

Macoratti.net Xamarin Android - Apresentando e usando o componente Spinner - I



Neste artigo vou apresentar os conceitos básicos sobre o componente **Spinner** e sua utilização em aplicações **Android** usando o **Visual Studio com Xamarin** e a linguagem **C#**.

Curso C# Vídeo Aulas
Do básico ao intermediário

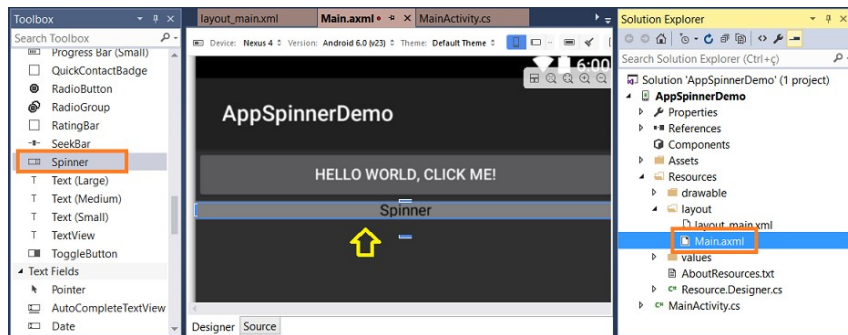
Por um preço justo

Um **Spinner** é um componente visual do Android (*um widget*) que permite selecionar um valor de um conjunto de dados.

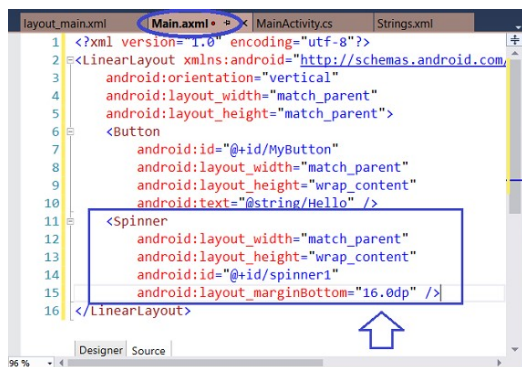
No estado padrão ele exibe o valor atualmente selecionado e, quando você interage com o controle ele exibe um menu suspenso com os demais valores e o usuário pode selecionar um novo item. (Por padrão a classe **Spinner** espera que o recurso fornecido seja referenciado como um simples **TextView**).

Nota: Um **Spinner** se parece com o controle **Combobox** do Windows Forms, **DropDownlist** da ASP .NET ou com a tag **Select** do HTML.

Para incluir um **Spinner** em uma view podemos usar a caixa de ferramentas (**ToolBox**) no ambiente do Visual Studio:



Ou podemos também definir o componente no arquivo XML que representa a view :



Os elementos que serão exibidos no **Spinner** podem ser obtidos de diversas formas :

- **ArrayList** ;
- Array de strings e/ou imagens definido no arquivo de recursos (Resources);
- De uma consulta a um banco de dados ;

De forma geral para preencher um **Spinner** com os dados desejados precisamos usar um **ArrayAdapter** que é uma classe concreta que implementa **BaseAdapter** e se baseia em um array de objetos e que funciona como um adapter.

Um **Adapter (adaptador)** serve como ponte entre os componentes de interface do usuário (UI) e a fonte de dados que preenche os dados nestes componentes. Ele trata e envia os dados para a respectiva view que pode obter esses dados exibindo-os a seguir em diferentes tipos de views como **spinner**, **list view**, **grid view**, etc.

O Xamarin Android fornece várias subclasses de adaptador que são úteis para recuperar diferentes tipos de dados e **views** para uma **AdapterView**.

Os adaptadores mais comuns são **ArrayAdapter**, **BaseAdapter**, **CursorAdapter**, **SimpleCursorAdapter**, **SpinnerAdapter** e **WrapperListAdapter**.

Vamos ver na prática como exibir uma lista de itens de uma fonte de dados usando o **Spinner** em uma aplicação Android criada no Visual Studio com Xamarin.

Recursos usados:

- [Visual Studio Community 2015](#) ou **Xamarin Studio**
- [Xamarin](#)
- Emulador Android virtual ou físico ([veja como emular usando o Vysor](#))

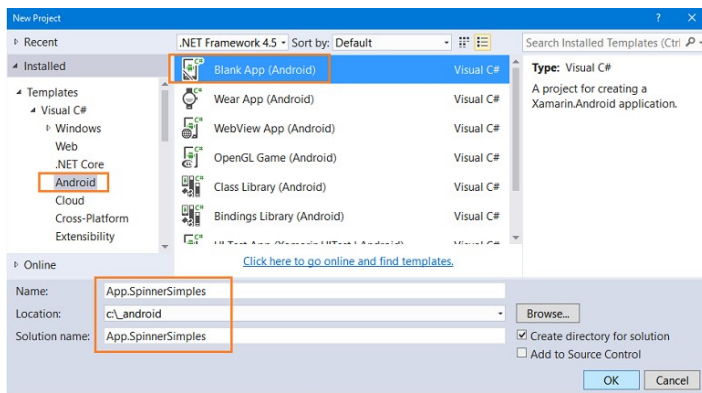
Nota: Baixe e use a versão **Community 2015** do VS ela é grátis e é equivalente a versão **Professional**.

Criando o projeto no Visual Studio 2015 Community

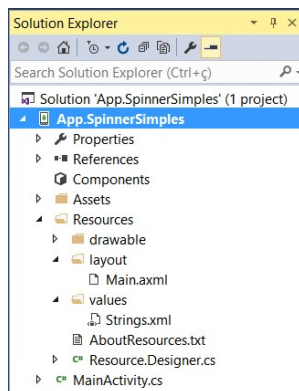
Abra o **VS 2015 Community** e clique em **New Project**;

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

Informe o nome **App.SpinnerSimples** e clique no botão **OK**;



Será criada uma solução com a seguinte estrutura:



- **Properties** - Contém o arquivo **AndroidManifest.xml** que descreve as funcionalidades e requisitos da sua aplicação Android, e o arquivo **AssemblyInfo.cs** contém informação sobre o projeto como número de versão e build.

- **References** - Contém as bibliotecas **Mono.Android**, **System.Core** e todas as bibliotecas usadas no seu projeto;

- **Components** - Contém componentes de terceiros ou desenvolvidos por você usados no seu projeto.

A maioria dos componentes está disponíveis diretamente do **Xamarin Component Store** e são **free** (não todos) e prontos para serem usados; (Para incluir um componente clique com o botão direito sobre **Components** e a seguir em **Get More Components**);

- **Assets e Resources** - Contém arquivos que não são código, como imagens, sons, arquivos XML e qualquer outro recurso que sua aplicação for usar. Os arquivos externos colocados na pasta **Assets** são facilmente acessíveis em tempo de execução através do **Asset Manager**.

Já os arquivos colocados na pasta **Resources** precisam ser declarados e mantidos em uma lista com os **IDs** dos recursos que você deseja usar em tempo de execução.

De forma geral, todas as imagens, ícones, sons e outros arquivos externos são colocados na pasta **Resources** enquanto que dicionários e arquivos XML são postos na pasta **Assets**;

Na subpasta **layout** temos os arquivos **.axml** que definem as **views** usadas no projeto;

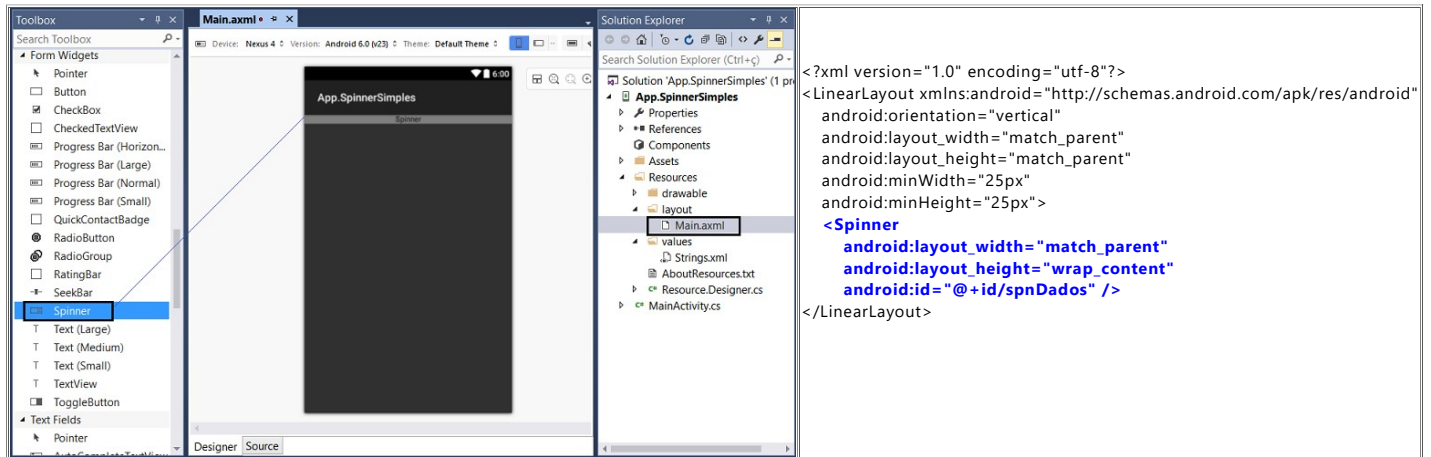
Na subpasta **values** temos o arquivo **Strings.xml** onde definimos as strings usadas no projeto;

O arquivo **MainActivity.cs** é um arquivo C# que define a atividade principal da aplicação Android.

1- Vamos Abrir o arquivo **Main.axml** na pasta **Resources/layout** e no modo **Designer** incluir um novo controle **Spinner** a partir da **ToolBox** e definir as seguintes propriedades:

- **id = @+id/spnDados**

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



Agora podemos iniciar a implementação do código para exibir itens no **Spinner** no arquivo **MainActivity.cs**.

Vamos começar definindo os namespaces usados no projeto:

```
using Android.App;
using Android.OS;
using Android.Widget;
using System.Collections;
```

A seguir a declaração da Activity como sendo a principal e a definição do ícone da aplicação:

```
[Activity(Label = "App.SpinnerSimples", MainLauncher = true, Icon = "@drawable/icon")]
```

Antes de implementar o método **OnCreate()** vamos definir as variáveis que iremos usar :

```
Spinner spinner;
ArrayAdapter adapter;
ArrayList estados;
```

No método **OnCreate()** vamos incluir o código abaixo:

```
protected override void OnCreate(Bundle bundle)
{
    base.OnCreate(bundle);
    //define a view a ser exibida
    SetContentView(Resource.Layout.Main);
    //preenche o arraylist com os dados
    GetEstados();
    //cria a instância do spinner declarado no arquivo Main
    spinner = FindViewById<Spinner>(Resource.Id.spnDados);
    //cria o adapter usando o leiaute SimpleListItem e o arraylist
    adapter = new ArrayAdapter(this, Android.Resource.Layout.SimpleListItem1, estados);
    //vincula o adaptador ao controle spinner
    spinner.Adapter = adapter;
    //define o evento ItemSelected para exibir o item selecionado
    spinner.ItemSelected += Spinner_ItemSelected;
}
```

Por padrão, o **ArrayAdapter** cria uma view para cada item do array chamando **toString()** em cada item e coloca o conteúdo em um **TextView**.

No exemplo estamos usando a lista de strings definida em **estados**, 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.

adapter = new ArrayAdapter(this, Android.Resource.Layout.SimpleListItem1, estados);

Os argumentos usados são:

- O primeiro argumento é **this** : é o contexto da aplicação;
- O segundo argumento é o leiaute 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;

Concluindo usamos a propriedade **Adapter** que retorna o adaptador atualmente em uso nesta **ListView** e exibe os dados na view:

spinner.Adapter = adapter;

O método **GetEstados()** vai apenas preencher o arraylist com as informações que desejamos exibir:

```
private void GetEstados()
{
    estados = new ArrayList();
    estados.Add("São Paulo");
    estados.Add("Rio de Janeiro");
    estados.Add("Minas Gerais");
    estados.Add("Paraná");
    estados.Add("Santa Catarina");
    estados.Add("Rio Grande do Sul");
    estados.Add("Espírito Santo");
}
```

Concluindo definindo o evento **ItemSelected** do Spinner para exibir o item selecionado:

spinner.ItemSelected += Spinner_ItemSelected;

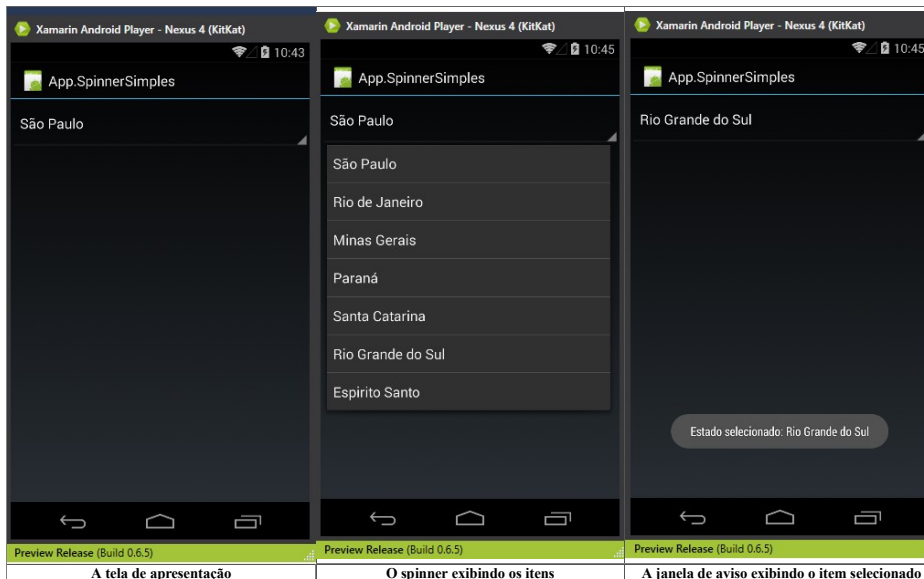
No evento **ItemSelected** do **Spinner** definimos uma janela de aviso usando a classe **Toast** para exibir qual o item foi selecionado :

```
private void Spinner_ItemSelected(object sender, AdapterView.ItemSelectedEventArgs e)
{
    Spinner spinner = (Spinner)sender;

    string toast = string.Format("Estado selecionado: {0}", spinner.GetItemAtPosition(e.Position));
    Toast.MakeText(this, toast, ToastLength.Long).Show();
}
```

Para saber mais sobre diálogos de alerta e avisos veja o artigo : [Xamarin Android - Exibindo uma janela de Alerta com AlertDialog.Builder](http://www.macoratti.net/16/07/xamand_spn1.htm)

Executando o projeto iremos obter o seguinte resultado:



Aguarde em breve mais artigos sobre o componente **Spinner** em aplicações Xamarin Android.

Pegue o projeto completo aqui : [App.SpinnerSimples.zip](#) (sem as referências)

Por isso sinto prazer nas fraquezas, nas injúrias, nas necessidades, nas perseguições, nas angústias por amor de Cristo. Porque quando estou fraco então sou forte.
2 Coríntios 12:10-10

[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# Básico - Vídeo 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](#)
- [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.Widget.ListView/>
- <https://developer.xamarin.com/api/property/Android.Widget.ListView.Adapter/>

[José Carlos Macoratti](#)