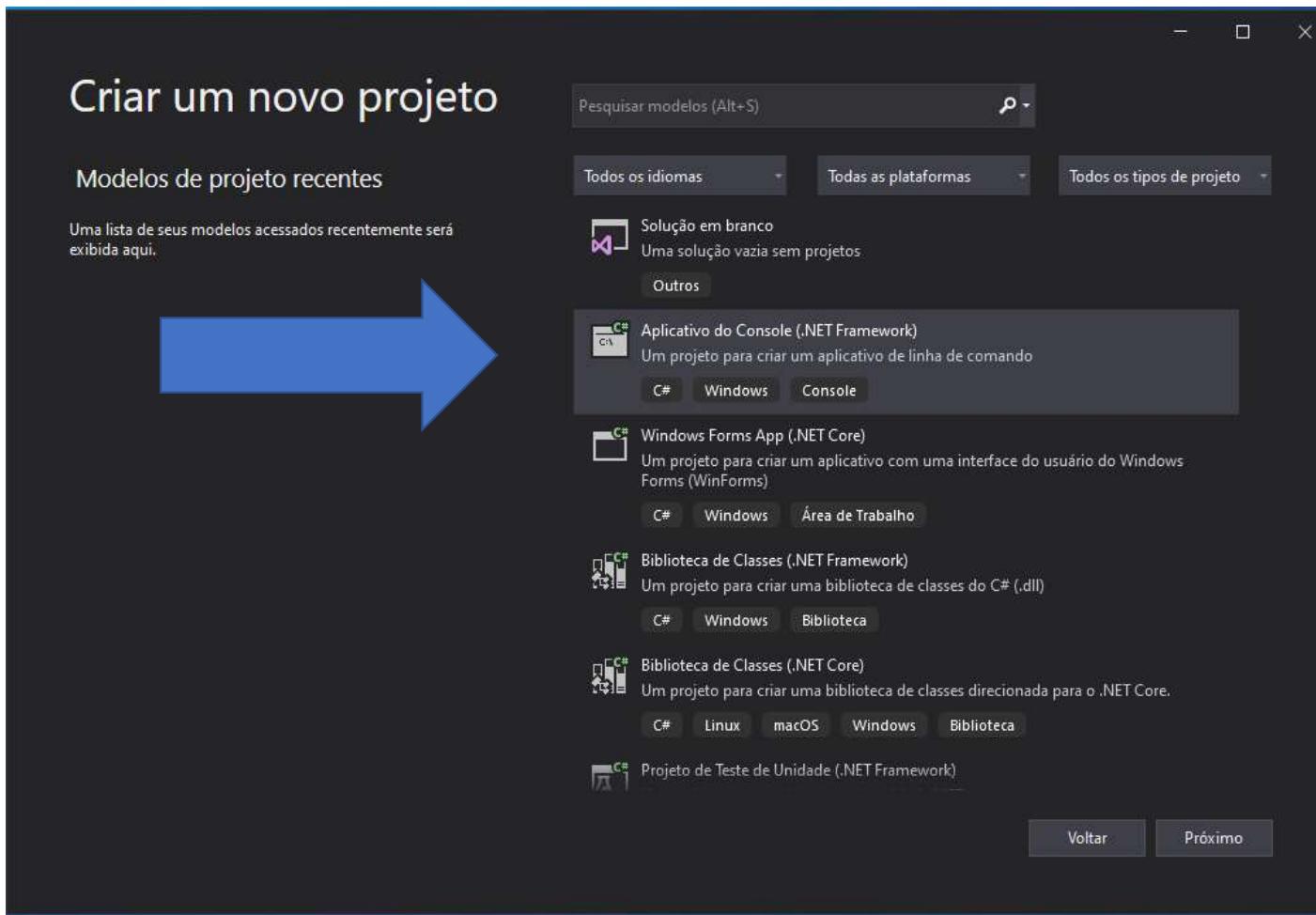


IDE

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

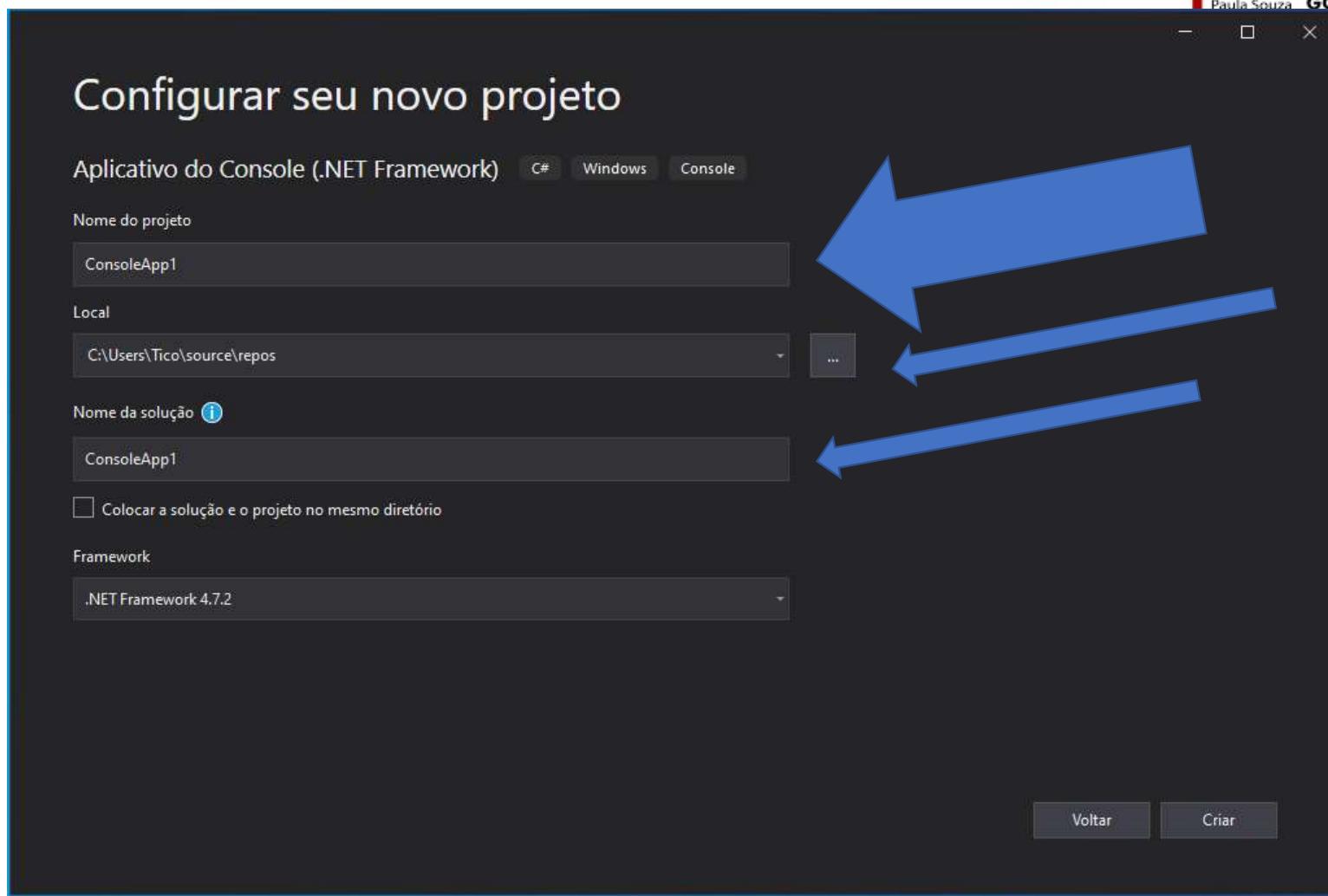


IDE

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO



IDE

Fatec
São Paulo

CPS
Centro
Paula Souza



GOVERNO DO ESTADO
DE SÃO PAULO

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for a C# console application named 'ConsoleApp1'. The code in 'Program.cs' is as follows:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp1
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14     }
15 }
16
```

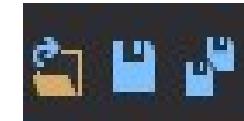
The Solution Explorer on the right shows the project structure:

- Solução 'ConsoleApp1' (1 de 1 projeto)
 - ConsoleApp1
 - Properties
 - Referências
 - App.config
 - Program.cs

The status bar at the bottom indicates: 100% completion, no errors found ('Não foi encontrado nenhum problema'), and file statistics (Ln: 1, Car: 1, SPC, CRLF).

IDE

- São as opções de salvar e abrir novos arquivos.
- Diz respeito ao teste, neste botão rodamos o programa que programamos para testar e operar seus resultados.



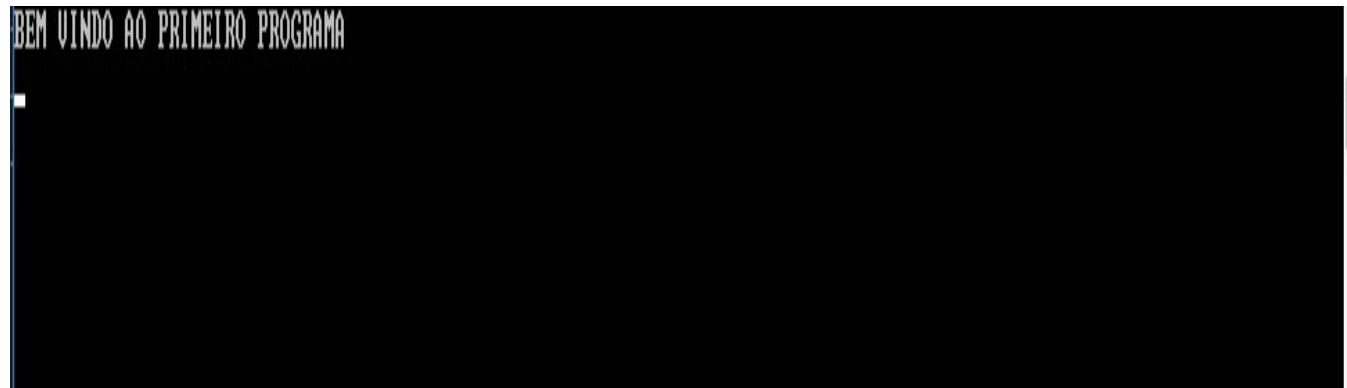
Primeiro Programa

```
namespace primeiroprojeto
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("BEM VINDO AO PRIMEIRO PROGRAMA");
            Console.ReadKey();
        }
    }
}
```

O que estes comandos fazem?

- *Console.WriteLine("AQUI");* Tem a função de imprimir uma mensagem na tela de seu programa.
- *Console.ReadKey();* Trava a tela, espera o usuário digitar alguma coisa antes de fechar o programa.

Resultado



Crie um programa que imprima na tela uma mensagem de vindo e seu nome.

Entrada, Saída e processamento

- Todo tipo de software que construímos, ou programamos, tem em seu interior, uma divisão simples que, mesmo que não percebida, sempre está lá. Esta divisão é:

Entrada, processamento e saída.

Entrada, Saída e processamento

```
double n1, n2, r;  
Console.WriteLine("Digite o primeiro numero");  
n1 = Convert.ToDouble(Console.ReadLine());  
Console.WriteLine("Digite o segundo numero");  
n2 = Convert.ToDouble(Console.ReadLine());  
r = n1 + n2;  
Console.WriteLine("A Soma: "+r);  
Console.ReadKey();
```

Entrada, Saída e processamento

- **Double n1,n2,r;** Nesta linha criamos as **variáveis**. Explicaremos melhor sobre elas adiante mas em um pequeno resumo são como as variáveis matemáticas, uma letra que representa seu conteúdo.
- **Console.WriteLine("Digite o primeiro numero");** Este comando já visto anteriormente simboliza a escrita na tela, escrevendo a mensagem entre as aspas duplas na tela.
- **n1 = Convert.ToDouble(Console.ReadLine());** este comando faz com que o usuário digite uma informação e esta informação será armazenada na variável 'n1'. Este comando pode ser dividido em duas partes:
 - **Convert.ToDouble()** Converte qualquer coisa entre parênteses para o tipo de variável Double(veremos mais sobre isso no próximo capítulo).
 - **Console.ReadLine()** Faz com que o usuário do programa possa digitar alguma informação através do teclado.
- As linhas 4 e 5 são iguais as linhas 2 e 3, com a diferença que o que o usuário digitar será armazenado na variável 'n2'.
- **r = n1 + n2;** Aqui é executada a operação matemática, no caso a variável 'r' irá conter a soma das variáveis 'n1' e 'n2'.
- No caso das linhas 7 e 8 temos a execução de comandos que já conhecemos, um para exibir a resposta e outro para travar a tela até o usuário digitar uma tecla qualquer.



```
1  using System;
2     using System.Collections.Generic;
3     using System.Linq;
4     using System.Text;
5     using System.Threading.Tasks;
6
7     namespace projeto2
8     {
9         class Program
10        {
11            static void Main(string[] args)
12            {
13                double n1, n2, r;
14                Console.WriteLine("Digite o primeiro numero");
15                n1 = Convert.ToDouble(Console.ReadLine());
16                Console.WriteLine("Digite o segundo numero");
17                n2 = Convert.ToDouble(Console.ReadLine());
18                r = n1 + n2;
19                Console.WriteLine("A Soma: "+r);
20                Console.ReadKey();
21            }
22        }
23    }
```

Sites para desenvolver diagramas

- <https://www.gliffy.com/>
- <http://www.draw.io>

Site para desenvolver script

- <https://dotnetfiddle.net/>
- <https://repl.it/languages/csharp>

Prática

- 1 - Desenvolva um programa que permita a entrada do nome do usuário e exiba na tela.
- 2 – Crie um programa e fluxograma que permita ao usuário entrar com dois números. Executar as quatro operações com esses dois números (Soma, Subtração, multiplicação, divisão). Não se esquecer de mostrar as respostas na tela.
- 3 – Desenvolva um programa e fluxograma para calcular a idade de uma pessoa, entrando com o ano de nascimento e ano atual.
- 4- Elaborar um programa e fluxograma para calcular a média aritmética de três números digitados pelo usuário.

Prática

- 5) Escreva um código para ler um valor (do teclado) e escrever (na tela) o seu antecessor.
- 6) Escreva um programa para ler as dimensões de um retângulo (base e altura), calcular e escrever a área do retângulo.
- 7) Faça um código em C# que leia a idade de uma pessoa expressa em anos, meses e dias e escreva a idade dessa pessoa expressa apenas em dias. Considerar ano com 365 dias e mês com 30 dias.
- 8) Faça um código em C# que leia três notas de um aluno, calcule e escreva a média final deste aluno. Considerar que a média é ponderada e que o peso das notas é 2, 3 e 5. Fórmula para o cálculo da média final é:

$$\text{mediafinal} = \frac{n1 * 2 + n2 * 3 + n3 * 5}{10}$$

Funções

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

Uma função nada mais é do que uma sub-rotina usada em um programa.

Denominamos função a um conjunto de comandos que realiza uma tarefa específica em um módulo dependente de código

Por que utilizar função?

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

A utilização de funções visa modularizar um programa, o que é muito comum em programação estruturada.

Principais Funções Matemáticas

Já aprendemos como lidar com as entradas numéricas, agora vamos aprender a fazer cálculos um pouco mais complexos com elas.

Podemos ver que para operações simples é bem fácil de fazer, para contas mais complicadas podemos utilizar um recurso presente no C# que permite utilizar **funções** matemáticas prontas, minimizando o trabalho em vários momentos.

Principais Funções Matemáticas

- Principais Funções Matemáticas em C# – Classe Math:
 - Sin(x), Cos(x), Tan(x): calcula o seno, o cosseno e a tangente de um numero qualquer. Atenção, apenas números radianos.
 - Pow(x,y): Obtém o valor de ‘x’ elevado a ‘y’. Uma forma de calcular potencias.
 - Sqrt(x): Calcula a raiz quadrada de um numero.
 - Ceiling(x), Floor(x): Arredonda o numero para cima ou para baixo.
 - Round(x,y): arredonda o numero ‘x’ em ‘y’ casas decimais.
 - PI: Retorna no número pi.

Exemplo

- Entrar com um número e calcular o quadrado do número e sua raiz.

```
static void Main(string[] args)
{
    double num, raiz, quadrado;
    Console.WriteLine("Digite o numero:");
    num = Convert.ToDouble(Console.ReadLine());
    raiz = Math.Sqrt(num);
    quadrado = Math.Pow(num, 2);
    Console.WriteLine(" O quadrado: " + quadrado + " A raiz: " + raiz);
    Console.ReadKey();
}
```

Exercícios

- 1 – Desenvolva um programa para calcular a área, diâmetro e perímetro do círculo.
 - $A = \pi \cdot r^2$
 - $d=2 \cdot r$
 - $P = 2 \cdot \pi \cdot r$
- 2 – Elabore um programa que calcule a equação de segundo grau utilizando a formula de Bhaskara. OBS: O usuário deve entrar com ‘a’, ‘b’ e ‘c’ e devemos calcular o valor de ‘x1’ e ‘x2’.
- Formula:
 - $\Delta = b^2 - 4 \cdot a \cdot c$
 - $x = (-b \pm \sqrt{\Delta}) / (2 \cdot a)$

Funções Alfanuméricas

- Alfanumérico é um conjunto de caracteres alfabéticos e numéricos, e é utilizado para descrever a coleção de letras latinas e algarismos arábicos, ou um texto construído a partir desta coleção.
- Função Alfanuméricas é o nome utilizado para os caracteres, por exemplo, as letras, os caracteres especiais e até os números.

Exemplos

- Fica mais fácil de entender com um exemplo:
 - No caso de tratarmos como caracteres numéricos, $1 + 1 = 2$.
 - Porem se tratarmos como caracteres alfanuméricos ficariam $1+1 = 11$.

Exemplos de Funções

- +
- ToUpper
- ToLower
- IndexOf
- Substring
- Replace
- Trim

Concatenação.

- A concatenação é quando queremos juntas duas ou mais variáveis, sendo pelo menos uma delas uma variável alfanumérica.
- Esta concatenação é feita utilizando o operador ‘+’ entre estas variáveis.

```
string nome;  
Console.WriteLine("Digite seu nome:");  
nome = Console.ReadLine();  
Console.WriteLine("Bom dia " + nome + " Seja bem vindo.");  
Console.ReadKey();
```

Funções

- Transformar em maiúsculo.

- nome = nome.ToUpper();

```
string nome;
Console.WriteLine("Digite seu nome:");
nome = Console.ReadLine();
nome = nome.ToUpper();
Console.WriteLine("Seu nome em maiúsculo:" + nome);
Console.ReadKey();
```

Funções

- Transformar em minúsculo.

- nome = nome.ToLower();

```
string nome;
Console.WriteLine("Digite seu nome:");
nome = Console.ReadLine();
nome = nome.ToLower();
Console.WriteLine("Seu nome em minúsculo:" + nome);
Console.ReadKey();
```

Funções

- Encontrar um carácter dentro da string

- pos = fra.IndexOf("o");

```
string fra;
int pos;
Console.WriteLine("Digite uma frase:");
fra = Console.ReadLine();
pos = fra.IndexOf("o");
Console.WriteLine("a posição da letra 'o' e:" + pos);
Console.ReadKey();
```

Funções

- **Dividir uma string em duas.**
 - parte = completo.Substring(4, 8);

```
string completo,parte;
Console.WriteLine("Digite uma frase:");
completo = Console.ReadLine();
parte = completo.Substring(4, 8);
Console.WriteLine("A parte quebrada e:" + parte);
Console.ReadKey();
```

Funções

- **Substituir caractere de uma string.**

- fra = fra.Replace("b", "j");

```
string fra;
Console.WriteLine("Digite uma palavra:");
fra = Console.ReadLine();
fra = fra.Replace("o", "a");
Console.WriteLine("substituindo o 'o' pelo 'a': "+fra);
Console.ReadKey();
```

Funções

- Remove todos os caracteres de espaço em branco à esquerda e à direita da cadeia de caracteres atual.
 - Trim();

```
String texto;
Console.WriteLine("Entrar com o algum texto");
texto = Console.ReadLine();
texto = texto.Trim();
Console.WriteLine(texto);
Console.ReadKey();
```

Exercícios

3 – Elaborar um programa que permita que o usuário escreva seu nome completo. Mostrar na tela:

- Nome e o sobrenome separadamente.
- O conteúdo da string do caractere cinco ao dez.
- Substituir todos os caracteres ‘a’ pelo ‘o’.

4 – Crie um gerador de e-mail que solicite o nome do usuário completo e mostre ao final o email: nome@fatec.sp.gov.br

5 – DESAFIO(Crie um gerador de e-mail que solicite o nome do usuário completo e mostre ao final o email: nome.ultimosobrenome@fatec.sp.gov.br

Layouts e cores

Fatec
São Paulo

CPS
Centro
Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

Layout é a maneira de exibição do programa ou sistema para o usuário.

Designer, Responsável por desenvolver projetos gráficos ou de comunicação visual.

A área de Interação Humano-Computador (IHC) se dedica a investigar a interação entre pessoas e artefatos de base computacional.



Exemplos

The screenshot shows a web-based application interface for 'CADASTRO DE EQUIPAMENTOS' (Equipment Registration). The top navigation bar includes the ArcelorMittal logo, a menu with links to PRINCIPAL, CADASTRO, OPERAÇÃO, RELATÓRIOS, GESTÃO, and SOBRE, and the LYNX Process logo. A 'CRIAR' button is visible on the left. The main form contains six dropdown menus labeled 'Nome' and two buttons at the bottom: 'Limpar' (Clear) and 'Buscar' (Search).

PRINCIPAL CADASTRO OPERAÇÃO RELATÓRIOS GESTÃO SOBRE

CRIAR

CADASTRO DE EQUIPAMENTOS

Nome Nome

Nome

Nome

Nome

Nome

Limpar Buscar

© Copyright 2016 by LYNX PROCES | Todos direitos reservados



SISTEMA DE CONTROLE DE CHAMADOS

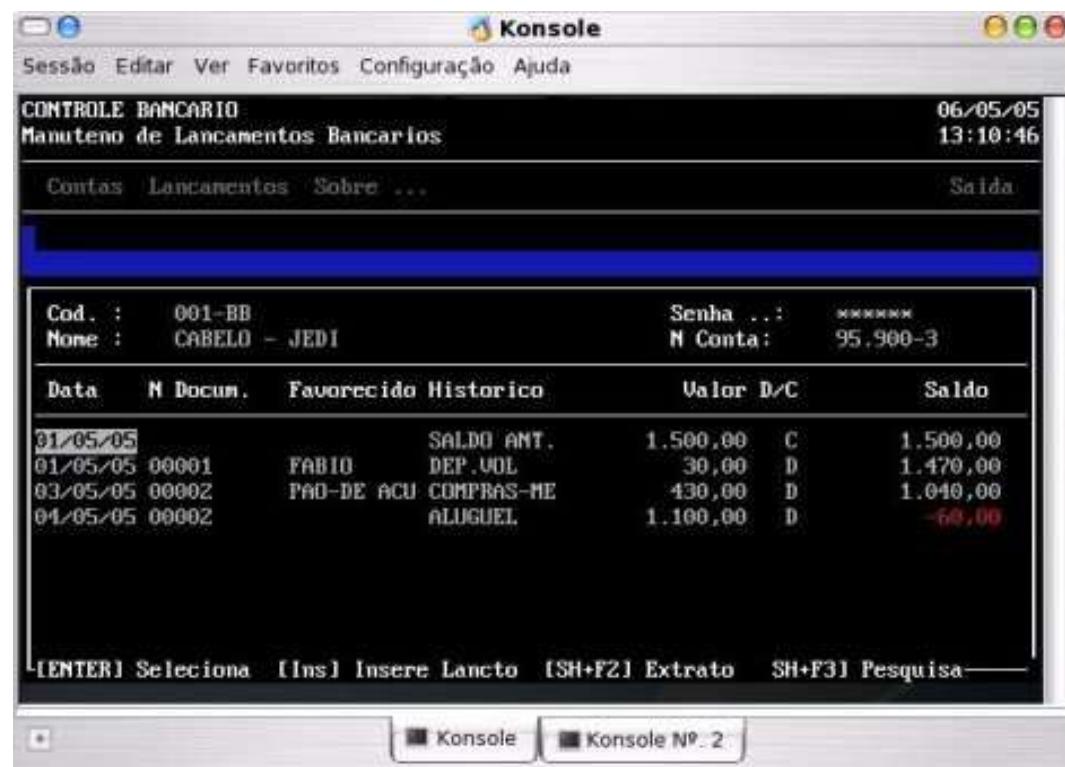
ABRIR CHAMADO
CONSULTAR
CONFIGURAÇÕES

Filtro

Código	Data Abertura	Data Resolução	Status	Cliente	Unidade
<input type="text"/>	<input type="text"/>	<input type="text"/>	Selecionar	<input type="text"/>	<input type="text"/>
Área	Tecnologia	Problema	Gravidade	Prioridade	SLA
Selecionar	Selecionar	Selecionar	Selecionar	Selecionar	Selecionar

Pesquisar
 Limpar/Redefinir

Código	Descrição	Data Abertura	Data Resolução	Unidade	Problema	Tecnologia	Área	SLA	Status	Visualizar
00553	realizar faturamento diário dos clientes.	05/05/2016 ...	05/05/2016 ...	Figueira	Faturamento D...	Faturamento	Faturamento	-	Resolvido	
00552	Validação diária dos clientes recebidos via e-...	05/05/2016 ...	05/05/2016 ...	Figueira	Validação de C...	PI System	Sala de Cont...	-	Resolvido	
00551	Favor verificar o status de "configure" na tag...	05/05/2016 ...	-	Figueira	Interfaces PI S...	PI System	Sala de Cont...	-	Atendimento	
00550	Após realizado no banco de dados, favor rea...	05/05/2016 ...	-	Figueira	Retrocesso de...	PI System	Sala de Cont...	-	Aberto	
00549	Favor realizar o insert dos dados da Gerdau...	05/05/2016 ...	-	Figueira	Insert de dados	Banco de Dados	Sala de Cont...	-	Aberto	
00548	Favor verificar a estação ERPTA2103, a me...	05/05/2016 ...	05/05/2016 ...	Figueira	Outros	CELLO OPC	Sala de Cont...	-	Resolvido	
00547	Favor verificar as tags que se encontram co...	04/05/2016 ...	-	Figueira	Interfaces PI S...	PI System	Sala de Cont...	-	Atendimento	
00546	A ERPP000059 está com o parâmetro de alar...	04/05/2016	04/05/2016	Figueira	Outros	PI System	Sala de Cont...	-	Resolvido	





MS
CG Prompt do MS-DOS - FAT



< MASTER >

< TESTE > V-2.13

Tabelas Estoque Vendas Faturamento Relatórios Utilitários Sair

Nota Fiscal
Duplicatas
Emissão Notas
Emissão Boleto
Mod.Tabulações
Relatórios
Cobr.Eletrônica
Carta Correção

Emissão de Notas
Emissão de Boletos
Emissão de Duplicata
Duplicatas p/Emissão
Duplicatas p/Vencimento
Faturamento p/Grupo
Faturamento p/Cliente
Estatística-Faturam.
Registro de Notas
Notas por Natureza
Etiqueta por Nome
Etiqueta por Tipo
Etiqueta por Nota
Produtos Faturados
Faturam. p/Vendedor

Estatística dos Produtos Faturados.

Layouts e cores

Podemos deixar nossos programas mais intuitivos, utilizando melhorando o layout adicionando cores e posicionamento.

- **Cor da letra**

- Console.ForegroundColor

- **Cor do fundo**

- Console.BackgroundColor

- **Posição na tela**

- SetCursorPosition(col,lin);

Cor da letra

```
Console.ForegroundColor = ConsoleColor.Red;  
Console.WriteLine("Bem vindo!");  
Console.ReadKey();
```



- Black – Preto.
- Blue – Azul.
- Cyan – Cyano.
- Grey – Cinza.
- Green – Verde.
- Magenta – Rosa.
- Red – Vermelho.
- White – Branco.
- Yellow – Amarelo.
- DarkBlue – Azul Escuro.
- DarkCyan – Cyano Escuro.
- DarkGrey – Cinza Escuro.
- DarkGreen – Verde Escuro.
- DarkMagenta – Rosa Escuro.
- DarkRed – Vermelho Escuro.
- DarkYellow – Amarelo Escuro.

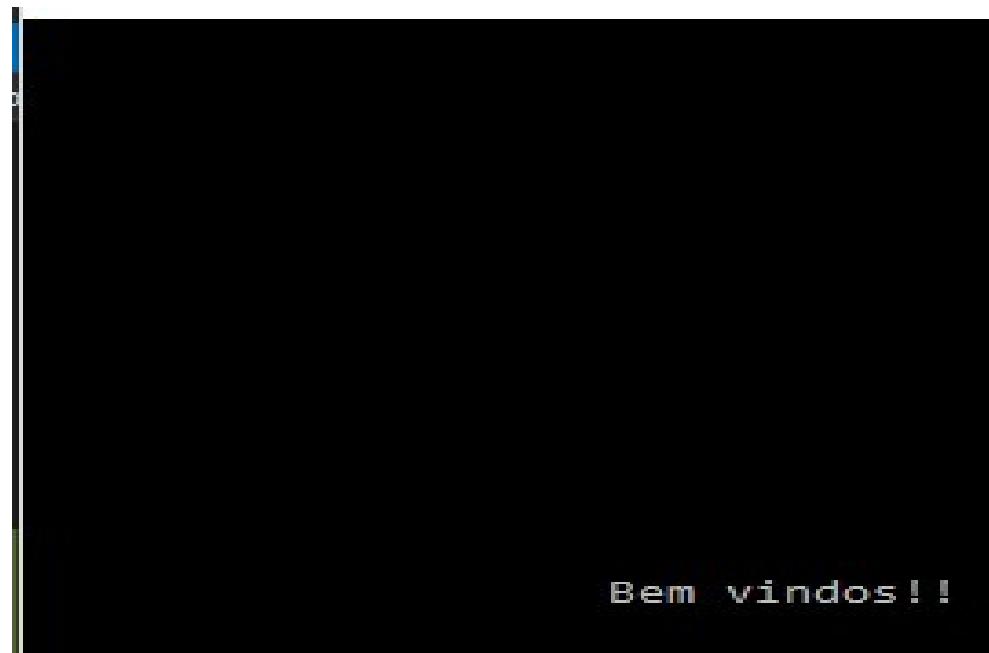
Cor do fundo

```
Console.BackgroundColor = ConsoleColor.Blue;  
Console.Clear();  
Console.WriteLine("Bem vindos!!");  
Console.ReadKey();
```

Bem vindos!!

Posição na Tela

```
Console.SetCursorPosition(20, 15);  
Console.WriteLine("Bem vindos!!");  
Console.ReadKey();
```



Prática

Utilizar comandos de cor de fundo, cor de letra e moldura.

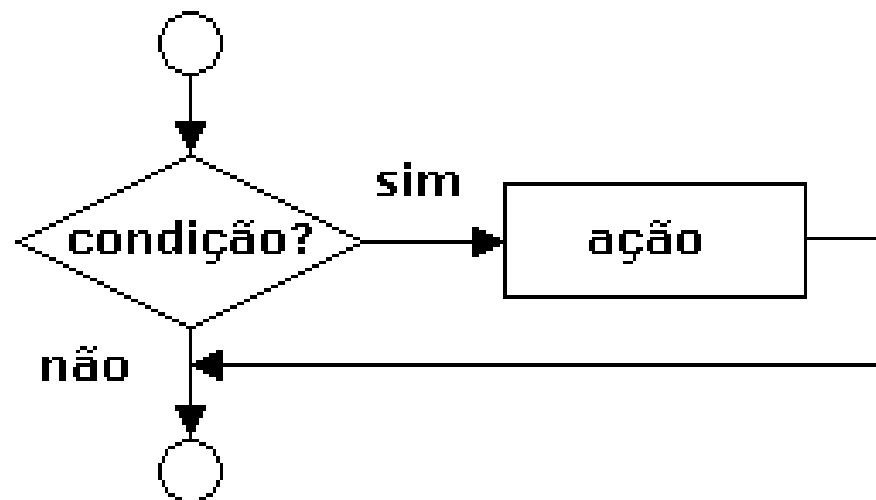
- 1 – Criar um programa que permita a digitação do salário mínimo e do salário do usuário, calcular quantos salários mínimos a pessoa ganha.
- 2 – Elaborar um programa que permita que o usuário digite uma temperatura em graus Celsius e o programa converta para Kelvin.
- 3 - Escreva um algoritmo para ler uma temperatura em graus Fahrenheit, calcular e escrever o valor correspondente em graus Celsius
- 4 – Criar um programa que permita que o usuário digite seu peso em kg, calcular e mostrar:

Peso em gramas.

O peso com um acréscimo de 10%

Estrutura de decisão IF

Esta estrutura tem uma finalidade simples, nos permitindo tomar decisões dentro de nossos programas e guiar o programa por caminhos diferentes dependendo da **condição** expressa.



Estrutura de decisão IF

Operadores relacionais	Descrição.
<code>==</code>	Igual
<code>!=</code>	Diferente
<code>></code>	Maior
<code><</code>	Menor
<code>>=</code>	Maior Igual
<code><=</code>	Menor Igual

Exemplo de operadores relacionais

a = 15

b = 5

c = 3

d = 15

a > b

c > a

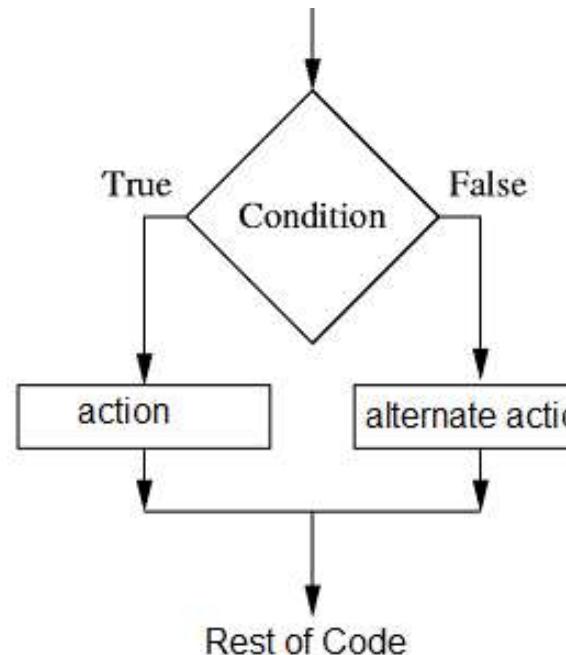
d <= c

a == d

a >= d

IF-ELSE

O if-else, pode ser visto como uma extensão do comando if "se isso for verdadeiro, então faça aquilo, caso contrário, faça outra coisa". Ele segue o seguinte formato:



```
if(condição){  
}  
}else{  
}  
}
```

Exemplo

```
double n1, n2, media;
Console.WriteLine("Digite a primeira nota:");
n1 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Digite a segunda nota:");
n2 = Convert.ToDouble(Console.ReadLine());
media = (n1 + n2) / 2;
Console.WriteLine("A media:" + media);
if (media >= 6)
{
    Console.WriteLine("Aprovado");
}
else
{
    Console.WriteLine("Reprovado");
}
Console.ReadKey();
```



GOVERNO DO ESTADO
DE SÃO PAULO

```
Digite a primeira nota:
7
Digite a segunda nota:
5
A media:6
Aprovado
```

```
Digite a primeira nota:
4
Digite a segunda nota:
5
A media:4,5
Reprovado
```

IF-ELSE-IF

```
//inicio
    Console.Title = "IF/ELSE/IF";
    Console.WriteLine("Digite uma opção de 0 a 3:");
    int op = Convert.ToInt32(Console.ReadLine());
    if (op == 0)
    {
        Console.WriteLine("Zero");
    }
    else if (op == 1)
    {
        Console.WriteLine("Um");
    }
    else if (op == 2)
    {
        Console.WriteLine("Dois");
    }
    else if (op == 3)
    {
        Console.WriteLine("Três");
    }
    else {
        Console.WriteLine("Opção Incorreta");
    }
    Console.ReadKey();
}//fim
```

Uma construção comum em programação é a forma if-else-if. Para exemplificar vamos reescrever o programa

Operador MOD %

Este operador tem a função de pegar o resto de uma divisão inteira.

$$6 \% 2 = 0$$

$$5 \% 2 = 1$$

$$8 \% 5 = 3$$

$$7 \% 6 = 1$$

```
Digite um numero:  
22  
PAR
```

```
Digite um numero:  
5  
IMPAR
```

```
double n, resto;  
Console.WriteLine("Digite um numero:");  
n = Convert.ToDouble(Console.ReadLine());  
resto = n % 2;  
if (resto == 0)  
{  
    Console.WriteLine("PAR");  
}  
else  
{  
    Console.WriteLine("IMPAR");  
}  
Console.ReadKey();
```

Exercícios

5 – Criar um programa que permita que o usuário digite um número, verificar se este número é maior que 20, se for maior mostrar na tela a metade do número digitado, caso seja menor 20 mostrar o dobro do número digitado.

6 – Criar um programa que permita que o usuário digite um número, verificar se o mesmo é múltiplo de 5.

7 – A prefeitura esta concedendo empréstimos a seus funcionários seguindo a regra de que a prestação não pode ser maior que 30% do valor do salário. Criar um programa que permita a entrada do salário e da prestação do empréstimo, verificar se a prestação digitada corresponde a regra de empréstimo descrita acima.

8 – Criar um programa que permita que o usuário digite um número, verificar e mostrar na tela se o número é **positivo, negativo ou zero**.

Fatec
São Paulo

CPS
Centro
Paula Souza



**GOVERNO DO ESTADO
DE SÃO PAULO**