

Programação Para Dispositivos Móveis I

AppBar, Toolbar e NAVIGATION DRAWERS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

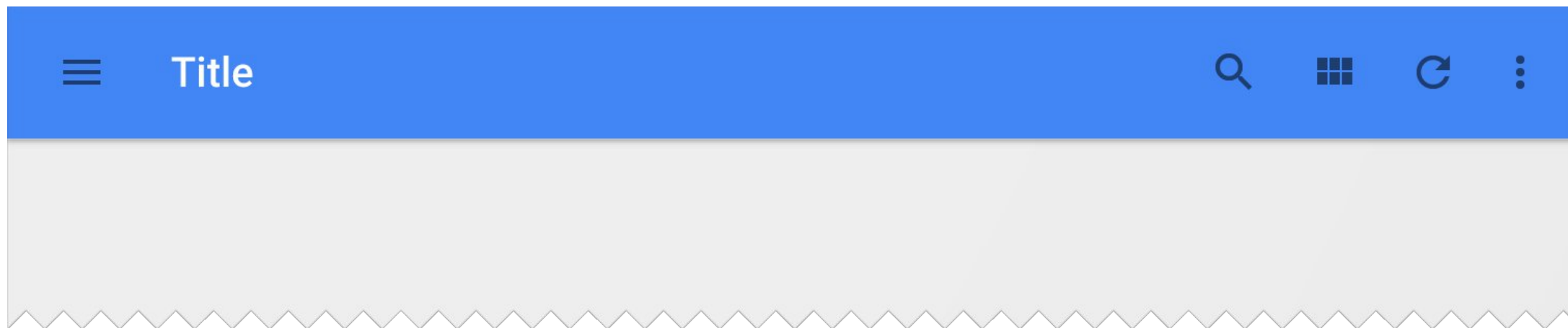
Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Índice

- AppBar;
- Criar uma AppBar;
- AppBar Menu;
- Navigation Drawer;
- Leitura Adicional.

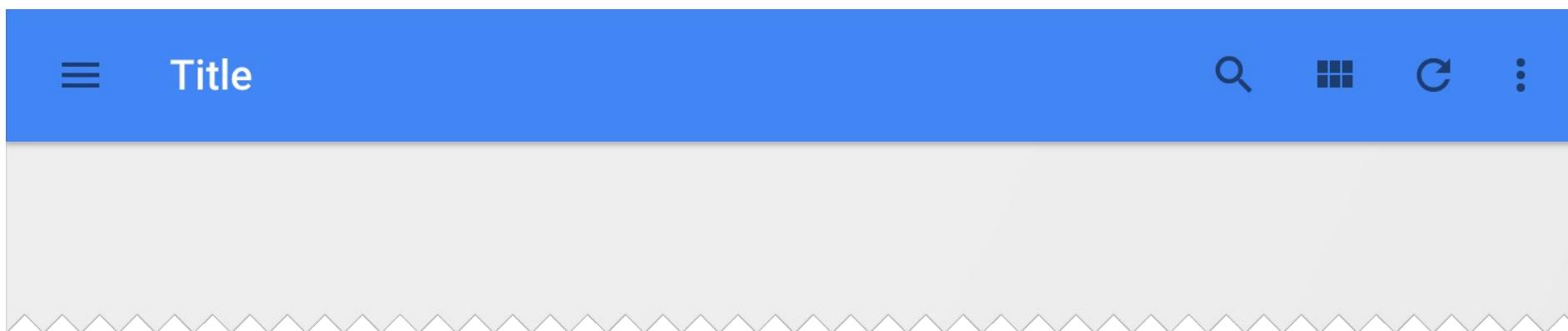
App Bar & Toolbar



App Bar

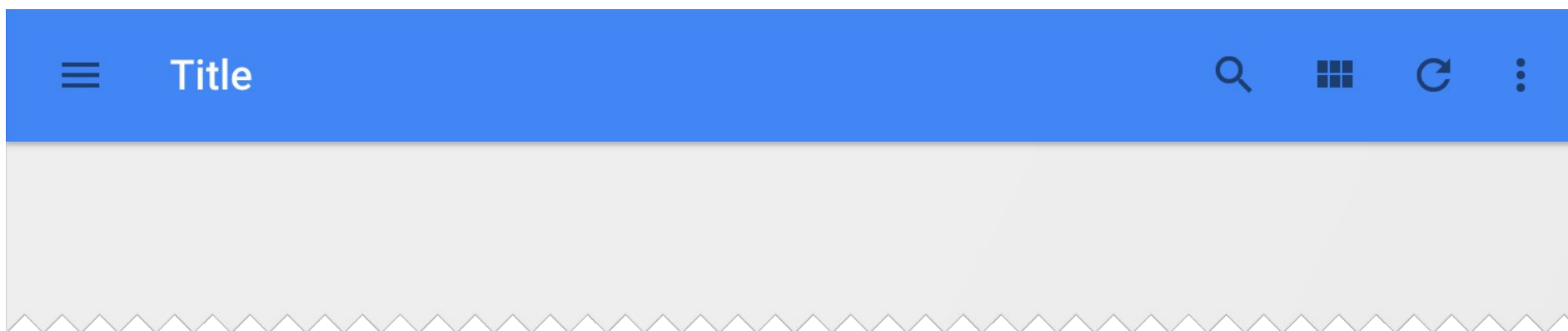
A App Bar foi introduzida no Android Lollipop (API 21) e é o sucessor da ActionBar.

- É um dos principais elementos de design de uma activity pois define uma estrutura visual que é familiar para o utilizador;
- Ao utilizar a App Bar as aplicações funcionam todas de forma consistente e similar;



App Bar

- **Principal objetivo:**
 - Indicar a localização do utilizador na aplicação (navegação);
 - Acesso a ações comuns a toda a aplicação (ex.: procura);
 - Suportar diferentes conceitos de navegação: tabs ou dropdown lists.



App Bar

Estrutura App Bar



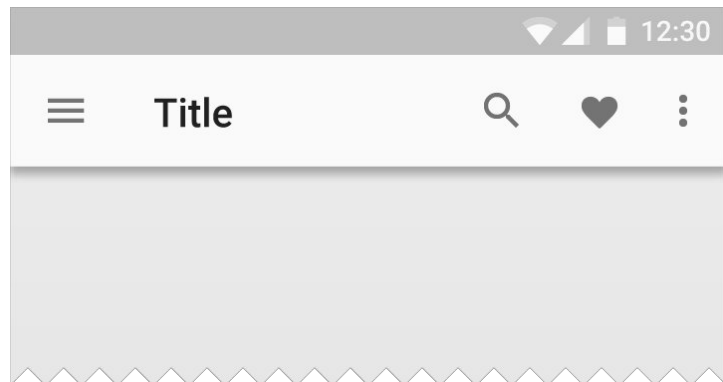
Nav icon: pode controlar a abertura/fecho do NavigationDrawer ou uma seta de navegação;

Title: normalmente reflete a pagina atual do utilizador (pode ser o titulo da app, titulo da activity ou filtro);

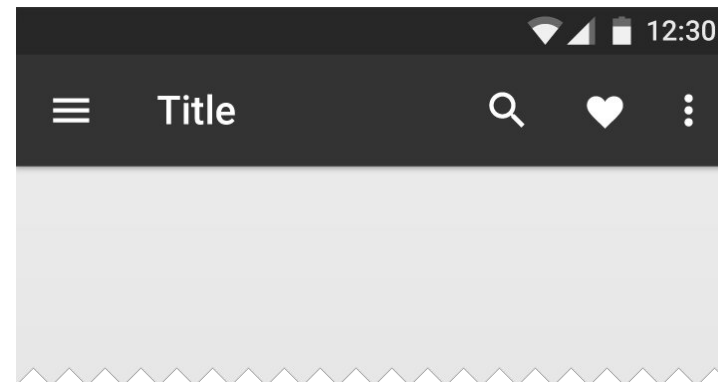
Action Icons: contem ações de fácil acesso ao utilizador: procura, definições, etc.

App Bar

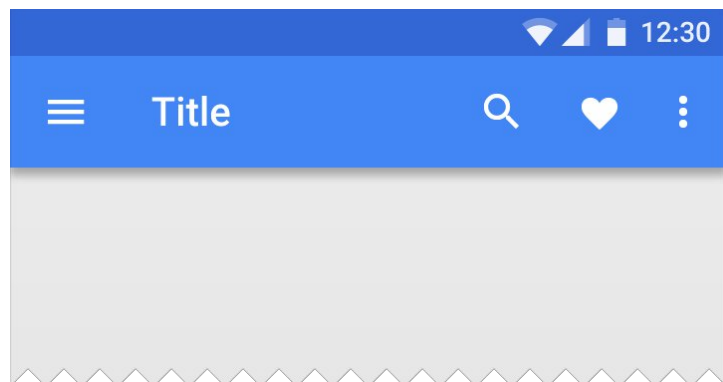
Themes



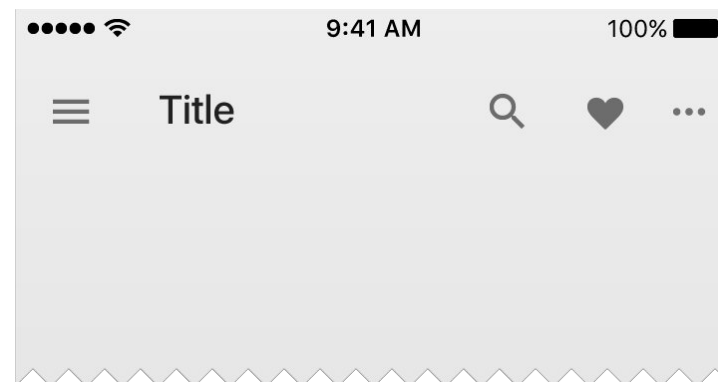
Light



Dark



Colored



Transparent

Adicionar AppBar

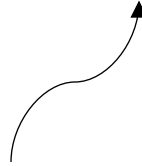
1. Verificar as dependências no ficheiro `build.gradle(module)`

```
dependencies {  
    ( ... )  
    implementation 'androidx.appcompat:appcompat:1.7.0'  
    ( ... )  
}
```


Adicionar AppBar

2. Modificar o tema da aplicação [themes -> themes.xml]

```
<resources xmlns:tools="http://schemas.android.com/tools">  
    <!-- Base application theme. -->  
    <style name="Theme.AppTheme" parent="Theme.MaterialComponents.DayNight.NoActionBar">  
        ( ... )  
    </style>  
</resources>
```



Como vamos substituir a ActionBar, necessitamos de um tema que não a inclua

Adicionar AppBar

3. Adicionar a toolbar ao layout da activity [activity_main.xml]

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

    <androidx.appcompat.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        android:elevation="4dp" />

</LinearLayout>
```

O atributo elevation faz parte das recomendações do Material Design

Adicionar AppBar

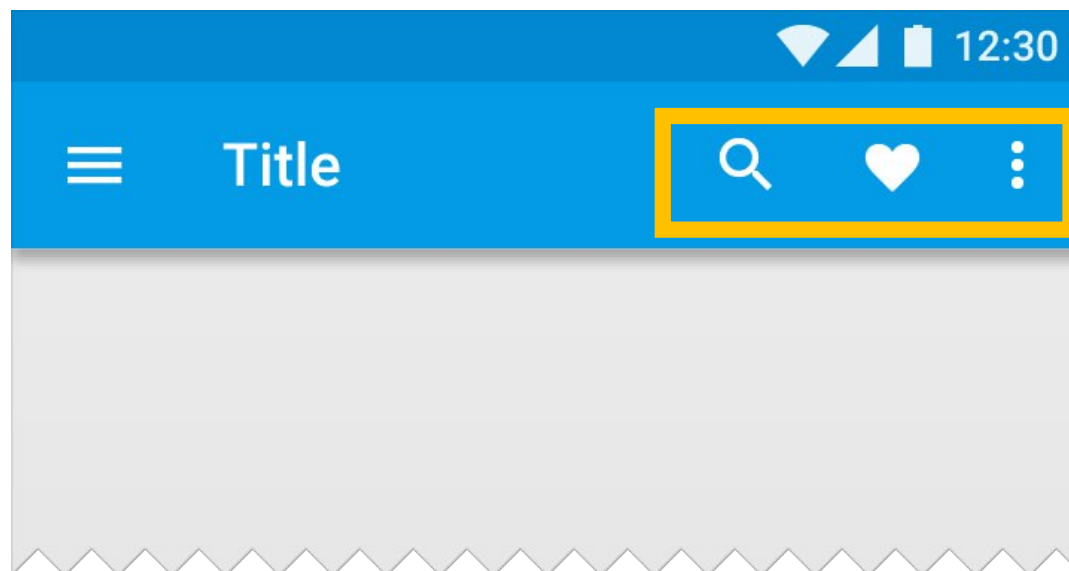
4. Adicionar a toolbar [MainActivity.java]

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Toolbar toolbar = (Toolbar) findViewById(R.id.my_toolbar);
    setSupportActionBar(toolbar);
}
```

AppBar Menu

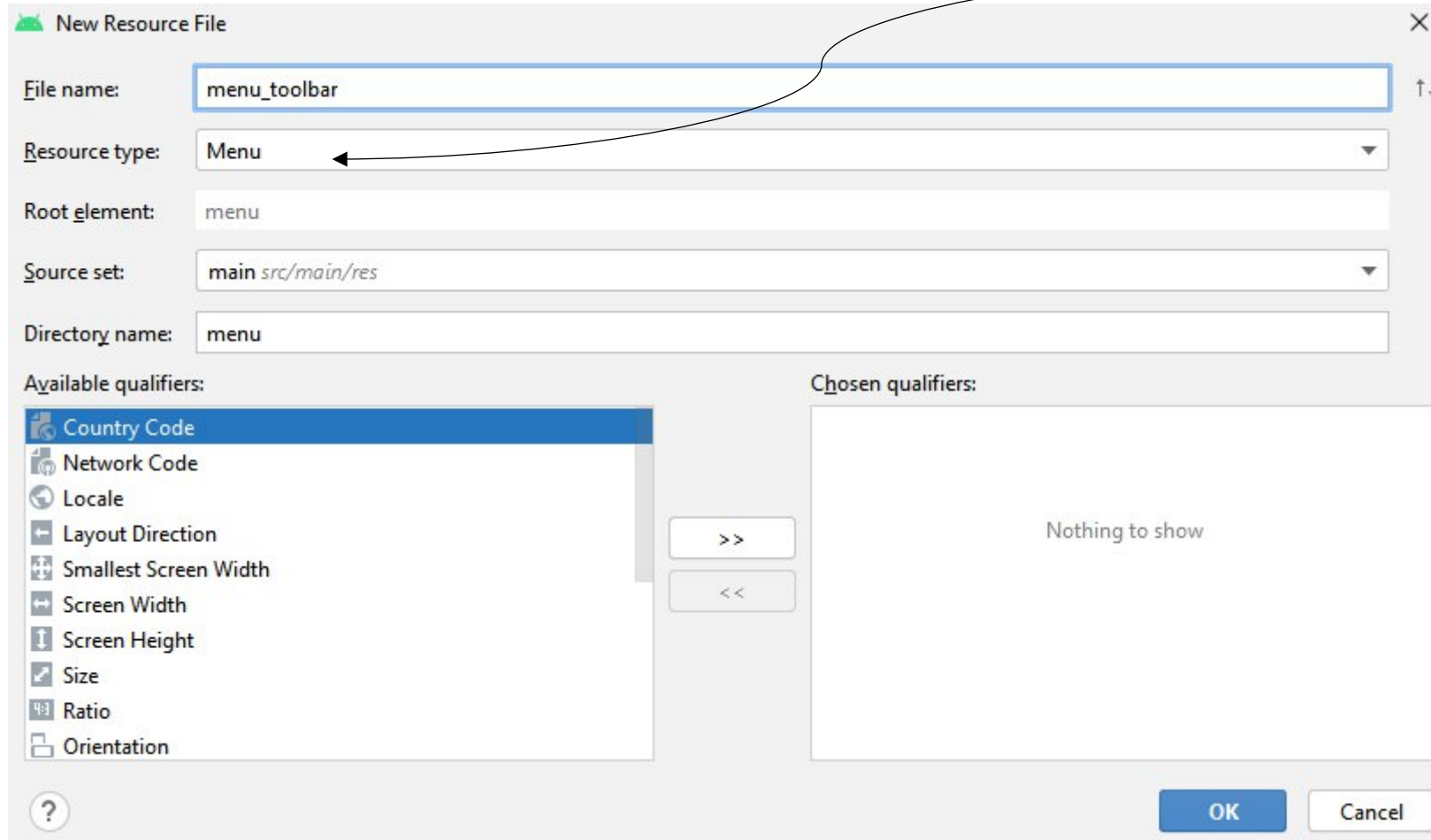
- A Toolbar permite definir um conjunto de ações através de um conceito chamado “Menu”.
 - Cada activity ou fragment é responsável por definir um menu e de que forma são mostradas essas ações (icon ou dropdown);



AppBar Menu

1. Criar ficheiro menu[res/menu/menu_toolbar.xml]

Garantir que escolhemos Menu como valor de Resource Type



AppBar Menu

2. Adicionar items[res/menu/menu_toolbar.xml]

```
<?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">
  <!-- "Mark Favorite", should appear as action button if possible -->
  <item
    android:id="@+id/action_favorite"
    android:icon="@drawable/ic_favorite_black_48dp"
    android:title="@string/action_favorite"
    app:showAsAction="ifRoom" />

  <item android:id="@+id/action_search"
    android:title="@string/action_search"
    android:icon="@drawable/ic_search"
    app:showAsAction="ifRoom|collapseActionView"
    app:actionViewClass="androidx.appcompat.widget.SearchView" />

  <!-- Settings, should always be in the overflow -->
  <item android:id="@+id/action_settings"
    android:title="@string/action_settings"
    app:showAsAction="never" />
</menu>
```

Este tipo de ação indica que o item irá aparecer na AppBar sempre que haja espaço disponível

Este tipo de ação indica que o item apenas irá sempre aparecer na zona de overflow

AppBar Menu

3. Adicionar o menu[MainActivity.java]

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

AppBar Menu

4. Adicionar comportamento aos items[MainActivity.java]

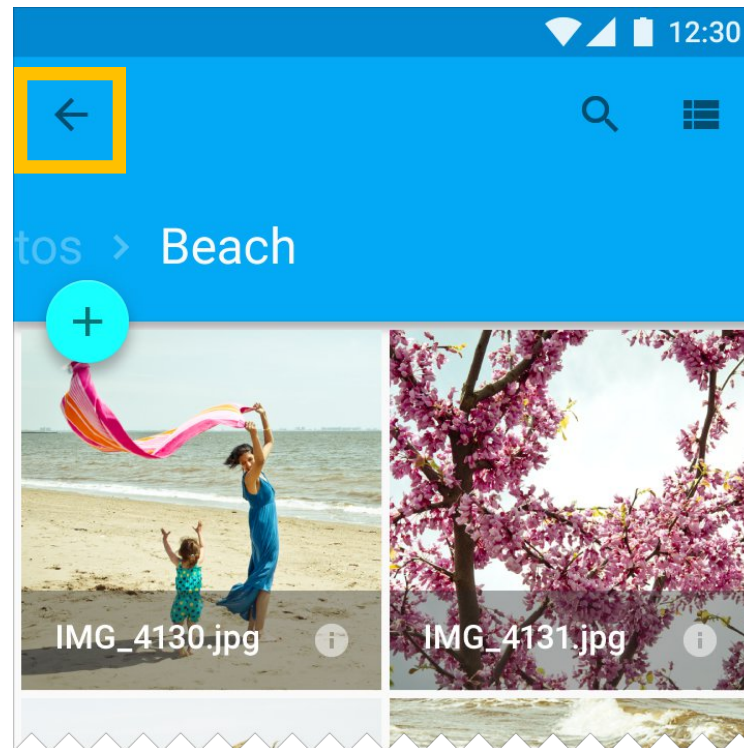
```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_settings:
            // User chose the "Settings" action, should be prompted
            // with settings activity
            return true;

        case R.id.action_favorite:
            // User chose the "Favorite" action, mark the current item
            // as a favorite...
            return true;

        default:
            // If we got here, the user's action was not recognized.
            // Invoke the superclass to handle it.
            return super.onOptionsItemSelected(item);
    }
}
```


AppBar Menu (Up Action)

Permite ao utilizador navegar para o ecrã anterior



AppBar Menu (Up Action)

1. Reutilizar a AppBar[toolbar.xml]

```
<androidx.appcompat.widget.Toolbar  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="?attr/actionBarSize"  
    android:background="?attr/colorPrimary"  
    android:elevation="4dp" />
```

AppBar Menu (Up Action)

2. Incluir a toolbar na Activity[activity_settings.xml]

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

    <include
        android:id="@+id/my_toolbar"
        layout="@layout/toolbar" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_settings" />
</LinearLayout>
```

Inclusão do layout da AppBar

AppBar Menu (Up Action)


3. Adicionar toolbar na Activity [SettingsActivity.java]

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_settings);

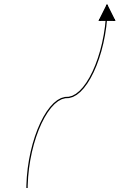
    Toolbar toolbar = (Toolbar) findViewById(R.id.my_toolbar);
    toolbar.setTitle("Settings");
    setSupportActionBar(toolbar);

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

Modificação do
título da toolbar



Adição do botão
UpAction



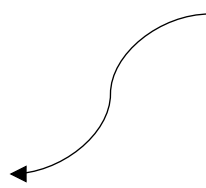
AppBar Menu (Up Action)

4. Adicionar Hierarquia [AndroidManifest.xml]

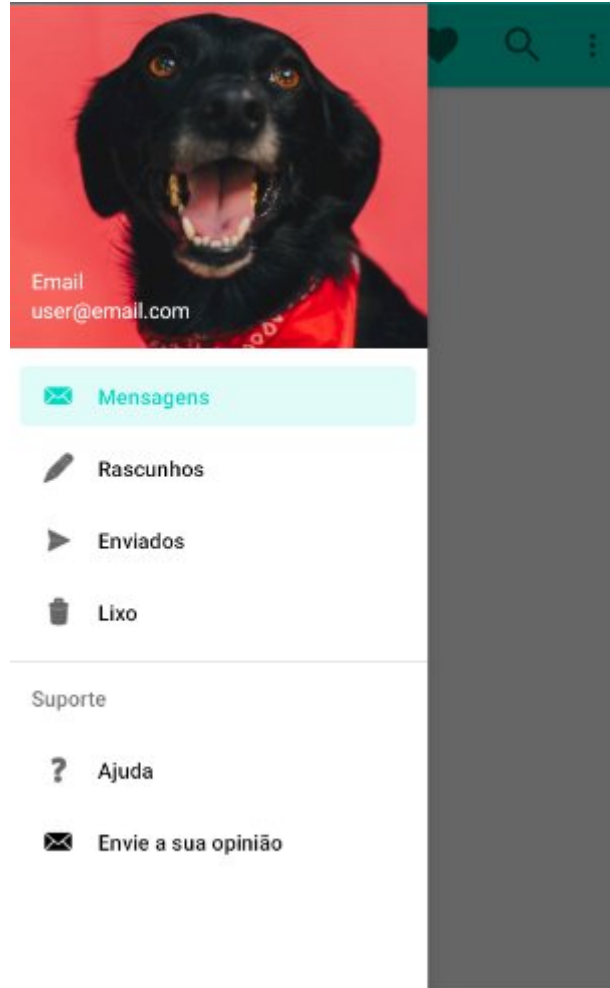
```
<activity android:name=".SettingsActivity"
    android:parentActivityName=".MainActivity">

    <!-- Parent activity meta-data to support 4.0 and lower -->
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity" />
</activity>
```

O click na
UpAction irá
retornar à
MainActivity



Navigation Drawer



Navigation Drawer

1. Definir o header [nav_header.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="230dp">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleType="centerCrop"
        android:src="@drawable/profile_pic" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="bottom|left"
        android:layout_margin="16dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/app_name"
            android:textAppearance="@style/TextAppearance.AppCompat.Body1"
            android:textColor="@color/white" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/user_name"
            android:textColor="@color/white" />

    </LinearLayout>
</FrameLayout>
```

Este atributo coloca
o texto no canto
inferior esquerdo da
imagem

Navigation Drawer

2. Adicionar items [menu_nav.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_inbox"
            android:icon="@android:drawable/sym_action_email"
            android:title="@string/nav_inbox"
            android:checked="true" />
        <item
            android:id="@+id/nav_drafts"
            android:icon="@android:drawable/ic_menu_edit"
            android:title="@string/nav_drafts" />
        <item
            android:id="@+id/nav_sent"
            android:icon="@android:drawable/ic_menu_send"
            android:title="@string/nav_sent" />
        <item
            android:id="@+id/nav_trash"
            android:icon="@android:drawable/ic_menu_delete"
            android:title="@string/nav_trash" />
    </group>

    (...)
</menu>
```



O item que vai estar selecionado por defeito

Navigation Drawer

2. Adicionar items [menu_nav.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

(...)

<item android:title="@string/nav_support">
    <menu>
        <item
            android:id="@+id/nav_help"
            android:icon="@android:drawable/ic_menu_help"
            android:title="@string/nav_help" />
        <item
            android:id="@+id/nav_feedback"
            android:icon="@android:drawable/sym_action_email"
            android:title="@string/nav_feedback" />
    </menu>
</item>
</menu>
```

Suporte

? Ajuda

✉ Envie a sua opinião

Navigation Drawer

3. Adicionar a Navigation Drawer ao Layout [activity_main.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

O element de Root é
um DrawerLayout

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <include
        android:id="@+id/my_toolbar"
        layout="@layout/toolbar" />
    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

O "antigo" layout que
tinhamos

Definição de um
container para os
fragments

```
<com.google.android.material.navigation.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="left"
    app:headerLayout="@layout/nav_header"
    app:menu="@menu/menu_nav" />
```

Definição da
NavigationView

Definição do layout e
dos items do menu da
NavigationView

```
</androidx.drawerlayout.widget.DrawerLayout>
```

Navigation Drawer

4. Adicionar a ação de toggle [MainActivity.java]

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    (...)
```

```
        drawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);  
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this,
```

O elemento
DrawerLayout

```
        drawerLayout,  
        toolbar,  
        R.string.nav_open_drawer,  
        R.string.nav_close_drawer);
```

A toolbar previamente
criada

```
        drawerLayout.addDrawerListener(toggle);
```

```
        toggle.syncState();
```

```
    }
```

Navigation Drawer

5. Responder aos clicks[MainActivity.java]

Implementar a interface do listener

```
public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener
```

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

```
(...)
```

```
    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);  
    navigationView.setNavigationItemSelectedListener(this);
```

Adicionar o listener

Navigation Drawer

5. Responder aos clicks[MainActivity.java]

```
@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
    Fragment fragment = null;

    switch (item.getItemId()) {
        case R.id.nav_drafts:
            fragment = new DraftsFragment();
            break;
        case R.id.nav_trash:
            fragment = new TrashFragment();
        case R.id.nav_sent:
            fragment = new SentFragment();
            break;
        default:
            fragment = new InboxFragment();
    }
    if(fragment != null){
        fragmentTransaction = getSupportFragmentManager().beginTransaction();
        fragmentTransaction.replace(R.id.fragment_container, fragment);
        fragmentTransaction.addToBackStack(null);
        fragmentTransaction.commit();
    }
    drawerLayout.closeDrawer(GravityCompat.START);

    return true;
}
```

Navigation Drawer

6. Fechar a Navigation quando se utiliza o Back Button[MainActivity.java]

```
@Override
public boolean onSupportNavigateUp() {
    onBackPressed();
    return super.onSupportNavigateUp();
}
```

O método é chamado quando se carrega no Back Button

```
@Override
public void onBackPressed() {
    if(drawerLayout.isDrawerOpen(GravityCompat.START)){
        drawerLayout.closeDrawer(GravityCompat.START);
    }else{
        super.onBackPressed();
    }
}
```

Se a Drawer estiver aberta, então dizemos a aplicação para a fechar

Leitura Adicional

- AppBar:

<https://developer.android.com/training/appbar>

- Navigation Drawer:

<https://developer.android.com/guide/navigation/navigation-ui>

Programação Para Dispositivos Móveis I

AppBar, Toolbar e NAVIGATION DRAWERS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa, rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos jrmr@estg.ipp.pt e Fábio Silva fas@estg.ipp.pt