

Macoratti.net Xamarin Android - Apresentando os controles RadioButton e RadioGroup



Neste artigo vou apresentar os conceitos básicos sobre o controle **RadioButton** do **Xamarin Android** usando o Visual Studio 2015 e a linguagem C#.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

Os **radio buttons** ou *botões de rádio* permitem ao usuário selecionar uma opção de um conjunto de opções. Este controle apresenta dois estados que pode estar marcado ou desmarcado. (*checked/unchecked*)

Quando o botão de opção estiver desmarcado, o usuário pode pressionar ou clicar nele para marcá-lo. No entanto, ao contrário de um *CheckBox*, um radio button não pode ser desmarcado pelo usuário uma vez marcado.

Os radio buttons em geral são usados em conjunto com um **RadioGroup**. Assim, quando vários botões de rádio estiverem no interior de **RadioGroup** se você marcar um radio button todos os demais serão desmarcados.

Você deve usar os botões de opção para conjuntos opcionais que são mutuamente exclusivos se você acha que o usuário precisa ver todas as opções disponíveis lado a lado. Se não for necessário mostrar todas as opções lado a lado, use um controle **Spinner**.

Para utilizar mais de um radio button em seu layout você deve agrupar os controles no interior de um **RadioGroup** pois eles são mutualmente exclusivos. Ao agrupá-los, o sistema garante que somente um botão de rádio pode ser selecionado de cada vez.

Assim, a classe **RadioGroup** é usada para criar um escopo de exclusão múltipla para um conjunto de radio buttons. A verificação de um botão de rádio que pertence a um **RadioGroup** desmarca qualquer botão de rádio previamente marcado dentro do mesmo grupo.

Neste artigo veremos um exemplo básico de utilização do controle **radio button** em conjunto com o **RadioGroup**.

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)**

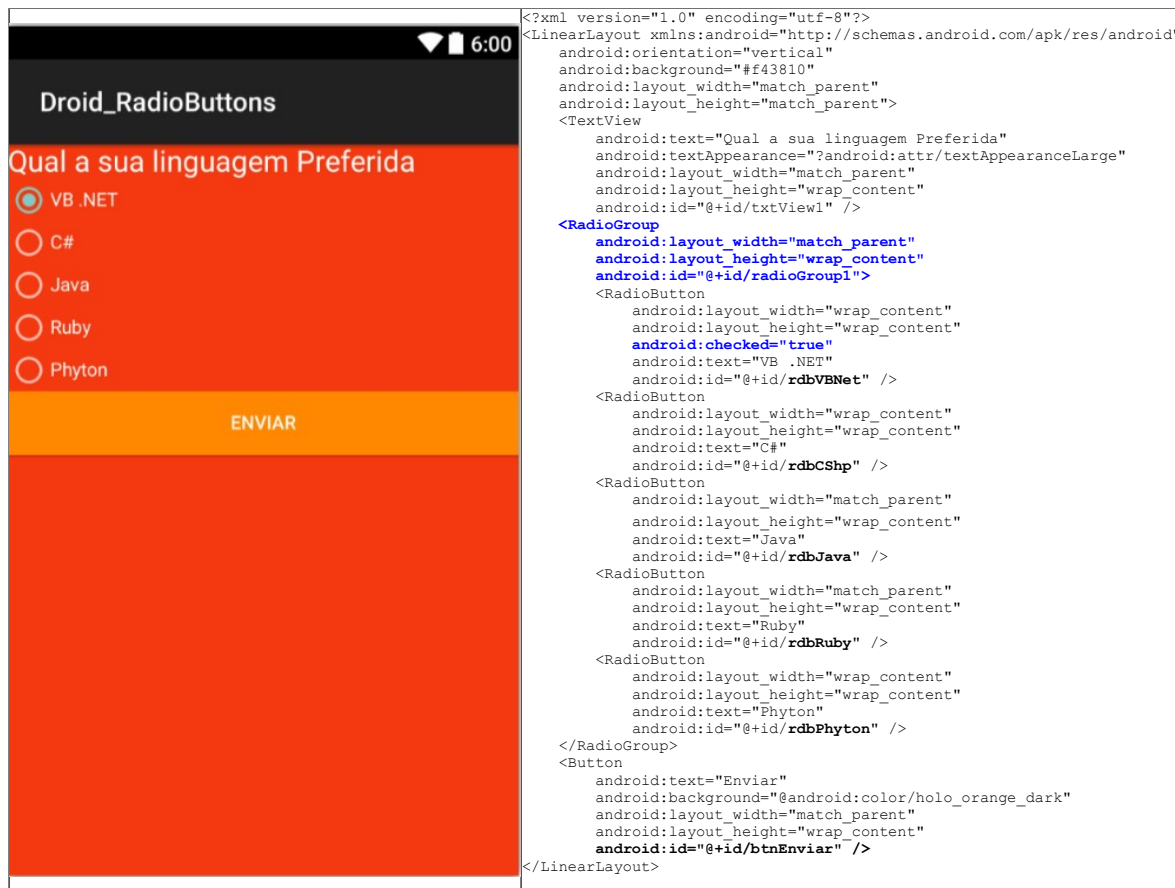
Informe o nome **Droid_RadioButton** e clique no botão **OK**;

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 Text (Large) - txtView1**
- **1 RadioGroup - radrioGroup1 contendo os radiobuttons : rdbVBNet, rdbCShp, rdbJava**
- **2 RadioButtons - rdbRuby e rdbPhyton**
- **1 Button - btnEnviar**

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 :



Observe que o **RadioGroup** contem os **RadioButtons** e que somente um dos controles possui a propriedade **android:checked** marcada como **true**.

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

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;

namespace Droid_RadioButtons
{
    [Activity(Label = "Droid_RadioButtons", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        Button btnEnviar;
        RadioGroup rdgrp1;

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

            // Set our view from the "main" layout resource
            SetContentView (Resource.Layout.Main);
            btnEnviar = FindViewById<Button>(Resource.Id.btnEnviar);
            rdgrp1 = FindViewById<RadioGroup>(Resource.Id.radioGroup1);

            btnEnviar.Click += BtnEnviar_Click;
        }

        private void BtnEnviar_Click(object sender, System.EventArgs e)
        {
            RadioButton checkedRadioButton = FindViewById<RadioButton>(rdgrp1.CheckedRadioButtonId);
            if (checkedRadioButton.Text == "C#")
            {
            }
        }
    }
}
```

```
        Toast.MakeText(this, "Muito bem, C# é mesmo uma ótima linguagem...", ToastLength.Short).Show();
    }
    else
    {
        Toast.MakeText(this, "Esta linguagem também é muito boa.", ToastLength.Short).Show();
    }
}
}
```

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 **RadioGroup** para receber a instância do RadioGroup usado:

```
Button btnEnviar;
RadioGroup rdgrp1;
```

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

```
SetContentView (Resource.Layout.Main);
btnEnviar = FindViewById<Button>(Resource.Id.btnEnviar);
rdgrp1 = FindViewById<RadioGroup>(Resource.Id.radioGroup1);

btnEnviar.Click += BtnEnviar_Click;
```

3- No evento **Click** do botão :

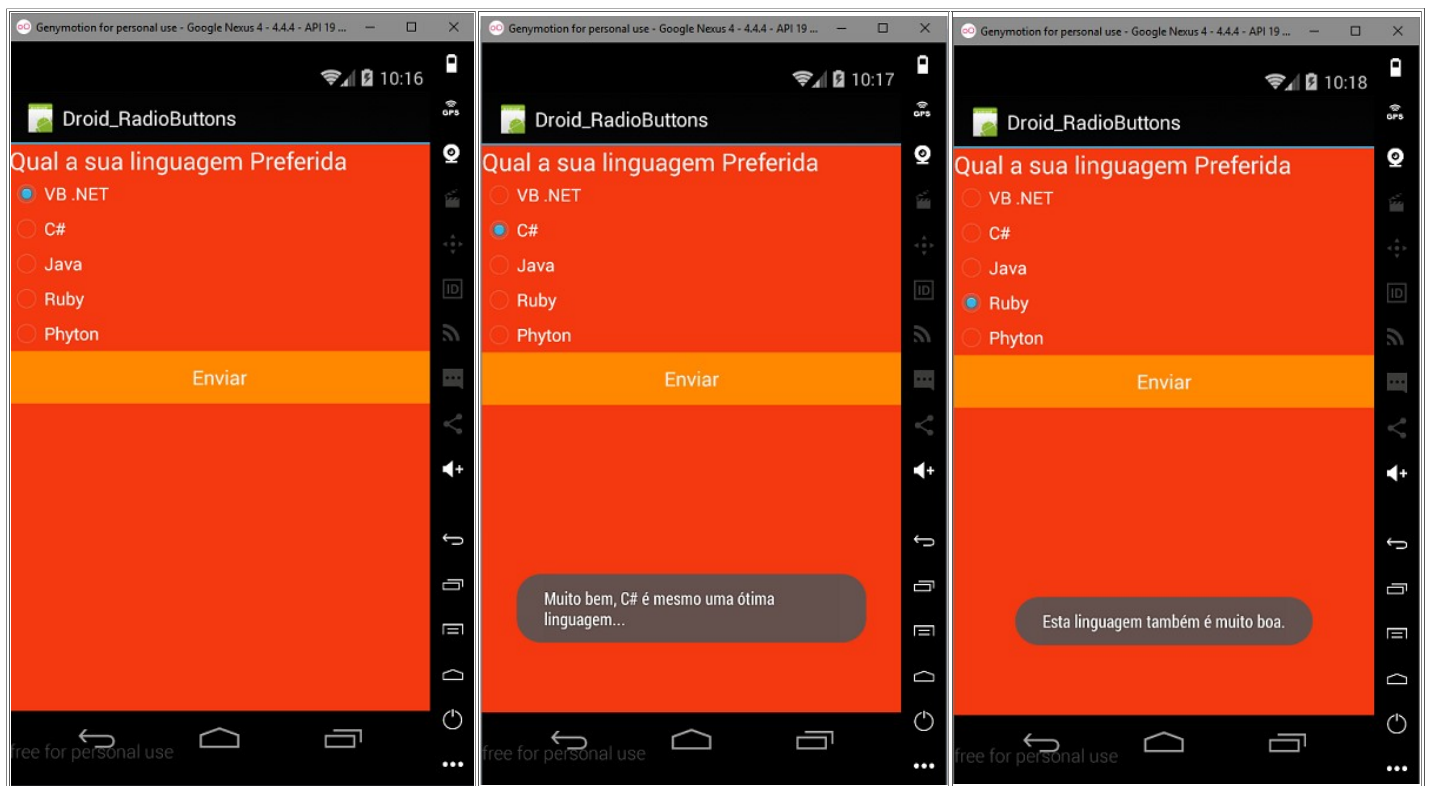
a - Criamos uma instância de um **RadioButton** e utilizamos a propriedade **CheckedRadioButtonId** do RadioGroup - **rdgrp1** - que permite localizar o id do RadioButton que esta marcado.

```
RadioButton checkedRadioButton = FindViewById<RadioButton>(rdgrp1.CheckedRadioButtonId);
```

b - Verificamos se o texto do radiobutton marcado é igual ao texto **"C#"** e emitimos uma mensagem. Caso contrário exibimos outra mensagem:

```
if (checkedRadioButton.Text == "C#")
{
    Toast.MakeText(this, "Muito bem, C# é mesmo uma ótima linguagem...", ToastLength.Short);
}
else
{
    Toast.MakeText(this, "Esta linguagem também é muito boa.", ToastLength.Short);
}
```

Executando o projeto usando o emulador **Genymotion** iremos obter o seguinte resultado:



Podemos usar outra abordagem para verificar se um RadioButton esta marcado definindo em cada RadioButton um evento Click.

Então nessa abordagem teremos que alterar o código do arquivo **Main.axml** e vamos incluir o evento **onClick** nos três primeiros RadioButtons:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:background="#f43810"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:text="Qual a sua linguagem Preferida"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/txtView1" />
    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/radioGroup1">
        <RadioButton
            android:onClick="radioButton_OnClick"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="VB .NET"
            android:id="@+id/rdbVBNet" />
        <RadioButton
            android:onClick="radioButton_OnClick"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="C#"
            android:id="@+id/rdbCShp" />
        <RadioButton
            android:onClick="radioButton_OnClick"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Java"
            android:id="@+id/rdbJava" />
        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Ruby"
            android:id="@+id/rdbRuby" />
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Phyton"
            android:id="@+id/rdbPhyton" />
    </RadioGroup>
    <Button
        android:text="Enviar"
        android:background="@android:color/holo_orange_dark"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btnEnviar" />
</LinearLayout>
```

A seguir vamos definir a implementação do evento **onClick** para o **RadioButton** no arquivo **MainActivity.cs** incluindo o código a seguir:

```
[Java.Interop.Export("radioButton_OnClick")]
```

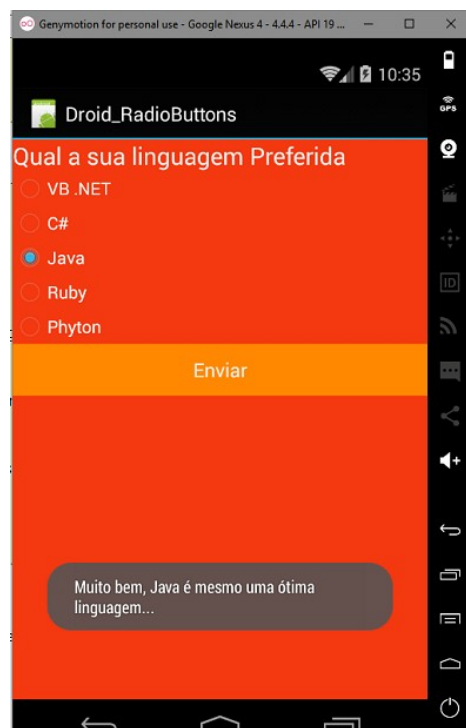
```
public void radioButton_OnClick(View v)
```

```
{
    switch(v.Id)
    {
        case Resource.Id.rdbCSharp:
            Toast.MakeText(this, "Muito bem, C# é mesmo uma ótima linguagem...", ToastLength.Short).Show();
            break;
        case Resource.Id.rdbJava:
            Toast.MakeText(this, "Muito bem, Java é mesmo uma ótima linguagem...", ToastLength.Short).Show();
            break;
        case Resource.Id.rdbVBNet:
            Toast.MakeText(this, "Muito bem, VB .NET é mesmo uma ótima linguagem...", ToastLength.Short).Show();
            break;
    }
}
```

Para que o código funcione temos que incluir uma referência à biblioteca **Mono.Android.Export**.

Nota: O método Export é usado para indicar ao gerador de código Java para exportar um método Java que se torna um Wrapper Callable do Android (ACW).

Executando o projeto novamente e marcando qualquer um dos 3 RadioButtons para os quais definimos o evento **onClick** teremos a respectiva mensagem:



Este é um exemplo bem básico que mostra o uso **RadioButton** e **RadioGroup** no Xamarin Android.

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

(Disse Jesus) "Quem ama a sua vida perdê-la-á, e quem neste mundo odeia a sua vida, guardá-la-á para a vida eterna."

João 12:25

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