



Layout

XML – eXtensible Markup Language



O que é XML?

- - Linguagem de marcação
- `<?xml version="1.0" encoding="utf-8" ?>`
- `<receita nome="pão">`
 - `<ingrediente quantidade="3" unidade="xicara" itens="farinha" />`
 - `<ingrediente quantidade="7" unidade="gramas" item="fermento" />`
- `</receita>`



```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools">
4
5      <application
6          android:allowBackup="true"
7          android:dataExtractionRules="@xml/data_extraction_rules"
8          android:fullBackupContent="@xml/backup_rules"
9          android:icon="@mipmap/ic_launcher"
10         android:label="Projeto_03"
11         android:supportsRtl="true"
12         android:theme="@style/Theme.Projeto_03"
13         tools:targetApi="31">
14         <activity
15             android:name=".MainActivity"
16             android:exported="true">
17             <intent-filter>
18                 <action android:name="android.intent.action.MAIN" />
19
20                 <category android:name="android.intent.category.LAUNCHER" />
21             </intent-filter>
22         </activity>
23     </application>
24
25 </manifest>
```

Configura o ícone do seu App

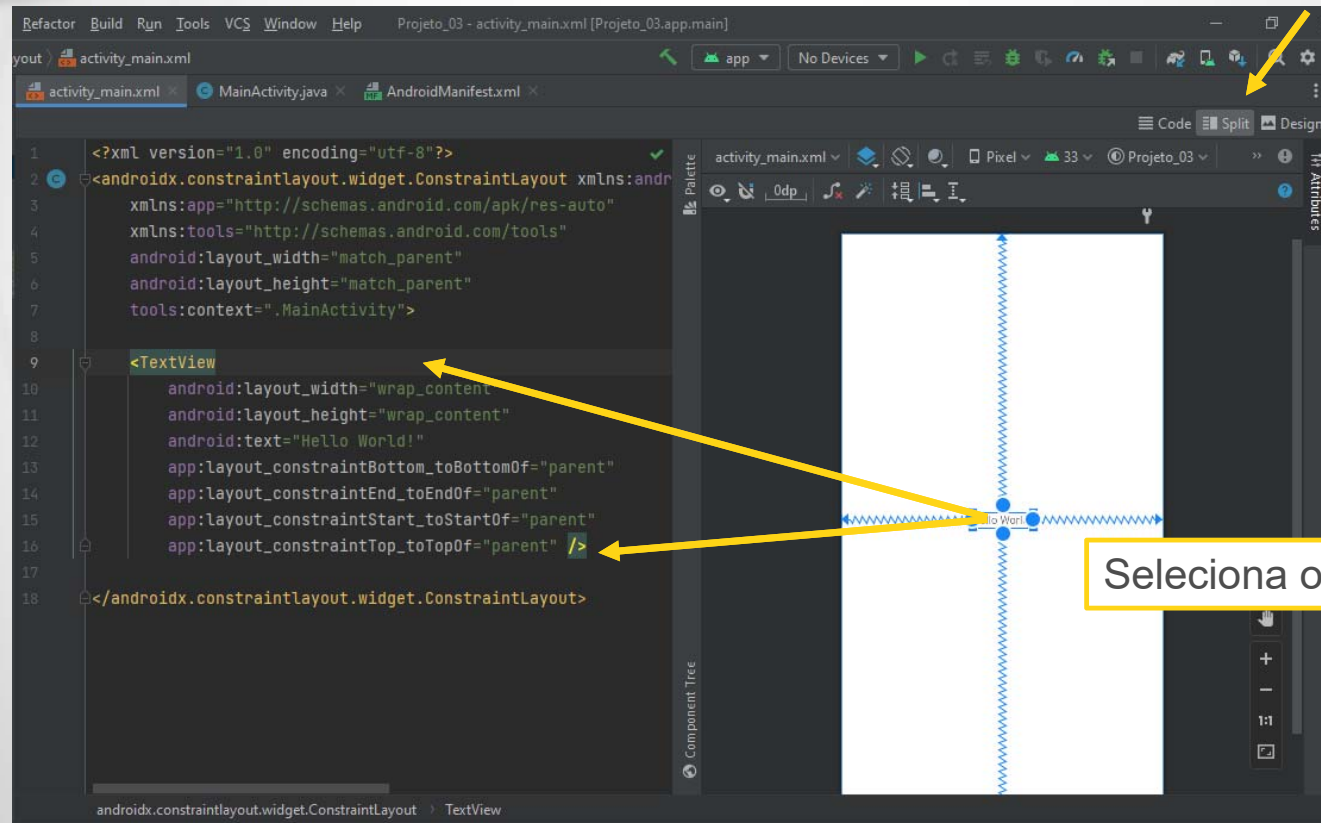
Nome do seu App

Define o tema do seu projeto

Interface principal

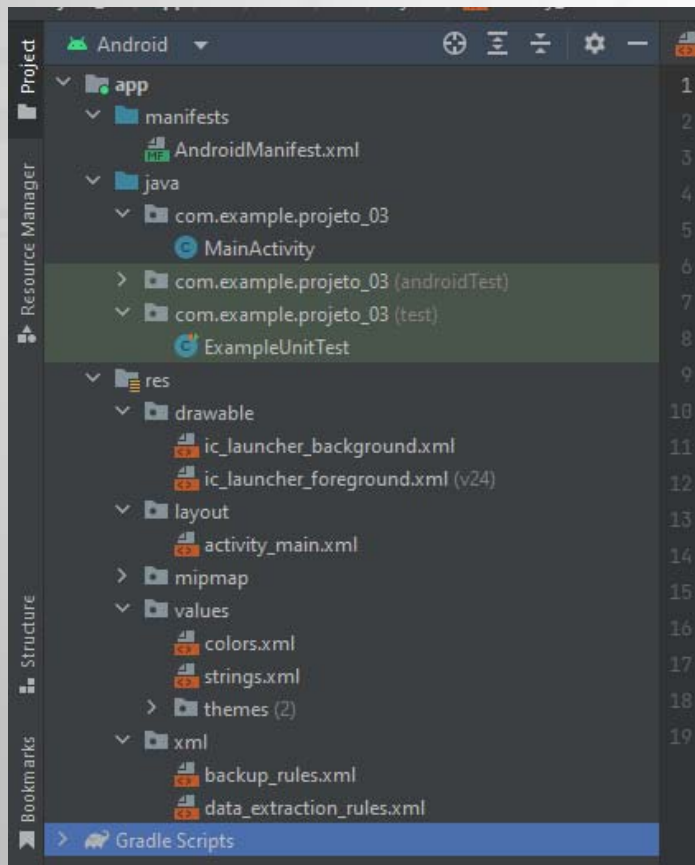


Split: Mostra o Código e o Design



Seleciona o código xml do componente

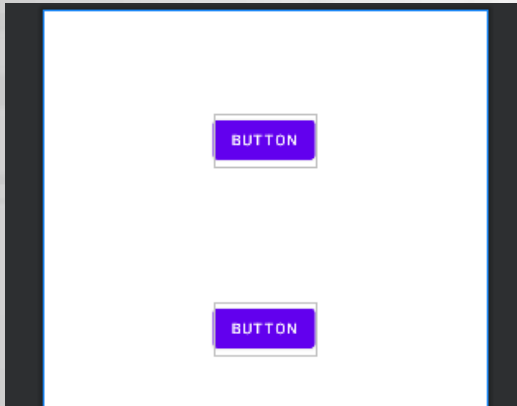
Perceba que ao selecionar o elemento da interface, o código .xml, correspondente, também é selecionado



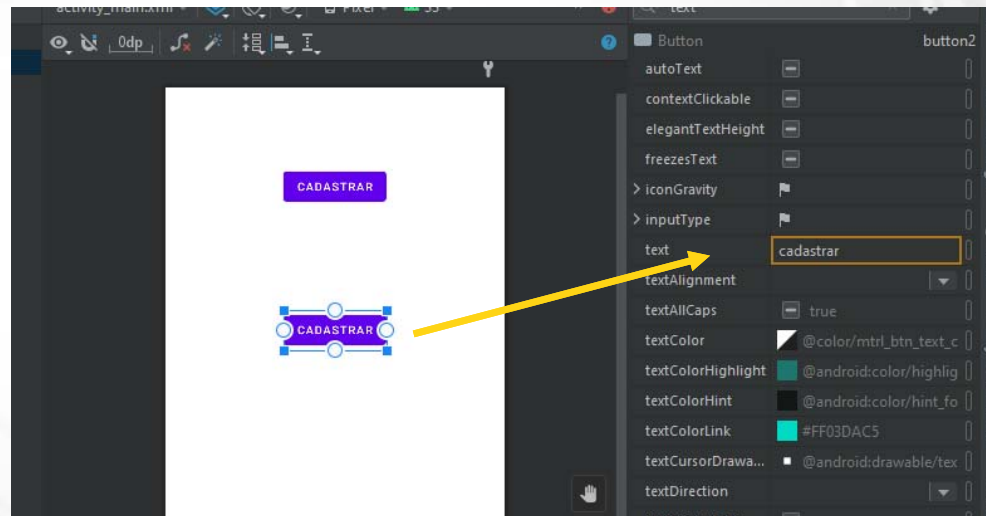
Observe que muitos arquivos do seu projeto usa arquivos xml
Na pasta **res** e **manifests** por exemplo você encontra vários desses arquivos



Estrutura de um projeto Android



Adicione 2 Button

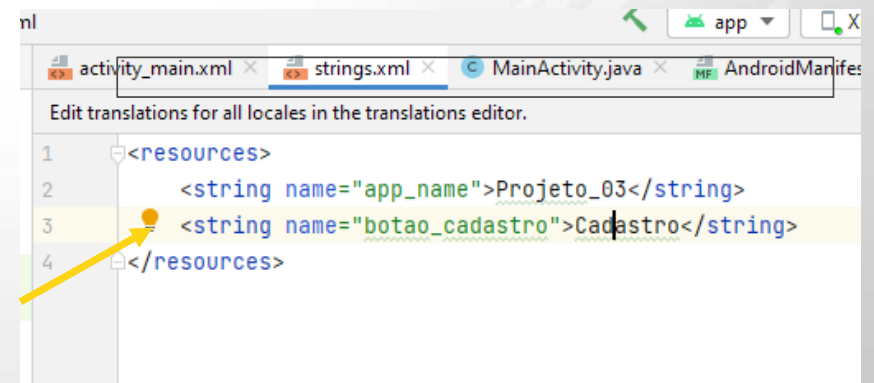
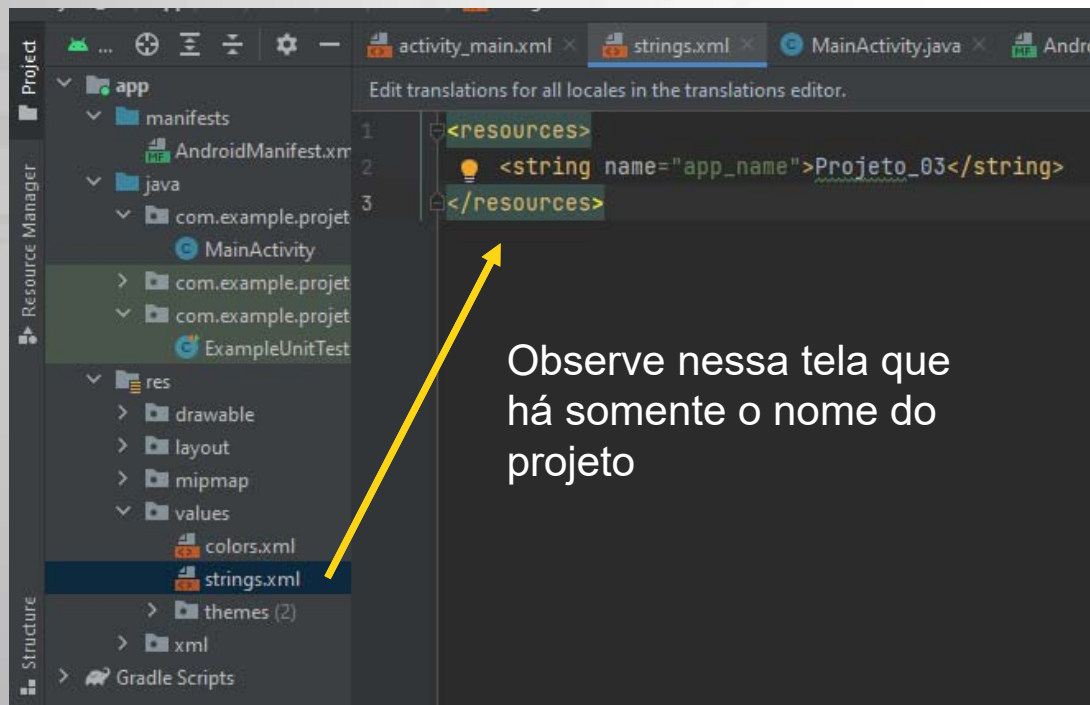


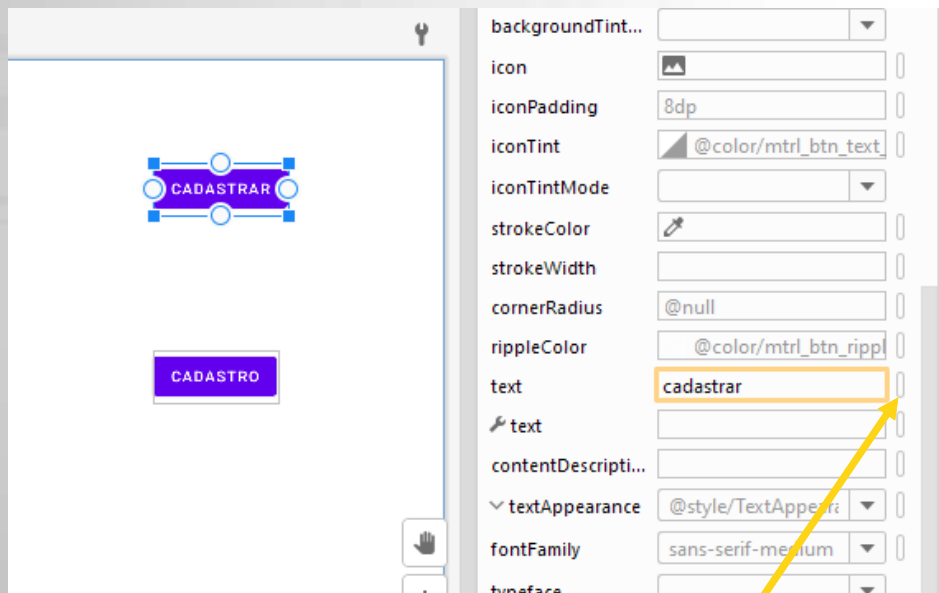
Perceba que você pode alterar o nome do botão simplesmente acessando Attributes



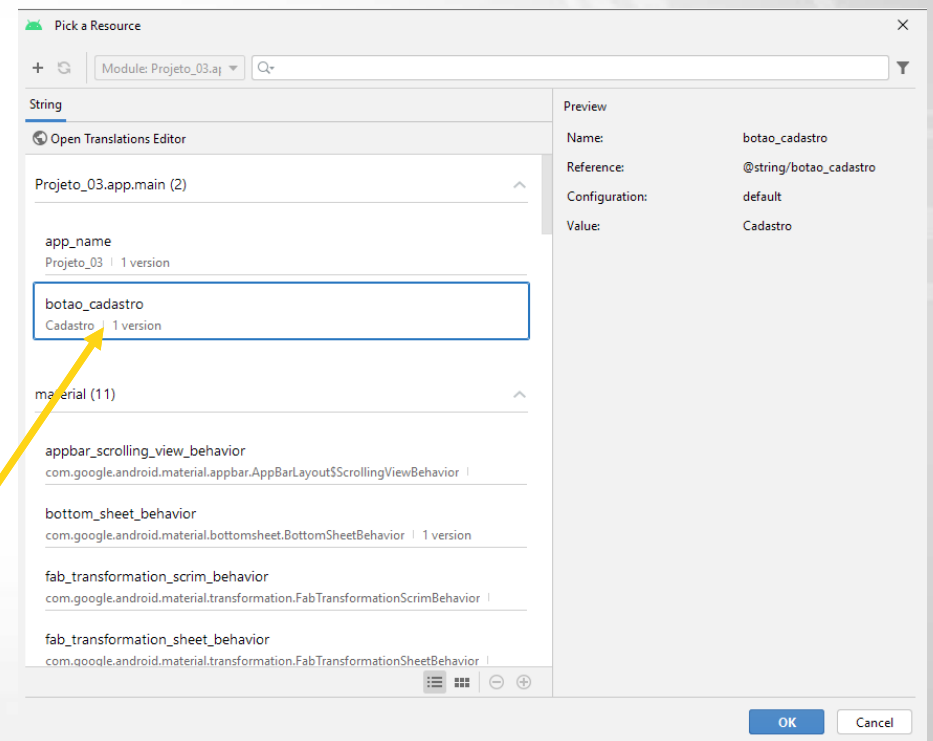
Padronizar o nomes dos botões

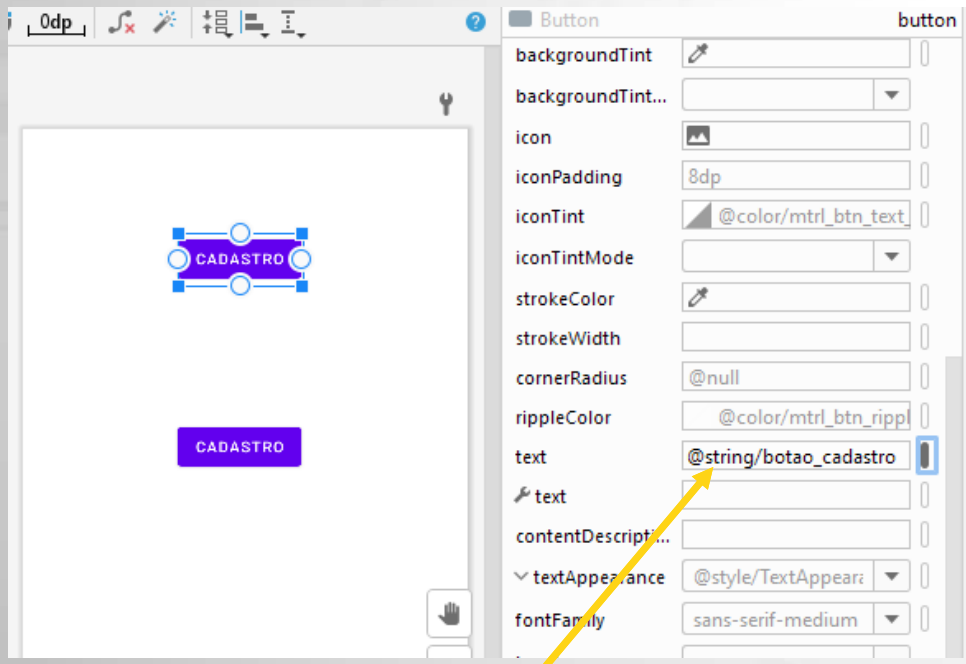
Para criar um padrão de texto de botões basta escrever no arquivo string.xml.



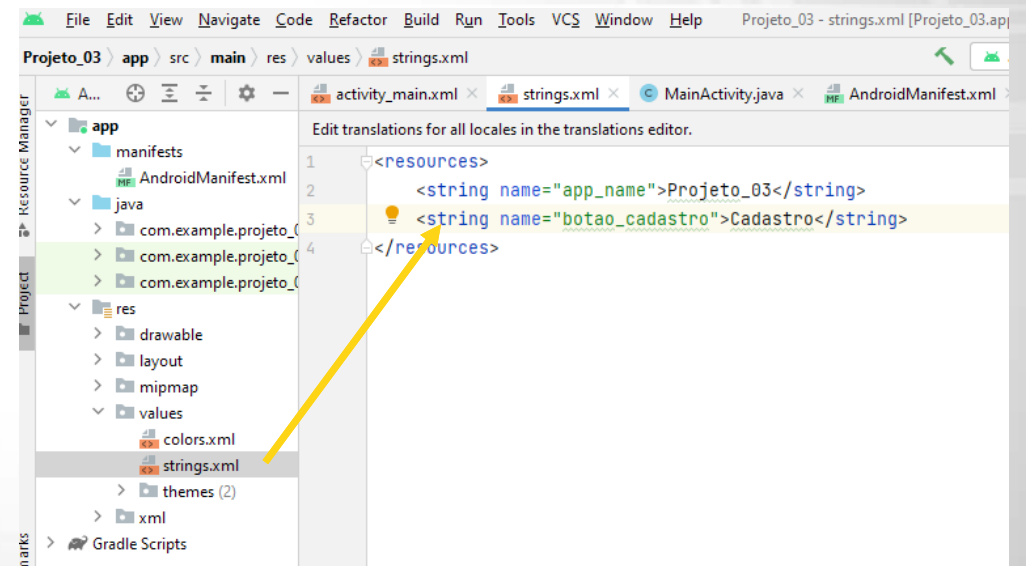


Clique no botão lateral a caixa de texto e procure o nome que você acabou de criar





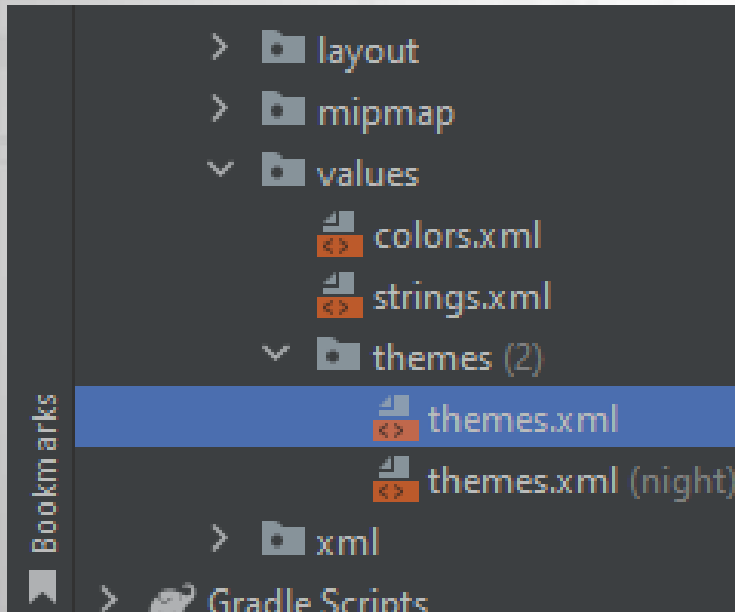
Observe que o nome faz referencia a **string.xml**



A mesma coisa você pode fazer para definir uma **cor de fundo**



Criando um Layout para o botão



Antes de tudo acesse o arquivo themes.xml e altere a definição do tema.

```
parent="Theme.MaterialComponents.DayNight.DarkActionBar"
```

Para

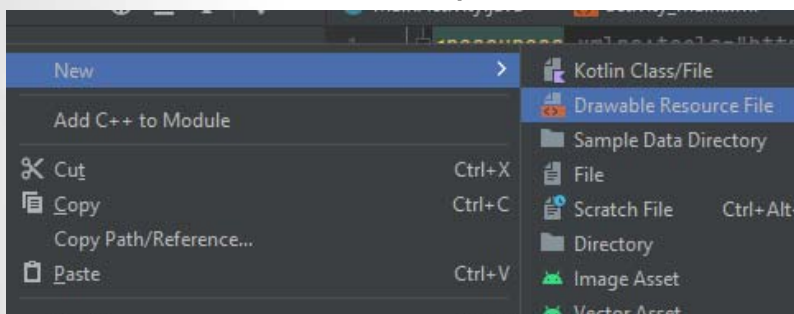
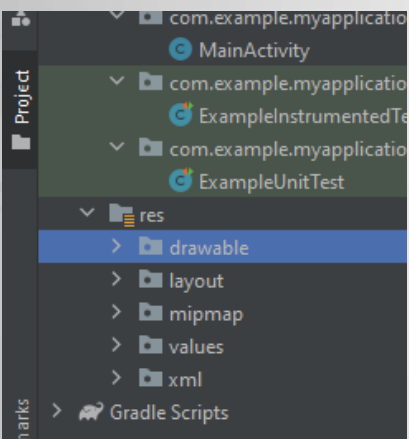
```
parent="Theme.AppCompat.Light.DarkActionBar">
```

Faça isso nos dois arquivos themes.xml

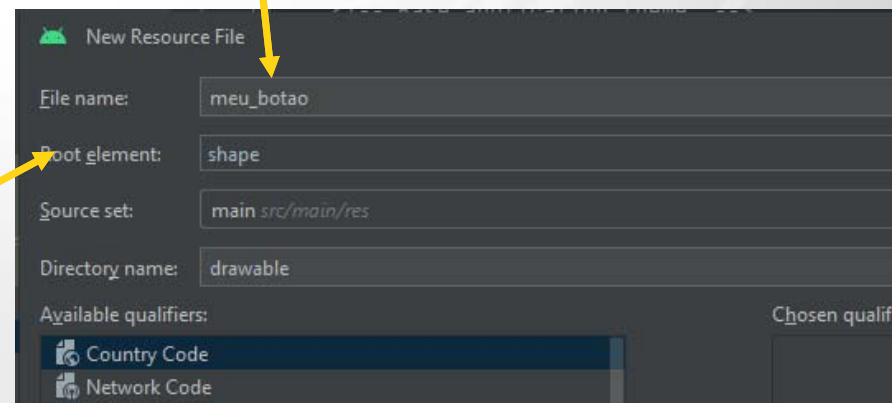


Drawable

Com o botão direito na pasta **drawable** crie um novo arquivo, **Drawable Resource File**



Dê um nome ao seu novo arquivo



Selecione o elemento **shape**

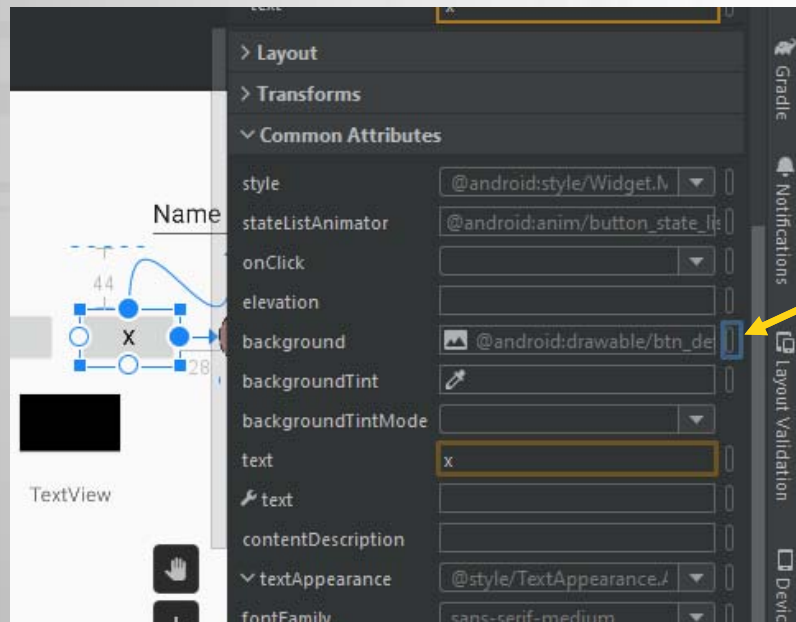


`android:shape="rectangle"`

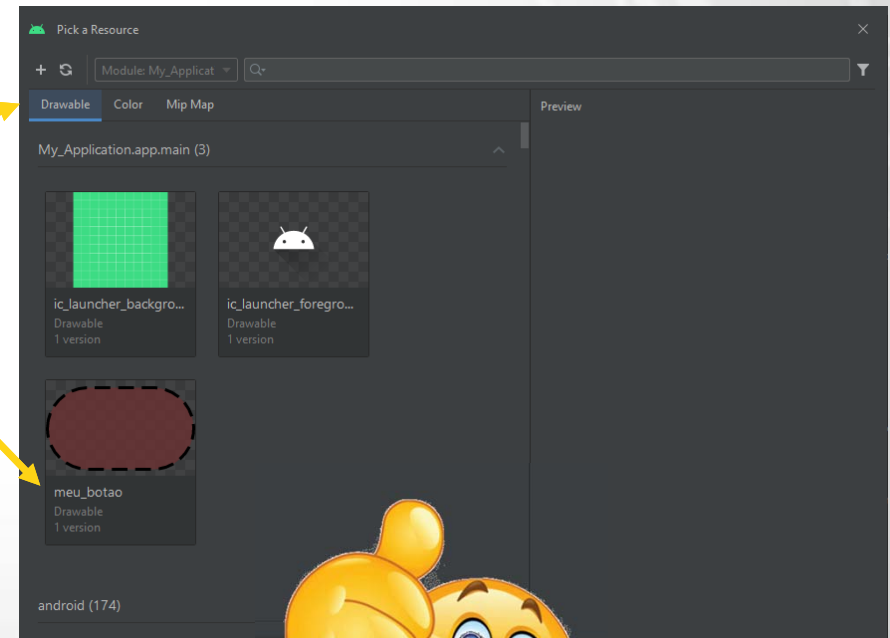
Adicione o shape retângulo na tag.

```
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle" xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#BC43646C" />
    <size android:height="50dp" android:width="20dp"/>
    <corners android:radius="20dp"/>
</shape>
```

Defina a cor, o tamanho e o formato do seu botão, além de muitos outros recursos

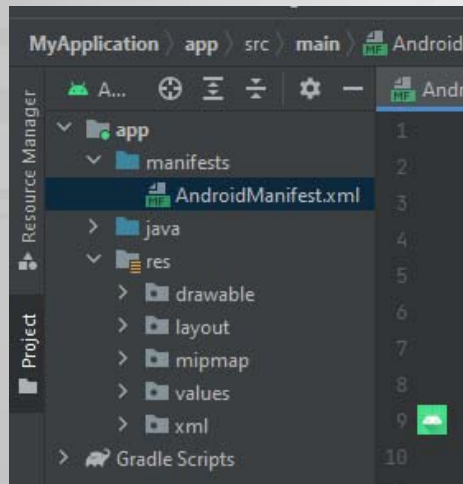


Selecione o
seu botão





Orientação de Tela



```
AndroidManifest.xml x MainActivity.java x activity_main.xml x values\themes.x
8      android:fullBackupContent="@xml/backup_rules"
9      android:icon="@mipmap/ic_launcher"
10     android:label="My Application"
11     android:supportRtl="true"
12     android:theme="@style/Theme.MyApplication"
13     tools:targetApi="31">
14     <activity
15         android:name=".MainActivity" android:exported="true"
16         android:screenOrientation="sensor"
17     <intent-filter>
18         <action android:name="android.intent.action.MAIN"
19             />
20         <category android:name="android.intent.category.LAUNCHER"
21             />
22     </intent-filter>
23 </activity>
```

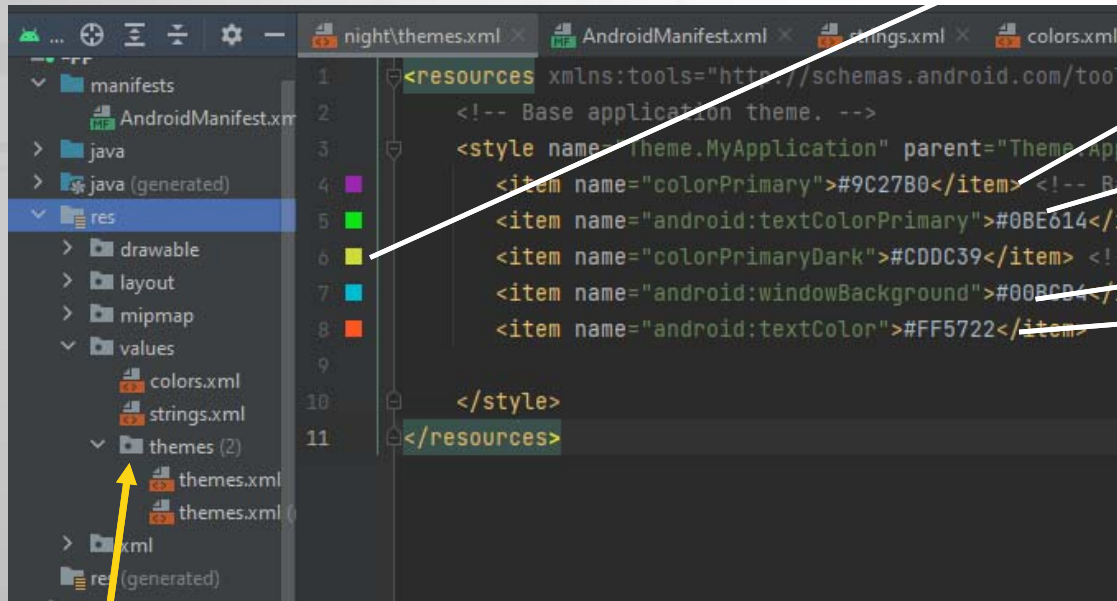
sensor: Se adapta a posição do celular

landscape: Fixa os conteúdos da tela no formato horizontal

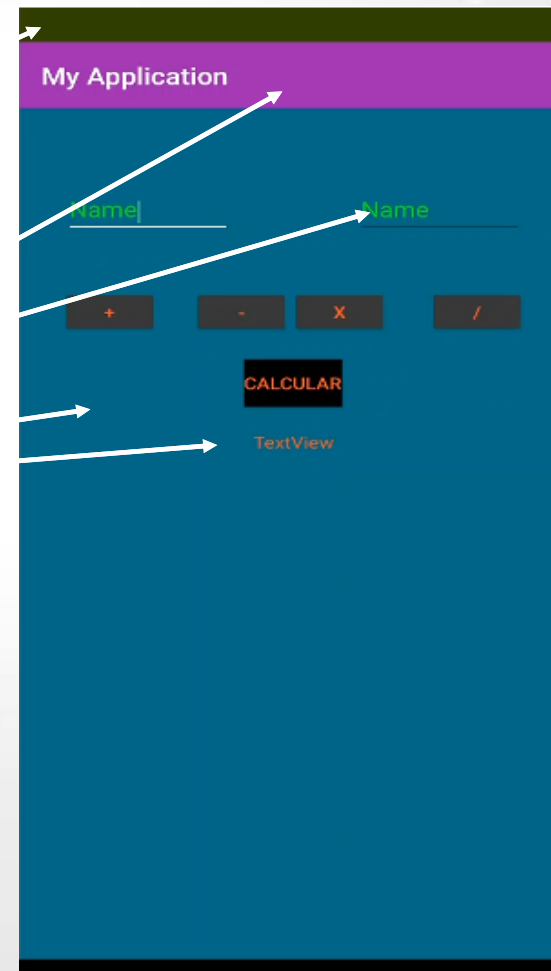
portrait: Fixa os conteúdos da tela no formato vertical

Adicione ao arquivo, AndroidManifest.xml, essa linha de código para definir a orientação da tela.

Temas



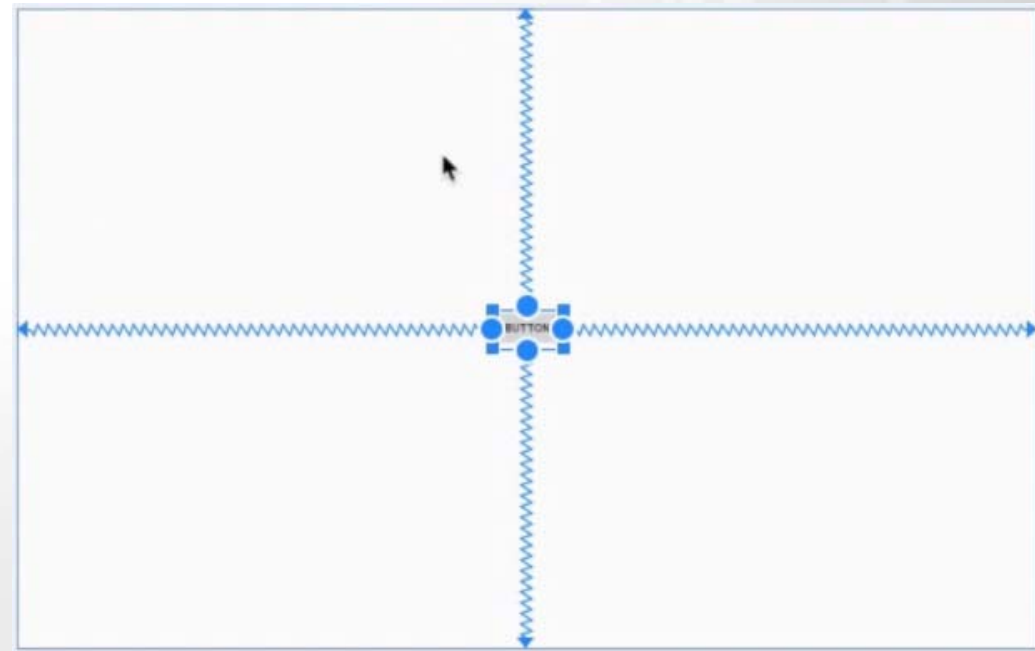
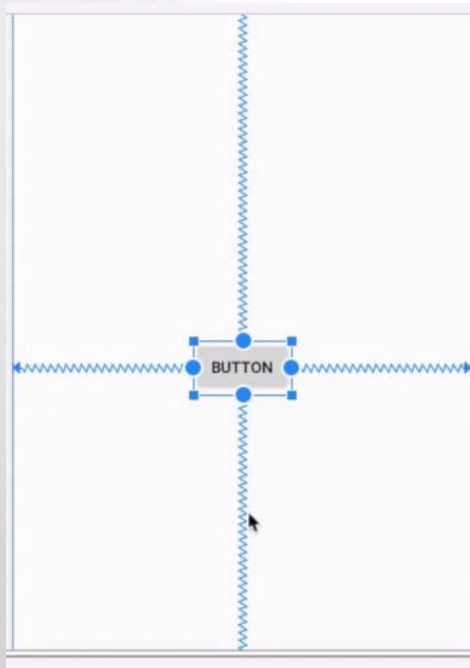
Defina um tema para seu App

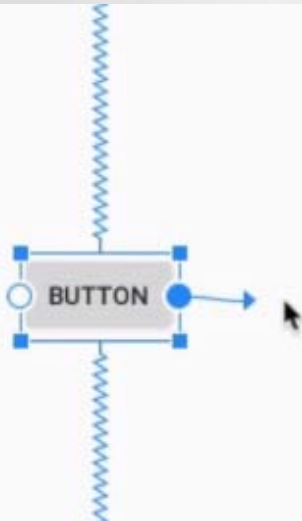




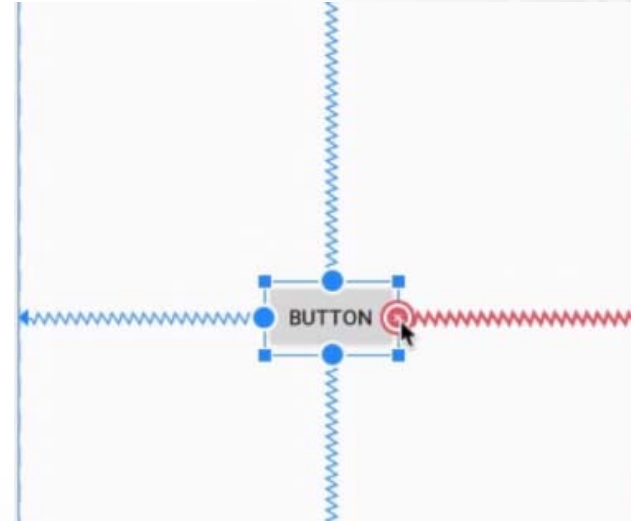
Alinhamento e regras - ConstraintLayout

- Usa regras matemáticas para alinhar e posicionar os componentes, independente dos dispositivos ou orientação da tela.





Pra adicionar uma conexão, click na bolinha lateral e arraste para o local que deseja.



Pra Excluir uma conexão, click na bolinha lateral pressionando a tecla **Ctrl**

As conexões podem ser feitas tanto nas laterais como em outros componentes

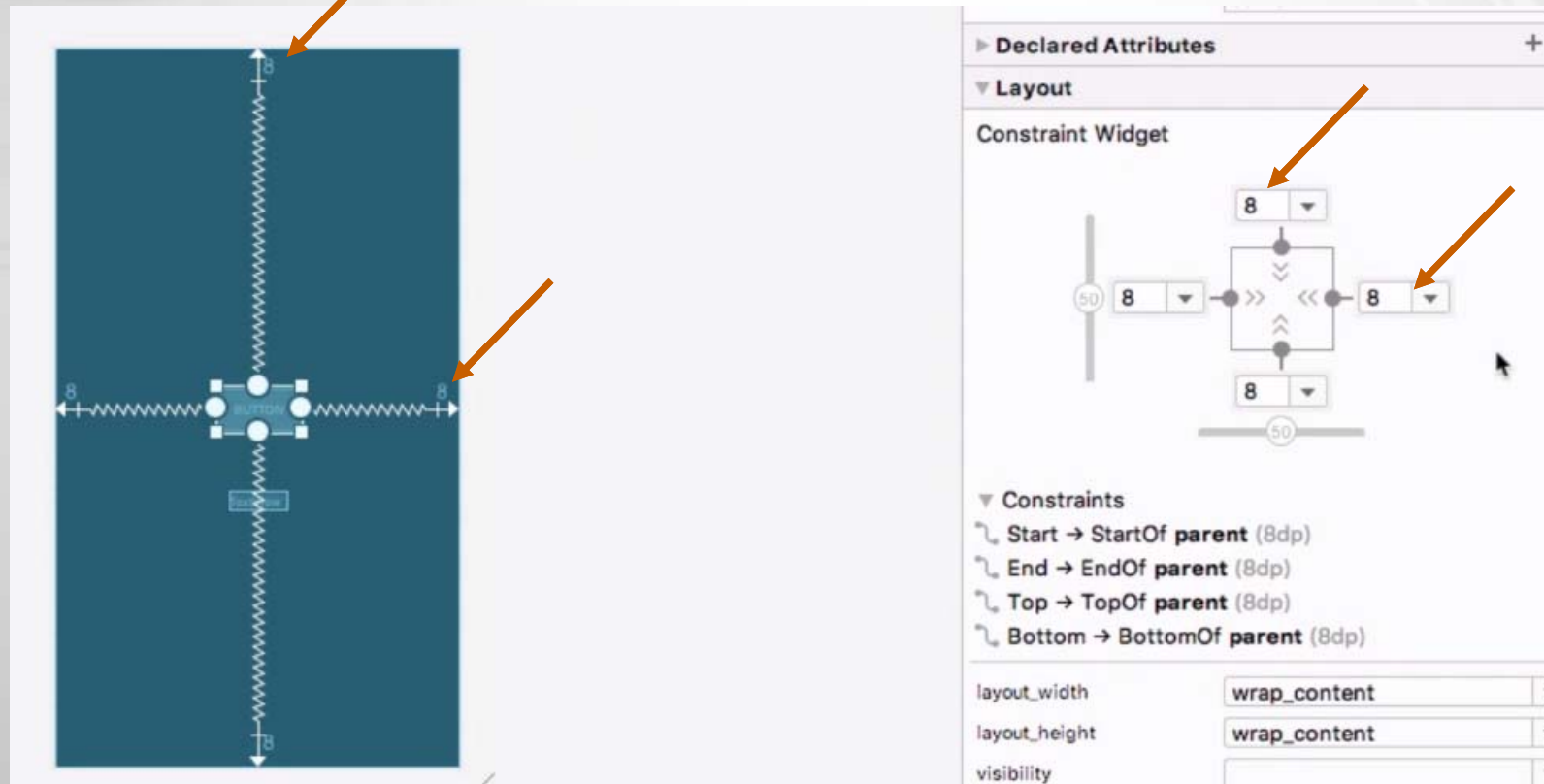


```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

Perceba que ao adicionar as regras de alinhamento são criados os códigos de referencia de conexões.



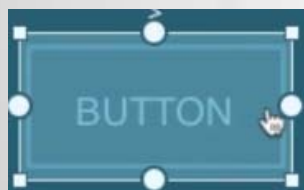
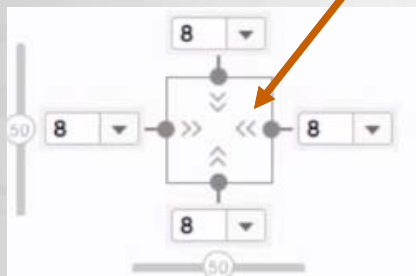
Espaçamentos nas laterais





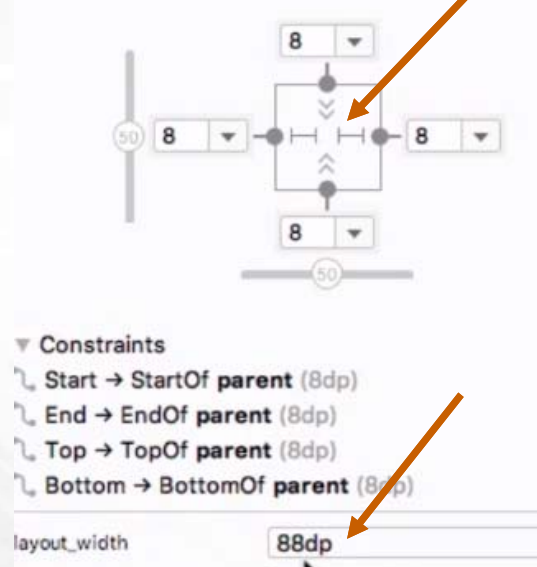
Espaçamento





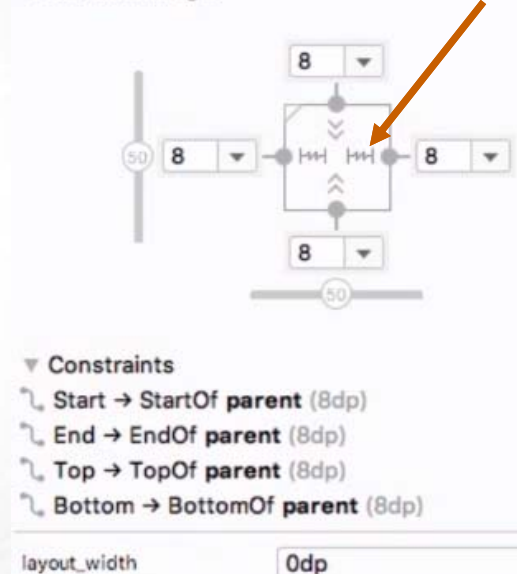
O Botão se adapta ao tamanho do texto

Constraint Widget



Defini um tamanho fixo para o elemento

Constraint Widget



O elemento se adapta ao tamanho da tela