

01

Trabalhando com Datas e Horas

Transcrição

Após trabalharmos com validações e formatações, veremos como trabalhar com **datas e horas** no formato brasileiro.

Para isso, adicionaremos um novo projeto em "Add > New Project... > Console App (.NET Framework)", e vamos chamá-lo de **TrabalhandoComDatas**. Definiremos esse projeto como *StartUp Project*, clicando com o direito no projeto.

Dentro da `main` desse projeto, instanciaremos uma "data" do tipo `DateTime`. E como passaremos uma data específica para o `DateTime`? Passaremos o **ano**, o **mês** e em seguida, o **dia**, separados por vírgulas:

```
static void Main(string[] args)
{
    DateTime data = new DateTime(2017, 9, 23);
}
```

Em seguida, imprimiremos essa data no *console*:

```
static void Main(string[] args)
{
    DateTime data = new DateTime(2017, 9, 23);
    Debug.WriteLine(data);
}
```

E rodamos a aplicação. Como podemos ver, a nossa data apareceu no console, mas também temos um horário. Isso aconteceu, porque o `DateTime` armazena não só a **data**, como também a **hora**.

Então, se nós queremos exibir somente o dia, mês e ano, o que precisamos fazer?

Precisamos formatar essa data de uma outra maneira!

```
static void Main(string[] args)
{
    DateTime data = new DateTime(2017, 9, 23);
    Debug.WriteLine(data);
    Debug.WriteLine(data.ToString("d"));
}
```

Com o `ToString`, nós podemos formatar a data de outra maneira. Quando colocamos a letra `d` em minúsculo, estamos pedindo para mostrar somente a data sem a hora!

Vamos rodar o projeto para ver o resultado no console.

E se quisermos, por exemplo, imprimir essa data no formato brasileiro?

Podemos fazer da seguinte maneira:

```
static void Main(string[] args)
{
    DateTime data = new DateTime(2017, 9, 23);
    Debug.WriteLine(data);
    Debug.WriteLine(data.ToString("d"));
    Debug.WriteLine(data.ToString("d", new CultureInfo("pt-BR")));
}
```

O que fizemos aqui foi passar um segundo parâmetro para o `ToString()`, que é a informação da *cultura*, passando o código da cultura do Brasil.

Rodando a aplicação, percebemos alguma diferença nos resultados? Aparentemente não. Mas, por que é importante colocar o `CultureInfo()` sendo ele um tipo de formatação de data? Pois, nós podemos estar trabalhando com um computador que talvez não esteja na **cultura brasileira**, pode ser um servidor internacional, uma máquina de outro país, ou outro cenário, o importante é garantir que a impressão dessa data seja no formato **pt-BR**.

Podemos formatar essa data de outras maneiras. Se quisermos exibir somente o **dia** e o **mês**, trocamos o "d" do `ToString()` por "dd/mm":

```
static void Main(string[] args)
{
    DateTime data = new DateTime(2017, 9, 23);
    Debug.WriteLine(data);
    Debug.WriteLine(data.ToString("d"));
    Debug.WriteLine(data.ToString("d", new CultureInfo("pt-BR")));
    Debug.WriteLine(data.ToString("dd/MM"));
}
```

Vamos rodar a aplicação. Apareceu para nós esse resultado:

23/00

Por que apareceu **00** no lugar do mês? Lembre-se que o `ToString()` é **Case Sensitive**, isto significa que o "m" minúsculo é o **minuto** e não o mês! Então devemos trocar o "m" minúsculo pelo "M" maiúsculo.

```
Debug.WriteLine(data.ToString("dd/MM"));
```

Vamos rodar novamente a aplicação. Agora sim obtemos **23/09**.

E se quisermos também exibir o ano, porém somente com dois dígitos? Adicionaremos ao `ToString()` a letra que representa o ano em inglês.

```
static void Main(string[] args)
{
    DateTime data = new DateTime(2017, 9, 23);
    Debug.WriteLine(data);
    Debug.WriteLine(data.ToString("d"));
    Debug.WriteLine(data.ToString("d", new CultureInfo("pt-BR")));
    Debug.WriteLine(data.ToString("dd/MM"));
```

```
        Debug.WriteLine(data.ToString("dd/MM/yy"));
    }
```

Muito bem! A partir de agora, trabalharemos um pouco com as **horas**.

Para instanciar uma data com a hora, podemos fazer assim:

```
static void Main(string[] args)
{
    // formatação da data

    data = new DateTime(2017, 9, 23, 13, 14, 15, 987);
    Debug.WriteLine(data);
}
```

Aqui, nós acrescentamos a hora logo após a data, sendo horas, minutos, segundos e milissegundos, respectivamente.

E se quisermos imprimir somente a hora e o minuto? Formatamos dessa forma:

```
static void Main(string[] args)
{
    // formatação da data

    data = new DateTime(2017, 9, 23, 13, 14, 15, 987);
    Debug.WriteLine(data);
    Debug.WriteLine(data.ToString("hh:mm"));
}
```

O resultado que obtemos é:

01:14

Repare que está sendo exibido no formato americano, e nós queremos o formato brasileiro. Para isso, trocaremos "hh" por "HH".

E o nosso resultado é **13:14**. Como fazemos para mostrar os segundos e os milissegundos? Somente precisamos acrescentar "ss.fff" após os minutos.

```
static void Main(string[] args)
{
    // formatação da data

    data = new DateTime(2017, 9, 23, 13, 14, 15, 987);
    Debug.WriteLine(data);
    Debug.WriteLine(data.ToString("HH:mm"));
    Debug.WriteLine(data.ToString("HH:mm:ss.fff"));
}
```

Mais a frente, veremos mais formatos de data e hora!

