

# Tutorial de Configurações De IDE com Pydroid-Project

Resumo	2
Mas como isso funciona?	2
Sem mais delongas, enfim o tutorial:	2
Eclipse-luna	2
1.Passo - Instalação Do Eclipse-Luna	2
2.Passo – Instalação do plugin ADT	5
3.Passo – Configurações do SDK no Eclipse Luna	9
4.Passo – Carregamentos de Projetos Android	11
5.Passo – Configurações na IDE e instalação do Android 19.	15
Outras IDES	19
Codificação no sublime	20
Codificação no Notepad ++	20
Codificação JetBrains IntelliJ IDEA	21
InstallDebug.py	22

## Resumo

O intuito do Pydroid é de certa forma, permitir que as pessoas programem para o Android em outras IDEs além do Android Studio, fazendo com que máquinas mais fracas ou máquinas abaixo da média, consigam programar para o Android, o programa visa dar uma opção alternativa de programação do Android.

### Mas como isso funciona?

Em suma o Pydroid cria os pacotes, pastas e configurações similares ao Android Studio, enquanto a codificação fica a cargo de uma IDE já instalada como o eclipse, IntelliJ IDE, Notepad e vários outros editores de código.

Lembrando que existem algumas alterações na criação desses arquivos, uma delas, por exemplo, é o Gradle. O Pydroid cria apenas um, enquanto o Android Studio cria dois arquivos Gradle, porém não é nada que afete a execução do Android ou afete significativamente algo, não se preocupe, e como se o App/Gradle estivesse dentro do Gradle! Na verdade, é isso!


### Sem mais delongas, enfim o tutorial:

**ALIÁS, SE PREPRA PORQUE ELE É GRANDINHO 😊 (não me venha com essa de mucho texto beleza)**

## Eclipse-luna















### 1.Passo - Instalação Do Eclipse-Luna

Assim que você baixar o Eclipse Luna ele vira zipado:

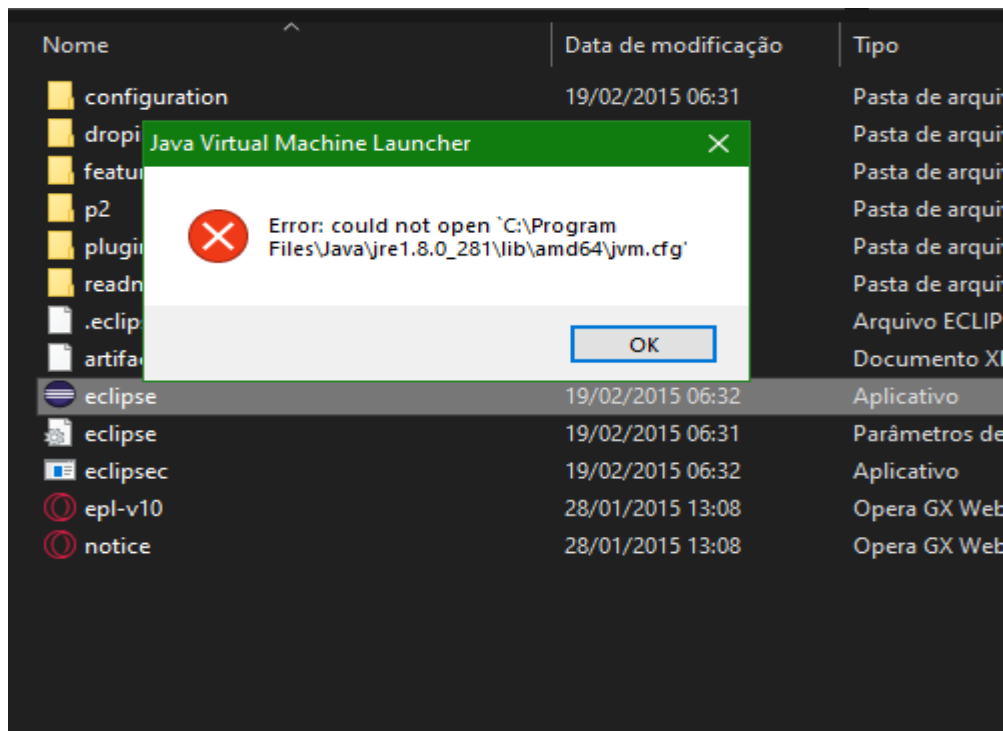
 eclipse-jee-luna-SR2-win32-x86_64	24/03/2021 15:38	WinRAR ZIP archive	260.529 KB
---	------------------	--------------------	------------

Se atente a estrutura do seu PC e ao SO para não baixar uma versão incompatível.

Extraia a pasta do eclipse para um local de sua preferência e em seguida abra a pasta

Nome	Data de modificação	Tipo
 configuration	25/03/2021 15:41	Pasta de arquivo
 dropins	19/02/2015 06:31	Pasta de arquivo
 features	24/03/2021 22:38	Pasta de arquivo
 jre	24/03/2021 22:22	Pasta de arquivo
 p2	24/03/2021 22:28	Pasta de arquivo
 plugins	24/03/2021 22:38	Pasta de arquivo
 readme	19/02/2015 06:31	Pasta de arquivo
 .eclipseproduct	28/01/2015 13:08	Arquivo ECLIPSE
 artifacts	24/03/2021 22:39	Documento XM
 eclipse	19/02/2015 06:32	Aplicativo
 eclipse	24/03/2021 22:45	Parâmetros de c
 eclipsec	19/02/2015 06:32	Aplicativo
 epl-v10	28/01/2015 13:08	Opera GX Web I
 notice	19/01/2015 18:59	Opera GX Web I

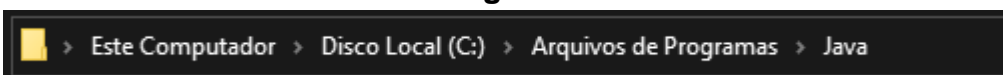
Agora de dois cliques na aplicação do Eclipse e...



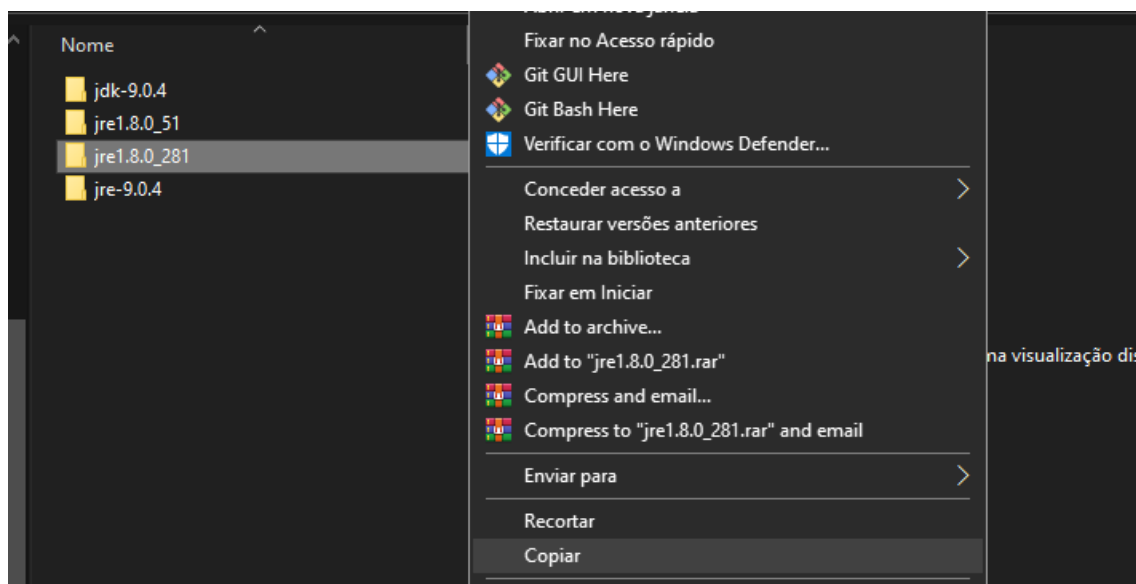
Quem me dera fosse tão fácil, o Eclipse vai exibir um erro no JRE, e não se preocupe e bem fácil arrumar isso, aparentemente nesta IDE do eclipse você precisava colocar manualmente o JRE para que o programa execute.

Bom como já citado, você vai precisar ter instalado pelo menos um JRE versão 8 acima do update 51, pois foram testadas 4 JRE'S, e apenas as da versão 8 funcionaram, as de versões superiores crasharam o Luna, como a 9.0.4 que também foi testada.

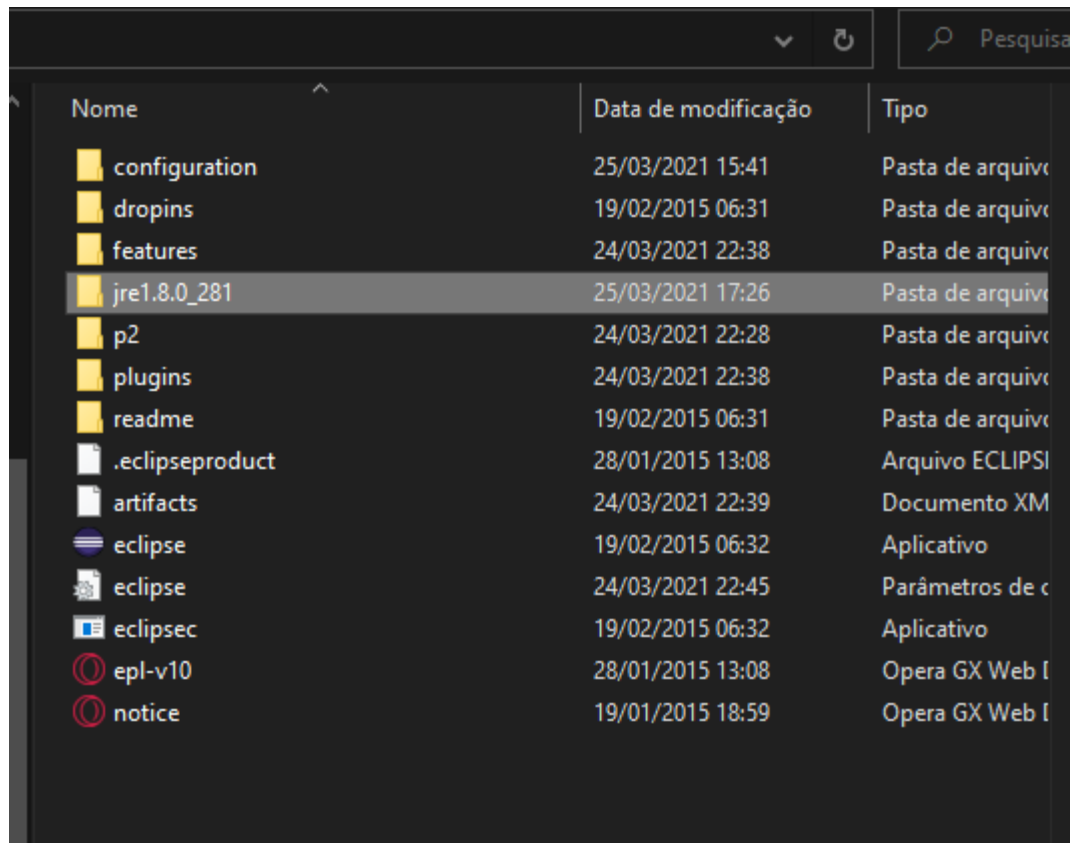
Primeiramente você precisa ir até onde os JRE estão localizadas no seu PC e por padrão o launcher do Java instala elas no caminho **C:\Program Files\Java**



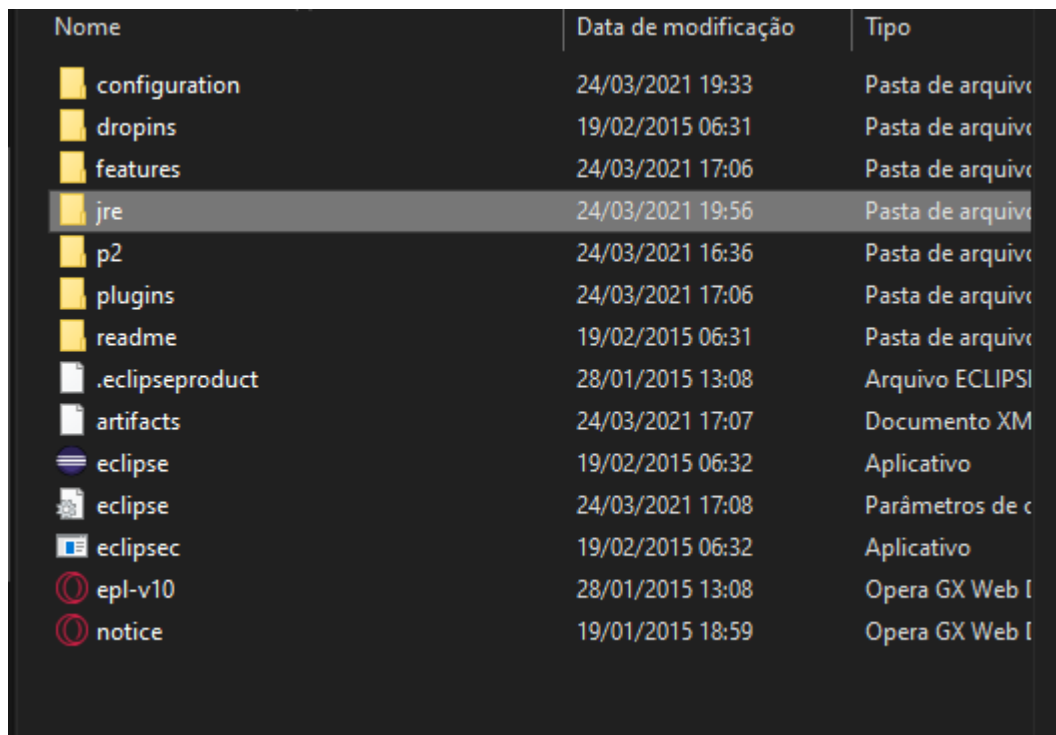
La estarão todas os JRE e os JDK que você possui, você vai copiar a pasta de um dos JRE da versão 8, clique com o botão direito sobre a pasta e selecione copiar.



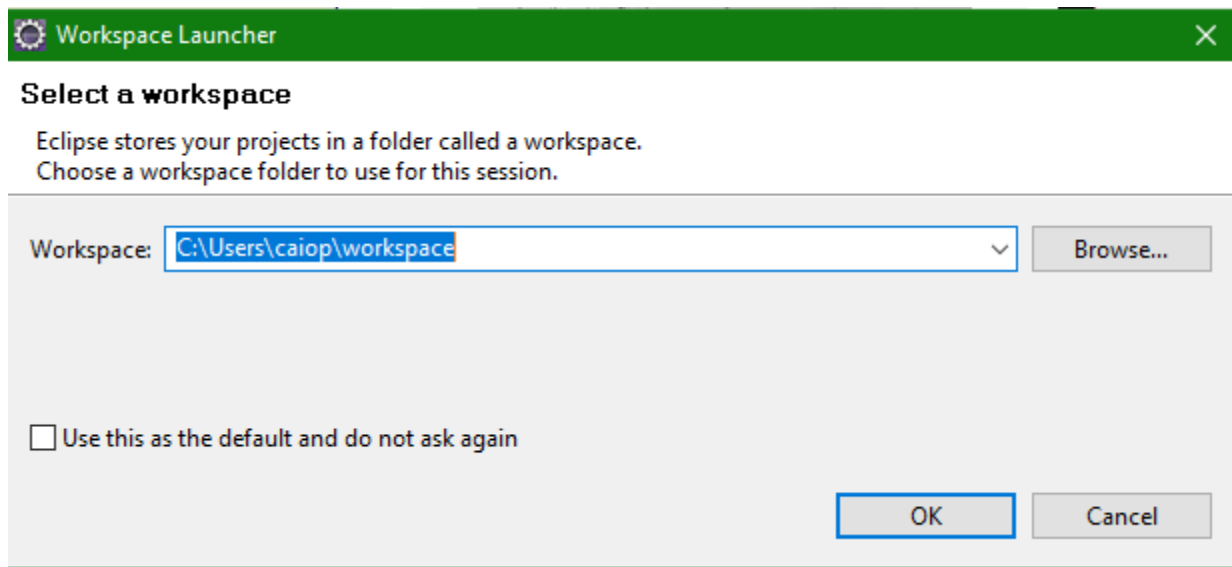
Agora você vai ir até a pasta do Eclipse Luna e cole o JRE copiado na pasta



Após isso você vai renomear a pasta que você colou apenas para **jre**



Após isso, Double click na aplicação do eclipse e voilà, ele vai dar início, pode ser que apareça uma tela durante esse processo sobre o seu **workspace** padrão do eclipse, marque a caixinha e clique em **ok**, ou caso queira mudar onde os seus projetos ficarão salvos, sinta-se a vontade para coloca-los onde achar melhor

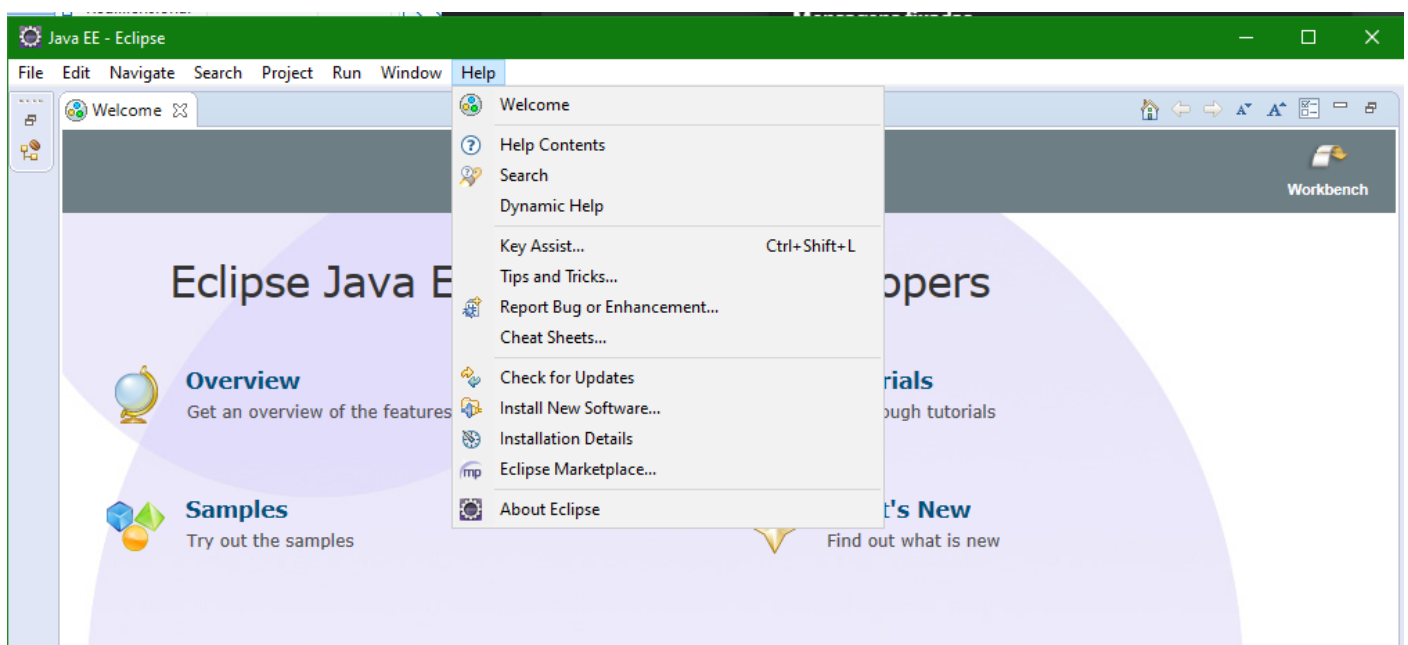


Agora com a tela inicial do eclipse aberta, podemos ir para o próximo passo

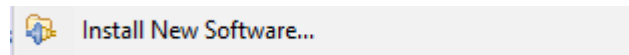
## 2.Passo – Instalação do plugin ADT

Bom o plugin ADT se faz necessário no Eclipse Luna porque é ele quem dá o suporte das API's do Android no Luna.

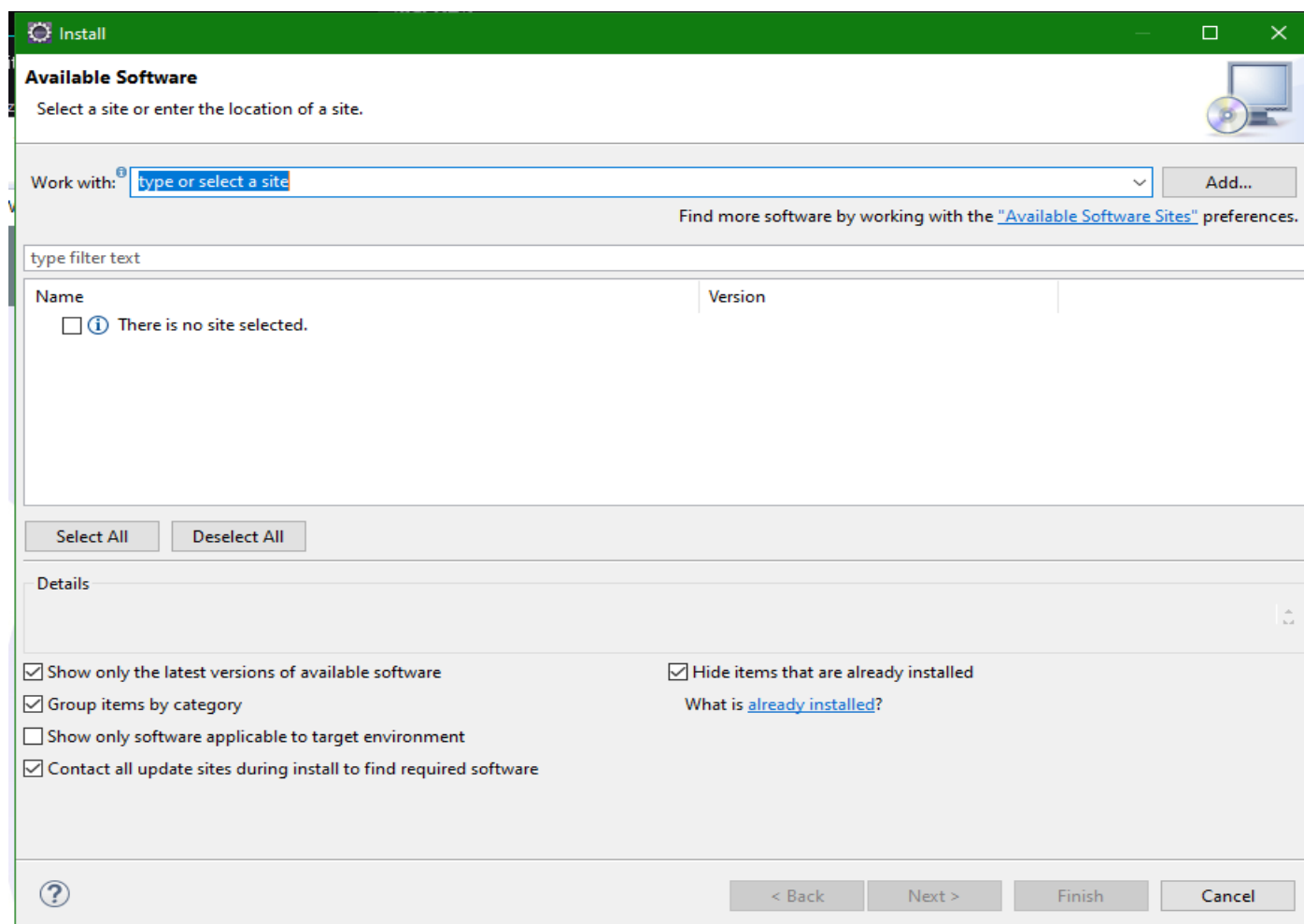
Primeiramente vá até **help**.



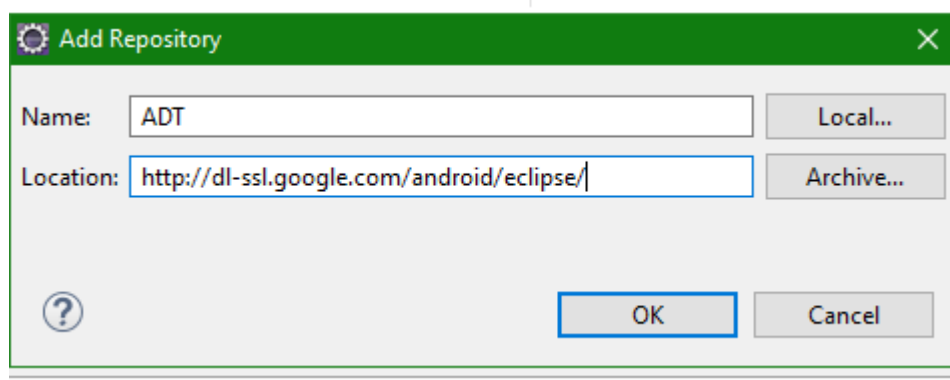
Agora vá em **Install New Software**



Assim será exibida essa tela




Clique em add, será exibida uma tela com dois campos, um para o nome e o outro para o link do download do plugin, no nome pode ser colocado “qualquer coisa”, mas para que não haja erros coloque apenas **ADT**, e na localização **<https://dl-ssl.google.com/android/eclipse/>**




Feito isso, clique em **ok** e será exibida essa tela

type filter text


Name	Version
> <input type="checkbox"/>  Developer Tools	

Marque a opção **Developer Tools** e clique em next

type filter text







Name	Version
> <input checked="" type="checkbox"/>  Developer Tools	

Então será exibida essa tela, com os recursos que serão adicionados, clique em **next**

 Install


**Install Details**

Review the items to be installed.

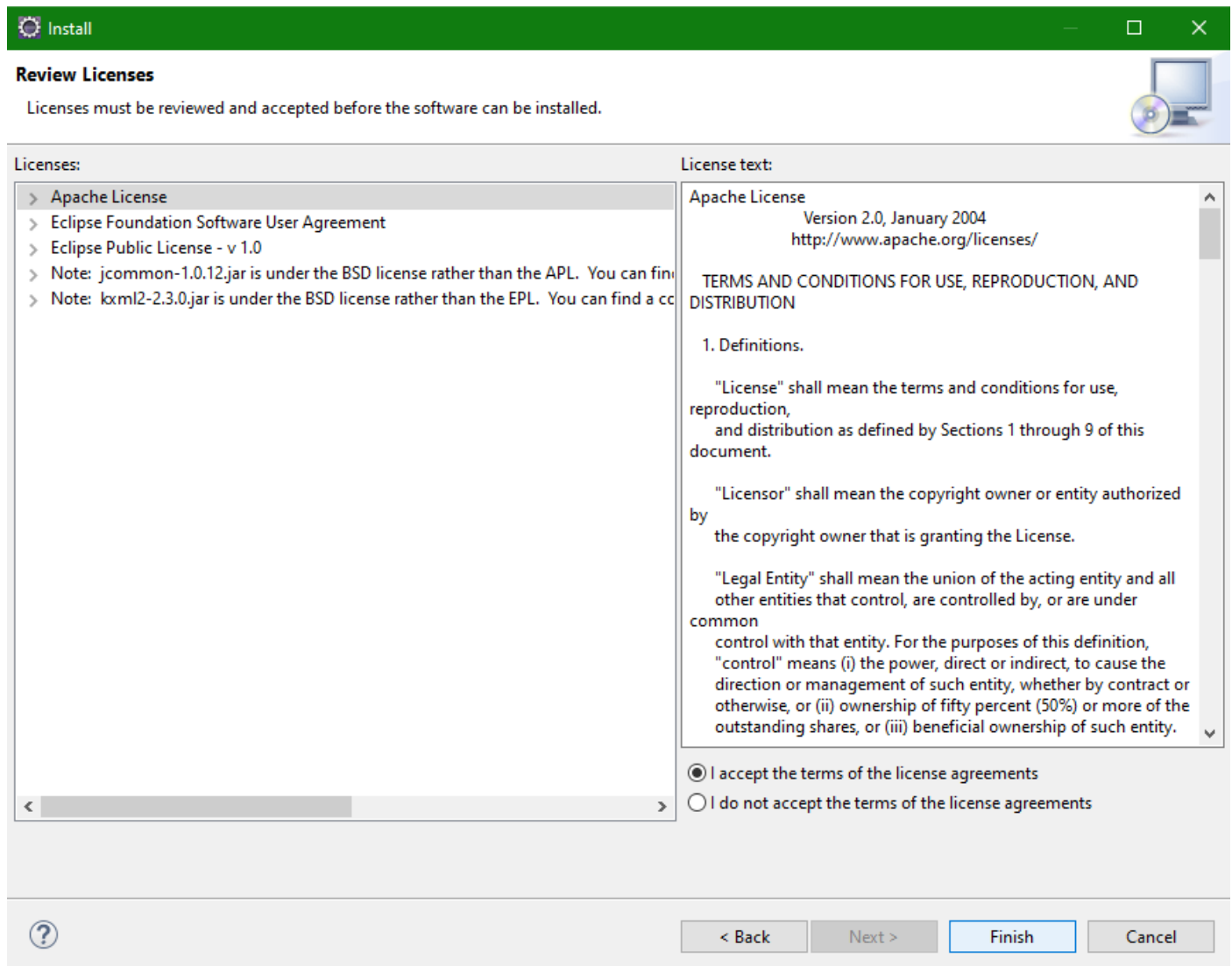
Name	Version	Id
 Android DDMS	23.0.7.2120684	com.android.ide.eclipse.ddms.featur...
 Android Development Tools	23.0.7.2120684	com.android.ide.eclipse.adt.featur...
 Android Hierarchy Viewer	23.0.7.2120684	com.android.ide.eclipse.hierarchy...
>  Android Native Development Tools	23.0.7.2120684	com.android.ide.eclipse.ndk.featur...
 Android Traceview	23.0.7.2120684	com.android.ide.eclipse.traceview....
 Tracer for OpenGL ES	23.0.7.2120684	com.android.ide.eclipse.gldebugg...

Size: Unknown

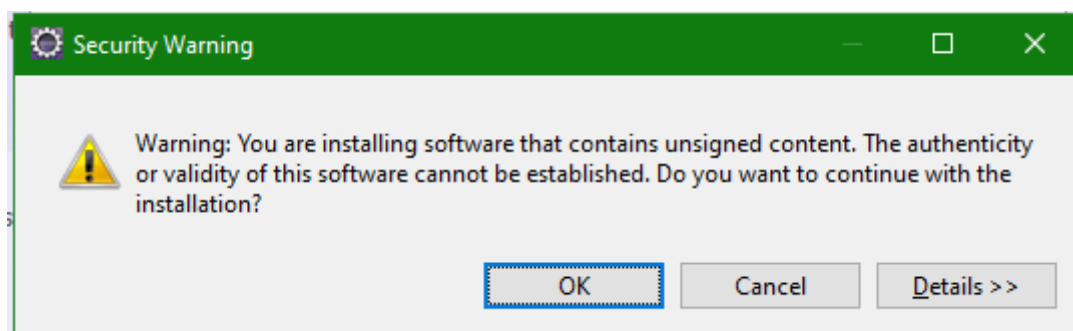
Details

 < Back Next > Finish Cancel

Será exibida essa tela, **marque a opção que concorda com os termos de uso** e por fim clique em **finish**.



Após isso o eclipse vai fazer o download do plugin, a demora vai depender da sua largura de banda, no meio do download, aparecerá essa tela, falando sobre downloads de terceiros e questões de segurança, apenas clique em ok.



Após isso aparecerá uma caixa com a opção de **restart**, selecione essa opção para que as alterações em relação ao plugin sejam feitas.

Agora que você baixou o plugin e reiniciou o Eclipse Luna, vamos ao próximo passo.



### 3.Passo – Configurações do SDK no Eclipse Luna

Esta parte vai parecer um pouco complexa, por isso é de suma importância que você tenha o SDK instalado em seu computador.

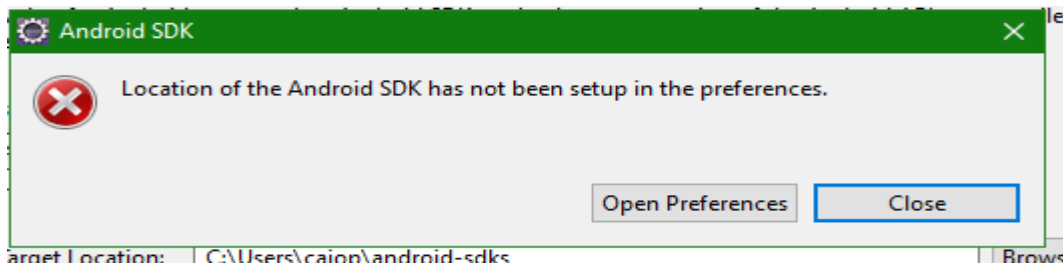
Antes de continuarmos peço que você verifique se você tem o SDK do Android, porém essa verificação depende de onde você instalou sua Sdk do Android.

EX.:

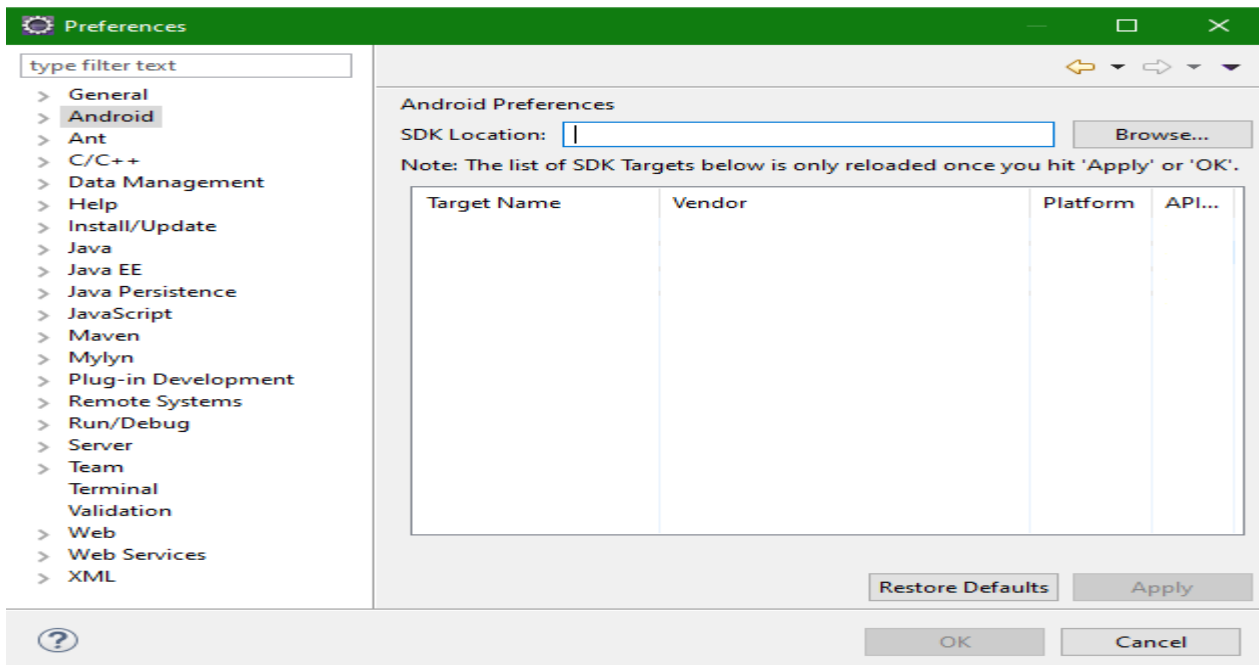
C:\Users\\*seu usuário\*\AppData\Local\Android\Sdk

C:\Users\\*seu usuario\*\Documents

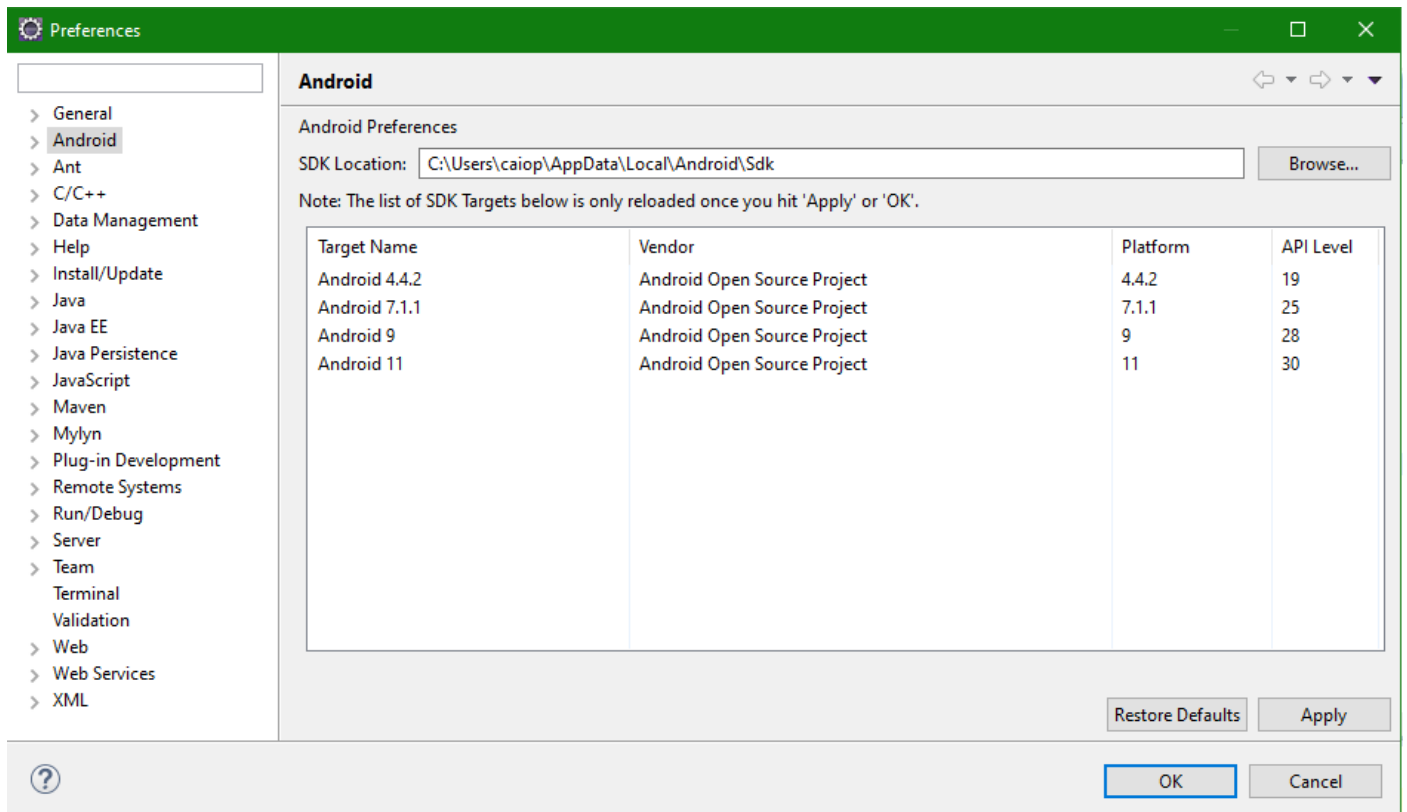
Depois de ter reiniciado o Luna apareceram duas telas uma em cima da outra, sendo essa.



Clique em **Open Preferences**, será exibido essa tela

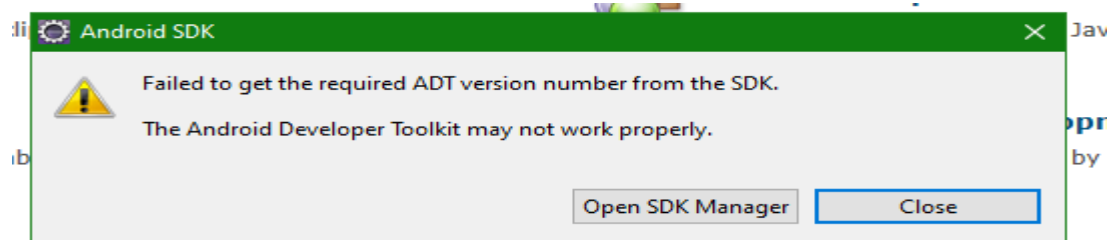


Nesta tela, você vai por o caminho do seu SDK do Android na barra **sdk location**, ou encontra-lo através do **browser**, e vai ficar assim.

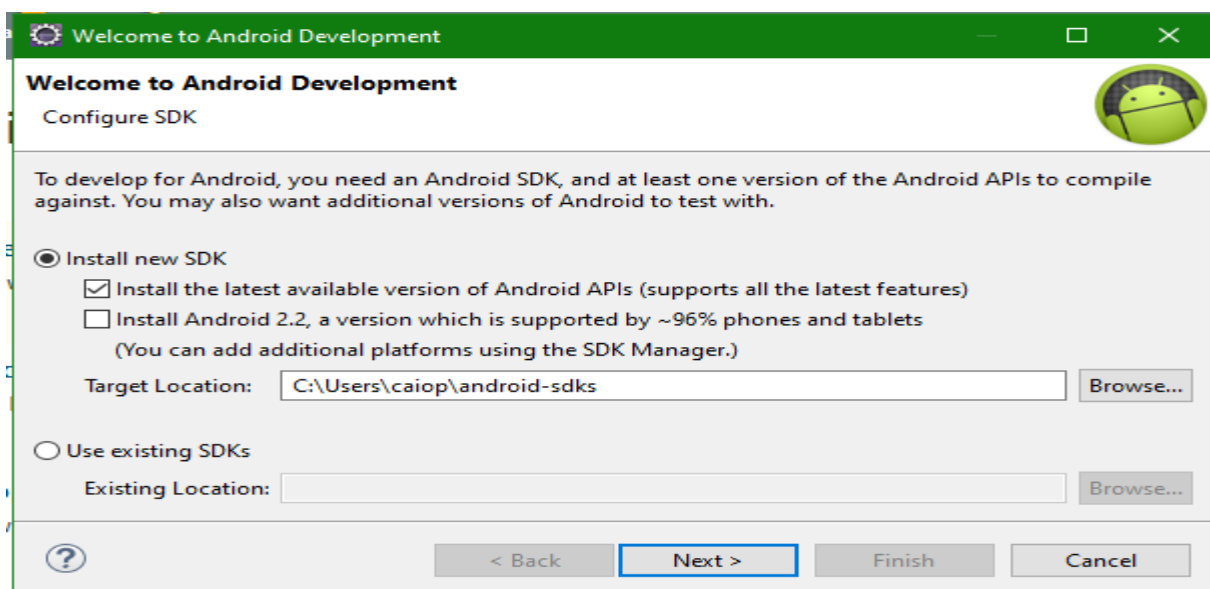


Feito isso você vai clicar em **Ok**

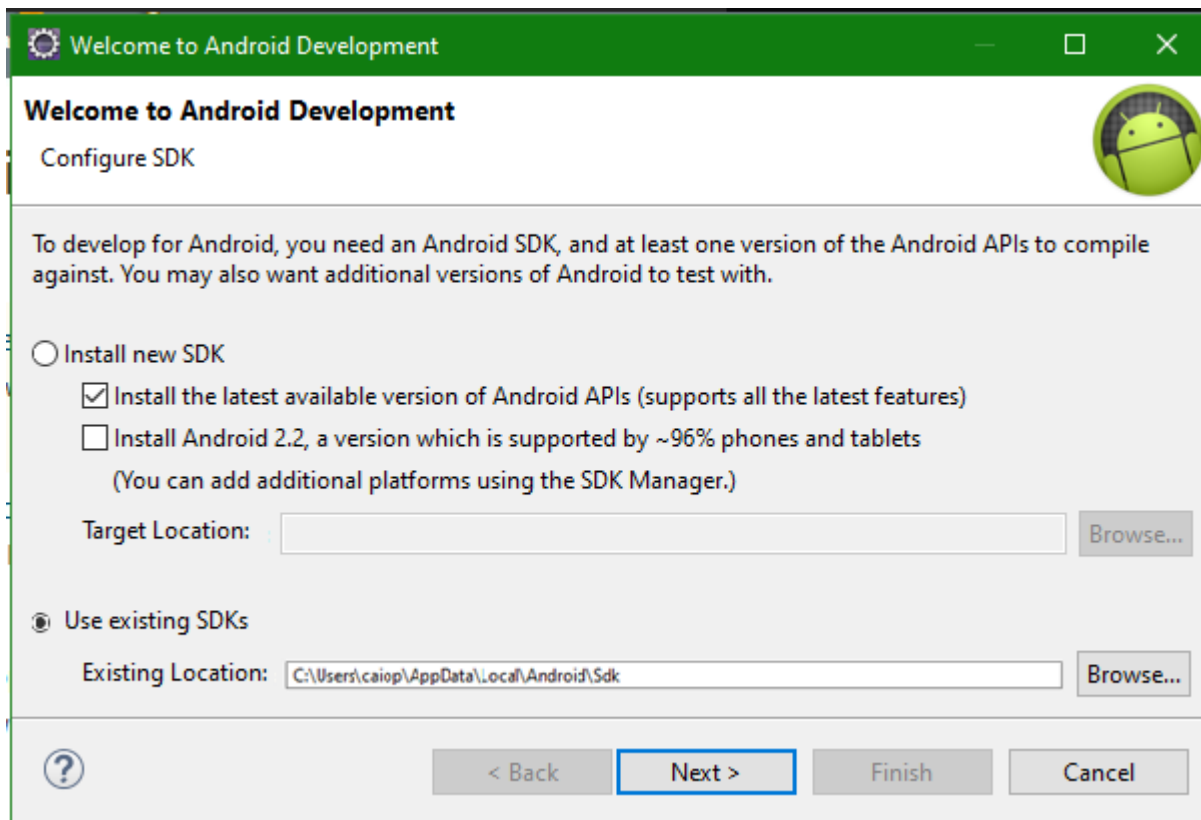
Em seguida essa tela vai aparecer clique em **Open SDK Manager**.



Você será direcionado para esta outra tela,



Assim que esta tela surgir, você vai marcar a opção **Use existing Sdks**, e colocar o caminho do seu SDK em **existing location**, ou dar browse e encontra-lo no Explorer.



Após isso clique em **next**, você será redirecionado para a tela inicial do Eclipse.

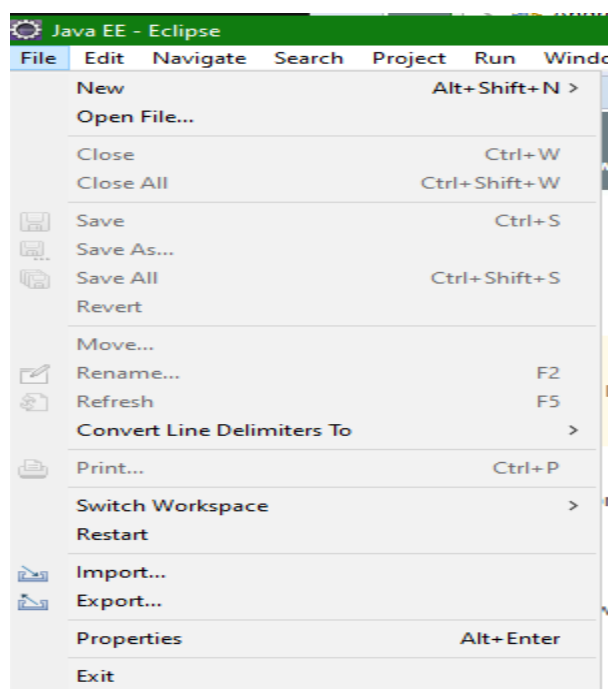
Com isso vamos para o próximo passo

**Obs.: talvez a ordem das caixas seja aleatória, então se atente aos botes Open Preferences e Open Sdk Manager**

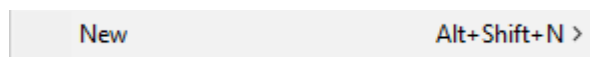
## 4.Passo – Carregamentos de Projetos Android

É de suma importância que você saiba onde está a pasta que o Pydroid criou para você, pois ela será usada neste passo para a sua codificação.

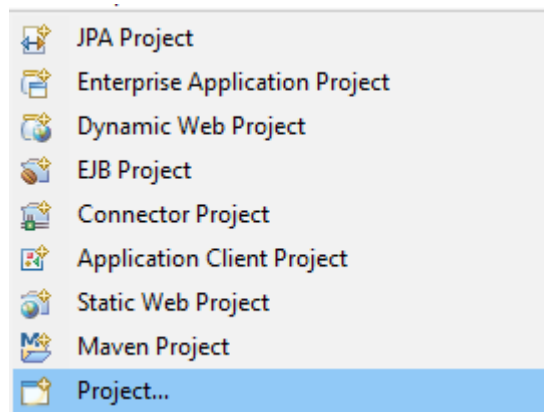
Va até a aba **file** na barra superior



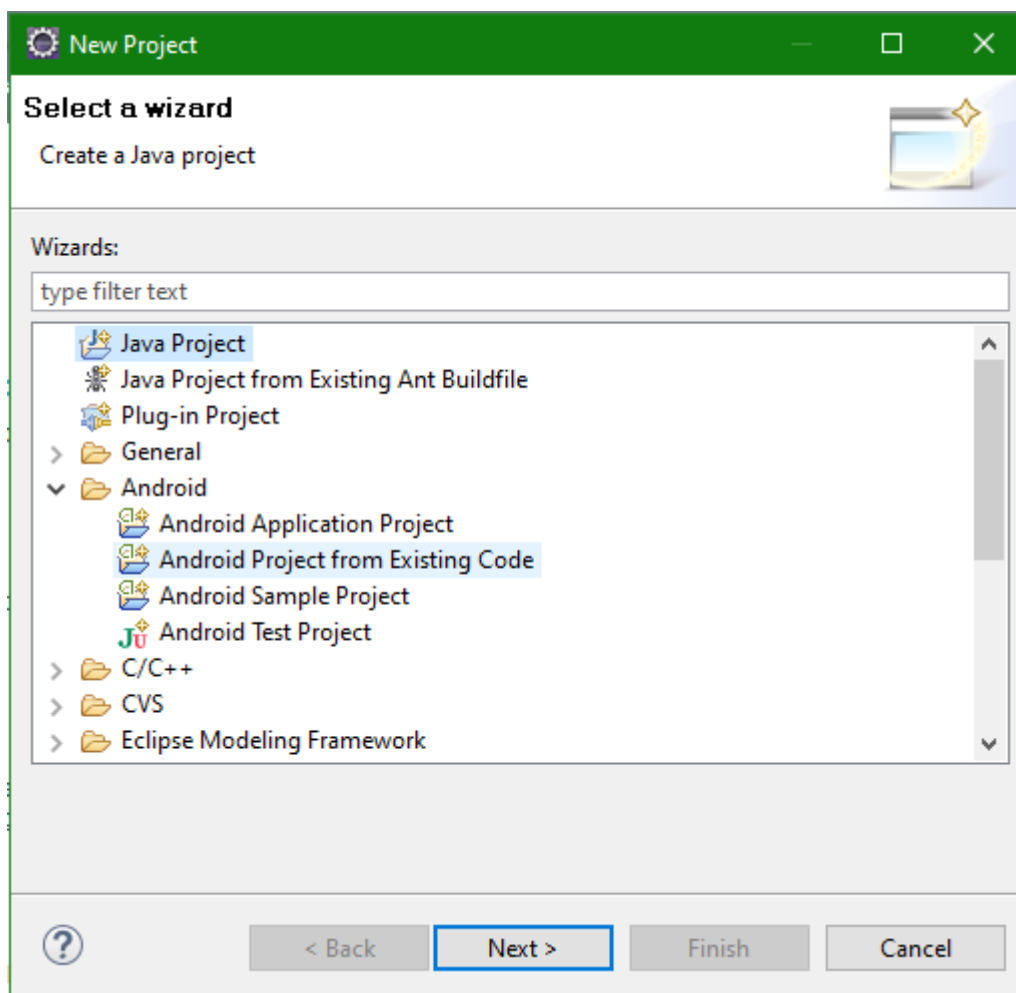
Em seguida vá até **new**



Em new vá até **Project**



Abra a pasta **Android**, e selecione **Android Project From existing code**



Após isso clique em **next**, será exibida essa outra tela.

**Import Projects**

Select a directory to search for existing Android projects

Root Directory:  Browse...

Projects:

Project to Import	New Project Name

Select All  
Deselect All  
Refresh

☐ Copy projects into workspace

Working sets

☐ Add project to working sets

Working sets:  Select...

? < Back Next > Finish Cancel

Em **Root Directory**, coloque o caminho do seu projeto criado pelo Pydroid, ou de **browse** e encontre-o pelo Explorer.

Root Directory:  Browse...

Após isso aparecerão várias caixas marcadas

Root Directory:  Browse...

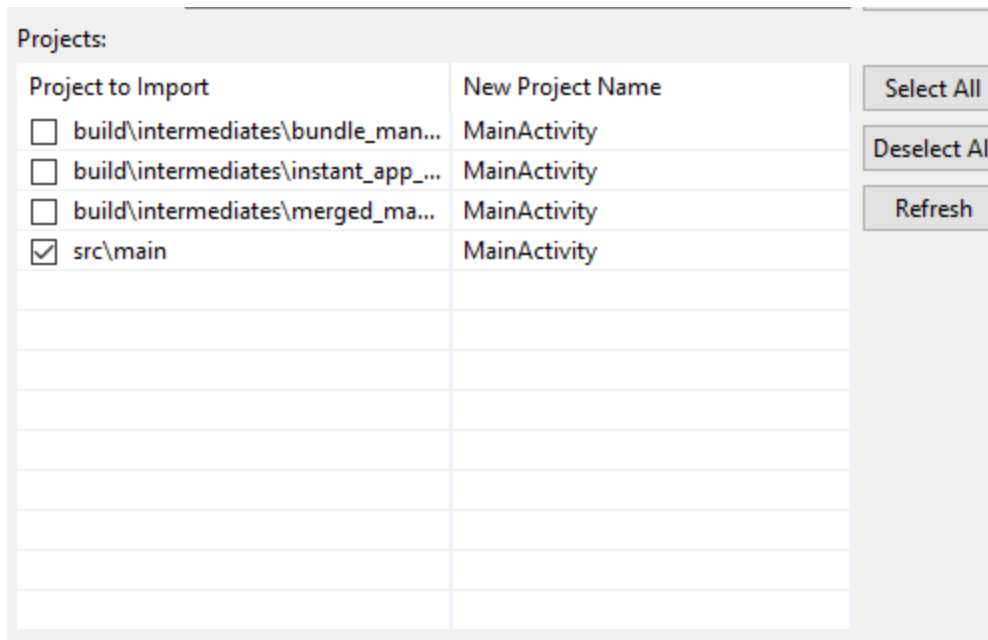
Projects:

Project to Import	New Project Name
<input checked="" type="checkbox"/> build\intermediates\bundle_man...	MainActivity
<input checked="" type="checkbox"/> build\intermediates\instant_app_...	MainActivity
<input checked="" type="checkbox"/> build\intermediates\merged_ma...	MainActivity
<input checked="" type="checkbox"/> src\main	MainActivity

Select All  
Deselect All  
Refresh

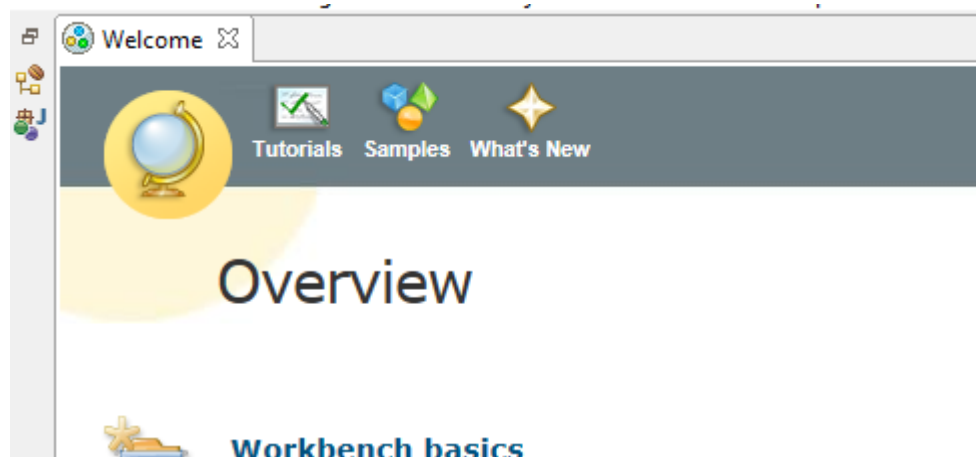
☐ Copy projects into workspace

Desmarque todas deixando somente a **src/main**



Após isso clique em **finish**.

Mesmo clicando em **finish** você voltara para a tela inicial do eclipse, para abrir o projeto vá até o canto superior esquerdo

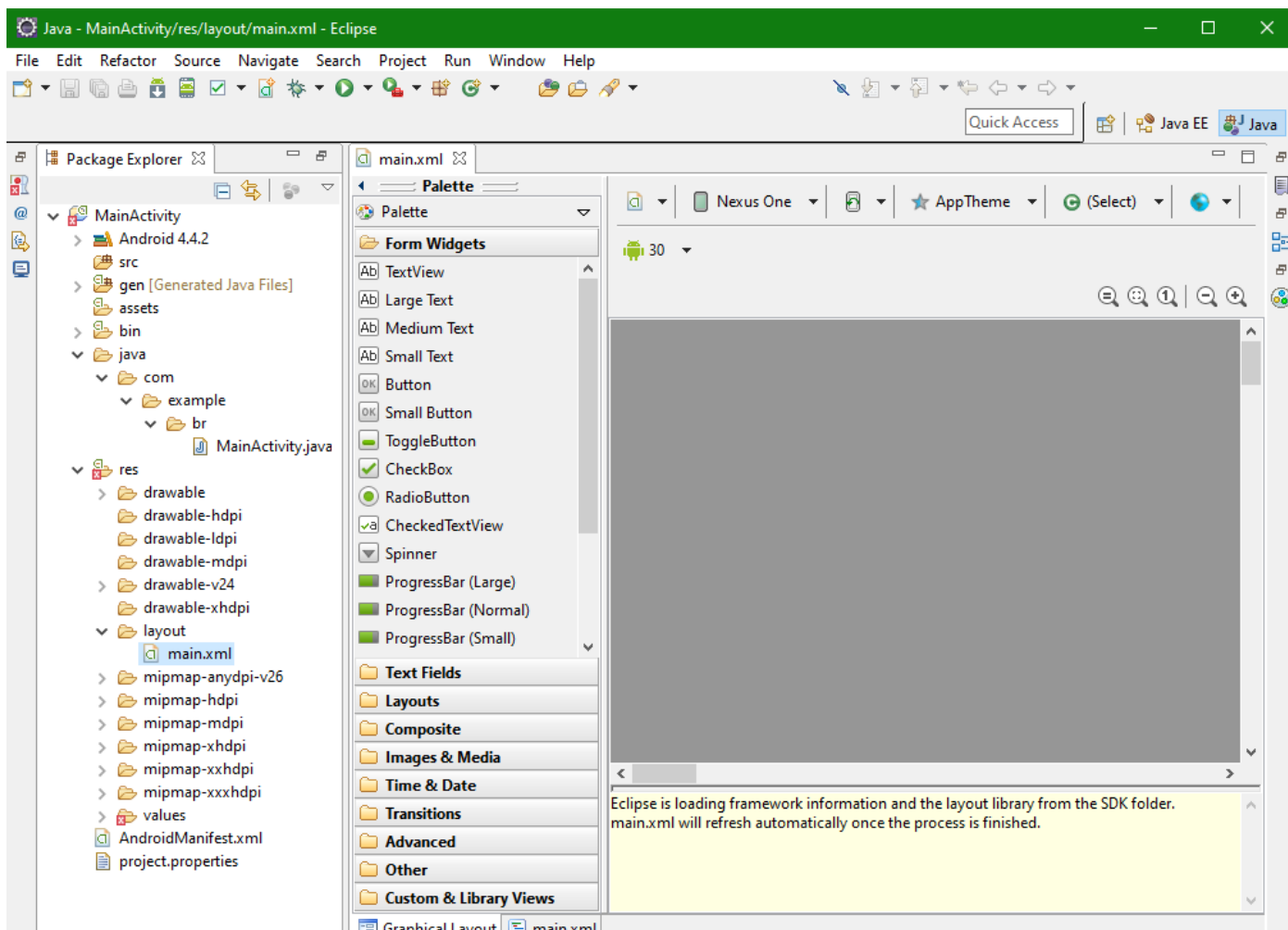


Clique no símbolo de exibição do Projeto



Abra a pasta **mainActivity**

E enfim o projeto está criado no Eclipse Luna



Enfim a tela inicial do projeto amigos, com isso vamos ao próximo passo e um dos mais importantes para sua codificação.

## 5. Passo – Configurações na IDE e instalação do Android 19.

**Este é o último passo para quem irá rodar o Android no eclipse-luna.**

Bom como iremos rodar o Android numa IDE antiga, alguns arranjos técnicos precisam ser feitos.

O primeiro deles é a instalação do SDK-Platform 19 do Android, pois as versões superiores não possuem suporte ao Luna.

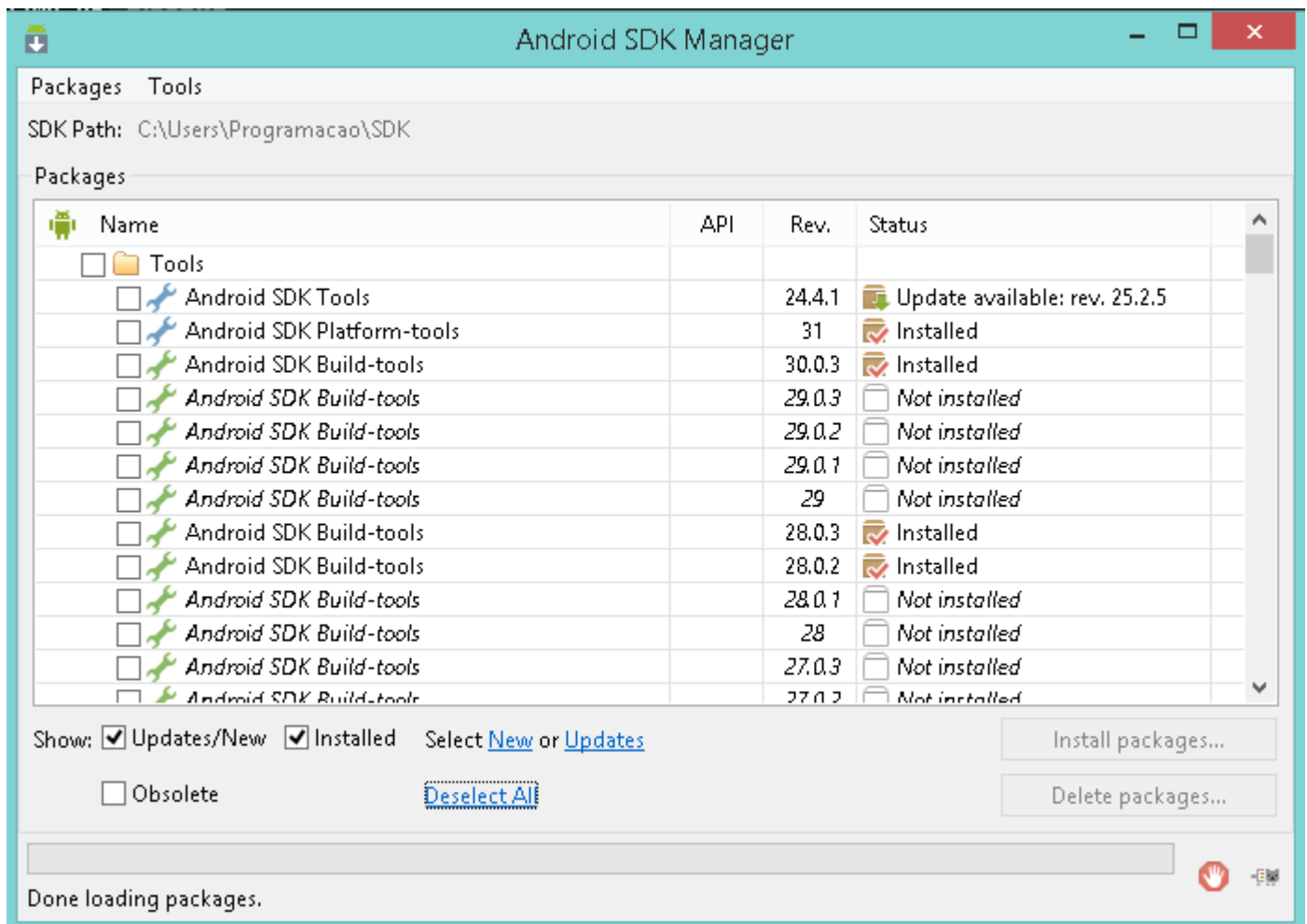
Bom para baixar é muito simples vá até o canto superior esquerdo, bem aqui



Em seguida clique nesse ícone



Será mostrada então a tela do SDK Manager com todas as versões do Android disponíveis para download



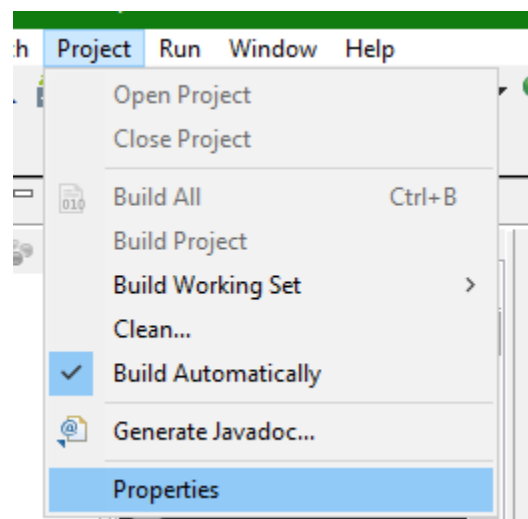
Procure e marque a versão **19**

<input type="checkbox"/>	Android 4.4.2 (API 19)			
<input checked="" type="checkbox"/>	SDK Platform	19	4	Installed

Após isso faça a instalação clicando em **Install packages**.

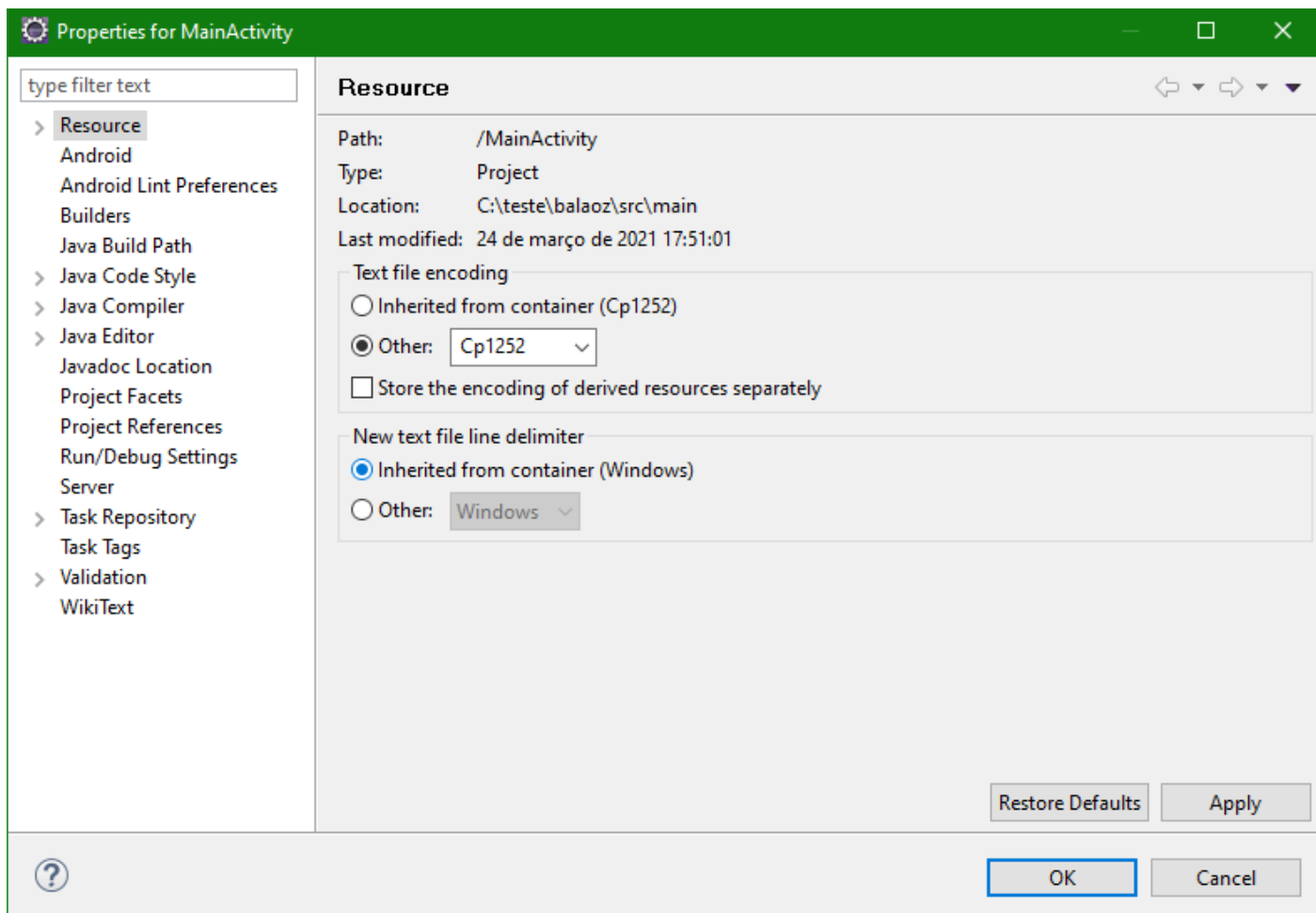
Feito a instalação vamos para a próxima configuração.

Na barra superior, vá até **Project**

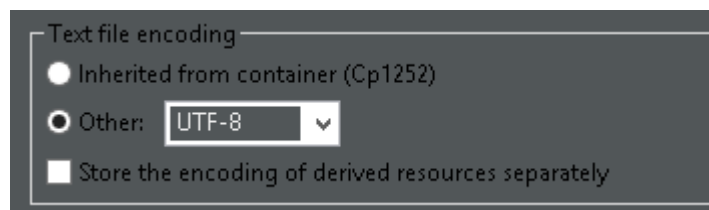


Clique em **properties** será exibido essa tela



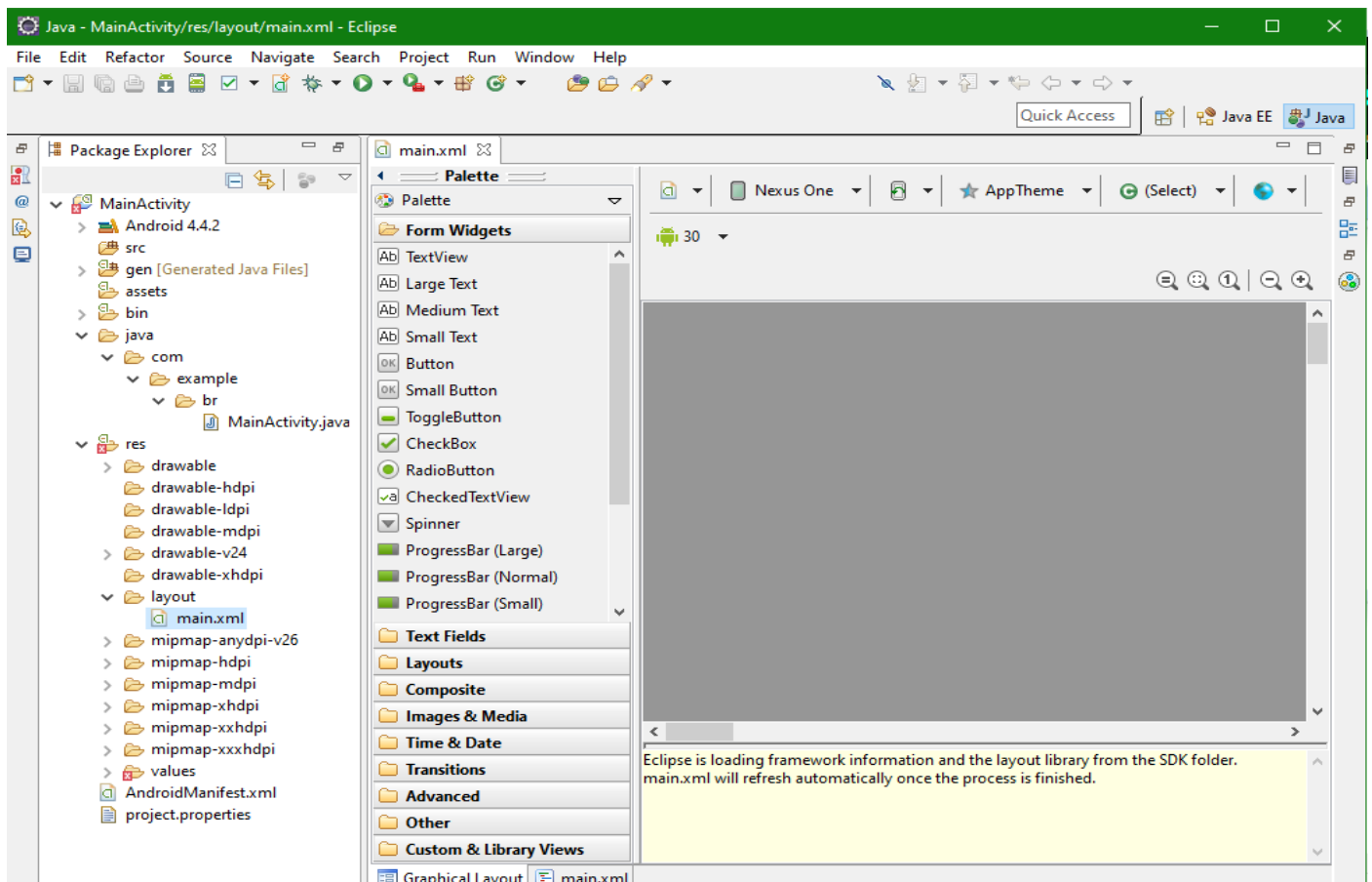


Em **Text file encoding**, selecione a opção Other e escolha UTF-8 como padrão texto.

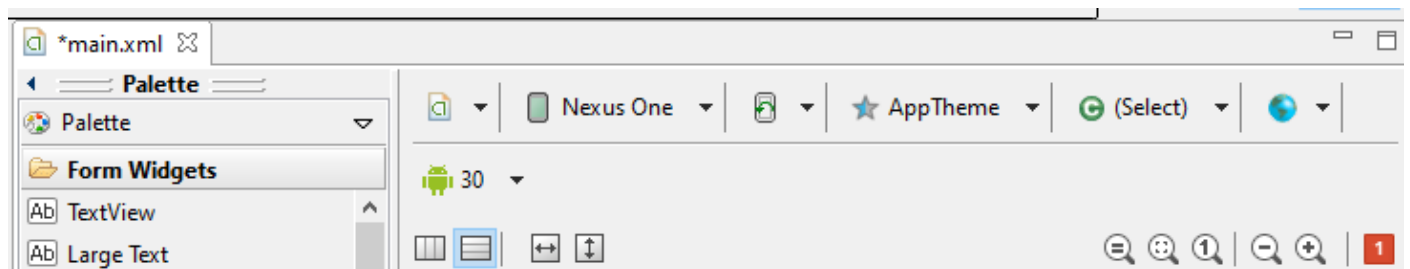


Feito isso vamos para a próxima e última configuração.

Abra seu arquivo **main.xml**, na pasta **res\layout**

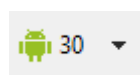


Na barra superior ao **view** do seu arquivo main, estarão algumas opções



Primeiro iremos mudar o Android do 30, para o 19.

Clique no ícone

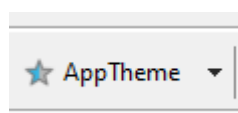


E troque para o 19

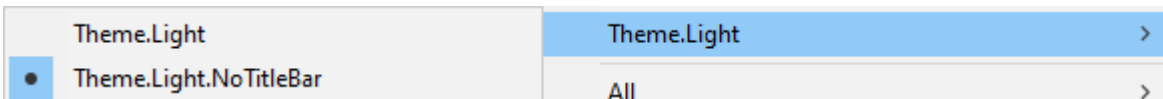


Agora iremos configurar o **view**

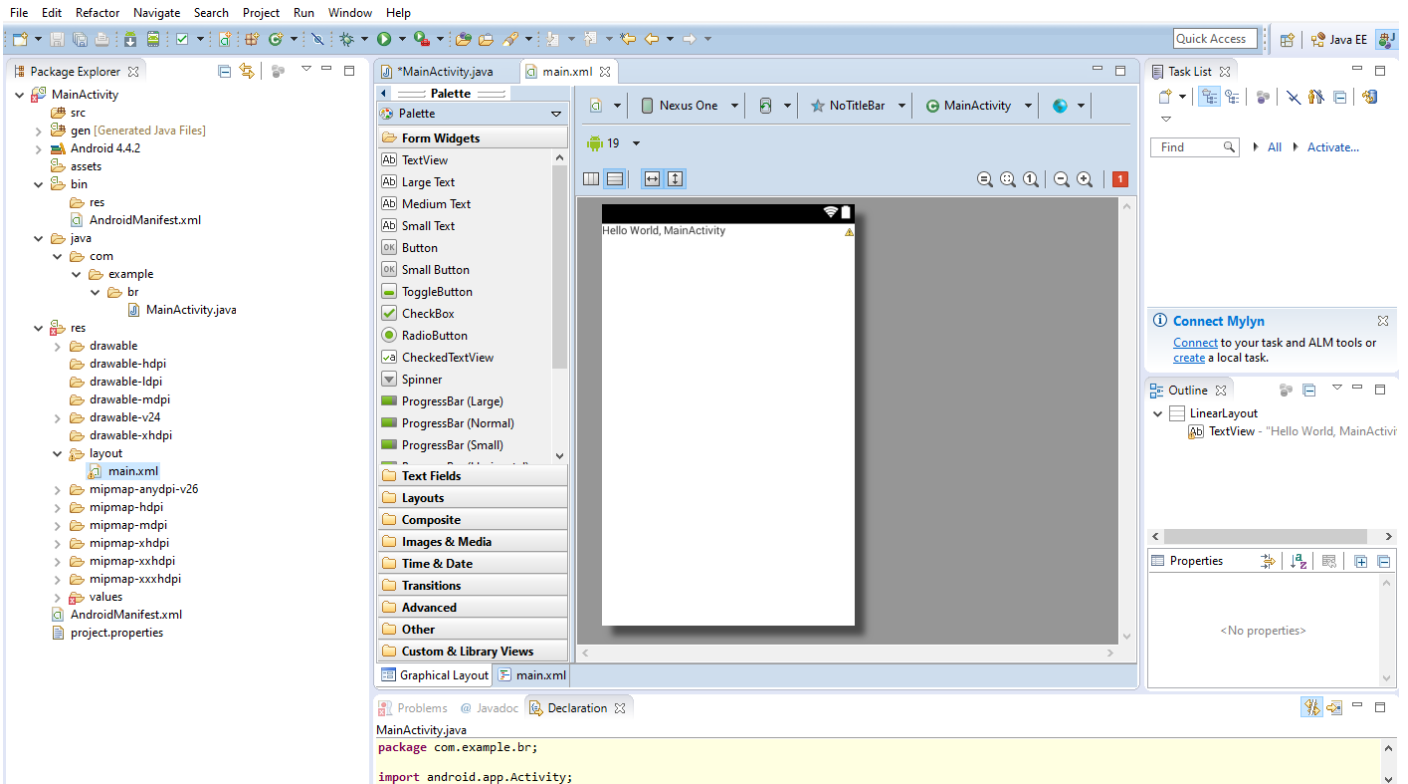
Clique em **AppTheme**



Em **AppTheme** vá até **Theme.Light** e selecione a opção **Theme.light.noTitleBar**



E prontinho, sua ide Eclipse Luna esta configurada para codificar APP's Android.

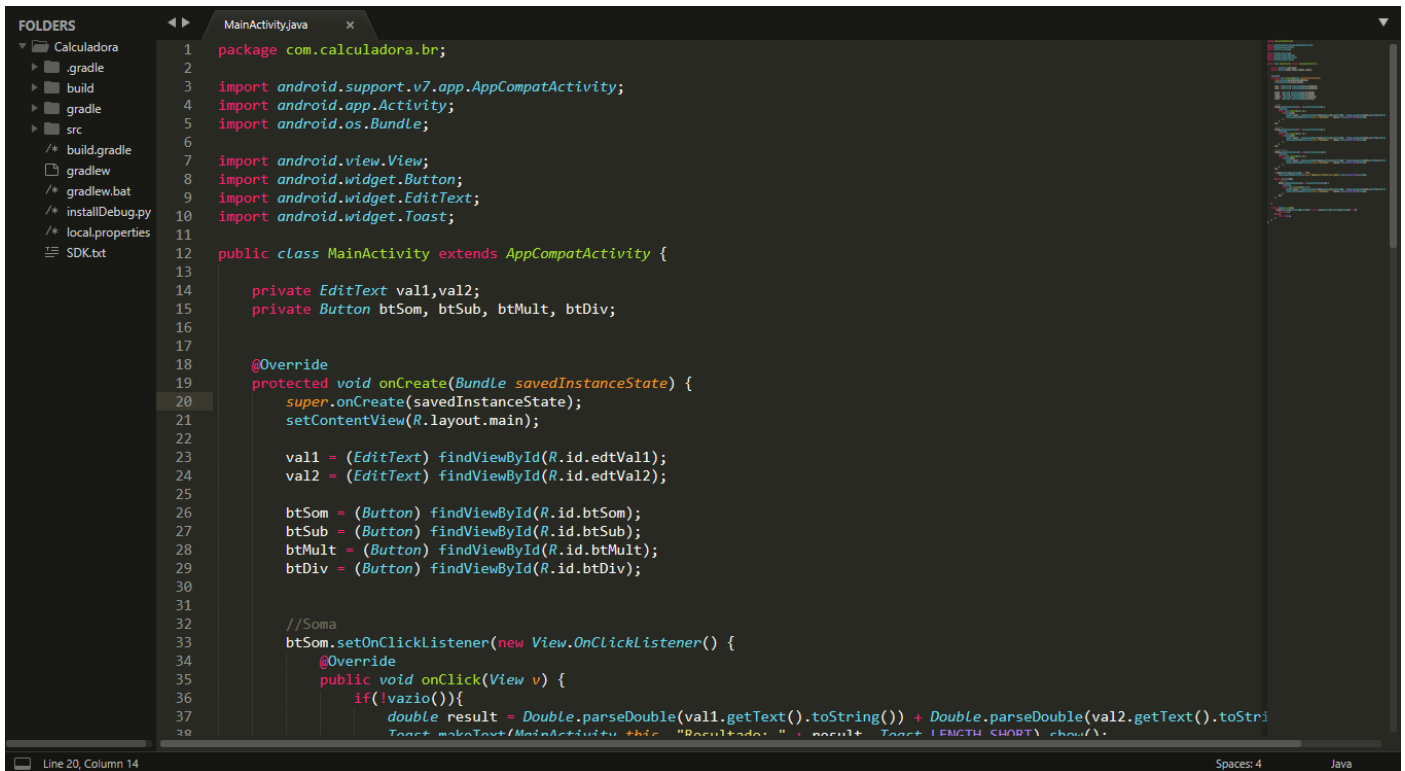


E esse e o “fim” do tutorial para aqueles que vão rodar no Eclipse Luna, obrigado por terem visto até aqui, **vão até a última seção** InstallDebug **para aprenderem a rodar o programa criado no Luna no celular!**

## Outras IDES

Assim como já citado anteriormente outras IDES são capazes de codificar o Android.

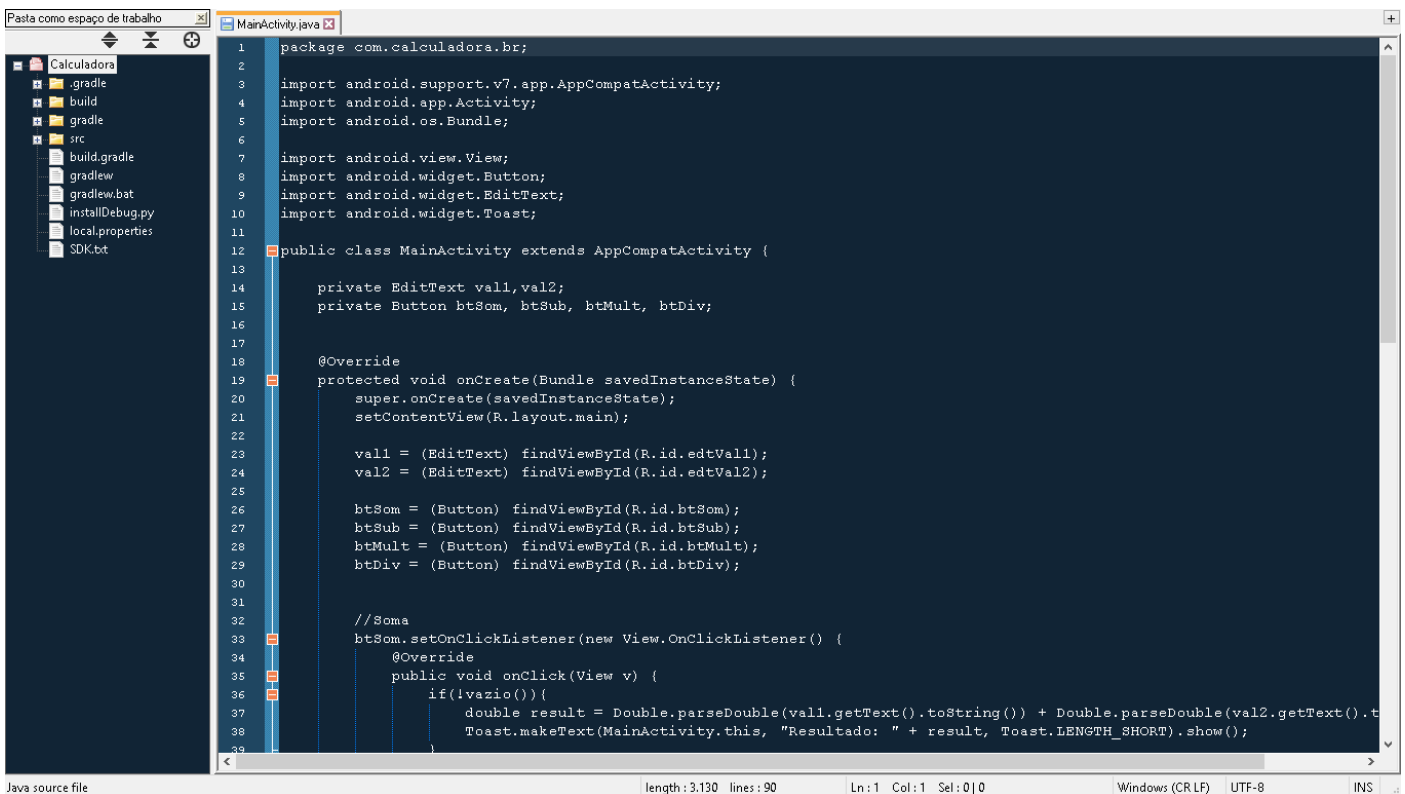
## Codificação no sublime



The screenshot shows the Sublime Text editor with a file named MainActivity.java. The code is for an Android application package com.calculadora.br. It imports necessary Android classes and defines the MainActivity class, which extends AppCompatActivity. The class contains private variables for EditTexts (val1, val2) and Buttons (btSom, btSub, btMult, btDiv). The onCreate method is overridden to initialize these variables and find views by ID. A click listener is set for the btSom button, which calculates the sum of the values in val1 and val2 and displays the result using Toast.

```
1 package com.calculadora.br;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.app.Activity;
5 import android.os.Bundle;
6
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.Toast;
11
12 public class MainActivity extends AppCompatActivity {
13
14     private EditText val1, val2;
15     private Button btSom, btSub, btMult, btDiv;
16
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.main);
22
23         val1 = (EditText) findViewById(R.id.edtVal1);
24         val2 = (EditText) findViewById(R.id.edtVal2);
25
26         btSom = (Button) findViewById(R.id.btSom);
27         btSub = (Button) findViewById(R.id.btSub);
28         btMult = (Button) findViewById(R.id.btMult);
29         btDiv = (Button) findViewById(R.id.btDiv);
30
31
32         //Soma
33         btSom.setOnClickListener(new View.OnClickListener() {
34             @Override
35             public void onClick(View v) {
36                 if(!vazio()){
37                     double result = Double.parseDouble(val1.getText().toString()) + Double.parseDouble(val2.getText().toString());
38                     Toast.makeText(MainActivity.this, "Resultado: " + result, Toast.LENGTH_SHORT).show();
39                 }
40             }
41         });
42     }
43 }
```

## Codificação no Notepad ++

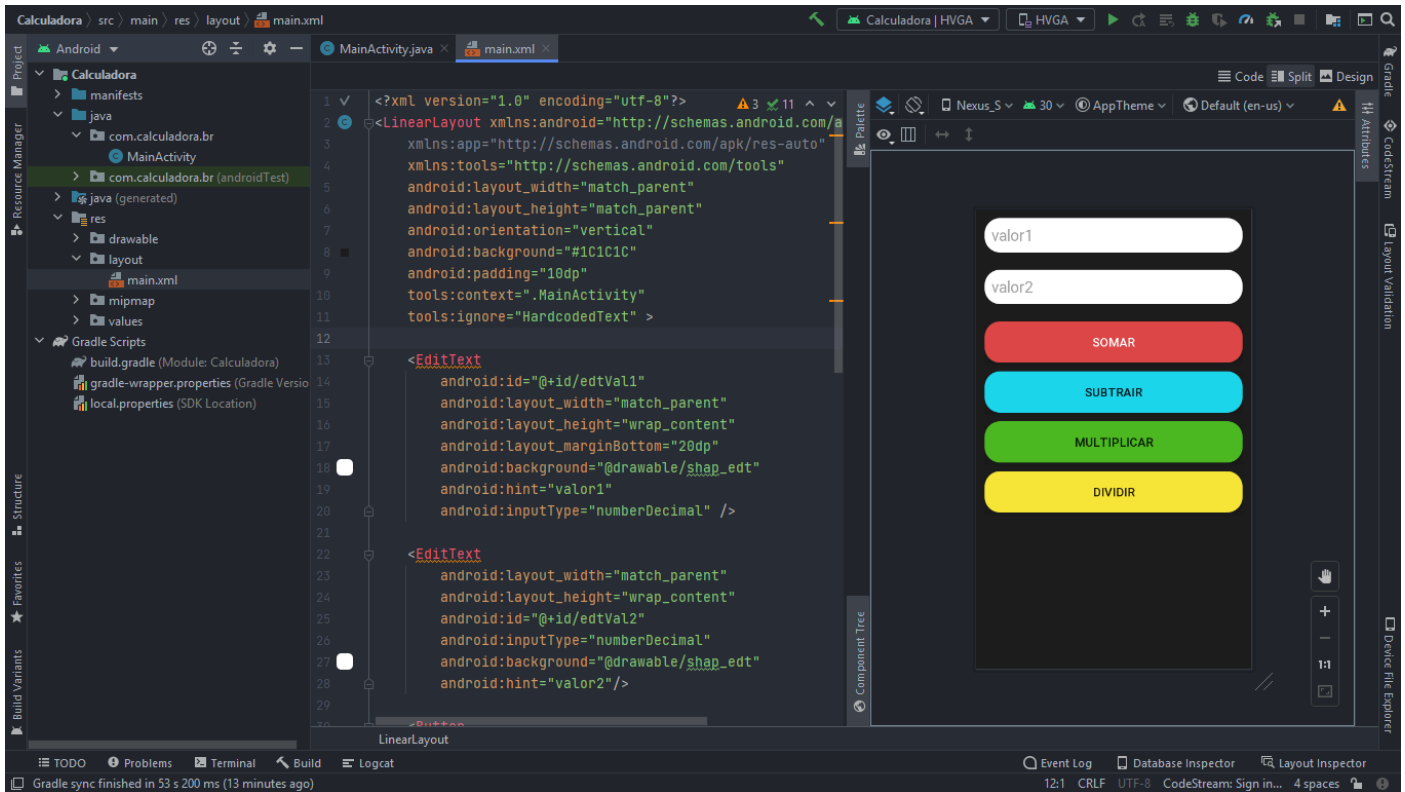


The screenshot shows the Notepad++ editor with a file named MainActivity.java. The code is identical to the one in the Sublime Text editor. The interface is different, with a menu bar and a toolbar at the top. The status bar at the bottom shows the file is a Java source file, has a length of 3,130, and 90 lines.

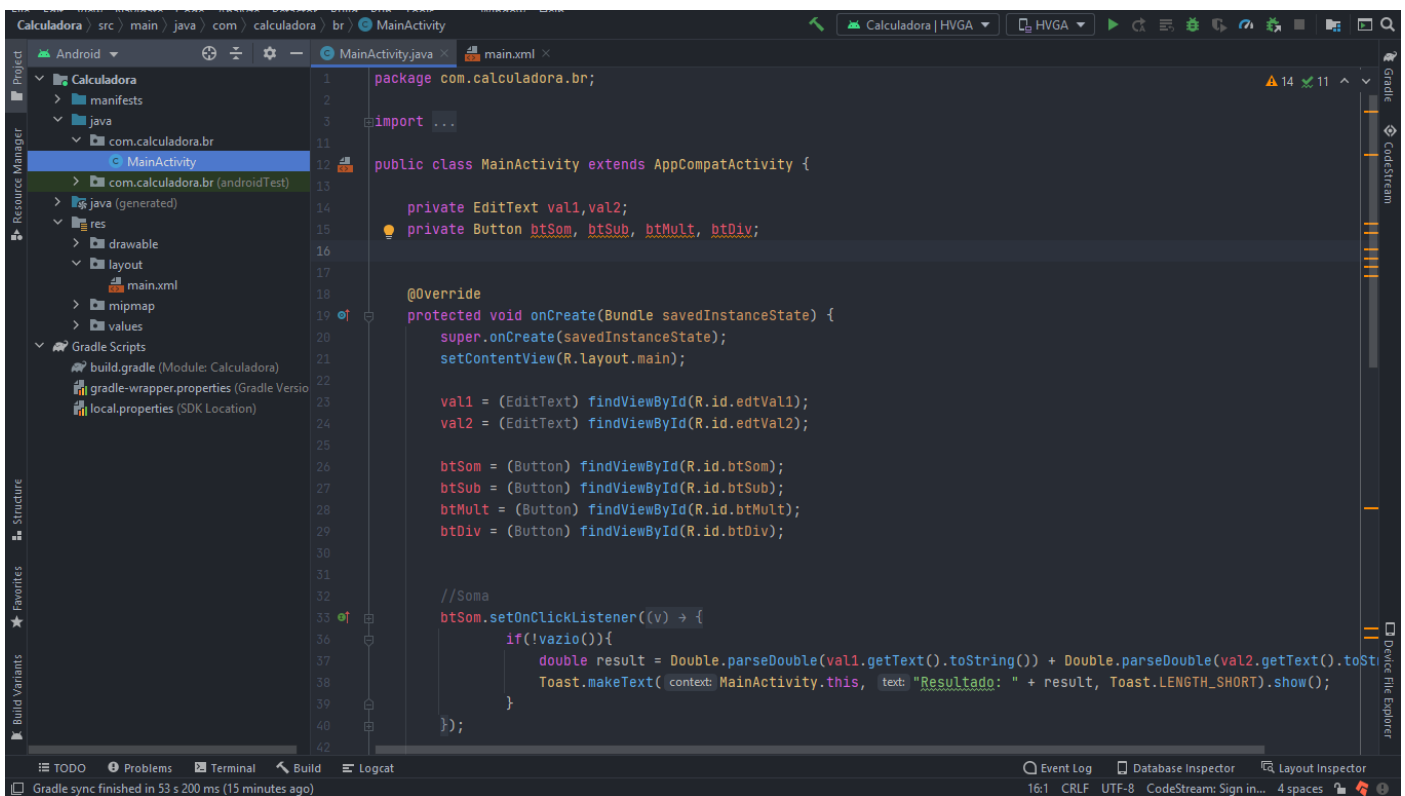
```
1 package com.calculadora.br;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.app.Activity;
5 import android.os.Bundle;
6
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.Toast;
11
12 public class MainActivity extends AppCompatActivity {
13
14     private EditText val1, val2;
15     private Button btSom, btSub, btMult, btDiv;
16
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.main);
22
23         val1 = (EditText) findViewById(R.id.edtVal1);
24         val2 = (EditText) findViewById(R.id.edtVal2);
25
26         btSom = (Button) findViewById(R.id.btSom);
27         btSub = (Button) findViewById(R.id.btSub);
28         btMult = (Button) findViewById(R.id.btMult);
29         btDiv = (Button) findViewById(R.id.btDiv);
30
31
32         //Soma
33         btSom.setOnClickListener(new View.OnClickListener() {
34             @Override
35             public void onClick(View v) {
36                 if(!vazio()){
37                     double result = Double.parseDouble(val1.getText().toString()) + Double.parseDouble(val2.getText().toString());
38                     Toast.makeText(MainActivity.this, "Resultado: " + result, Toast.LENGTH_SHORT).show();
39                 }
40             }
41         });
42     }
43 }
```

# Codificação Jetbrains IntelliJ IDEA

## Xml:



## Java:



## InstallDebug.py

O Pydroid também conta com um arquivo para rodar o programa em um celular ou emulador.

Antes **você precisa ligar um emulador**, ou **conectar um celular no PC através de usb**

Para usá-lo vá até a pasta do seu projeto Pydroid, e em seguida execute o arquivo **installDebug.py** e aguarde a instalação e abertura do projeto em seu celular ou emulador.

Nome	Data de modificaç...	Tipo	Tamanho
 .gradle	24/03/2021 18:59	Pasta de arquivos	
 .idea	25/03/2021 18:18	Pasta de arquivos	
 build	24/03/2021 19:02	Pasta de arquivos	
 gradle	24/03/2021 18:58	Pasta de arquivos	
 src	24/03/2021 18:58	Pasta de arquivos	
 build.gradle	24/03/2021 18:58	Arquivo GRADLE	2 KB
 gradlew	24/03/2021 18:58	Arquivo	6 KB
 gradlew.bat	24/03/2021 18:58	Arquivo em Lotes ...	3 KB
 <u>installDebug.py</u>	24/03/2021 18:58	Python File	1 KB
 local.properties	24/03/2021 18:58	Arquivo PROPERT...	1 KB
 SDK.txt	24/03/2021 18:58	Documento de Te...	1 KB

Finalmente chegamos ao fim do tutorial, espero que tenha ficado claro e compreensível para todos.

Quaisquer dúvidas ou report de erros, favor nos contatar em algum meio de comunicação e tentaremos ajudá-lo.

Só pra lembrar, quem participa deste projeto ate o momento é o [Thigos \(Thiago M.\)](#) e [Cayozada \(Caio\)](#).