

## Macoratti.net Xamarin Android - Apresentando e usando o controle autoCompleteTextView



Neste artigo vou apresentar o controle (*widget*)

**AutoCompleteTextView** e mostrar como usar o controle em uma aplicação **Xamarin Android** usando o **Visual Studio Community 2015** e a linguagem **C#**.

Curso C# Vídeo Aulas  
Do básico ao intermediário

Por um preço justo

O **AutoCompleteTextView** é um *text view* editável que mostra sugestões automaticamente enquanto o usuário esta digitando o texto.

A lista de sugestões é geralmente exibida em um menu suspenso no qual o usuário pode escolher um item para substituir o conteúdo da caixa de edição.

As sugestões exibidas no menu suspenso podem ser descartadas a qualquer momento pressionando a tecla de retorno ou, se nenhum item for selecionado no menu suspenso, pressionando a tecla **Enter**.

A lista de sugestões é obtida a partir de um adaptador de dados e aparece apenas depois de um determinado número de caracteres definidos na propriedade **Threshold** do controle.

No exemplo deste artigo vou definir no arquivo **Main.axml** da pasta **Resources/layout** uma view contendo um controle **AutoCompleteTextView** que vai exibir uma lista de sugestões autocompletar com nomes de estados enquanto o usuário digita na *Text View*.

No exemplo mostrado iremos realizar as seguintes tarefas:

- Criar um array de strings contendo os nomes dos estados que desejamos exibir como sugestão;
- Criar um **Adapter** contendo o array de strings e definindo o modo de exibição;
- Definir a propriedade **Threshold** com o número de caracteres que devem ser digitados para que a sugestão comece a ser exibida;

### 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**;

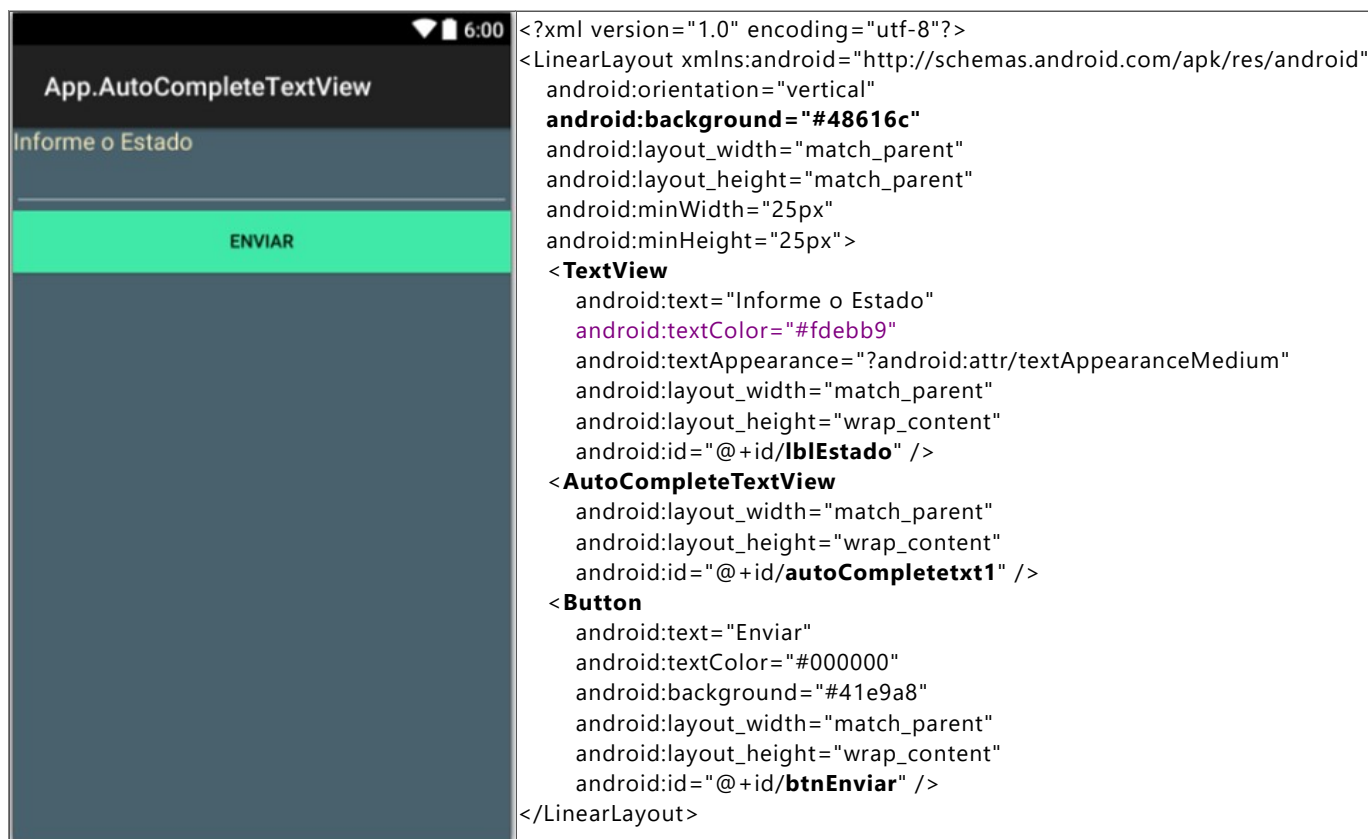
Selecione a linguagem Visual C# e o template **Android -> Blank App(Android)**

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

Abra o arquivo **Main.axml** na pasta **Resources/layout** e no modo **Designer** inclua os seguintes controles

- 1 **TextView (Mediu)** - Text = Informe o estado , id = lblEstado
- 1 **AutoCompleteTextView** - id = autoCompleteetxt1
- 1 **Button** - id = btnEnviar

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



A seguir abra o arquivo **MainActivity.cs** e altere o código desse arquivo conforme abaixo:

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

namespace App.DemoAutoCompleteTextView
{
    [Activity(Label = "App.AutoCompleteTextView", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        AutoCompleteTextView autoComplete1;
        Button btnEnviar;
        string[] estados;

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

            // define a view a partir do recurso de layout Main
            SetContentView(Resource.Layout.Main);

            //define um array de strings com os nomes dos estados a serem usados como sugestão
            estados = new string[] { "Amazonas", "Alagoas", "Amapá", "Santa Catarina", "Sergipe", "São Paulo", "Roraima",
                "Rio de Janeiro", "Rio Grande do Sul", "Minas Gerais", "Mato Grosso" };

            //cria uma instância do AutoCompleteTextView e atribui 1 caractere a partir do qual a sugestão começa a ser exibida
            autoComplete1 = FindViewById<AutoCompleteTextView>(Resource.Id.autoCompletetxt1);
            autoCompletar1.Threshold = 1;

            //cria uma instância do controle Button
            btnEnviar = FindViewById<Button>(Resource.Id.btnEnviar);

            //cria um Adapter contendo o array de strings e o modo de exibição
            ArrayAdapter adapter = new ArrayAdapter<string>(this, Android.Resource.Layout.SimpleListItem1, estados);

            //atribui o adapter à propriedade Adapter do controle
            autoCompletar1.Adapter = adapter;
        }
    }
}
```

```

//tratamento do evento Click do Button
btnEnviar.Click += delegate
{
    if (autoCompletar1.Text != "")
    { //exibe o nome do estado selecionado
        Toast.MakeText(this, " Estado = " + autoCompletar1.Text, ToastLength.Short).Show();
    }
    else
    {
        Toast.MakeText(this, " Informe o nome do Estado ", ToastLength.Short).Show();
    }
};
}
}
}

```

O código já está comentado mas vale a pena destacar que :

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.

```
ArrayAdapter adapter = new ArrayAdapter<string>(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.


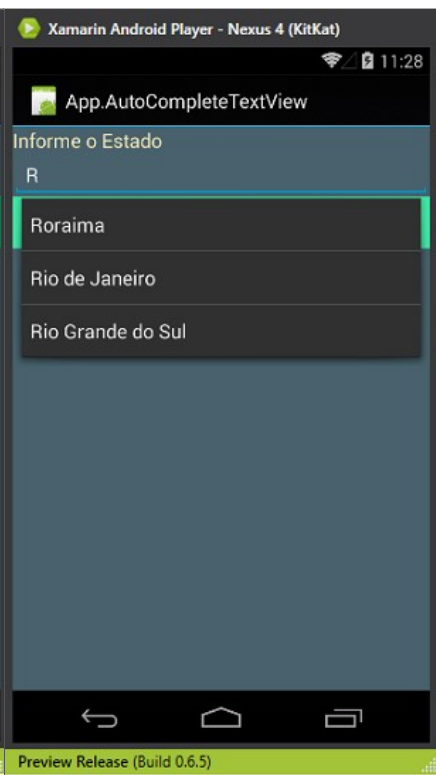

Estamos usando : **SimpleListItem1**

- 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 no **AutoCompleteTextView** e exibe os dados na view:

```
autoCompletar1.Adapter = adapter;
```

Executando o código acima teremos o seguinte resultado :

		
<p>A tela inicial do dispositivo apresentando o controle.</p>	<p>Ao digitar um caractere são apresentadas as sugestões no autocompletar exibidas em um lista suspensa</p>	<p>Ao selecionar um item da lista de sugestão e clicar no botão Enviar temos a mensagem exibindo o nome selecionado.</p>

Podemos customizar a exibição no controle definindo um arquivo XML na pasta **Resources/layout** e usá-lo como leiaute para exibição. Como exemplo criamos o arquivo **Estados.xml** com código abaixo:

```
<?xml version="1.0" encoding="utf-8" ?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    android:textSize="16sp"
    android:textColor="#000">
</TextView>
```

A seguir basta atribuir este leiaute na definição do **ArrayAdapter**:

```
ArrayAdapter adapter = new ArrayAdapter<string>(this, Android.Resource.Layout.Estados, estados);
```

Pegue o projeto completo aqui : [App.AutoCompleteTextView.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

#### 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.App.AlertDialog/>
- [Xamarin Android - Tratando eventos de forma declarativa](#)

---

[José Carlos Macoratti](#)