SENAI CIMATEC

SISTEMA FIEB

TECNOLOGIA E INOVAÇÃO EM PROL DA INDÚSTRIA

SENAI CIMATEC

SISTEMA FIEB

Curso técnico em Informática

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Desenvolvimento de Sistemas II
180h

Profª: Francisleide Almeida

Introdução

- C# foi desenvolvida pela Microsoft. Foi apresentada junto a plataforma .NET.
- C# é uma mistura de C++ e Java.
- Criador: Anders Hejlsberg

Características

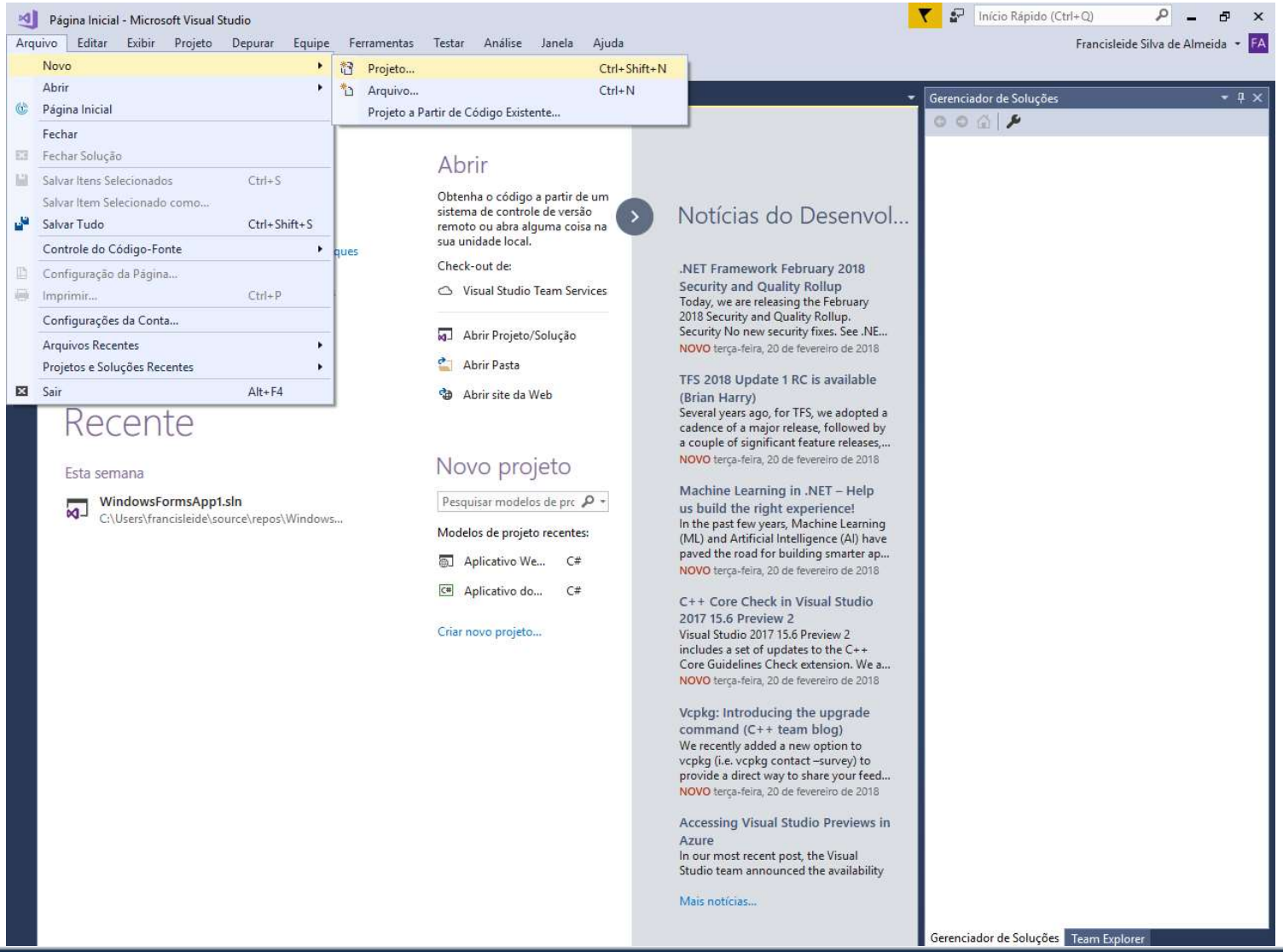
- É orientada a objetos.
- Possui um alto nível de abstração.
- Possui coletor de lixo.
- Suporta tipagem dinâmica e estática.
  - A tipagem dinâmica não exige declarações de tipos de dados (a partir da versão 3.0 do C#)
  - A tipagem estática exige a declaração de quais dados poderão ser associados a cada variável antes da sua utilização.

Características

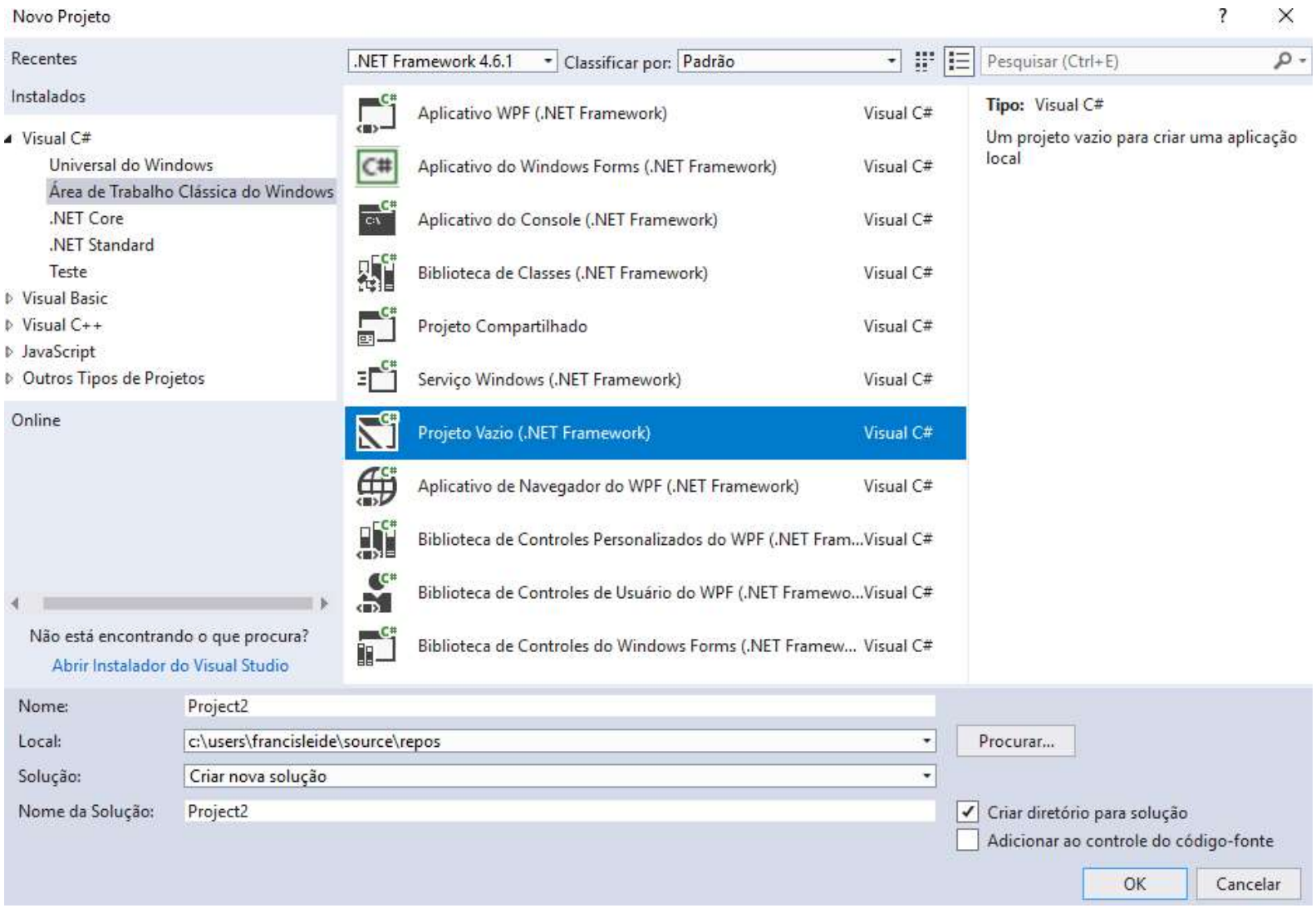
- C# está bastante vinculada ao framework .NET
  - O framework .NET suporta várias linguagens de programação.
- C# usa a biblioteca de classe do framework .NET
  - O framework .NET possui mais de 4 mil classes.

Ambiente de desenvolvimento

- Visual Studio



No projeto...


Novo Projeto
Recentes
Instalados
Visual C#
Universal do Windows
Área de Trabalho Clássica do Windows
.NET Core
.NET Standard
Teste
Visual Basic
Visual C++
JavaScript
Outros Tipos de Projetos
Online
Não está encontrando o que procura?
Abrir Instalador do Visual Studio
.NET Framework 4.6.1
Classificar por: Padrão
Pesquisar (Ctrl+E)
Aplicativo WPF (.NET Framework) Visual C#
Aplicativo do Windows Forms (.NET Framework) Visual C#
Aplicativo do Console (.NET Framework) Visual C#
Biblioteca de Classes (.NET Framework) Visual C#
Projeto Compartilhado Visual C#
Serviço Windows (.NET Framework) Visual C#
Projeto Vazio (.NET Framework) Visual C#
Aplicativo de Navegador do WPF (.NET Framework) Visual C#
Biblioteca de Controles Personalizados do WPF (.NET Fram... Visual C#
Biblioteca de Controles de Usuário do WPF (.NET Framewo... Visual C#
Biblioteca de Controles do Windows Forms (.NET Framew... Visual C#
Tipo: Visual C#
Um projeto vazio para criar uma aplicação local
Nome: Project2
Local: c:\users\francisleide\source\repos
Procurar...
Solução: Criar nova solução
Nome da Solução: Project2
Criar diretório para solução
Adicionar ao controle do código-fonte
OK
Cancelar

Nova classe

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              Console.WriteLine("hello");
14          }
15      }
16  }
17
```

Explicação

- Ctrl + F5 no Visual C# executa sem debugging. Isso força uma pausa no final da execução permitindo que você possa visualizar o resultado da execução.
- Já o atalho F6 é o mesmo que ir no menu Debug e depois clicar em “Build Solution”.

Namespace

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              Console.WriteLine("hello");
14          }
15      }
16  }
17
```

- As classes são organizadas em namespace
- System é um namespace
- Console é uma classe do namespace

Explicação

- Declarar seus próprios namespaces pode ajudar no controle do escopo da classe e nomes de métodos em grandes projetos.
- Os namespaces ajudam na manutenção de um programa!

Explicação

- Quando você criou um projeto em C# utilizando o Visual C#, ele já adicionou:

```
1    using System;
2    using System.Collections.Generic;
3    using System.Linq;
4    using System.Text;
5    using System.Threading.Tasks;
6
```

- Você faz referência a um namespace utilizando a palavra reservada “using”.

Explicação

- WriteLine é um método da classe Console. É passada a string “Hello World!

```
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              Console.WriteLine("hello");
14          }
15      }
16  }
17
```

Observações

- C# é case-sensitive, ou seja, diferencia letras minúsculas de maiúsculas.
- Com o “//” é possível fazer comentários de uma linha. Começando com “/*” e fechando com “*/” é possível fazer comentários de múltiplas linhas.

Leitura de dados

- Método: ReadLine() – Retorna String.

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              int x = Int32.Parse(Console.ReadLine());
14              Console.WriteLine("Valor de x: " + x);
15          }
16      }
17  }
18
```

Tipos de dados C#

Nome abreviado	Classe do .NET	Type (Tipo)	Width	Intervalo (bits)
byte	Byte	Inteiro sem sinal	8	0 a 255
sbyte	SByte	inteiro com sinal com sinal	8	-128 a 127
int	Int32	inteiro com sinal com sinal	32	-2,147,483,648 to 2,147,483,647
uint	UInt32	Inteiro sem sinal	32	0 a 4294967295
short	Int16	inteiro com sinal com sinal	16	-32.768 a 32.767
ushort	UInt16	Inteiro sem sinal	16	0 a 65535
long	Int64	inteiro com sinal com sinal	64	-922337203685477 508 to 9223372036854775 07
ulong	UInt64	Inteiro sem sinal	64	0 a 1844674407370955 1615
float	Single	Tipo de ponto flutuante de precisão simples	32	-3.402823e38 para 3.402823e38
double	Double	Tipo de ponto flutuante de precisão dupla	64	-1.79769313486232 e308 para 1.79769313486232e 308
char	Char	Um único caractere Unicode	16	Unicode símbolos usados no texto
bool	Boolean	Tipo booliano lógico	8	True ou false
object	Object	tipo de base de todos os outros tipos		
string	String	Uma sequência de caracteres		
decimal	Decimal	Preciso tipo fracionário ou integral que pode representar números Decimal com 29 dígitos significativos	128	±1.0 × 10e−28 para ±7.9 × 10e28

Tipos de dados C#

- Descobrindo o tipo de dados de sua variável:

GetType():

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              Console.WriteLine("hello");
14              int x = 55;
15              Console.WriteLine(x.GetType());
16          }
17      }
18  }
19
```

- System.Int32 ??

```
System.Int32
Pressione qualquer tecla para continuar. . .
```

- Cada tipo de dados em C# possui um apelido.

Conversões

- Implícita;
  - Não é necessário nenhum código especial, pois se trata de uma conversão segura, sem risco de perda de dados.
  - Ex:

```
int numero = 1234;
decimal numero2 = numero;
```

De	Para
sbyte	short,int,long,float,double, ou decimal
Byte	short,ushort,int,uint,long,ulong,float,double, ou decimal
short	int,long,float,double, oudecimal
ushort	int,uint,long,ulong,float,double, ou decimal
int	long,float,double, oudecimal
uint	long,ulong,float,double, oudecimal
Long	float,double, oudecimal
char	ushort,int,uint,long,ulong,float,double, ou decimal
float	double
ulong	float,double, ou decimal

Conversões

- **Conversão explícita (casts):**
  - Necessita de um operador. É realizado quando há a necessidade de se converter um valor e pode ocorrer perda de informações.
  - Para fazer um cast, deve ser informado o tipo entre parênteses na frente da variável que será convertida

```
double num = 123.4;
int num2;
num2 = (int)num;
```

Conversões

- **Conversão explícita (casts)** continuação**:**

De	Para
sbyte	byte,ushort,uint,ulong, ouchar
Byte	Sbyteou char
short	sbyte,byte,ushort,uint,ulong, ouchar
ushort	sbyte,byte,short, ouchar
int	sbyte,byte,short,ushort,uint,ulong ou char
uint	sbyte,byte,short,ushort,int ouchar
Long	sbyte,byte,short,ushort,int,uint,ulong ouchar
ulong	sbyte,byte,short,ushort,int,uint,long ouchar
char	sbyte,byte oushort
float	sbyte,byte,short,ushort,int,uint,long,ulong,char oudecimal
double	sbyte,byte,short,ushort,int,uint,long,ulong,char,float ou decimal
decimal	sbyte,byte,short,ushort,int,uint,long,ulong,char,float oudouble

Conversões

- **Conversões com classes auxiliares:**
  - São as conversões em que, por exemplo, o desenvolvedor utiliza a classe System.Convert para converter para o tipo desejado ou o método Parse dos tipos de dados

```
int num;
        num = Convert.ToInt32("123");
```

Classe do System

```
int num;
num = Int32.Parse("123");
```

SubString

- Para obter uma substring, basta utilizar o método Substring().

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              String nome = "Francisleide Almeida";
14              String sobrenome = nome.Substring(13, 7);
15              Console.WriteLine(sobrenome);
16          }
17      }
18  }
19
```

SubString

- Substring() : o primeiro parâmetro é o índice de onde começa e o segundo é a quantidade de caracteres

Tamanho da String

- Utiliza-se o Length.

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              String nome = "Francisleide Almeida";
14              int tamanho = nome.Length;
15              Console.WriteLine("Tamanho da String: " + tamanho);
16          }
17      }
18  }
19
```

Método Contains()

- Verifica se uma string está contida na outra. Retorna True se for verdade e False caso contrário. A saída do programa será True.

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace Project1
8   {
9       class Class1
10      {
11          static void Main(String[] args)
12          {
13              String nome = "Francisleide Almeida";
14              int tamanho = nome.Length;
15              Console.WriteLine(nome.Contains("Alm"));
16          }
17      }
18  }
```

Outros comandos de strings

- Concat(): Concatena Strings;
- TrimStart() e TrimEnd(): Retira caracteres em branco do inicio e do final (respectivamente)
- Split(): Divide uma String, passando o separador por parâmetro;
- Join(): Junta o conteúdo de uma String.

Declarando constante

- Basta usar a palavra reservada “const”antes do tipo da variável.
- Na declaração de uma constante, é obrigatório atribuir um valor a ela no momento da declaração.

```
class Program
{
    static void Main(string[] args)
    {
        const int n = 2000;
        Console.WriteLine(n);
    }
}
```

Condicionais

- If
- Switch

```
class Program
{
    static void Main(string[] args)
    {
        int n1 = 10, n2 = 20;
        if (n1 > n2)
            Console.WriteLine("n1 maior");
        else
            Console.WriteLine("n2 maior ou igual");
    }
}
```

Condicionais

```
class Program
{
    static void Main(string[] args)
    {
        int num = 2;
        switch (num)
        {
            case 1:
                Console.WriteLine("Numero 1");
                break;
            case 2:
                Console.WriteLine("Numero 2");
                break;
            default:
                Console.WriteLine("Numero não encontrado");
                break;
        }
    }
}
```

Condicionais

- O switch pode ser usado com String!

```
class Program
{
    static void Main(string[] args)
    {
        string nome = "Marcos";
        switch (nome)
        {
            case "Joao":
                Console.WriteLine("Joao!");
                break;
            case "Marcos":
                Console.WriteLine("Marcos");
                break;
            default:
                Console.WriteLine("Nome não encontrado");
                break;
        }
    }
}
```

Loops

- do-while
- for
- while
- foreach-in

Do-while

- Segue um exemplo utilizando o do-while que soma os 10 primeiros números naturais não nulos

```
class Program
{
    static void Main(string[] args)
    {
        int soma = 0, i = 1;
        do
        {
            soma += i;
            i++;
        } while (i <= 10);
        Console.WriteLine(soma);
    }
}
```

For

- Mesmo exemplo anterior

```
class Program
{
    static void Main(string[] args)
    {
        int soma = 0;
        for (int i = 1; i <= 10; i++)
            soma += i;
        Console.WriteLine(soma);
    }
}
```

While

- Mesmo exemplo anterior

```
class Program
{
    static void Main(string[] args)
    {
        int soma = 0, i = 1;
        while(i <= 10)
        {
            soma += i;
            i++;
        }
        Console.WriteLine(soma);
    }
}
```

Foreach-in

- Mesmo exemplo anterior.

```
class Program
{
    static void Main(string[] args)
    {
        int[] vet = {1,2,3};

        foreach (int i in vet)
        {
            Console.WriteLine(i);
        }
    }
}
```

Atividades

- Construa um programa em C# que leia um nome e imprima esse nome tantas vezes quanto fores seus caracteres.
- Construa um programa em C# que leia uma String e imprima a maior sequência de letras “A” (maiúsculo ou minúsculo) tem no nome.