

## Macoratti.net Xamarin Android - Tratando eventos de forma declarativa



Neste artigo vou mostrar como podemos realizar o tratamento de eventos em uma aplicação **Xamarin Android** de forma declarativa usando o **VS 2015 Community** e a linguagem **C#**.

Curso C# Vídeo Aulas  
Do básico ao intermediário

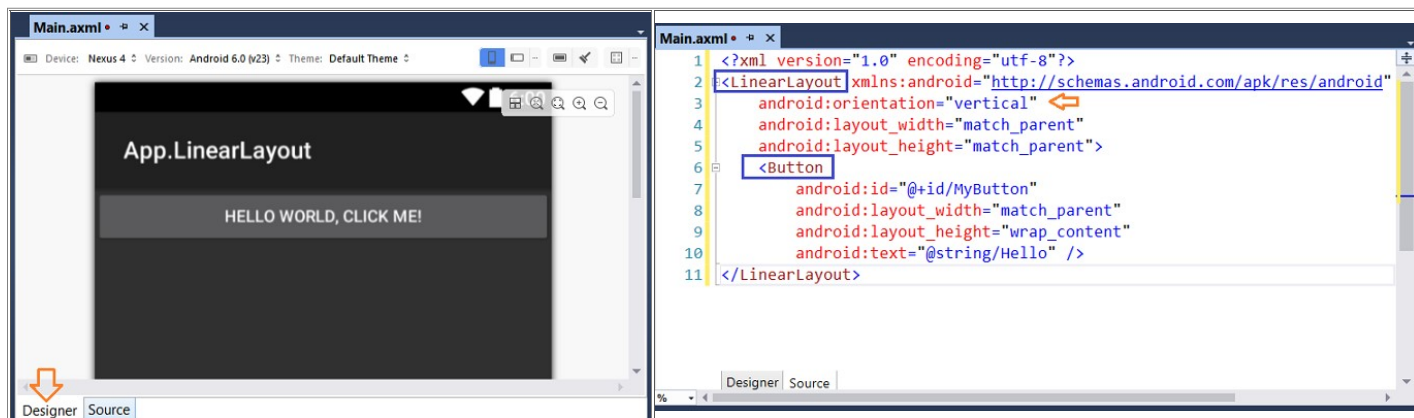
Por um preço justo

Em aplicações **Android** os componentes da interface do usuário (UI) são totalmente isolados da sua **Activity** associada.

Para realizar a vinculação ou para ligar os componentes com a sua atividade precisamos realizar uma abordagem parecida com a adotada pelo JavaScript que é: *primeiro localizar o elemento e então interagir com ele*.

Vamos supor que temos uma view representada por um arquivo **.axml** contendo um **button** com o id **MyButton** (esse cenário ocorre sempre que você cria uma aplicação Android vazia no Visual Studio)

Neste caso teremos o seguinte código **Main.axml**:



Na respectiva **Activity** associada para poder usar o evento **Click** do botão de comando temos o seguinte código:

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

namespace App.PrimeiraApp
{
    [Activity(Label = "App.PrimeiraApp", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        int count = 1;

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

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

            // Get our button from the layout resource,
            // and attach an event to it
            Button button = FindViewById<Button>(Resource.Id.MyButton);

            button.Click += delegate
            {
                button.Text = string.Format("{0} clique!", count++);
            };
        }
    }
}
```

Observe que tivemos que instanciar um objeto do tipo **Button** usando **FindViewById** e depois definimos o evento **Click** para a instância criada.

Mas será que essa é a única abordagem que temos para poder realizar o tratamento de eventos ???

**Não. !!**

Podemos realizar o tratamento de eventos de forma declarativa no arquivo **.axml**.

### Realizando o tratamento de eventos de forma declarativa

Para realizar o tratamento de eventos de forma declarativa no arquivo **.axml** basta seguir o seguinte roteiro:

1- Inclua uma referência à biblioteca **Mono.Android.Export** no seu projeto;

No menu **Project** clique em **Add Reference** e selecione **Framework** e marque a opção **Mono.Android.Export**;

2- Declara o evento que deseja tratar no arquivo **.axml**. Veja o exemplo a seguir :

```
...
<Button
    android:id="@+id/MyButton"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/Hello"
    android:onClick="btn_Evento"/>
...
```

3 - Na respectiva Activity defina o seguinte código :

```
using Android.App;
using Android.OS;
using Android.Views;
using Android.Widget;
using Java.Interop;

namespace App.EventosDeclarativos
{
    [Activity(Label = "App.EventosDeclarativos", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        int count = 1;

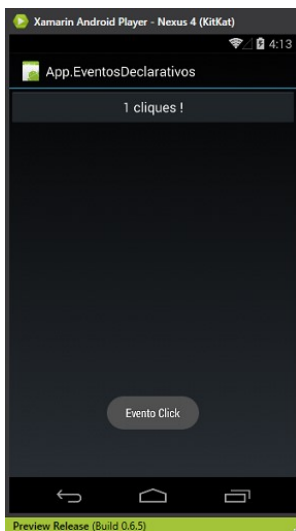
        [Export("btn_Evento")]
        public void button_Click(View v)
        {
            Toast.MakeText(this, "Evento Click", ToastLength.Short).Show();
            (v as Button).Text = string.Format("{0} cliques !", count++);
        }

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

No código usamos o atributo `[Export("nome_evento_no_XML")]` e definimos o código no evento **Click**.

Note que declaramos no namespace **Java.Interop**.

Executando o código acima teremos o mesmo comportamento para o evento **Click** e veremos também o aviso - "Evento Click" - sendo exibido no dispositivo:



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

**Porque Deus enviou o seu Filho ao mundo, não para que condenasse o mundo, mas para que o mundo fosse salvo por ele. Quem crê nele não é condenado; mas quem não crê já está condenado, porquanto não crê no nome do unigênito Filho de Deus. João 3:17,18**

[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 - 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](#)
- [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/>

---

[José Carlos Macoratti](#)