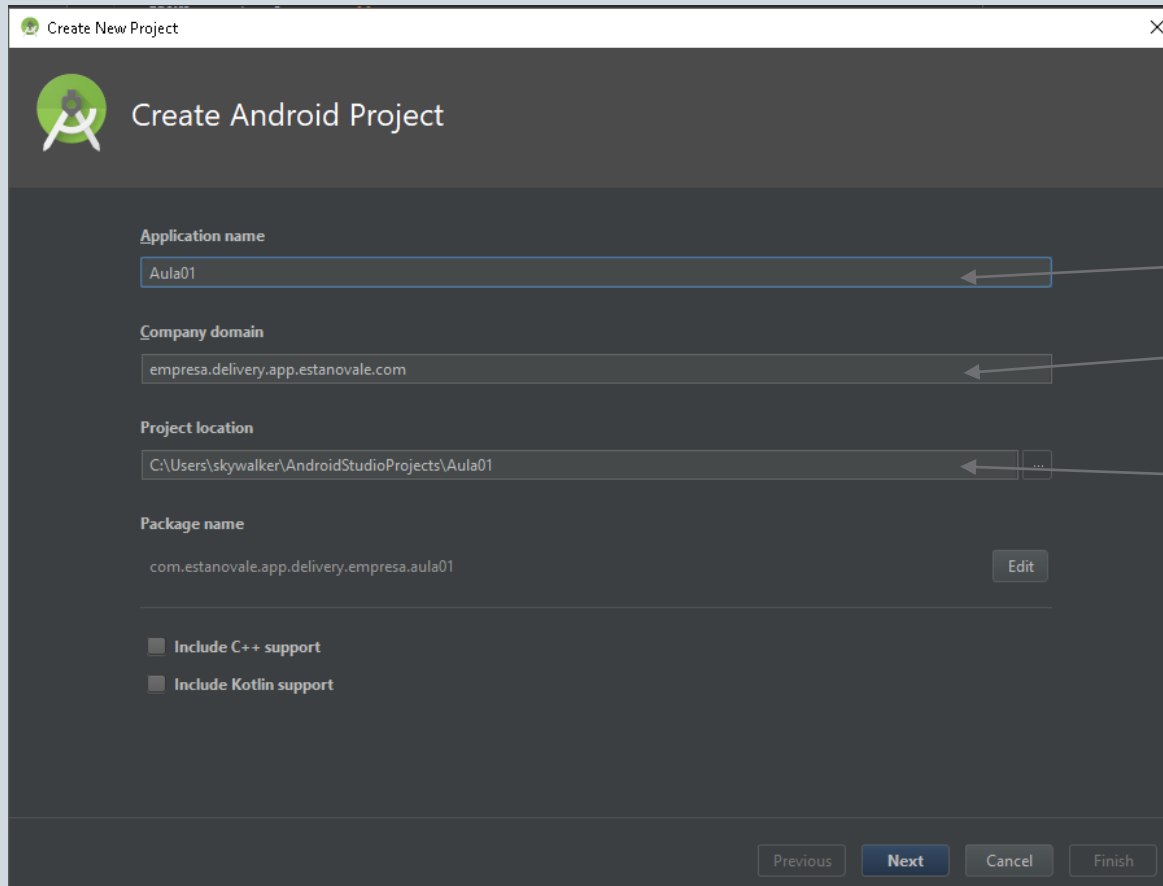


ANDROID STUDIO

PROF. ME. HÉLIO ESPERIDIÃO

Criar um projeto Android



Application name
Aula01

Company domain
empresa.delivery.app.estanovale.com

Project location
C:\Users\skywalker\AndroidStudioProjects\Aula01

Package name
com.estanovale.app.delivery.empresa.aula01

Include C++ support
 Include Kotlin support

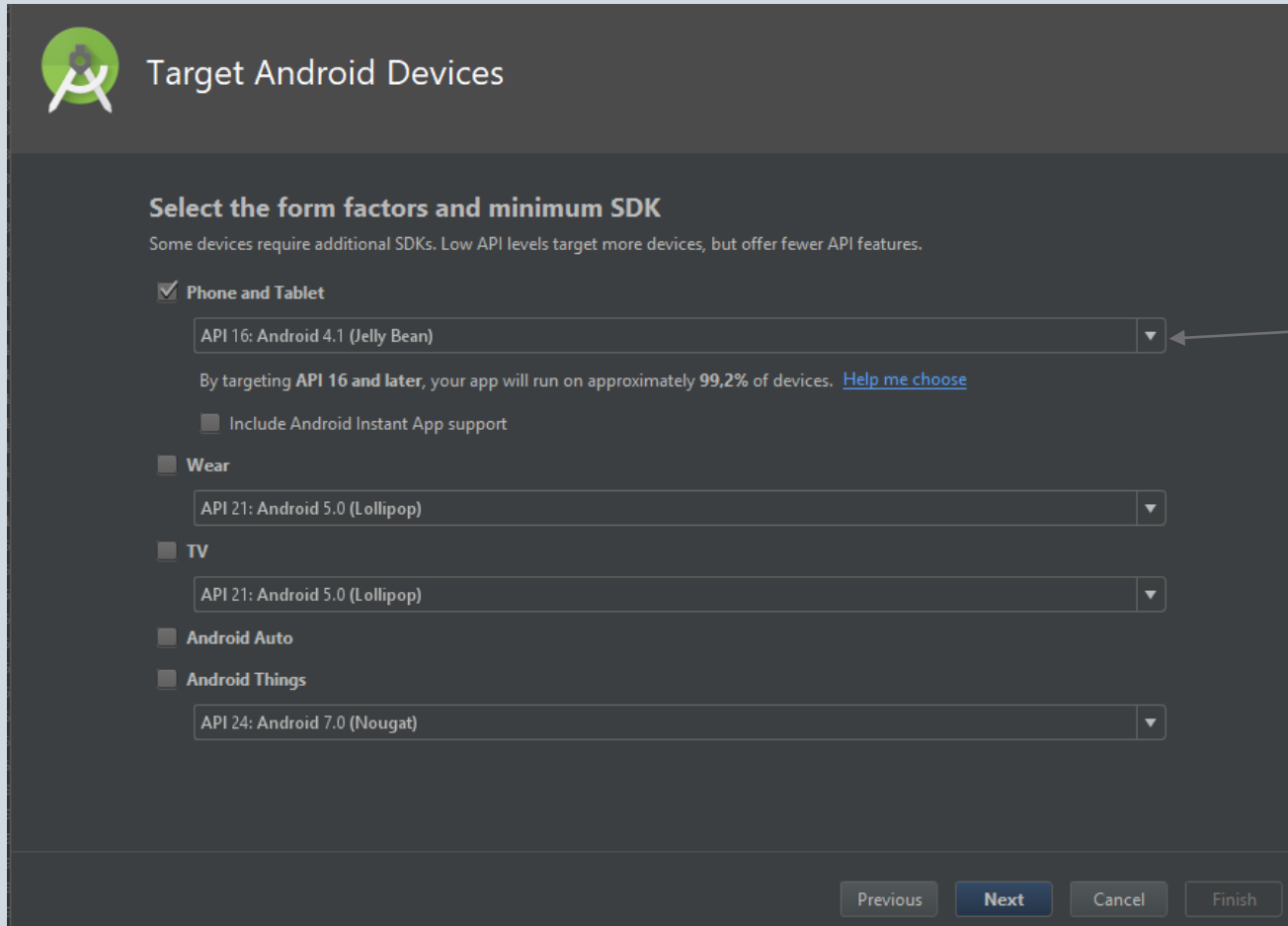
Previous Next Cancel Finish


Nome do projeto

Nome do pacote (nome do domínio ao contrário)

Local onde será salvo o projeto

Configurações do projeto.



 Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

Phone and Tablet

API 16: Android 4.1 (Jelly Bean) ▼

By targeting **API 16 and later**, your app will run on approximately 99,2% of devices. [Help me choose](#)

Include Android Instant App support

Wear

API 21: Android 5.0 (Lollipop) ▼

TV

API 21: Android 5.0 (Lollipop) ▼

Android Auto

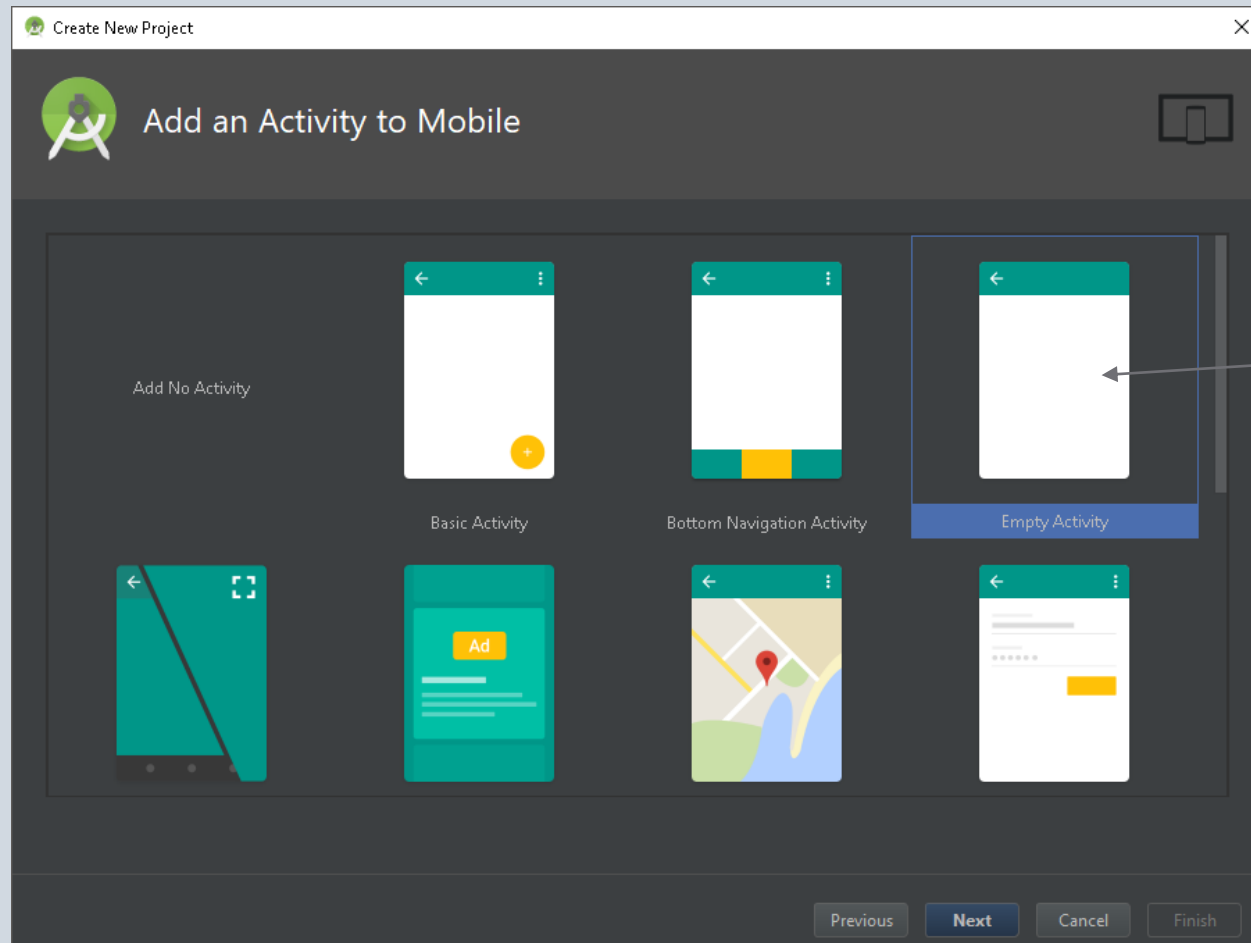
Android Things

API 24: Android 7.0 (Nougat) ▼

Previous **Next** Cancel Finish

Versão mínima do sistema operacional

Layouts prontos



Simple com as principais funcionalidades

Activity

Uma Activity é um módulo único e independente que normalmente está relacionada diretamente com uma tela de interface de usuário e suas funcionalidades correspondentes.

Configuração da atividade Principal.

Create New Project

Configure Activity

Creates a new empty activity

←

Activity Name
MainActivity

Generate Layout File

Layout Name
activity_main

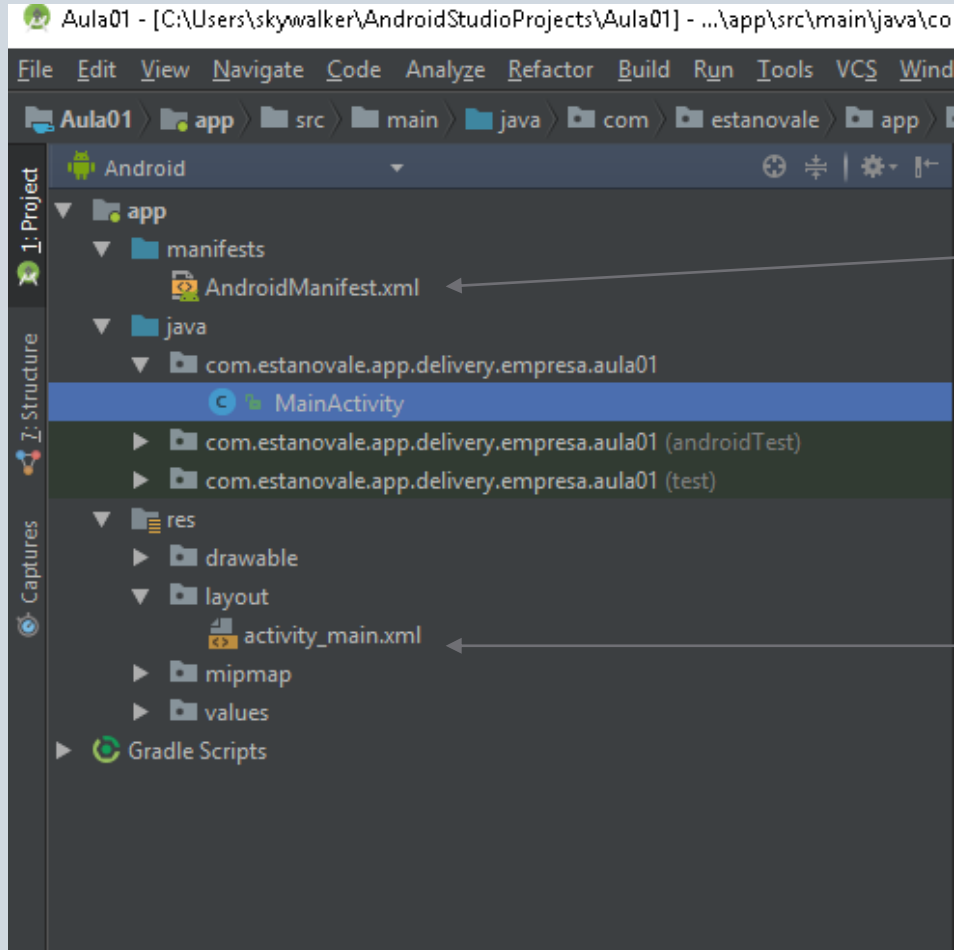
Backwards Compatibility (AppCompat)

Previous Next Cancel Finish

Nome do arquivo java

Nome do arquivo xml

Estrutura de arquivos



Arquivos java

Arquivos xml

Main Activity

```
delivery > empresa > aula01 > MainActivity >
activity_main.xml x MainActivity.java x
1 package com.estanovale.app.delivery.empresa.aula01;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

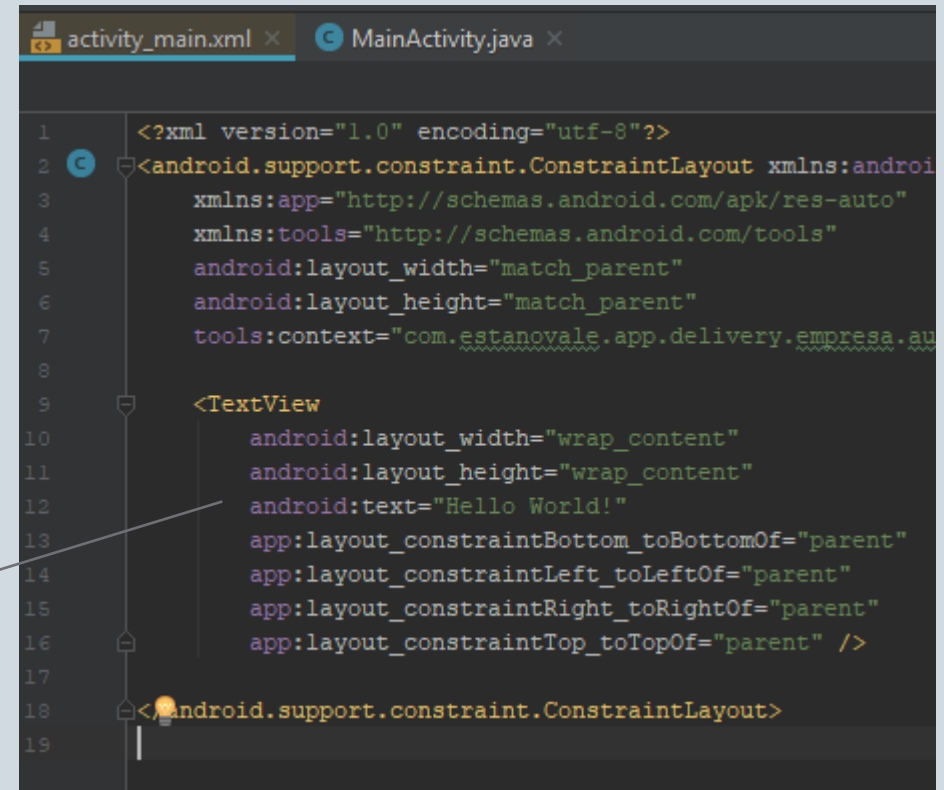
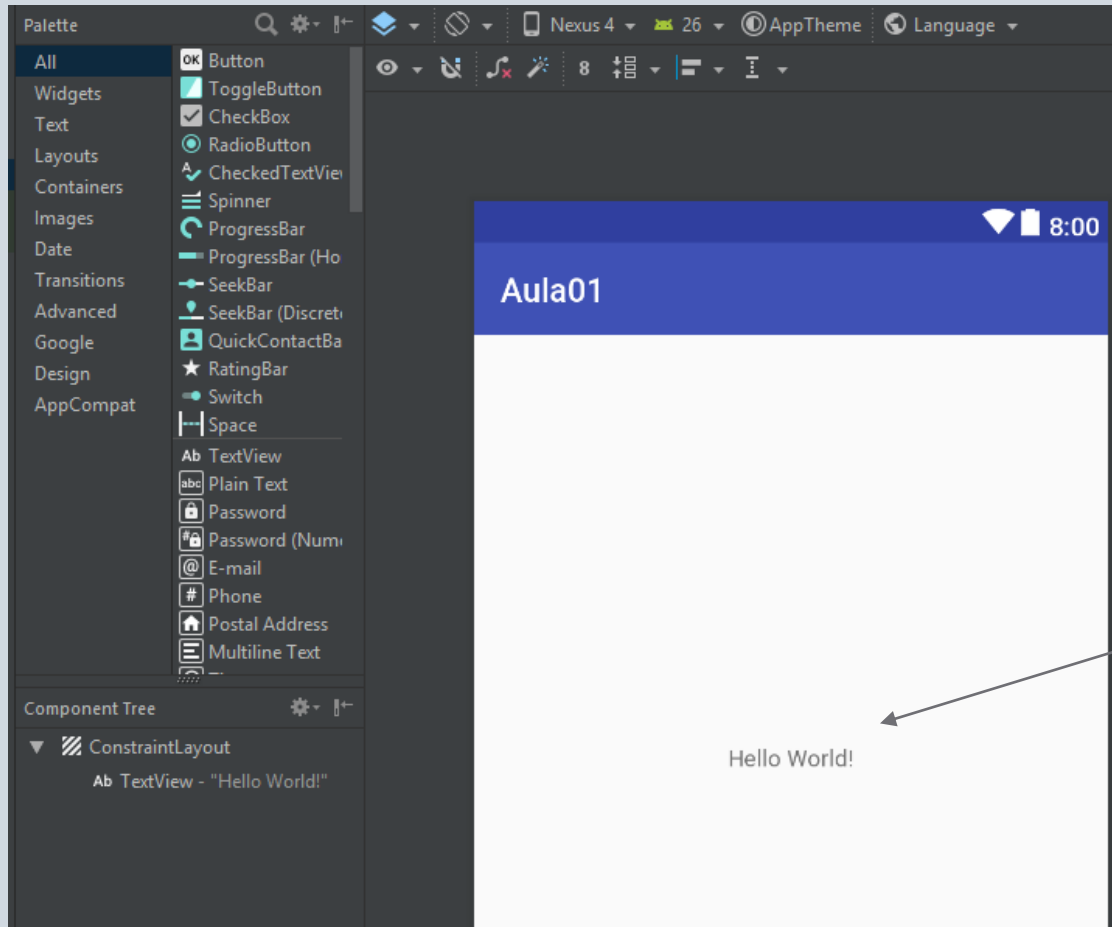
O método onCreate é o método que é executado Primeiro. Semelhante ao evento onload de outras linguagens

Arquivo xml

```
activity_main.xml x MainActivity.java x
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout 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="com.estanovale.app.delivery.empresa.aula01.MainActivity">
8
9   <TextView
10     android:layout_width="wrap_content"
11     android:layout_height="wrap_content"
12     android:text="Hello World!"
13     app:layout_constraintBottom_toBottomOf="parent"
14     app:layout_constraintLeft_toLeftOf="parent"
15     app:layout_constraintRight_toRightOf="parent"
16     app:layout_constraintTop_toTopOf="parent" />
17
18 </android.support.constraint.ConstraintLayout>
19
```

O arquivo xml é responsável pelo layout do aplicativo, nele são definidas as posições de todos os componentes visuais.

Editor Gráfico





All	ConstraintLayout
Widgets	GridLayout
Text	FrameLayout
Layouts	LinearLayout (horiz
Containers	LinearLayout (verti
Images	RelativeLayout
Date	TableLayout
Transitions	TableRow
Advanced	<fragment>
Google	
Design	
AppCompat	

Paleta de componentes visuais

Os layouts determinam como os componentes visuais serão distribuídos na janela.

O Layout LinearLayout é o mais simples pois permite que os componentes sejam posicionados um abaixo do outro ou lado a lado.

Estrutura básica do xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context="com.estanovale.app.delivery.empresaa.aula01.MainActivity">
    |
</android.support.constraint.ConstraintLayout>
```

Programe aqui seus componentes visuais.

Programme o LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context="com.estanovale.app.delivery_empresa.aula01.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
    >

    </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

Seu tamanho é definido de acordo
Com o tamanho da tela

Sua organização é na vertical, ou seja,
Os componentes ficarão um abaixo
Do outro.

XML e .java

Recupera o label e altera o seu texto

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
>

  <TextView
    android:id="@+id/lblOlamundo"
    android:text="Ola mundo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
  />

</LinearLayout>
```

```
public class MainActivity extends AppCompatActivity {

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

        TextView lblOlamundo = (TextView) findViewById(R.id.lblOlamundo);

        lblOlamundo.setText("Novo Ola mundo :D");
    }
}
```

Todos os arquivos xml a princípio devem ser associados a um arquivo java

Label, caixa de texto e botão

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
>

  <TextView
    android:id="@+id/lblOlamundo"
    android:text="Ola mundo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
  />

  <EditText
    android:id="@+id/txtNome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName" />

  <Button
    android:id="@+id/btnConfirmar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Confirmar" />

</LinearLayout>
```

Id do componente

Texto do componente

Largura e altura

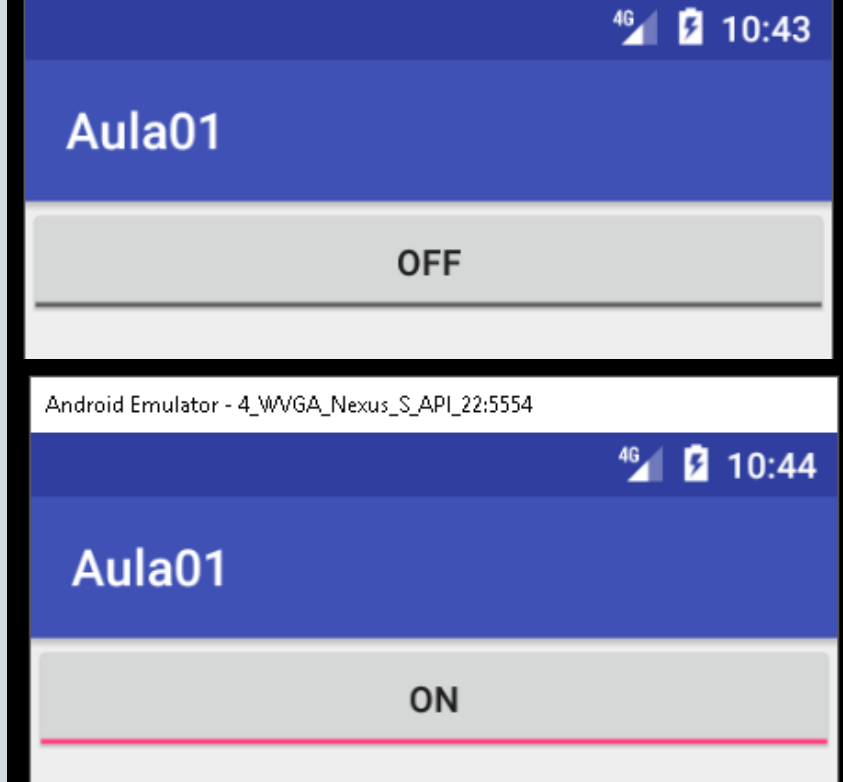
Evento click

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

    final TextView lblOlamundo = (TextView) findViewById(R.id.lblOlamundo);
    final EditText txtNome= (EditText) findViewById(R.id.txtNome);
    final Button btnConfirmar = (Button) findViewById(R.id.btnConfirmar);

    btnConfirmar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String nome = txtNome.getText().toString();
            Toast.makeText(view.getContext(), text: "entrou!", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Toast envia uma mensagem para o usuário



ToggleButton

<ToggleButton

```
    android:id="@+id/toggleButton"
```

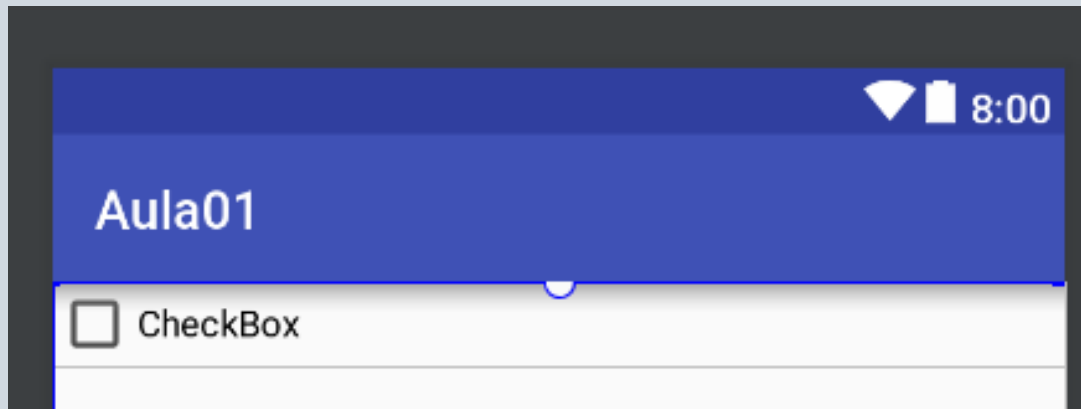
```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="ToggleButton" />
```

```
String value =toggleButton1.getText().toString();
```

```
Toast.makeText(this, value, Toast.LENGTH_SHORT).show();
```



```
<CheckBox  
    android:id="@+id/checkBox"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="CheckBox" />
```

checkBox

```
boolean value =checkBox1.isChecked();
```

Programme o LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    tools:context="com.estanovale.app.delivery_empresa.aula01.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
    >

    </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

Seu tamanho é definido de acordo
Com o tamanho da tela

Sua organização é na vertical, ou seja,
Os componentes ficarão um abaixo
Do outro.

XML e .java

Recupera o label e altera o seu texto

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
>

  <TextView
    android:id="@+id/lblOlamundo"
    android:text="Ola mundo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
  />

</LinearLayout>
```

```
public class MainActivity extends AppCompatActivity {

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

        TextView lblOlamundo = (TextView) findViewById(R.id.lblOlamundo);

        lblOlamundo.setText("Novo Ola mundo :D");
    }
}
```

Todos os arquivos xml a princípio devem ser associados a um arquivo java

Label, caixa de texto e botão

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
>

  <TextView
    android:id="@+id/lblOlamundo"
    android:text="Ola mundo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
  />

  <EditText
    android:id="@+id/txtNome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName" />

  <Button
    android:id="@+id/btnConfirmar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Confirmar" />

</LinearLayout>
```

Id do componente

Texto do componente

Largura e altura

Evento click

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

    final TextView lblOlamundo = (TextView) findViewById(R.id.lblOlamundo);
    final EditText txtNome= (EditText) findViewById(R.id.txtNome);
    final Button btnConfirmar = (Button) findViewById(R.id.btnConfirmar);

    btnConfirmar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String nome = txtNome.getText().toString();
            Toast.makeText(view.getContext(), text: "entroui", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Id do label

Id da caixa de texto

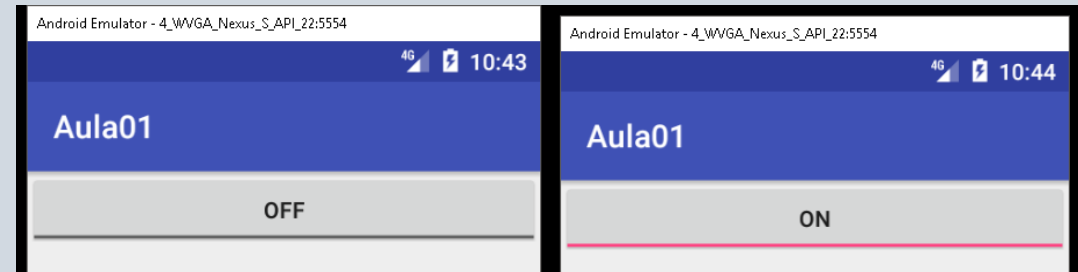
Id do botão

Toast envia uma mensagem para o usuário

ToggleButton

<ToggleButton

```
    android:id="@+id/toggleButton"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="ToggleButton" />
```

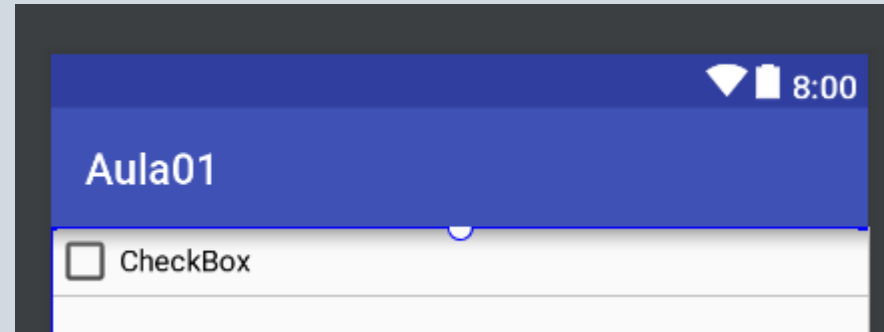


```
String value =toggleButton1.getText().toString();
```

```
Toast.makeText(this, value, Toast.LENGTH_SHORT).show();
```

checkBox

```
<CheckBox  
  android:id="@+id/checkBox"  
  android:layout_width="match_parent"  
  android:layout_height="wrap_content"  
  android:text="CheckBox" />
```



```
boolean value =checkBox1.isChecked();
```


Botões de Rádio

```
<RadioGroup
    android:id="@+id/radioSex"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/radioSexM"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="RadioButton" />

    <RadioButton
        android:id="@+id/radioSexF"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="RadioButton" />
</RadioGroup>

<Button
    android:id="@+id/btnOk"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button" />

</LinearLayout>
```

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

    Button btnClickMe = (Button) findViewById(R.id.btnOk);

    btnClickMe.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {

            final RadioButton radioSex = (RadioButton) findViewById(R.id.radioSexM);
            String value = String.valueOf(radioSex.isChecked());
            Toast.makeText(view.getContext(), value, Toast.LENGTH_SHORT).show();

        }
    });
}
```

Combo Box

```
final Spinner cboNomes = (Spinner) findViewById(R.id.cboNomes);
List<String> list = new ArrayList<String>();
list.add("Maria");
list.add("Marina");
list.add("Marina");
ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>( context: this, android.R.layout.simple_spinner_item, list);
dataAdapter.setDropDownViewResource (android.R.layout.simple_spinner_dropdown_item);
cboNomes.setAdapter (dataAdapter);

Button btnClickMe = (Button) findViewById(R.id.btnOk);

btnClickMe.setOnClickListener (new View.OnClickListener() {

    @Override
    public void onClick (View view) {

        String x = cboNomes.getSelectedItem().toString();

        Toast.makeText (view.getContext(), x, Toast.LENGTH_SHORT).show();

    }

});
```

```
<Spinner
    android:id="@+id/cboNomes"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/btnOk"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button" />
```

Arquivos

```
public void saveFile(String nomeArquivo, String dados, Context context){  
  
    FileOutputStream outputStream;  
  
    try {  
        outputStream = context.openFileOutput(nomeArquivo, Context.MODE_PRIVATE);  
        outputStream.write(dados.getBytes());  
        outputStream.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
public String loadFile(String nomeArquivo, Context context) {  
    String textoArquivo="";  
    FileInputStream inputStream;  
    try {  
        inputStream = context.openFileInput(nomeArquivo);  
  
        int content;  
  
        while ((content = inputStream.read()) != -1) {  
            textoArquivo+= (char) content;  
        }  
        inputStream.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return textoArquivo;  
}
```

Como utilizar:

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

    Button btnClickMe = (Button) findViewById(R.id.btnOk);

    btnClickMe.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {
            String nomeArquivo="nome.txt";
            String dados="helio Esperidião";
            saveFile(nomeArquivo,dados,view.getContext());

            String leitura= loadFile(nomeArquivo,view.getContext());

            Toast.makeText(view.getContext(), leitura,Toast.LENGTH_SHORT).show();
        }
    });
}
```

parseInt()

Converte uma string em número inteiro

```
String a = "123";
```

```
int x = Integer.parseInt(a);
```

parseFloat()

Converte uma string em número ponto flutuante

```
String a = "123.75";
```

```
double x = Double.parseDouble(a);
```

Expressões Aritméticas

Operador	Descrição	Hierarquia da Operação
+	Soma	3º
-	Subtração	3º
/	Multiplicação	2º
*	Divisão	2º
(%)	Resto da divisão	2º
()	Parênteses	1º 1º

$$7+2/2=?$$

$$(7+2)/2=?$$

Operadores relacionais

Operador	Função	Exemplo	Resultado
>	Maior	5>3	?
<	Menor	5<3	?
>=	Maior ou igual	5>=6	?
<=	Menor ou igual	7<=3	?
!=	Diferente de	5!=5	?
==	Igual a	5==5	?
!	Inversor	!5==5	?

Operadores lógicos

Operador	Função
&&	E
	Ou

```
float media=7;
```

```
if (media>=6) {
```

```
.....
```

```
}else{
```

```
.....
```

```
}
```

Condiciona
(IF) -
Exemplo

Sintaxe (WHILE)

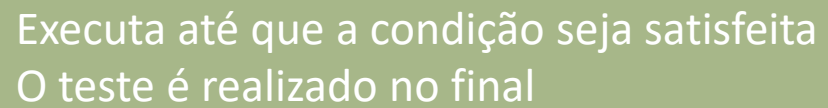
```
while( <condição> ) {  
    <Instrução>  
}
```

Sintaxe (FOR)

```
for (<variável de inicialização>;< condição >;<variável de incremento>) {  
    <Bloco de Instruções>  
}
```

Sintaxe (do-while)

```
Do{  
    <Bloco de Instruções>  
} while( <condição> );
```



Executa até que a condição seja satisfeita
O teste é realizado no final