

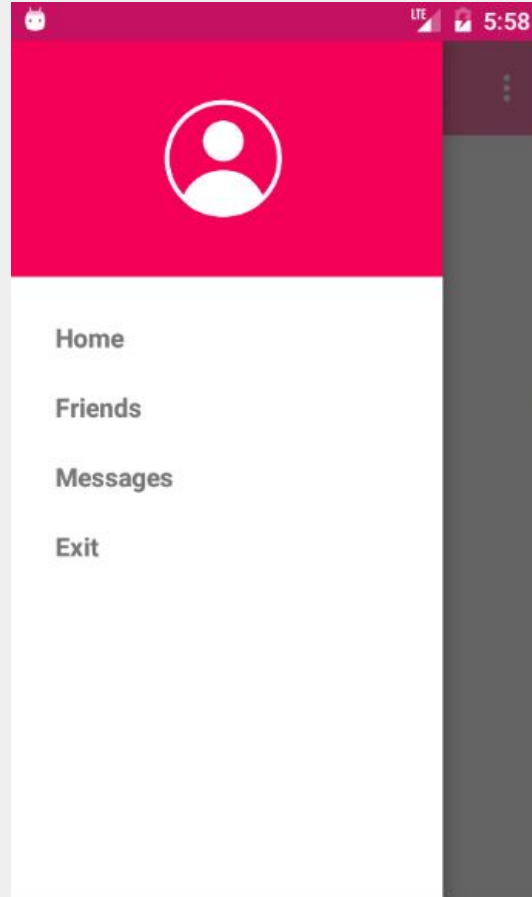
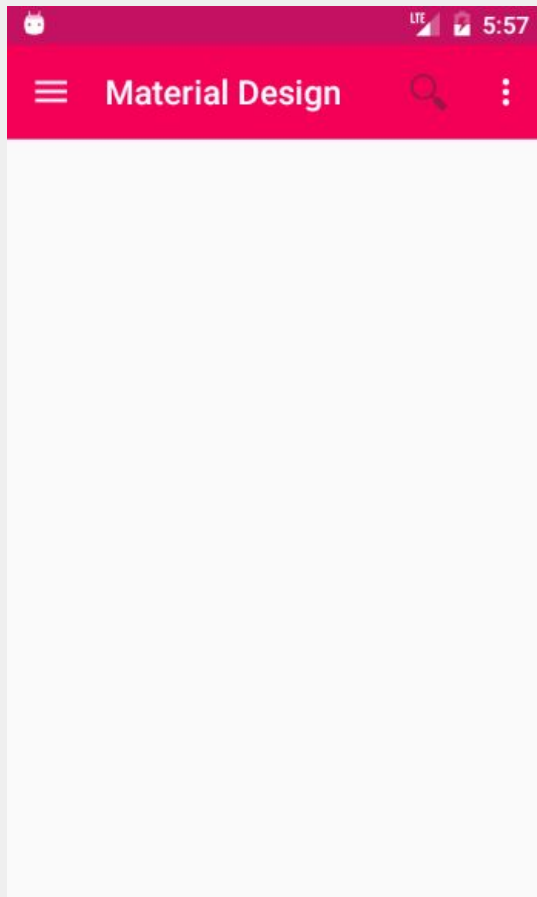
# Desenvolvimento de Layouts com os Recursos do Material Design

**Curso:** Sistemas para Internet

**Disciplina:** Dispositivos Móveis

**Integrantes:** Déborah, Maurício e Renato

# Nossa aplicação

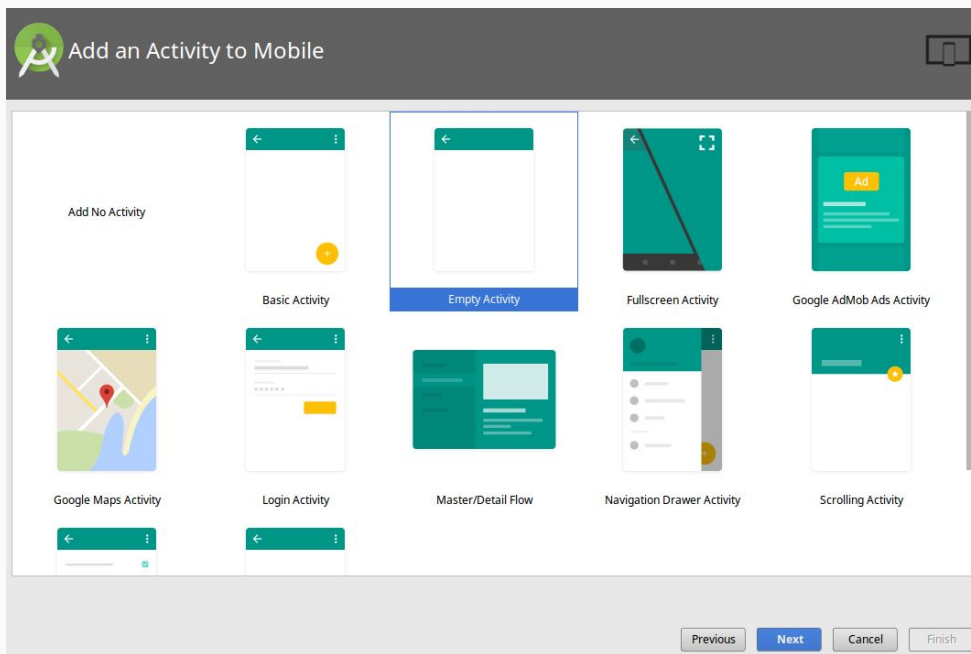


# Introdução

O Material Design é um guia abrangente para design visual, de movimento e de interação para diversas plataformas e dispositivos. Para usar o Material Design nos aplicativos Android, siga as orientações descritas nas [especificações do Material Design](#) e use os novos componentes e funcionalidades disponíveis no Android 5.0

# Criando um novo projeto

Com o Android Studio aberto vá em **File ⇒ New Project**. Preencha os campos necessários e na última tela selecione **Empty Activity**.



# Definindo Strings utilizadas na aplicação

O primeiro passo em nosso projeto será definir o arquivo **strings.xml**. Nele iremos colocar todas as strings utilizadas em nossa aplicação.

Vá em  
**res ⇒ values ⇒ strings.xml**  
e adicione as strings.

[Strings.xml](#)

```
1  <resources>
2      <string name="app_name">Material Design</string>
3      <string name="action_settings">Configurações</string>
4      <string name="action_search">Pesquisar</string>
5      <string name="drawer_open">Abrir</string>
6      <string name="drawer_close">Fechar</string>
7
8      <string name="nav_item_home">Início</string>
9      <string name="nav_item_friends">Amigos</string>
10     <string name="nav_item_notifications">Notificações</string>
11     <string name="nav_item_exit">Sair</string>
12
13     <!-- navigation drawer item labels -->
14     <string-array name="nav_drawer_labels">
15         <item>@string/nav_item_home</item>
16         <item>@string/nav_item_friends</item>
17         <item>@string/nav_item_notifications</item>
18         <item>@string/nav_item_exit</item>
19     </string-array>
20 </resources>
```

## Definindo Cores do tema padrão

Agora iremos configurar as cores padrões de nosso projeto no arquivo **colors.xml**.

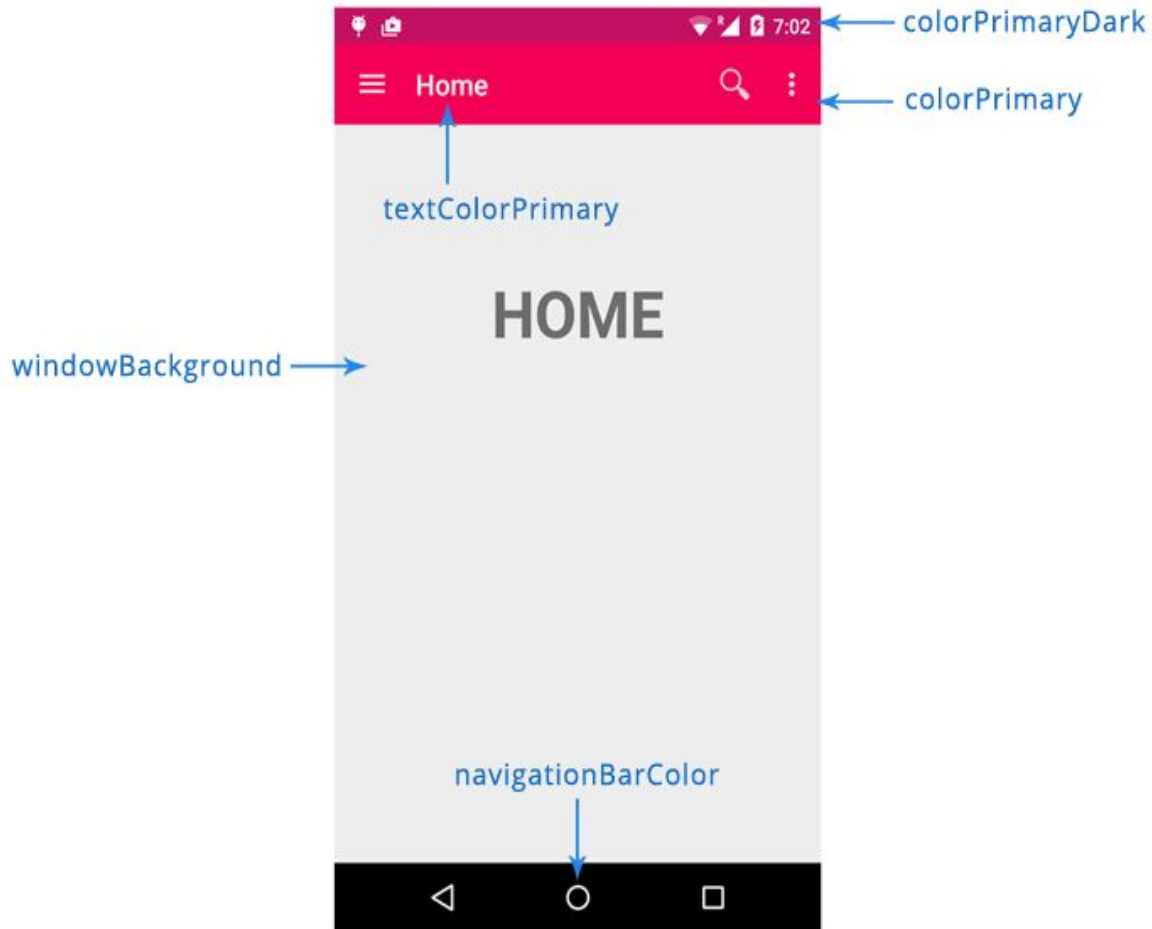
Nele iremos colocar todas as strings utilizadas em nossa aplicação.

Vá em **res** ⇒ **values** ⇒ **colors.xml** e adicione a tabela de cores abaixo.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="colorPrimary">#F50057</color>
4      <color name="colorPrimaryDark">#C51162</color>
5      <color name="textColorPrimary">#FFFFFF</color>
6      <color name="windowBackground">#FFFFFF</color>
7      <color name="navigationBarColor">#000000</color>
8      <color name="colorAccent">#FF80AB</color>
9  </resources>
```

[colors.xml](#)

Posição das cores definidas no  
colors.xml



## Definindo as dimensões

Em seguida vamos definir as dimensões do nosso aplicativo. Para isso abra **res ⇒ values ⇒ dimens.xml** e adicione as seguintes dimensões:

```
1 <resources>
2     <!-- Default screen margins, per the Android Design guidelines. -->
3     <dimen name="activity_horizontal_margin">16dp</dimen>
4     <dimen name="activity_vertical_margin">16dp</dimen>
5     <dimen name="nav_drawer_width">260dp</dimen>
6 </resources>
```

dimens.xml



## Nomear meu tema para MyThemeMaterial

Agora vamos abrir o arquivo **res** ⇒ **values** ⇒ **styles.xml** e definir os estilos abaixo. Os estilos definidos neste arquivo são comuns a todas as versões do Android.

```
1  <resources>
2      <style name="MyMaterialTheme" parent="MyMaterialTheme.Base">
3
4      </style>
5
6      <style name="MyMaterialTheme.Base" parent="Theme.AppCompat.Light.DarkActionBar">
7          <item name="windowNoTitle">true</item>
8          <item name="windowActionBar">false</item>
9          <item name="colorPrimary">@color/colorPrimary</item>
10         <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
11         <item name="colorAccent">@color/colorAccent</item>
12     </style>
13
14 </resources>
```

styles.xml

## Aplicando o tema

Agora temos os estilos básicos do material design pronto. Para aplicar o tema, abra o arquivo **AndroidManifest.xml** e modifique a tag **android:theme**.

```
android:theme="@style/MyMaterialTheme"
```

## Aplicando o tema

Feito isso, o **AndroidManifest.xml** deve estar como o exemplo ao lado:

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3    package="br.com.mjuniior.myapplication">
4
5    <application
6        android:allowBackup="true"
7        android:icon="@mipmap/ic_launcher"
8        android:label="@string/app_name"
9        android:supportsRtl="true"
10       android:theme="@style/MyMaterialTheme">
11       <activity android:name=".MainActivity">
12           <intent-filter>
13               <action android:name="android.intent.action.MAIN" />
14               <category android:name="android.intent.category.LAUNCHER" />
15           </intent-filter>
16       </activity>
17   </application>
18
19 </manifest>
```

## Executando o aplicativo

Agora, se você executar o aplicativo, verá que a cor da barra de notificação mudou para a cor que mencionamos em nossos arquivos de estilo.



## Adicionando a barra de ferramentas

Crie um arquivo xml chamado **toolbar.xml** em **res** ⇒ **layout** e adicione o elemento **android.support.v7.widget.Toolbar**. Ele criará a barra de ferramentas com altura e tema específico.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.v7.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:local="http://schemas.android.com/apk/res-auto"
4      android:id="@+id/toolbar"
5      android:layout_width="match_parent"
6      android:layout_height="wrap_content"
7      android:minHeight="?attr/actionBarSize"
8      android:background="?attr/colorPrimary"
9      local:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
10     local:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
```

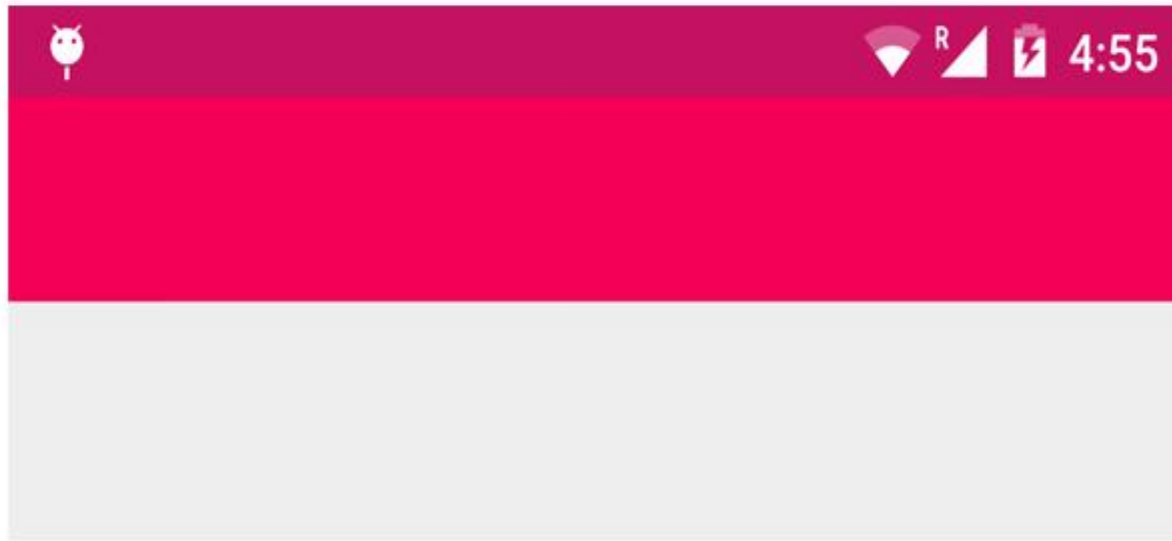
## Adicionando a barra de ferramentas

Em seguida, abra o arquivo **activity\_main.xml** e usando a tag **<include>** adicione a barra de ferramentas, como no exemplo abaixo:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/drawer_layout"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent">
8
9
10     <LinearLayout
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         android:orientation="vertical">
14
15         <LinearLayout
16             android:id="@+id/container_toolbar"
17             android:layout_width="match_parent"
18             android:layout_height="wrap_content"
19             android:orientation="vertical">
20
21             <include
22                 android:id="@+id/toolbar"
23                 layout="@layout/toolbar" />
24         </LinearLayout>
```

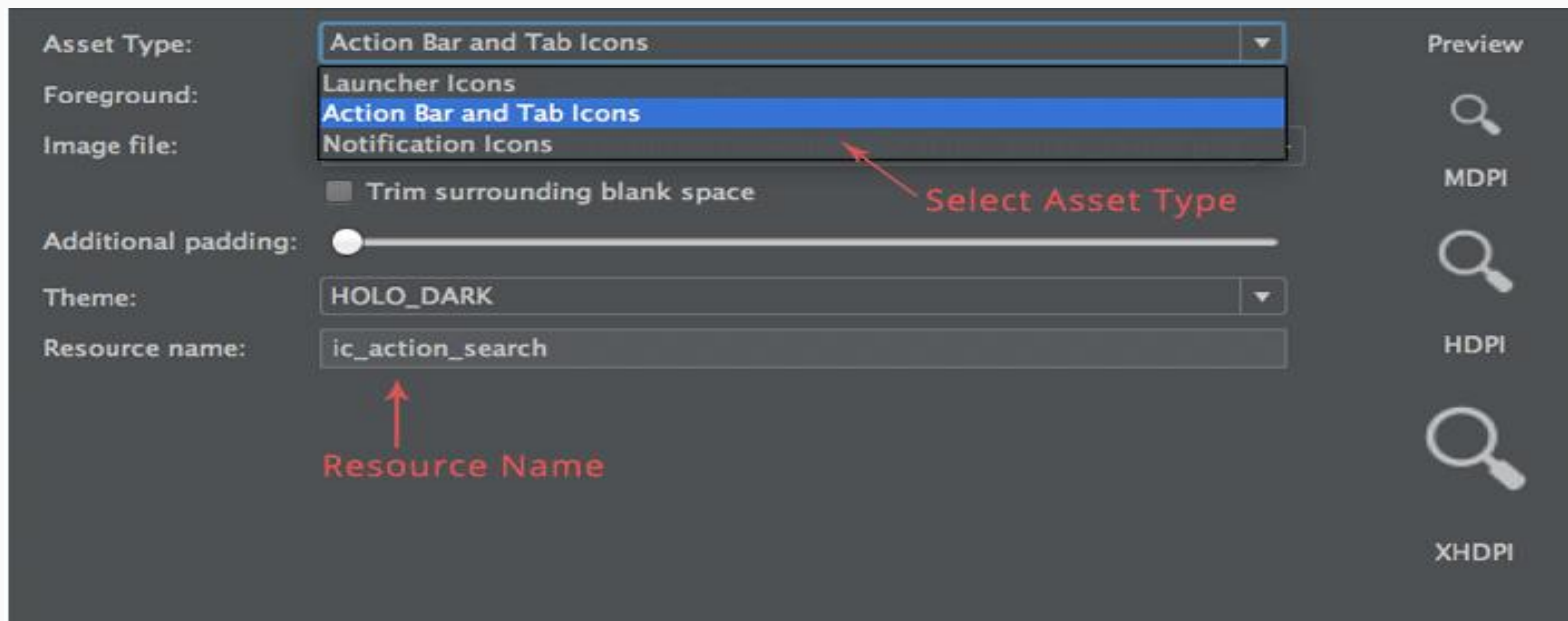
## Executando o aplicativo

Executando, aplicativo a barra de ferramentas será exibida. Como no exemplo abaixo:



## Inserindo ícones da barra de título

Inicialmente faça o download de um ícone de pesquisa e importe para o Android Studio como recurso de imagem. Em seguida clique com botão direito em **res** ⇒ **New** ⇒ **Image Asset**. Essa ação exibirá uma tela como a do exemplo abaixo. Nela você irá navegar até o ícone de pesquisa que vc baixou e selecionará a opção **Action Bar and Tab Icons** e dará o nome **ic\_search\_action** ao ícone.





## Inserindo ícones da barra de título

Uma vez que o ícone foi importado, abra o **menu\_main.xml** localizado dentro de **res** ⇒ **menu** e adicione o item de menu de busca como mostrado no exemplo ao lado:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3        xmlns:app="http://schemas.android.com/apk/res-auto"
4        xmlns:tools="http://schemas.android.com/tools"
5        tools:context=".MainActivity">
6
7      <item
8          android:id="@+id/action_search"
9          android:title="@string/action_search"
10         android:orderInCategory="100"
11         android:icon="@drawable/ic_action_name"
12         app:showAsAction="ifRoom" />
13
14      <item
15          android:id="@+id/action_settings"
16          android:title="@string/action_settings"
17          android:orderInCategory="100"
18          app:showAsAction="never" />
19  </menu>
```

# Inserindo ícones da barra de título

Agora abra seu **MainActivity.java** e faça as seguintes mudanças:

- Estender a classe **MainActivity.java** para **AppCompatActivity**.

```
17  
18     public class MainActivity extends AppCompatActivity {  
19
```

- Ativar a barra de ferramentas chamando **setSupportActionBar ()** e passar o objeto barra de ferramentas.

```
23         @Override  
24         protected void onCreate(Bundle savedInstanceState) {  
25             super.onCreate(savedInstanceState);  
26             setContentView(R.layout.activity_main);  
27  
28             mToolbar = (Toolbar) findViewById(R.id.toolbar);  
29  
30             getSupportActionBar(mToolbar);  
31             getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
32
```

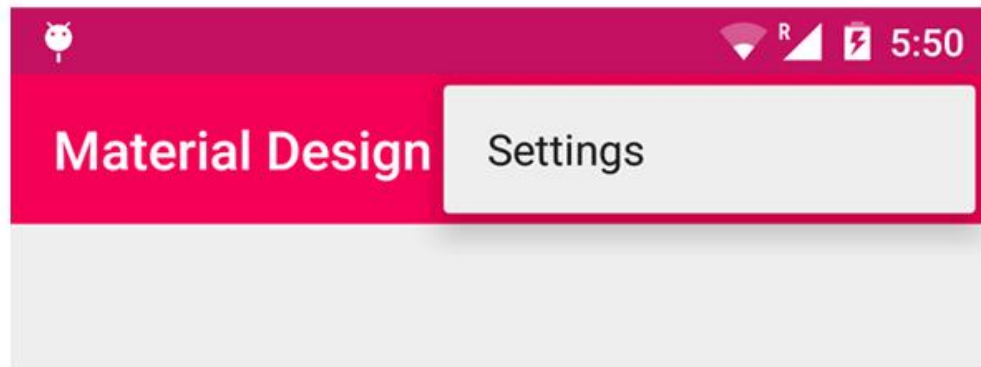
## Inserindo ícones da barra de título

- Fazer o override os métodos **onCreateOptionsMenu ()** e **onOptionsItemSelected()** para ativar os itens da barra de tarefas.

```
39
40
41 @Override
42 public boolean onCreateOptionsMenu(Menu menu) {
43     // Inflate the menu; this adds items to the action bar if it is present.
44     getMenuInflater().inflate(R.menu.menu_main, menu);
45     return true;
46 }
47
48 @Override
49 public boolean onOptionsItemSelected(MenuItem item) {
50     // Handle action bar item clicks here. The action bar will
51     // automatically handle clicks on the Home/Up button, so long
52     // as you specify a parent activity in AndroidManifest.xml.
53     int id = item.getItemId();
54
55     //noinspection SimplifiableIfStatement
56     if (id == R.id.action_settings) {
57         return true;
58     }
59
60     return super.onOptionsItemSelected(item);
61 }
```

## Executando o aplicativo

Assim que essas alterações estiverem sido realizadas, você verá um ícone na barra de pesquisa quando o aplicativo for executado.



## Adicionando navegação Drawer

Primeiramente, na pasta java do seu projeto crie três pacotes chamados: **activity**, **adapter** e **model**. Mova seu **Main Activity.java** para o pacote activity. Isso manterá seu projeto organizado.

Em seguida, abra o **build.gradle**, localizado dentro no módulo app, adicione as dependências exemplificadas na figura abaixo. Depois disso é necessário ir em **Build ⇒ Rebuild Project** para baixar as bibliotecas necessárias.

```
21
22 dependencies {
23     compile fileTree(dir: 'libs', include: ['*.jar'])
24     compile 'com.android.support:appcompat-v7:23.4.0'
25     compile 'com.android.support:recyclerview-v7:23.4.0'
26     compile 'com.android.support:support-v4:23.4.0'
27     testCompile 'junit:junit:4.12'
28 }
```

## Adicionando navegação Drawer

No pacote `model` crie a classe **NavDrawerItem.java** com o código da figura abaixo. Esta classe define cada linha do menu drawer.

```
1  package br.com.mjunior.model;
2
3  /**
4   * Created by mjunior on 28/11/16.
5   */
6
7  public class NavDrawerItem {
8      private boolean showNotify;
9      private String title;
10
11
12      public NavDrawerItem() {
13
14      }
15
16      public NavDrawerItem(boolean showNotify, String title) {
17          this.showNotify = showNotify;
18          this.title = title;
19      }
20
```

```
20
21      public boolean isShowNotify() {
22          return showNotify;
23      }
24
25      public void setShowNotify(boolean showNotify) {
26          this.showNotify = showNotify;
27      }
28
29      public String getTitle() {
30          return title;
31      }
32
33      public void setTitle(String title) {
34          this.title = title;
35      }
36
37  }
```

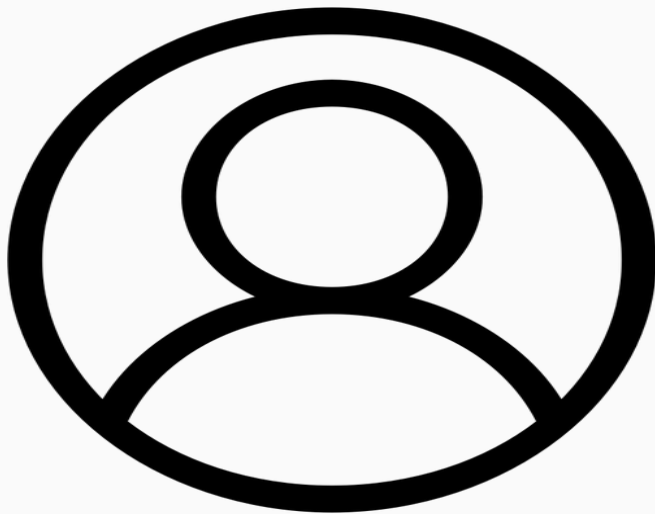
## Adicionando navegação Drawer

Dentro de **res** ⇒ **layout**, crie um layout chamado **nav\_drawer\_row.xml** com o código exemplificado abaixo. Se você desejar personalizar o item do menu, você deverá fazer as mudanças nesse arquivo.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="wrap_content"
5      android:clickable="true">
6
7      <TextView
8          android:id="@+id/title"
9          android:layout_width="fill_parent"
10         android:layout_height="wrap_content"
11         android:paddingLeft="30dp"
12         android:paddingTop="10dp"
13         android:paddingBottom="10dp"
14         android:textSize="15dp"
15         android:textStyle="bold" />
16
17  </RelativeLayout>
```

## Adicionando navegação Drawer

Faça download de um ícone de usuário e coloque na pasta **drawable**. É um passo opcional, mas esse ícone será usado no cabeçalho da navegação Drawer.





# Adicionando navegação Drawer

A seguir, crie um outro layout chamado **fragment\_navigation\_drawer.xml** e adicione o código ao lado.

Esse layout contém uma seção de cabeçalho para exibir as informações do usuário e uma RecyclerView para exibir a visualização da lista.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:background="@android:color/white">
6
7
8      <RelativeLayout
9          android:id="@+id/nav_header_container"
10         android:layout_width="match_parent"
11         android:layout_height="140dp"
12         android:layout_alignParentTop="true"
13         android:background="@color/colorPrimary">
14
15         <ImageView
16             android:layout_width="70dp"
17             android:layout_height="70dp"
18             android:scaleType="fitCenter"
19             android:layout_centerInParent="true" />
20
21     </RelativeLayout>
22
23
24     <android.support.v7.widget.RecyclerView
25         android:id="@+id/drawerList"
26         android:layout_width="match_parent"
27         android:layout_height="wrap_content"
28         android:layout_below="@id/nav_header_container"
29         android:layout_marginTop="15dp" />
30
31
32 </RelativeLayout>
```

## Adicionando navegação Drawer

À medida que o RecyclerView é personalizado, precisamos de uma classe adaptadora para processar o XML. Então, no pacote adapter crie a classe **NavigationDrawerAdapter.java** com o código do link abaixo. Essa classe usa o **nav\_drawer\_row.xml** e torna o **RecyclerView** um menu Drawer.

<https://github.com/mjunior/Learning-android-material-design/blob/master/app/src/main/java/br/com/mjunior/adapter/NavigationDrawerAdapter.java>

## Adicionando navegação Drawer

No pacote no activity crie um fragment chamado **FragmentManager.java**. No Android Studio, para criar um fragment clique com o botão direito sobre a **activity** ⇒ **New** ⇒ **Fragment** ⇒ **Fragment (Blank)**. Siga o exemplo do link abaixo:

<https://github.com/mjunior/Learning-android-material-design/blob/master/app/src/main/java/br/com/mjunior/activity/FragmentManager.java>

## Adicionando navegação Drawer

Por fim, abra o **activity\_main.xml** e modifique o layout como no código do link abaixo. Assim, estamos adicionando **android.support.v4.widget.DrawerLayout** para exibir o menu drawer de navegação.

Além disso não se esqueça de dar o caminho correto para o seu **FragmentManager** na tag **<fragment>**.

[https://github.com/mjunior/Learning-android-material-design/blob/master/app/src/main/res/layout/activity\\_main.xml](https://github.com/mjunior/Learning-android-material-design/blob/master/app/src/main/res/layout/activity_main.xml)

## Adicionando navegação Drawer

Agora, temos todos os arquivos de layout e classes java prontos. Vamos fazer as seguintes mudanças necessárias no MainActivity .

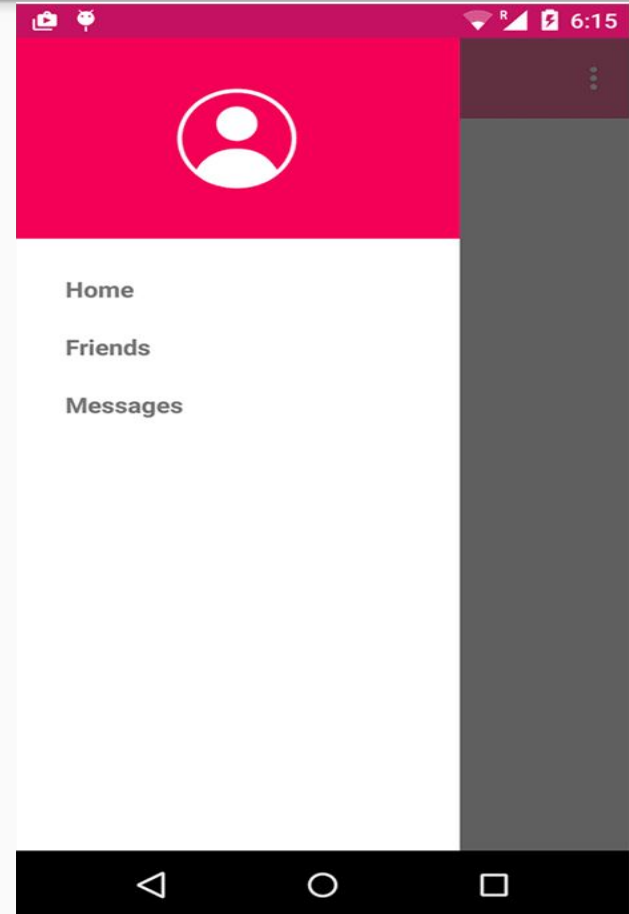
- Fazer a classe implementar **FragmentDrawer.FragmentDrawerListener** e adicionar o método override **onDrawerItemSelected()**
- Criar uma instância de **FragmentDrawer** e setar o listeners do drawer.

Veja o exemplo do código no link abaixo:

<https://github.com/mjunior/Learning-android-material-design/blob/master/app/src/main/java/br/com/mjunior/myapplication/MainActivity.java>

## Executando o aplicativo

Agora, se você executar o aplicativo e ver o resultado.



**Obrigado!**