Macoratti.net Xamarin.Forms - Trabalhando com Lista de dados - ListView - II

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.

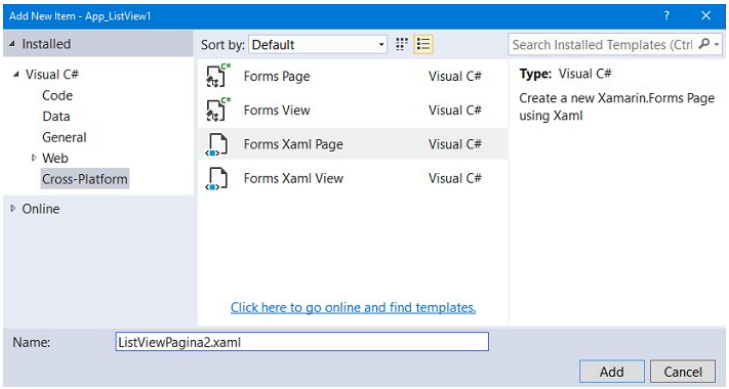
Na **primeira parte do artigo** mostrei como exibir uma lista de itens simples em uma view ListView. Vamos continuar agora exibindo uma lista de objetos complexos, uma lista de contatos.

Abra o projeto criado no artigo anterior e vamos incluir uma nova página axml.

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 Plataform -> Forms Xaml Page** e informe o nome **ListViewPagina2.xaml** e clique em **Add**;



Lembre-se que agora precisamos alterar o código da classe **App.cs** para criar uma instância da nossa página **ListViewPagina2** 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 ListViewPagina2();
        }
      .....

}
```

Agora podemos definir o código nos arquivos **ListViewPagina2**.

Definindo o modelo de dados

Vamos criar uma pasta chamada **Models** no projeto compartilhado para incluir nesta pasta a nossa classe de domínio.

No menu **Project** selecione **New Folder** e informe o nome **Models**.

Selecione a pasta **Models** e no menu **Project** clique em **Add Class** e informe o nome

```
    public class Contato
    {
        public string Nome { get; set; }
        public string Status { get; set; }
        public string ImagemUrl { get; set; }
    }
```

Agora vamos para o code-behind para definir os dados que vamos usar para os contatos. Vamos iniciar diretamente o ListView usando o código abaixo:

```
using App_ListView1.Models;
using System.Collections.Generic;

using Xamarin.Forms;

namespace App_ListView1
{
    public partial class ListViewPagina2 : ContentPage
    {
        public ListViewPagina2()
        {
            InitializeComponent();
```

```
            lv2.ItemsSource = new List<Contato>
            {
               new Contato { Nome="Jimi", Status="Ativo", ImagemUrl="http://www.macoratti.net/Imagens/pessoas/jimi1.jpg" },
               new Contato { Nome="Janis", Status="Ativo", ImagemUrl="http://www.macoratti.net/Imagens/pessoas/janis1.jpg" },
               new Contato { Nome="Lenon", Status="Pendente", ImagemUrl="http://www.macoratti.net/Imagens/pessoas/lenon1.jpg" },
               new Contato { Nome="House", Status="Ativo", ImagemUrl="http://www.macoratti.net/Imagens/pessoas/house1.jpg" }
            };
        }
    }
}
```

Para poder exibir dados complexos em um ListView vamos usar as células já existentes no Listview:

- **TextCell** - usada para exibir texto com uma segunda linha como texto detalhe. Para isso basta definir as propriedades:

  Text - para exibir a primeira linha de texto em fonte maior;
  Detail - para exibir uma segunda linha de texto em fonte menor;

- **ImageCell** - usada para exibir imagem e texto e um linha com texto detalhe. Para isso basta definir as propriedades:

  Text - para exibir a primeira linha de texto em fonte maior;
  Detail - para exibir uma segunda linha de texto em fonte menor;
  ImageSource - para exibir uma imagem próxima ao texto

  Nota: *Tanto TextCell como ImageCell possuem as propriedades **TextColor** e **DetailColor** para definir a cor do texto principal e do detalhe.*

Observação : Existem também as células **SwitchCell** e **EntryCell** que não são muito usadas com o ListView.

Então vamos usar esses recursos para exibir os dados dos contatos no ListView usando código XAML.

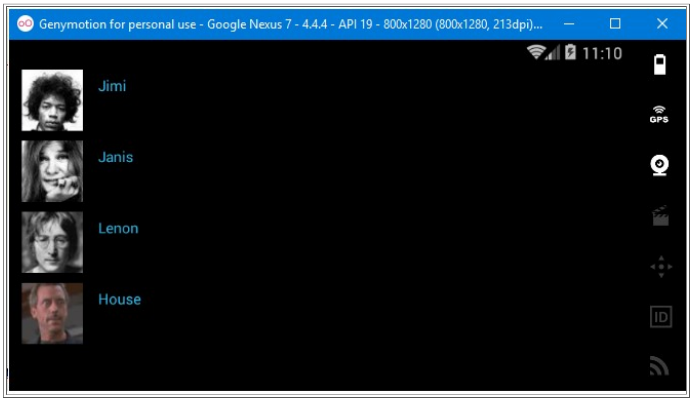
No arquivo AXML vamos definir uma view ListView e usar um **ItemTemplate** e um **DataTemplate** conforme o código a seguir:

```
<?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.ListViewPagina2"
             BackgroundColor="Black">

  <ListView x:Name="lv2">
    <ListView.ItemTemplate>
      <DataTemplate>
        <ImageCell Text="{Binding Nome}" Detail="{Binding Status}" ImageSource="{Binding ImagemUrl}"/>
      </DataTemplate>
    </ListView.ItemTemplate>
  </ListView>

</ContentPage>
```

Estamos usando um **DataTemplate** para exibir dados de uma coleção de objetos em um ListView.

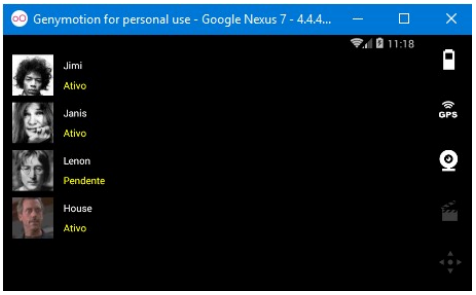
A aparência dos dados em cada célula em um ListView pode ser gerenciada pela definição da propriedade **ItemTemplate** para um **DataTemplate**, onde os elementos especificados no DataTemplate definem a aparência de cada célula.

Lembrando que um filho de um **DataTemplate** tem que ser to tipo ou derivar de um **ViewCell** onde estamos usando um **ImageCell** para exibir o texto e a imagem dos contatos:



Podemos melhorar a aparência na exibição dos itens usando as propriedades **TextColor** e **DetailColor** do **ImageCell**:

```
<?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.ListViewPagina2"
             BackgroundColor="Black">

  <ListView x:Name="lv2">

    <ListView.ItemTemplate>
      <DataTemplate>
        <ImageCell Text="{Binding Nome}" Detail="{Binding Status}" ImageSource="{Binding ImagemUrl}"
                   TextColor="White" DetailColor="Yellow"/>
      </DataTemplate>
    </ListView.ItemTemplate>
  </ListView>
</ContentPage>
```



Na **terceira parte do artigo** vamos continuar a tratar do ListView.

Pegue o projeto aqui : **AppListView2.zip** *(sem as referências)*

Jesus Cristo é o mesmo, ontem, e hoje, e eternamente.
Hebreus 13:8

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 (@macorati) | Twitter**
- **An Introduction to Xamarin.Forms - Xamarin**
- **https://www.visualstudio.com/pt-br/features/xamarin-vs.aspx**
- **https://xamarin.com/starter**
- **Xamarim Studio - Desenvolvimento Multiplataforma com C# (Android, iOS e Windows)**
- **Xamarim - Criando Apps com o Visual Studio e C#** **(vídeo aula)**
- **https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_binding_basics/**

José Carlos Macoratti