



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 [terceira parte do artigo](#) mostrei como customizar a aparência das células do ListView e agora vamos podemos tratar a interatividade via seleção dos itens.

A view ListView suporta a interação com os dados da lista através dos seguintes recursos:

- **Seleção e Toque (Tap)** - Responde a toques e *seleção/deseleção* de itens. Habilita ou desabilita a seleção de linha (*habilitada por padrão*);
- **Ações de Contexto (Context Actions)** - Expõe a funcionalidade por item;
- **Arraste para Atualizar (Pull to Refresh)** - Implementa o recurso nativo *puxe-para-atualizar* das plataformas;

Seleção e Toque (Tap)

A ListView suporta seleção de um item de cada vez, sendo que a seleção esta ativada por padrão. Assim, quando um usuário toca em um item da lista, dois eventos são disparados:

1. **ItemTapped**
2. **ItemSelected**

Na primeira vez que você seleciona ou toca um item o evento **ItemTapped** é disparado e logo a seguir o evento **ItemSelected** é disparado.

Mas se você tocar/selecionar várias vezes o mesmo item vai disparar vários eventos **ItemTapped** mas não vai disparar vários eventos **ItemSelected**.

Vamos mostrar na prática como isso funciona.

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 Platform -> 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 **ListViewPagina4** 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 ListViewPagina4();
        }
        .....
    }
}
```

Agora podemos definir o código nos arquivos **ListViewPagina4**.

Vamos usar o mesmo modelo de dados e o código do arquivo Code-Behind **ListViewPagina4.xaml.cs** será idêntico ao do artigo anterior, assim vamos apenas alterar o código do arquivo **ListViewPagina4.xaml**.

Tratando a seleção de itens : ItemSelected e ItemTapped

Para tratar a seleção de itens vamos definir no arquivo **ListViewPagina4.xaml** os eventos **ItemSelected** e **ItemTapped** que deverão ser tratados no arquivo code-behind.



Agora no arquivo code-behind **ListViewPagina4.xaml.cs** vamos implementar o tratamento dos eventos exibindo os itens selecionados usando a propriedade **SelectedItem** e o método **DisplayAlert()**.

```

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

using Xamarin.Forms;

namespace App_ListView1
{
    public partial class ListViewPagina4 : ContentPage
    {
        public ListViewPagina4()
        {
            InitializeComponent();
            lv4.ItemsSource = new List<Contato>
            {
                new Contato { Nome="Adele", Status="Ativo", ImageUrl="http://www.macoratti.net/Imagens/semfoto641.jpg" },
                new Contato { Nome="Beatles", Status="Ativo", ImageUrl="http://www.macoratti.net/Imagens/semfoto641.jpg" },
                new Contato { Nome="Jimi", Status="Pendente", ImageUrl="http://www.macoratti.net/Imagens/jimi1.jpg" },
                new Contato { Nome="Janis", Status="Ativo", ImageUrl="http://www.macoratti.net/Imagens/janis1.jpg" },
                new Contato { Nome="Lady Gaga", Status="Ativo", ImageUrl="http://www.macoratti.net/Imagens/semfoto641.jpg" },
                new Contato { Nome="Lenon", Status="Pendente", ImageUrl="http://www.macoratti.net/Imagens/lenon1.jpg" },
                new Contato { Nome="Shakira", Status="Ativo", ImageUrl="http://www.macoratti.net/Imagens/semfoto641.jpg" },
                new Contato { Nome="Pearl Jam", Status="Suspense", ImageUrl="http://www.macoratti.net/Imagens/semfoto641.jpg" },
                new Contato { Nome="House", Status="Ativo", ImageUrl="http://www.macoratti.net/Imagens/house1.jpg" }
            };
        }

        public void Handle_ItemSelected(object sender, SelectedItemChangedEventArgs e)
        {
            var contato = e.SelectedItem as Contato;
            DisplayAlert("Item Selecionado (SelectedItem) ", contato.Nome, "Ok");
        }
    }
}

```

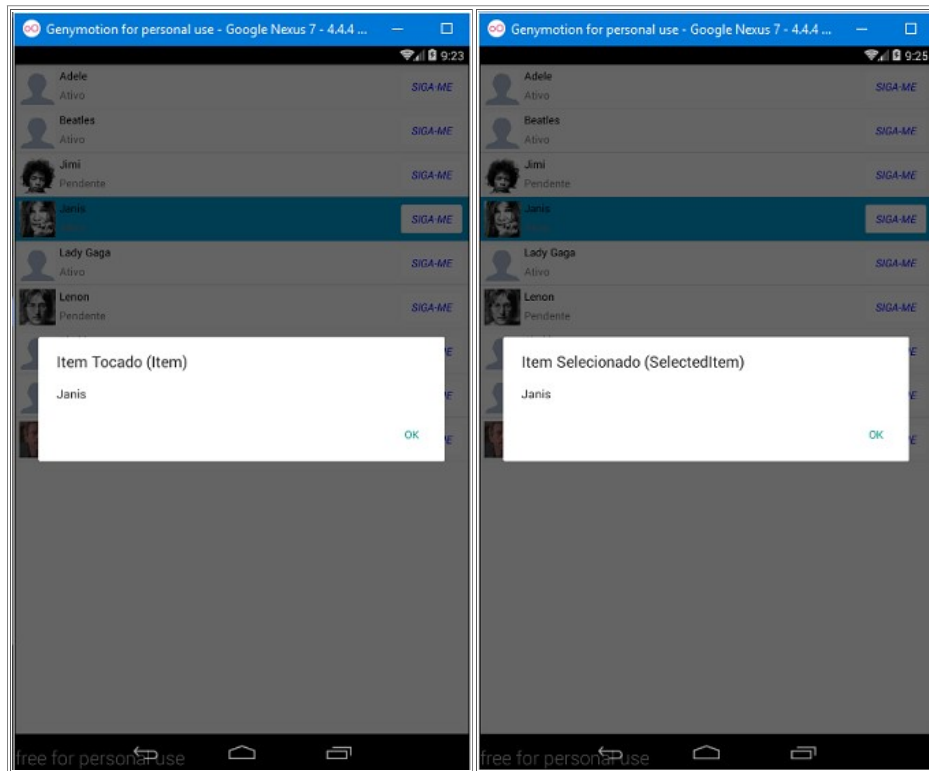
```

    }

    public void Handle_ItemTapped(object sender, ItemTappedEventArgs e)
    {
        var contato = e.Item as Contato;
        DisplayAlert("Item Tocado (Item) ", contato.Nome, "Ok");
    }
}

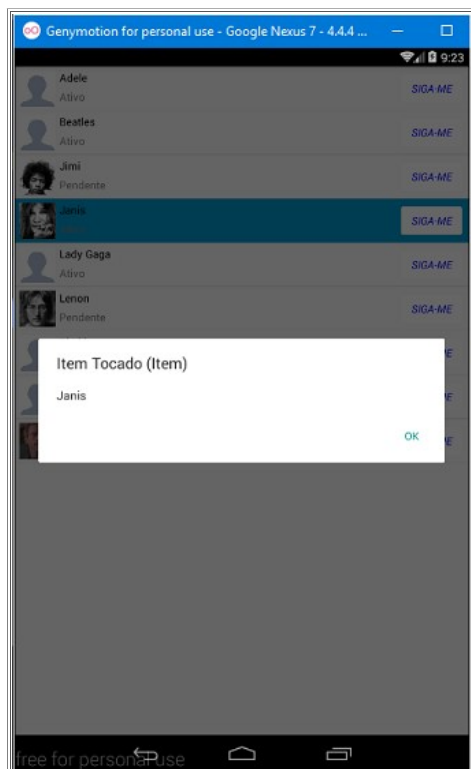
```

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



Testando no Android, ao selecionar um item, primeiro o evento **ItemTapped** é disparado e depois o evento **ItemSelected**.

Agora clicando novamente no item já selecionado iremos obter:



Agora temos que ao selecionar novamente um item já selecionado somente o evento **ItemTapped** é disparado.

E se você desejar desabilitar o evento **ItemSelected** ?

Simples.

Basta atribuir **null** à propriedade **SelectedItem** do ListView:

```
public void Handle_ItemSelected(object sender, SelectedItemChangedEventArgs e)
{
    lv4.SelectedItem = null;
}
```

Agora ao selecionar um item somente o evento **ItemTapped** irá ser disparado.

E para fazer a mesma coisa com o evento **ItemTapped** ?

Mais simples ainda. Basta não tratar o evento:

```
public void Handle_ItemTapped(object sender, ItemTappedEventArgs e)
{
}
```

Agora nenhum item poderá ser selecionado ou tocado na lista.

Pegue o código da página usada neste artigo:: [📁 ListViewPagina4.zip](#) (sem as referências)

Porque toda a carne é como a erva, e toda a glória do homem como a flor da erva. Secou-se a erva, e caiu a sua flor; Mas a palavra do Senhor permanece para sempre. E esta é a palavra que entre vós foi evangelizada.

1 Pedro 1:24-25

[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 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](#)
- **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](#)