Sumário

[1. Informativo 2](#_Toc503346555)

[2. Pré-Requisitos 3](#_Toc503346556)

[3. Objetivo 4](#_Toc503346557)

[4. Versão 5](#_Toc503346558)

[4.1. minSdkVersion 5](#_Toc503346559)

[4.2. targetSdkVersion 5](#_Toc503346560)

[5. SplashScreen 6](#_Toc503346561)

[6. Novo Projeto 6](#_Toc503346562)

[6.1. styles.xml 9](#_Toc503346563)

[6.2. AndroidManifest.xml 10](#_Toc503346564)

[6.3. colors.xml 10](#_Toc503346565)

[6.4. activity\_splash\_screen.xml 11](#_Toc503346566)

[6.5. Animação 13](#_Toc503346567)

[6.6. Tela Secundária 17](#_Toc503346568)

[7. Resumo 20](#_Toc503346569)

[8. Referências 21](#_Toc503346570)

# Informativo

Autor(a): Helena Strada

Data de criação: jan/2018

Escola Senai de Informática

CodeXP – Mobile

# Pré-Requisitos

Java (Orientação a Objetos, APIs e Bibliotecas);

Básico de Android (Activites e Intents).

# Objetivo

Entender o funcionamento de uma splashscreen (tela inicial), animações e como construí-la.

# Versão

## minSdkVersion

15

## targetSdkVersion

26

# SplashScreen

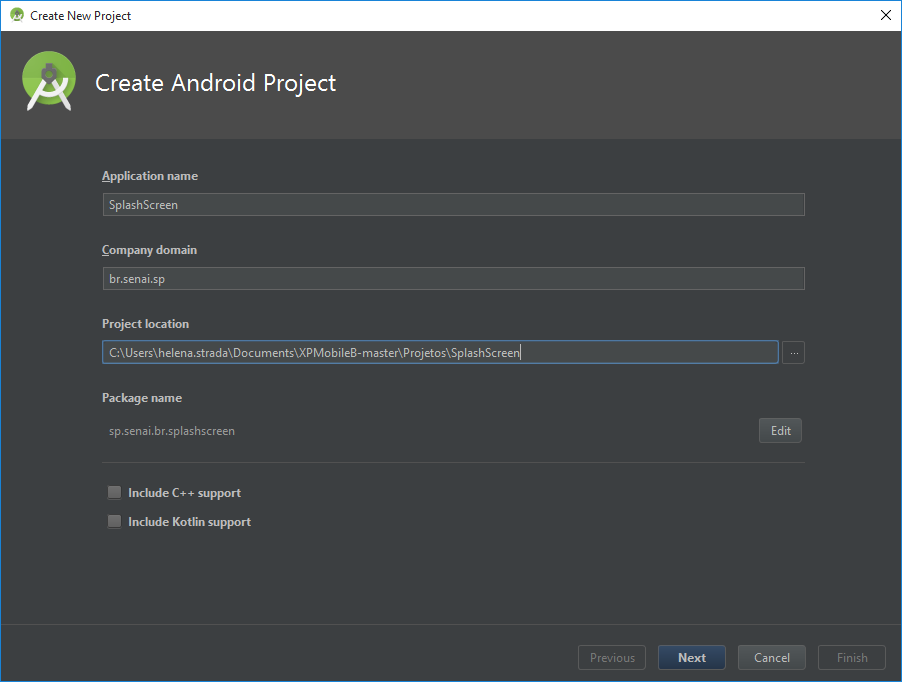
A splashscreen ou tela de apresentação é a tela inicial que aparece ao iniciarmos um aplicativo. Usualmente, as empresas utilizam a splashscreen para mostrar o logo da empresa ou o logo de quem desenvolveu o aplicativo.

Exemplo:



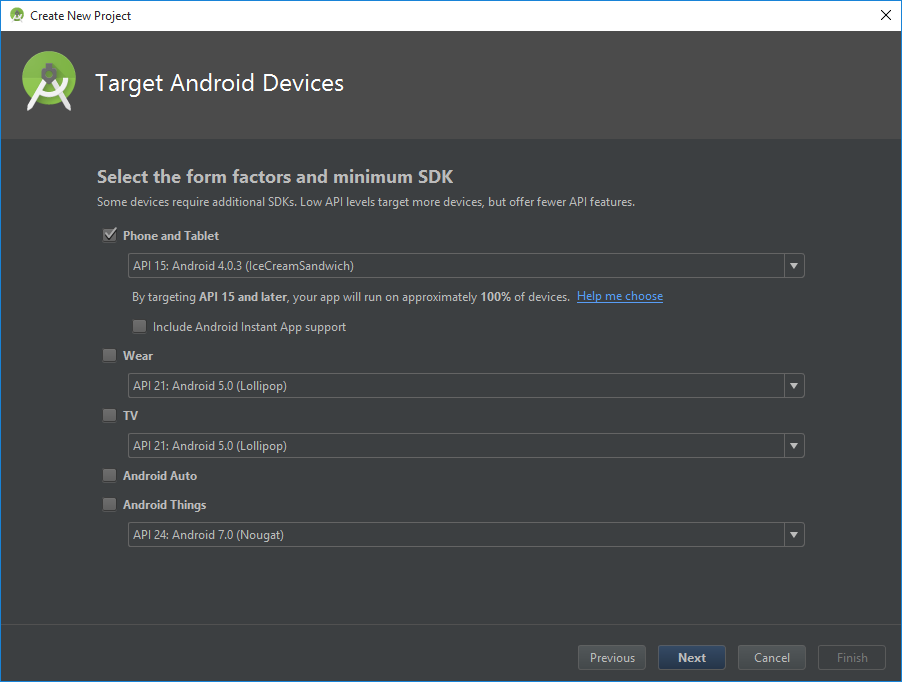
# Novo Projeto

Vamos criar um novo projeto chamado SplashScreen.



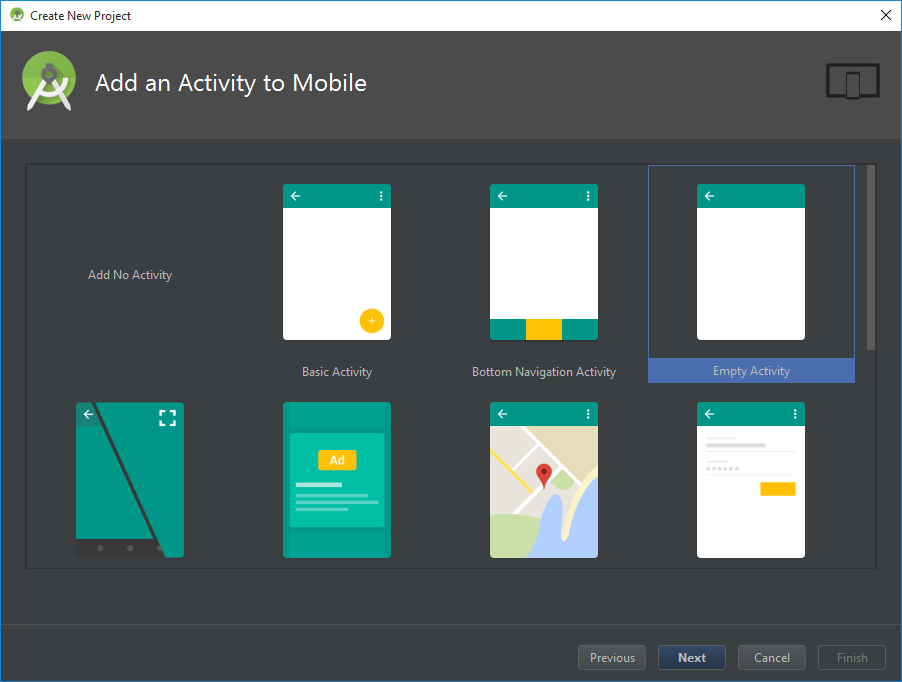
Clicar em “*Next*”.

Escolheremos a API mínima que queremos utilizar.



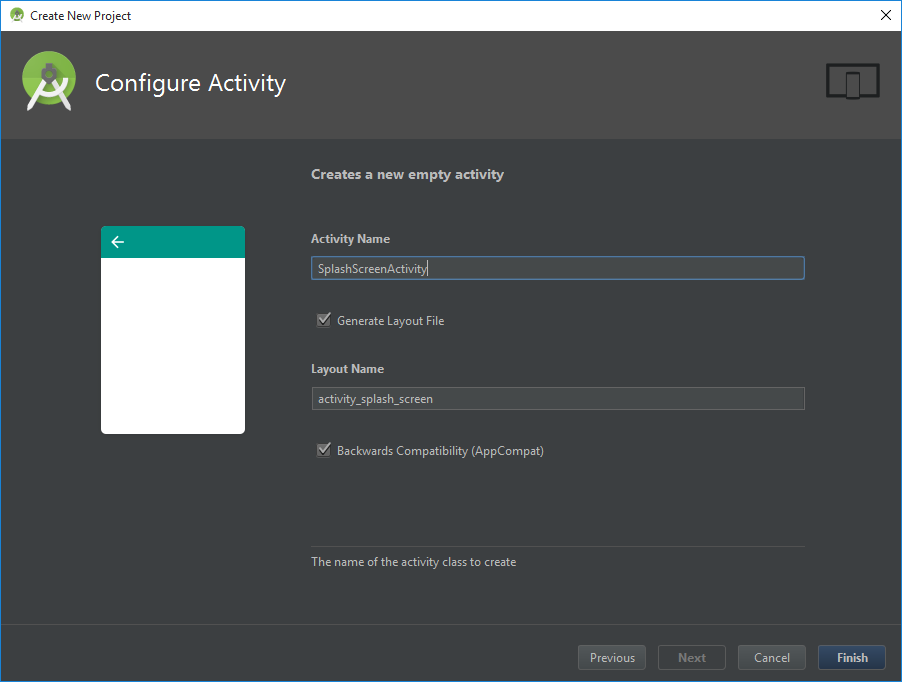
Clicar em “*Next*”.

Vamos manter como *Empty Activity.*



Clicar em “*Next*”.

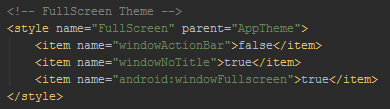
Vamos alterar o nome da nossa activity principal para SplashScreenActivity.



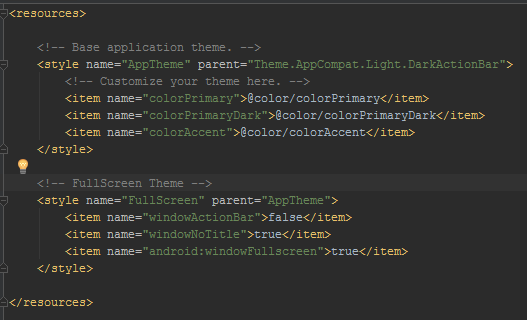
Clicar em “*Finish*”.

### styles.xml

Vamos abrir o nosso arquivo styles.xml e adicionar as nossas alterações:



O código completo deste arquivo:

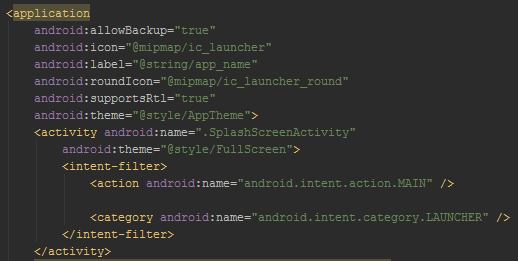


Vamos agora alterar o estilo da nossa tela principal, no nosso AndroidManifest.xml.

### AndroidManifest.xml

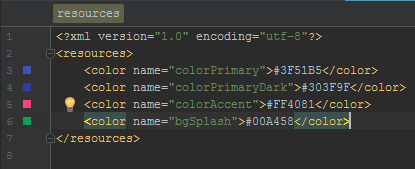
Iremos altera o estilo da nossa activity principal (SplashScreenActivity) para utilizar o tema atual que acabamos de adicionar no nosso arquivo *styles.xml*.

Além disso, iremos alterar a nossa cor de fundo da nossa tela principal.



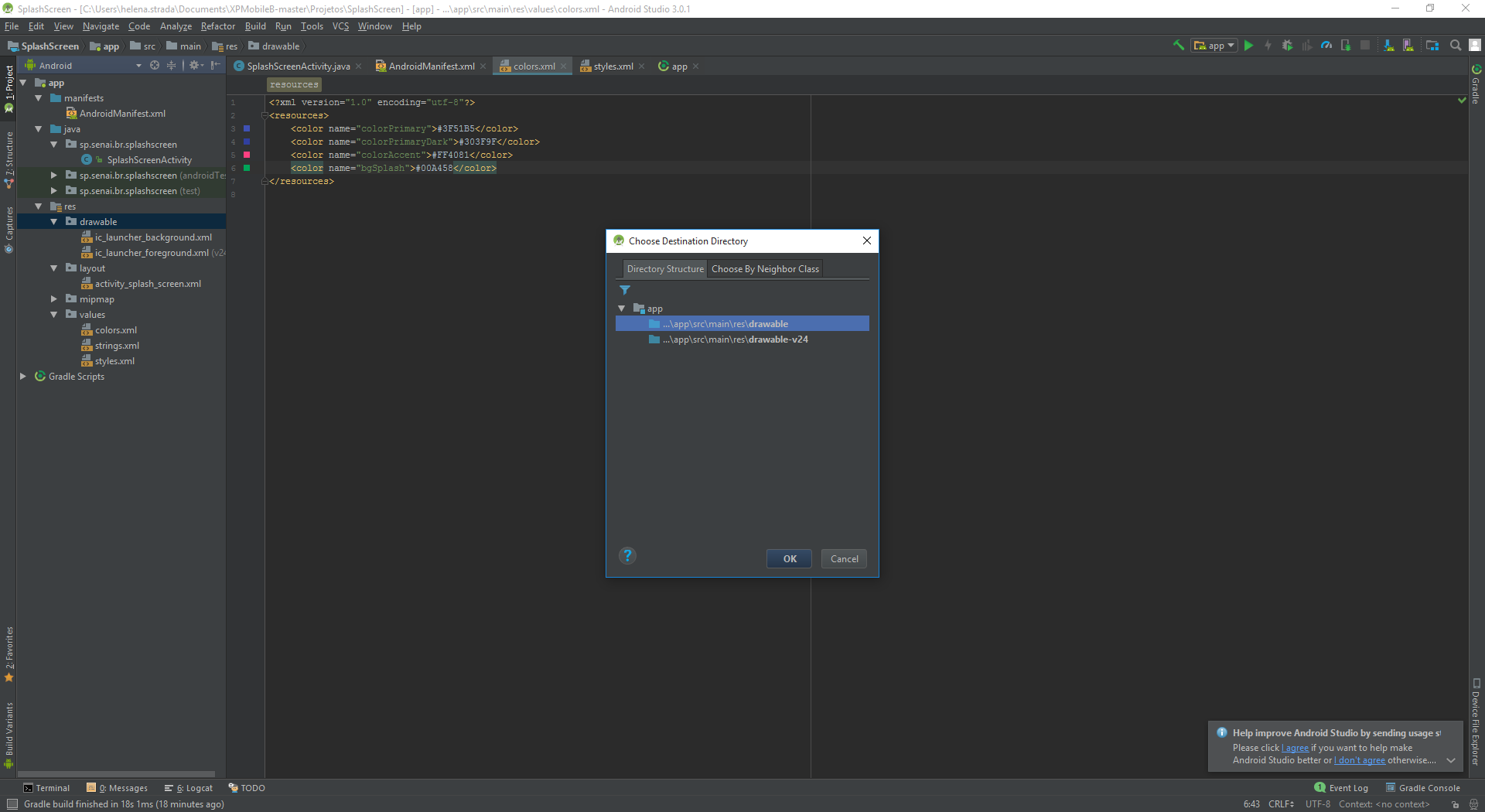
### colors.xml

Iremos colocar uma cor apenas para teste. Neste caso, nosso colors.xml ficará assim:



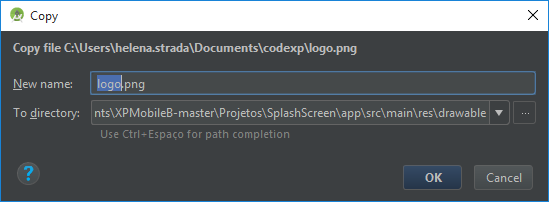
Iremos também, além do fundo, iremos escolher um logo para a nossa tela principal.

Escolha um logo e o inclua na pasta: res -> drawable -> arquivo.png



Clique em “*Ok*”.

Escolha o nome do arquivo que deseja e clique em “*Ok*”.

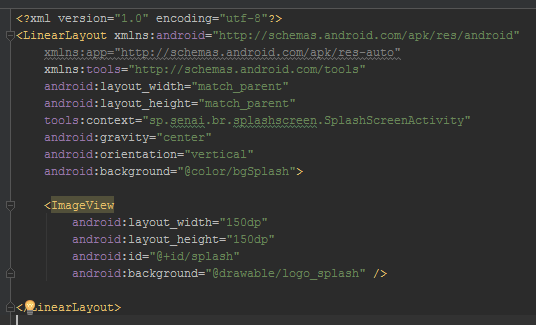


No nosso exemplo, iremos alterar o nome para “*logo\_splash.png*”.

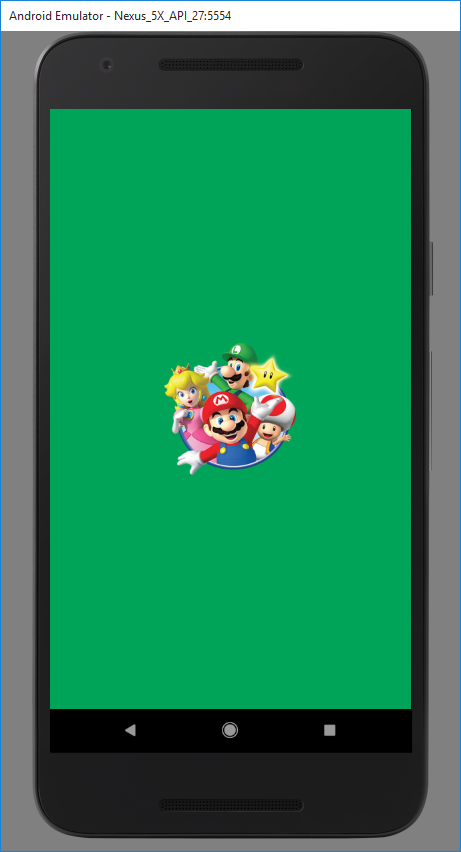
### activity\_splash\_screen.xml

Nós incluímos o logo na nossa aplicação, alteramos a cor, porém, ainda não fizemos a alteração na nossa activity.

Nosso próximo passo será realizar as alterações que seguem:



Atualmente a nossa tela está assim:

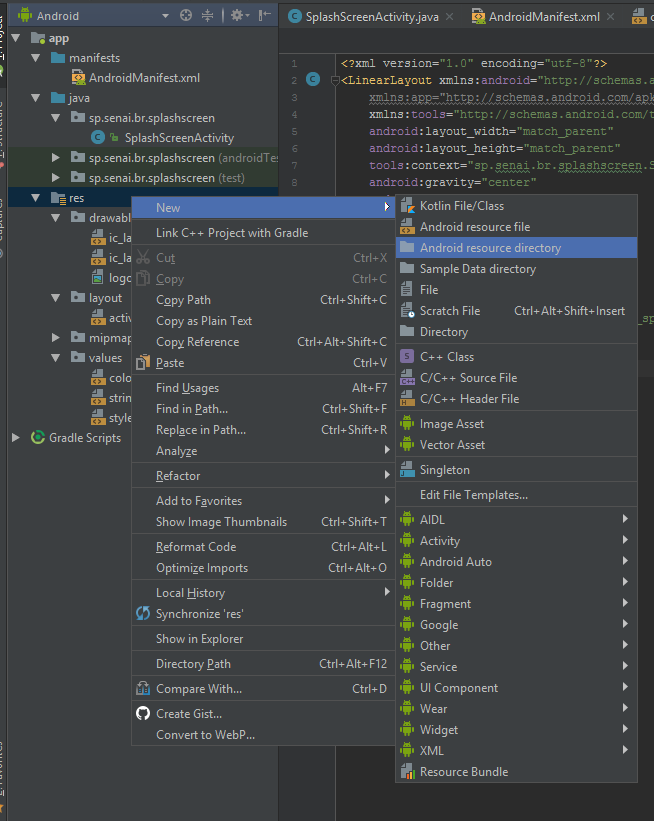


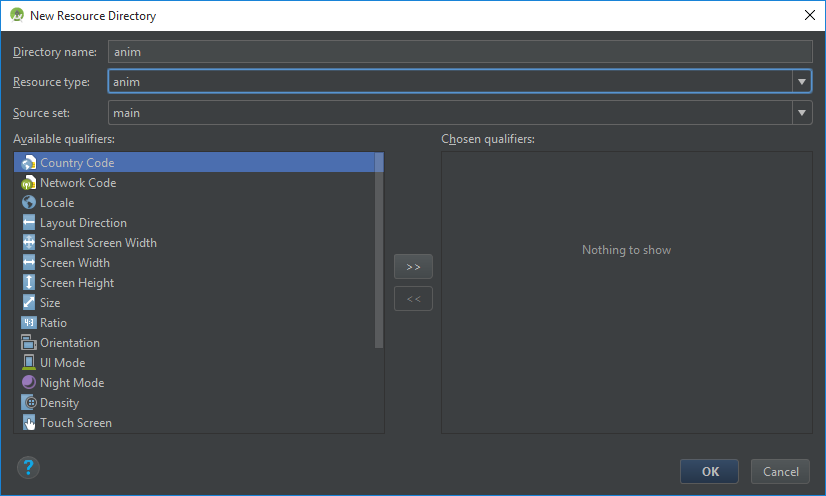
### Animação

Já finalizamos a primeira parte do nosso projeto. Adicionamos o nosso logo e o nosso plano de fundo na tela principal. Porém, queremos que abra o nosso logo e logo em seguida, vá para uma outra tela de login, por exemplo.

Para isto, iremos criar uma animação na nossa tela principal e em seguida iremos redicioná-la para outra tela.

Dentro da pasta res, iremos criar uma pasta do tipo “*Android resource directory*”.

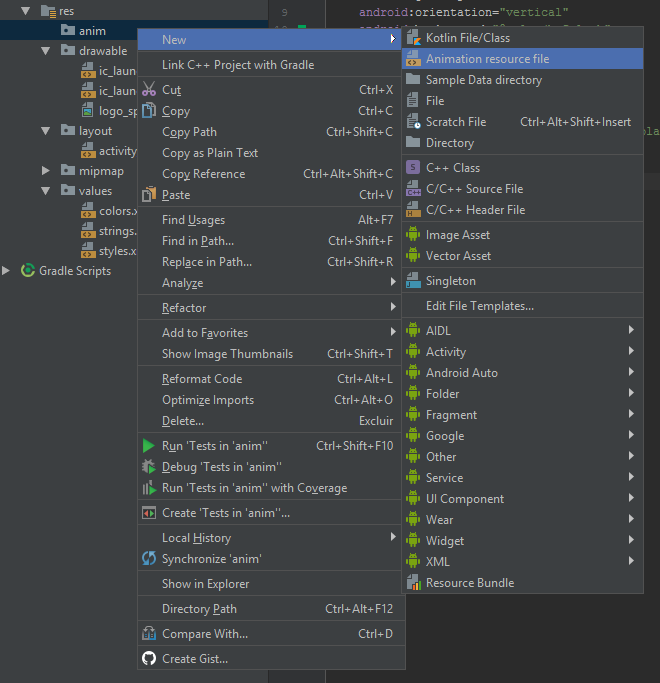




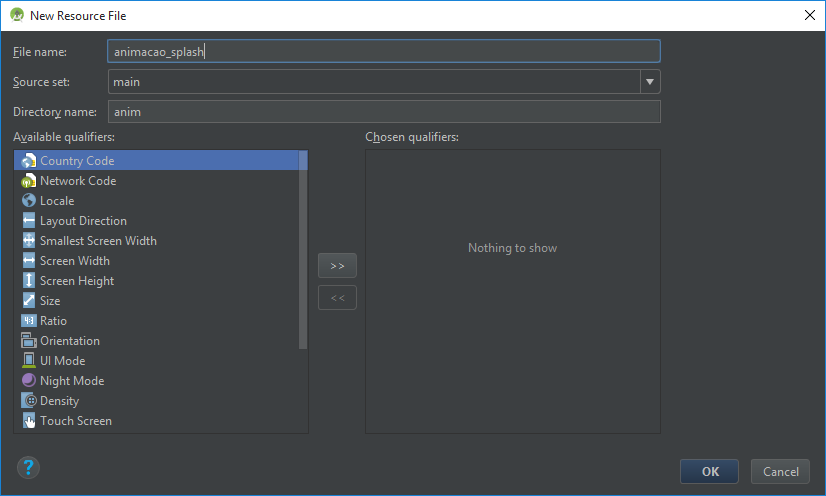
Escolheremos o nome do diretório de “*anim*” e o tipo de recurso que iremos utilizar também será *anim*.

Clicar em “*Ok*” para finalizar a criação da nossa pasta.

Iremos criar agora um xml que conterá as informações que desejamos incluir em nossa animação.

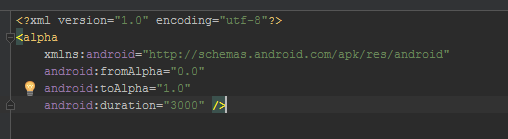


Iremos colocar o nome de “*animacao\_splash*”.



Clicar em “*Ok*” para finalizar a criação do nosso arquivo.

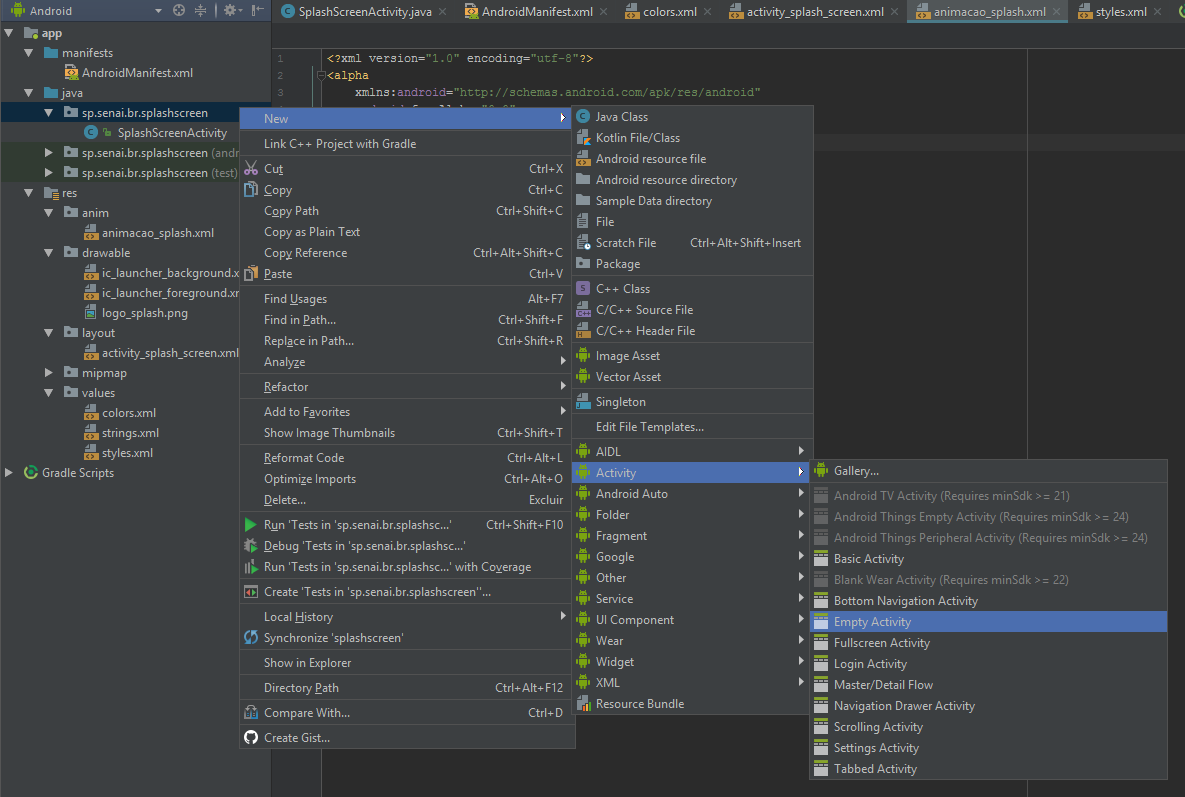
Iremos editar o arquivo que acabamos de criar (animacao\_splash.xml) da seguinte maneira:



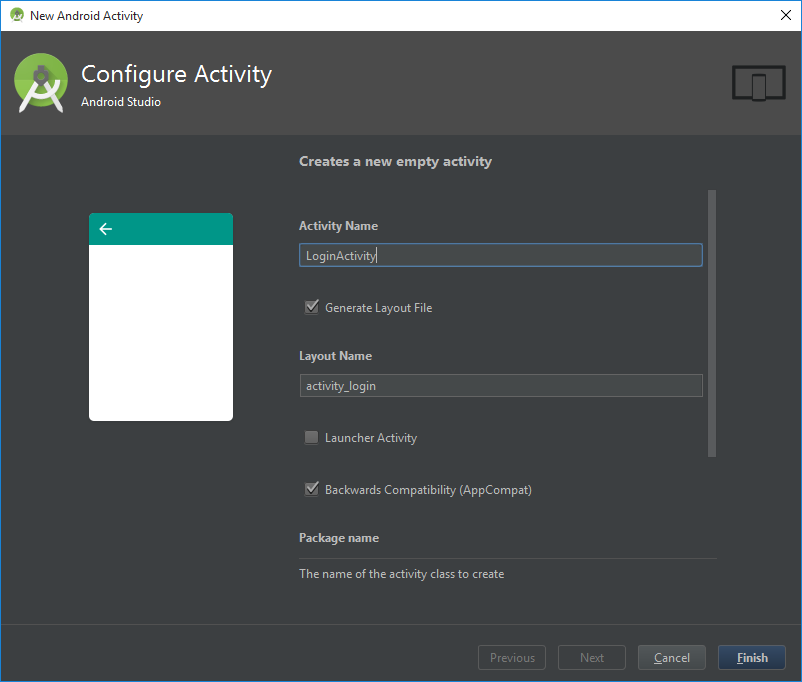
Com essas informações, iremos realizar um *fade-in* na nossa imagem.

### Tela Secundária

Iremos agora adicionar uma segunda tela para o nosso aplicativo.

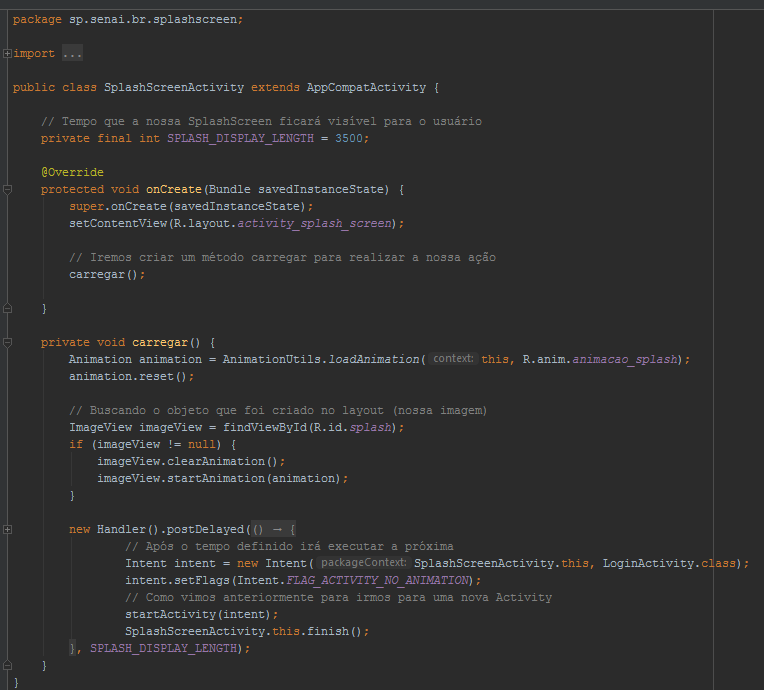


Iremos colocar o nome de *LoginActivity*.



Clicar em “*Finish*” para finalizar a criação da nossa segunda activity.

Criamos a nossa segunda tela, precisamos alterar agora o comportamento da nossa *SplashScreenActivity* para que, ao terminar de carregar o logo e tempo de carregamento do aplicativo, ele seja redirecionado para a tela de login ou qualquer outra tela correspondente.



# Resumo

Neste tutorial vimos como criar uma splashscreen e a importância para a nossa aplicação. Além de somente termos uma tela inicial, nós podemos carregar algo em segundo plano enquanto carregamos a tela inicial para o usuário.

# Referências

<http://blog.alura.com.br/criando-uma-tela-de-abertura-no-android-splash-screen/>

<https://android.jlelse.eu/right-way-to-create-splash-screen-on-android-e7f1709ba154?gi=eecd127ab15f>