

Desenvolvimento de Jogos

Linguagem de programação

PROF. ME. HÉLIO ESPERIDIÃO

O que é um Algoritmo?

- ✓ É qualquer procedimento computacional bem definido que informa algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída.

Programas

- ✓ Um computador nada mais faz do que executar programas.
- ✓ Um programa é simplesmente uma sequencia de instruções definida por um programador em uma linguagem de especifica que pode ser executada em um computador.
- ✓ Um programa pode ser comparado a uma receita que indica os passos elementares que devem ser seguidos para desempenhar uma tarefa.

Programas

- ✓ Cada instrução é executada no computador por seu principal componente, o processador ou CPU (Unidade Central de Processamento).
- ✓ Formalmente um programa:
 - ✓ É um conjunto de instruções que descrevem uma tarefa a ser realizada por um dispositivo computacional.



Linguagem de programação

- ✓ É um método padronizado para comunicar instruções que serão processadas em um processador.
- ✓ Permite que um programador especifique precisamente em quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas em circunstâncias pré determinadas
- ✓ Exemplos:
 - ✓ C
 - ✓ C++
 - ✓ Java

Multiplataforma

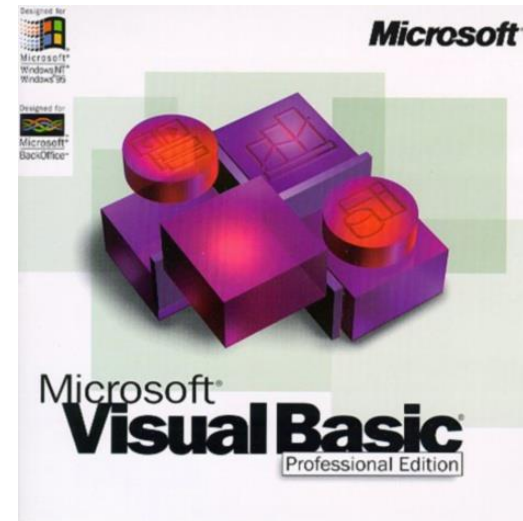
- ✓ São aquelas linguagens que podem ser executadas em sistemas operacionais distintos.



Não portáteis

- ✓ São aquelas desenvolvidas para um equipamento ou sistema operacional específico.
- ✓ Exemplos:
 - ✓ Visual Basic
 - ✓ Plataforma .net da Microsoft

Microsoft®
.net™

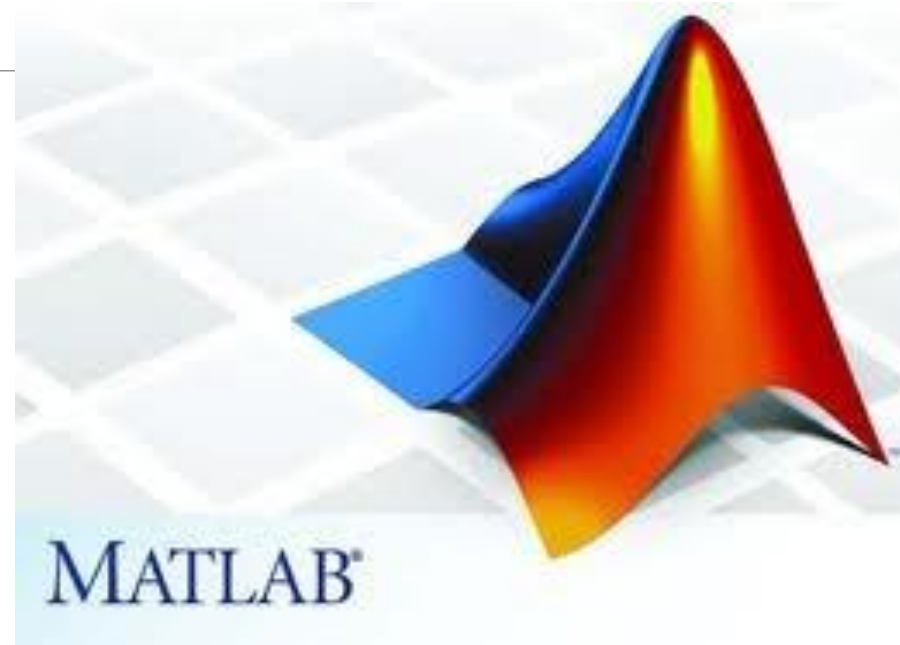


Programação de alto nível

- ✓ É uma linguagem com um nível de abstração relativamente elevado, longe do código de máquina e mais próximo à linguagem humana.
- ✓ O programador de uma linguagem de alto nível não precisa conhecer características do processador, como instruções e registradores. Essas características são abstraídas na linguagem de alto nível.

Linguagens de alto nível:

- ✓ ASP
- ✓ Java
- ✓ MATLAB
- ✓ PHP
- ✓ Visual Basic



Programação de baixo nível

- ✓ É uma linguagem de programação que não abstrai as características da arquitetura do computador.
- ✓ É necessário conhecer detalhes físicos da máquina que está sendo programada.
- ✓ As linguagens de baixo nível estão diretamente relacionadas com a arquitetura do computador.
- ✓ Exemplo: Assembly.

Baixo nível VS Alto nível

	Vantagens	Desvantagens
Baixo nível	Programas são processados mais rapidamente. Ocupam menos espaço na memória	Programas tem pouca portabilidade, ou seja, são específicos para cada tipo de processador) e não são estruturados
Alto Nível	Programas tem maior portabilidade e as linguagens são mais estruturadas	Programas exigem mais tempo de processamento

Programação de baixo nível

- ✓ É uma linguagem de programação que não abstrai as características da arquitetura do computador.
- ✓ É necessário conhecer detalhes físicos da máquina que está sendo programada.
- ✓ As linguagens de baixo nível estão diretamente relacionadas com a arquitetura do computador.
- ✓ Exemplo: Assembly.

Exemplo de Algoritmo

- ✓ Um exemplo simples de um problema que pode ser resolvido por meio de uma sequencia de passos é uma receita de bolo.
- ✓ A Montagem de uma casa de lego.
- ✓ A construção de um carro



Algoritmos e programas

- ✓ Algoritmos não utilizam uma linguagem de programação formal.
- ✓ Um programa de computador é a formalização de um algoritmo em qualquer linguagem de programação.

Características de um programa

- ✓ De forma geral a maioria dos programas são compostos por dados de entrada, processamento e dados de saída.



Termos utilizados

Termo	Descrição
Compilar	Transformar código fonte em código de máquina.
Rodar	é o mesmo que executar
Executar	Carregar o programa na memória e iniciar o processamento
Em execução	O programa está na memória, pode estar ou não sendo executado pelo processador
Debug	Análise linha a linha do código a procura de erro
“Debugar”	O mesmo que Debug
Case Sensitive	Maiúsculas e minúsculas fazem diferença

IDE

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

Mais utilizadas

Posição	Linguagem de programação	Avaliação
1	Java	17.050%
2	C	16.523%
3	C#	8.653%
4	C++	7.853%
5	Objective-C	7.062%
6	PHP	5.641%
7	(Visual) Basic	4.315%
8	Python	3.148%
9	Perl	2.931%
10	JavaScript	2.465%

Variáveis

- ✓ Definem um espaço de memória para armazenar dados.
- ✓ Possuem tipos específicos para armazenamento de dados.
- ✓ Um espaço destinado para guardar números é diferente de um espaço criado para guardar letras ou caracteres.

$$X = 5$$

$$Y = 3 * X$$

Variáveis

- ✓ O nome de uma variável deve começar com uma letra
- ✓ Os caracteres subsequentes devem ser letras, números ou sublinhado
- ✓ O nome de uma variável não pode ser igual a uma palavra reservada nem igual ao nome de uma função declarada pelo programador

Variáveis

- ✓ Todas as variáveis devem ser declaradas antes de serem usadas.

Tipos Básicos

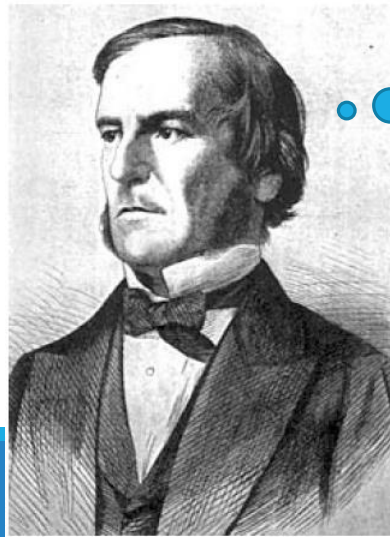
- ✓ Caractere
 - ✓ O valor armazenado é um caractere ou um conjunto de caracteres(Letras).
- ✓ Inteiro
 - ✓ Número inteiro é o tipo padrão e o tamanho do conjunto que pode ser representado normalmente depende da máquina em que o programa está rodando.

Tipos Básicos


- ✓ Real:
 - ✓ Número em ponto flutuante de precisão simples. São conhecidos normalmente como números reais.
- ✓ Lógico:
 - ✓ Verdadeiro ou falso

Revisão da numeração Computacional.

- ✓ O matemático inglês George Boole (1815-1864) publicou em 1854 os princípios da lógica booleana.
- ✓ Segundo Boole tudo poderia ser representado utilizando apenas os números 0 e 1.



George Boole

A blue thought bubble with a white border, containing three lines of binary code. The bubble is connected to the portrait of George Boole by three small blue circles of increasing size.

```
010000111010101011  
110110101010110101  
010110101010101101
```


Bit

- ✓ Simplificação de “dígito binário”(Binary digit em inglês)
- ✓ É a menor unidade de informação que pode ser armazenada ou transmitida.
- ✓ Um bit pode assumir somente 2 valores, por exemplo: 0 ou 1, verdadeiro ou falso.

Byte

- ✓ Um byte nada tem de especial, é apenas um número binário de oito algarismos

0 1 0 1 0 1 1 1

Bytes

- ✓ 1 Byte é representado por uma cadeia de 8 bits

1 byte = 8 bits

1024 bytes = 1 K byte

1.048.576 bytes = 1 Mega byte

Noção de tamanho

Bit		2^0	0 ou 1 = 1 bit
Byte		2^3	8 bits
Kilo	1 Kbyte	2^{10}	1024 Bytes
Mega	1 Mbyte	2^{20}	1 024 kB
Giga	1 Gbyte	2^{30}	1 024 MB
Tera	1 Tbyte	2^{40}	1 024 GB
peta	1 Pbyte	2^{50}	1 024 TB
Exa	1 Ebyte	2^{60}	1 024 PB
Zetta	1 Zbyte	2^{70}	1 024 EB
Yotta	1 Ybyte	2^{80}	1 024 ZB

Tipos de dados

Tipo	descrição	Bits	
byte	Inteiro sem sinal	8	0 a 255
sbyte	inteiro com sinal com sinal	8	-128 a 127
int	inteiro com sinal com sinal	32	-2,147,483,648 to 2,147,483,647
uint	Inteiro sem sinal	32	0 a 4294967295
short	inteiro com sinal com sinal	16	-32.768 a 32.767
long	inteiro com sinal com sinal	64	-922337203685477508 to 922337203685477507
ulong	Inteiro sem sinal	64	0 a 18446744073709551615

Tipo	descrição	Bits	
float	Tipo de ponto flutuante de precisão simples	32	-3.402823e38 para 3.402823e38
double	Tipo de ponto flutuante de precisão dupla	64	-1.79769313486232e308 para 1.79769313486232e308
char	Um único caractere Unicode	16	Unicode símbolos usados no texto
bool	Tipo booliano lógico	8	True ou false
object	tipo de base de todos os outros tipos		
string	Uma sequência de caracteres		
decimal	Preciso tipo fracionário ou integral que pode representar números Decimal com 29 dígitos significativos	128	$\pm 1.0 \times 10e-28$ para $\pm 7.9 \times 10e28$

Importância da escolha correta do tipo de dados

Economia de memória.

Economia de processador.

Economia de Disco.

Qual o resultado da economia?

Expressões Aritméticas

Operador	Descrição	Hierarquia da Operação
+	Soma	3º
-	Subtração	3º
/	Multiplicação	2º
*	Divisão	2º
()	Parênteses	1º 1º

Horizontalização

- ✓ Para o desenvolvimento de algoritmos que possuam cálculos matemáticos, as expressões aritméticas devem estar horizontalizadas

Matemática Tradicional	Algoritmo
$M = \frac{N1 + N2}{2}$	$M \leftarrow (N1 + N2) / 2$

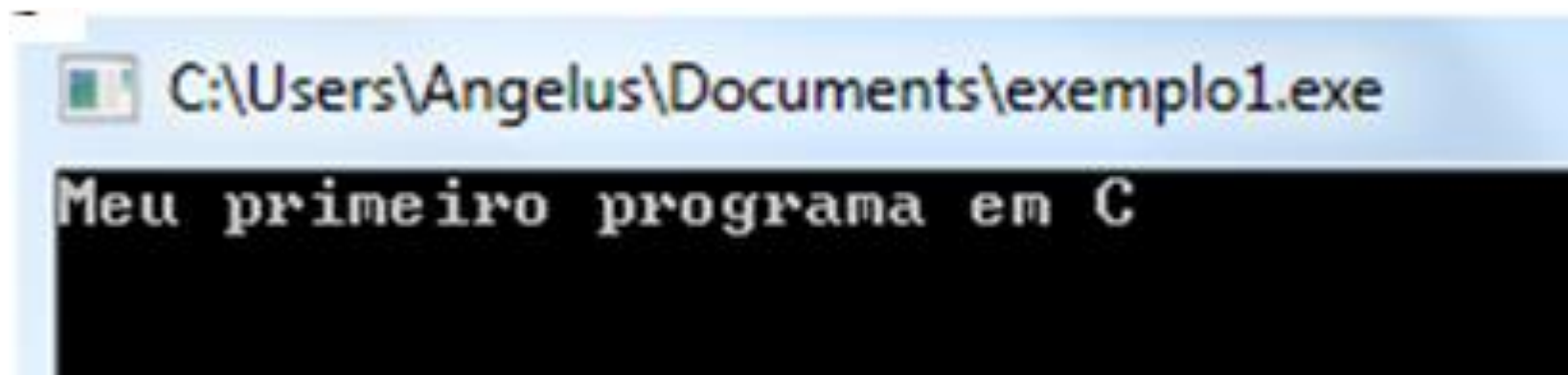
- ✓ As expressões matemáticas na forma horizontalizada não são apenas utilizadas em algoritmos, mas também na maioria das linguagens de programação

Vantagens do C#

- ✓ Faça um aplicativo RAPIDAMENTE.
- ✓ Faça uma interface de usuário com boa aparência.
- ✓ Crie e interaja com bases de dados.
- ✓ Concentre-se em resolver seus problemas REAIS.

Olá mundo em c

```
#include<stdio.h>
main() {
    printf("Meu primeiro programa em C");
    getch();
}
```



Ola mundo

Bibliotecas

```
]using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

```
]namespace ConsoleApplication1  
{  
] class Program  
  {  
]     static void Main(string[] args)  
      {  
          Console.WriteLine("Ola mundo");  
          Console.ReadLine();  
      }  
  }  
}
```

Primeiro método
que é executado

Ola mundo

```
#include<stdio.h>
main() {
    printf("Meu primeiro programa em C");
    getch();
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Ola mundo");
            Console.ReadLine();
        }
    }
}
```

Declarando variáveis.

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            string nome;
        }
    }
}
```

The diagram illustrates the components of the variable declaration `string nome;` within the `Main` method. A blue box labeled "Tipo do dado" (Data type) has an arrow pointing to the word `string`. Another blue box labeled "Nome da variável" (Variable name) has an arrow pointing to the word `nome`. The word `nome` is underlined in the code.

Ler valores

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine(x.ToString());
        }
    }
}
```

Conversão de tipos

No exemplo anterior vimos que os tipos de dados são incompatíveis, portanto é necessário fazer conversões.

As conversões são tipicamente

- De inteiro, real, double para string.
- De string para real, float ou double.

Operadores Aritméticos

Operador	Função
+	Soma
-	Subtração
/	Divisão
%	Resto da divisão
*	multiplicação.

Operadores Relacionais

Operador	Função	Exemplo
==	Comparação	x==y
>	Maior que	x>y
<	Menor que	x<y
>=	Maior ou igual a	x>=y
<=	Menor ou igual a	x<=y
!=	Diferente de	x!=y

Considere: x=1; y=3; z=true;

Operadores de atribuição

Operador	Função	Exemplo
=	Atribuição Simples	X=5;
+=	Atribuição aditiva	X+=5;
-=	Atribuição Subtrativa	X-=5;
=	Atribuição Multiplicativa	X=5;
/=	Atribuição de divisão	X/=5;
%=	Atribuição de módulo	X%=5;

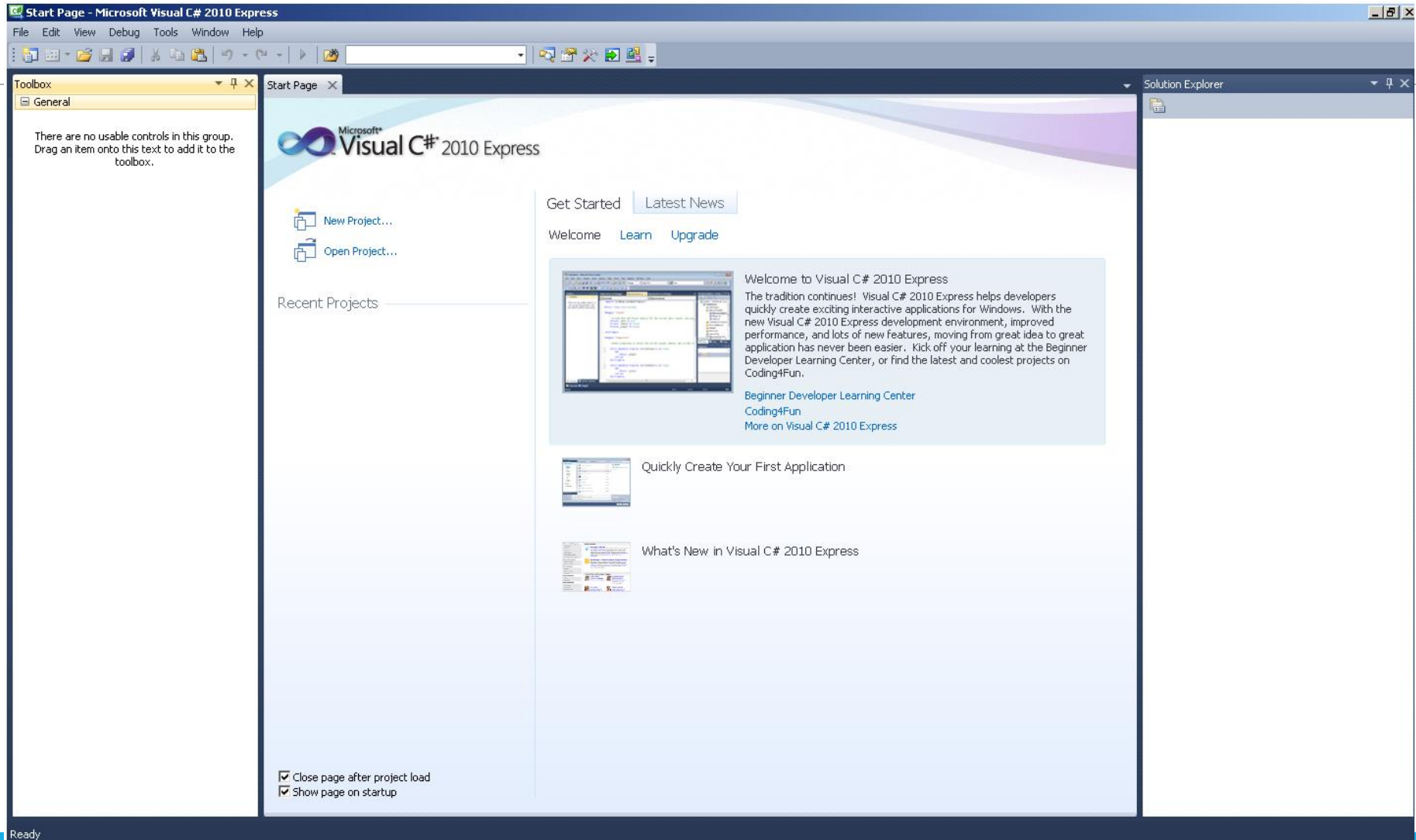
Operadores Lógicos

Operador	Função	Exemplo
&&	.e. Lógico.	(A>B) && (A<B)
	.ou. Lógico.	(A>B) (A<B)
!	Negação	!z

Expressões Aritméticas

Operador	Descrição	Hierarquia da Operação
+	Soma	3º
-	Subtração	3º
/	Multiplicação	2º
*	Divisão	2º
()	Parênteses	1º 1º

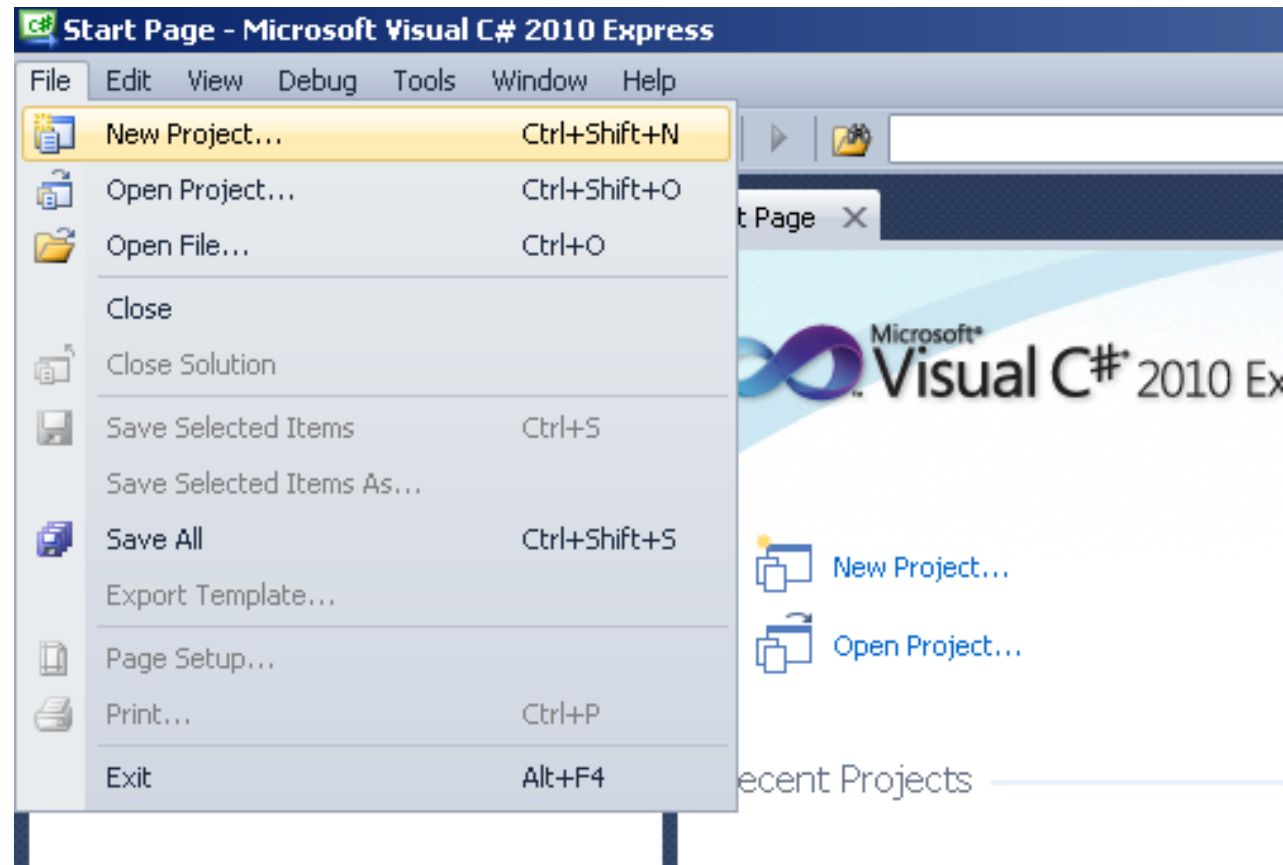
7+4/2 ??????

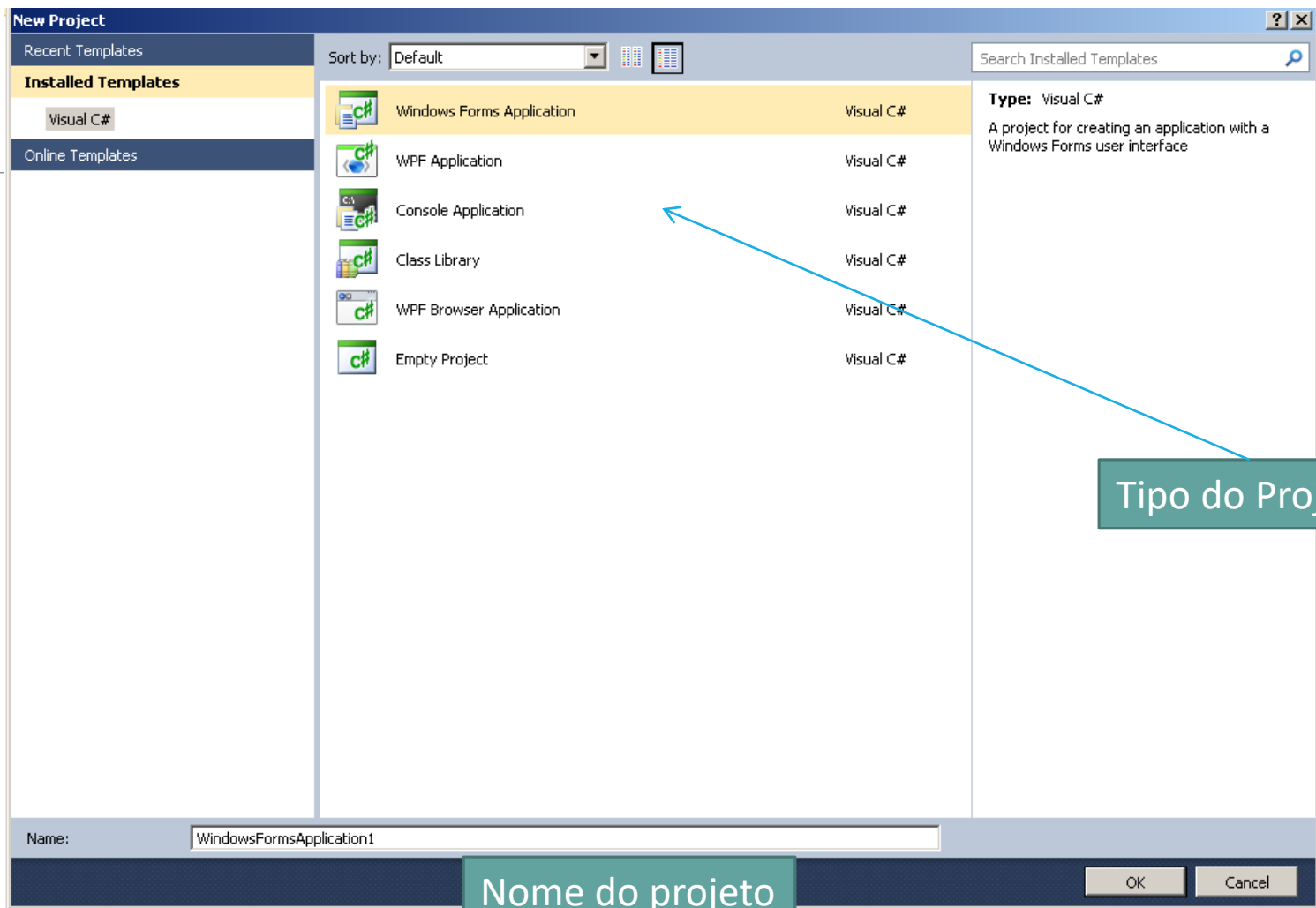


Criando Projetos

Menu File

- *New Project*

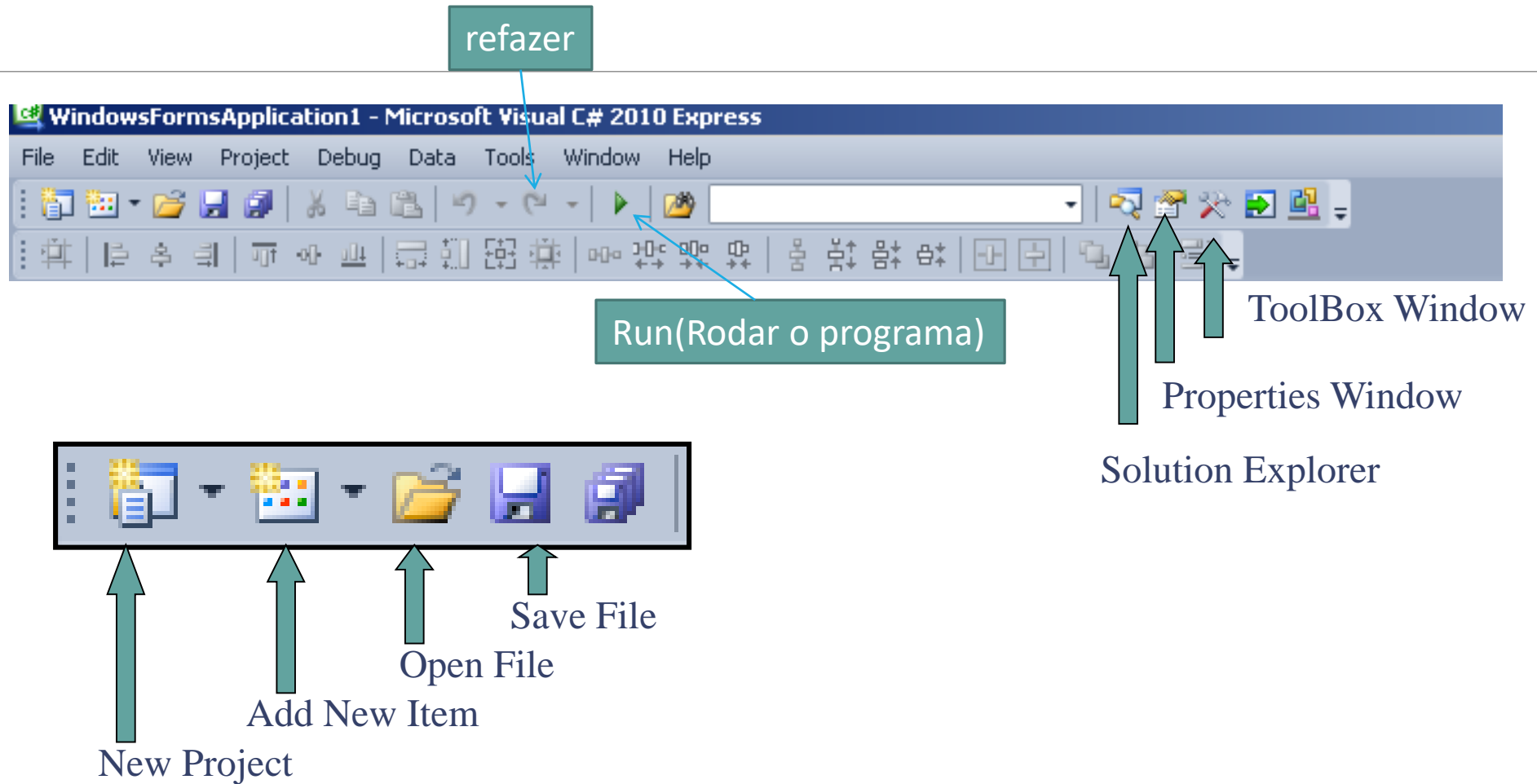




Tipo do Projeto

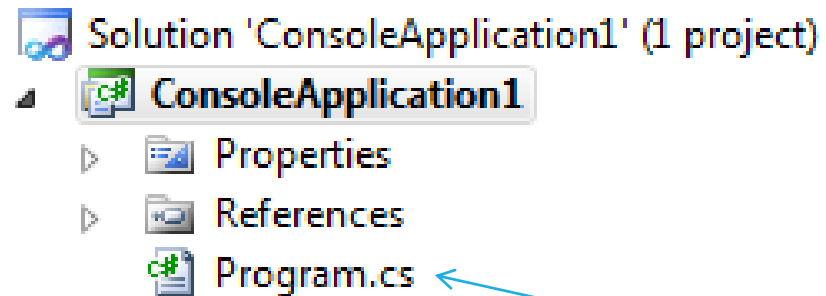
Nome do projeto

standard toolbar options



SOLUTION EXPLORER

Esta caixa de diálogo é usada para gerenciar os arquivos que foram criados no nosso projeto.



Arquivo onde se
deve programar