

## Macoratti.net Xamarin Android - Tocando um Som mp3



Neste artigo vou mostrar como podemos executar um arquivo de som **mp3** em uma aplicação Xamarin Android.

**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.

Existe um sem número de ocasiões na qual você precisa ou deseja que sua aplicação emita sons, quer seja um simples bip até uma música mais elaborada.

Neste artigo eu vou mostrar como executar um arquivo **mp3** quando o usuário clica em um botão de comando em uma aplicação Xamarin Android.

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

Vou mostrar como executar arquivos locais e arquivos remotos, e, neste caso, teremos que 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](#))

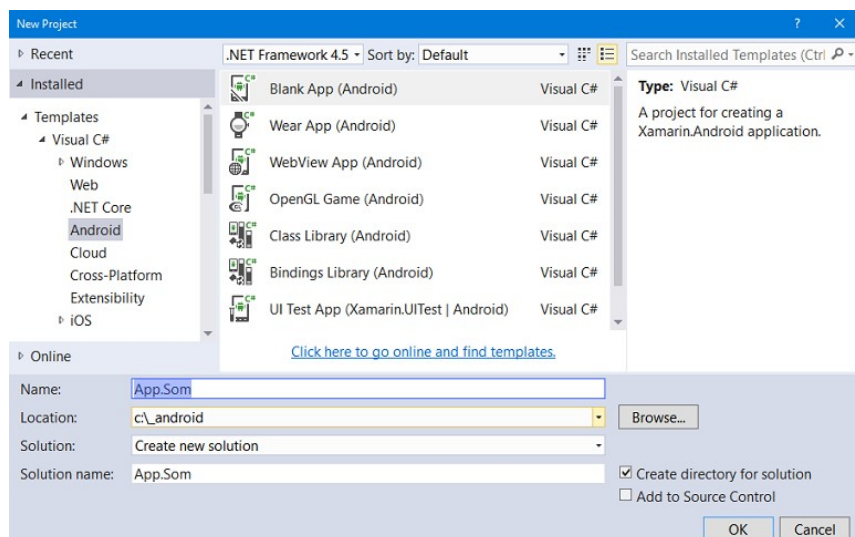
**Nota:** Baixe e use a versão **Community 2015** do VS ela é grátis e é equivalente a versão **Professional**.

### 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.Som** e clique no botão OK:

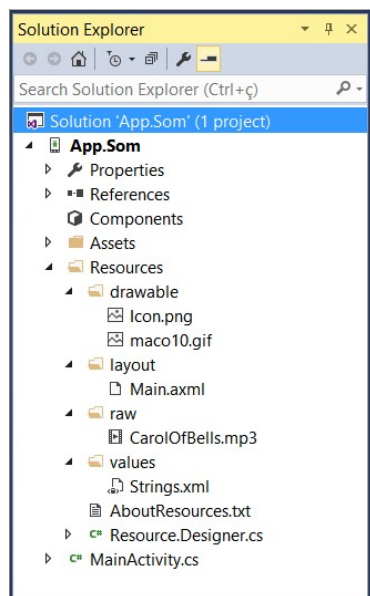


Será criada uma solução com a seguinte estrutura:

- **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**;

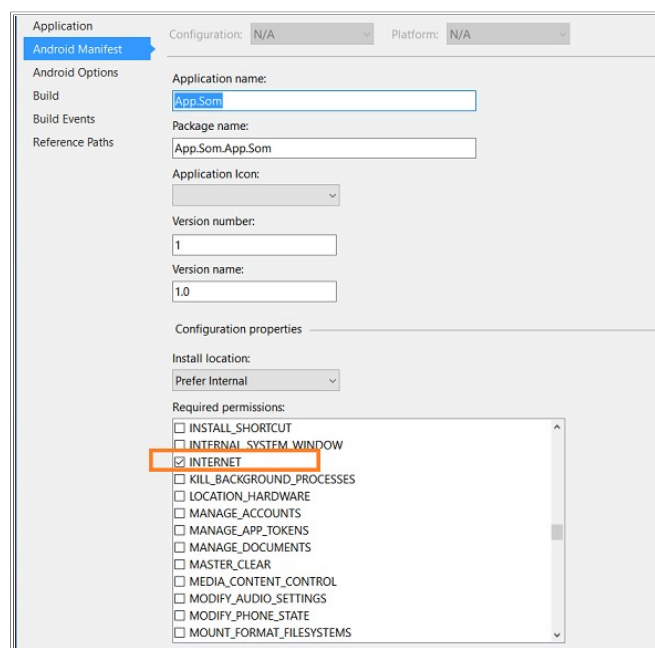
Na subpasta **layout** temos os arquivos **.xml** que definem as **views** usadas no projeto;

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

**Nota** : A pasta **Drawable** contém recursos como imagens png, jpg, etc., usadas no aplicativo. Ela contém múltiplas pastas específicas para cada resolução possível em uma aplicação Android. Numa aplicação típica Android você vai acabar encontrando as pastas: **Drawable-LDPI**, **Drawable-mdpi**, **Drawable-hdpi**, **Drawable-xhdpi**, **Drawable-xxhdpi**, etc.

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 deseja gravar áudio também deverá dar acesso para realizar essa tarefa.

Agora, inclua na pasta **Resources/drawable** um arquivo de imagem que deverá ser exibido na view **ImageView**. No exemplo eu estou usando o arquivo **maco10.gif**.

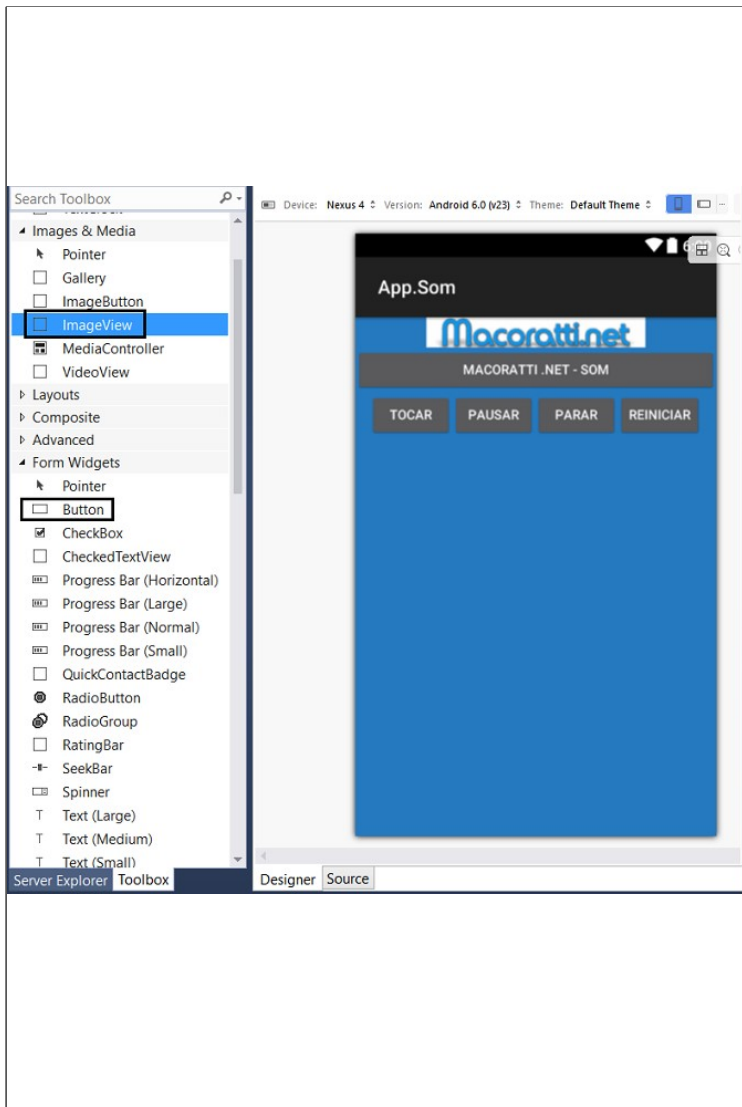
Crie a pasta **raw** no interior da pasta **Resources** e nesta pasta inclua o arquivo **mp3** que deseja executar. No exemplo estou usando o arquivo **CarolOfBells.mp3**.

A pasta **raw** é uma pasta especial que deve ser criada sempre dentro da pasta **Resources** e que é utilizada para salvar arquivos na sua forma natural.

Para saber mais veja a documentação do Android : <https://developer.android.com/guide/topics/resources/providing-resources.html>

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

- 1 **LinearLayout (vertical)**
- 1 **ImageView**
- 1 **Button - MyButton**
- 1 **LinearLayout (horizontal), gravity = center**
- 4 **Buttons - btnTocar, btnPara, btnPausar e btnReiniciar**



```
<?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"
    android:background="#2579BF">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="0dp"
        android:src="@drawable/maco10" />
    <Button
        android:id="@+id/MyButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/Hello" />
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout1"
        android:gravity="center">
        <Button
            android:text="Tocar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/btnTocar" />
        <Button
            android:text="Pausar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/btnPausar" />
        <Button
            android:text="Parar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/btnParar" />
        <Button
            android:text="Reiniciar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/btnReiniciar" />
    </LinearLayout>
</LinearLayout>
```

Ao lado da figura vemos o código XML gerado.

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

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

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

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

        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);
            SetContentView(Resource.Layout.Main);

            //criando a instância do media player para o recurso que desejamos tocar
            _myPlayer = MediaPlayer.Create(this, Resource.Raw.CarolOfBells);

            Button button = FindViewById<Button>(Resource.Id.MyButton);
            button.SetBackgroundColor(Android.Graphics.Color.Red);

            Button tocar = FindViewById<Button>(Resource.Id.btnTocar);
            Button parar = FindViewById<Button>(Resource.Id.btnParar);
            Button pausar = FindViewById<Button>(Resource.Id.btnPausar);
            Button reiniciar = FindViewById<Button>(Resource.Id.btnReiniciar);
```

```

        button.Click += delegate
        {
            _myPlayer.Start();
        };
        tocar.Click += Tocar_Click;
        parar.Click += Parar_Click;
        pausar.Click += Pausar_Click;
        reiniciar.Click += Reiniciar_Click;
    }

    private void Reiniciar_Click(object sender, System.EventArgs e)
    {
        _myPlayer.Start();
    }

    private void Pausar_Click(object sender, System.EventArgs e)
    {
        _myPlayer.Pause();
    }

    private void Parar_Click(object sender, System.EventArgs e)
    {
        _myPlayer.Stop();
    }

    private void Tocar_Click(object sender, System.EventArgs e)
    {
        _myPlayer.Start();
    }
}
}

```

O código é bem simples:

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

Executando o projeto iremos obter o seguinte resultado:



Para concluir vamos implementar agora uma novo botão em nossa interface e criar um evento **Click** que permita ao usuário executar um arquivo **mp3** que não é local.

Neste cenário o arquivo esta em um servidor na internet.

Abra o arquivo **Main.axml** no modo **Designer** e a partir da **ToolBox** inclua um novo **Button** com o id igual a **btnOnline**, abaixo dos demais controles conforme mostra a figura a seguir:



Ao lado vemos o código XML gerado.

A seguir basta incluir as duas linhas de código abaixo para criar uma instância deste novo Button e definir um evento **Click** ao button no método **OnCreate()** do arquivo **MainActivity.cs**:

```
Button online = FindViewById<Button>(Resource.Id.btnOnLine);
online.Click += Online_Click;
```

Após isso defina o seguinte código no evento **Click** do button :

```
private void Online_Click(object sender, System.EventArgs e)
{
    string url = "http://macoratti.net/testeaudio/CarolOfBells.mp3";
    _myPlayer = new MediaPlayer();
    _myPlayer.SetAudioStreamType(Stream.Music);
    _myPlayer.SetDataSource(url);
    _myPlayer.Prepare();
    _myPlayer.Start();
}
```

A primeira linha define a URI para o arquivo de áudio a ser executado.

Depois criamos uma nova instância de **MediaPlayer()**, definimos o tipo de stream e passamos a url via **SetDataSource()**.

O método **Prepare()** prepara o player para ser executado de forma síncrona, o que significa, que esta instrução bloqueia o programa até que o player esteja pronto para executar. (Poderíamos ter usando o método **PrepareAsync()** que executa a tarefa de forma assíncrona).

Executando o projeto novamente teremos o resultado a seguir:



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

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

**Porque a palavra da cruz é loucura para os que perecem; mas para nós, que somos salvos, é o poder de Deus.**  
**1 Coríntios 1:18**

[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 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 \(@macorati\) | 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](#)