

Macoratti.net Xamarin Android - Criando um menu de opções



Neste artigo vou mostrar como criar um menu de opções em uma aplicação **Android** usando **Visual Studio** com **Xamarin**.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

O Xamarin Android fornece ao usuário a habilidade de exibir um tipo especial de menu que contém um conjunto de itens que diz respeito a toda a aplicação e não somente à **Activity** atual.

Com esse recurso adicionar um menu de opções em nossa aplicação Android se torna uma tarefa simples e somente duas coisas são necessárias :

1. Uma estrutura com as opções do menu
2. O código para vincular o menu com a Activity

Vejamos isso na prática usando o Visual Studio com Xamarin.

Recursos usados:

- [Visual Studio Community 2015](#) ou **Xamarin Studio**
- **Xamarin**
- Emulador Android virtual ou físico ([veja como emular usando o Vysor](#))

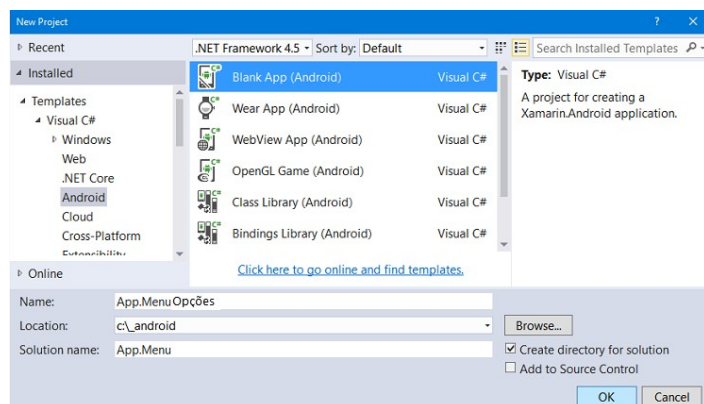
Nota: Baixe e use a versão **Community 2015** do VS ela é grátis e é equivalente a versão **Professional**.

Criando o projeto no Visual Studio 2015 Community

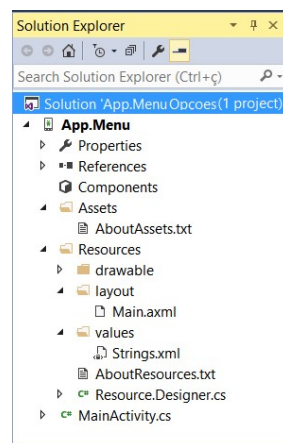
Abra o **VS 2015 Community** e clique em **New Project**;

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

Informe o nome **App.MenuOpcoes** 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;

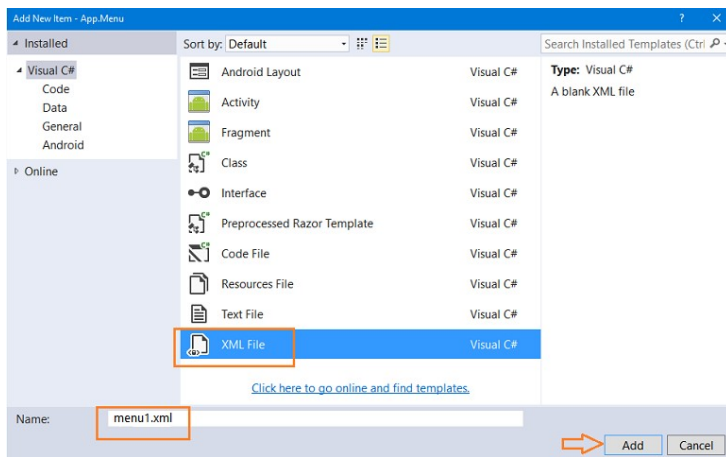
Na subpasta **values** temos o arquivo **Strings.xml** onde definimos as strings usadas no projeto;

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 começar criando uma nova pasta chamada **Menu** sob a pasta **Resources**.

A seguir vamos criar um arquivo XML na pasta menu com o nome **menu1.xml**:

Clique com o botão direito do mouse sobre a pasta **menu** e depois clique em **Add New Item**;



Clique no botão **Add** e a seguir inclua o código abaixo no arquivo XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item android:id="@+id/Adicionar"
    android:icon="@drawable/Add"
    android:title="@string/Adicionar"
    android:showAsAction="ifRoom"/>
<item android:id="@+id/Editar"
    android:icon="@drawable/Edit"
    android:title="@string/Editar"
    android:showAsAction="ifRoom"/>
<item android:id="@+id/Cancelar"
    android:icon="@drawable/Cancel"
    android:title="@string/Cancelar"
    android:showAsAction="ifRoom"/>
<item android:id="@+id/Deletar"
    android:icon="@drawable/Deleta"
    android:title="@string/Deletar"
    android:showAsAction="ifRoom"/>
</menu>
```

Neste arquivo definimos as opções do menu e para isso usamos as seguinte tags:

- **id** - definimos o Id para identificar a opção do menu;
- **icon** - definimos nome do arquivo de ícone que será exibido no menu (*deveremos incluir este arquivo na pasta drawable*);
- **title** - o título do menu (*deveremos incluir o texto no arquivo strings.xml*);
- **ShowAsAction** - Aqui estamos usando a opção **ifRoom** que indica que os ícones serão exibido se houver espaço na tela;

Cada item do menu contém uma identificação única, que permite que o sistema reconhecer o item quando o usuário selecionar uma opção, e um título, o qual é usado para apresentar a opção ao usuário.

Há também Propriedades adicionais, no caso estamos usando um ícone. Se estivéssemos usando uma ActionBar, o ícone seria exibido lado a lado.

1- Vamos Abrir o arquivo **Strings.xml** na pasta **Resources/values** e vamos incluir os textos definidos no arquivo menu1.xml conforme abaixo:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="Hello">Macoratti .net</string>
    <string name="Adicionar">Novo</string>
    <string name="Editar">Editar</string>
    <string name="Cancelar">Cancela</string>
    <string name="Deletar">Deleta</string>
    <string name="AppName">App.MenuOpcoes</string>
</resources>
```

Agora que já definimos as opções do menu e seus textos temos que sobrescrever o método **OnCreateOptionsMenu** para inflar o nosso menu e a seguir sobrescrever o método **OnOptionsItemSelected** para capturar o item selecionado pelo usuário.

Abra o arquivo **MainActivity.cs** e altere o código do arquivo conforme mostrado a seguir:

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

namespace App.MenuOpcoes
{
    [Activity(Label = "App.MenuOpcoes", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);
            SetContentView(Resource.Layout.Main);
        }
    }
}
```

```

    }

    public override bool OnCreateOptionsMenu(IMenu menu)
    {
        base.OnCreateOptionsMenu(menu);
        MenuInflater inflater = this.MenuInflater;
        inflater.Inflate(Resource.Menu.menu1, menu);

        return true;
    }

    public override bool OnOptionsItemSelected(IMenuItem item)
    {
        switch (item.ItemId)
        {
            case Resource.Id.Adicionar:
                var atividadeAdicionar = new Intent(this, typeof(ActivityAdicionar));
                StartActivity(atividadeAdicionar);
                break;
            case Resource.Id.Editar:
                Toast.MakeText(this, "Editar", ToastLength.Short).Show();
                return true;
            case Resource.Id.Deletar:
                Toast.MakeText(this, "Deletar", ToastLength.Short).Show();
                return true;
            case Resource.Id.Cancelar:
                Toast.MakeText(this, "Cancelar", ToastLength.Short).Show();
                return true;
        }
        return base.OnOptionsItemSelected(item);
    }
}

```

No método **OnCreateOptionsMenu**, a instância **MenuInflater** cria a estrutura de menu a partir do arquivo de recurso (*menu1.xml*) inflando-o em uma instância **IMenu**. Todos os itens de menu no recurso serão adicionados como filhos para o menu.

Quando lidamos com as seleções dos itens do menu, o item de menu que foi selecionado é passado para o método **OnOptionsItemSelected()**. Se o evento foi tratado, será retornado **true**, caso contrário, o sistema irá continuar a processar o evento.

Podemos usar qualquer uma das propriedades no item de menu, mas uma das mais usadas é a propriedade **ItemId**, que contém o ID que foi utilizado no arquivo de recursos. Esta identificação pode ser usada para determinar qual item foi selecionado.

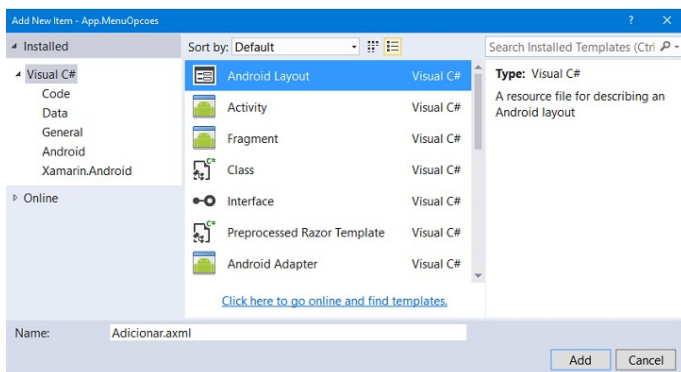
No exemplo a opção do menu **Adicionar** está invocando a Activity **ActivityAdicionar** que iremos criar a seguir usando o recurso **Intent**. Nas demais opções estamos usando também o recurso do método **Toast.MakeText** que cria uma pequena mensagem de retorno da operação exibindo um texto em um pequeno *pop up* durante um certo tempo.

Ex: **Toast.MakeText(this, "Cancelado !", ToastLength.Short).Show();**

Vamos iniciar definindo um layout para a Activity **ActivityAdicionar**.

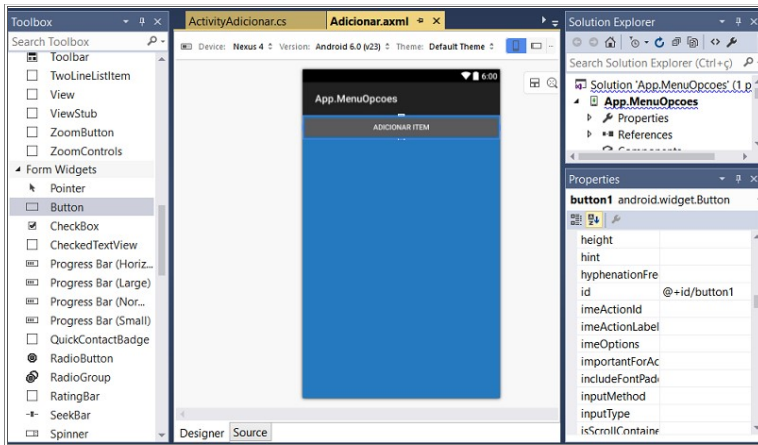
No menu **Project** clique em **Add New Item**;

Selecione o template **Android Layout** e informe o nome **Adicionar.xml**:



A seguir vamos definir um layout bem simples incluindo um **Button**, a partir da Toolbox, com a propriedade **Text** igual a "**Adicionar Item**".

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

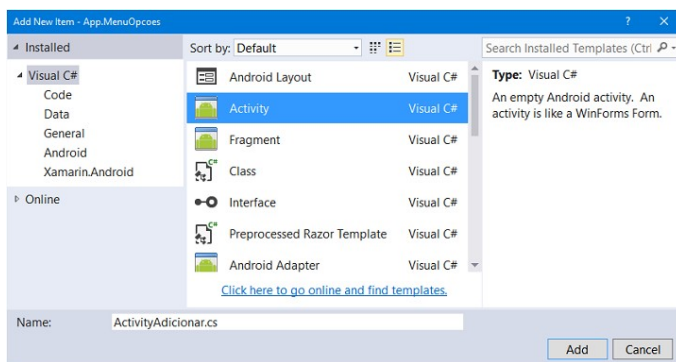


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:minWidth="25px"
    android:minHeight="25px"
    android:background="#2579BF">
    <Button
        android:text="Adicionar Item"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button1" />
</LinearLayout>
```

Agora podemos criar a Activity.

No menu **Project** clique em **Add New Item**;

Selecione o template **Activity** e informe o nome **ActivityAdicionar.cs**.



Veja como deve ficar o código do arquivo **MainActivity.cs**: (O código em azul foi o que incluímos)

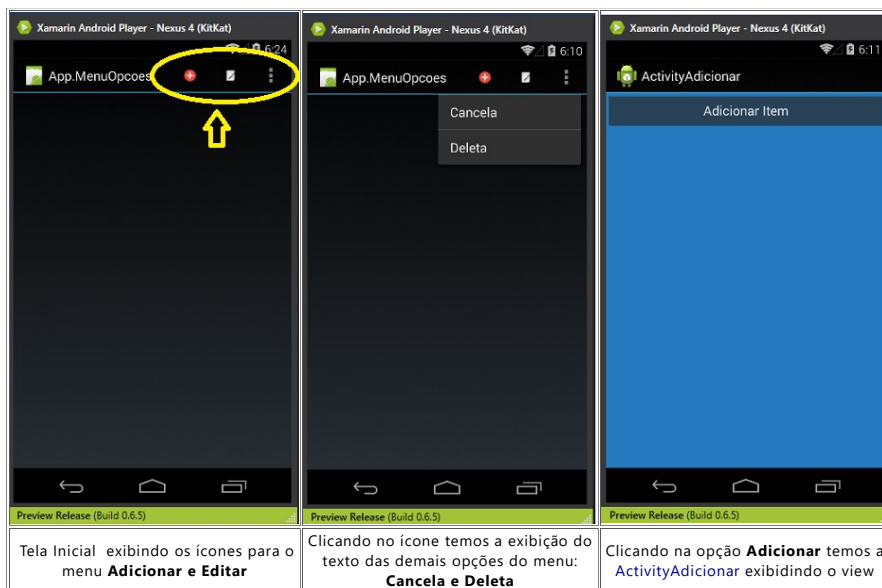
```
using Android.App;
using Android.OS;

namespace App.MenuOpcoes
{
    [Activity(Label = "ActivityAdicionar")]
    public class ActivityAdicionar : Activity
    {
        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

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

No código desta Activity apenas estamos definindo o layout vinculado para exibir a interface na tela do dispositivo.

Executando o projeto iremos obter o seguinte resultado:



Aguarde mais artigos sobre Xamarin Android.

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

Porque há um só Deus, e um só Mediador entre Deus e os homens, Jesus Cristo homem.

1 Timóteo 2:5

[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# Básico - 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