

## Macoratti.net Xamarin Android - Exibindo um arquivo de vídeo



Neste artigo vou mostrar como podemos executar arquivos de vídeo em uma aplicação Xamarin Android usando o [VS 2015 Community](#) e a linguagem C#.

Curso C# Vídeo Aulas  
Do básico ao intermediário

Por um preço justo

O sistema operacional Android fornece amplo suporte para multimídia, englobando tanto o áudio como o vídeo. Para tratar com vídeo temos as classe **VideoView** e **MediaRecord** com recursos de gravação e playback de arquivos de vídeos.

Existe um sem número de ocasiões na qual você precisa ou deseja que sua aplicação exiba vídeos.

Neste artigo eu vou mostrar como executar um arquivo de vídeo quando o usuário informa uma uri e clica em um botão de comando em uma aplicação Xamarin Android.

Para isso vamos usar a classe **VideoView** do namespace **Android.Media** que executa arquivos de [vídeo](#) e [stream](#).

Como vamos ter que ter acesso a internet vamos requisitar a permissão para acessar a **INTERNET** no arquivo **AndroidManifest.xml**.

Então vamos ao trabalho...

### Recursos usados:

- [Visual Studio Community 2015](#)
- [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**.

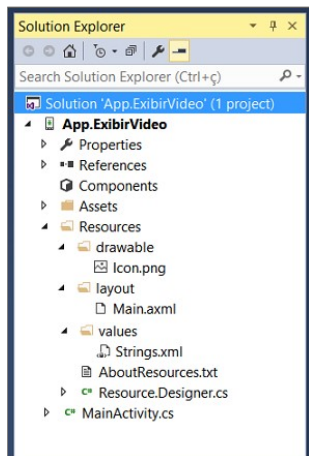
### Criando o projeto no VS Community com Xamarin

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

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

Informe o nome **App.ExibirVideo** 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.

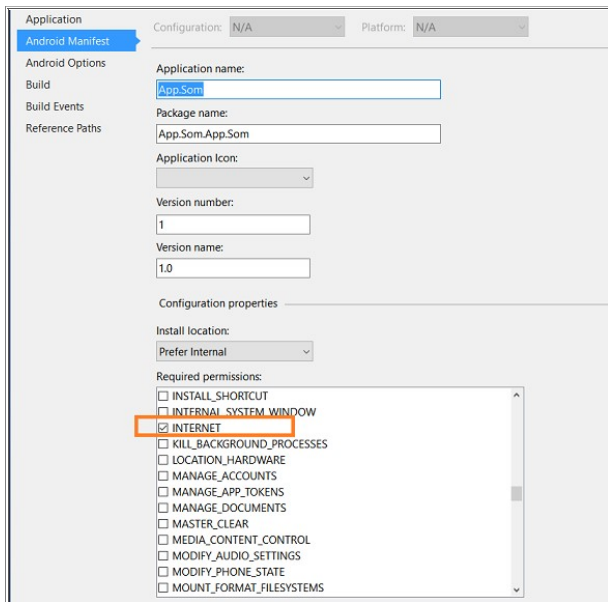
De forma geral, todas as imagens, ícones, sons e outros arquivos externos são colocados na pasta **Resources** enquanto que dicionários e arquivos XML são postos na pasta **Assets**;

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;

Vamos agora acessar o arquivo **AndroidManifest.xml** e requisitar a permissão para acessar a internet.

Abra a janela de propriedades do projeto e marque a permissão **INTERNET**:



**Nota:** Se você desejar gravar áudio/vídeo também deverá dar acesso para realizar essa tarefa.

A seguir localize o arquivo **Main.axml** na pasta **Resources\layout** e inclua a partir da **ToolBox** os seguintes controles:

- 1 **EditText** - **txtVideo**
- 1 **Button** - **btnExibir**
- 1 **VideoView** - **videoView**

```
<?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">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="http://video/animation-hd.mp4"
        android:id="@+id/txtVideo" />
    <Button
        android:text="Exibir Vídeo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btnExibir" />
    <VideoView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/videoView" />
</LinearLayout>
```

O arquivo **MainActivity.cs**, como o nome já sugere, é onde está definida a **Activity** da aplicação Android.

Vamos agora abrir o arquivo **MainActivity.cs** e incluir o seguinte código:

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

namespace App.ExibirVideo
{
    [Activity(Label = "App.ExibirVideo", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        VideoView video;

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

            SetContentView(Resource.Layout.Main);
        }
    }
}
```

```

Button button = FindViewById<Button>(Resource.Id.btnExibir);
video = FindViewById<VideoView>(Resource.Id.videoView);
EditText txtVideo = FindViewById<EditText>(Resource.Id.txtVideo);

var uri = Android.Net.Uri.Parse(txtVideo.Text);

button.Click += delegate
{
    if (uri != null)
    {
        video.SetVideoURI(uri);
        video.Start();
    }
    else
    {
        Toast.MakeText(this, "Informe a URI do vídeo", ToastLength.Short).Show();
    }
};
}
}

```

O código é bem simples:

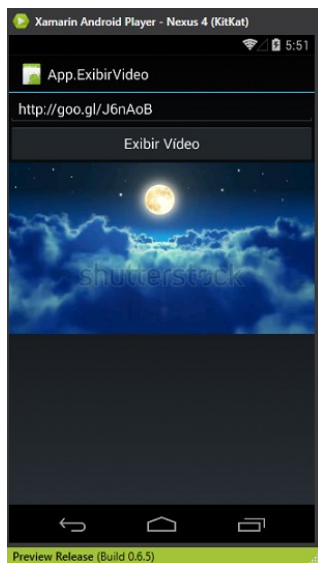
- criamos instâncias dos componentes definidos no arquivo de layout;
- definimos para o objeto Button um evento Click onde usamos os métodos adequados da classe VideoView;

Primeiro criamos usamos o método **Android.Net.Uri.Parse** que cria um Uri que analisa a string URI codificada.

Depois usamos o método **SetVideoUri()** para passar a URI para a classe VideoView.

E depois iniciamos a execução do vídeo com o método **Start()**.

Executando o projeto iremos obter o seguinte resultado:



Aguarde mais artigos sobre o desenvolvimento de aplicativos Android usando o Visual Studio e o Xamarin.

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

**(Disse Jesus aos fariseus)** Hipócritas, bem profetizou Isaías a vosso respeito, dizendo:

**Este povo se aproxima de mim com a sua boca e me honra com os seus lábios, mas o seu coração está longe de mim.**

**Mas, em vão me adoram, ensinando doutrinas que são preceitos dos homens.**

**Mateus 15:7-9**

[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 - Video 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 video 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](#)

---

[José Carlos Macoratti](#)