Sumário

[Criando um novo projeto 2](#_Toc515971862)

[Organizando a estrutura do nosso projeto 5](#_Toc515971863)

[Criando a tela de login 6](#_Toc515971864)

[Alterando a activity de início do aplicativo 8](#_Toc515971865)

[Adicionando as bibliotecas 10](#_Toc515971866)

[Realizando a configuração do Retrofit 11](#_Toc515971867)

[Criando a classe de configuração do Retrofit 11](#_Toc515971868)

[Criando a classe de utilitários 12](#_Toc515971869)

[Criando a interface para o Retrofit 12](#_Toc515971870)

[Criando a classe de modelo do Usuário 13](#_Toc515971871)

[Resumo 15](#_Toc515971872)

[Realizando login 16](#_Toc515971873)

[Efetuando o login 16](#_Toc515971874)

[Atualizando o AndroidManifest 17](#_Toc515971875)

[Armazenando o token 18](#_Toc515971876)

[Imprimindo a lista de categorias 20](#_Toc515971877)

[Criando o modelo categoria 20](#_Toc515971878)

[Adicionando um novo método na interface 20](#_Toc515971879)

[Incluindo o Authorization no Header 20](#_Toc515971880)

[Imprimindo as categorias com Log.d 21](#_Toc515971881)

[Imprimindo as categorias com RecyclerView 21](#_Toc515971882)

[Adicionando as dependências 21](#_Toc515971883)

[Adicionando o RecyclerView ao layout 21](#_Toc515971884)

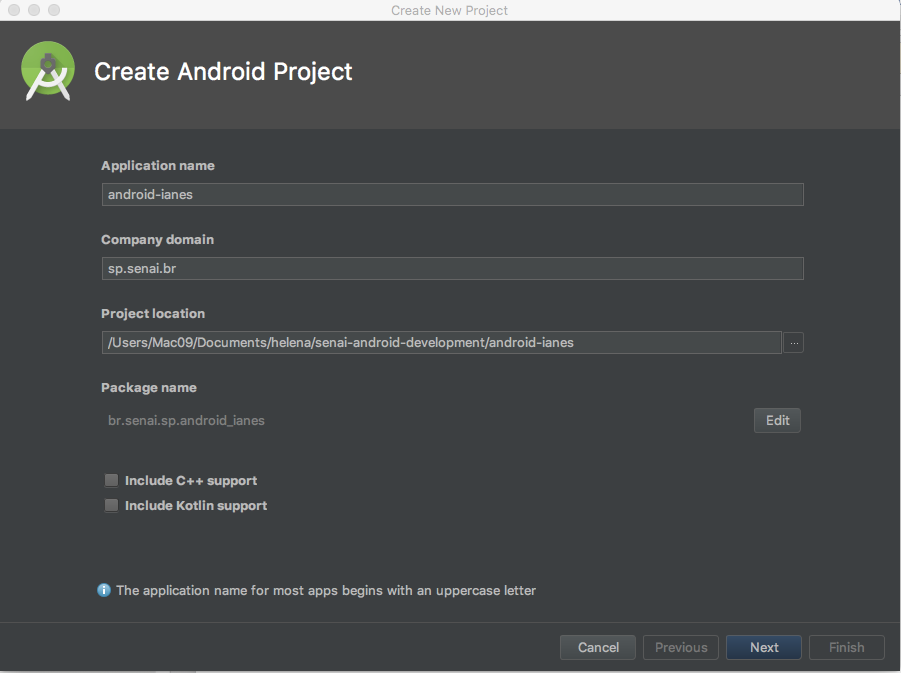
[Criando a linha personalizada 22](#_Toc515971885)

[CategoriaViewHolder 23](#_Toc515971886)

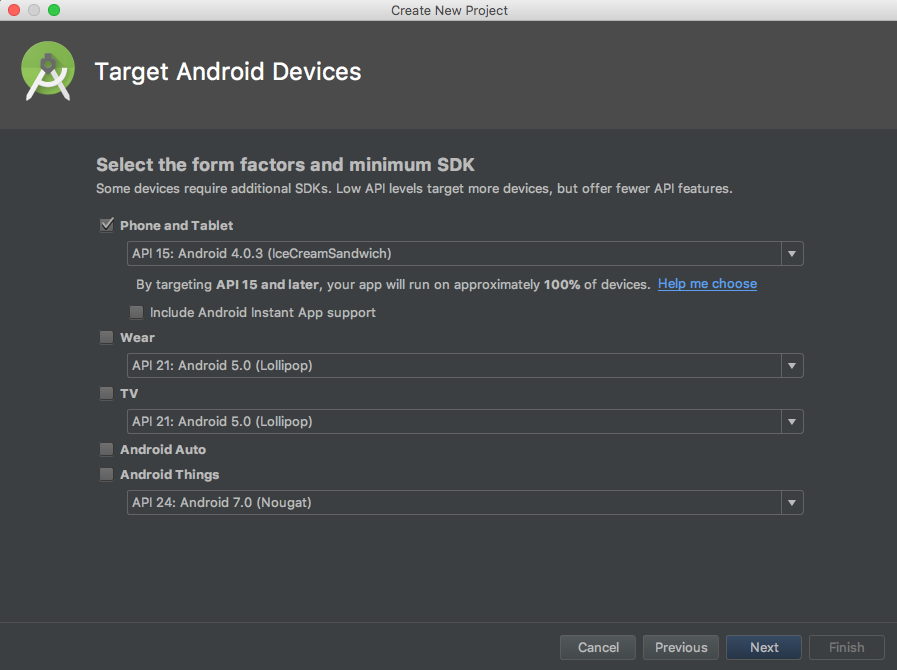
[CategoriaAdapter 23](#_Toc515971887)

[CategoriaActivity.java 24](#_Toc515971888)

# Criando um novo projeto

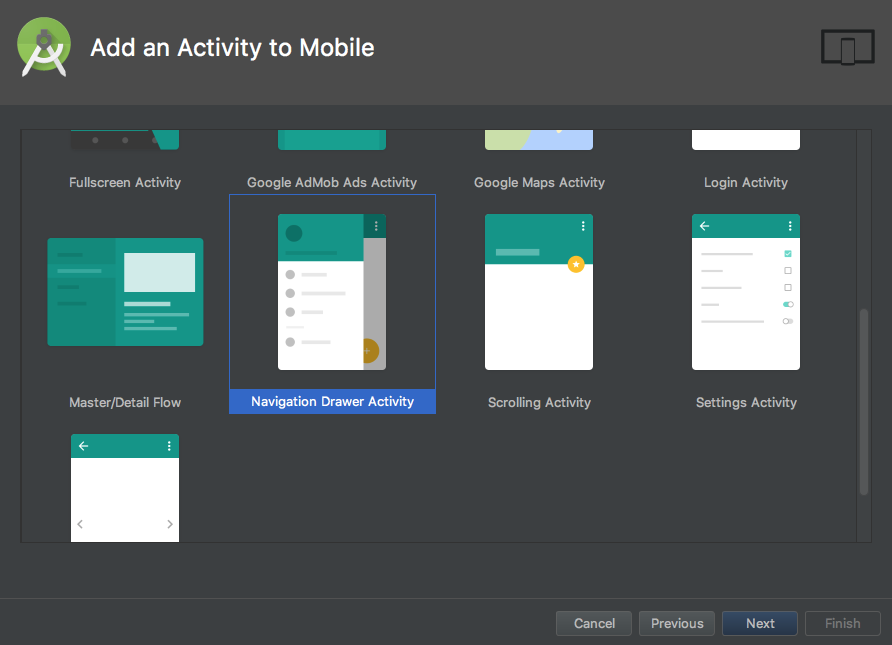


Clicar em ‘*Next’*.



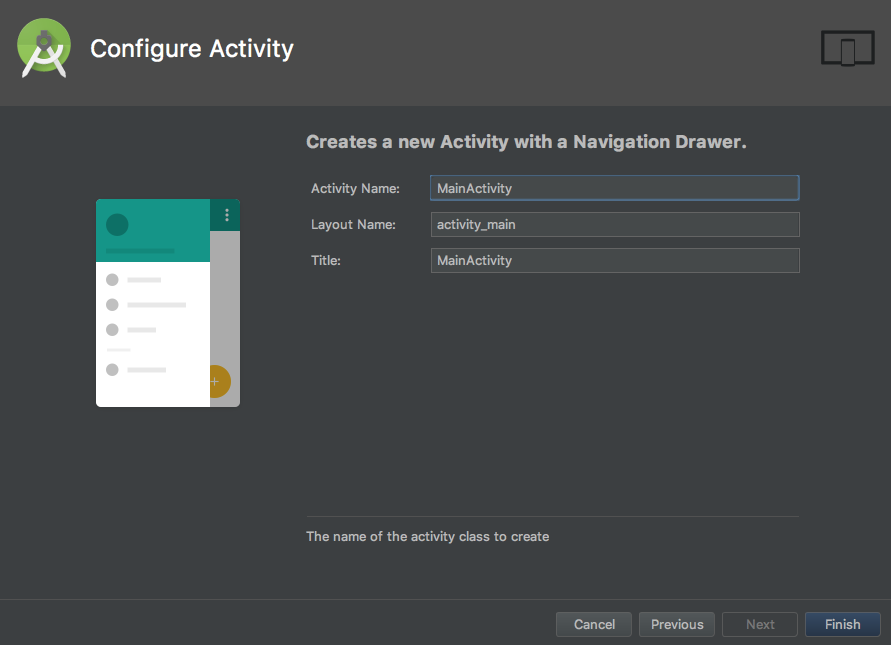
Clicar em ‘*Next’*.

Escolher o modelo da primeira activity. No meu exemplo, vou selecionar o tipo ‘Navigation Drawer Activity’.



Clicar em ‘Next’.

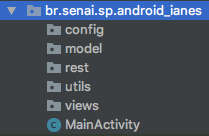
E manterei o nome como ‘*MainActivity’*.



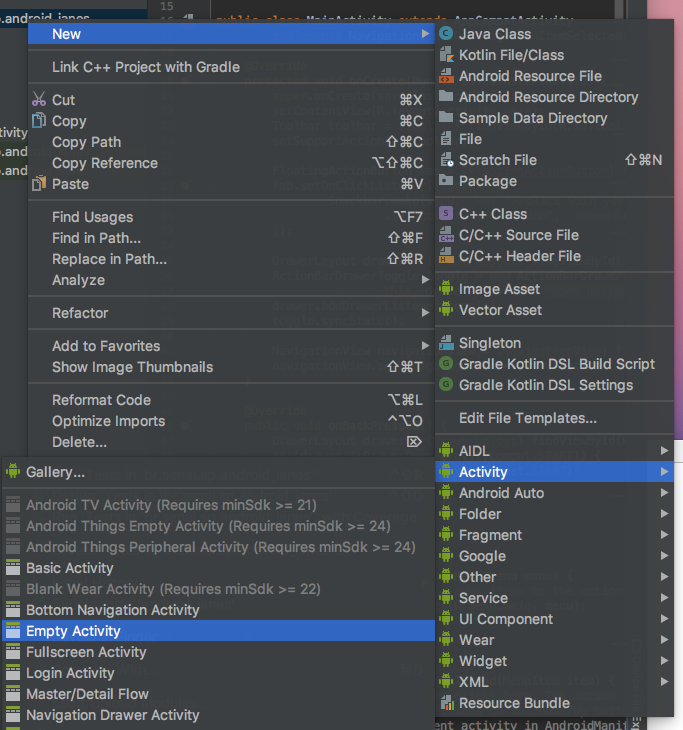
Clicar em ‘*Finish*’.

## Organizando a estrutura do nosso projeto

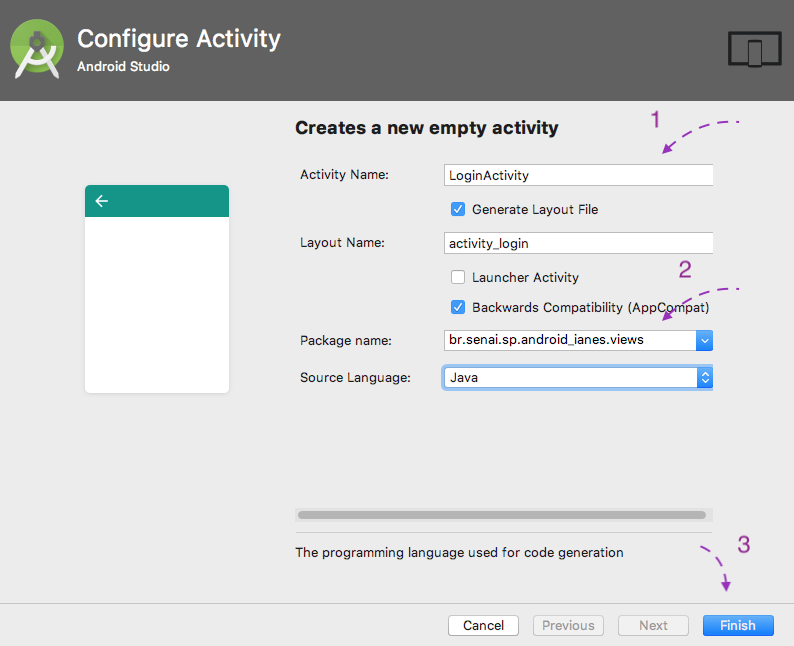
Vamos começar com a organização do nosso projeto. Vamos criar os pacotes:



## Criando a tela de login

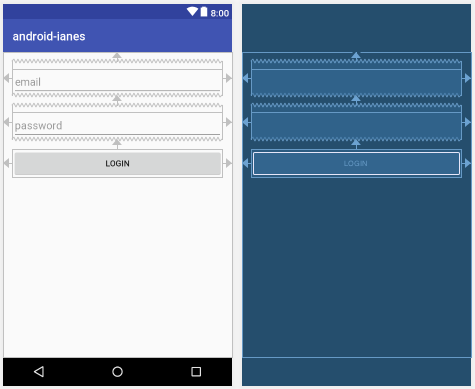


Essa nova activity será a de Login.



Clicar em ‘*Finish*’.

Após criarmos a activity de login, vamos colocar os campos correspondentes para o usuário inserir com os dados de acesso de sua conta.



2 TextInputLayout (para e-mail e senha) e 1 botão para realizar a ação de clicar.

## Alterando a activity de início do aplicativo

Após criarmos a activity de login, ela deverá a de início do nosso aplicativo. Sendo assim, precisamos alterar no AndroidManifest.xml essa configuração.



Por enquanto, vamos apenas deixar a activity de login criada, e começar a criar as nossas classes de configuração.

## Adicionando as bibliotecas

Mas antes de criarmos as classes, já conhecemos algumas bibliotecas que queremos utilizar. Vamos adicioná-las.

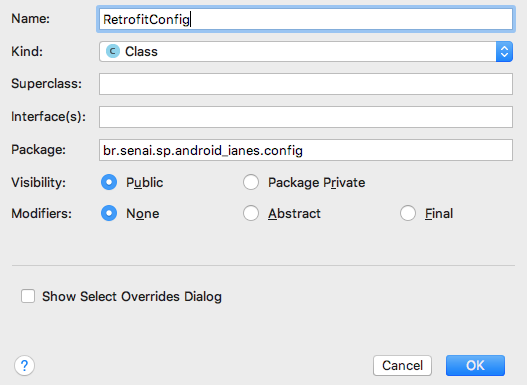
Observação: na versão, o termo *compile* para as bibliotecas, ficou obsoleto. Utilize o termo *implementation.*



# Realizando a configuração do Retrofit

## Criando a classe de configuração do Retrofit

Criar uma nova classe chamada RetrofitConfig dentro do pacote config.



Clique em ‘*Ok*’.

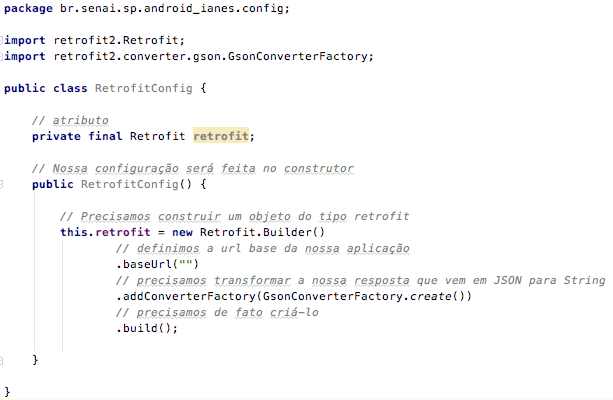


Figura 1 - Classe de configuração do Retrofit

Para não inserirmos manualmente a nossa URL base do projeto, vamos criar uma nova classe que nos ajudará a não repetir informações em nosso aplicativo e deixá-las desorganizadas.

## Criando a classe de utilitários

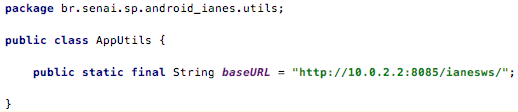


Figura 2 - Criando a classe AppUtils dentro do pacote utils

Voltando a classe RetrofitConfig, podemos alterar a baseURL para a que foi definida em AppUtils. Assim, sempre que precisarmos alterar a url, podemos manter nosso projeto ordenado e organizado.

../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2012.21.00.

Já configuramos a classe que ficará responsável pelas requisições. Vamos criar a nossa interface. Ela conterá os endpoints que queremos acessar da API.

## Criando a interface para o Retrofit

../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2012.34.13.

Figura 3 - Criando a interface que conterá os endpoints

Dentro do pacote rest, criar uma interface chamada RestInterface.

O primeiro endpoint importante que devemos acessar, será o que ficará responsável por gerar o nosso token. Esse passo é muito importante.

../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2012.36.18.

Ainda não está completo. Vamos lembrar. Uma vez que o post será feito, o que eu preciso informar no corpo para que os meus dados sejam validados?

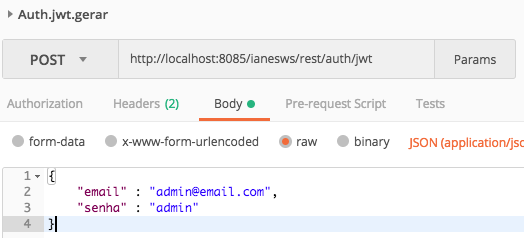


Figura 4 - Gerando o token no postman

Sendo assim, eu preciso informar e enviar os dados do usuário. Nós ainda não criamos nenhum modelo, vamos criar o nosso primeiro modelo.

## Criando a classe de modelo do Usuário

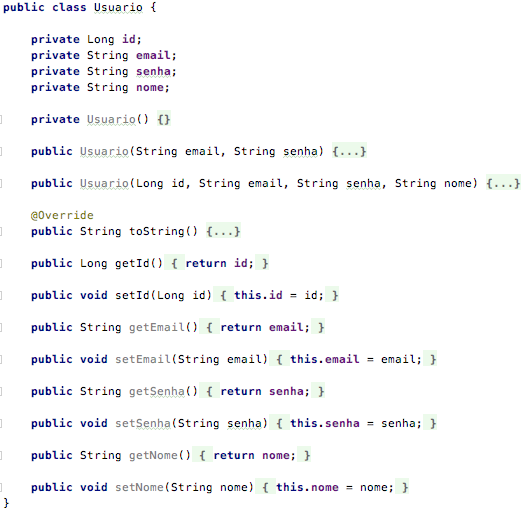
Não esquecer das informações de dentro do construtor

private String email;

public Usuario(String email) {

this.email = email;

}

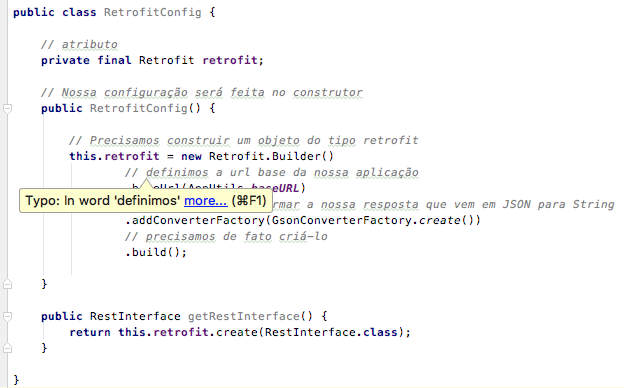


Depois de criar o nosso modelo, vamos alterar a nossa interface para enviar os dados do usuário.

INTERFACE

../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2012.42.17.

Como criamos essa interface, vamos informá-la em nossa classe do RetrofitConfig.



Observação: Para qualquer requisição que eu venha a fazer dentro da nossa aplicação (qualquer endpoint que esteja restrito), eu precisarei enviar o token para que ele valide e saiba as minhas credenciais. Além disso, através do token, eu consigo pegar o nome do usuário, e outro detalhe muito importante, o tipo de permissão que ele contém, caso eu venha a desenvolver duas telas distintas (permissão de acesso), eu consigo redirecioná-lo para a tela correta.

## Resumo

Vamos recapitular o que foi feito até agora.

1. Criamos a classe de configuração do Retrofit, para realizar as nossas chamadas;
2. Criamos a interface que conterá as nossas “referências” aos endpoints que serão chamados.
3. Criamos o nosso modelo Usuario que conterá as informações do token;
4. Criamos a tela de login.

Vamos começar a segunda etapa do projeto. Eu preciso pegar os dados que o usuário digitar na nossa aplicação, fazer a requisição para o endpoint que foi informado (rest/auth/jwt) e na resposta, eu preciso armazenar a informação no retorno. No caso, o token.

# Realizando login

Em um primeiro momento, nós já havíamos construído a tela de login. Faça as devidas referências para conseguirmos capturar as informações que o usuário digitar.



## Efetuando o login

Precisamos enviar as informações do usuário e receber o token, caso as informações enviadas, coincidam com os registros no banco de dados.

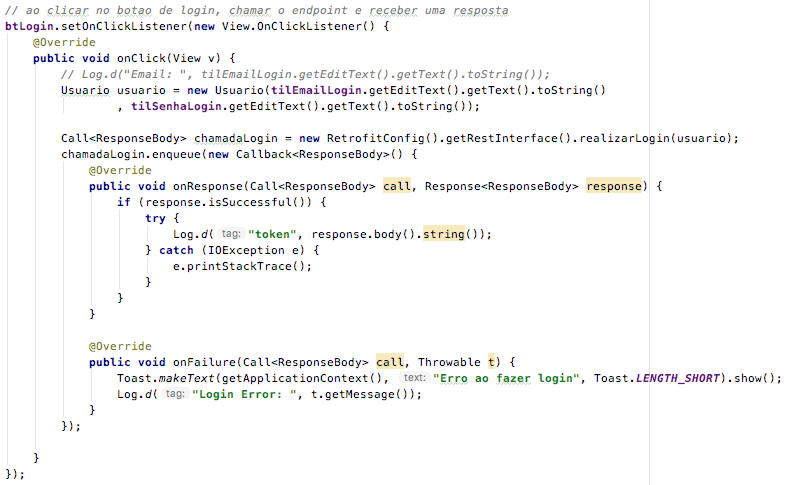


Figura 5 - Realizando a chamada.

Após realizar a chamada, você verá um erro no log.

../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2014.03.04.

Figura 6 - Erro

O que ficou faltando realizar na nossa configuração para que funcionasse?

## Atualizando o AndroidManifest

AndroidManifest.xml

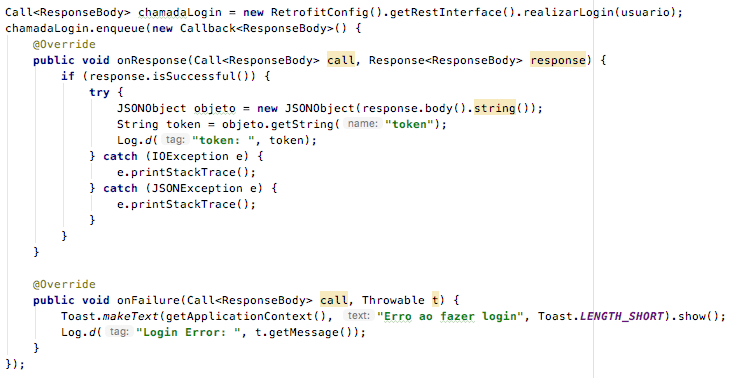
../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2014.07.57.

Figura 7 - Permissão de acesso

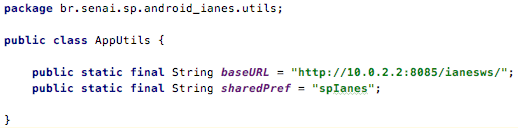
# Armazenando o token

LoginActivity

JSON

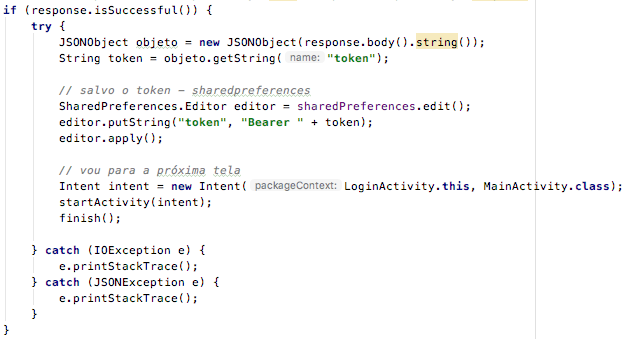


SharedPreferences



LoginActivity

Lembre-se de declarar o SharedPreferences anteriormente.



MainActivity

Dentro do onCreate().

../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2014.20.53.

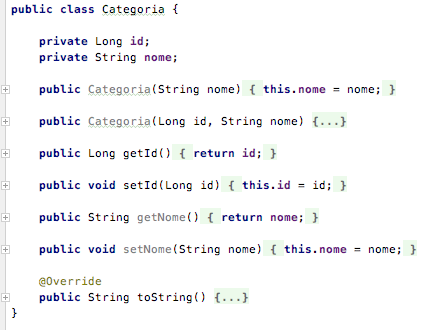
Imprimindo o token

../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2014.34.23.

# Imprimindo a lista de categorias

## Criando o modelo categoria

Categoria.java



## Adicionando um novo método na interface

RestInterface

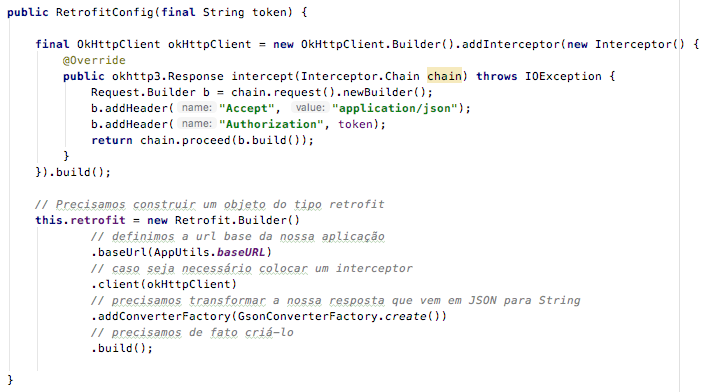
../../Desktop/Captura%20de%20Tela%202018-05-30%20às%2015.54.39.

## Incluindo o Authorization no Header

RetrofitConfig

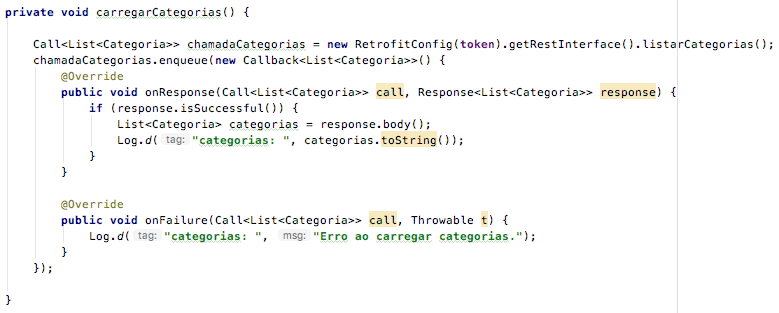
Obs.: ele terá dois construtores. Um que recebe o token como parâmetro e outro que não recebe (que foi o que criamos anteriormente).

S2



## Imprimindo as categorias com Log.d

MainActivity

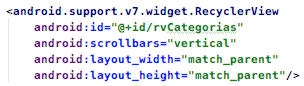


# Imprimindo as categorias com RecyclerView

## Adicionando as dependências

*// recyclerview*implementation **'com.android.support:recyclerview-v7:27.1.1'**

## Adicionando o RecyclerView ao layout

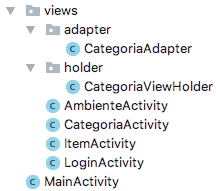


## Criando a linha personalizada

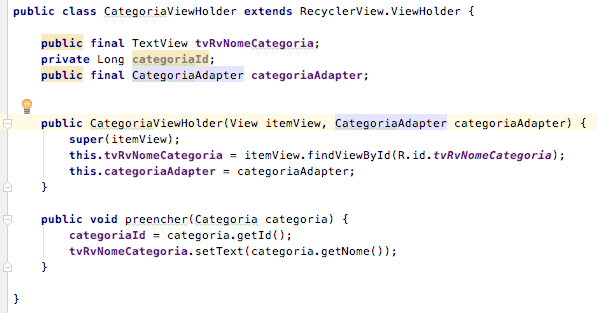


Criando as pastas para organizar o RecyclerView

../../Desktop/Captura%20de%20Tela%202018-06-05%20às%2007.19.03.



## CategoriaViewHolder



## CategoriaAdapter



## CategoriaActivity.java

