

Interface de Usuário

Bruno Lopes Alcantara Batista

Prazer em conhecer,

Bruno Lopes A Batista

Me. Ciência da Computação - UECE

Esp. em Desenvolvimento Web com JavaEE - FJN

Bel. em Sistemas de Informação - FJN

Professor Pesquisador - LARCES

Gerente de Projetos - LARCES

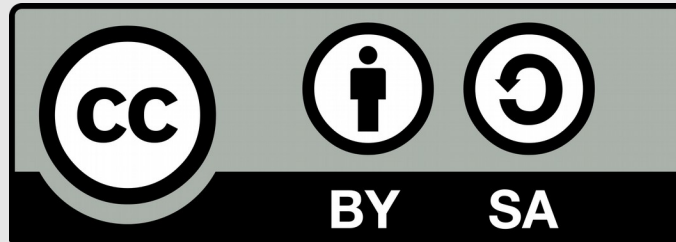
Gerente de Projetos - Athomustec

bruno@larces.uece.br

<http://www.larces.uece.br/bruno>



Copyleft



Esta obra está licenciado com uma Licença
Creative Commons Atribuição-CompartilhaIgual 4.0 Internacional

Agenda

- ☐ Introdução
- ☐ View e ViewGroup
- ☐ Unidades de Medida
- ☐ Gerenciadores de Layout
- ☐ Toolbar
- ☐ Let's try it!
- ☐ RecyclerView e CardView
- ☐ Let's try it!
- ☐ Navigation Drawer
- ☐ Let's try it!
- ☐ Trabalho Opcional #1



Introdução




Introdução

- ❑ Apendemos até o momento que a Activity é a unidade básica do Android.
- ❑ Uma Activity exibe a interface da sua aplicação, que pode conter widgets.
- ❑ Um widget pode ser um botão, caixas de textos, labels e etc.
- ❑ Tipicamente a interface de usuário é definida via XML.

Introdução

```
1  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:paddingBottom="16dp"
6      android:paddingLeft="16dp"
7      android:paddingRight="16dp"
8      android:paddingTop="16dp"
9      tools:context=".MainActivity">
10
11      <TextView
12          android:layout_width="wrap_content"
13          android:layout_height="wrap_content"
14          android:text="@string/hello_world" />
15
16  </RelativeLayout>
17
```

Introdução

```
14  public class NewActivity extends Activity {  
15  
16     private MyBroadcastReceiver myBroadcastReceiver;  
17  
18     @Override  
19   protected void onCreate(Bundle savedInstanceState) {  
20         super.onCreate(savedInstanceState);  
21         setContentView(R.layout.activity_new);  
22     }  
23  
24 }  
25
```


Introdução

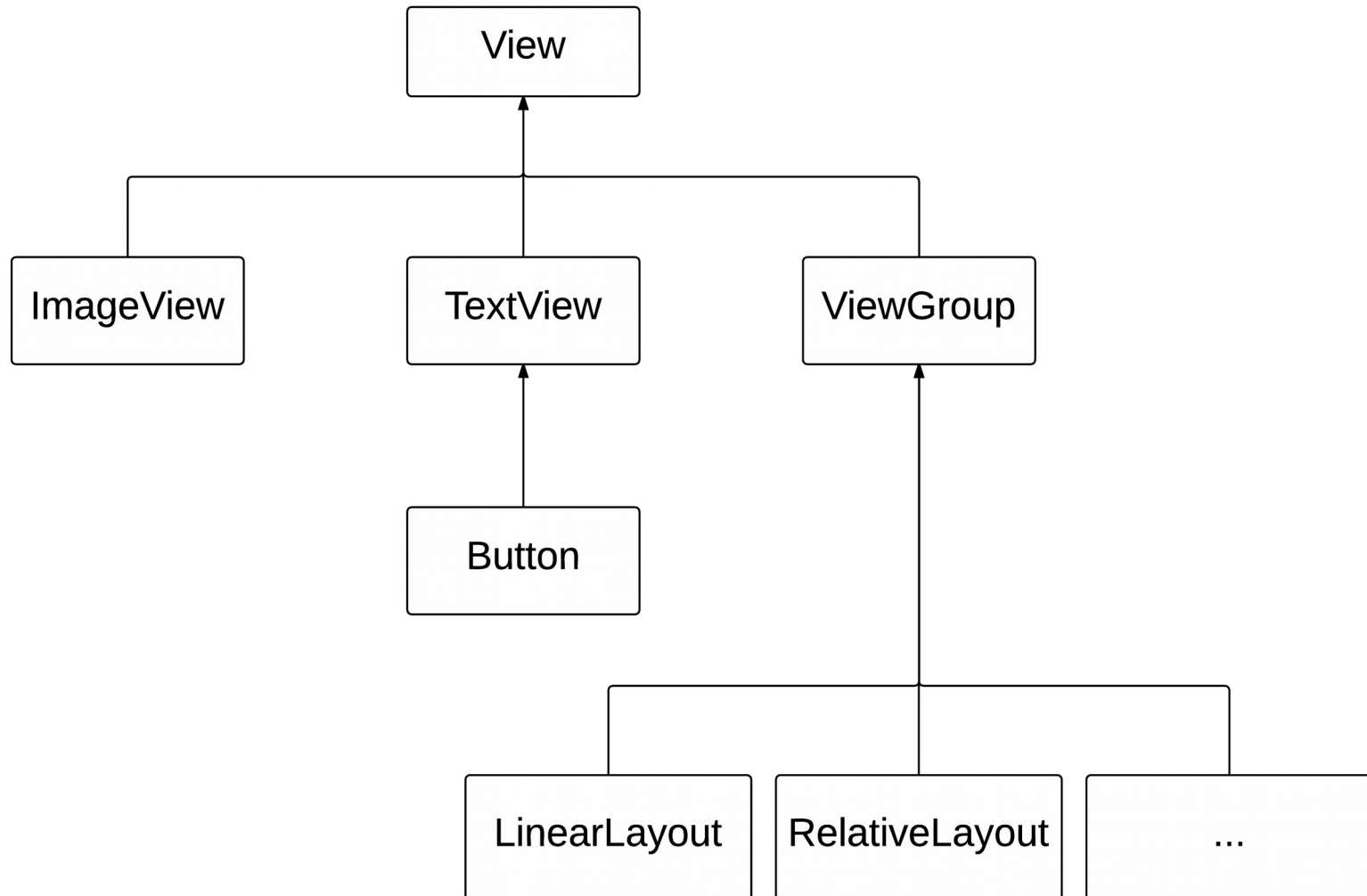
- ❑ Durante a compilação, cada elemento declarado via XML é compilado na classe Java equivalente.
- ❑ O sistema Android cria a interface de usuário da Activity quando a mesma é carregada.

View e ViewGroup

View e ViewGroup

- ❑ Uma Activity contem View e ViewGroup.
- ❑ Uma View é um widget que possui uma aparência na tela, ex:
 - ❑ Botões;
 - ❑ Labels;
 - ❑ Caixas de textos.
- ❑ Uma ou mais Views podem ser agrupadas dentro de um ViewGroup.
- ❑ Um ViewGroup descreve como as Views serão dispostas na tela.
- ❑ ViewGroups também podem ser chamados de gerenciadores de layout.

View e ViewGroup



Unidades de Medida

Unidades de Medida

- Quando especificamos o tamanho de um elemento em uma interface de usuário, deve-se estar atento a unidade de medida utilizada:

- dp – Density Independent Pixels:

1 dp é equivalente a um pixel em uma tela de 160 dpi.
Esta é a unidade de medida recomendada para especificar dimensões no layout.

- sp – Scale Independent Pixels:

Similar ao dp e é recomendada para especificar tamanho de fontes

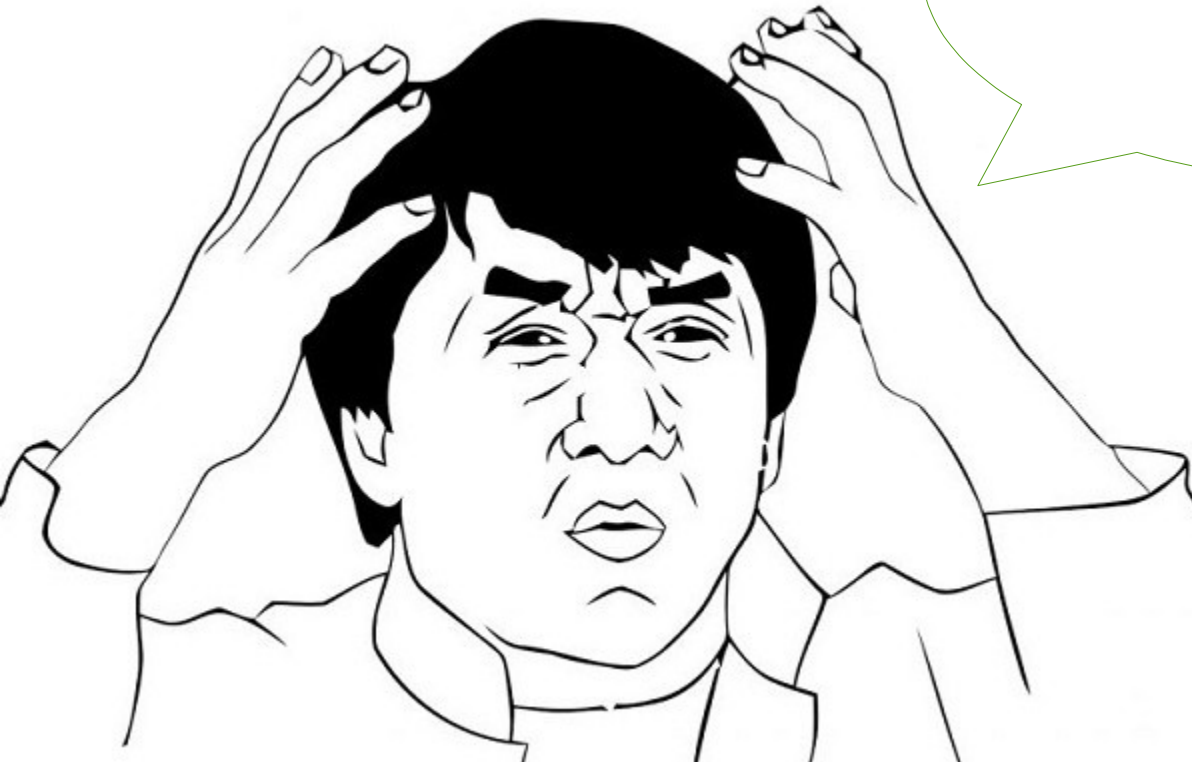
- pt – Point:

É definido como 1/72 de uma polegada.

- px – Pixel:

Corresponde aos pixels de uma tela.

Unidades de Medida



O que significa
“uma tela de
160dpi”?

Unidades de Medida

- Vamos responder a essa pergunta analisando os seguintes dispositivos



LG Nexus 5

Tela: 4,95"

Resolução: 1080x1920

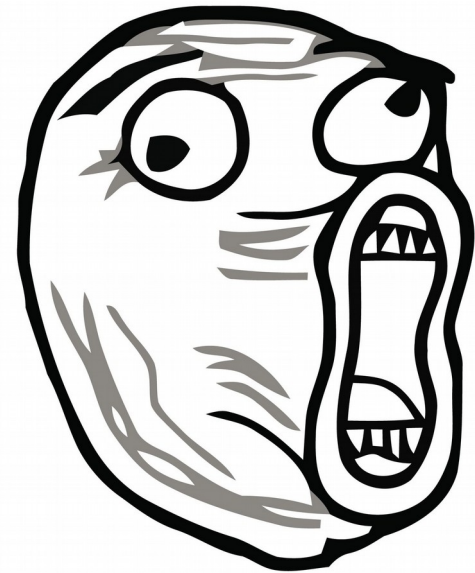


Samsung Galaxy S

Tela: 4"

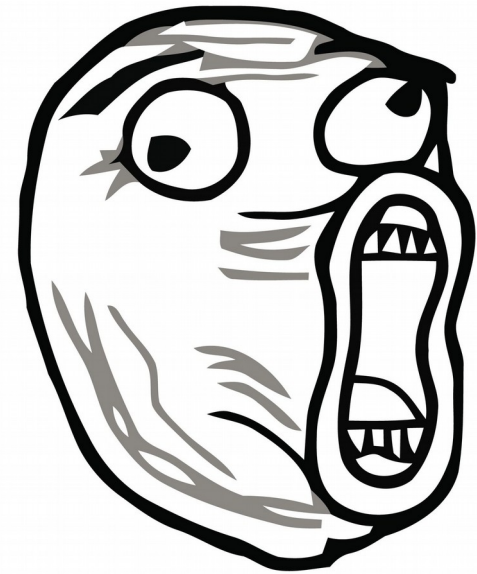
Resolução: 480x800

Unidades de Medida



$$\frac{\sqrt{(1920^2 + 1080^2)}}{4,95} = 445,031751531 \approx 445 \text{ dpi}$$

Unidades de Medida



$$\frac{\sqrt{(800^2 + 480^2)}}{4} = 233,238075794 \approx 233 \text{ dpi}$$

Unidades de Medida

- ❑ Podemos concluir que a densidade de pixels varia de acordo com a resolução e o tamanho da tela.
- ❑ O Android define reconhece as seguintes densidades:
 - ❑ ldpi (Low) ~ 120dpi
 - ❑ mdpi (Medium) ~ 160dpi
 - ❑ hdpi (High) ~ 240dpi
 - ❑ xdpi (Extra High) ~ 320dpi
 - ❑ xxdpi (Extra Extra High) ~ 480dpi
 - ❑ xxxdpi (Extra Extra Extra High) ~ 640dpi



Layout

Layout

- ❑ Um layout define a estrutura visual para a interface de usuário.
- ❑ Podemos declarar um layout de duas maneiras:
 - ❑ Declarando elementos via XML
 - ❑ Instanciando objetos de layout em tempo de execução.
- ❑ O Android permite utilizar um mix de ambas as formas.
- ❑ Podemos criar diferentes layouts para diferentes tamanhos de telas em uma mesma aplicação.

Layout

Escrevendo o XML

Layout

Escrevendo o XML

- ❑ Utilizando o vocabulário de XML do android, podemos criar layout rapidamente.
- ❑ Cada arquivo de layout deve conter apenas um único nó raiz.
- ❑ Pode-se adicionar outros elementos de layout como elementos filhos do elemento raiz.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Layout

Carregando o XML de Layout

Layout

Carregando o XML de Layout

- ❑ Quando compilamos a aplicação, cada arquivo de XML é compilado para um recurso do tipo View.
- ❑ Deve-se carregar os recursos de layout para o código fonte da aplicação.
- ❑ Realizamos tal tarefa utilizando o método setContentView()

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main_layout);  
}
```

Layout

□ Mais informações sobre layouts:

`http://developer.android.com/guide/topics/ui/declaring-layout.html#CommonLayouts`

Layout

Gerenciadores de Layout

Layout

Gerenciadores de Layout

- Cada subclasse de ViewGroup fornece uma maneira única para mostrar os componentes.

Linear Layout



A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

Relative Layout



Enables you to specify the location of child objects relative to each other (child A to the left of child B) or to the parent (aligned to the top of the parent).

Web View



Displays web pages.

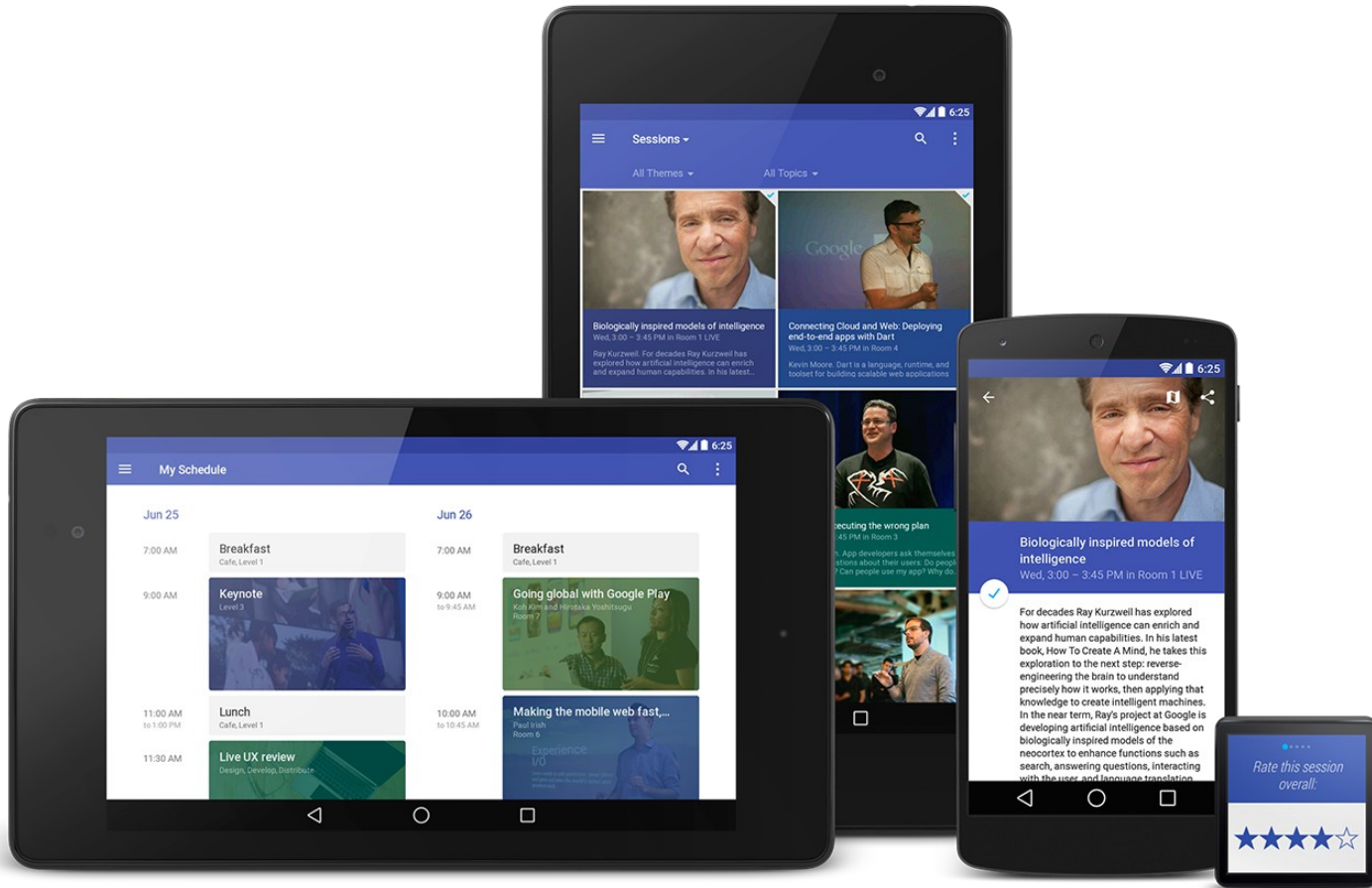


Toolbar

Toolbar

- ❑ Componente adicionado no Android Lollipop.
- ❑ Substitui a antiga ActionBar.
- ❑ Maior flexibilidade na customização.
- ❑ Estende de ViewGroup, ou seja, pode-se adicionar outros componentes.
- ❑ Utilizaremos a versão da biblioteca de retrocompatibilidade em nossos exemplos.

Toolbar



Toolbar

- ❑ Adicionar a dependência de retrocompatibilidade no arquivo build.gradle

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 22
5      buildToolsVersion "22.0.1"
6
7      defaultConfig {
8          applicationId "br.edu.fa7.uiexample"
9          minSdkVersion 10
10         targetSdkVersion 22
11         versionCode 1
12         versionName "1.0"
13     }
14     buildTypes {
15         release {
16             minifyEnabled false
17             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18         }
19     }
20 }
21
22 dependencies {
23     compile fileTree(dir: 'libs', include: ['*.jar'])
24     compile "com.android.support:appcompat-v7:22.2.1"
25 }
26
27
```


Toolbar

- ❑ Criar o arquivo colors.xml dentro de res/values com o seguinte conteúdo:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="colorPrimary">#2196F3</color>
4   <color name="colorPrimaryDark">#1976D2</color>
5   <color name="colorAccent">#03A9F4</color>
6   <color name="colorControlNormal">@android:color/white</color>
7 </resources>
8
```

Toolbar

- ❑ Alterar o arquivo styles.xml dentro de res/values com o seguinte conteúdo:

```
1 <resources>
2
3     <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
4         <item name="colorPrimary">@color/colorPrimary</item>
5         <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
6         <item name="colorAccent">@color/colorAccent</item>
7         <item name="colorControlNormal">@color/colorControlNormal</item>
8     </style>
9
10    <style name="ToolbarTheme" parent="AppTheme">
11        <item name="android:textColorPrimary">@android:color/white</item>
12        <item name="android:textColorSecondary">@android:color/white</item>
13    </style>
14
15 </resources>
16
```

Toolbar

- Adicionar no arquivo de layout a Toolbar com o seguinte conteúdo.

```
1  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context=".MainActivity">
7
8      <android.support.v7.widget.Toolbar
9          android:id="@+id/main_toolbar"
10         android:elevation="4dp"
11         android:layout_width="match_parent"
12         android:layout_height="?attr/actionBarSize"
13         android:minHeight="?attr/actionBarSize"
14         android:background="?attr/colorPrimary"
15         app:theme="@style/ToolbarTheme"
16         app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
17
18     <TextView
19         android:layout_below="@+id/main_toolbar"
20         android:layout_width="wrap_content"
21         android:layout_height="wrap_content"
22         android:text="Hello world!" />
23
24 </RelativeLayout>
25
```

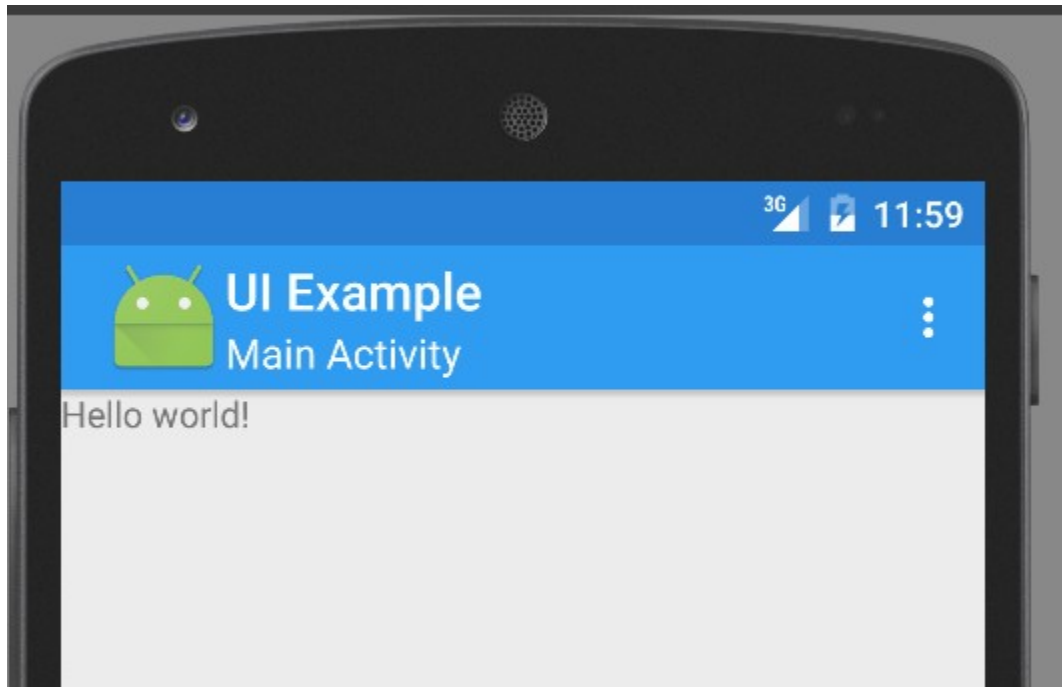
Toolbar

- ❑ Modificar o arquivo MainActivity.java com o seguinte conteúdo

```
1 package br.edu.fa7.uiexample;
2
3 import ...
4
5
6
7
8
9
10
11
12 public class MainActivity extends AppCompatActivity {
13
14     private Toolbar mMainToolbar;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         mMainToolbar = (Toolbar) findViewById(R.id.main_toolbar);
23         mMainToolbar.setTitle("UI Example");
24         mMainToolbar.setSubtitle("Main Activity");
25         mMainToolbar.setLogo(R.mipmap.ic_launcher);
26         setSupportActionBar(mMainToolbar);
27
28     }
29
30     @Override
31     public boolean onCreateOptionsMenu(Menu menu) {
32         // Inflate the menu; this adds items to the action bar if it
```

Toolbar

- Ao executar o resultado será o seguinte



Toolbar

□ Maiores detalhes em:

`http://android-developers.blogspot.com.br/2014/10/appcompat-v21-material-design-for-pre.html`


`http://android-developers.blogspot.com.br/2014/10/implementing-material-design-in-your.html`



Let's try it

Let's try it!

- ☐ Criar o projeto **Demo App**
- ☐ Criar duas Activities: MainActivity e SettingsActivity
- ☐ Implementar a Toolbar em ambas a Activities

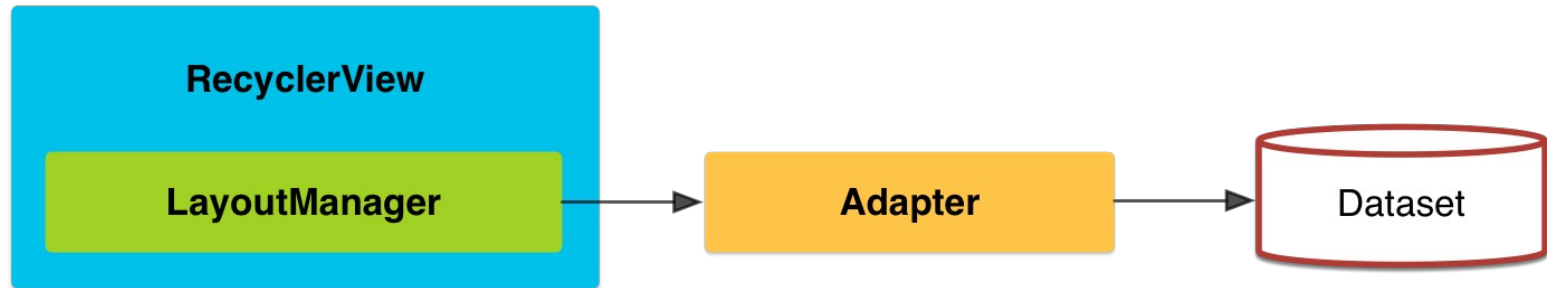


RecyclerView e CardView

RecyclerView e CardView

- ❑ O RecyclerView é uma versão avançada e flexível do ListView.
- ❑ Este componente é um container para exibição de grandes conjuntos de dados.
- ❑ Implementa o scroll eficiente.
- ❑ O RecyclerView simplifica a exibição e a manipulação de grandes conjuntos de dados fornecendo:
 - ❑ Gerenciadores de layout para o posicionamento dos itens.
 - ❑ Animações padrão para operações comuns dos itens, tais como remoção e adição de um item.

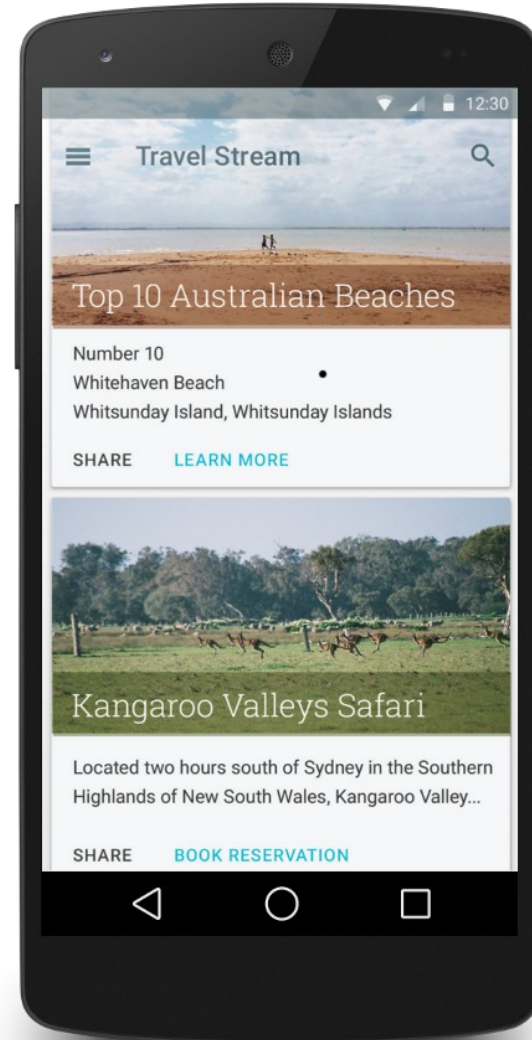
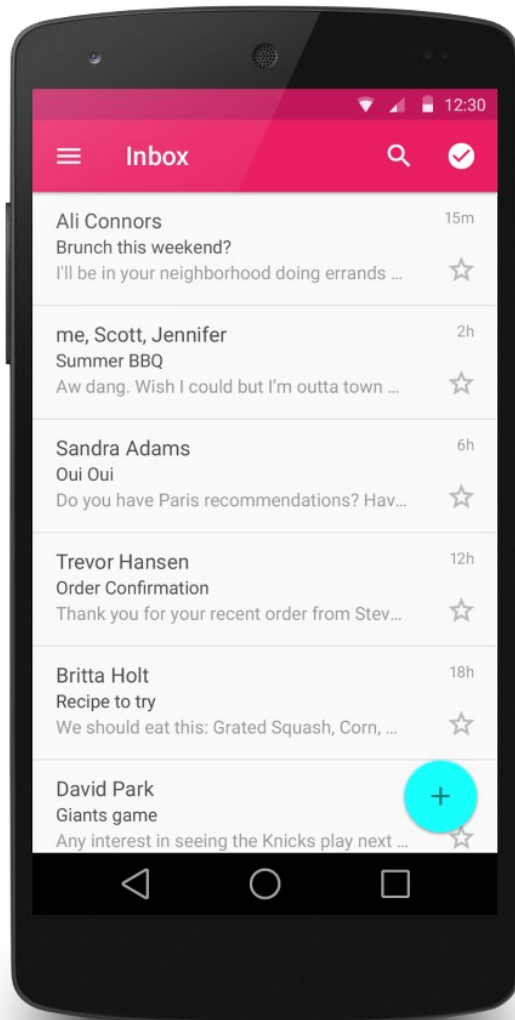
RecyclerView e CardView



RecyclerView e CardView

- ❑ O CardView estende o gerenciador de layout `FrameLayout`, permitindo exibir um visual consistente.
- ❑ Podem possuir sombras e arestas arredondadas.
- ❑ Foi adicionado no Android Lollipop, mas está presente na API de retrocompatibilidade.

RecyclerView e CardView



RecyclerView e CardView

- ❑ Para utilizar o RecyclerView e o CardView, adicione as seguintes dependências:

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 22
5      buildToolsVersion "22.0.1"
6
7      defaultConfig {
8          applicationId "br.edu.fa7.uiexample"
9          minSdkVersion 10
10         targetSdkVersion 22
11         versionCode 1
12         versionName "1.0"
13     }
14     buildTypes {
15         release {
16             minifyEnabled false
17             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18         }
19     }
20 }
21
22 dependencies {
23     compile fileTree(dir: 'libs', include: ['*.jar'])
24     compile "com.android.support:appcompat-v7:22.2.1"
25     compile "com.android.support:recyclerview-v7:22.2.1"
26     compile "com.android.support:cardview-v7:22.2.1"
27 }
28
```

RecyclerView e CardView

- ❑ Criar o arquivo de layout contendo o CardView, e.g. **cardview_item.xml**:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.v7.widget.CardView
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:card_view="http://schemas.android.com/apk/res-auto"
5     android:id="@+id/card_view"
6     android:layout_width="match_parent"
7     android:layout_height="wrap_content"
8     android:layout_gravity="center"
9     android:layout_marginTop="5dp"
10    android:layout_marginBottom="5dp"
11    android:layout_marginLeft="16dp"
12    android:layout_marginRight="16dp"
13    card_view:cardCornerRadius="4dp"
14    android:elevation="4dp">
15
16    <RelativeLayout
17        android:layout_width="match_parent"
18        android:layout_height="match_parent">
19
20
21        <ImageView
22            android:layout_width="wrap_content"
23            android:layout_height="wrap_content"
24            android:id="@+id/imageView"
25            android:layout_alignParentTop="true"
26            android:layout_alignParentLeft="true"
27            android:layout_alignParentStart="true"
28            android:src="@mipmap/ic_launcher" />
29
30    ...
```

```
29
30
31    <TextView
32        android:layout_width="wrap_content"
33        android:layout_height="match_parent"
34        android:text="New Text"
35        android:id="@+id/textView"
36        android:layout_alignBottom="@+id/imageView"
37        android:layout_toRightOf="@+id/imageView"
38        android:layout_toEndOf="@+id/imageView"
39        android:layout_marginLeft="30dp"
40        android:layout_marginStart="30dp"
41        android:textSize="30sp"
42        android:layout_alignParentTop="true"
43        android:gravity="center" />
44
45    </RelativeLayout>
46
47 </android.support.v7.widget.CardView>
```

RecyclerView e CardView

- ❑ Criar o arquivo de layout contendo o RecyclerView, e.g. **activity_main.xml**:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout 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:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8     <android.support.v7.widget.Toolbar
9         android:id="@+id/main_toolbar"
10        android:layout_width="match_parent"
11        android:layout_height="?attr/actionBarSize"
12        android:background="?attr/colorPrimary"
13        android:elevation="4dp"
14        android:minHeight="?attr/actionBarSize"
15        app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
16        app:theme="@style/ToolbarTheme" />
17
18    <android.support.v7.widget.RecyclerView
19        android:id="@+id/main_recycler_view"
20        android:layout_width="match_parent"
21        android:layout_height="match_parent"
22        android:layout_marginTop="10dp"
23        android:layout_marginBottom="10dp"
24        android:scrollbars="vertical"
25        android:layout_below="@+id/main_toolbar"
26        android:layout_alignParentLeft="true"
27        android:layout_alignParentStart="true" />
28
29 </RelativeLayout>
```


RecyclerView e CardView

- ❑ Criar o adapter customizado para o RecyclerView:

```
13 public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {
14
15     private List<Person> people;
16     private LayoutInflater mLayoutInflater;
17
18     public MyAdapter(Context context, List<Person> people) {
19         this.people = people;
20         this.mLayoutInflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
21     }
22
23     @Override
24     public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
25         View v = mLayoutInflater.inflate(R.layout.cardview_item, parent, false);
26         MyViewHolder myViewHolder = new MyViewHolder(v);
27         return myViewHolder;
28     }
29
30     @Override
31     public void onBindViewHolder(MyViewHolder holder, int position) {
32         Person p = people.get(position);
33         holder.imageView.setImageResource(p.getImage());
34         holder.textView.setText(p.getName());
35     }
```














RecyclerView e CardView

- ❑ Criar o adapter customizado para o RecyclerView (continuação):

```
36
37
38  @Override
39  public int getItemCount() {
40      return people.size();
41  }
42
43  public class MyViewHolder extends RecyclerView.ViewHolder{
44
45      public ImageView imageView;
46      public TextView textView;
47
48      public MyViewHolder(View itemView) {
49          super(itemView);
50          imageView = (ImageView) itemView.findViewById(R.id.imageView);
51          textView = (TextView) itemView.findViewById(R.id.textView);
52      }
53  }
54
55 }
```

RecyclerView e CardView

- ❑ Recuperar a instancia do RecyclerView e adicionar o adapter e a lista, e.g MainActivity:

```
34
35
36  people.add(new Person(R.mipmap.ic_launcher, "Joao"));
37  people.add(new Person(R.mipmap.ic_launcher, "Maria"));
38  people.add(new Person(R.mipmap.ic_launcher, "José"));
39  people.add(new Person(R.mipmap.ic_launcher, "Francisca"));
40  people.add(new Person(R.mipmap.ic_launcher, "Guilhermina"));
41  people.add(new Person(R.mipmap.ic_launcher, "Joana"));
42  people.add(new Person(R.mipmap.ic_launcher, "Eurico"));
43  people.add(new Person(R.mipmap.ic_launcher, "Carlos"));
44  people.add(new Person(R.mipmap.ic_launcher, "Eugênio"));
45  people.add(new Person(R.mipmap.ic_launcher, "Marcos"));
46  people.add(new Person(R.mipmap.ic_launcher, "Verônica"));
47  people.add(new Person(R.mipmap.ic_launcher, "Marta"));
48  people.add(new Person(R.mipmap.ic_launcher, "Hortência"));
49
50
51
52
53
54
55
56
```

```
List<Person> people = new ArrayList<>();
people.add(new Person(R.mipmap.ic_launcher, "Joao"));
people.add(new Person(R.mipmap.ic_launcher, "Maria"));
people.add(new Person(R.mipmap.ic_launcher, "José"));
people.add(new Person(R.mipmap.ic_launcher, "Francisca"));
people.add(new Person(R.mipmap.ic_launcher, "Guilhermina"));
people.add(new Person(R.mipmap.ic_launcher, "Joana"));
people.add(new Person(R.mipmap.ic_launcher, "Eurico"));
people.add(new Person(R.mipmap.ic_launcher, "Carlos"));
people.add(new Person(R.mipmap.ic_launcher, "Eugênio"));
people.add(new Person(R.mipmap.ic_launcher, "Marcos"));
people.add(new Person(R.mipmap.ic_launcher, "Verônica"));
people.add(new Person(R.mipmap.ic_launcher, "Marta"));
people.add(new Person(R.mipmap.ic_launcher, "Hortência"));

LinearLayoutManager llm = new LinearLayoutManager(this);
MyAdapter adapter = new MyAdapter(this, people);
mRecyclerView = (RecyclerView) findViewById(R.id.main_recycler_view);
mRecyclerView.setHasFixedSize(true);
mRecyclerView.setLayoutManager(llm);
mRecyclerView.setAdapter(adapter);
```



Let's try it

Let's try it!

- Adicionar o RecyclerView e o CardView no projeto **Demo App** criado anteriormente.

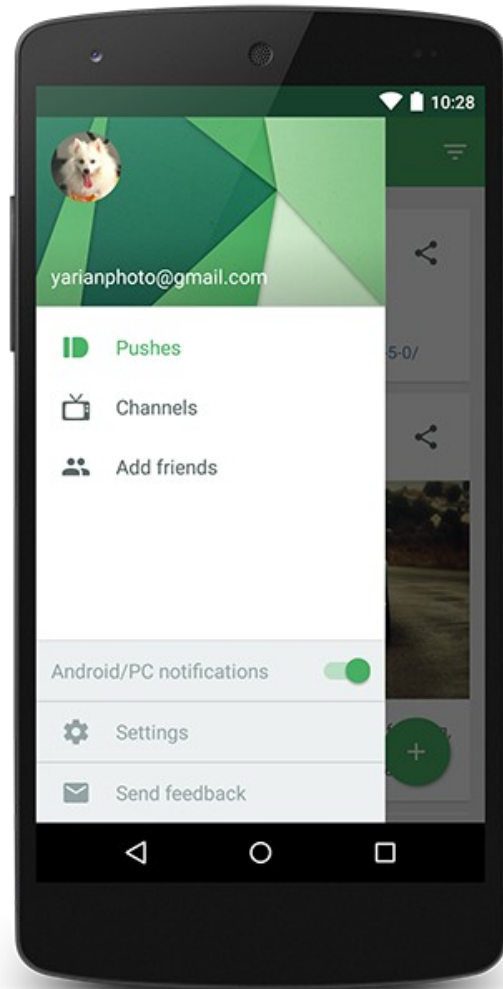


Naviagtion Drawer

Navigation Drawer

- ❑ É um painel que mostra o painel principal de navegação do aplicativo.
- ❑ Fica na margem esquerda da tela.
- ❑ O mesmo fica escondido na maior parte do tempo.
- ❑ Entretanto é revelado quando o usuário faz um swipe da esquerda para direita ou quando toca no ícone da aplicação na action bar.

Navigation Drawer



Navigation Drawer

- Para utilizar o Navigation Drawer, devemos utilizar o DrawerLayout em nosso arquivo de layout:

```
1 <android.support.v4.widget.DrawerLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:id="@+id/drawer_layout"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent">
6
7   <!-- The main content view -->
8   <FrameLayout
9     android:id="@+id/content_frame"
10    android:layout_width="match_parent"
11    android:layout_height="match_parent" />
12
13   <!-- The navigation drawer -->
14   <LinearLayout
15     android:layout_width="240dp"
16     android:layout_height="match_parent">
17
18     <ListView
19       android:id="@+id/left_drawer"
20       android:layout_width="match_parent"
21       android:layout_height="match_parent"
22       android:layout_gravity="start"
23       android:background="#111"
24       android:choiceMode="singleChoice"
25       android:divider="@android:color/transparent"
26       android:dividerHeight="0dp" />
27
28   </LinearLayout>
29
30 </android.support.v4.widget.DrawerLayout>
31
```

Navigation Drawer

- E na Activity devemos adicionar o seguinte código:


```
27 protected void onCreate(Bundle savedInstanceState) {  
28  
29     super.onCreate(savedInstanceState);  
30     setContentView(R.layout.activity_main_drawer);  
31  
32     List<String> list = new ArrayList<>();  
33     list.add("Item 1");  
34     list.add("Item 2");  
35     list.add("Item 3");  
36     list.add("Item 4");  
37     list.add("Item 5");  
38  
39     mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);  
40     mDrawerList = (ListView) findViewById(R.id.left_drawer);  
41  
42     mDrawerList.setAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, list));  
43  
44     mMainToolbar = (Toolbar) findViewById(R.id.main_toolbar);  
45     mMainToolbar.setTitle("UI Example");  
46     mMainToolbar.setSubtitle("Main Activity");  
47     mMainToolbar.setLogo(R.mipmap.ic_launcher);  
48     setSupportActionBar(mMainToolbar);  
49  
50     MainFragment mainFragment = new MainFragment();  
51  
52     mFragmentManager = getSupportFragmentManager();  
53     FragmentTransaction fragmentTransaction = mFragmentManager.beginTransaction();  
54     fragmentTransaction.replace(R.id.content_frame, mainFragment);  
55     fragmentTransaction.commit();  
56  
57  
58 }
```



Let's try it

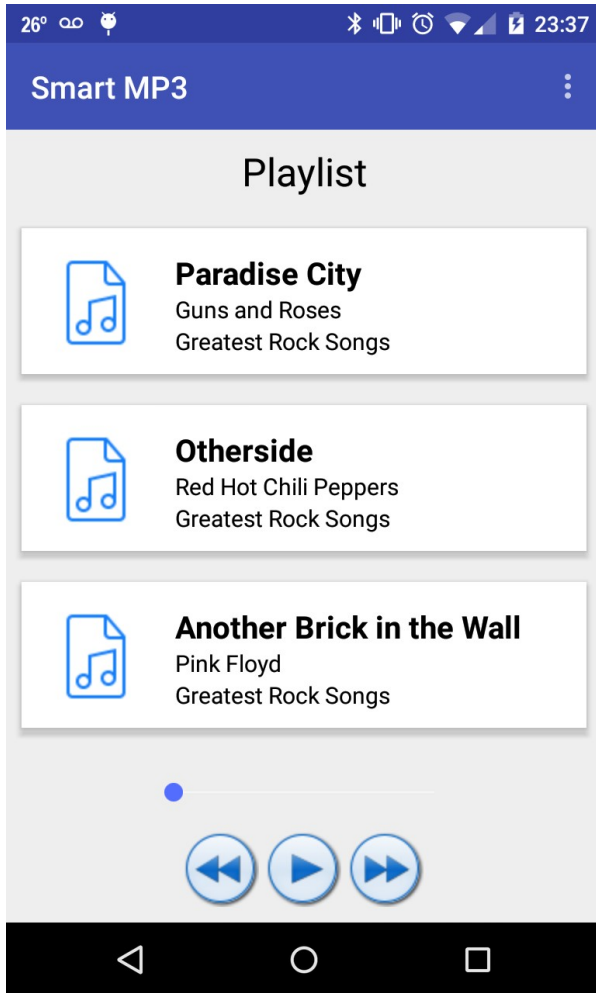
Let's try it!

- ☐ Adicionar o Navigation Drawer no projeto **Demo App** criado anteriormente.



Trabalho Opcional #1

Trabalho Opcional #1



- ☐ Criar um App Android com o layout ao lado
- ☐ Não é necessário implementar nenhuma funcionalidade
- ☐ Data de entrega: 31/08/15
- ☐ Dica: Use o Relative Layout.

Obrigado!

