

Introdução aos dispositivos móveis

Interfaces e seus componentes

Marcelo Zamith

e-mail: mzamith@ufrj.br
<https://www.dcc.ufrj.br/~marcelo/>

Universidade Federal Rural do Rio de Janeiro - DCC



Introdução

Introdução

- Layouts:
 - ▶ define a **interface** do usuário.

Introdução

- Layouts:
 - ▶ define a **interface** do usuário.
- O que é interface ?

Introdução

- Layouts:
 - ▶ define a **interface** do usuário.
- O que é interface ?
 - ▶ interação do usuário com o sistema;
 - ▶ contém os componentes e controles;
 - ▶ é a **camada de comunicação com usuário**; ←

Introdução

- Layouts:
 - ▶ define a **interface** do usuário.
- O que é interface ?
 - ▶ interação do usuário com o sistema;
 - ▶ contém os componentes e controles;
 - ▶ **é a camada de comunicação com usuário;** ←

Observação

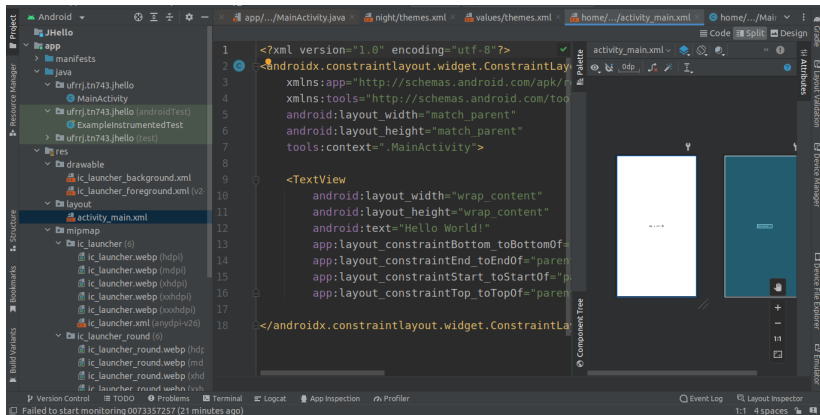
Interfaces do sistema: permite a troca de informações entre sistemas!

Introdução

- interface mal feita, fracasso do app;
- fluidez na interação;
- dados e ações claras para o usuário (*affordance*);
- alguns exemplos:
 - ▶ usuário não sabe o que fazer ou não acha a opção desejada;
 - ▶ o app não implementa a função ou induz usuário ao erro;
 - ▶ o app trava ou fecha “sem explicação”;
 - ▶ muda a interface da versão mais nova;

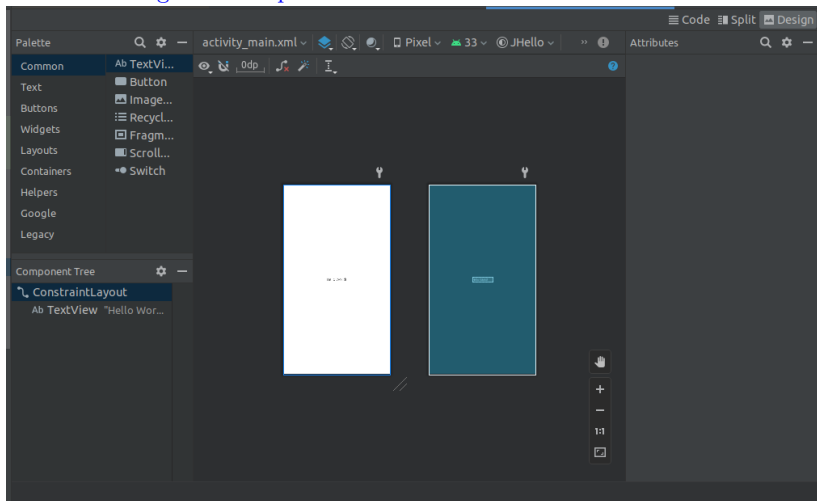
Layout & interface do usuário

- Layout tem três componentes:
 - o código dos eventos
 - os componentes de interface (WYSIWYG)
 - código dos componentes de interface



Layout & interface do usuário

- Layout tem três componentes:
 - ▶ o código dos eventos
 - ▶ os componentes de interface (WYSIWYG)
 - ▶ código dos componentes de interface



Layout & interface do usuário

- Interface com usuário:
 - ▶ viabiliza a navegação através de componentes e define o que pode ser feito pelo usuário;
 - ▶ define como as entradas devem ser feitas;
 - ▶ define as saídas para o usuário;
- GUI - *graphical user interface*: paradigma mais natural de interação com o sistema.

Layout & interface do usuário

- Interface com usuário:
 - ▶ layouts;
 - ▶ atenção do usuário ao conteúdo;
 - ▶ estética;
 - ▶ experiência do usuário;
 - ▶ consistência;
- Desafio de programar para dispositivos móveis:
 - ▶ trabalhar com uma tela reduzida
 - 5 polegadas o tamanho médio ¹

¹ <https://www.tudocelular.com/android/noticias/n119931/tamanho-tela-mais-popular-mundo.html>

Interface

Layouts

- View / ViewGroup
 - ▶ FrameLayout
 - ▶ LinearLayout
 - ▶ TableLayout
 - ▶ RelativeLayout
 - ▶ AbsoluteLayout

Interface

Layouts

- define como os componentes são organizados no tela;
- organizam de diferentes formas:
 - ▶ horizontal
 - ▶ vertical
 - ▶ de maneira mais desejada :)
- organizam os componentes independente da resolução do aparelho

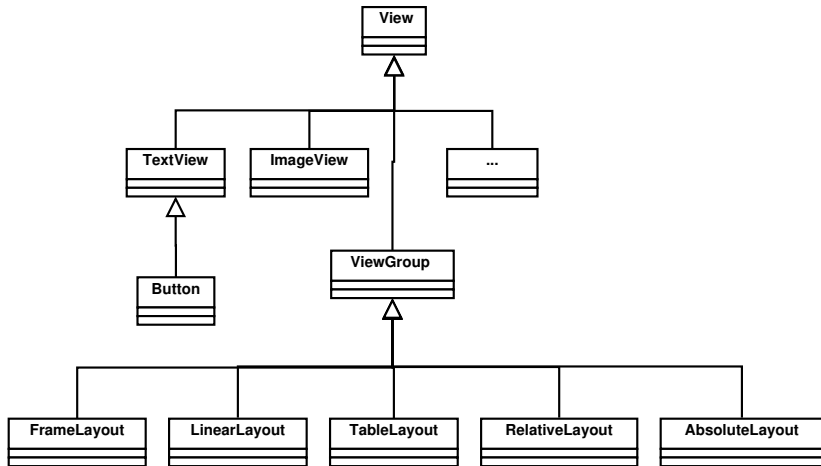
Interface

Layouts

- define como os componentes são organizados no tela;
- organizam de diferentes formas:
 - ▶ horizontal
 - ▶ vertical
 - ▶ de maneira mais desejada :)
- organizam os componentes independente da resolução do aparelho

Interface

Layouts



Interface

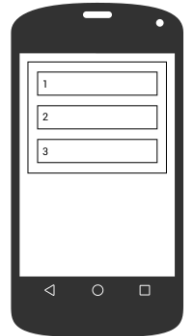
Layouts

- Listeners: captura a interação do usuário;
- Métodos callback: método responsável pela captura e tratamento do evento;
- View: renderiza o componente dia `onDraw (Canvas)`
- distinção:
 - ▶ layouts: como organiza a interface;
 - ▶ widgets: componentes de interfaces que permitem a interação com usuário (UI).

Interface

Layouts - LinearLayout

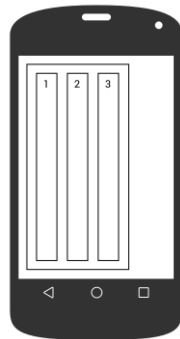
```
1 <LinearLayout
2   android:layout_width="match_parent"
3   android:layout_height="wrap_content"
4   android:orientation="vertical">
5   <TextView
6     android:layout_width="match_parent"
7     android:layout_height="wrap_content"
8     android:text="1"/>
9   <TextView
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:text="2"/>
13   <TextView
14    android:layout_width="match_parent"
15    android:layout_height="wrap_content"
16    android:text="3"/>
17 </LinearLayout>
```



Interface

Layouts - LinearLayout

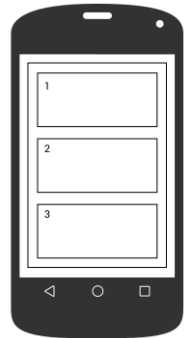
```
1 <LinearLayout
2     android:layout_width="wrap_content"
3     android:layout_height="match_parent"
4     android:orientation="horizontal">
5     <TextView
6         android:layout_width="wrap_content"
7         android:layout_height="match_parent"
8         android:text="1"/>
9     <TextView
10        android:layout_width="wrap_content"
11        android:layout_height="match_parent"
12        android:text="2"/>
13    <TextView
14        android:layout_width="wrap_content"
15        android:layout_height="match_parent"
16        android:text="3"/>
17 </LinearLayout>
```



Interface

Layouts - LinearLayout

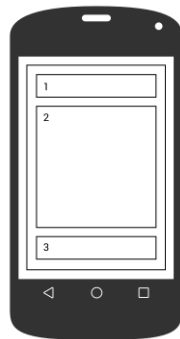
```
1  <LinearLayout
2      android:layout_width="match_parent"
3      android:layout_height="match_parent"
4      android:orientation="vertical">
5      <TextView
6          android:layout_width="match_parent"
7          android:layout_height="0dp"
8          android:layout_weight="1"
9          android:text="1" />
10     <TextView
11         android:layout_width="match_parent"
12         android:layout_height="0dp"
13         android:layout_weight="1"
14         android:text="2" />
15     <TextView
16         android:layout_width="match_parent"
17         android:layout_height="0dp"
18         android:layout_weight="1"
19         android:text="3" />
20 </LinearLayout>
```



Interface

Layouts - LinearLayout

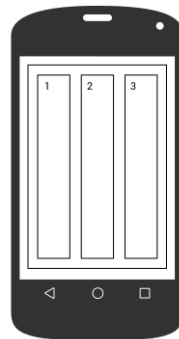
```
1 <LinearLayout
2     android:layout_width="match_parent"
3     android:layout_height="match_parent"
4     android:orientation="vertical">
5     <TextView
6         android:layout_width="match_parent"
7         android:layout_height="wrap_content"
8         android:layout_weight="0"
9         android:text="1" />
10    <TextView
11        android:layout_width="match_parent"
12        android:layout_height="0dp"
13        android:layout_weight="1"
14        android:text="2" />
15    <TextView
16        android:layout_width="match_parent"
17        android:layout_height="wrap_content"
18        android:layout_weight="0"
19        android:text="3" />
20 </LinearLayout>
```



Interface

Layouts - LinearLayout

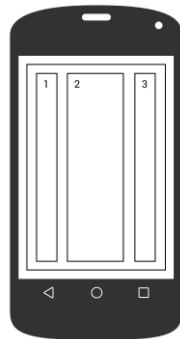
```
1 <LinearLayout
2     android:layout_width="match_parent"
3     android:layout_height="match_parent"
4     android:orientation="horizontal">
5     <TextView
6         android:layout_width="0dp"
7         android:layout_height="match_parent"
8         android:layout_weight="1"
9         android:text="1"/>
10    <TextView
11        android:layout_width="0dp"
12        android:layout_height="match_parent"
13        android:layout_weight="1"
14        android:text="2"/>
15    <TextView
16        android:layout_width="0dp"
17        android:layout_height="match_parent"
18        android:layout_weight="1"
19        android:text="3"/>
20 </LinearLayout>
```



Interface

Layouts - LinearLayout

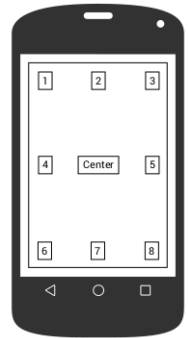
```
1  <LinearLayout
2      android:layout_width="match_parent"
3      android:layout_height="match_parent"
4      android:orientation="horizontal">
5      <TextView
6          android:layout_width="wrap_content"
7          android:layout_height="match_parent"
8          android:layout_weight="0"
9          android:text="1" />
10     <TextView
11         android:layout_width="0dp"
12         android:layout_height="match_parent"
13         android:layout_weight="1"
14         android:text="2" />
15     <TextView
16         android:layout_width="wrap_content"
17         android:layout_height="match_parent"
18         android:layout_weight="0"
19         android:text="3" />
20 </LinearLayout>
```



Interface

Layouts - RelativeLayout

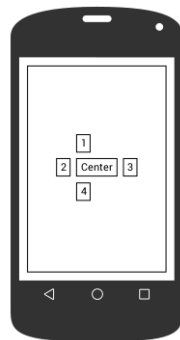
```
1 <RelativeLayout
2     android:layout_width="match_parent"
3     android:layout_height="match_parent">
4     <TextView
5         android:layout_width="wrap_content"
6         android:layout_height="wrap_content"
7         android:layout_alignParentTop="true"
8         android:layout_alignParentLeft="true"
9         android:text="1"/>
10    <TextView
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:layout_alignParentTop="true"
14        android:layout_centerHorizontal="true"
15        android:text="2"/>
16    <TextView
17        android:layout_width="wrap_content"
18        android:layout_height="wrap_content"
19        android:layout_alignParentTop="true"
20        android:layout_alignParentRight="true"
21        android:text="3"/>
22    <TextView
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_centerVertical="true"
26        android:layout_alignParentLeft="true"
27        android:text="4"/>
28    <TextView
29        android:layout_width="wrap_content"
30        android:layout_height="wrap_content"
31        android:layout_centerInParent="true"
32        android:text="Center"/>
33    ...
```



Interface

Layouts - RelativeLayout

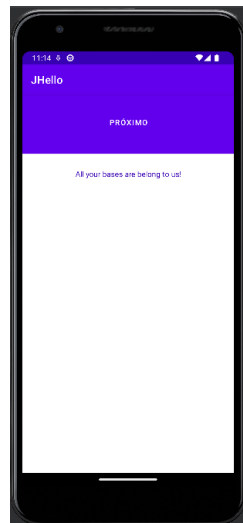
```
1 <RelativeLayout
2     android:layout_width="match_parent"
3     android:layout_height="match_parent" >
4     <TextView
5         android:id="@+id/center"
6         android:layout_width="wrap_content"
7         android:layout_height="wrap_content"
8         android:layout_centerInParent="true"
9         android:text="Center" />
10    <TextView
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:layout_toLeftOf="@id/center"
14        android:layout_alignBottom="@id/center"
15        android:text="2" />
16    <TextView
17        android:layout_width="wrap_content"
18        android:layout_height="wrap_content"
19        android:layout_toRightOf="@id/center"
20        android:layout_alignBottom="@id/center"
21        android:text="3" />
22    <TextView
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_above="@id/center"
26        android:layout_alignLeft="@id/center"
27        android:text="1" />
28    <TextView
29        android:layout_width="wrap_content"
30        android:layout_height="wrap_content"
31        android:layout_below="@id/center"
32        android:layout_alignLeft="@id/center"
33        android:text="4" />
34 </RelativeLayout>
```



Interface

Layouts - LinearLayout - activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <LinearLayout
11         android:layout_width="409dp"
12         android:layout_height="729dp"
13         android:layout_marginStart="2dp"
14         android:layout_marginTop="2dp"
15         android:layout_marginEnd="2dp"
16         android:layout_marginBottom="2dp"
17         android:orientation="vertical"
18         app:layout_constraintBottom_toBottomOf="parent"
19         app:layout_constraintEnd_toEndOf="parent"
20         app:layout_constraintStart_toStartOf="parent"
21         app:layout_constraintTop_toTopOf="parent">
22         <Button
23             android:id="@+id/btn_main_next"
24             android:layout_width="match_parent"
25             android:layout_height="128dp"
26             android:text="@string/btn_next_label" />
27         <TextView
28             android:id="@+id/textView"
29             android:layout_width="match_parent"
30             android:layout_height="64dp"
31             android:gravity="center"
32             android:text="@string/txt_view_default"
33             android:textAlignment="center"
34             android:textColor="@color/purple_700" />
35     </LinearLayout>
36 </androidx.constraintlayout.widget.ConstraintLayout>
```



Interface

Layouts - FrameLayout - activity_main.xml

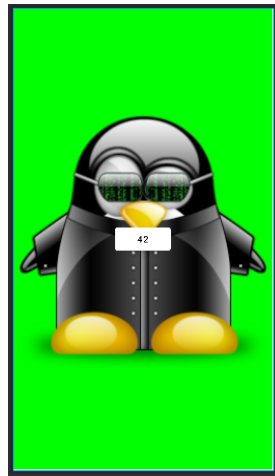
```
1 <FrameLayout
2     android:layout_width="409dp"
3     android:layout_height="729dp"
4     android:layout_marginStart="8dp"
5     android:layout_marginTop="8dp"
6     android:layout_marginEnd="8dp"
7     android:layout_marginBottom="8dp"
8     app:layout_constraintBottom_toBottomOf="parent"
9     app:layout_constraintEnd_toEndOf="parent"
10    app:layout_constraintStart_toStartOf="parent"
11    app:layout_constraintTop_toTopOf="parent">
12
13    <ImageView
14        android:id="@+id/img_linux"
15        android:layout_width="wrap_content"
16        android:layout_height="wrap_content"
17        android:layout_gravity="top|center"
18        android:background="@color/green"
19        app:srcCompat="@drawable/linux_matrix" />
20
21    <Button
22        android:id="@+id/btn_login"
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_gravity="bottom|center"
26        android:backgroundTint="@color/green"
27        android:text="@string/btnlabellogin"
28        android:textColor="@color/black" />
29 </FrameLayout>
```



Interface

Layouts - FrameLayout - activity_main.xml

```
1 <FrameLayout
2     android:layout_width="409dp"
3     android:layout_height="729dp"
4     android:layout_marginStart="8dp"
5     android:layout_marginTop="8dp"
6     android:layout_marginEnd="8dp"
7     android:layout_marginBottom="8dp"
8     app:layout_constraintBottom_toBottomOf="parent"
9     app:layout_constraintEnd_toEndOf="parent"
10    app:layout_constraintStart_toStartOf="parent"
11    app:layout_constraintTop_toTopOf="parent">
12
13    <ImageView
14        android:id="@+id/img_linux"
15        android:layout_width="match_parent"
16        android:layout_height="match_parent"
17        android:layout_gravity="top|center"
18        android:background="@color/green"
19        app:srcCompat="@drawable/linux_matrix" />
20
21    <Button
22        android:id="@+id/btn_login"
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_gravity="center"
26        android:backgroundTint="@color/white"
27        android:text="@string/btnlabellogin"
28        android:textColor="@color/black" />
29 </FrameLayout>
```



Interface

Componentes de interface

- Text
- Buttons
- Widgets
- Layouts
- Containers
- Helpers
- Google
- Legacy

Interface

Componentes de interface

```
1      <FrameLayout
2      android:layout_width="409dp"
3      android:layout_height="729dp"
4      android:layout_marginStart="8dp"
5      android:layout_marginTop="8dp"
6      android:layout_marginEnd="8dp"
7      android:layout_marginBottom="8dp"
8      app:layout_constraintBottom_toBottomOf="parent"
9      app:layout_constraintEnd_toEndOf="parent"
10     app:layout_constraintStart_toStartOf="parent"
11     app:layout_constraintTop_toTopOf="parent">
12
13     <ImageView
14     android:id="@+id/img_linux"
15     android:layout_width="match_parent"
16     android:layout_height="match_parent"
17     android:layout_gravity="top|center"
18     android:background="@color/green"
19     app:srcCompat="@drawable/linux_matrix" />
20
21     <Button
22     android:id="@+id/btn_login" <--
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:layout_gravity="center"
26     android:backgroundTint="@color/white"
27     android:text="@string/btnlabellogin"
28     android:textColor="@color/black" />
29 </FrameLayout>
```



Interface

Componentes de interface

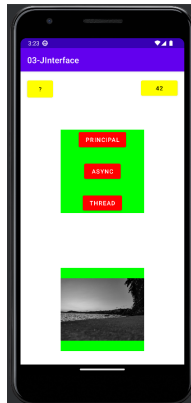
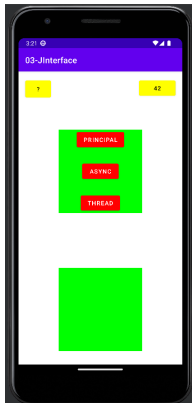
```
1 public class MainActivity extends AppCompatActivity {
2     private Button mBtnLogin = null;
3
4     @Override
5     protected void onCreate(Bundle savedInstanceState) {
6         super.onCreate(savedInstanceState);
7         setContentView(R.layout.activity_main);
8         mBtnLogin = findViewById(R.id.btn_login);
9     }
10
11     private class onButtonClickEvent implements View.OnClickListener{
12         @Override
13         public void onClick(View view) {
14             if (view.getId() == R.id.btn_login){
15                 Toast t = Toast.makeText(getApplicationContext(),
16                     "All your bases are belong to us!",
17                     Toast.LENGTH_LONG);
18                 t.show();
19             } //if (view.getId() == R.id.btn_main_next){
20             } //public void onClick(View view){
21             } //private class onButtonClickEvent implements View.OnClickListener{
22     }
```

Interface

Componentes de interface

- interface, seus elementos e os eventos envolvidos;
- app, programa ou software e o ele faz;
- estrutura da Activity:
 - gerência da interface;
 - lógica envolvida;
- Exemplo disponível em:

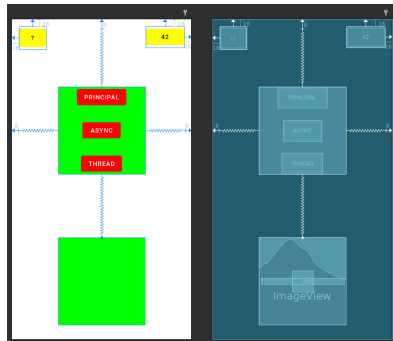
<https://github.com/mzamith-ufrrj/Android-examples/tree/main/03JInterface>



Interface

Componentes de interface

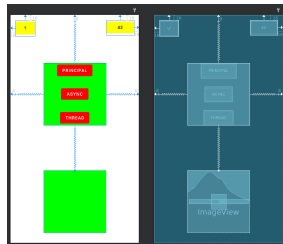
- `ConstraintLayout`
- `mainframelayout`
 - ▶ `btn_main` - "Principal"
 - ▶ `btn_async` - "Async"
 - ▶ `btn_thread` - "Thread"
- `imageframelayout`
 - ▶ `imgSource`
 - ▶ `progressBarAsync`
 - ▶ `progressBarThread`
- `btn_Clean` - "?"
- `btn_ShowMsg` - "42"



Interface

Componentes de interface

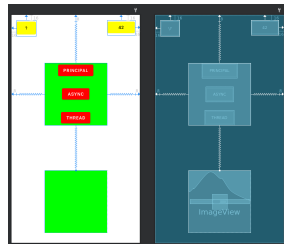
```
1 <FrameLayout
2     android:id="@+id/mainframelayout"
3     android:layout_width="200dp"
4     android:layout_height="200dp"
5     android:layout_marginStart="8dp"
6     android:layout_marginTop="8dp"
7     android:layout_marginEnd="8dp"
8     android:background="@color/green"
9     app:layout_constraintBottom_toTopOf="@+id/imageframelayout"
10    app:layout_constraintEnd_toEndOf="parent"
11    app:layout_constraintStart_toStartOf="parent"
12    app:layout_constraintTop_toTopOf="parent">
13    <Button
14        android:id="@+id/btn_main"
15        android:layout_width="wrap_content"
16        android:layout_height="wrap_content"
17        android:layout_gravity="top|center"
18        android:backgroundTint="@color/red"
19        android:text="Principal" />
20    <Button
21        android:id="@+id/btn_async"
22        android:layout_width="wrap_content"
23        android:layout_height="wrap_content"
24        android:layout_gravity="center"
25        android:backgroundTint="@color/red"
26        android:text="Async" />
27    <Button
28        android:id="@+id/btn_thread"
29        android:layout_width="wrap_content"
30        android:layout_height="wrap_content"
31        android:layout_gravity="bottom|center"
32        android:backgroundTint="@color/red"
33        android:text="Thread" />
34 </FrameLayout>
```



Interface

Componentes de interface

```
1 <FrameLayout
2     android:id="@+id/imageframelayout"
3     android:layout_width="200dp"
4     android:layout_height="200dp"
5     android:layout_marginStart="4dp"
6     android:layout_marginEnd="4dp"
7     android:layout_marginBottom="32dp"
8     android:background="@color/green"
9     app:layout_constraintBottom_toBottomOf="parent"
10    app:layout_constraintEnd_toEndOf="parent"
11    app:layout_constraintStart_toStartOf="parent">
12    <ImageView
13        android:id="@+id/imgSource"
14        android:layout_width="match_parent"
15        android:layout_height="match_parent" />
16    <ProgressBar
17        android:id="@+id/progressBarAsync"
18        style="?android:attr/progressBarStyleHorizontal"
19        android:layout_width="190dp"
20        android:layout_height="15dp"
21        android:layout_gravity="center"
22        android:progressTint="@color/red"
23        android:visibility="invisible" />
24    <ProgressBar
25        android:id="@+id/progressBarThread"
26        style="?android:attr/progressBarStyle"
27        android:layout_gravity="center"
28        android:visibility="invisible"
29        android:layout_width="wrap_content"
30        android:layout_height="wrap_content" />
31 </FrameLayout>
```



Interface

Lógica por trás da interface

● Inicializando os componentes

```
1  protected void onCreate(Bundle savedInstanceState) {
2      super.onCreate(savedInstanceState);
3      setContentView(R.layout.activity_main);
4      onClick buttonClick = new onClick();
5
6      mImgSource = findViewById(R.id.imgSource);
7      mBtnMain = findViewById(R.id.btn_main);
8      mBtnAsync = findViewById(R.id.btn_async);
9      mBtnThread = findViewById(R.id.btn_thread);
10     mBtnClean = findViewById(R.id.btn_Clean);
11     mBtnShowMsg = findViewById(R.id.btn_ShowMsg);
12
13     mProgressBarAsync = findViewById(R.id.progressBarAsync);
14     mProgressBarThread = findViewById(R.id.progressBarThread);
15
16     //mProgressBarAsync.setVisibility(View.VISIBLE);
17
18     mBtnMain.setOnClickListener(buttonClick);
19     mBtnAsync.setOnClickListener(buttonClick);
20     mBtnThread.setOnClickListener(buttonClick);
21     mBtnClean.setOnClickListener(buttonClick);
22     mBtnShowMsg.setOnClickListener(buttonClick);
23
24
25     mHandler = new Handler() {...};
26
27 } //protected void onCreate(Bundle savedInstanceState) {
28
```

Interface

Lógica por trás da interface

- Capturando o evento dos botões

```
1 private class onButtonClick implements View.OnClickListener{
2     @Override
3     public void onClick(View view) {
4         switch (view.getId()){
5             case R.id.btn_main: btnMainEvent(); break;
6             case R.id.btn_async: btnAsyncEvent(); break;
7             case R.id.btn_thread: btnThreadEvent(); break;
8             case R.id.btn_Clean: mImgSource.setImageBitmap(null); break;
9             case R.id.btn_ShowMsg: ShowMessage(); break;
10        }
11    }
12 }
13 } //private class onButtonClick implements View.OnClickListener{
```

Interface

Lógica por trás da interface

- Capturando o evento dos botões

```
1      private class onButtonClick implements View.OnClickListener{
2          @Override
3          public void onClick(View view) {
4              switch (view.getId()){
5                  case R.id.btn_main:btnMainEvent(); break;
6                  case R.id.btn_async:btnAsyncEvent(); break;
7                  case R.id.btn_thread:btnThreadEvent(); break;
8                  case R.id.btn_Clean:mImgSource.setImageBitmap(null); break;
9                  case R.id.btn_ShowMsg:ShowMessage(); break;
10             }
11         }
12     }
13 } //private class onButtonClick implements View.OnClickListener{
```

Interface

Lógica por trás da interface

- Computação na thread principal

```
1 private void btnMainEvent(){
2     Bitmap bitmap_in = BitmapFactory.
3         decodeResource(getResources(), R.drawable.img_2048_1536);
4     int w = bitmap_in.getWidth();
5     int h = bitmap_in.getHeight();
6
7     int[] pixels = new int[w * h];
8     bitmap_in.getPixels(pixels, 0, w, 0, 0, w, h);
9     bitmap_in = null;
10
11     for (int i = 0; i < w * h; i++){
12         pixels[i] = RGB2BW(pixels[i]);
13     }
14
15     Bitmap bitmap_out = Bitmap.createBitmap(w, h, Bitmap.Config.ARGB_8888);
16     bitmap_out.setPixels(pixels, 0, w, 0, 0, w, h);
17     mImgSource.setImageBitmap(bitmap_out);
18 }
```

Interface

Lógica por trás da interface

● Computação usando AsyncTask

```
1 private class AsyncTaskRunner extends AsyncTask<String, String, String>{
2     private Bitmap mBitmap_out = null;
3     @Override
4     protected String doInBackground(String... strings) {
5         Bitmap bitmap_in = BitmapFactory.
6             decodeResource(getResources(), Integer.parseInt(strings[0]));
7         ...
8         mProgressBarAsync.setMax(w * h);
9         ...
10        for (int i = 0; i < w * h; i++){
11            if ((i % percent) == 0)
12                publishProgress(Integer.toString(i+1));
13            pixels[i] = RGB2BW(pixels[i]);
14        }
15        ...
16        return null;
17    }
18
19    @Override
20    protected void onProgressUpdate(String... progress) {
21        mProgressBarAsync.setProgress(Integer.parseInt(progress[0]));
22    }
23
24    @Override
25    protected void onPostExecute(String result) {
26        if (mBitmap_out != null) {
27            mImgSource.setImageBitmap(mBitmap_out);
28            mProgressBarAsync.setVisibility(View.INVISIBLE);
29            mProgressBarAsync.setProgress(0);
30        }
31    }
32 }
```

Interface

Lógica por trás da interface

● Computação usando AsyncTask

```
1      private class AsyncTaskRunner extends AsyncTask<String, String, String>{
2          private Bitmap mBitmap_out = null;
3          @Override
4          protected String doInBackground(String... strings) {
5              Bitmap bitmap_in = BitmapFactory.
6                  decodeResource(getResources(), Integer.parseInt(strings[0]));
7              ...
8              mProgressBarAsync.setMax(w * h);
9              ...
10             for (int i = 0; i < w * h; i++){
11                 if ((i % percent) == 0)
12                     publishProgress(Integer.toString(i+1));
13                 pixels[i] = RGB2BW(pixels[i]);
14             }
15             ...
16             return null;
17         }
18
19         @Override
20         protected void onProgressUpdate(String... progress) {
21             mProgressBarAsync.setProgress(Integer.parseInt(progress[0]));
22         }
23
24         @Override
25         protected void onPostExecute(String result) {
26             if (mBitmap_out != null) {
27                 mImgSource.setImageBitmap(mBitmap_out);
28                 mProgressBarAsync.setVisibility(View.INVISIBLE);
29                 mProgressBarAsync.setProgress(0);
30             }
31         }
32     }
```

Interface

Lógica por trás da interface

● Computação usando Runnable, thread e Handler

```
1 private class taskInRunnable implements Runnable{
2     ...
3
4     @Override
5     public void run() {
6         Bitmap bitmap_in = BitmapFactory.decodeResource(getResources(), mFileID);
7         ...
8         Message message = m_Handler.obtainMessage();
9         Bundle b = new Bundle();
10        b.putParcelable(MSGKEY, mBitmap_out);
11        message.setData(b);
12        m_Handler.sendMessage(message);
13    }
14 }
```

```
1 mHandler = new Handler() {
2     @Override
3     public void handleMessage(Message msg) {
4         Bundle bundle = msg.getData();
5         Bitmap bitmap = bundle.getParcelable(MSGKEY);
6         mImgSource.setImageBitmap(bitmap);
7         mProgressBarThread.setVisibility(View.INVISIBLE);
8     }
9 };
```

Considerações finais

- arquivos de cores: colors.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="purple_200">#FFBB86FC</color>
4     <color name="purple_500">#FF6200EE</color>
5     <color name="purple_700">#FF3700B3</color>
6     <color name="teal_200">#FF03DAC5</color>
7     <color name="teal_700">#FF018786</color>
8     <color name="black">#FF000000</color>
9     <color name="white">#FFFFFFFF</color>
10    <color name="red">#FFFF0000</color>
11    <color name="green">#FF00FF00</color>
12    <color name="blue">#FF0000FF</color>
13    <color name="yellow">#FFFFFF00</color>
14 </resources>
```

- arquivos de cores: strings.xml

```
1 <resources>
2     <string name="app_name">03-JInterface</string>
3     <string name="txt_number_enter_lbl">Digite um n.:</string>
4     <string name="txt_is_prime_lbl">-----</string>
5 </resources>
```

Considerações finais

- interfaces, celular e comunicação com usuário;
- componentes, layouts e organização na tela;
- arquivos com cores e *strings* separados;
- problemas de interfaces, poluição visual e defeitos.
- Exercício:
 - ▶ Montar uma interface de login contendo:
 - usuário e senha;
 - botão de login;
 - botão “não sou cadastrado”;
 - botão “esqueci a minha senha”;
 - usar esquema próprio de cores;
 - ▶ apresentar na próxima aula e explicar o uso dos componentes escolhidos

