



Neste artigo vou mostrar como usar os controles **SearchView** e **ListView** para realizar uma busca dinâmica em uma lista de itens 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

Eu já apresentei a [view/widget/controle](#) **ListView** no Xamarin Android mostrando como usar alguns de seus recursos; e, hoje vou mostrar como usar um controle **SearchView** que atua junto com um **ListView** para realizar consultas dinâmicas nos itens da lista.

O controle **SearchView** é um **widget** que fornece uma interface de usuário para o usuário digitar uma consulta de pesquisa e enviar uma solicitação para um provedor de pesquisa. Ele mostra uma lista de sugestões de consulta ou de resultados, se disponível, e, permite ao usuário escolher uma sugestão ou resultado.

No exemplo deste artigo vou usar o evento **QueryTextChange** que é chamado quando o texto da consulta do **SearchView** é alterado pelo usuário, e, vou usar o parâmetro **NewText** que é o novo conteúdo da consulta do campo do controle.

O retorno é um booleano que pode ser **false** se o **SearchView** deve executar a ação padrão de mostrar a sugestão, se disponível, ou **true** se a ação foi tratada pelo ouvinte.

No tratamento do evento vamos usar um Filtro no **Adapter** usando **InvokeFilter** que inicia uma operação assíncrona de filtragem.

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 VS Community 2015

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

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

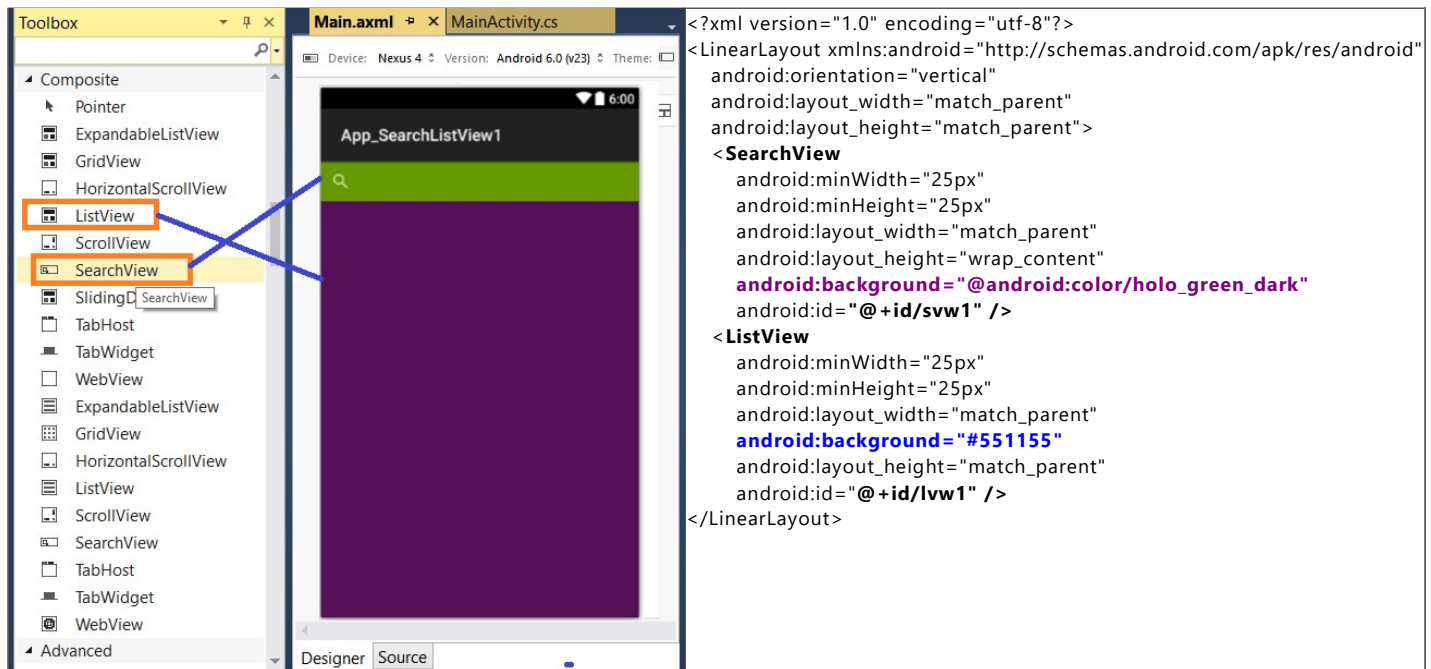
Informe o nome **App_SearchListView1** e clique no botão **OK**;

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

Primeiro remova o controle **Button** que vem definido por padrão e a seguir inclua o seguinte controle a partir da **ToolBox**:

- **1 SearchView - id = svw1**
- **1 ListView - id = lvw1**

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



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <SearchView
        android:minWidth="25px"
        android:minHeight="25px"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@android:color/holo_green_dark"
        android:id="@+id/svw1" />
    <ListView
        android:minWidth="25px"
        android:minHeight="25px"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#551155"
        android:id="@+id/lvw1" />
</LinearLayout>
```

Definimos apenas um controle **SearchView** e um controle **ListView** usando XML.

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

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

namespace App_SearchListView1
{
    [Activity(Label = "App_SearchListView1", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        private SearchView sv1;
        private ListView lv1;
        private ArrayAdapter adp1;
        private ArrayList lista;

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

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

            sv1 = FindViewById<SearchView>(Resource.Id.svw1);
            lv1 = FindViewById<ListView>(Resource.Id.lvw1);

            AdicionarDados();

            adp1 = new ArrayAdapter<this, Android.Resource.Layout.SimpleListItem1, lista>;
            lv1.Adapter = adp1;

            sv1.QueryTextChange += Sv1_QueryTextChange;
            lv1.ItemClick += Lv1_ItemClick;
        }

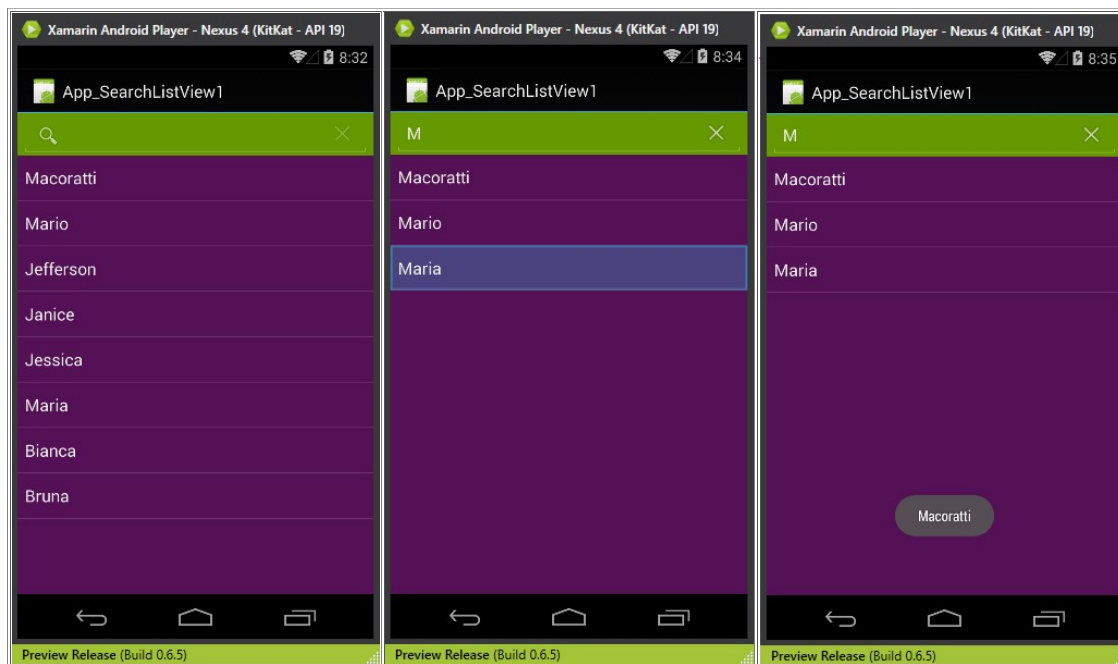
        private void Lv1_ItemClick(object sender, AdapterView.ItemClickEventArgs e)
        {
            Toast.MakeText(this, adp1.GetItem(e.Position).ToString(), ToastLength.Short).Show();
        }

        private void Sv1_QueryTextChange(object sender, SearchView.QueryTextChangeEventArgs e)
        {
            adp1.Filter.InvokeFilter(e.NewText);
        }

        private void AdicionarDados()
        {
            lista = new ArrayList();
            lista.Add("Macoratti");
            lista.Add("Mario");
            lista.Add("Jefferson");
            lista.Add("Janice");
            lista.Add("Jessica");
            lista.Add("Maria");
            lista.Add("Bianca");
            lista.Add("Bruna");
        }
    }
}
```

Neste código criamos uma instância dos controles **SearchView**, **ListView**, **ArrayAdapter** e **ArrayList** e a seguir definimos os eventos **QueryTextChange** do **SearchView** onde iremos realizar o filtro dinâmico **ItemClick** do **ListView** onde iremos exibir o texto do item clicado pelo usuário usando a classe **Toast** que permite exibir mensagens ao usuário de maneira simples e rápida em uma janela usando uma pequena view.

Executando o projeto usando o emulador do **Xamarin Android Player** e emulando o **KitKat(API 19)** iremos obter o seguinte resultado:



Ao digitar um texto no campo do **SearchView** o evento **QueryTextChanged** apresenta as sugestões dos itens que atendem o critério exibindo-os no ListView.

Pegue o projeto aqui : [App_SearchListView1.zip](#) (sem as referências)

Jesus lhes respondeu, e disse: A minha doutrina não é minha, mas daquele que me enviou.

Se alguém quiser fazer a vontade dele, pela mesma doutrina conhecerá se ela é de Deus, ou se eu falo de mim mesmo.

João 7:16-17

[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 - 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](#)
- [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](#)
- [Xamarin Android - Busca dinâmica com SearchView e ListView ...](#)
- [Seção Mobile/Xamarin do site Macoratti .net](#)

[José Carlos Macoratti](#)