

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



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

Na [segunda parte do artigo](#) mostrei como exibir uma lista de objetos complexos, vamos continuar mostrando como customizar a aparência das células do **ListView**.

A customização da aparência das células da **ListView** é feita através da utilização de **ViewCells** que podem ser usadas para exibir imagens e texto, indicar true ou false e receber a entrada do usuário.

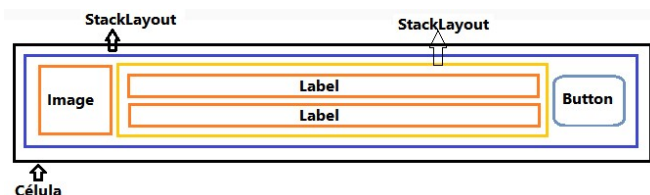
Existem duas formas de customizar a aparência das células em um **ListView**:

- Customizar usando os recursos de células já existentes como **TextCell** e **ImageCell**. (Como fizemos no artigo anterior)
- Criar células personalizadas para implementar um recurso específico;

Neste artigo vamos usar a segunda abordagem para alterar a aparência das células do **ListView** vamos incluir um controle **Button** com o texto **Siga-me** à direita no final da linha.

Abaixo temos um esboço da célula que queremos criar. Para isso vamos precisar usar as seguintes views no interior da **ViewCell**:

- **StackLayout - Orientação Horizontal**
  - **Image**
- **StackLayout - Orientação Vertical**
  - **Label**
  - **Label**
- **Button**



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 Plataforma -> Forms Xaml Page** e informe o nome **ListViewPagina3.xaml** e clique em **Add**;

Lembre-se que agora precisamos alterar o código da classe **App.cs** para criar uma instância da nossa página **ListViewPagina3** 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 ListViewPagina3();
        }
        .....
    }
}
```

Agora podemos definir o código nos arquivos **ListViewPagina3**.

Vamos usar o mesmo modelo de dados e o código do arquivo Code-Behind **ListViewPagina3.xaml.cs** será idêntico ao do artigo anterior, assim vamos apenas alterar o código do arquivo **ListViewPagina3.xaml**.

### Customizando a aparência das células do ListView

Para customizar a aparência das células temos que criar um template usando uma **ViewCell** que é uma célula que pode ser adicionado no **ListView** ou **TableView** e que pode conter views definidas pelo usuário. (A renderização de uma **ViewCell** é feita de forma diferente nas plataformas iOS, Android e Windows Phone.)

No arquivo **ListViewPagina3.xaml** vamos definir uma view **ListView** e usar um **ItemTemplate** e um **DataTemplate** e usar uma **ViewCell** conforme o código abaixo:

```
<?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.ListViewPagina3">

    <ListView x:Name="lv2">
        <ListView.ItemTemplate>
            <DataTemplate>

                <ViewCell>
                    <StackLayout Orientation="Horizontal">

                        <Image Source="{Binding ImageUrl}" />
```

```

<StackLayout HorizontalOptions="StartAndExpand">
  <Label Text="{Binding Nome}" TextColor="Black"/>
  <Label Text="{Binding Status}" TextColor="Gray"/>
</StackLayout>

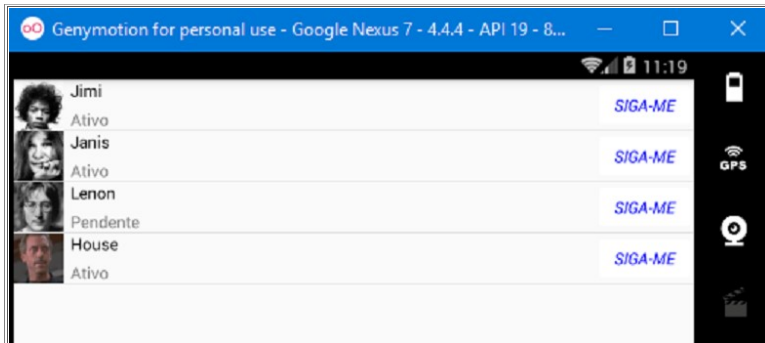
  <Button BackgroundColor="White" TextColor="Blue" Text="Siga-me" FontAttributes="Italic" />
</StackLayout>
</ViewCell>

</DataTemplate>
</ListView.ItemTemplate>
</ListView>
</ContentPage>

```

No interior da **ViewCell** definimos um **StackLayout Horizontal** que irá conter a view **Image**, o layout **StackLayout** (vertical) contendo duas **Labels** e a view **Button**.

Executando o projeto usando um emulador **Android GenyMotion** iremos obter o seguinte resultado:

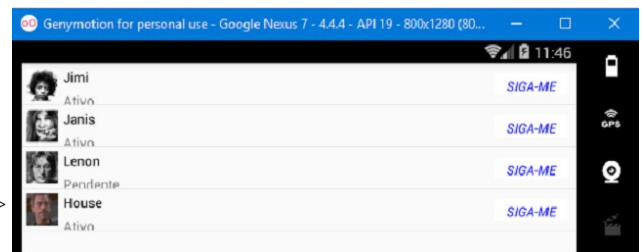


Podemos melhorar a aparência na exibição dos itens aumentando a distância entre os objetos e célula definindo um **Padding de 5 dpi** no primeiro **StackLayout**:

```

<?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.ListViewPagina3">
  <ListView x:Name="lv2">
    <ListView.ItemTemplate>
      <DataTemplate>
        <ViewCell>
          <StackLayout Orientation="Horizontal" Padding="5">
            <Image Source="{Binding ImagemUrl}" />
            <StackLayout HorizontalOptions="StartAndExpand">
              <Label Text="{Binding Nome}" TextColor="Black"/>
              <Label Text="{Binding Status}" TextColor="Gray"/>
            </StackLayout>
            <Button BackgroundColor="White" TextColor="Blue" Text="Siga-me" FontAttributes="Italic" />
          </StackLayout>
        </ViewCell>
      </DataTemplate>
    </ListView.ItemTemplate>
  </ListView>
</ContentPage>

```



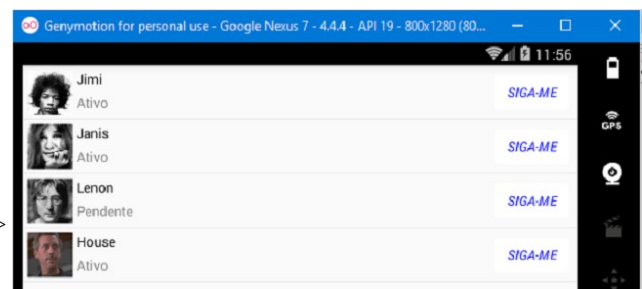
Ao executar o projeto você verá que o resultado não ficou legal pois agora a altura da célula está sobrepondo uma parte do texto da **Label**.

Para resolver este problema precisamos definir a propriedade **HasUnevenRows** do **ListView** como **true** e assim ajustar o tamanho da altura da linha ao conteúdo da célula de forma automática.

```

<?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.ListViewPagina3">
  <ListView x:Name="lv2" HasUnevenRows="true">
    <ListView.ItemTemplate>
      <DataTemplate>
        <ViewCell>
          <StackLayout Orientation="Horizontal" Padding="5">
            <Image Source="{Binding ImagemUrl}" />
            <StackLayout HorizontalOptions="StartAndExpand">
              <Label Text="{Binding Nome}" TextColor="Black"/>
              <Label Text="{Binding Status}" TextColor="Gray"/>
            </StackLayout>
            <Button BackgroundColor="White" TextColor="Blue" Text="Siga-me" FontAttributes="Italic" />
          </StackLayout>
        </ViewCell>
      </DataTemplate>
    </ListView.ItemTemplate>
  </ListView>
</ContentPage>


```



Verificando o resultado vemos que agora temos uma aparência melhor e cumprimos o nosso objetivo mostrando como customizar a aparência de uma célula usando

**ViewCell.**

Na [última parte do artigo](#) veremos como tratar a seleção de itens.

Pegue o código da página usada neste artigo:  [ListViewPagina3.zip](#) (sem as referências)

**Deitar-me faz em verdes pastos, guia-me mansamente a águas tranqüilas.Refrigera a minha alma; guia-me pelas veredas da justiça, por amor do seu nome.**

**Salmos 23:2,3**

[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 - Video Aulas](#)

**Quer aprender C# ??**

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e video 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 - Video 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 Video Aulas - Video Aula sobre VB .NET, ASP .NET e C#](#)
- [Super DVD C# - Recursos de aprendizagens e video aulas para C#](#)
- [Seção C# do site Macoratti.net](#)
- [Seção ASP .NET do site Macoratti .net](#)
- [Curso Básico VB .NET - Video Aulas](#)
- [Curso C# Básico - Video Aulas](#)
- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW
- [Macoratti .net | Facebook](#)
- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macoratti\) | 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# \(video aula\)](#)
- [https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data\\_binding\\_basics/](https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_binding_basics/)

---

[José Carlos Macoratti](#)