



++

45697096

ActionBar, Menu, Toolbar

Desenvolvimento Mobile e IoT - Android
Mobile Development and IoT - Android

45697096

45697096

...



Prof. Douglas Cabral <douglas.cabral@fiap.com.br>
<https://www.linkedin.com/in/douglascabral/>



++

45697096

ActionBar

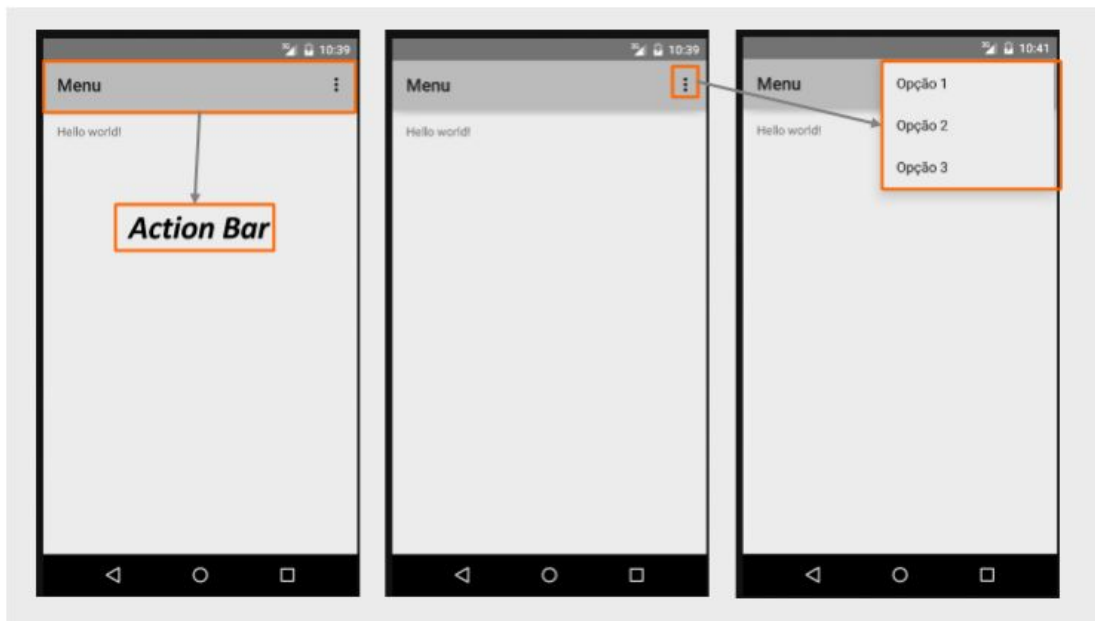


45697096

45697096



A ActionBar é uma toolbar primária que desde o Android 3.0 (API Level 11) aparece por padrão no topo dos aplicativos, conforme a imagem abaixo, podendo adicionar também um menu (opcional).



++

45697096

45697096

Menu

45697096



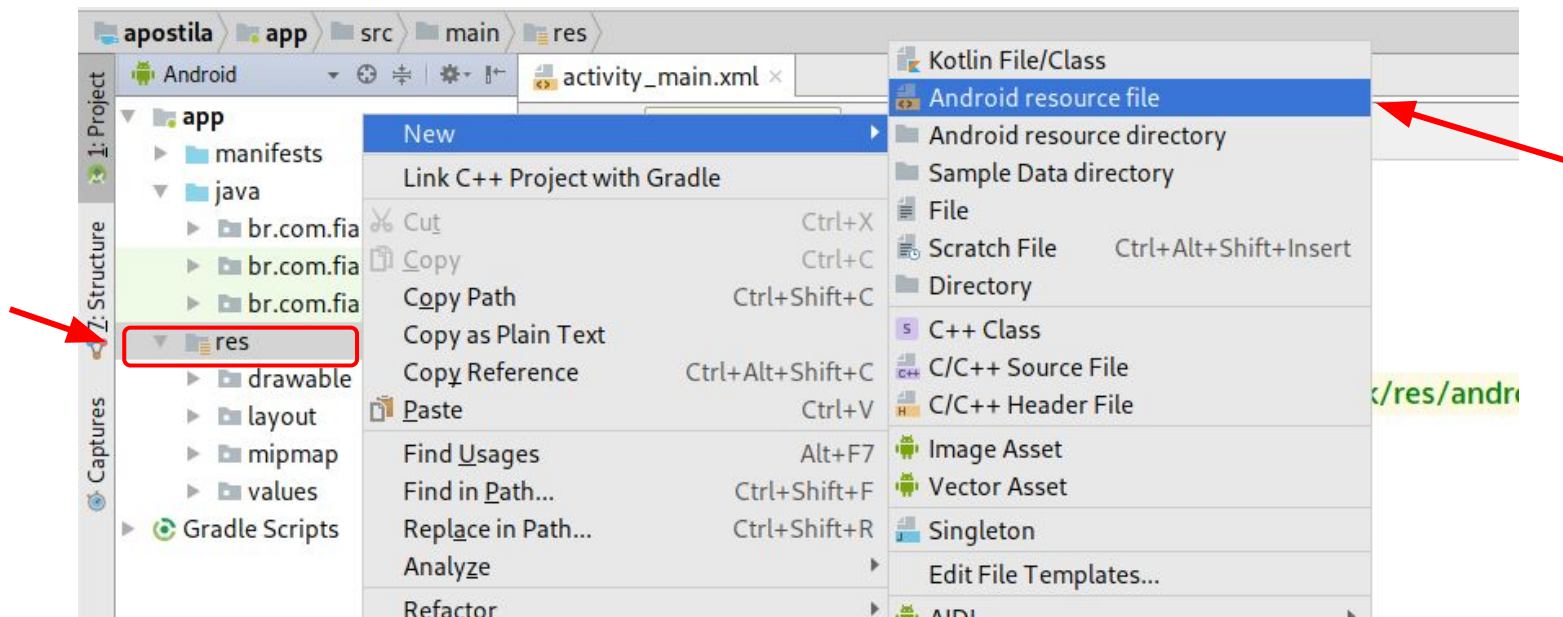
Para adicionar um menu, o processo é bem simples:

- Criar um arquivo de menu no projeto (**Ex: res/menu/menu_main.xml**)
- Adicionar os itens no arquivo de menu criado (tag item com atributos title, icon, id e showAsAction)
- Implementar o **ciclo de vida do menu** na **Activity** no qual o menu será apresentado.
 - método **onCreateOptionsMenu** (Infla o menu para o Activity)
 - método **onOptionsItemSelected** (Verifica qual item do menu foi clicado através do id do item)

Criação do menu

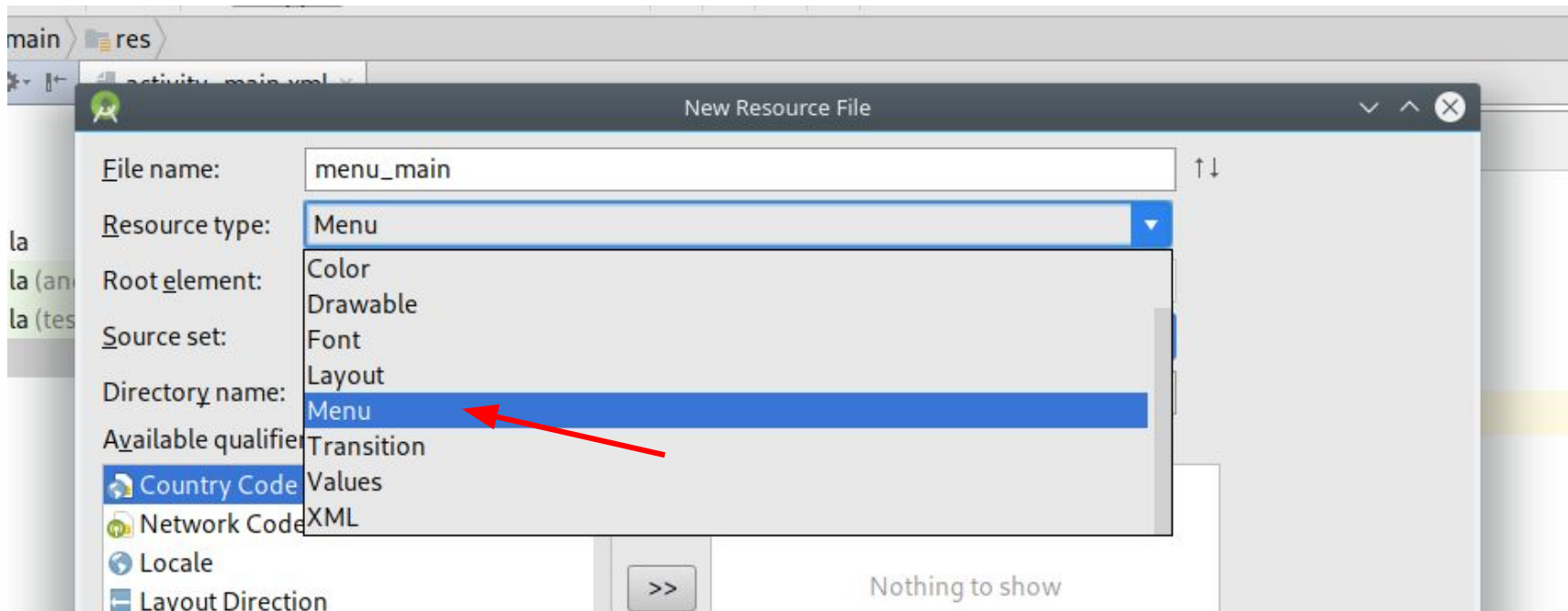
Para criar um arquivo de menu, basta clicar com o botão direito na pasta **res/** e escolher a opção:

New Android Resource File.



Criação do menu

Na próxima tela, basta dar um nome para o arquivo e escolher a opção **menu** em **Resource Type**:



Criação do menu

Agora é só adicionar os itens da seguinte maneira:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/mni_adicionar"
    android:title="Adicionar novo"
    app:showAsAction="always"
    android:icon="@drawable/ic_add_black_24dp"/>
</menu>
```


Criação do menu - Atributos do item



id ⇒ Identificador para o item do menu

title ⇒ Texto de exibição para o item

showAsAction ⇒ Opções de exibição

- always (Sempre visível na ActionBar)
- collapseActionView (oculta a opção para dentro de um menu expansível - tipo submenu)
- ifRoom (exibe se couber na ActionBar ou oculta dentro do menu caso necessário)
- never (nunca exibe direto na ActionBar, somente dentro no menu)

icon ⇒ Ícone para o item do menu

Para exibir o menu na Activity, basta implementar o método **onCreateOptionsMenu**, da seguinte forma:

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return super.onCreateOptionsMenu(menu);  
}
```



Criação do menu - Verificação do item clicado

Para implementar o evento de “**click**” no item do menu, basta implementarmos o método **onOptionsItemSelected**, verificando o id do item clicado:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    if (item.getItemId() == R.id.mni_adicionar) {
        // Implementar algo aqui...
        Toast.makeText(this, "Adicionar clicado...", Toast.LENGTH_SHORT).show();
    }

    return super.onOptionsItemSelected(item);
}
```

++

45697096

Submenu



Para criar um submenu, basta adicionar uma tag menu com suas opções dentro de um item já existente:

```
<item
  android:id="@+id/mni_submenu"
  android:title="Submenu">
  <menu>
    <item
      android:id="@+id/mni_submenu_01"
      android:title="Item 01 do submenu"/>

    <item
      android:id="@+id/mni_submenu_02"
      android:title="Item 02 do submenu"/>
  </menu>
</item>
```

++

45697096

ActionBar oculta

45697096

45697096

■ ■ ■

◇ ◇ ◇

Para remover uma **ActionBar** de uma Activity, dando a sensação de um aplicativo **FullScreen**, basta acessar o **AndroidManifest.xml** de sua aplicação e adicionar o atributo **android:theme** com um Style que não possua ActionBar. Exemplo:

```
<activity
    android:name=".MainActivity"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar">

    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>

        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```


++

45697096

45697096

Toolbar

45697096



A partir do Android API Level 21, é possível adicionar mais de uma Toolbar em um aplicativo.

Além disso, o uso de Toolbar no lugar da ActionBar padrão faz com que seja possível a utilização de muito mais recursos, como ícone da aplicação, formatação personalizada nas dimensões, posição, entre outras vantagens.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#F00"
        android:id="@+id/toolbarPrincipal"></android.support.v7.widget.Toolbar>

</LinearLayout>
```

Repare que, por questões de compatibilidade com outras versões da plataforma, foi utilizado a tag **android.support.v7.widget.Toolbar** e não apenas **Toolbar**.

Como dito anteriormente, é possível adicionar **mais de uma Toolbar** na Activity, porém, uma delas pode ser considerada a principal, através do método **setSupportActionBar** (quando utilizado a versão de compatibilidade do componente) ou **setActionBar** (quando utilizado a tag Toolbar apenas).

@Override

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    Toolbar toolbar = findViewById(R.id.toolbarPrincipal);
```

```
    setSupportActionBar(toolbar);
```

```
}
```



Outros atributos podem ser manipulados da Toolbar, conforme a imagem abaixo:

```
Toolbar toolbar = findViewById(R.id.toolbarPrincipal);

toolbar.setLogo(R.mipmap.ic_launcher);
toolbar.setTitle("Qualquer texto de título");
toolbar.setSubtitle("Subtítulo");
toolbar.setTitleTextColor(getResources().getColor(R.color.white));
toolbar.setSubtitleTextColor(getResources().getColor(R.color.white));

setSupportActionBar(toolbar);
```



Para inserir um ícone de voltar em uma Toolbar de uma Activity, basta colocar o seguinte código depois da definição da Toolbar:

```
Toolbar toolbar = findViewById(R.id.toolbarPrincipal);

toolbar.setLogo(R.mipmap.ic_launcher);
toolbar.setTitle("Qualquer texto de título");
toolbar.setSubtitle("Subtítulo");
toolbar.setTitleTextColor(getResources().getColor(R.color.white));
toolbar.setSubtitleTextColor(getResources().getColor(R.color.white));

setSupportActionBar(toolbar);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```



Para capturar o evento de clique do botão voltar, verificamos no mesmo método que verifica cliques do menu:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    if ( item.getItemId() == android.R.id.home) {
        finish();
    }

    return super.onOptionsItemSelected(item);
}
```


++

45697096

Dúvidas?

◇ ◇ ◇



Copyright © 2020 Prof. Douglas Cabral <douglas.cabral@fiap.com.br> <https://www.linkedin.com/in/douglascabral/>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).