



App NETFLIX





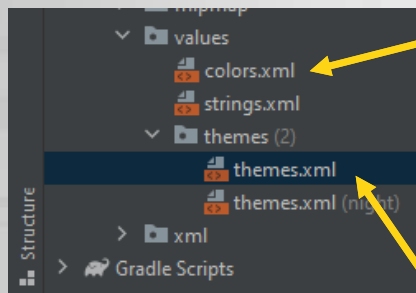
- Vamos criar uma interface parecida com essa.
- A intenção dessa aula é aprender a utilizar alguns recursos como: Cores, Botões, Menus etc,.
- Iremos apenas montar a interface profissional.





Configurando as Cores

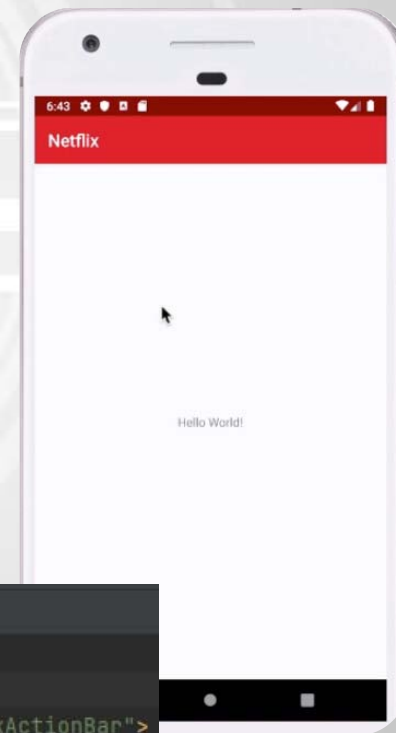
Defina as cores de cada elemento da interface



```
colors.xml
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <resources>
4      <color name="barraStatus">#BB1A11</color>
5      <color name="barraTitulo">#E62019</color>
6      <color name="fundoTela">#FF393939</color>
7  </resources>
```

Em seguida, adicione as cores ao tema do seu App

```
night/themes.xml
1  <resources xmlns:tools="http://schemas.android.com/tools">
2      <!-- Base application theme. -->
3      <style name="Theme.NetFlix" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
4          <item name="colorPrimaryDark">@color/barraStatus</item>
5          <item name="colorPrimary">@color/barraTitulo</item>
6          <!-- Customize your theme here. -->
7      </style>
8  </resources>
```





Dica de cores

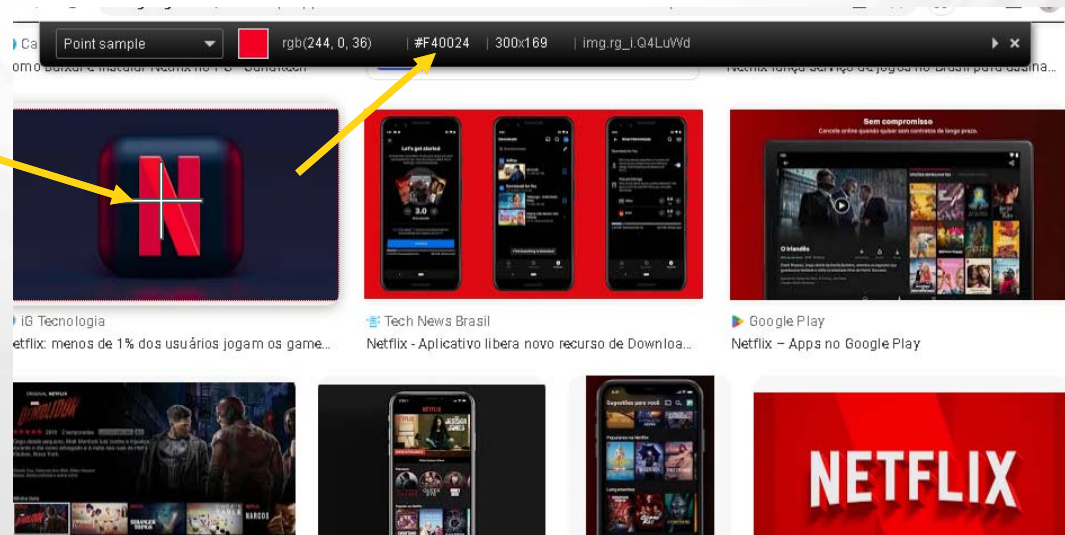
Você pode instalar a extensão, ColorZilla, no seu navegador Chrome



Em seguida você pode acessar alguns modelos de cores App e copiar essa cores.

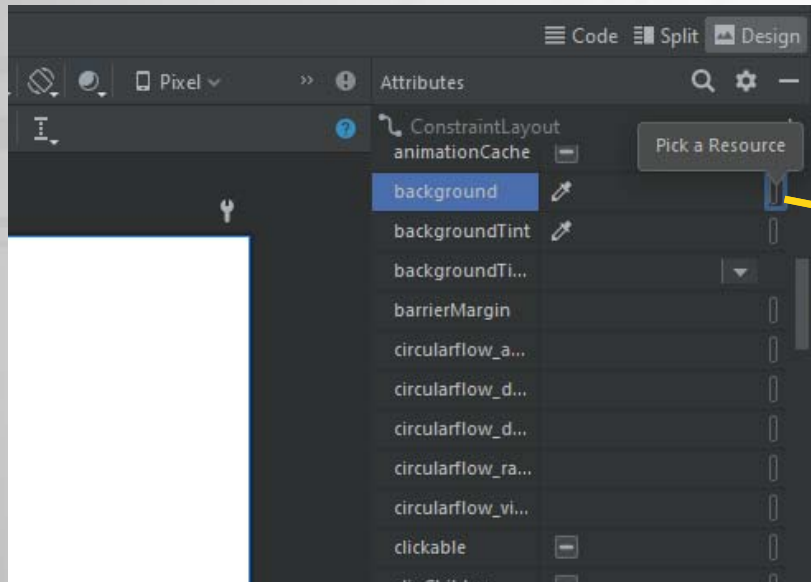
Selecione a cor desejada e copie o código gerado pelo navegador.

Exemplo: #393939

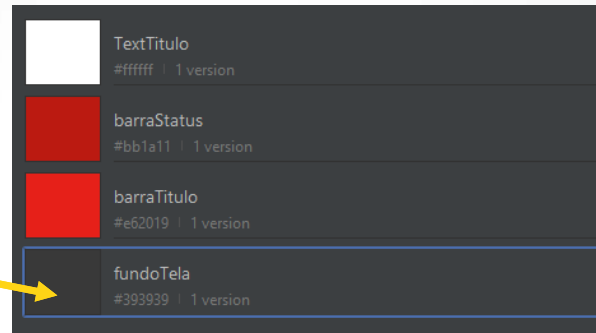




Cor de Fundo e texto

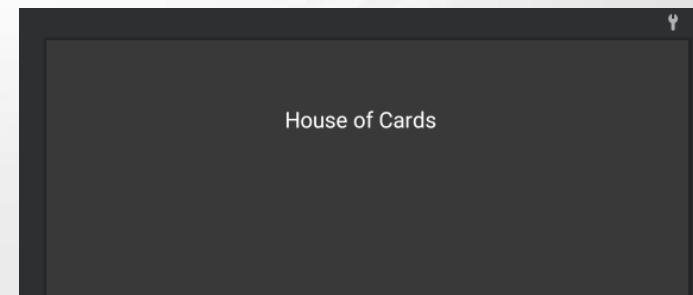


Adicione a color de fundo criado anteriormente em colors.xml



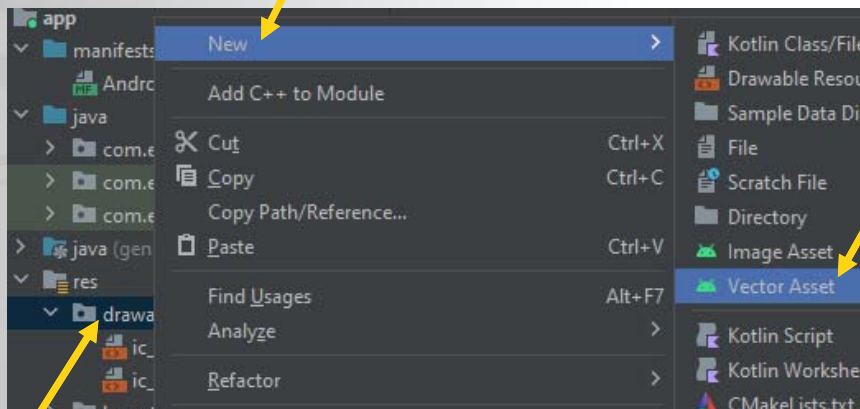
Para mudar a cor do Texto, basta acrescentar ao Themes a seguinte tag:

```
<item name="android:textColor">@android:color/white</item>
```

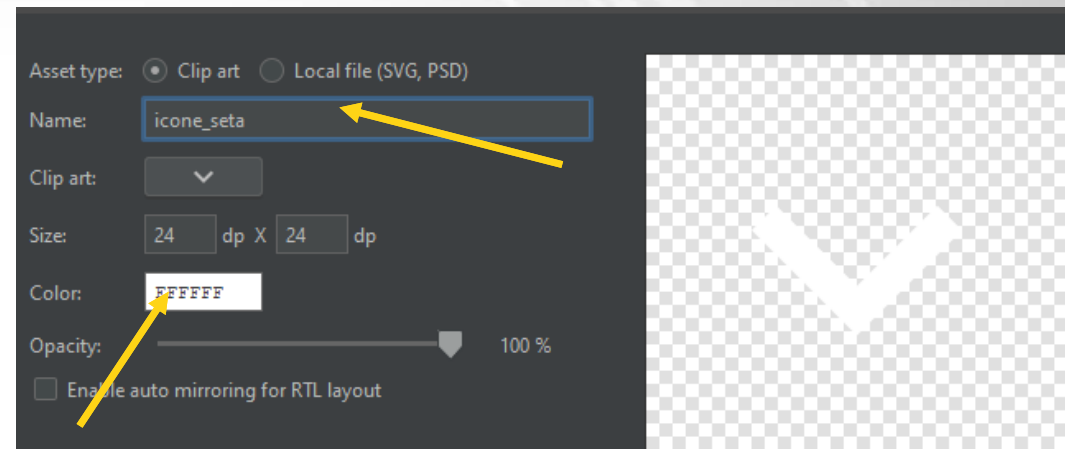




Adicionar ícone a cima do texto



**Crie um drawable: Vector Asset.
Pesquise por arrow e selecione o
ícone correspondente ao nosso App**

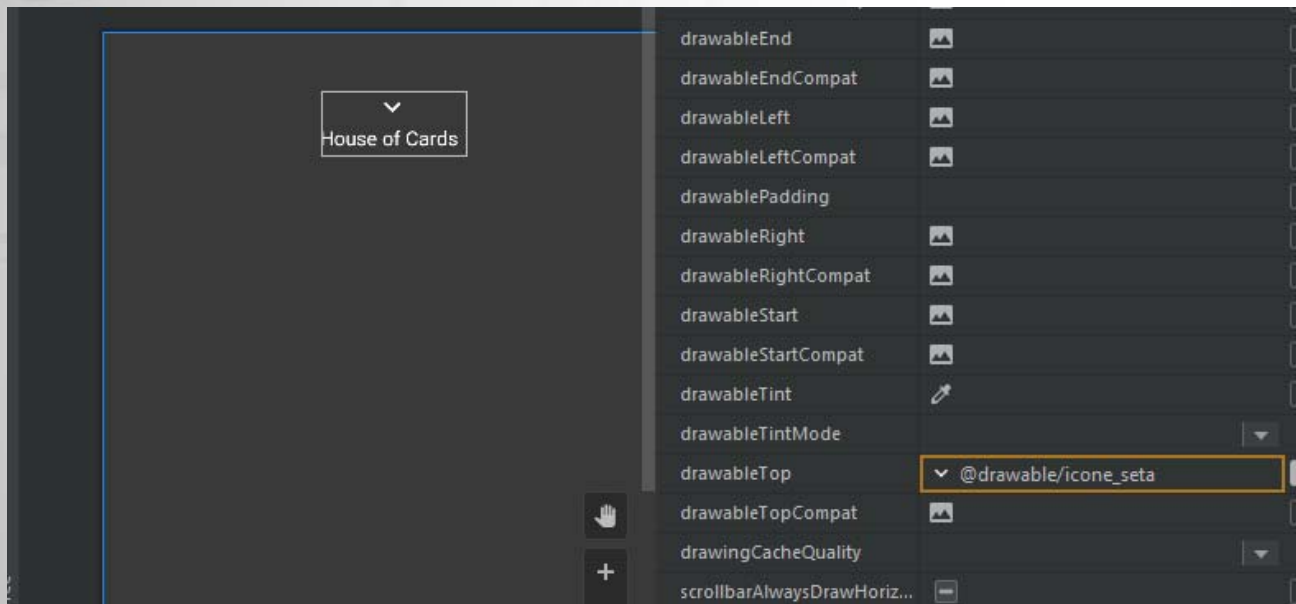


**De um nome ao seu ícone e
escolha a cor Branca
#FFFFFF**

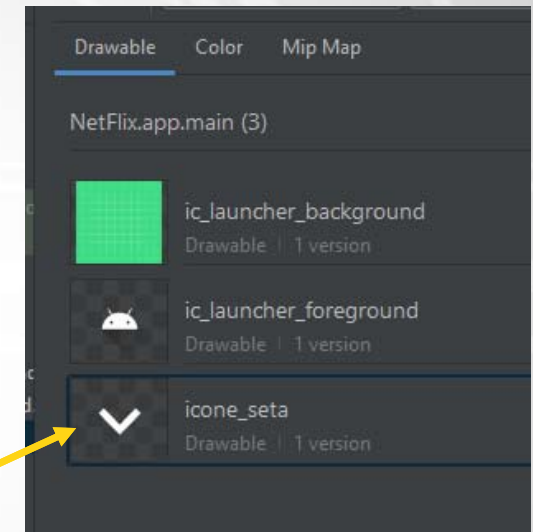
```
<vector android:height="24dp" android:tint="#FFFFFF"
    android:viewportHeight="24" android:viewportWidth="24"
    android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/andr
    <path android:fillColor="@android:color/white" android:pathData="M7.41,8.59
</vector>
```



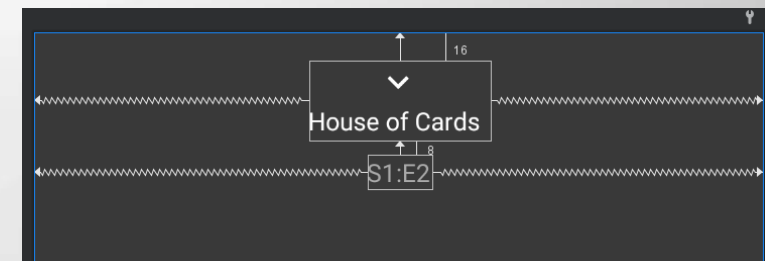
Adicionar ícone a cima do texto



Selecione o atributo **drawableTop** e escolha o ícone criado anteriormente

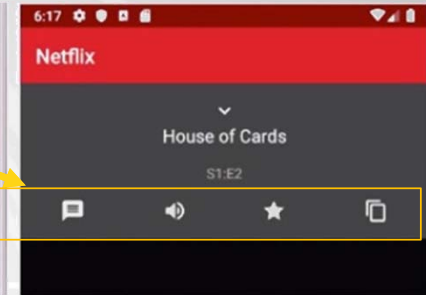


Em seguida conect o textView nas laterais e topo do **parent**

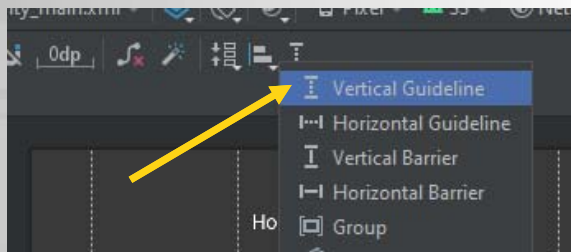




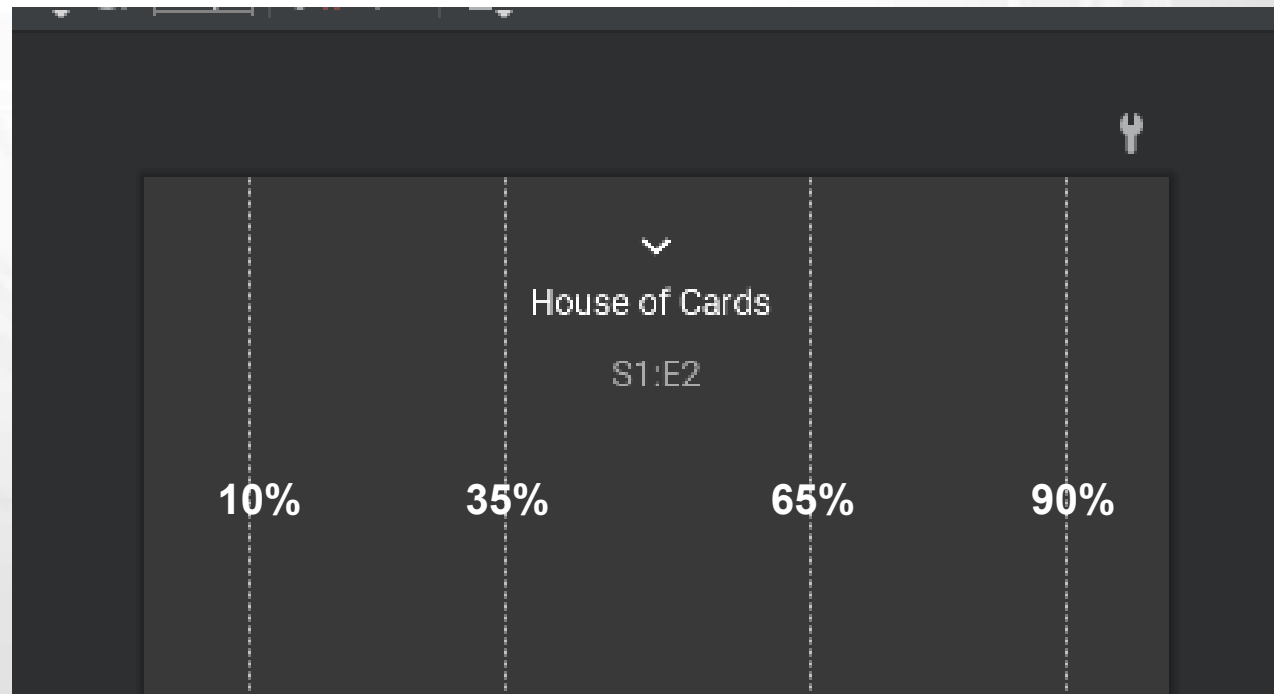
Adicionando ícones e Guideline



Guideline: Pontos de referencia do seu App

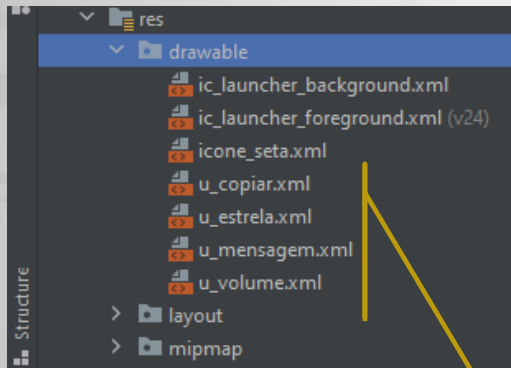


Clique para adicionar um Guideline

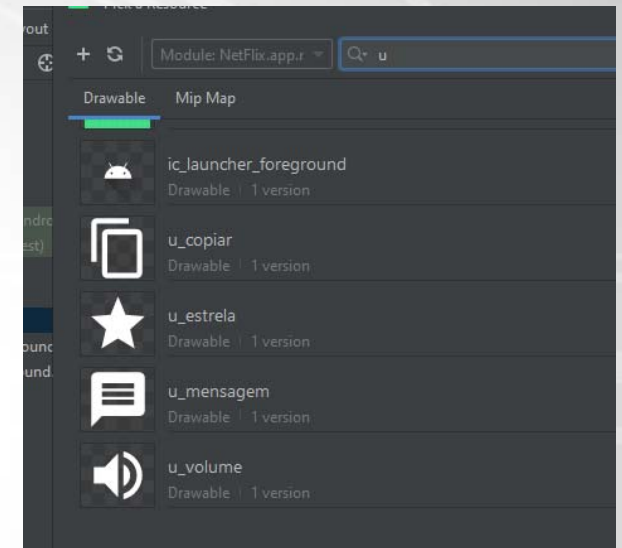
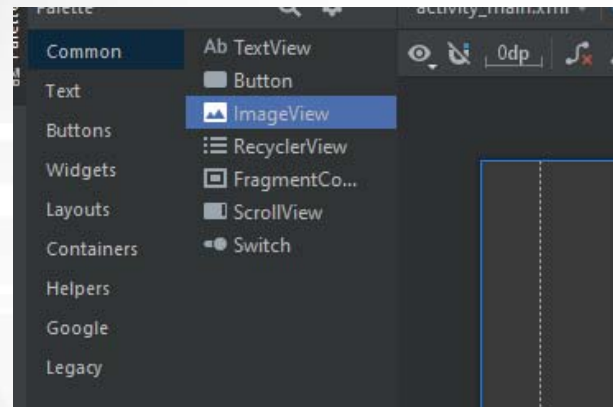




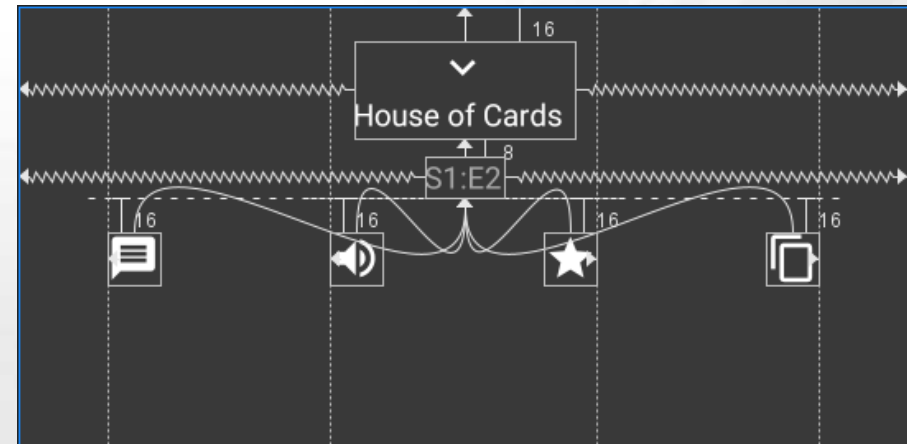
Criando os ícones



Adicione os ícones na pasta **drawable**, conforme visto anteriormente

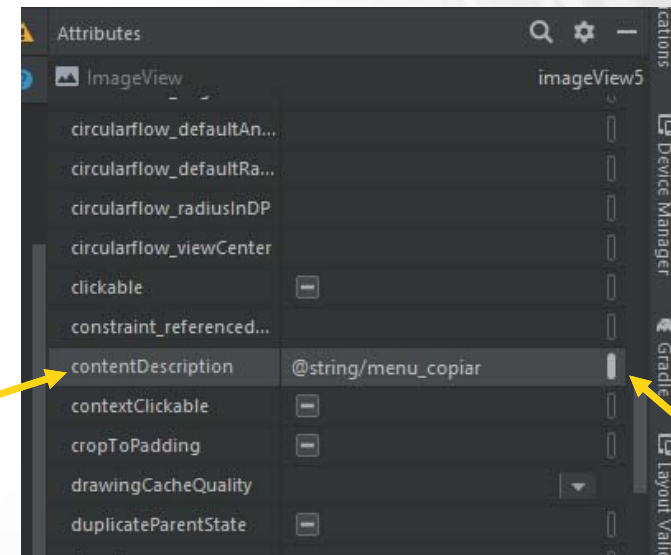
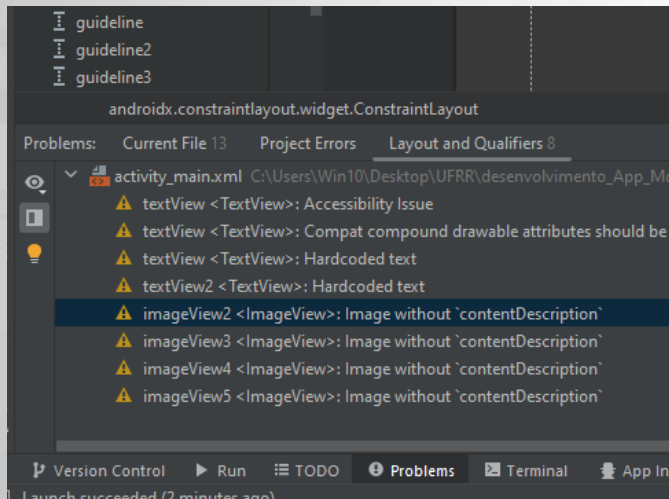


Adicione uma imagem e em seguida escolha **drawable**





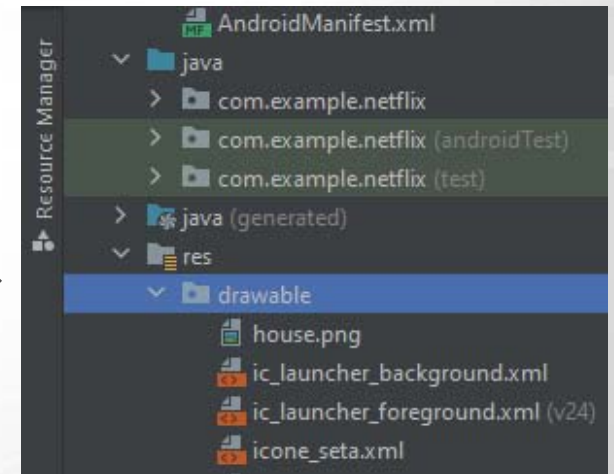
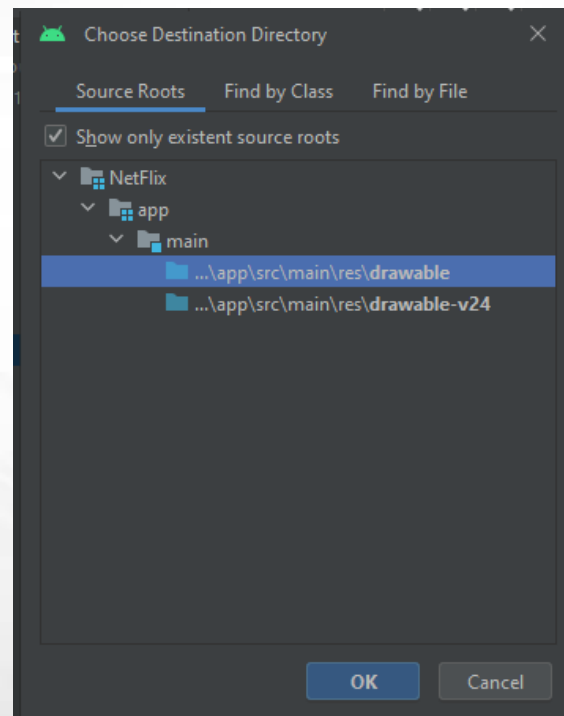
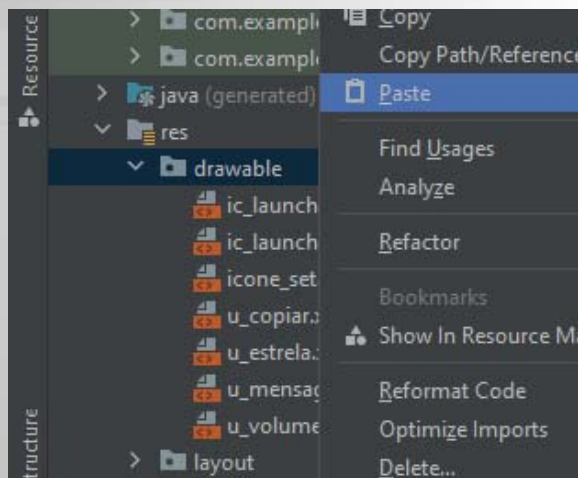
ContentDescription

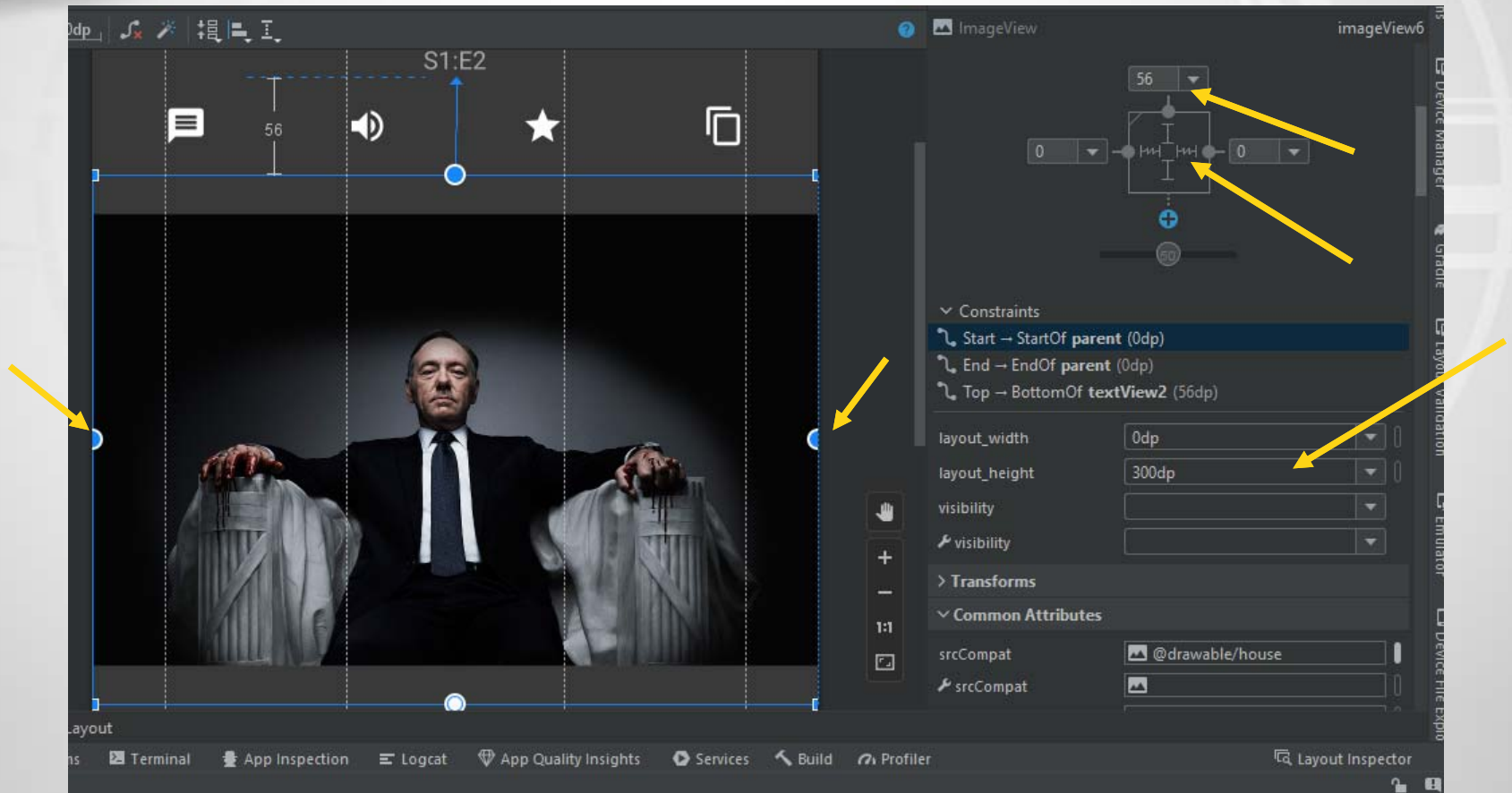


Para eliminar essas mensagem de alertas, crie uma string para adicionar ao **contentDescription**

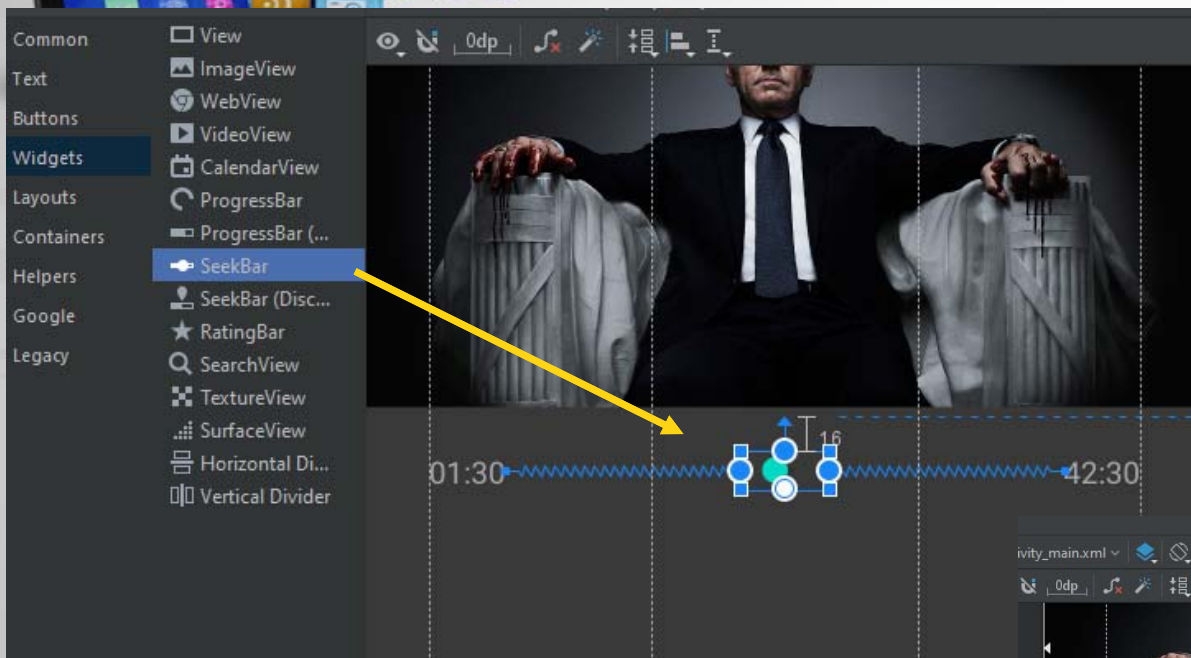


Interface



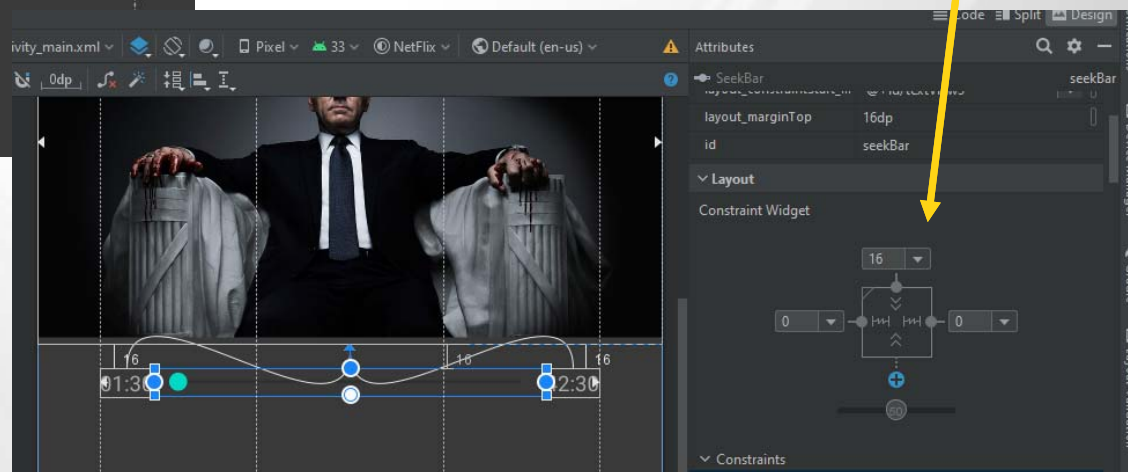


seekBar



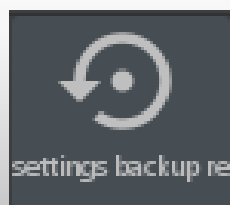
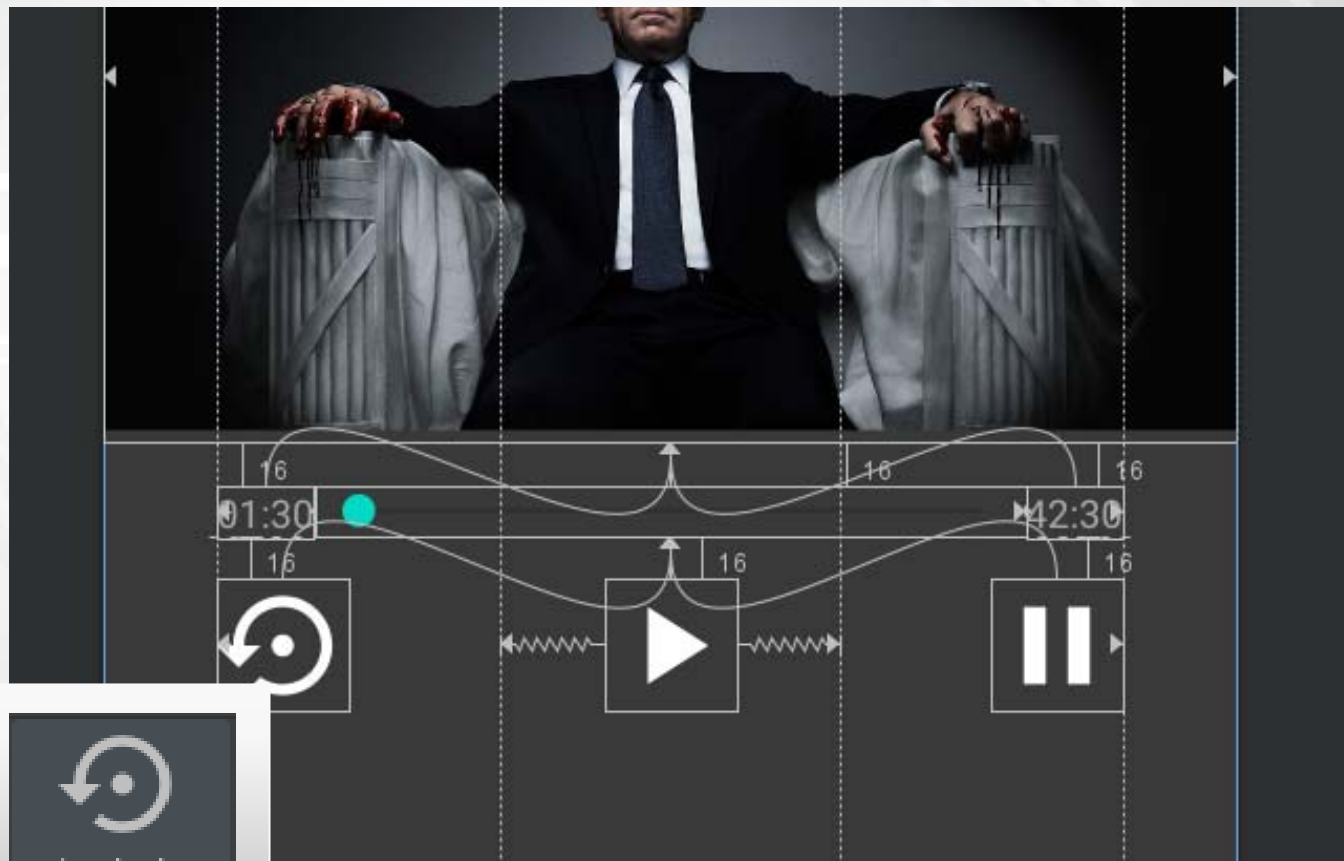
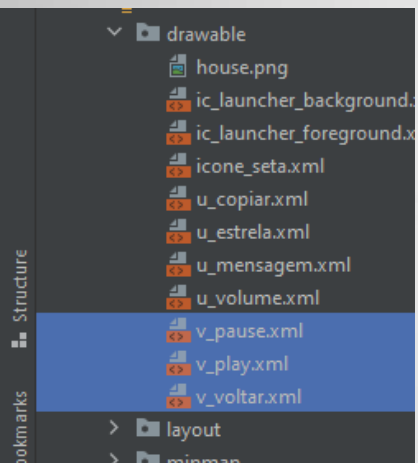
Adicione o SeekBar e alinhe aos textView

Deixe um espaçamento de 16





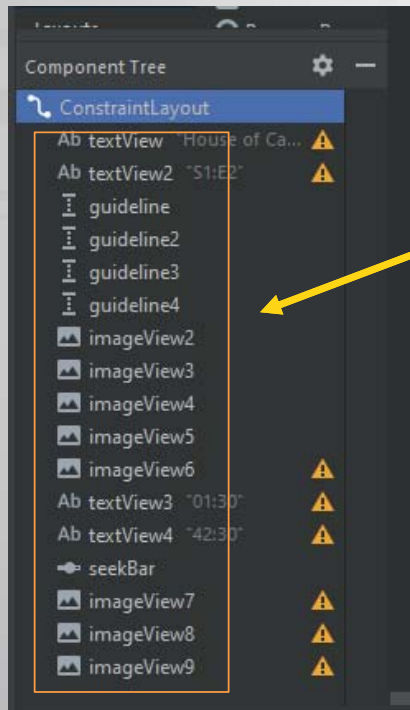
Adicione os outros ícones





ScrollView

Veja que todos os componentes que adicionamos até agora, estão dentro de ConstraintLayout



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
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   android:background="@color/fundoTela"
8   tools:context=".MainActivity">
9
10
11 <TextView
12     android:id="@+id/textView"
13     android:layout_width="102dp"
14     android:layout_height="45dp"
15     android:layout_marginTop="16dp"
16     android:drawableTop="@drawable/icone_sete"
17 />
```



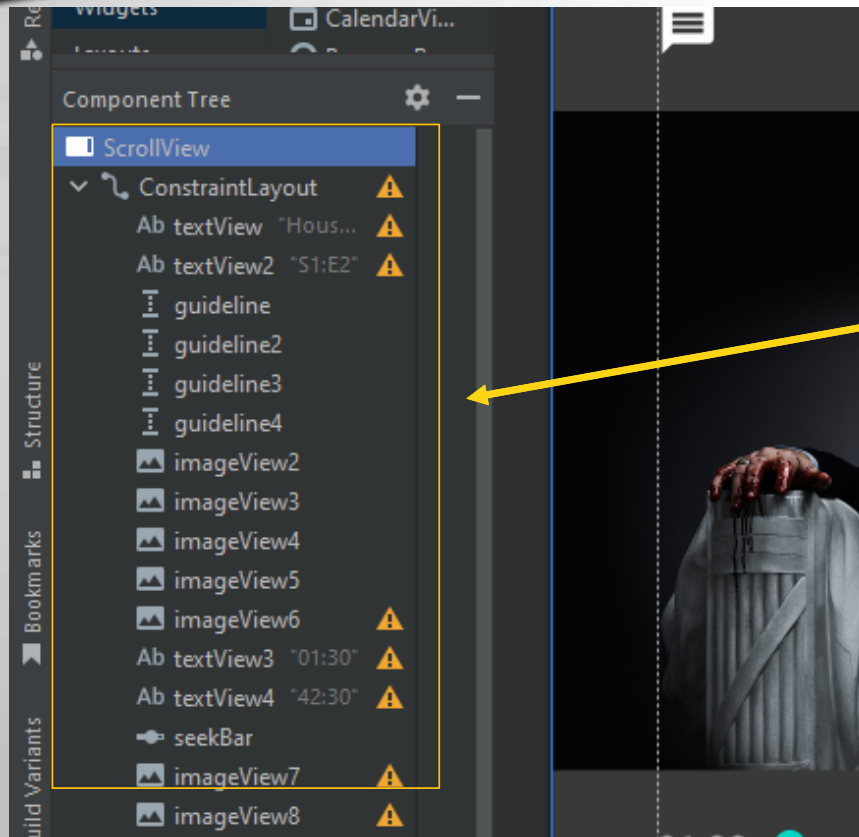
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_height="match_parent"
7     android:layout_width="match_parent">
8
9     <androidx.constraintlayout.widget.ConstraintLayout id="@+id/constraintLayout"
10         android:layout_width="match_parent"
11         android:layout_height="match_parent"
12         android:background="@color/fundoTela"
13         tools:context=".MainActivity">
```

Adicione logo acima a tag ScrollView e remova os xmls e cole dentro de ScrollView

Adicione também a linha: android: fillViewport="true"

```
app:layout_
app:layout_
app:srcCompa

</androidx.constraint
</ScrollView>
```



Perceba que agora tudo está dentro de ScrollView