



Neste artigo vou mostrar como usar a câmera do dispositivo em aplicações **Xamarin Android** usando o Visual Studio 2015 e a linguagem C#.

**Curso C# Vídeo Aulas**  
Do básico ao intermediário

Por um preço justo

Existem diversas abordagens que podemos adotar para usar a câmera do dispositivo no Xamarin Android, e, neste artigo, eu vou usar o provider **MediaStore**.

O **provedor (provider)** de mídia - **MediaStore** - contém metadados para todas as mídias disponíveis em dispositivos de armazenamento interno e externo.

**Nota:** O recurso **Provider** do Android fornece classes para acessar os provedores de conteúdo fornecidos pelo Android. O Android é fornecido com vários provedores de conteúdo que armazenam dados comuns, como informações de contatos, informações de calendário e arquivos de mídia. Essas classes fornecem métodos simplificados para adicionar ou recuperar dados desses provedores de conteúdo.

Podemos usar o campo **ActionImageCapture** que é uma **Intent** padrão que pode ser enviada para que a câmera do dispositivo capture uma imagem e a retorne.

```
Intent intent = new Intent(MediaStore.ActionImageCapture);  
StartActivityForResult(intent, 0);
```

A seguir usamos o método **StartActivityForResult** para chamar a Intent com a expectativa de obter algum resultado de volta; no caso a imagem capturada pela câmera.

Também definimos o método **OnActivityResult** que é chamado quando uma atividade que foi lançada termina, dando-nos o **requestCode** que usamos, o **resultCode** retornado e dados adicionais (**data**).

```
protected override void OnActivityResult(int requestCode, [GeneratedEnum] Result resultCode, Intent data)  
{  
    base.OnActivityResult(requestCode, resultCode, data);  
    global::Android.Graphics.Bitmap bitmap = (Bitmap)data.Extras.Get("data");  
    imageView1.SetImageBitmap(bitmap);  
}
```

Observe que tivemos que usar o atributo **global::Android.Graphics.Bitmap** para usar o tipo **Bitmap** e contornar o não reconhecimento do namespace : **using Android.Graphics;**

Vamos então mostrar como implementar o recurso no Xamarin Android.

**Recursos usados:**

- [Visual Studio Community 2015](#) ou **Xamarin Studio**
- [Xamarin](#)

**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)**

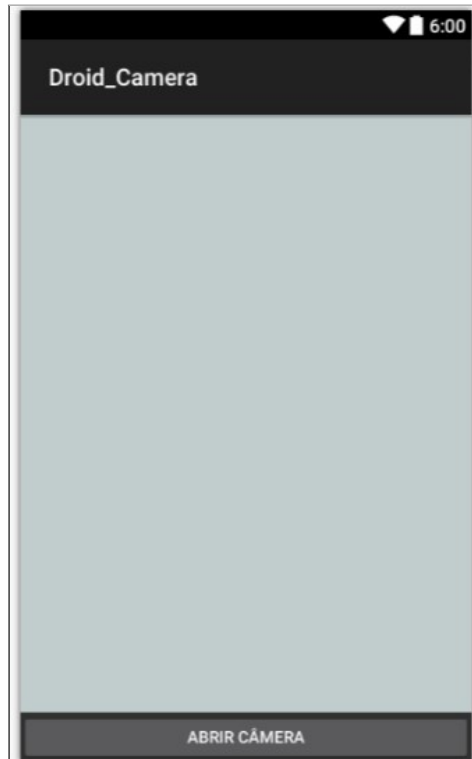
Informe o nome **Droid\_Camera** e clique no botão **OK**;

Abra o arquivo **Main.axml** na pasta **Resources/layout** e no modo **Designer** e a seguir inclua a partir da ToolBox os seguintes controles:

- **1 ImageView - imgbw1**
- **1 Button - btnCamera**

Ao inserir o **RadioGroup** já teremos o controle contendo 3 RadioButtons. Acrescentamos assim mais dois RadioButtons a partir da ToolBox no RadioGroup para ficar com 5 RadioButtons.

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

	<pre>&lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"     android:orientation="vertical"     android:layout_width="match_parent"     android:layout_height="match_parent"     android:weightSum="10"&gt;     &lt;ImageView         android:layout_width="match_parent"         android:layout_height="fill_parent"         android:background="#c1cdcd"         android:layout_weight="9"         android:id="@+id/imgvw1" /&gt;     &lt;Button         android:text="Abrir Câmera"         android:layout_weight="1"         android:layout_width="match_parent"         android:layout_height="wrap_content"         android:id="@+id/btnCamera" /&gt; &lt;/LinearLayout&gt;</pre>
--	--

Observe que usamos **android:weightSum** no layout em conjunto com **android:weight** em cada widget filha (*ImageView* e *Button*) para obter o layout exibido. (Veja abaixo o que diz a documentação)

#### Android: weightSum

Define a soma máxima do peso. Se não especificado, a soma é calculada adicionando o **layout\_weight** de todas as views filhas. Isso pode ser usado, por exemplo, para dar a uma única view filha 50% do espaço disponível total, dando-lhe um **layout\_weight** de 0,5 e definindo o **weightSum** para 1,0.

Agora vamos definir o código no arquivo **MainActivity.cs** vinculado a nossa view **Main.xml**.

## Tratando a seleção nos RadioButtons

Abra o arquivo **MainActivity.cs** e altere o código desse arquivo conforme abaixo:

```
using Android.App;
using Android.Widget;
using Android.OS;
using Android.Content;
using Android.Provider;
using Android.Runtime;
using Android.Graphics;

namespace Droid_Camera
{
```

```
[Activity(Label = "Droid_Camera", MainLauncher = true, Icon = "@drawable/icon")]
public class MainActivity : Activity
{
    ImageView imgView1;
    Button btnCamera;

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

        // Set our view from the "main" layout resource
        SetContentView (Resource.Layout.Main);
        btnCamera = FindViewById<Button>(Resource.Id.btnCamera);
        imgView1 = FindViewById<ImageView>(Resource.Id.imgvw1);

        btnCamera.Click += BtnCamera_Click;
    }

    protected override void OnActivityResult(int requestCode, [GeneratedEnum] Result resultCode, Intent data)
    {
        base.OnActivityResult(requestCode, resultCode, data);
        global::Android.Graphics.Bitmap bitmap = (Bitmap)data.Extras.Get("data");
        imgView1.SetImageBitmap(bitmap);
    }

    private void BtnCamera_Click(object sender, System.EventArgs e)
    {
        Intent intent = new Intent(MediaStore.ActionImageCapture);
        StartActivityForResult(intent, 0);
    }
}
```

Vamos entender o código :

1- Definimos uma variável do tipo Button que vai receber a instância do **Button** definido no Layout e uma variável do tipo **ImageView** para receber a instância do ImageView usado:

```
ImageView imgView1;
Button btnCamera;
```

2- Fizemos a vinculação da nossa Activity com o arquivo de layout **Main** e criamos a instância do botão e do ImageView usados no Layout e definimos o evento **Click** do botão :

```
SetContentView (Resource.Layout.Main);
btnCamera = FindViewById<Button>(Resource.Id.btnCamera);
imgView1 = FindViewById<ImageView>(Resource.Id.imgvw1);

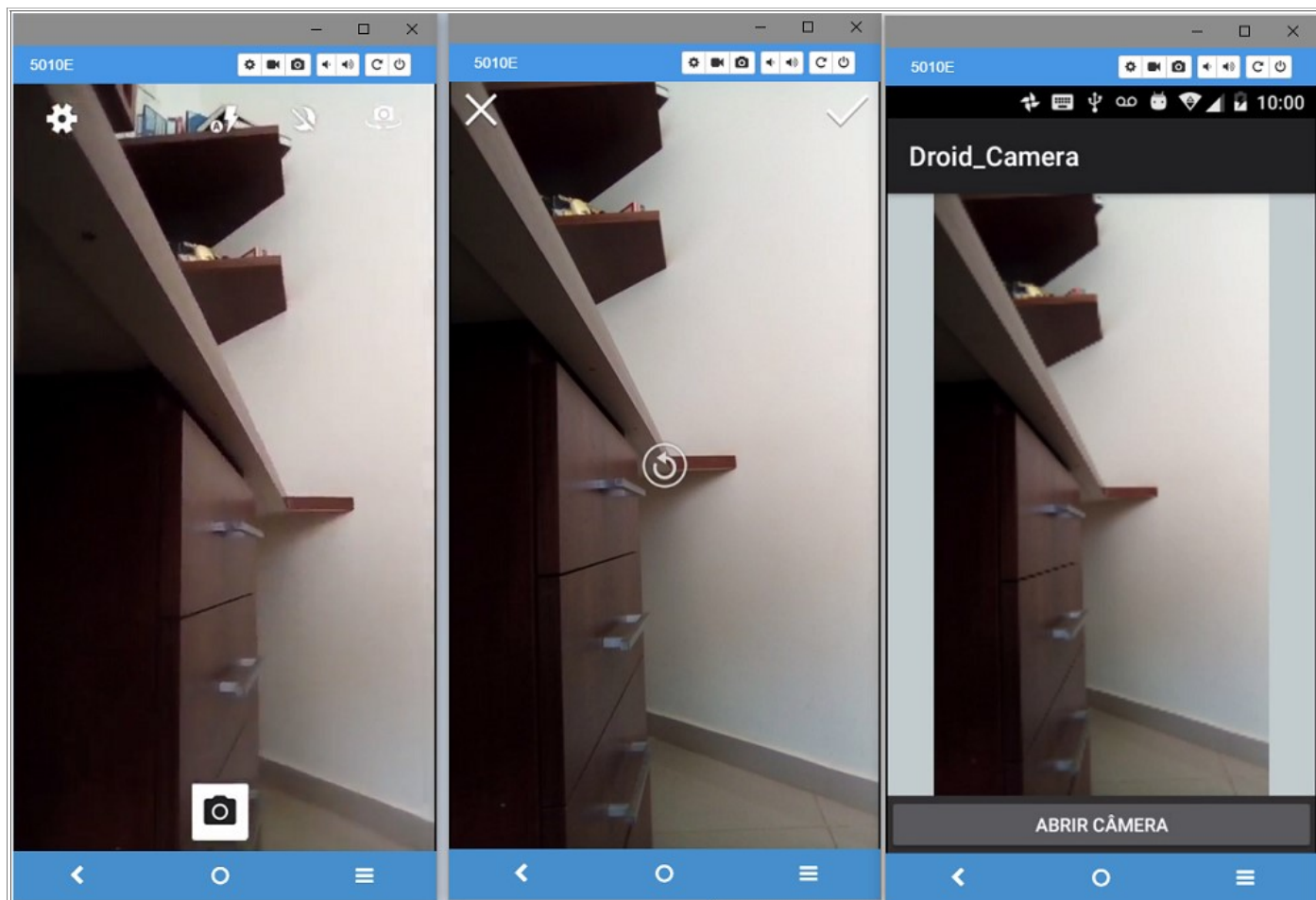
btnCamera.Click += BtnCamera_Click;
```

3- No evento **Click** do botão : *(já foi explicado no início do artigo)*

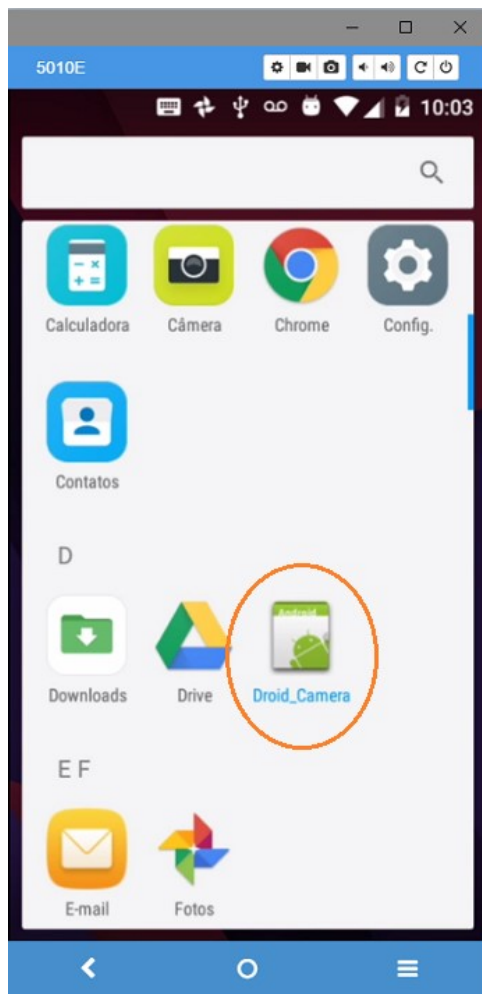
Usamos o método **ActionImageCapture** do provedor **MediaStore** para capturar a imagem usando a câmera do dispositivo e exibindo a imagem na widget **ImageView**.

Executando o projeto e fazendo a emulação no dispositivo físico usando o **Vysor** iremos obter o seguinte resultado:

**Nota:** Para saber como usar o seu dispositivo físico usando o Vysor leia este artigo : [Veja como emular usando o Vysor - Macoratti](#)



**Obs:** Eu estou usando um Alcatel PIXI 4 (lembre-se de instalar o driver adb para fazer a ponte entre o Windows e o Android). Nesta abordagem o deploy do seu aplicativo android será feito no dispositivo conforme mostra a figura abaixo:



Pegue o projeto aqui : [Droid Camera.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. Porque está escrito: Destruirei a sabedoria dos sábios, E aniquilarei a inteligência dos inteligentes.**

**1 Coríntios 1:18,19**

[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](#)
- [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](#)
- [Seção Mobile/Xamarin do site Macoratti .net](#)
- <https://developer.android.com/reference/android/app/Activity.html>
- <https://developer.xamarin.com/api/type/Android.Widget.ProgressBar/>

---

[José Carlos Macoratti](#)