

Macoratti.net Xamarin Android - Ordenando uma ListView



Neste artigo vou mostrar como fazer a ordenação dos dados exibidos em uma ListView no [Xamarin Android](#) usando o Visual Studio 2015 e a linguagem C#.



Curso de Xamarin Forms Vídeo Aulas

Desenvolva para Android, iOS e Windows Phone

Este é um artigo para iniciantes que mostra de forma bem simples como podemos ordenar informações em um controle ListView.

O artigo trabalha basicamente com array de dados e, não usa **MVVM**, o que seria mais indicado para o caso.

Em meu artigo [Apresentando e usando o controle ListView](#) eu apresentei o controle **ListView** e mostrei uma forma de utilizar o controle em aplicações [Xamarin Android](#).

O **ListView** é um componente UI importante de aplicações Android, muito usado para exibir listas curtas de opções de menu a até longas listas de opções. Ele fornece uma maneira simples de apresentar uma lista de rolagem de linhas que podem ser formatadas com um estilo embutido ou personalizados.

O ListView requer um adaptador para alimentá-lo com dados e, de maneira geral para adicionar linhas em um ListView, precisamos incluí-lo em nosso Layout e implementar um **ListAdapter** com os métodos que o ListView chama para se auto preencher.

Usar um **ArrayAdapter<string>** é a maneira mais fácil de usar o ListView devido à sua simplicidade, mas ele serve basicamente para exibir apenas uma linha por vez e seus recursos são limitados.

Vamos usar essa abordagem e mostrar como ordenar os itens em uma ListView.

Recursos usados:

- [Visual Studio Community 2017](#) ou [Xamarin Studio](#)
- [Xamarin](#)

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 2017

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

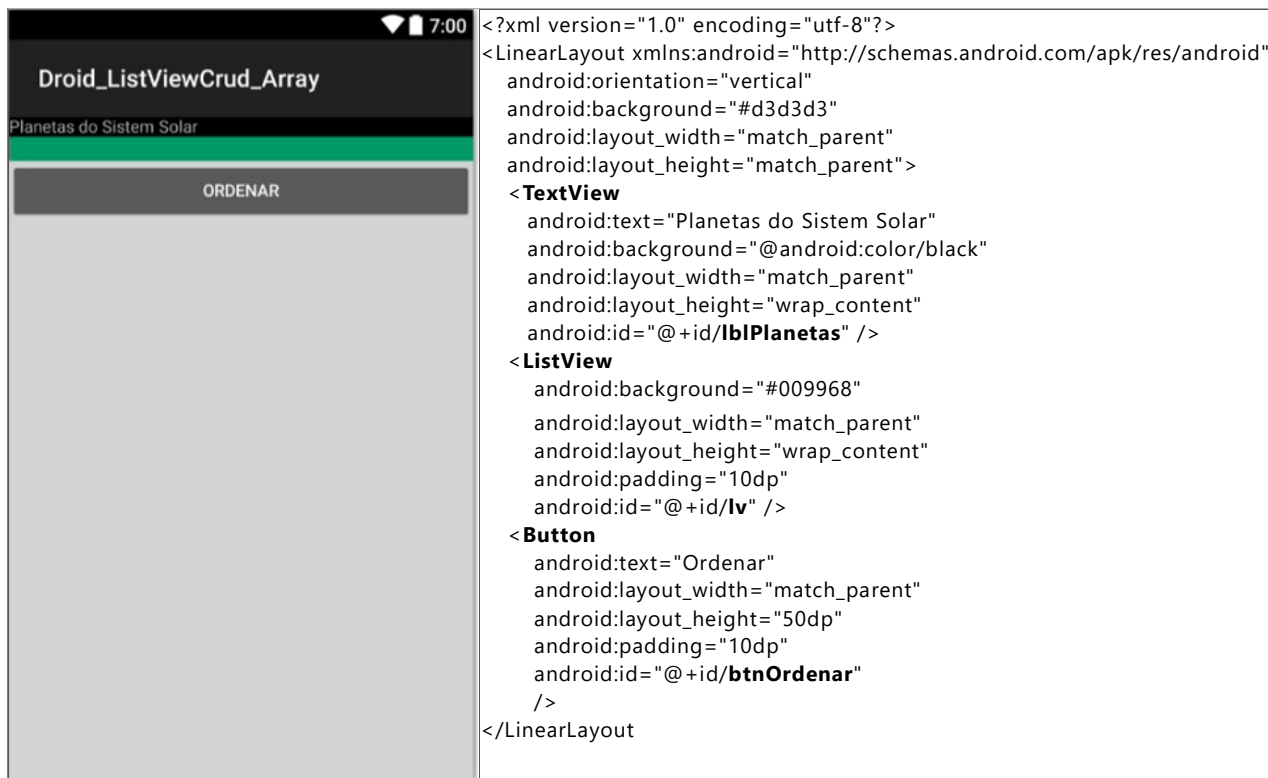
Selecione a linguagem Visual C# e o template **Android -> Blank App(Android)**

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

Abra o arquivo **Main.axml** na pasta **Resources/layout** e no modo **Designer** e a seguir inclua a partir da ToolBox os seguintes controles:

- [TextView - lblPlanetas](#)
- [ListView - lv](#)
- [Button - btnOrdenar](#)

Abaixo vemos o leiaute no emulador do Xamarin e ao lado o respectivo código XML gerado :



Agora vamos definir o código no arquivo **MainActivity.cs** vinculado a nossa view **Main.axml**.

Definindo os dados e fazendo a ordenação

Abra o arquivo **MainActivity.cs** e altere o código desse arquivo conforme abaixo:

```
using System;

namespace Droid_ListViewCrud_Array
{
    [Activity(Label = "Droid_ListView_Ordenacao", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        //Views ListView e Button
        private ListView lv;
        private Button btnOrdenar;

        //define um array de strings como dados e define a ordem ascendente
        private readonly string[] planetas = { "Júpiter", "Netuno", "Urano", "Saturno", "Marte", "Terra", "Mércurio", "Vênus" };
        private bool ascendente = true;

        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);

            // Set our view from the "main" layout resource
            SetContentView (Resource.Layout.Main);

            //inicializa as views e ordena os dados
            this.InicializarViews();
            this.OrdenaDados(ascendente);
            this.ascendente = !ascendente;
            btnOrdenar.Click += BtnOrdenar_Click;
        }

        private void BtnOrdenar_Click(object sender, EventArgs e)
        {
            //ordena os dados ascendente
        }
    }
}
```

```

        OrdenaDados(ascendente);
        this.ascendente = !ascendente;
    }

    //Inicializar Views
    private void InicializarViews()
    {
        lv = FindViewById<ListView>(Resource.Id.lv);
        btnOrdenar = FindViewById<Button>(Resource.Id.btnOrdenar);
        lv.ItemClick += Lv_ItemClick;
    }

    //exibe o nome o planeta selecionado ao clicar em um item no ListView
    private void Lv_ItemClick(object sender, AdapterView.ItemClickEventArgs e)
    {
        using (var dialog = new AlertDialog.Builder(this))
        {
            int posicao = e.Position;
            string valor = planetas[posicao];
            dialog.SetTitle("Planeta Selecionado");
            dialog.SetMessage(valor);
            dialog.Show();
        }
    }

    //Preencher Listview
    private void PopularListView()
    {
        lv.Adapter = new ArrayAdapter(this, Android.Resource.Layout.SimpleListItem1, planetas);
    }
    /*
    * Ordenar
    */
    private void OrdenaDados(bool asc)
    {
        //Ordena o array : ascendente e descendente
        if (asc)
        {
            Array.Sort(planetas);
        }
        else
        {
            Array.Reverse(planetas);
        }
        //Limpa e preenche o Listview (sem MVVM)
        PopularListView();
    }
}

```

O código é bem simples e já está comentado mas basicamente definimos os controles a serem usados , preenchemos o ListView e implementamos o método **OrdenaDados()** para ordenação.

A ordenação utiliza os seguintes métodos da classe Array:

- **Sort** - Classifica os elementos em todo um Array unidimensional usando a implementação [Comparable](#) de cada elemento do Array.
- **Reverse** - Inverte a sequência dos elementos em todo Array unidimensional.

No exemplo estamos usando a lista de strings definida em **planetas**, e estamos usando um leiaute de linha existente chamado **SimpleListItem1**, que representa uma única linha de texto, para definir a aparência do ListView. Este layout de item contém um único **TextView** permitindo exibir uma única linha de texto.

Nota: Existem diversos layouts de itens de lista incorporados ao Xamarin.Android como : *SimpleListItem2* , *TwoLineListItem* , *ActivityListItem* , *SimpleListItem2* , *TestListItem* , etc.

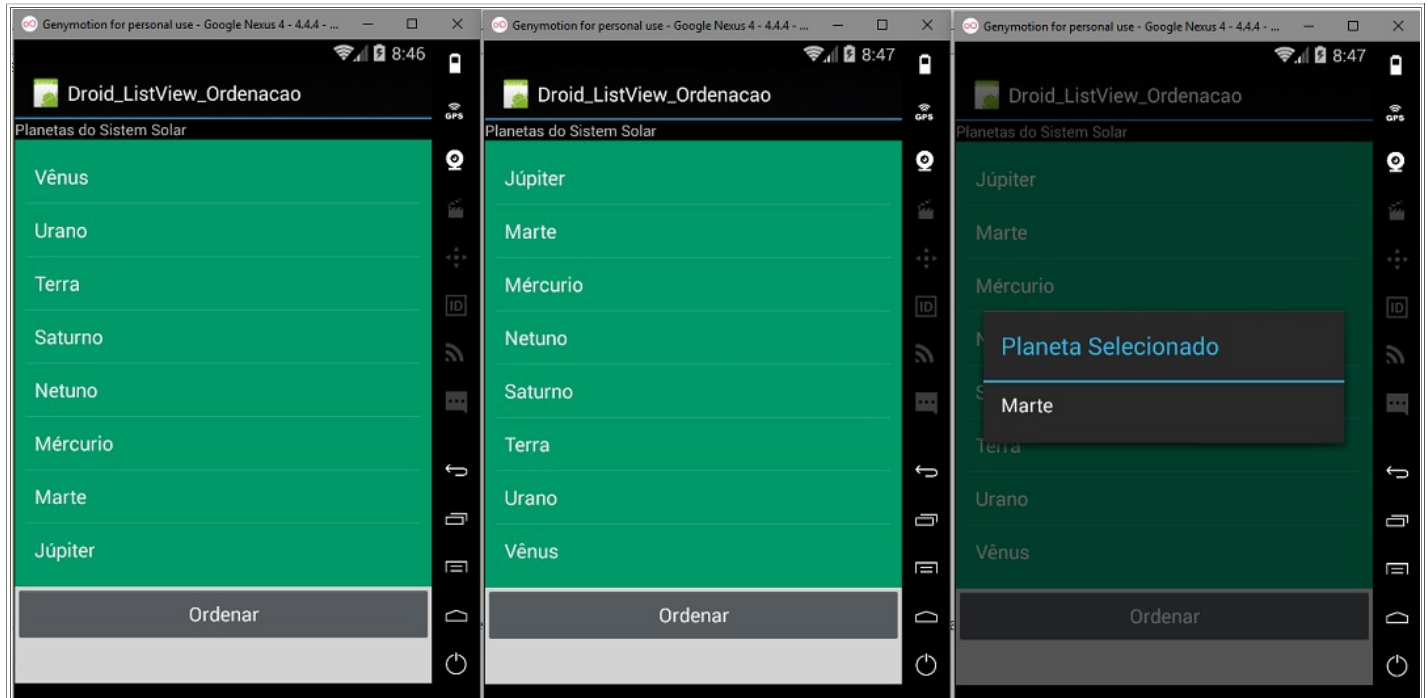
lv.Adapter = new ArrayAdapter(this, Android.Resource.Layout.SimpleListItem1, planetas);

Os argumentos usados são:

- O primeiro argumento é **this** : é o contexto da aplicação;
- O segundo argumento é o layout definido no arquivo XML que possui o **TextView** para cada item do array. Estamos usando : **SimpleListItem1**
- O terceiro argumento é o **array** de strings que será usado para preencher o texto da view;

Executando o projeto usando o emulador **Genymotion** iremos obter o seguinte resultado:

Na primeira imagem temos a ordenação ascendente, ao clicar no botão a ordenação é invertida e ao selecionar um item o nome do item é exibido na mensagem de alerta:



Pegue o projeto aqui : [Droid ListView Ordem.zip](#) (sem as referências)

O Senhor reinará eterna e perpetuamente; Êxodo 15:18

[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](#)
- <https://developer.android.com/reference/android/app/Activity.html>
- <https://developer.xamarin.com/api/type/Android.Widget.ProgressBar/>
- [Xamarin Android - ListView - Customizando um Adapter - Macoratti](#)
- [Xamarin Android - Usando SearchView com ListView - Macoratti](#)
- [Xamarin Android - ListView - Usando uma ListActivity - Macoratti](#)

[José Carlos Macoratti](#)