



Neste artigo vou apresentar os conceitos básicos sobre a view **DatePicker** do **Xamarin Forms** usando o Visual Studio 2015 e a linguagem C#.

Na [primeira parte do artigo](#) eu mostrei como usar a view **DatePicker** em um projeto Xamarin Forms usando código C#.

Vamos ver como obter o mesmo resultado usando o código **XML** e também os recursos do **Data Binding**.

Recursos usados:

- [Visual Studio Community 2015](#) ou Xamarin Studio
- [Xamarin](#)
- Emulador Android virtual ou físico ([veja como emular usando o Vysor](#))

Nota: Baixe e use a versão Community 2015 do VS ela é grátis e é equivalente a versão Professional.

Criando o projeto no VS Community 2015

Abra o **VS 2015 Community** e clique em **New Project**;

Selecione a linguagem Visual C# e o template **Android -> Blank App(Android)**

Informe o nome **App_DatePicker2** e clique no botão **OK**;

Ao clicar no botão OK, será criada uma solução contendo 4 projetos. *(Dependendo do seu ambiente pode haver alguma variação nos projetos.)*

- **1 Projeto Comum** contendo a maior parte do código da aplicação : *(Este é o projeto Portable cuja compilação pode ser compartilhada com outras plataformas)*
- **1 Projeto para plataforma Android**
- **1 Projeto para plataforma iOS**
- **1 Projeto para Windows 8.1 ou para Windows 10**
- **1 Projeto para plataforma Windows Phone**

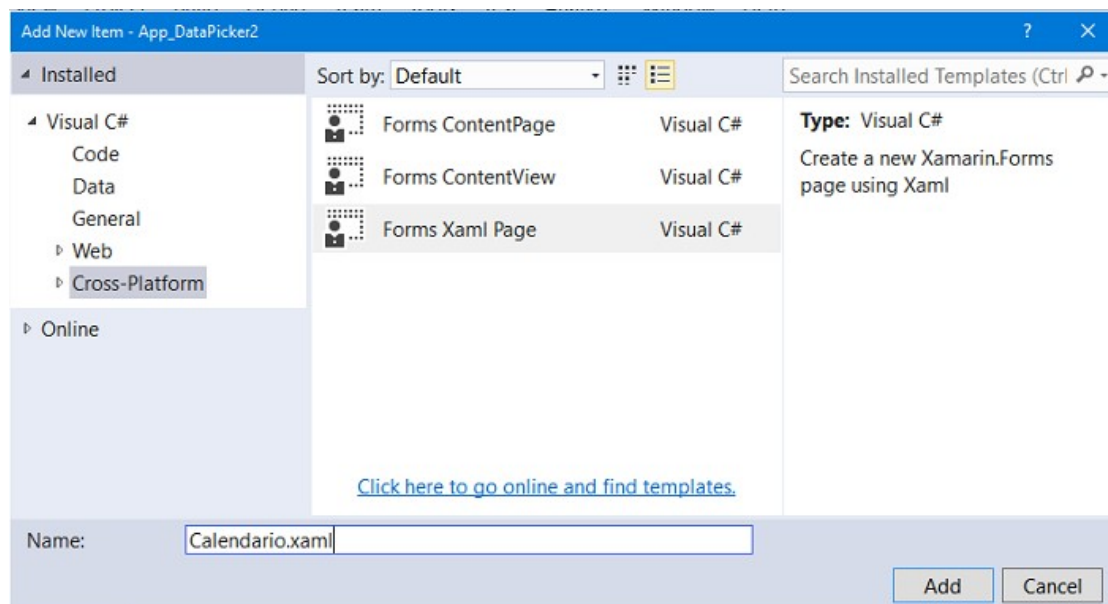
Nota: Durante o processo de criação dos projetos, ao criar o projeto para o iOS será apresentada a janela : **Xamarin Mac Agent Instructions** , com instruções para identificar e localizar o Mac (a própria máquina ou em uma rede local) Se você não tiver um Mac (como eu) clique no botão OK e a seguir clique no botão Close. Com isso não poderemos compilar nem testar projetos iOS no Visual Studio.

O projeto comum possui a classe **App.cs** que irá conter o código compartilhado e que vamos usar neste artigo.

Podemos na janela de propriedades da solução marcar a opção - **Multiple startup projects** - e selecionar quais o projetos vamos fazer o deploy e executar.

Criando a UI via XML

Vamos incluir uma nova página em nosso projeto via menu **Project -> Add New Item** usando o template **Cross Platform -> Forms Xaml Page** e informando o nome **Calendario.xaml**;



A seguir vamos definir o código abaixo neste arquivo onde definimos uma página contendo:

- 1 **StackLayout** - vertical
- 1 **Label**
- 1 **DatePicker** - **DateSelected = OnDateSelected**
- 1 **Label** - **x:Name=lblData**

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="App_DataPicker2.Calendario">

    <StackLayout Orientation="Vertical" VerticalOptions="Center">
        <Label
            Text="DatePicker - XML" FontSize="20" HorizontalOptions="Center" TextColor="#e7d692"
        />

        <DatePicker
            MinimumDate="01/01/1970"
            MaximumDate="01/01/2100"
            Format="dd/MM/yyyy"
            DateSelected="OnDateSelected"
        />

        <Label
            x:Name="lblData" FontSize="30" HorizontalOptions="Center" TextColor="#e7d692"
        />
    </StackLayout>
</ContentPage>
```

Definimos o evento **DateSelected** em nosso **DatePicker** e precisamos implementar via código esse evento no code-behind.

Definindo o evento no Code-behind : Calendario.xaml.cs

Abra o arquivo **Calendario.xaml.cs** e inclua o código a seguir onde definimos o código do método **OnDateSelected** que vai ocorrer sempre que uma data for selecionada:

```
using Xamarin.Forms;

namespace App_DataPicker2
{
    public partial class Calendario : ContentPage
    {
        public Calendario()
        {
            InitializeComponent();
        }

        private void OnDateSelected(object sender, DateChangedEventArgs e)
        {
            lblData.Text = e.NewDate.Day + "/" + e.NewDate.Month + "/" + e.NewDate.Year;
        }
    }
}
```

Neste código obtemos o dia, mês e ano da data selecionada e exibimos na Label.

Para podermos usar a página que criamos precisamos alterar o código do arquivo **App.cs** no projeto compartilhado criando uma instância da página **Calendario.xaml** :

```
using Xamarin.Forms;

namespace App_DataPicker2
{
    public class App : Application
    {
        public App()
        {
            // The root page of your application
            MainPage = new Calendario();
        }

        protected override void OnStart()
        {
            // Handle when your app starts
        }

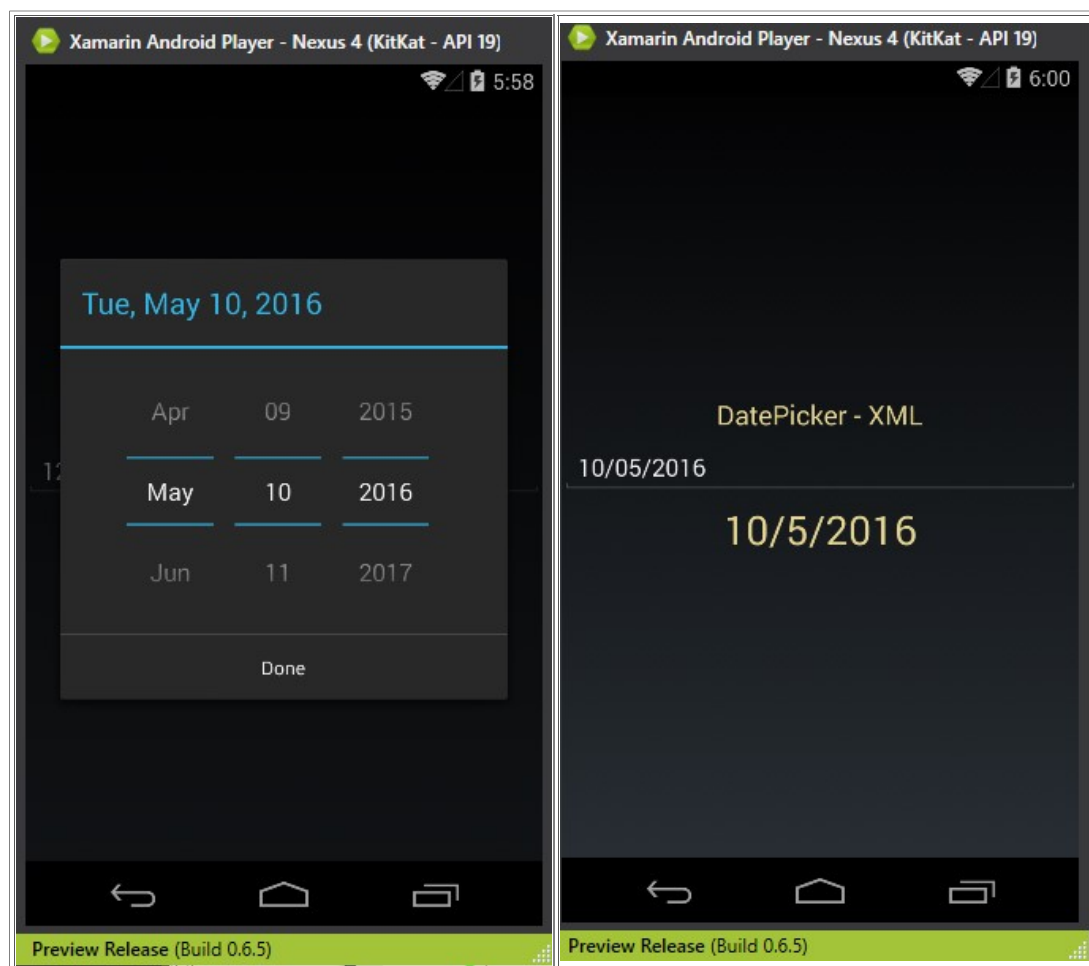
        protected override void OnSleep()
        {
            // Handle when your app sleeps
        }

        protected override void OnResume()
        {
            // Handle when your app resumes
        }
    }
}
```

```
}
```

Agora é só alegria...

Executando o projeto usando o emulador do **Xamarin Android Player** e emulando o **KitKat(API 19)** iremos obter o seguinte resultado:



Este exemplo bem simples apenas mostra como usar e selecionar datas no **DatePicker** usando o código XML.

A seguir vamos simplificar ainda mais o exemplo usando os recursos do data binding.

Usando o Data Binding

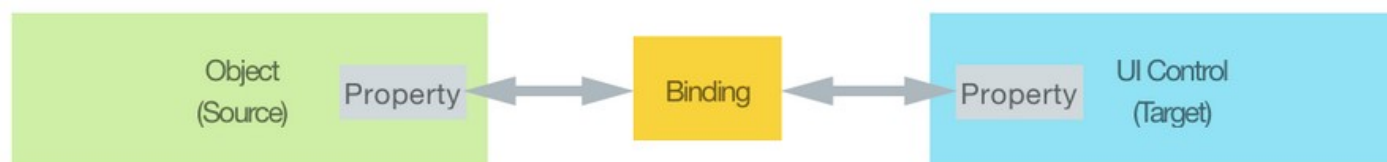
As vinculações de dados, ou **data bindings**, permitem que as propriedades de dois objetos sejam ligadas de modo que uma alteração numa provoca uma alteração na outra.

Este recurso conecta as propriedades de dois objetos chamados **source** (origem) e **target** (destino).

Os componentes do **Data Binding** :

- **Target(Destino)** - É o objeto que vai usar o resultado da vinculação (*binding*);
- **Target Property(Propriedade de destino)** - A propriedade de destino do objeto que irá utilizar o resultado;
- **Source (Origem)** - O objeto que fornece um valor para o objeto de destino usar;

- **Path(Caminho)** - Um caminho que localiza o valor dentro do objeto de origem;



A definição do data binding é um processo feito em duas etapas :

1. A propriedade **BindingContext** do objeto **target** precisa ser definida para o objeto **source**;
2. Uma vinculação precisa ser estabelecida entre o target e o source. Via XAML isso é feito usando a tag **Binding**.

Usando Data Binding podemos vincular propriedades de duas views na mesma página. É o que vamos fazer agora.

Vamos vincular a propriedade **Text** da Label - **lblData** à propriedade **Date** do **DatePicker** usando a tag **Binding** e a marcação de extensão **x:Reference** onde referenciamos o **DatePicker** via atributo **x:Name**:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="App_DataPicker2.Calendario">

    <StackLayout Orientation="Vertical" VerticalOptions="Center">
        <Label
            Text="DatePicker - XML" FontSize="20" HorizontalOptions="Center" TextColor="#e7d692"
        />

        <DatePicker
            x:Name="macDatePicker"
            MinimumDate="01/01/1970"
            MaximumDate="01/01/2100"
            Format="dd/MM/yyyy"
        />

        <Label
            x:Name="lblData"
            FontSize="30"
            Text="{Binding Date, Source={ x:Reference macDatePicker }}"
            HorizontalOptions="Center"
            TextColor="#e7d692"
        />

        <Label
            BindingContext="{x:Reference macDatePicker}"
            Text="{Binding Date, StringFormat='{0:dd MMMM, yyyy}'}"
            FontSize="30"
            TextColor="#e7d692"
            HorizontalOptions="Center"
        />
    </StackLayout>
</ContentPage>
```

```
</StackLayout>
```

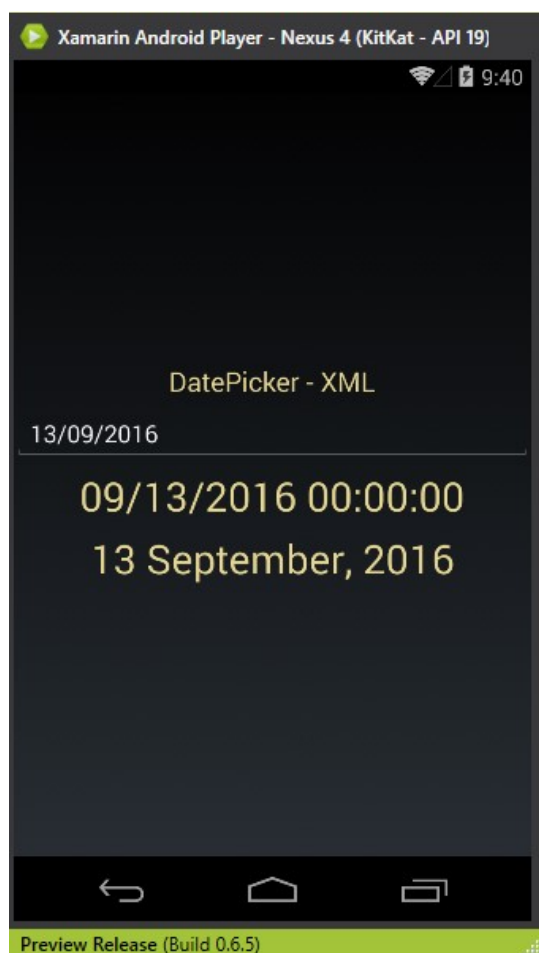
```
</ContentPage>
```

Na segunda **Label** que definimos no código XAML estamos usando o [data binding](#) usando outra sintaxe. Aqui definimos usando a propriedade **BindingContext** do objeto **target** (a *Label*) e definindo-a para o objeto **source** (o *DatePicker*).

A seguir usando a tag **Binding** definimos a propriedade **Date do source** a ser vinculada e realizamos a formatação da data usando **StringFormat**.

Dessa forma a vinculação é automática e não precisamos mais do evento **DateSelected**.

Executando novamente o projeto iremos obter:



Pegue o projeto aqui : [📁 App_DataPicker2.zip](#) (sem as referências)

Jesus lhes respondeu, e disse: A minha doutrina não é minha, mas daquele que me enviou. Se alguém quiser fazer a vontade dele, pela mesma doutrina conhecerá se ela é de Deus, ou se eu falo de mim mesmo.

[João 7:16-17](#)

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

[Quer migrar para o VB .NET ?](#)

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Video Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#) NEW

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Super DVD C# - Recursos de aprendizagens e vídeo aulas para C#](#)
- [Seção C# do site Macoratti.net](#)
- [Seção ASP .NET do site Macoratti .net](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW
- [Macoratti .net | Facebook](#)
- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macoratti\) | Twitter](#)
- [VB.NET 2005 - Controles - Macoratti.net](#)
- [Seção de Jogos do site Macoratti .net](#)
- [Xamarin - Desenvolvimento Multiplataforma com C# ... - Macoratti.net](#)
- [Xamarin - Apresentando Xamarin.Forms - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Criando sua primeira ... - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Anatomia da aplicação - Macoratti.net](#)

- <https://developer.xamarin.com/api/type/Android.App.AlertDialog/>
 - [Xamarin Android - Tratando eventos de forma declarativa](#)
 - **[Seção Mobile/Xamarin do site Macoratti .net](#)**
-

[José Carlos Macoratti](#)