



Neste artigo vou mostrar como trabalhar com lista de itens, exibindo textos e imagens em lista roláveis usando a view **ListView** e apresentando alguns dos seus recursos.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

Quando trabalhamos com listas no [Xamarin Android](#) precisamos usar um *adaptador*, geralmente a classe base **BaseAdapter**.

No [Xamarin Forms](#) não precisamos desse adaptador, e, podemos vincular a view diretamente a uma lista de strings usando a propriedade **ItemsSource** com o template de lista padrão. Podemos também vincular a view a um modelo de dados e usar lista personalizadas com a classe **ItemTemplate** interna do ListView.

Além disso podemos personalizar as linhas da lista com layouts, imagens e views como botões e realizar agrupamentos, usar cabeçalhos e rodapés e outros recursos.

A seguir veremos exemplos práticos de como usar a view **ListView**

Requisitos necessários :

- [Visual Studio 2015 Community](#) (Pode usar a partir do VS 2013)
- [Xamarin for VS](#)

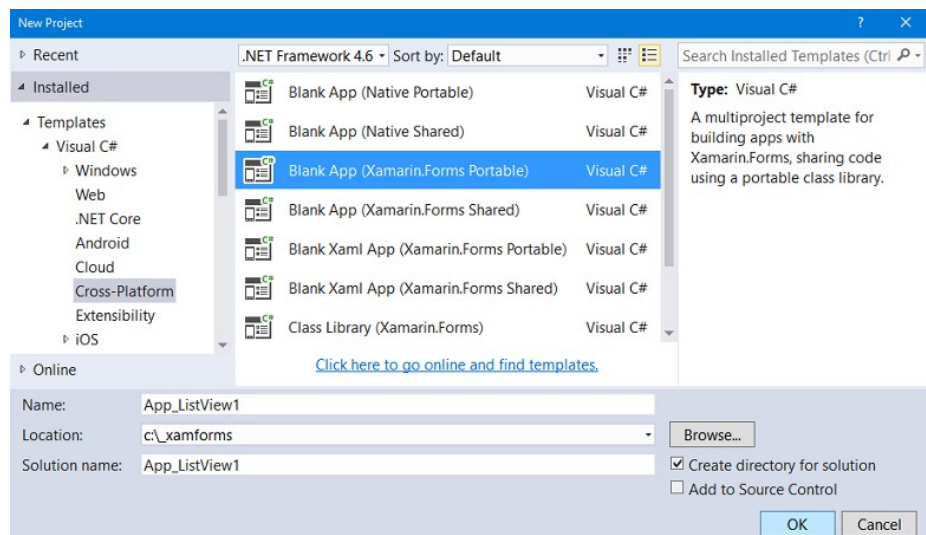
Criando a solução Xamarin.Forms no Visual Studio 2015

Abra o [Visual Studio Community 2015](#) e clique em **New Project**;

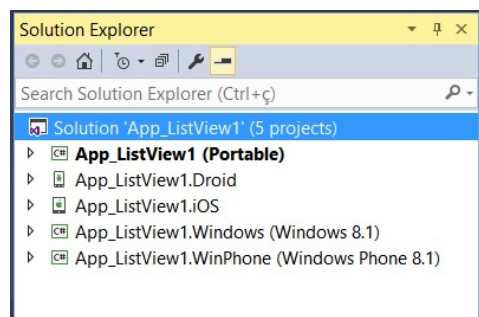
Selecione **Visual C#**, o template **Cross Platform** e a seguir **Blank App (Xamarin.Forms Portable)**;

Nota : A opção *Portable* (Portable Class Library - PCL) - Inclui todo o código comum em uma biblioteca de vínculo dinâmico (DLL) que pode então ser referenciada a partir de outros projetos;

Informe o nome **App_ListView1** e clique no botão OK;



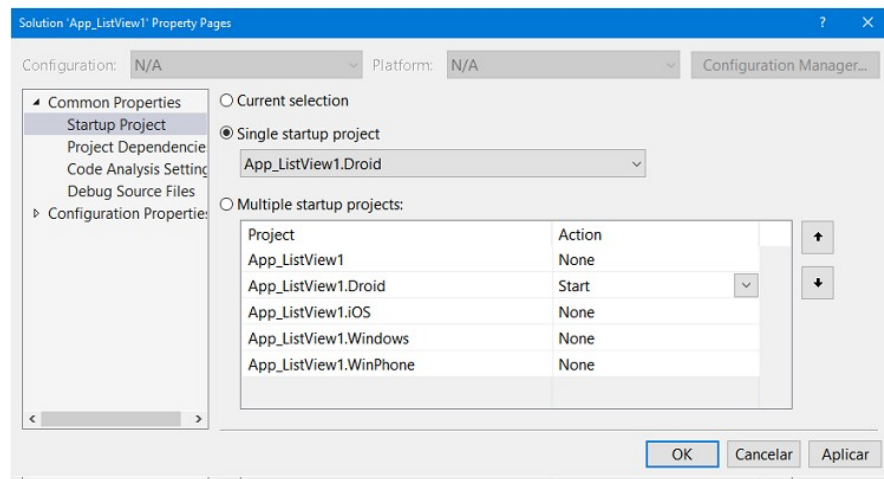
Ao clicar no botão OK, será criada uma solução contendo 5 projetos. (Dependendo do seu ambiente pode haver alguma variação nos projetos.)



Como não tenho um Mac (ainda), e a compilação e testes da aplicação ficará prejudicada, vou remover o projeto iOS da solução criada. Para isso clique com o botão sobre o projeto e a seguir clique em **Remove**. (Também vou remover o projeto Windows Phone e Windows

8.1). Dessa forma, para o meu caso, estou debugando e fazendo o deploy para o projeto Android.

Na janela de propriedades da minha solução vou definir o **startup project** conforme mostra a figura abaixo: (Você pode alterar conforme a sua necessidade esta opção)



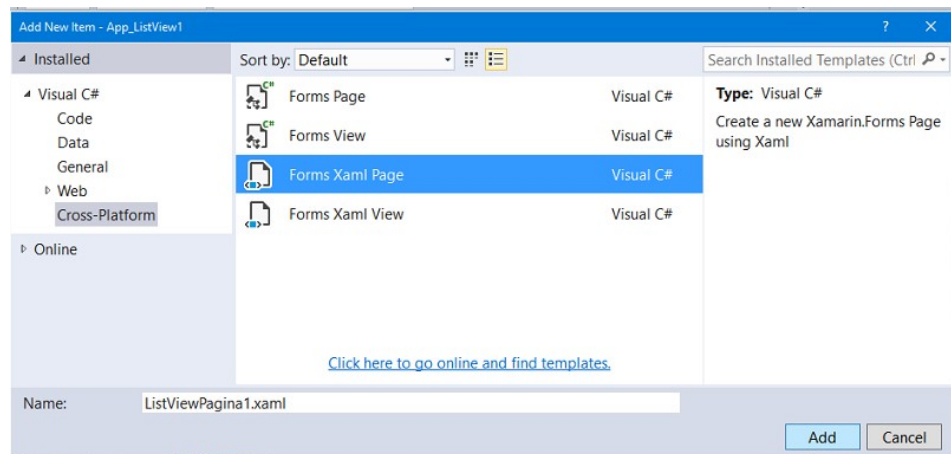
O código será definido no projeto comum **App_ListView1(Portable)** que possui a classe **App.cs** e que irá conter o código compartilhado que vamos usar neste artigo.

Assim o código poderá ser executado em qualquer plataforma (*iOs, Android, Windows Phone*).

Incluindo uma nova página XAML no projeto compartilhado (Portable)

Selecione o projeto **App_ListView1(Portable)** e no menu **Project** clique em **Add New Item**;

Selecione o template **Cross Platform -> Forms Xaml Page** e informe o nome **ListViewPagina1.xaml** e clique em **Add**;



Serão criados dois arquivos :

- **ListViewPagina1.xaml** - onde definimos o código AXML;
- **ListViewPagina1.xaml.cs** - onde definimos o código C#;

Após isso precisamos alterar o código da classe **App.cs** para criar uma instância da nossa página **ListViewPagina1** definindo-a como a página principal que será executada quando a aplicação rodar. O código deve ficar assim:

```
using Xamarin.Forms;
namespace App_ListView1
{
    public class App : Application
    {
        public App()
        {
            MainPage = new ListViewPagina1();
        }

        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 podemos definir o código nos arquivos **ListViewPagina1**.

Xamarin Forms - Criando um lista básica de itens

A forma mais simples e rápida de exibir uma lista de itens usando o ListView em um projeto Xamarin Forms e definir uma lista de strings e usar a propriedade **ItemsSource**.

Abra o arquivo **ListViewPagina1.axml** e defina um controle **ListView** com o nome **lv1**:

```

<?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_ListView1.ListViewPagina1">
    <ListView x:Name="lv1"/>
</ContentPage>

```

Agora vamos definir o código no arquivo code-behind **ListViewPagina1.axml.cs** :

```

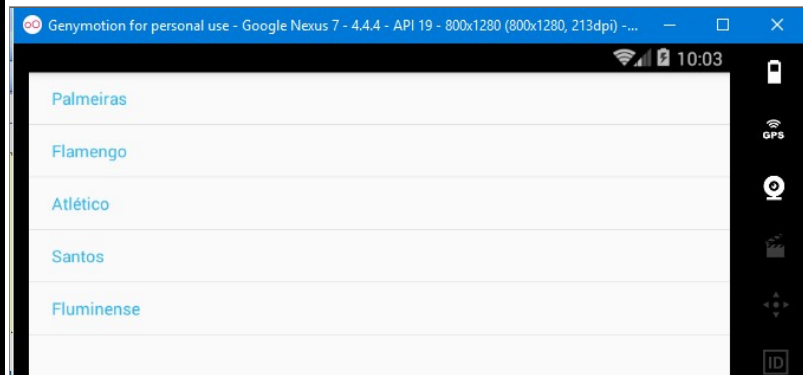
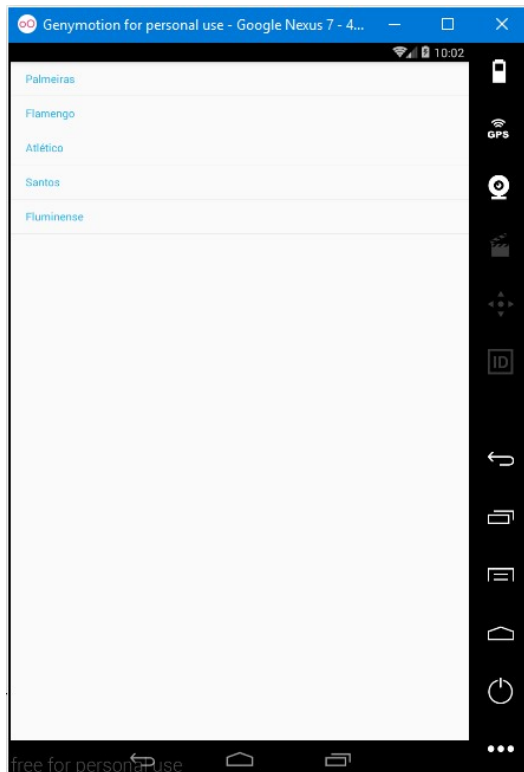
using System;
using System.Collections.Generic;
using Xamarin.Forms;

namespace App_ListView1
{
    public partial class ListViewPagina1 : ContentPage
    {
        public ListViewPagina1()
        {
            InitializeComponent();

            List<String> itens = new List<String>()
            {
                "Palmeiras", "Flamengo", "Atlético", "Santos", "Fluminense"
            };
            lv1.ItemsSource = itens;
        }
    }
}

```

Executando o projeto fazendo a emulação via projeto Android e usando o **Genymotion** como emulador iremos obter:



Podemos alterar a cor das linhas que separam os itens e também podemos ocultá-las. Para isso usamos as propriedades : **SeparatorColor** e **SeparatorVisibility (None** oculta)

```
<?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_ListView1.ListViewPagina1"
  BackgroundColor="Black">

  <ListView x:Name="lv1"
    SeparatorColor="White"
    SeparatorVisibility="Default"/>

</ContentPage>
```



Na [segunda parte do artigo](#) veremos como exibir objetos complexos em um ListView.

Pegue o projeto aqui : [App_ListView1.zip](#) (sem as referências)

E os que são de Cristo crucificaram a carne com as suas paixões e concupiscências.

Gálatas 5:24

[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

Gostou ?  [Compartilhe no Facebook](#)  [Compartilhe no Twitter](#)

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 aprendizagem 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](#)
- <https://www.visualstudio.com/pt-br/features/xamarin-vs.aspx>
- <https://xamarin.com/starter>
- [Xamarin Studio - Desenvolvimento Multiplataforma com C# \(Android, iOS e Windows\)](#)
- [Xamarin - Criando Apps com o Visual Studio e C# \(vídeo aula\)](#)
- https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_binding_basics/
- [Seção Xamarin Forms do site Macoratti .net](#)

[José Carlos Macoratti](#)