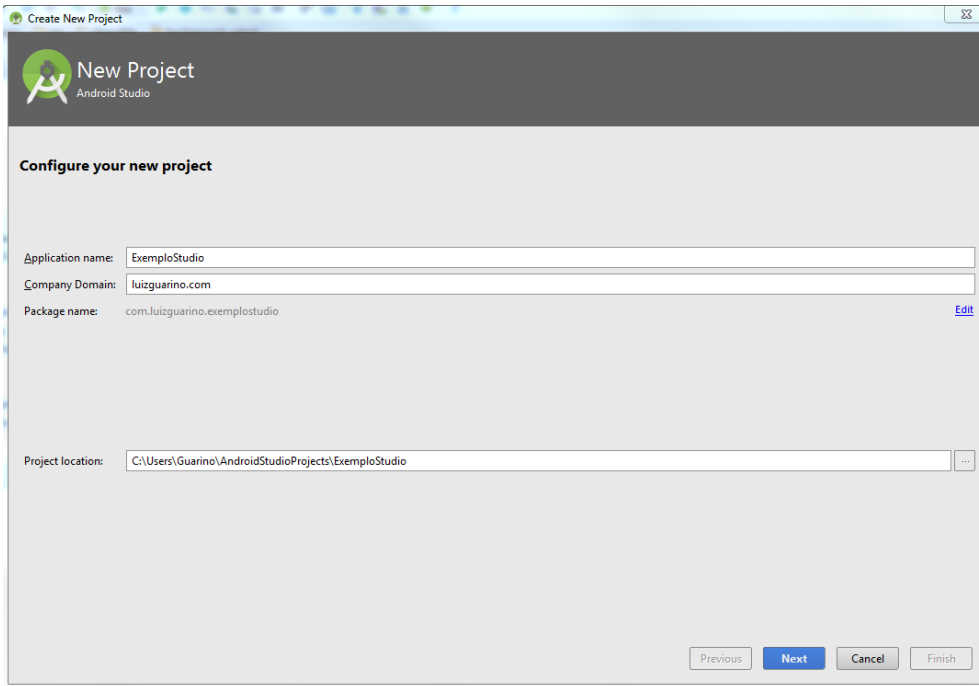
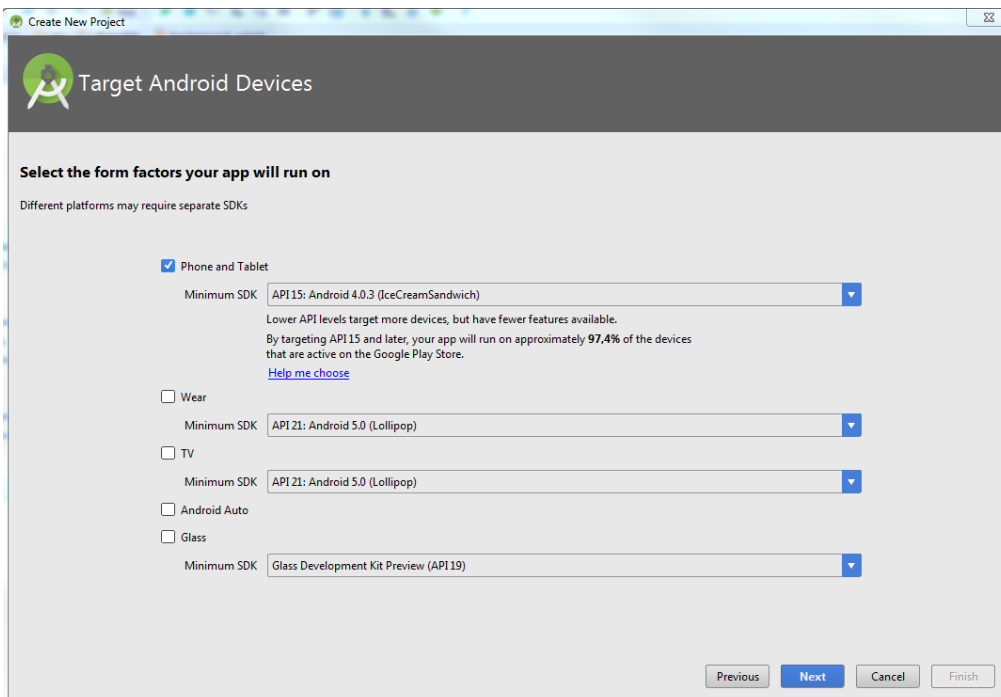


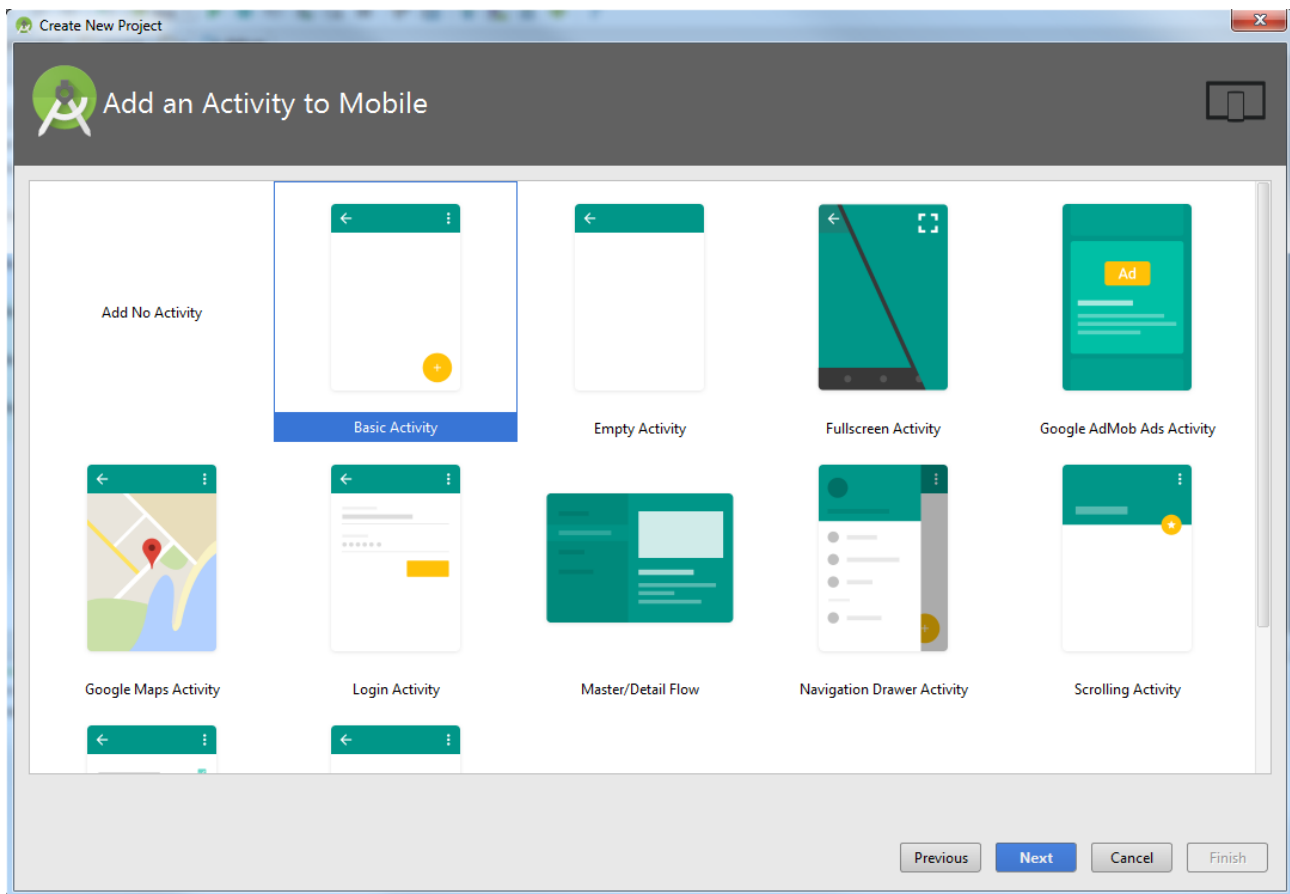
Criando o primeiro projeto para entender a estrutura do app

File > New > New Project

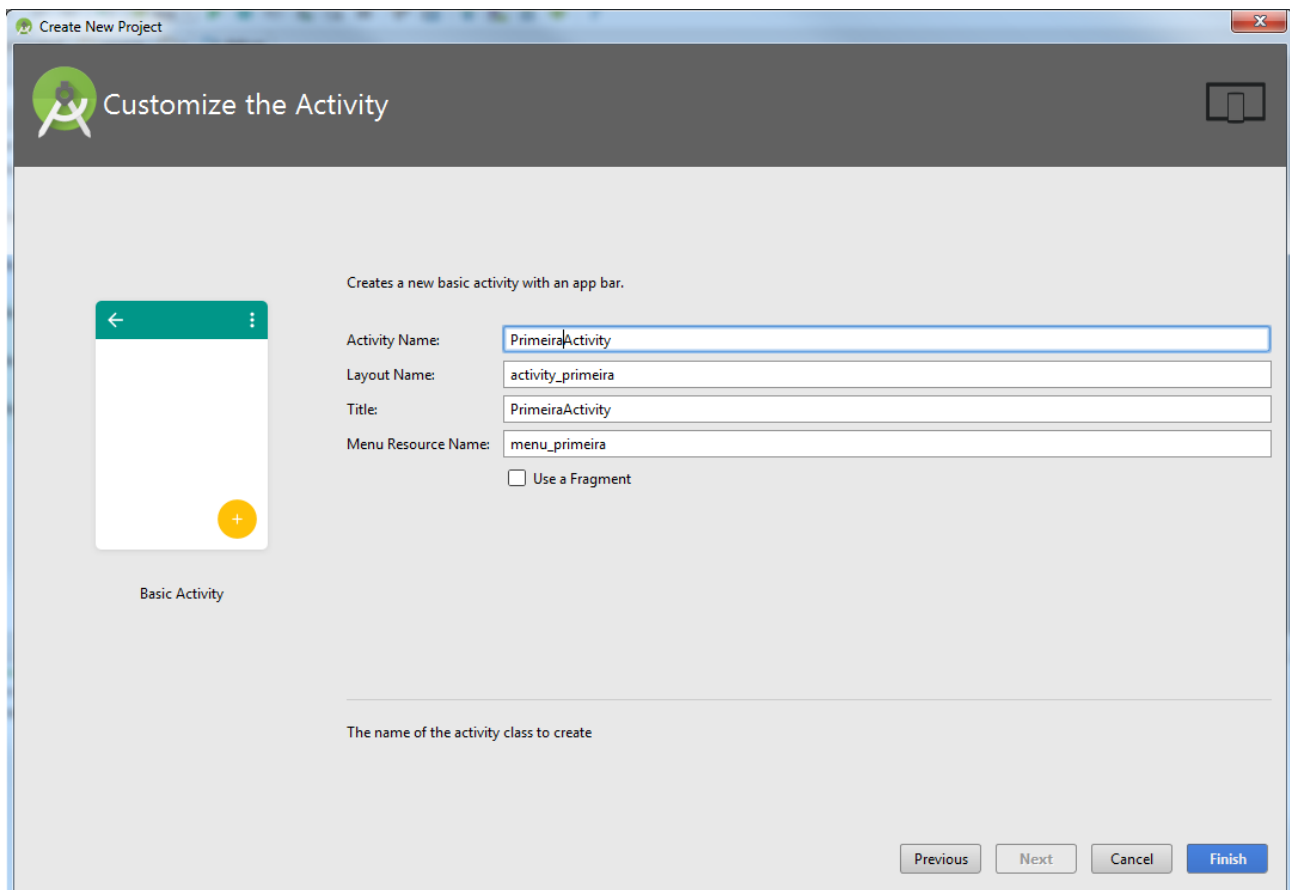


Escolha o tipo de projeto. Tenha sempre atenção a versão mínima da API a ser utilizada no projeto. Normalmente, quando menor a API mais dispositivos serão alcançados na distribuição. Porém, a retrocompatibilidade de componentes precisa ser considerada no desenvolvimento.

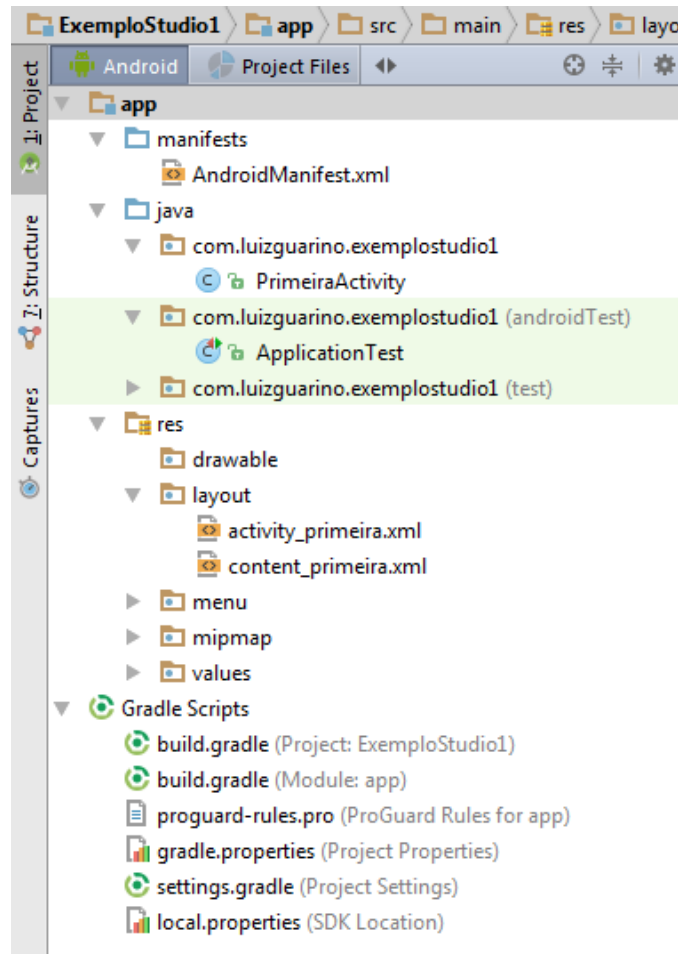




Criando a primeira Activity (Tela).



Estrutura do Projeto



Layout da tela (Activity – executar algo na tela)

app > res > layout > activity_primeira.xml.

Estrutura da tela. Pode ser visualizado em modo Design ou Text.

<http://developer.android.com/intl/pt-br/reference/android/app/Activity.html>

app > res > layout > content_primeira.xml.

Define conteúdo do elemento. Exemplo “Hello World”.

Arquivos Java

app > java > com.luizguarino.exemplostudio1 > PrimeiraActivity.java

Todas as activities, por convenção, **devem** estar no mesmo pacote.

- extends
- onCreate (primeiro método a ser disparado)
- classe R (intercâmbio entre as camadas – guarda todos os identificadores que foram criados no projeto – é uma classe autogerada com constantes para cada identificador, a fim de ligar o XML as classes Java)
- R.id.xxx – é uma constante que vem de activity_primeira.xml → @+id/xxx

Manifesto

app > manifests > AndroidManifest.xml

Características fundamentais do app. Coração do app. Principais configurações. Acessos de hardware. Permissões. Todas as activities são “declaradas” no manifesto.

Gradle Scripts

build.gradle – Android usa para compilar e construir a app.

- applicationId – sempre vai ter caminho completo do site, pois será usado na apk que será publicada na Google Play.
- dependencies – gerenciar as dependências do projeto. Mais simples que o Maven.

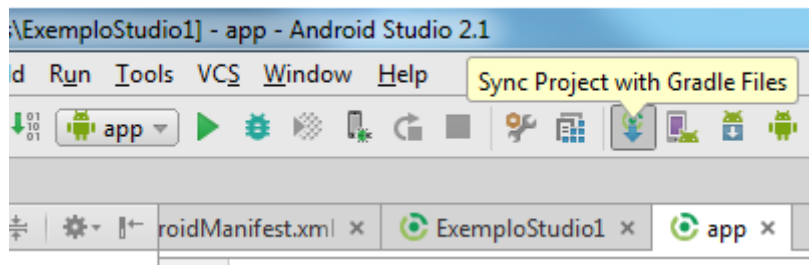
Pode buscar informações no Maven Repository - <http://mvnrepository.com/>

Se quiser adicionar uma nova dependência, basta ir no mvnrepository, encontrar a dependência, escolher a versão e depois ver como deve ser usado no Gradle. Em seguida, basta colocar no build.gradle a linha de comando compile 'nome_do_jar'.

Exemplo: <http://mvnrepository.com/artifact/log4j/log4j/1.2.17>

```
....
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
    compile 'com.android.support:design:23.3.0'
    compile 'log4j:log4j:1.2.17'
}
```

Ao alterar alguma coisa no Gradle, é importante **sincronizar** o projeto com o Gradle.



drawable – imagens, animação, canvases, entre outros

menu – itens de menu

mipmap – ícones

values

- strings – arquivo de propriedades. Guardam valores com identificador associado. Padronização de textos no projeto. Não deveríamos colocar texto dentro das telas, mas sim dentro desse arquivo. Texto de mensagens, botões, etc.
 - Pode ter string ou string-array

<resources>

<string name="ola_guarino">Olá Guarino</string>

<string-array name="lista_nomes"><item>Primeiro item</item></string-array>

</resources>

- dimens – arquivos de dimensões. Android lida com diferentes tipos de telas e dispositivos. O mesmo código tem que executar em todos.

- px – é um pixel. **Não trabalhe com pixels. Prefira sp e dip (dp).**
- sp – scale-independent pixels (**muito usados em font sizes**). Android redimensiona esse tipo de dimensão de acordo com o tamanho da tela. O texto será redimensionado de acordo com o tamanho ao redor.
- dip – density-independet pixels – **usado para praticamente todo o resto**. Margens internas. Espaço entre tabelas, entre componentes. dip == dp.
- in - polegadas
- mm - milímetros
- cm – centímetros.

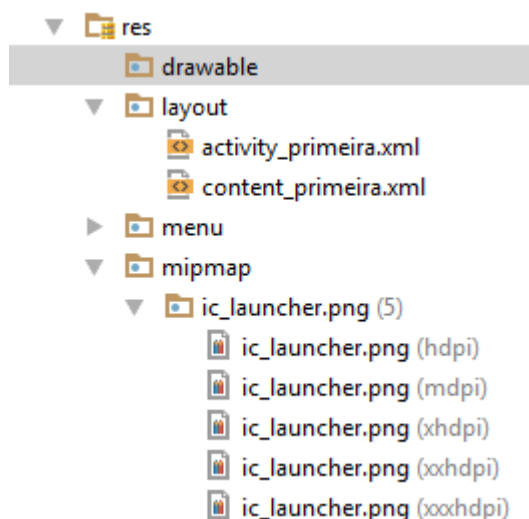
Density pixels

ldpi (low), mdpi (medium), hdpi (high), xhdpi, xxhdpi, xxxhdpi – arquivos de tamanhos diferentes.

Normalmente precisaremos criar imagens de tamanhos diferentes.

Density Bucket	Screen Display	Physical Size	Pixel Size
ldpi	120 dpi	0.5 x 0.5 in	0.5 in * 120 dpi = 60x60 px
mdpi	160 dpi	0.5 x 0.5 in	0.5 in * 160 dpi = 80x80 px
hdpi	240 dpi	0.5 x 0.5 in	0.5 in * 240 dpi = 120x120 px
xhdpi	320 dpi	0.5 x 0.5 in	0.5 in * 320 dpi = 160x160 px
xxhdpi	480 dpi	0.5 x 0.5 in	0.5 in * 480 dpi = 240x240 px
xxxhdpi	640 dpi	0.5 x 0.5 in	0.5 in * 640 dpi = 320x320 px

Ver, por exemplo, em res/mipmap/ic_launcher



Density	Description	Units Per Physical Inch	Density Independent	Same Physical Size On Every Screen
px	Pixels	Varies	No	No
in	Inches	1	Yes	Yes
mm	Millimeters	25.4	Yes	Yes
pt	Points	72	Yes	Yes
dp	Density Independent Pixels	~ 160	Yes	No
sp	Scale	~ 160	Yes	No

in, mm, pt – mesmo tamanho físico independente do tamanho da tela. NÃO É recomendado usar.

<resources>

```
<!-- Default screen margins, per the Android Design guidelines. -->
```

```
<dimen name="activity_horizontal_margin">16dp</dimen>
```

```
<dimen name="activity_vertical_margin">16dp</dimen>
```

```
<dimen name="fab_margin">16dp</dimen>
```

```
<dimen name="padding_layout">20dip</dimen>
```

```
<dimen name="texto_padrao">32sp</dimen>
```

```
</resources>
```

Alterando a content_primeira.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:padding="@dimen/padding_layout"
```

```
app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

```
tools:context="com.luizguarino.exemplostudio1.PrimeiraActivity"
```

```
tools:showIn="@layout/activity_primeira"
```

```
android:orientation="vertical"
```

```
>
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

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

```
android:text="Hello World!"
```

```
android:textSize="@dimen/texto_padrao"
```

```
android:id="@+id/textView1" />
```

```
<TextView
```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@+id/textView1"
    android:text="@string/ola_guarino"
    android:id="@+id/textView2"/>
</LinearLayout>

```

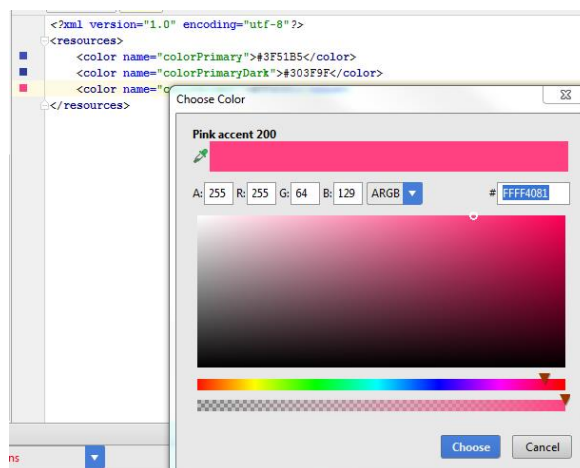
- colors – cores do app.
 - Só pode ter valores em Haxadecimal

```

<?xml version="1.0" encoding="utf-8" ?>
<resources>
  <color name="colorPrimary">#3F51B5</color>
  <color name="colorPrimaryDark">#303F9F</color>
  <color name="colorAccent">#FF4081</color>
</resources>

```

Ao clicar na cor, abre a paleta de cores



Exemplo no content_primeira.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout ...
  android:background="@color/colorAccent"
  >

<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Hello World!"
  android:textSize="@dimen/texto_padrao"
  android:textColor="@color/colorPrimary"
  android:id="@+id/textView1" />

```

- styles – estilos disponíveis.

```
<resources>
  <!-- Base application theme. -->
  ...
  <style name="Txt_teste">
    <item name="android:textSize">@dimen/texto_padrao</item>
    <item name="android:textColor">@color/colorPrimary</item>
  </style>
</resources>
```

No content_primeira.xml

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@+id/textView1"
  android:text="@string/ola_guarino"
  style="@style/Txt_teste"
  android:id="@+id/textView2"/>
```

Sempre que precisar acessar um **Resource (pasta res)**, vai precisar de @

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_toRightOf="@+id/textView1"
  android:text="@string/ola_guarino"
  android:id="@+id/textView2"/>
```

Tem-se ainda o @+id para acessar componentes.

Tem-se @android para acessar recursos nativos do Android.

Outros tipos de Resources

<http://developer.android.com/intl/pt-br/guide/topics/resources/more-resources.html>

Nomes dos resources

Não deve ter:

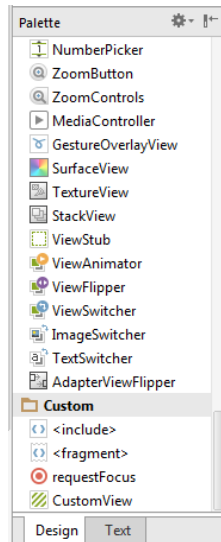
- caracteres especiais
- letras maiúsculas
- hífen

Palette (Modo Design no XML)

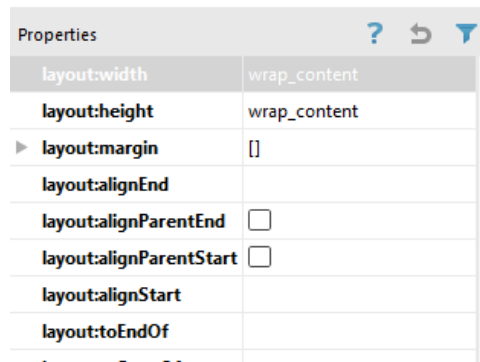
Controles que podem ser usados na tela.

Componente riscado está depreciado.

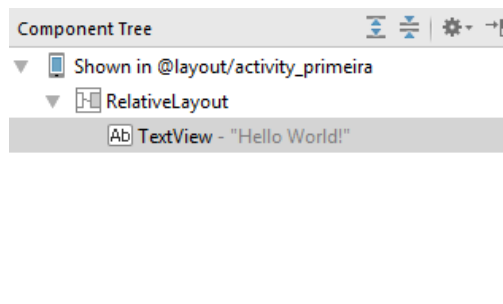
Os dispositivos mais novos não possuem esses componentes (SO Android mais novo).



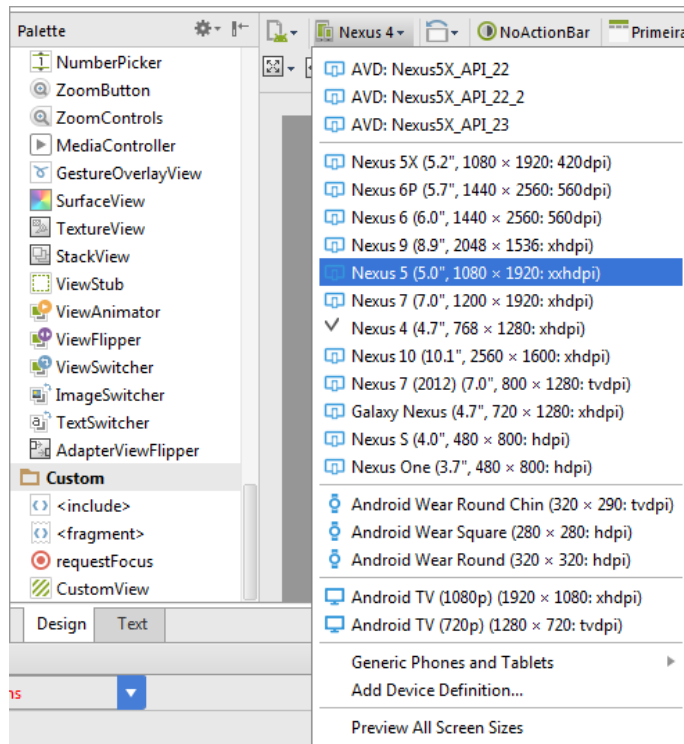
Cada componente tem **Propriedades**.



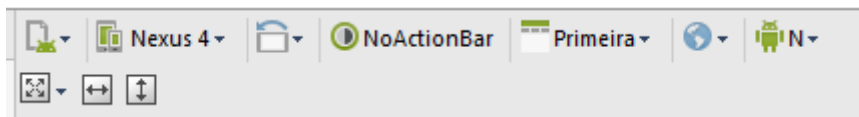
Estrutura de componentes (**Component Tree**)



Visualização em outros dispositivos



Outras opções interessantes



No emulador CTRL + F11 para mudar visualização (Portrait x Landscape)

Entendendo Layout

Abrindo res/layout/content_primeira.xml

Cada tela tem um Layout. Container com todos os elementos. Define como os componentes serão arranjados.

- LinearLayout – organiza um abaixo do outro ou um ao lado do outro.

<LinearLayout ...

android:orientation="vertical" >

- **android:orientation="vertical" OU**
- **android:orientation="horizontal"**

- RelativeLayout – um componente é relativo ao outro. Onde cada componente vai ficar. Acima, abaixo, ao lado de quem.

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Hello World!"
  android:id="@+id/textView1" />
```

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@+id/textView1"
  android:text="Hello World!"
  android:id="@+id/textView2"/>
```

below or **android:layout_toRightOf="@+id/textView1"**

Visualizar no Modo Design.

Pesquisar algo no Source – CTRL + SHIFT + R

Support Library

Compatibilidade com APIs antigas
Hoje, estamos na API 26.

AppCompat v4 – a partir da versão Android 1.6
AppCompat v7 – a partir da versão Android 2.1

Pode ser observado em C:\Program Files (x86)\Android\android-sdk\extras\android\support

Pode ser adicionado via Gradle dentro de dependencias

```
dependencies {
  compile fileTree(dir: 'libs', include: ['*.jar'])
  testCompile 'junit:junit:4.12'
  compile 'com.android.support:appcompat-v7:23.3.0'
  compile 'com.android.support:design:23.3.0'
}
```

Depois de adicionar, basta sincronizar o gradle no projeto. A dependência deve ser baixada com sucesso.

Para verificar se as dependências foram adicionadas corretamente, acessar
File > Project Structure...
Aba Dependencies