

**Tratamento de Datas no C#.NET****Datas e horas usando formatos customizados**

Especificador de formato	Descrição	Exemplos Utilizando <b>ToString</b>
Sem formato	Padrão de data/hora geral.	sem formato: 09/04/2018 19:40:21
"d"	Padrão de data abreviada.	d : 09/04/2018
"D"	Padrão de data por extenso.	D : segunda-feira, 9 de abril de 2018
"f"	Padrão de data completa e hora com hora,minutos	f : segunda-feira, 9 de abril de 2018 19:40
"F"	Padrão de data completa e hora com hora,minutos,segundos	F : segunda-feira, 9 de abril de 2018 19:40:21
"g"	Padrão de data/hora geral	g : 09/04/2018 19:40
"G"	Padrão de data/hora geral com segundos	G : 09/04/2018 19:40:21
"M"	Padrão de dia do mês.	M : 9 de abril
"t"	Padrão de tempo curto.	t: 19:40
"T"	Tempo padrão de tempo.	T: 19:40:21
"Y", "y"	Padrão de mês do ano.	Y ou y: abril de 2018

Atenção: Nos exemplos está sendo utilizado um componente RichText para mostrar os resultados.

Exemplo 1: Utilizando o método ToString

```
private void button1_Click(object sender, EventArgs e)
{
    DateTime dataExemplo = DateTime.Now;

    richTextBox1.Text += "\n" + "sem formato" + dataExemplo.ToString("");
    richTextBox1.Text += "\n" + "d : " + dataExemplo.ToString("d");
    richTextBox1.Text += "\n" + "D : " + dataExemplo.ToString("D");
}
```

```

richTextBox1.Text += "\n" + "f : " + dataExemplo.ToString("f");
richTextBox1.Text += "\n" + "F : " + dataExemplo.ToString("F");
richTextBox1.Text += "\n" + "g : " + dataExemplo.ToString("g");
richTextBox1.Text += "\n" + "G : " + dataExemplo.ToString("G");
richTextBox1.Text += "\n" + "M : " + dataExemplo.ToString("M");
richTextBox1.Text += "\n" + "m : " + dataExemplo.ToString("m");
richTextBox1.Text += "\n" + "s : " + dataExemplo.ToString("s");
richTextBox1.Text += "\n" + "t : " + dataExemplo.ToString("t");
richTextBox1.Text += "\n" + "Y : " + dataExemplo.ToString("Y");
richTextBox1.Text += "\n" + "y : " + dataExemplo.ToString("y");
richTextBox1.Text += "\n" + "dd/mm/yyyy:" +
dataExemplo.ToString("dd/mm/yyyy");
richTextBox1.Text += "\n" + "dd/MM/yyyy:" +
dataExemplo.ToString("dd/MM/yyyy");
}

```

```

-----
sem formato09/10/2018 16:15:50
d : 09/10/2018
D : terça-feira, 9 de outubro de 2018
f : terça-feira, 9 de outubro de 2018 16:15
F : terça-feira, 9 de outubro de 2018 16:15:50
g : 09/10/2018 16:15
G : 09/10/2018 16:15:50
M : 9 de outubro
m : 9 de outubro
T : 16:15:50
t : 16:15
Y : outubro de 2018
y : outubro de 2018
dd/mm/yyyy 09/15/2018
dd/MM/yyyy 09/10/2018

```

### Exemplo 2: Utilizando os métodos do DateTime para mostrar formatos curtos e longos

```

private void button2_Click(object sender, EventArgs e)
{
    DateTime dataExemplo = DateTime.Now;

    richTextBox1.Text += "\n" + "ToShortDateString: " +
dataExemplo.ToShortDateString();
    richTextBox1.Text += "\n" + "ToShortTimeString: " +
dataExemplo.ToShortTimeString();
    richTextBox1.Text += "\n" + "ToLongDateString : " +
dataExemplo.ToLongDateString();
    richTextBox1.Text += "\n" + "ToLongTimeString : " +
dataExemplo.ToLongTimeString();

}

```

```

ToShortDateString: 09/10/2018
ToShortTimeString: 16:12
ToLongDateString : terça-feira, 9 de outubro de 2018
ToLongTimeString : 16:12:15

```

Exercício

Supondo uma variável do tipo DateTime chamada dtCadastro e você deseja imprimir no formato dia/mês/ano, como ficaria no MessageBox.Show?

```
MessageBox.Show(dtCadastro.ToShortDateString());
```

```
MessageBox.Show(dtCadastro.ToString("d"));
```

```
MessageBox.Show(dtCadastro.ToString("dd/MM/yyyy"));
```

Exemplo 3: Usando o método Format

Converte o valor de objetos em cadeias de caracteres com base em formatos de especificado e as insere em outra cadeia de caracteres. O {0} no formato de cadeia de caracteres é um item de formato. 0 é o índice do objeto cujo valor de cadeia de caracteres será inserido nessa posição. (Os índices começam em 0)

Exemplo:

```
string s = String.Format("Hoje é {0:d} às {0:t}", DateTime.Now);  
// Output 'Hoje é 4/17/2018 às 10:04 AM'
```

```
private void button3_Click(object sender, EventArgs e)  
{
```

```
    DateTime dt = DateTime.Now; // supondo que fosse 9/10/2018 21:56
```

```
    richTextBox1.Text += "\n" + String.Format("{0:y yy yyy yyyy}", dt);  
    richTextBox1.Text += "\n" + String.Format("{0:M MM MMM MMMM}", dt);  
    richTextBox1.Text += "\n" + String.Format("{0:d dd ddd dddd}", dt);  
    richTextBox1.Text += "\n" + String.Format("{0:h hh H HH}", dt);  
    richTextBox1.Text += "\n" + String.Format("{0:m mm}", dt);  
    richTextBox1.Text += "\n" + String.Format("{0:s ss}", dt);  
    richTextBox1.Text += "\n" + String.Format("{0:dd/MM/yyyy}", dt);
```

```
18 18 2018 2018  
10 10 out outubro  
9 09 ter terça-feira  
4 04 16 16  
21 21  
56 56  
09/10/2018
```

Exercício

O que será impresso com o comando?

```
DateTime dt = DateTime.Today;  
MessageBox.Show(String.Format("{0:dd/MM/yyyy}", dt));
```

→ 03/11/2020

## Calendários

O C#.NET possui a diversas classes para outros calendários que estão no namespace - System.Globalization onde podemos usar classes para outros calendários: *Árabe, Coreano, Japonês, etc.*

### Exemplo 4: Datas em vários calendários

```
private void button5_Click(object sender, EventArgs e)
{
    HijriCalendar arabe = new HijriCalendar();
    JapaneseCalendar japonês = new JapaneseCalendar();
    KoreanCalendar coreano = new KoreanCalendar();

    richTextBox1.Text += "\n" + "No Calendário Ocidental hoje é dia : " +
    DateTime.Today;
    richTextBox1.Text += "\n" + "No Calendário Árabe hoje é dia : " +
    arabe.GetDayOfMonth(DateTime.Today) + "/" +
    arabe.GetMonth(DateTime.Today) + "/" + arabe.GetYear(DateTime.Today);
    richTextBox1.Text += "\n" + "No Calendário Japão hoje é dia : " +
    japonês.GetDayOfMonth(DateTime.Today) + "/" +
    japonês.GetMonth(DateTime.Today) + "/" + japonês.GetYear(DateTime.Today);
    richTextBox1.Text += "\n" + "No Calendário Coreano hoje é dia : " +
    coreano.GetDayOfMonth(DateTime.Today) + "/" +
    coreano.GetMonth(DateTime.Today) + "/" +
    coreano.GetYear(DateTime.Today);
}
```

ATENÇÃO : Desde 1º de janeiro de 1873 (ano 6 da Era Meiji), o Japão usa o calendário gregoriano. A forma de contar os dias e os meses é igual à ocidental, mas cada ano tem um nome, de acordo com a era. Por exemplo, 1999 corresponde ao ano 11 da Era Heisei.

```
No Calendário Ocidental hoje é dia : 09/10/2018 00:00:00
No Calendário Árabe hoje é dia : 29/1/1440
No Calendário Japão hoje é dia : 9/10/30
No Calendário Coreano hoje é dia : 9/10/4351
No Calendário Hebraico hoje é dia : 30/1/5779
```

## Métodos e Propriedades para tratamento de datas no C#

Ação a realizar	Método/Propriedade	Tipo Retorno
Obter a Data atual	DateTime.Today	DateTime
Obter a Data e	DateTime.Now	DateTime

hora atual		
Adicionar dias	DateTime.AddDays(n)	DateTime
Adicionar meses	DateTime.AddMonths(n)	DateTime
Adicionar anos	DateTime.AddYears(n)	DateTime
Adicionar Horas	DateTime.AddHours(n)	DateTime
Obter Dia da Semana	DateTime.DayOfWeek	DayOfWeek
Obter Hora do Dia	DateTime.TimeOfDay	TimeSpan
Obter Dias de um Mês	System.DateTime.DaysInMonth(ano,mês)	int
Obter Dia	DateTime.Day	Int
Obter Mês	DateTime.Month	Int
Obter Ano	DateTime.Year	int
Testar se ano é bissexto	DateTime.IsLeapYear(ano)	Booleano
Volta a diferença entre duas datas em dias	data2.Subtract(data1).TotalDays ** data2 deve ser a data mais recente senão volta valor negativo	Double

**Exemplo 4: Métodos e Propriedades para Date e Hora**

```
private void button6_Click(object sender, EventArgs e)
{
    DateTime dt = new DateTime(2008, 4, 10, 6, 30, 0); // ano, mes, dia,
    hora, minutos, segundo
    //ou
    //DateTime dt = Convert.ToDateTime("10/04/2008 6:30:00");

    richTextBox1.Text += "\n" + dt.AddDays(2).ToShortDateString();
    richTextBox1.Text += "\n" + dt.AddHours(2).ToLongTimeString();
    richTextBox1.Text += "\n" + dt.AddMonths(2).ToShortDateString();
    richTextBox1.Text += "\n" + dt.DayOfWeek.ToString();
    richTextBox1.Text += "\n" + dt.TimeOfDay.ToString();
    richTextBox1.Text += "\n" + dt.Day.ToString();
    richTextBox1.Text += "\n" + dt.Month.ToString();
    richTextBox1.Text += "\n" + dt.Year.ToString();
    richTextBox1.Text += "\n" + System.DateTime.DaysInMonth(2018, 4);

    if (DateTime.IsLeapYear(2018)) // Se o valor fosse 2010 retornaria false
        MessageBox.Show("Bissexto");
    else
        MessageBox.Show("Não Bissexto");

    DateTime dtVencimento = Convert.ToDateTime("15/04/2018");

    MessageBox.Show(dtVencimento.AddDays(15).ToShortDateString());
    MessageBox.Show(dtVencimento.AddMonths(2).ToShortDateString());
}
```

```
MessageBox.Show(dtVencimento.AddYears(-2).ToShortDateString());
```

```
DateTime data1 = Convert.ToDateTime("21/04/2017");
```

```
DateTime data2 = Convert.ToDateTime("26/04/2018");
```

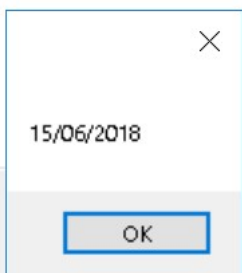
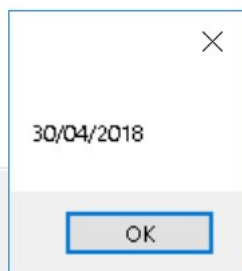
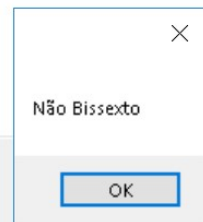
```
double dias;
```

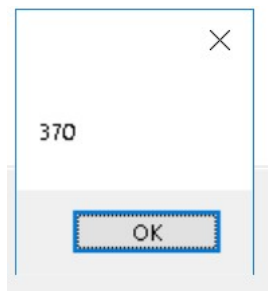
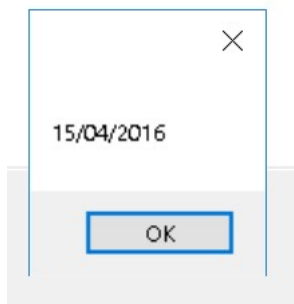
```
dias = data2.Subtract(data1).TotalDays;
```

```
MessageBox.Show(dias.ToString());
```

```
}
```

```
12/04/2008  
08:30:00  
10/06/2008  
Thursday  
06:30:00  
10  
4  
2008  
30
```





### Exercício

Supondo uma variável do tipo DateTime chamada dtVencimento que foi inicializada com a data 15/04/2018. Como faria para:

MessageBox.Show("15 dias" + dtVencimento.AddDays(15).ToShortDateString());

Somar 15 dias na data?

Somar 2 meses na data?

Subtrair 2 anos na data?

E para separar(pegar) somente o mês dessa data?

Fazer a diferença entre essa data e 15/04/2017?

DateTime dtAux = Convert.ToDateTime("14/04/2017");

int numDias;

numDias = dtVencimento.Subtract(dtAux).TotalDays;

### **Formatando números**

A formatação de números pode ser aplicada a qualquer tipo de número. A formatação abaixo mostra os especificadores de formato.

Especificador de formato	Descrição	Saída
"C" ou "c"	Moeda ** pode incluir casas decimais ex. C3	R\$2.592,300 -R\$2.592,300

Especificador de formato	Descrição	Saída
D ou d	Decimal ** pode incluir número total de casas, colocará zeros à esquerda	000250
"E" ou "e"	Científico	2,592300E+003
F ou f	Ponto fixo ** pode incluir casas decimais Ex. F3, F0	2592,30 2592
"G" ou "g"	Geral ** pode incluir total de casas decimais Ex.G6	2592,3
"N" ou "n"	Número ** pode incluir total de casas decimais Ex.N4	2.592,3000
"X" ou "x"	Hexadecimal	F A

Exemplo 5: Utilizando Format para números

```
private void button8_Click(object sender, EventArgs e)
{
    richTextBox1.Text += String.Format("{0:C3}", 2592.30);
    richTextBox1.Text += "\n" + String.Format("{0:C3}", -2592.30);
    richTextBox1.Text += "\n" + String.Format("{0:D6}", 250);
    richTextBox1.Text += "\n" + String.Format("{0:E}", 2592.30);
    richTextBox1.Text += "\n" + String.Format("{0:F2}", 2592.30);
    richTextBox1.Text += "\n" + String.Format("{0:F0}", 2592.30);
    richTextBox1.Text += "\n" + String.Format("{0:G6}", 2592.30);
    richTextBox1.Text += "\n" + String.Format("{0:N4}", 2592.30);
    richTextBox1.Text += "\n" + String.Format("{0:X}", 15);
    richTextBox1.Text += "\n" + String.Format("{0:X}", 10);
}
```



```

R$2.592,300
-R$2.592,300
000250
2.592300E+003
2592,30
2592
2592,3
2.592,3000
F
A

```

### Exemplo 6: Utilizando ToString

```

private void button9_Click(object sender, EventArgs e)
{
    double numero = 2592.30;
    int numero2 = 15;
    int numero3 = 10;
    int numero4 = 2592;

    richTextBox1.Text += numero.ToString("C3");
    richTextBox1.Text += "\n" + numero.ToString("C3");
    richTextBox1.Text += "\n" + numero4.ToString("D6");
    richTextBox1.Text += "\n" + numero.ToString("E");
    richTextBox1.Text += "\n" + numero.ToString("F2");
    richTextBox1.Text += "\n" + numero.ToString("F0");
    richTextBox1.Text += "\n" + numero.ToString("G6");
    richTextBox1.Text += "\n" + numero.ToString("N4");
    richTextBox1.Text += "\n" + numero2.ToString("X");
    richTextBox1.Text += "\n" + numero3.ToString("X");
}

```

```

R$2.592,300
R$2.592,300
002592
2.592300E+003
2592,30
2592
2592,3
2.592,3000
F
A

```

### Exercício

Supondo uma variável do tipo Double totalCompra=3525.45.

Como mostraria essa data formatada com 2 casas decimais no formato de moeda?

```

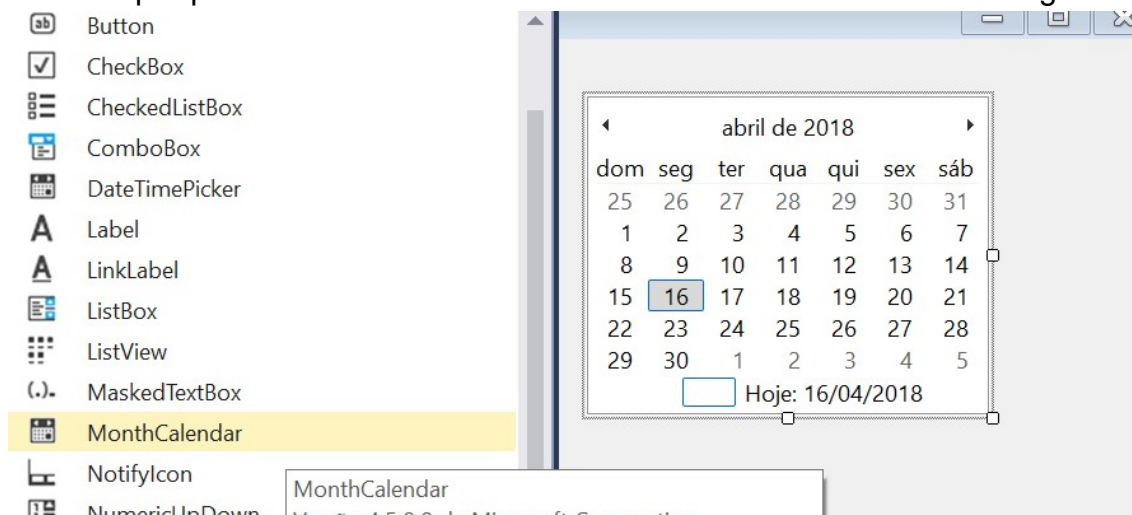
MessageBox.Show(String.Format("{0:C2}", totalCompra));

```

```
MessageBox.Show(totalCompra.ToString("C2"));
```

## Componente MonthCalendar

Na figura abaixo temos a aparência inicial do calendário: Uma grade que exibe os dias da semana e os dias do mês atual. O mês em curso e o dia atual são exibidos por padrão e o dia também é marcado e destacado na base da grade.



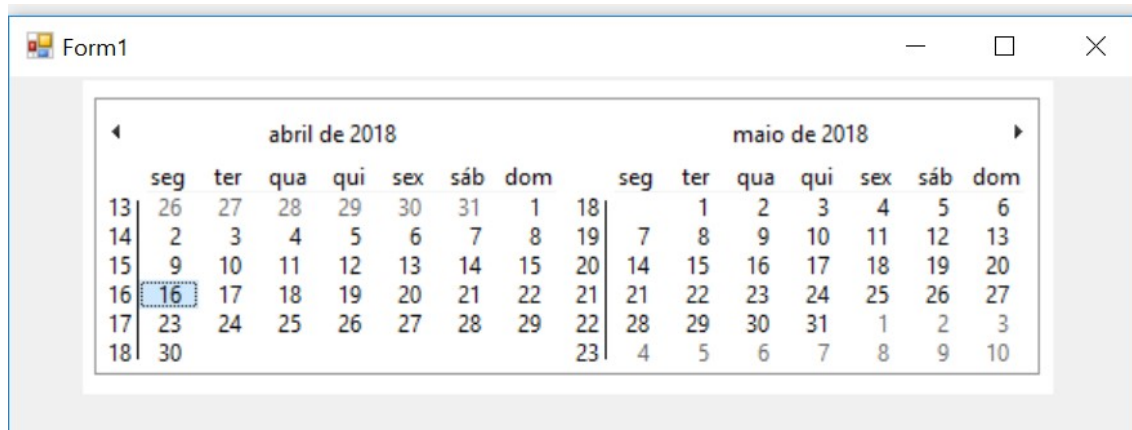
A propriedade chave do controle é *SelectionRange*, o intervalo de datas selecionadas no controle. O valor da propriedade não pode exceder o número máximo de dias que pode ser selecionado e está definido na propriedade *MaxSelectionCount*. A menor e a maior data que pode ser selecionada é determinada pelas propriedades *MaxDate* e *MinDate*.

## Alterando as configurações padrão do controle MonthCalendar

Podemos alterar diversas propriedades e métodos do controle personalizando sua aparência. Vejamos o que podemos fazer para isto:

- Para desabilitar a marcação do dia atual defina as propriedades *ShowToday* e *ShowTodayCircle* como *False*.
- Para exibir o número de semanas no ano no calendário defina a propriedade *ShowWeekNumbers* como *True*.
- Para exibir múltiplos meses altere o valor padrão da propriedade *CalendarDimensions* que é de 1;1 para outro valor; 2;1 por exemplo (que seriam 2 meses na mesma linha (horizontal) e colocar 1;2 seriam 2 meses na mesma coluna (vertical)).
- Para alterar o primeiro dia da semana padrão que é *sunday* defina a propriedade *FirstDayOfWeek* para um valor diferente de default.

Abaixo temos a figura que exibe o resultado obtido pelas alterações destas configurações:



A definição destas propriedades via código pode ser feita assim:

```
MonthCalendar1.ShowToday = False
MonthCalendar1.ShowTodayCircle = False
MonthCalendar1.ShowWeekNumbers = True
MonthCalendar1.FirstDayOfWeek = Day.Monday
```

Para exibir múltiplos meses via código:

```
MonthCalendar1.CalendarDimensions = New System.Drawing.Size (2,1)
```

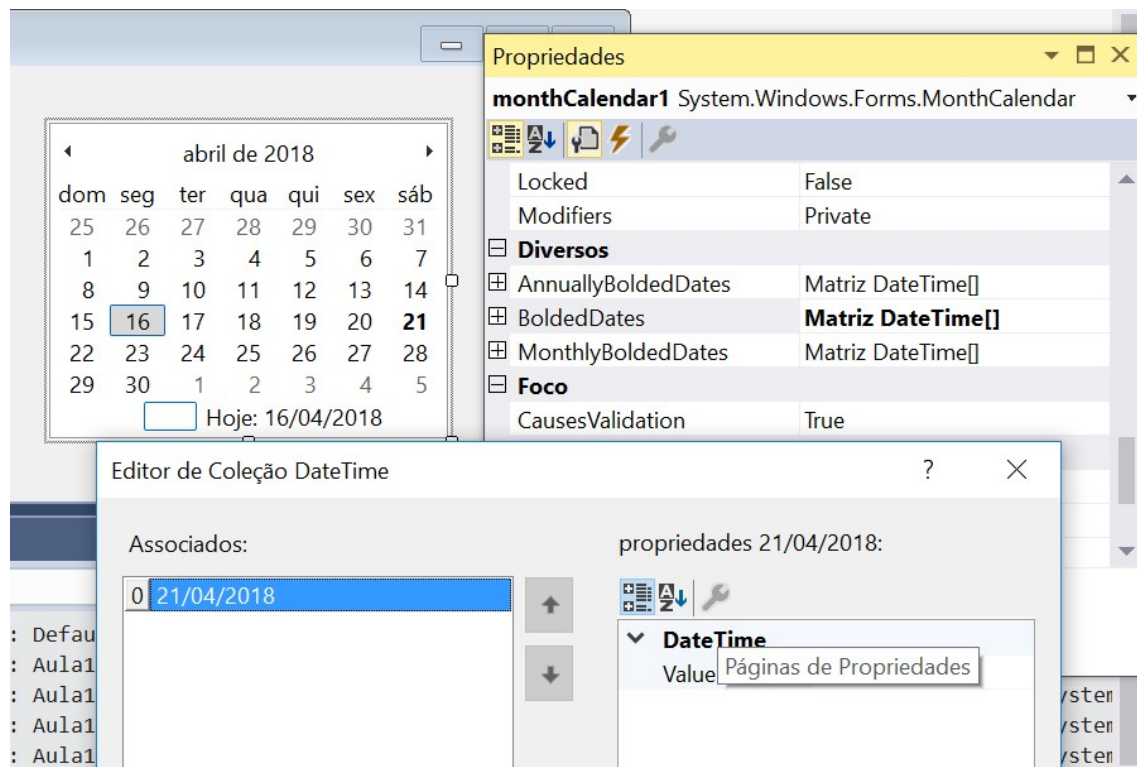
Você pode destacar em negrito datas especiais a seu critério usando as propriedades:

***BoldedDates*** - destaca uma única data uma única vez.

***AnnuallyBoldedDates***: contém datas que serão destacadas e irão ser exibidas a cada ano.

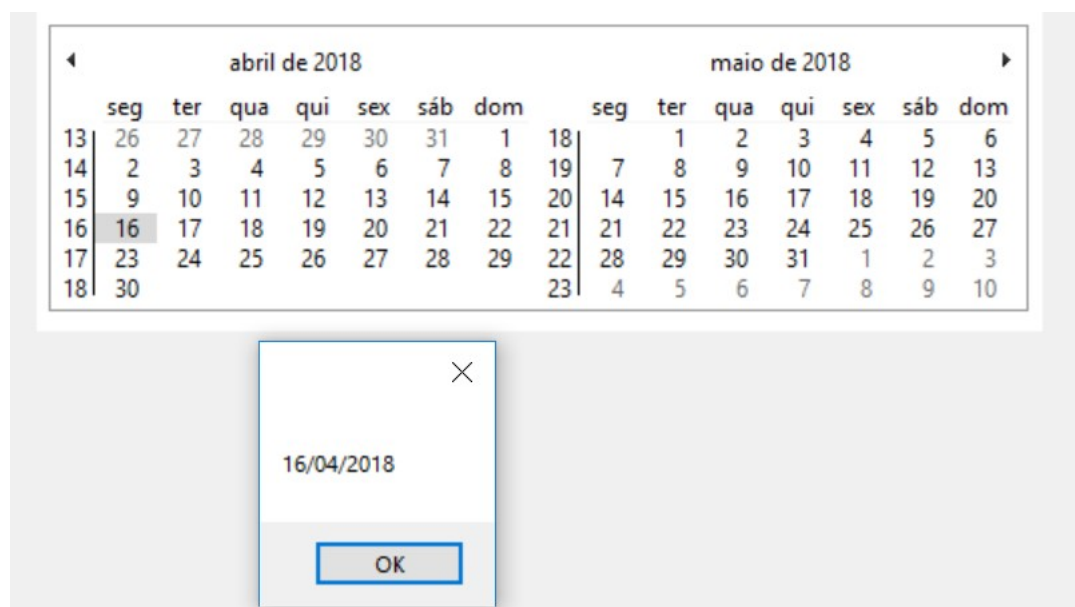
***MonthlyBoldedDates*** : contém datas que serão destacadas e irão ser exibidas a cada mês.

Cada uma destas propriedades possuem um array de objetos *DateTime* como mostra a figura abaixo:



### Para Pegar a Data Inicial do Calendário:

```
private void button1_Click(object sender, EventArgs e)
{
    DateTime dt = monthCalendar1.SelectionStart;
    MessageBox.Show(dt.ToShortDateString());
}
```



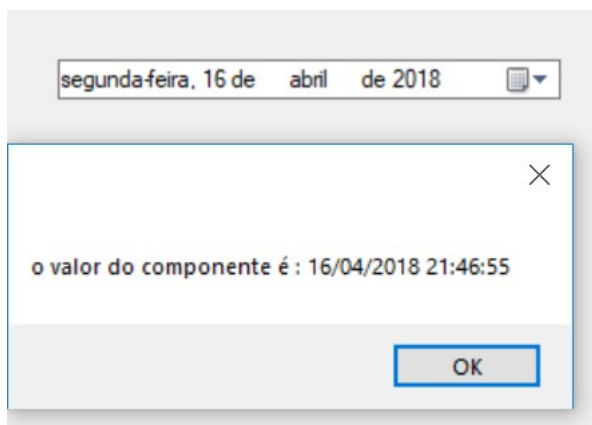
**O Componente DateTimePicker**

Na figura abaixo temos a aparência inicial do calendário:

**Para Pegar a Data Inicial do Calendário:**

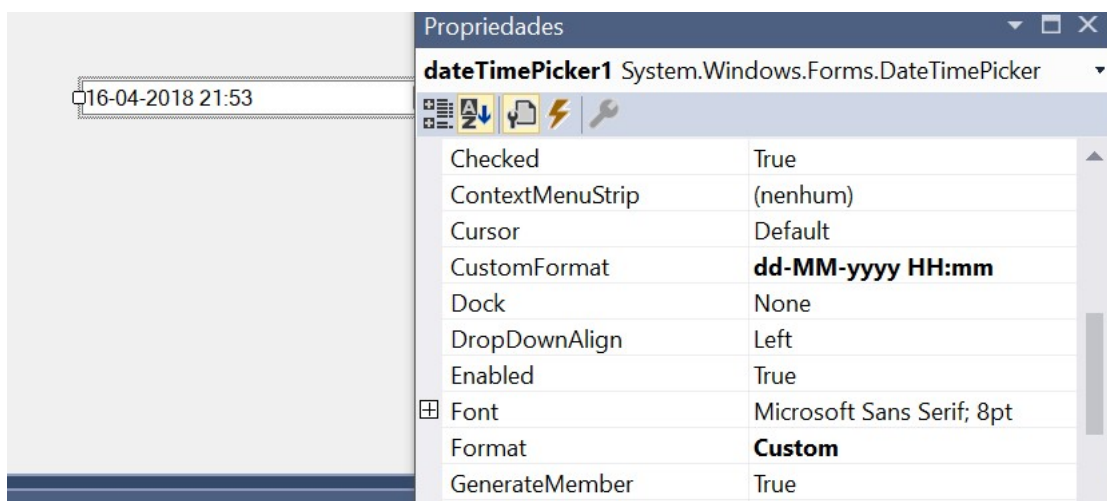
```
Private Sub Button1_Click(object sender, EventArgs e)
    DateTime Dt;
    Dt =DateTimePicker1.Value;

    MessageBox.show("o valor do componente é : " + Dt.ToString);
}
```



Para alterar o formato data/hora mostrados basta alterar a propriedade **FORMAT**, se desejar um formato customizado escolha **CUSTOM** e altere também a propriedade **CUSTOMFORMAT**.

Ex. Altere CustomFormat para dd-MM-yyyy HH:mm



**Referências**

FORMATOS. Disponível em: <[https://msdn.microsoft.com/pt-br/library/az4se3k1\(v=vs.100\).aspx](https://msdn.microsoft.com/pt-br/library/az4se3k1(v=vs.100).aspx)> Acesso: 05.ABR.2018.

FORMATOS2. Disponível em: <<https://docs.microsoft.com/en-us/dotnet/standard/base-types/standard-date-and-time-format-strings#the-short-date-d-format-specifier>> Acesso: 05.ABR.2018.

GLOBALIZATION. Disponível em: <[https://msdn.microsoft.com/en-us/library/system.globalization\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.globalization(v=vs.110).aspx)> Acesso: 05.ABR.2018.