Calculo de Data com Forms – C#

fatec – senai centro | Rua Caetés 741, Belo Horizonte

Utilizando o datetime

Rubem Cândido dos Santos

2020

Sumário

[Introdução 2](#_Toc58145981)

[Ferramentas necessárias 2](#_Toc58145982)

[Visual Studio 2](#_Toc58145983)

[Criando a camada View 2](#_Toc58145984)

[Codificando 4](#_Toc58145985)

[Código: 6](#_Toc58145986)

[Github 7](#_Toc58145987)

[Conclusão 7](#_Toc58145988)

# Introdução

O DateTime é um tipo struct do .Net.

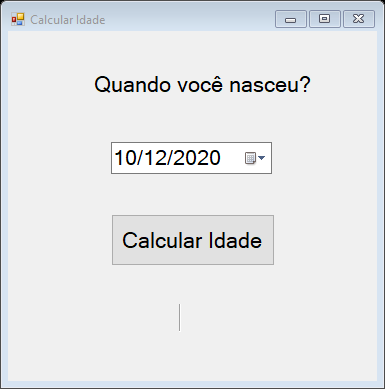


Figura1. Exemplo de aplicação

# Ferramentas necessárias

## Visual Studio

Visual Studio a partir do 2012

## Criando a camada View

No visual studio vamos criar o projeto juntamente com a primeira camada a View. Clique em Arquivo > Novo > Projeto > Visual C# > Aplicativo do Windows Forms (.NET Framework)

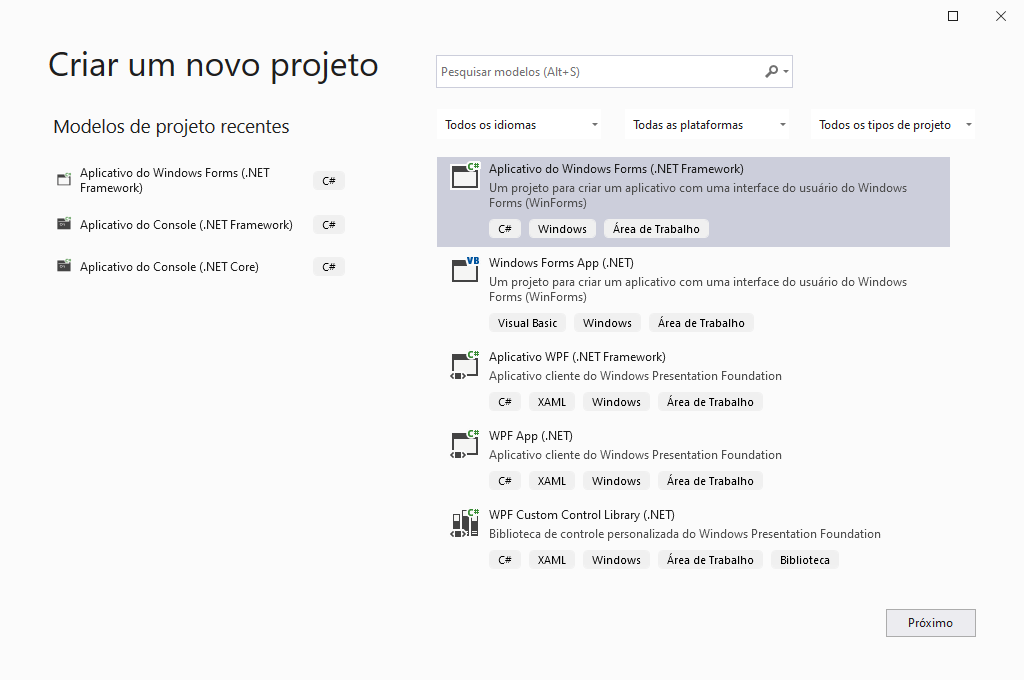


Figura 2. Criando um projeto

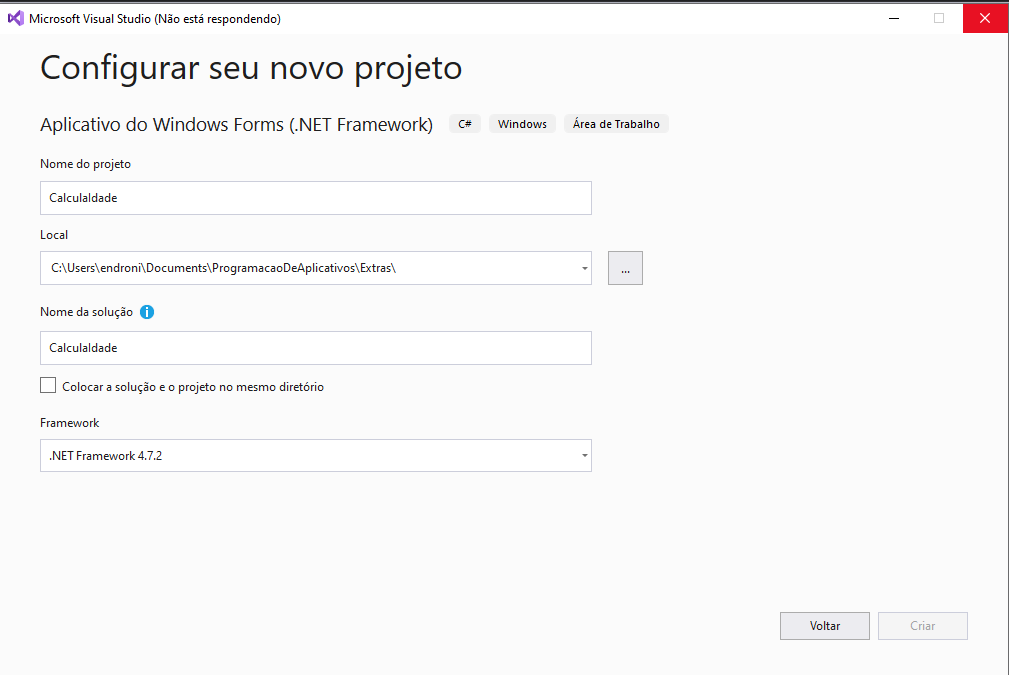


Figura 3. Nome do projeto.

No formulário Form1.cs inclua os seguintes controles a partir da ToolBox:

1 Label

1 DateTimePicker - dtpNascimento, format = short

1 Button - btnCalcularIdade

1 Label - lblIdade , BorderStyle=Fixed3D

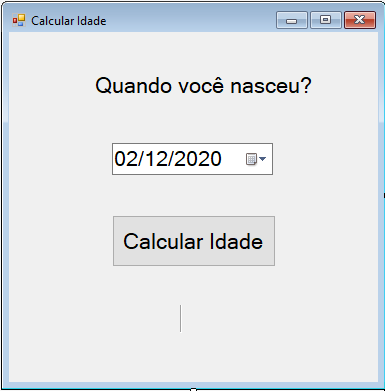


Figura 4. Janela com elementos.

# Codificando

Dê um duplo clique no botão Calcular Idade.

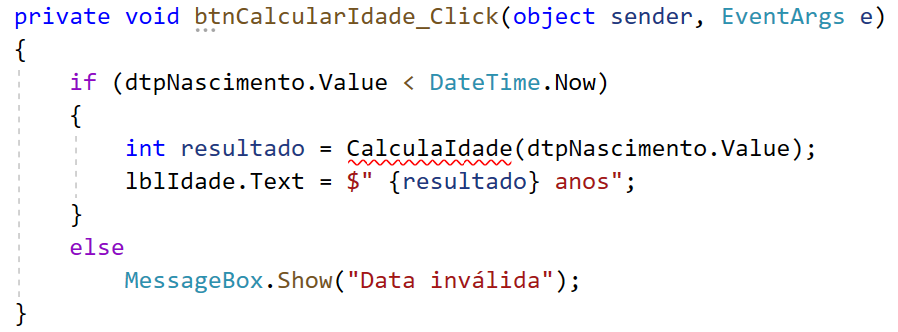
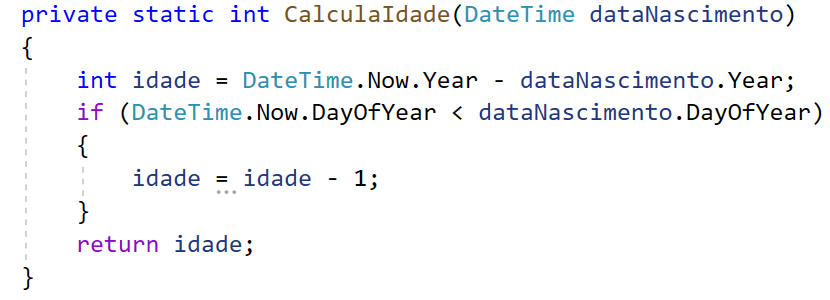


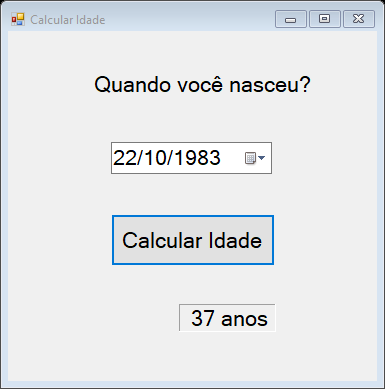
Figura 5. Obtendo a data do usuário

Crie o método CalculaIdade abaixo do evento do botão.



Obs. Se naquele ano a data não tiver ocorrido ainda, quer dizer que a idade (ano) ainda não foi concluído.

Compile e realize testes.



# Código:

using System;

using System.Windows.Forms;

namespace CalculaIdade

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnCalcularIdade\_Click(object sender, EventArgs e)

{

if (dtpNascimento.Value < DateTime.Now)

{

int resultado = CalculaIdade(dtpNascimento.Value);

lblIdade.Text = $" {resultado} anos";

}

else

MessageBox.Show("Data inválida");

}

private static int CalculaIdade(DateTime dataNascimento)

{

int idade = DateTime.Now.Year - dataNascimento.Year;

if (DateTime.Now.DayOfYear < dataNascimento.DayOfYear)

{

idade = idade - 1;

}

return idade;

}

}

}

**DateTime**

Representa um INSTANTE

É um tipo valor (struct)

# Representação interna

Um objeto DateTime internamente armazena:

O número de "ticks" (100 nanosegundos) desde a meia noite do dia 1 de janeiro do ano 1 da era comum

DateTime d1 = DateTime.Now;

Console.WriteLine(d1);

Console.WriteLine(d1.Ticks);

# Instanciação

## Construtores

* DateTime(ano, mes, dia)
* DateTime(ano, mes, dia, hora, minuto, segundo) *[opcional: kind]*
* DateTime(ano, mes, dia, hora, minuto, segundo, milissegundo) *[opcional: kind]*

## Builders

* DateTime.Now
* DateTime.UtcNow
* DateTime.Today *[time: 00:00:00]*
* DateTime.Parse(string)
* DateTime.ParseExact(string, string)

# Demonstração - construtores

DateTime d1 = new DateTime(2000, 8, 15);

DateTime d2 = new DateTime(2000, 8, 15, 13, 5, 58);

DateTime d3 = new DateTime(2000, 8, 15, 13, 5, 58, 275);

Console.WriteLine(d1);

Console.WriteLine(d2);

Console.WriteLine(d3);

# Demo - Now, UtcNow, Today

DateTime d1 = DateTime.Now;

DateTime d2 = DateTime.UtcNow;

DateTime d3 = DateTime.Today;

Console.WriteLine(d1);

Console.WriteLine(d2);

Console.WriteLine(d3);

# Demonstração - Parse

DateTime d1 = DateTime.Parse("2000-08-15");

DateTime d2 = DateTime.Parse("2000-08-15 13:05:58");

DateTime d3 = DateTime.Parse("15/08/2000");

DateTime d4 = DateTime.Parse("15/08/2000 13:05:58");

Console.WriteLine(d1);

Console.WriteLine(d2);

Console.WriteLine(d3);

Console.WriteLine(d4);

# Demonstração - ParseExact

DateTime d1 = DateTime.ParseExact("2000-08-15", "yyyy-MM-dd",

CultureInfo.InvariantCulture);

DateTime d2 = DateTime.ParseExact("15/08/2000 13:05:58", "dd/MM/yyyy HH:mm:ss", CultureInfo.InvariantCulture);

Console.WriteLine(d1);

Console.WriteLine(d2);

# TimeSpan

Representa uma DURAÇÃO

* É um tipo valor (struct)

# Representação interna

Um objeto TimeSpan internamente armazena uma duração na forma de ticks (100 nanosegundos)

TimeSpan t1 = new TimeSpan(0, 1, 30);

Console.WriteLine(t1);

Console.WriteLine(t1.Ticks);

# Construtores

* TimeSpan()
* TimeSpan(ticks)
* TimeSpan(horas, minutos, segundos)
* TimeSpan(dias, horas, minutos, segundos)
* TimeSpan(dias, horas, minutos, segundos, milissegundos)

# Demonstração - construtores

TimeSpan t1 = new TimeSpan();

TimeSpan t2 = new TimeSpan(900000000L);

TimeSpan t3 = new TimeSpan(2, 11, 21);

TimeSpan t4 = new TimeSpan(1, 2, 11, 21);

TimeSpan t5 = new TimeSpan(1, 2, 11, 21, 321);

Console.WriteLine(t1);

Console.WriteLine(t2);

Console.WriteLine(t3);

Console.WriteLine(t4);

Console.WriteLine(t5);

# Demonstração - métodos From

TimeSpan t1 = TimeSpan.FromDays(1.5);

TimeSpan t2 = TimeSpan.FromHours(1.5);

TimeSpan t3 = TimeSpan.FromMinutes(1.5);

TimeSpan t4 = TimeSpan.FromSeconds(1.5);

TimeSpan t5 = TimeSpan.FromMilliseconds(1.5);

TimeSpan t6 = TimeSpan.FromTicks(900000000L);

Console.WriteLine(t1);

Console.WriteLine(t2);

Console.WriteLine(t3);

Console.WriteLine(t4);

Console.WriteLine(t5);

Console.WriteLine(t6);

# Propriedades e Operações com DateTime

## Propriedades

* Date (DateTime)
* Day (int)
* DayOfWeek (DayOfWeek)
* DayOfYear (int)
* Hour (int)
* Kind (DateTimeKind)
* Millisecond (int)
* Minute (int)
* Month (int)
* Second (int)
* Ticks (long)
* TimeOfDay (TimeSpan)
* Year (int)

# Demonstração

DateTime d = new DateTime(2001, 8, 15, 13, 45, 58, 275);

Console.WriteLine(d);

Console.WriteLine("1) Date: " + d.Date);

Console.WriteLine("2) Day: " + d.Day);

Console.WriteLine("3) DayOfWeek: " + d.DayOfWeek);

Console.WriteLine("4) DayOfYear: " + d.DayOfYear);

Console.WriteLine("5) Hour: " + d.Hour);

Console.WriteLine("6) Kind: " + d.Kind);

Console.WriteLine("7) Millisecond: " + d.Millisecond);

Console.WriteLine("8) Minute: " + d.Minute);

Console.WriteLine("9) Month: " + d.Month);

Console.WriteLine("10) Second: " + d.Second);

Console.WriteLine("11) Ticks: " + d.Ticks);

Console.WriteLine("12) TimeOfDay: " + d.TimeOfDay);

Console.WriteLine("13) Year: " + d.Year);

# Formatação (DateTime -> string)

DateTime d = new DateTime(2001, 8, 15, 13, 45, 58);

string s1 = d.ToLongDateString();

string s2 = d.ToLongTimeString();

string s3 = d.ToShortDateString();

string s4 = d.ToShortTimeString();

string s5 = d.ToString();

string s6 = d.ToString("yyyy-MM-dd HH:mm:ss");

string s7 = d.ToString("yyyy-MM-dd HH:mm:ss.fff");

Console.WriteLine(s1);

Console.WriteLine(s2);

Console.WriteLine(s3);

Console.WriteLine(s4);

Console.WriteLine(s5);

Console.WriteLine(s6);

Console.WriteLine(s7);

# Operações com Datetime

DateTime x = ...

DateTime y = x.Add(timeSpan);

DateTime y = x.AddDays(double);

DateTime y = x.AddHours(double);

DateTime y = x.AddMilliseconds(double);

DateTime y = x.AddMinutes(double);

DateTime y = x.AddMonths(int);

DateTime y = x.AddSeconds(double);

DateTime y = x.AddTicks(long);

DateTime y = x.AddYears(int);

DateTime y = x.Subtract(timeSpan);

TimeSpan t = x.Subtract(dateTime);

# Demonstração: MaxValue, MinValue, Zero

TimeSpan t1 = TimeSpan.MaxValue;

TimeSpan t2 = TimeSpan.MinValue;

TimeSpan t3 = TimeSpan.Zero;

Console.WriteLine(t1);

Console.WriteLine(t2);

Console.WriteLine(t3);

# Demonstração - propriedades

TimeSpan t = new TimeSpan(2, 3, 5, 7, 11);

Console.WriteLine(t);

Console.WriteLine("Days: " + t.Days);

Console.WriteLine("Hours: " + t.Hours);

Console.WriteLine("Minutes: " + t.Minutes);

Console.WriteLine("Milliseconds: " + t.Milliseconds);

Console.WriteLine("Seconds: " + t.Seconds);

Console.WriteLine("Ticks: " + t.Ticks);

Console.WriteLine("TotalDays: " + t.TotalDays);

Console.WriteLine("TotalHours: " + t.TotalHours);

Console.WriteLine("TotalMinutes: " + t.TotalMinutes);

Console.WriteLine("TotalSeconds: " + t.TotalSeconds);

Console.WriteLine("TotalMilliseconds: " + t.TotalMilliseconds);

# Demonstração - operações

TimeSpan t1 = new TimeSpan(1, 30, 10);

TimeSpan t2 = new TimeSpan(0, 10, 5);

TimeSpan sum = t1.Add(t2);

TimeSpan dif = t1.Subtract(t2);

TimeSpan mult = t2.Multiply(2.0);

TimeSpan div = t2.Divide(2.0);

Console.WriteLine(t1);

Console.WriteLine(t2);

Console.WriteLine(sum);

Console.WriteLine(dif);

Console.WriteLine(mult);

Console.WriteLine(div);

# DateTimeKind e padrão ISO 8601

## DateTimeKind

Tipo enumerado especial que define três valores possíveis para a localidade da data:

* Local *[ fuso horário do sistema. Exemplo: Belo Horizonte = GMT -3 ]*
* Utc *[ fuso horário GMT (Greenwich Mean Time) ]*
* Unspecified

# Boa prática

Armazenar em formato UTC (texto: BD / Json / XML)

Instanciar e mostrar em formato Local

Para converter um DateTime para Local ou Utc, você deve usar:

* myDate.ToLocalTime()
* myDate.ToUniversalTime()

# Demonstração

DateTime d1 = new DateTime(2000, 8, 15, 13, 5, 58, DateTimeKind.Local);

DateTime d2 = new DateTime(2000, 8, 15, 13, 5, 58, DateTimeKind.Utc);

DateTime d3 = new DateTime(2000, 8, 15, 13, 5, 58);

Console.WriteLine("d1: " + d1);

Console.WriteLine("d1 Kind: " + d1.Kind);

Console.WriteLine("d1 to Local: " + d1.ToLocalTime());

Console.WriteLine("d1 to Utc: " + d1.ToUniversalTime());

Console.WriteLine();

Console.WriteLine("d2: " + d2);

Console.WriteLine("d2 Kind: " + d2.Kind);

Console.WriteLine("d2 to Local: " + d2.ToLocalTime());

Console.WriteLine("d2 to Utc: " + d2.ToUniversalTime());

Console.WriteLine();

Console.WriteLine("d3: " + d3);

Console.WriteLine("d3 Kind: " + d3.Kind);

Console.WriteLine("d3 to Local: " + d3.ToLocalTime());

Console.WriteLine("d3 to Utc: " + d3.ToUniversalTime());

# Formato:

yyyy-MM-ddTHH:mm:ssZ

\* Z indica que a data/hora está em Utc

# Demonstração

DateTime d1 = DateTime.Parse("2000-08-15 13:05:58");

DateTime d2 = DateTime.Parse("2000-08-15T13:05:58Z"); // cria local DateTime

Console.WriteLine("d1: " + d1);

Console.WriteLine("d1 Kind: " + d1.Kind);

Console.WriteLine("d1 to Local: " + d1.ToLocalTime());

Console.WriteLine("d1 to Utc: " + d1.ToUniversalTime());

Console.WriteLine();

Console.WriteLine("d2: " + d2);

Console.WriteLine("d2 Kind: " + d2.Kind);

Console.WriteLine("d2 to Local: " + d2.ToLocalTime());

Console.WriteLine("d2 to Utc: " + d2.ToUniversalTime());

Console.WriteLine();

Console.WriteLine(d2.ToString("yyyy-MM-ddTHH:mm:ssZ")); // cuidado!

Console.WriteLine(d2.ToUniversalTime().ToString("yyyy-MM-ddTHH:mm:ssZ"));

# Github

Subpasta **Extras/CalculoIdade**

<https://github.com/endroni/ProgramacaoDeAplicativos.git>

# Conclusão

O uso do tipo de dados DateTime é relativamente simples de usar.

Espero que tenha valia para seus estudos.

**BIBLIOGRAFIA BÁSICA:**

Documentação Microsoft. **DateTime Estrutura**: .NET. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/api/system.datetime?view=net-5.0 >. Acesso em: 06 de dezembro de 2020.

Documentação Microsoft. **Time Span Struct**: .NET. Disponível em: < https://docs.microsoft.com/en-us/dotnet/api/system.timespan?redirectedfrom=MSDN&view=net-5.0 >. Acesso em: 10 de dezembro de 2020.

Macoratti. **C# - Calculando a idade em anos, meses, dias,...** : Calcular a idade de uma pessoa. Disponível em: < http://www.macoratti.net/19/01/cshp\_calcidade.htm >. Acesso em: 10 de dezembro de 2020.

Nelio Alves. **DateTime**: Manipulando DateTime C#. Disponível em: <https://devsuperior.com.br/>. Acesso em: 10 de dezembro de 2020.

Iso.org. **ISO 8601 DATE AND TIME FORMAT**: Popular Standards. Disponível em: < https://www.iso.org/iso-8601-date-and-time-format.html >. Acesso em: 10 de dezembro de 2020.

Wikipedia. **ISO 8601**: Current date and time expressed according to ISSO 8601. Disponível em: < https://en.wikipedia.org/wiki/ISO\_8601 >. Acesso em: 10 de dezembro de 2020.