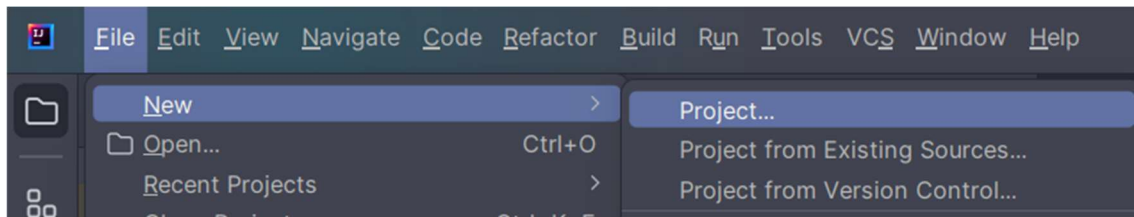
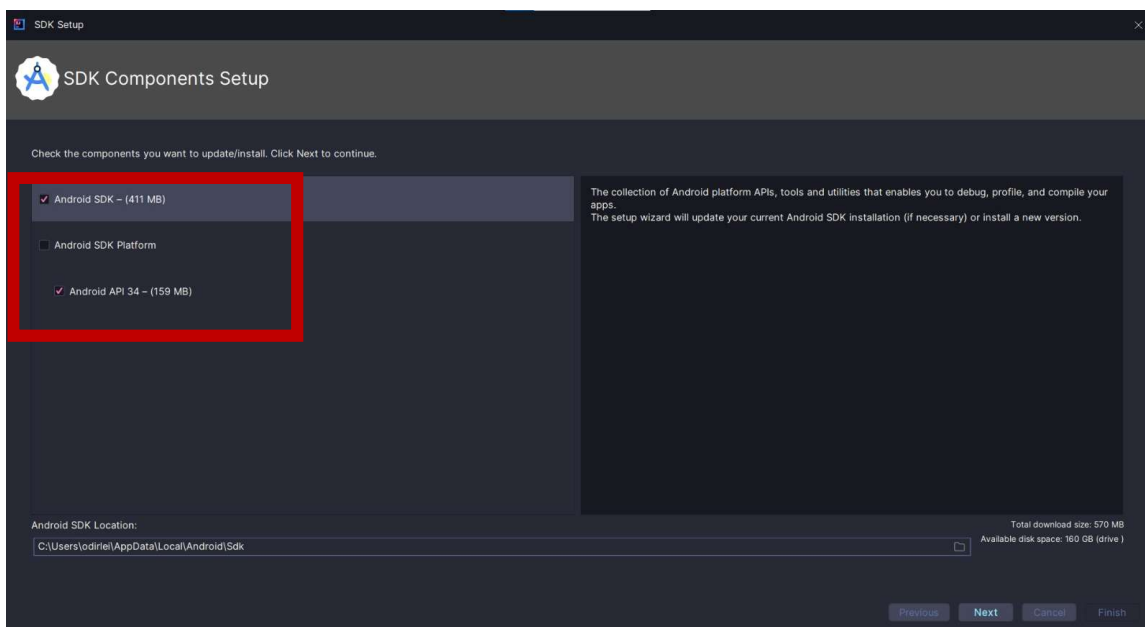
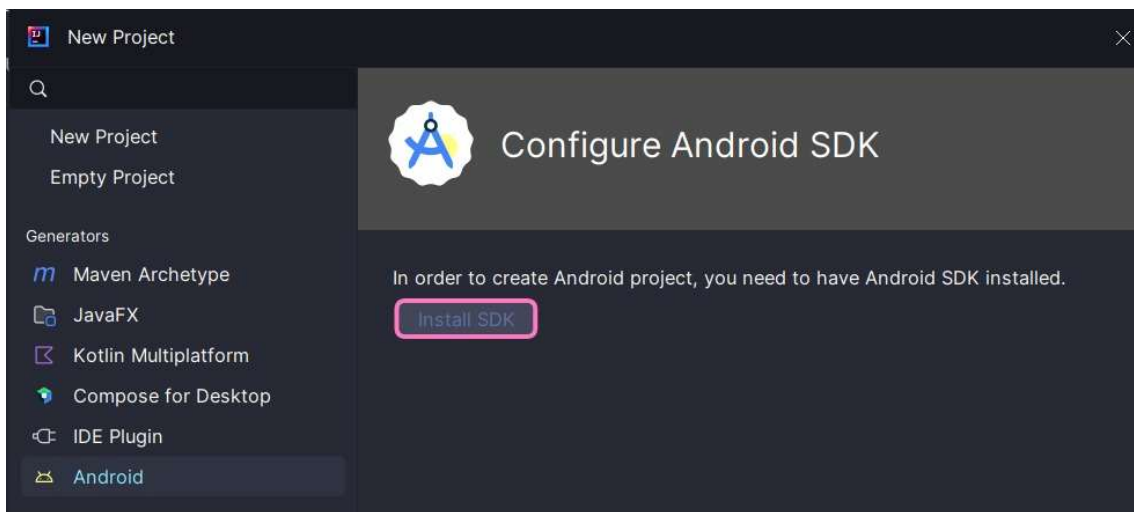
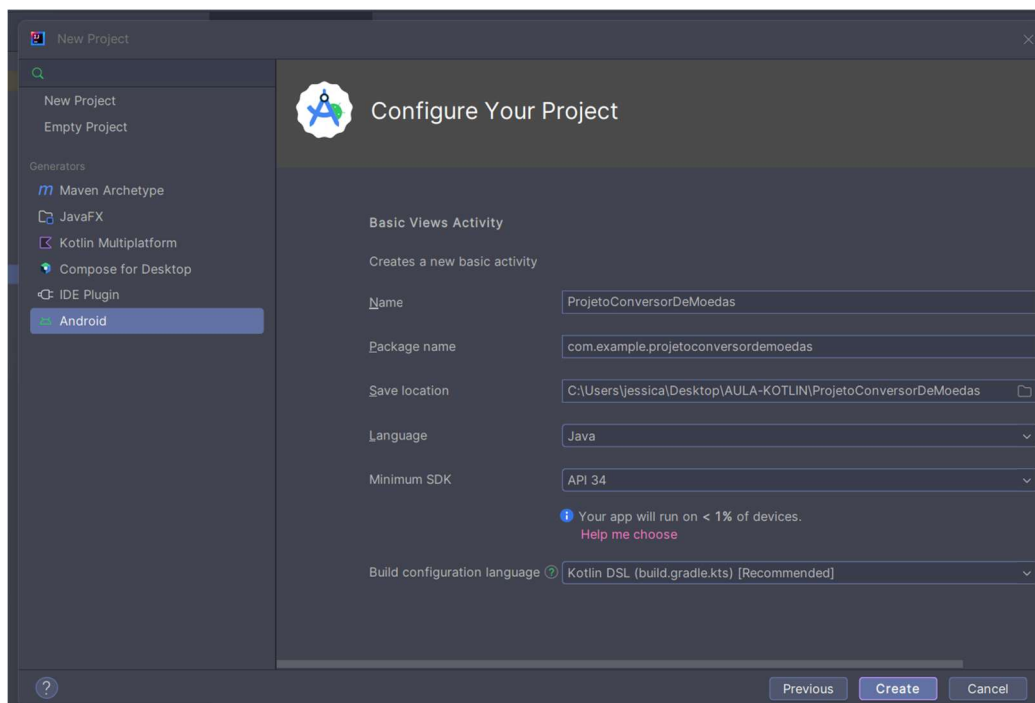
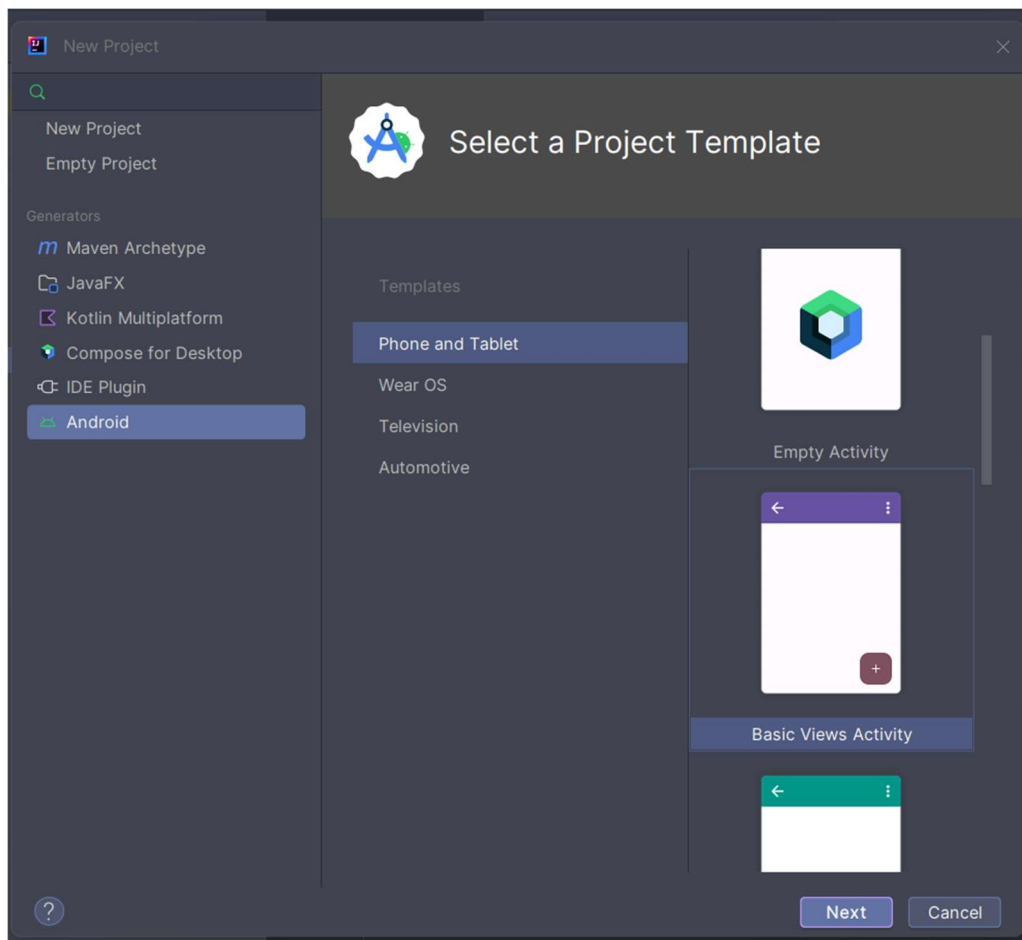


CRIAÇÃO DO PROJETO



Selecione as opções: Android > Phone and Tablet > **Basic Views Activity** ou **Basic Activity**





- Preencha o nome - HelloWorld
- Selecione a Linguagem **Java**

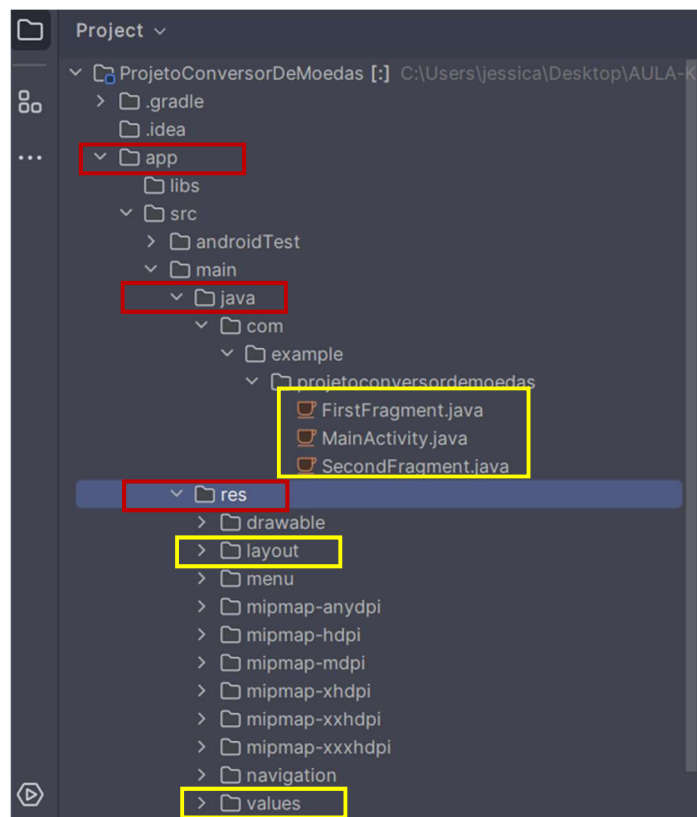
- SDK – **API 34**
- Build – **Kotlin**

Clicar em Create

Ele solicitará a aprovação de instalação de um arquivo chamado abs. Clique em aceitar.
Caso não apareça, será necessário a criação de um novo projeto.

Isso acontecerá apenas na primeira vez.

PROJETO



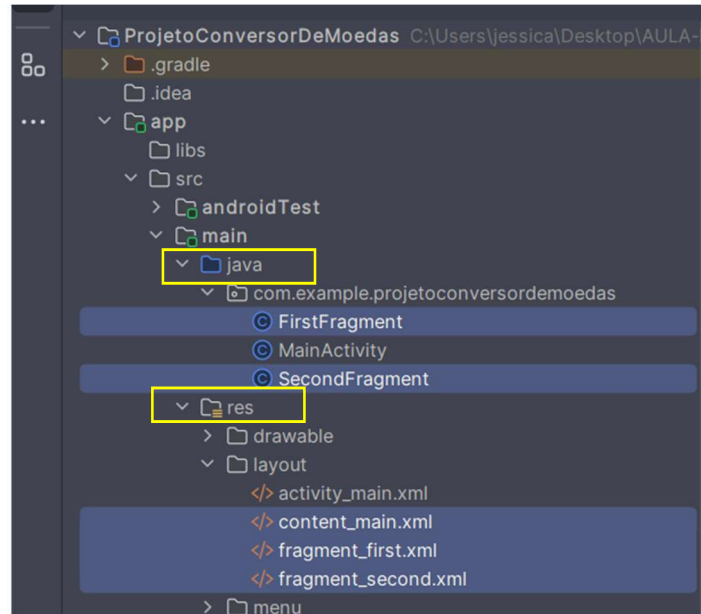
Trabalharemos com 3 arquivos:

- Java -> Escrevemos a estrutura lógica do projeto
- Res > Layout -> Montamos a tela (layout), podendo ser com drop and drag (clica e arrasta) ou com código.
- Res > Values -> Armazenamos textos que serão utilizados na aplicação em variáveis que serão chamadas na tela de Layout

1 PASSO:

Limpar arquivos que não serão utilizados.

Como faremos apenas 1 tela, podemos limpar as telas que estão a mais no nosso projeto.

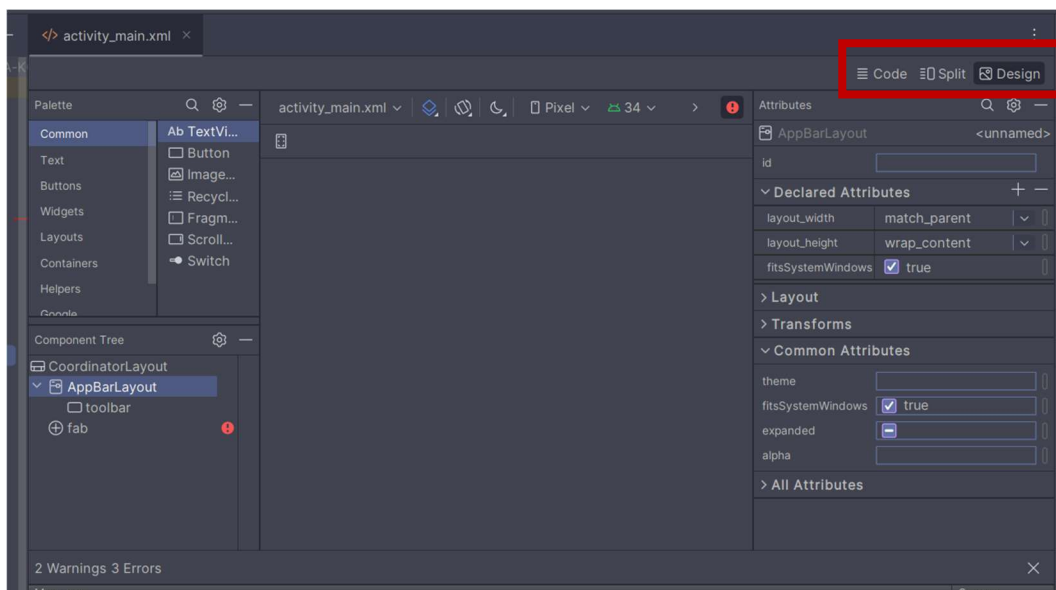


Faremos a limpeza dos arquivos nas pastas Res > Layout e Java. Manteremos apenas os arquivos Activity Main.

2 PASSO:

Vamos começar a estruturar nosso layout:

RES > LAYOUT > ACTIVITY_MAIN



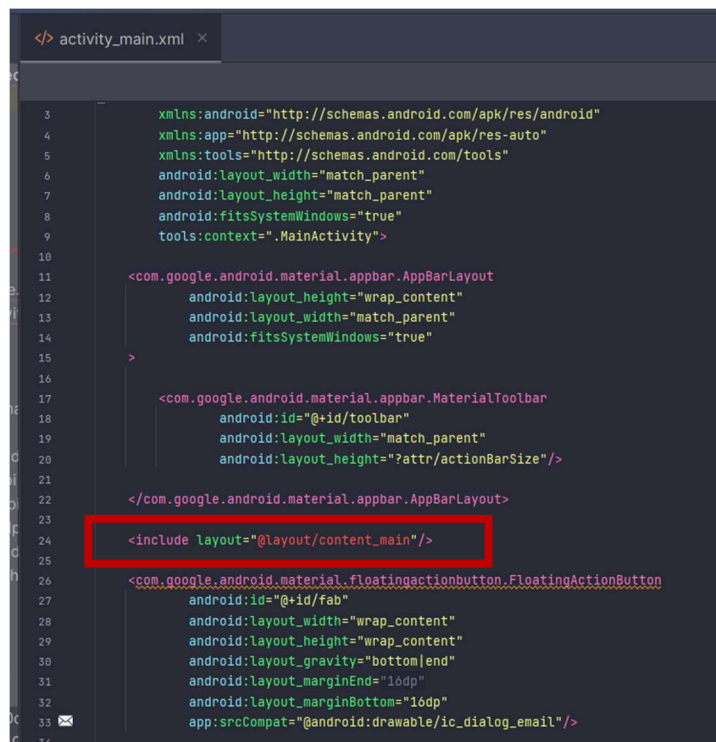
O IntelliJ disponibiliza 3 opções de visualização para montar o código.

1 – **CODE** -> Apenas o código

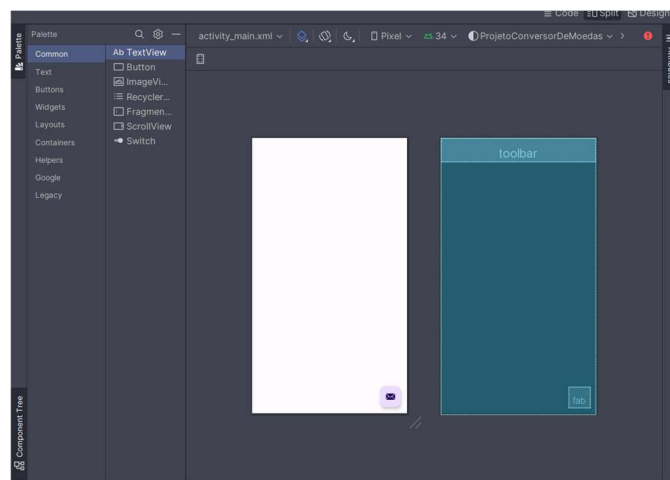
2 – **SPLIT** -> Código e Design (Clica e Arrasta)

3 – **DESIGN** -> Apenas o Clica e Arrasta

Entraremos no modo SPLIT para limpar uma linha que impede de visualizarmos o espaço de trabalho. Remova a linha 24, pois ela está chamando o layout que foi desenvolvido em um dos layouts que excluímos.



```
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_width="match_parent"
7      android:layout_height="match_parent"
8      android:fitsSystemWindows="true"
9      tools:context=".MainActivity">
10
11      <com.google.android.material.appbar.AppBarLayout
12          android:layout_height="wrap_content"
13          android:layout_width="match_parent"
14          android:fitsSystemWindows="true"
15      >
16
17          <com.google.android.material.appbar.MaterialToolbar
18              android:id="@+id/toolbar"
19              android:layout_width="match_parent"
20              android:layout_height="?attr/actionBarSize"/>
21
22      </com.google.android.material.appbar.AppBarLayout>
23
24      <include layout="@layout/content_main"/>
25
26      <com.google.android.material.floatingactionbutton.FloatingActionButton
27          android:id="@+id/fab"
28          android:layout_width="wrap_content"
29          android:layout_height="wrap_content"
30          android:layout_gravity="bottom|end"
31          android:layout_marginEnd="16dp"
32          android:layout_marginBottom="16dp"
33          app:srcCompat="@android:drawable/ic_dialog_email"/>
34  </android.support.design.widget.CoordinatorLayout>
```

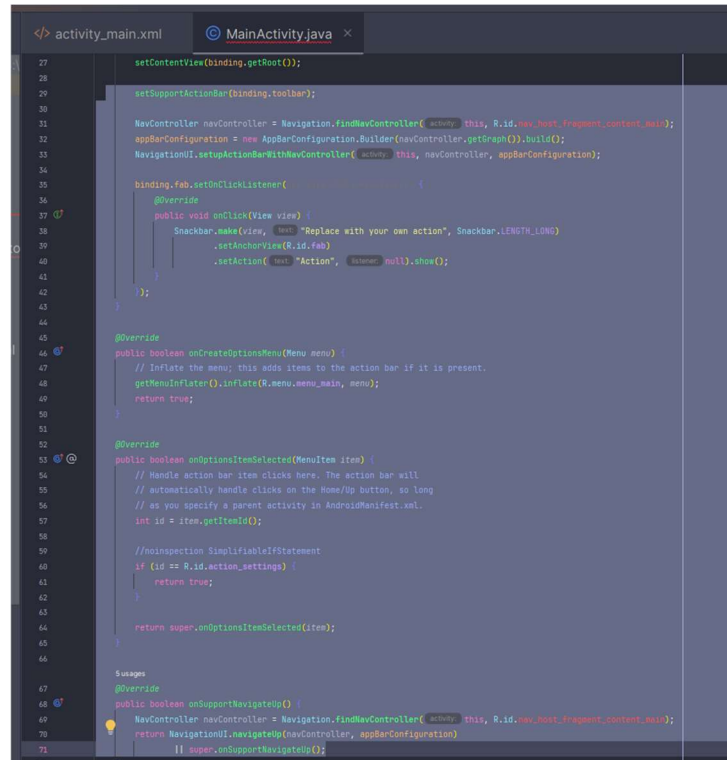


Automaticamente aparecerá a tela para trabalharmos.

1. Limpeza de Código

Vamos para o arquivo Java > **MainActivity**

Apagar código da linha 29 à 71

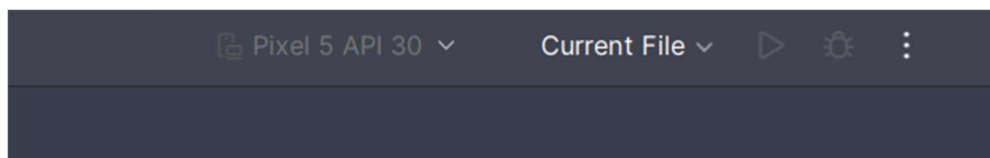


```
27 setContentView(binding.getRoot());
28
29 setSupportActionBar(binding.toolbar);
30
31 NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);
32 appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();
33 NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
34
35 binding.fab.setOnClickListener(new View.OnClickListener() {
36     @Override
37     public void onClick(View view) {
38         Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
39             .setAnchorView(R.id.fab)
40             .setAction("Action", listener null).show();
41     }
42 });
43
44
45 @Override
46 public boolean onCreateOptionsMenu(Menu menu) {
47     // Inflate the menu; this adds items to the action bar if it is present.
48     getMenuInflater().inflate(R.menu.menu_main, menu);
49     return true;
50 }
51
52 @Override
53 public boolean onOptionsItemSelected(MenuItem item) {
54     // Handle action bar item clicks here. The action bar will
55     // automatically handle clicks on the Home/Up button, so long
56     // as you specify a parent activity in AndroidManifest.xml.
57     int id = item.getItemId();
58
59     //noinspection SimplifiableIfStatement
60     if (id == R.id.action_settings) {
61         return true;
62     }
63
64     return super.onOptionsItemSelected(item);
65 }
66
67 @Override
68 public boolean onSupportNavigateUp() {
69     NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);
70     return NavigationUI.navigateUp(navController, appBarConfiguration)
71         || super.onSupportNavigateUp();
72 }
```

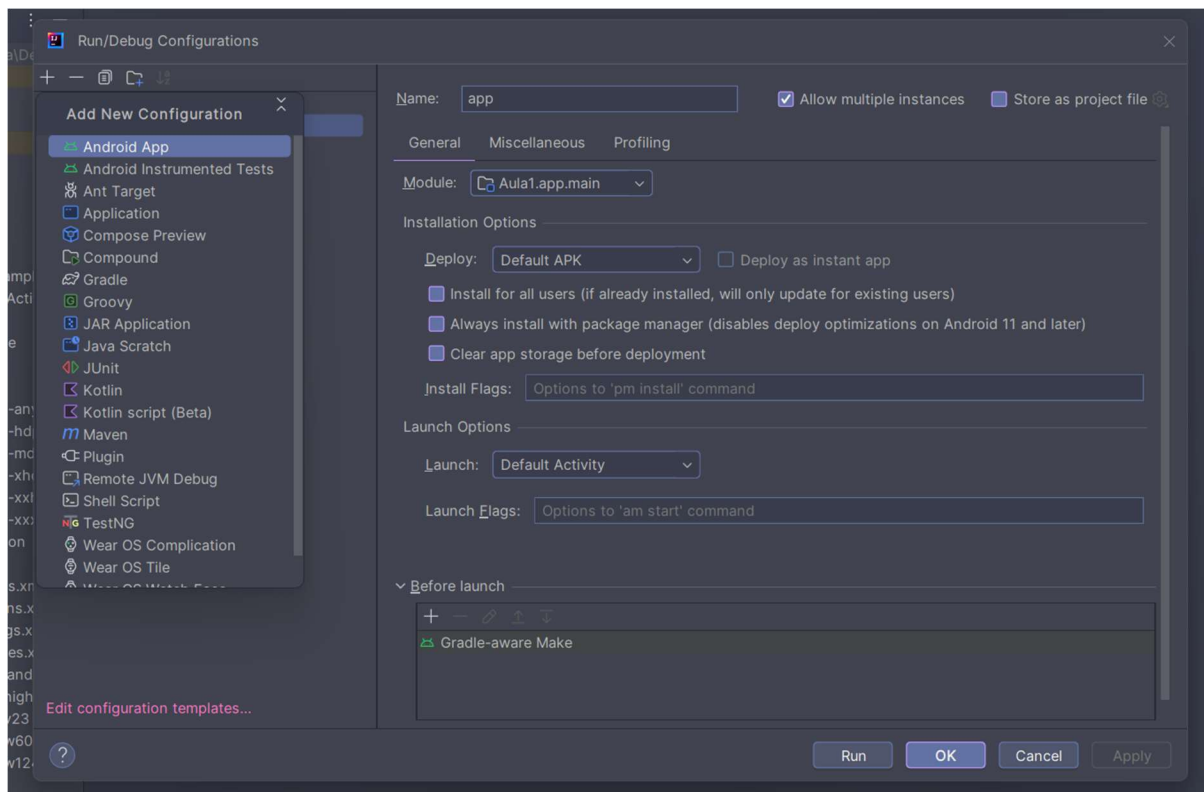
RODAR O PROJETO

Para rodar o projeto, precisamos configurar algumas coisas.

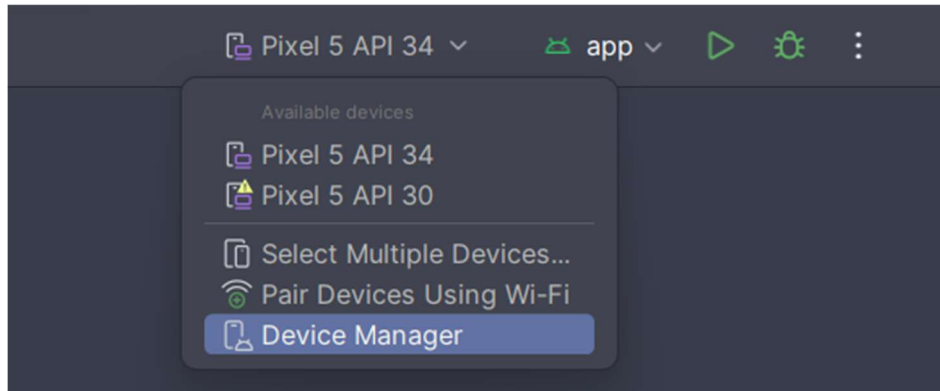
1. Current File



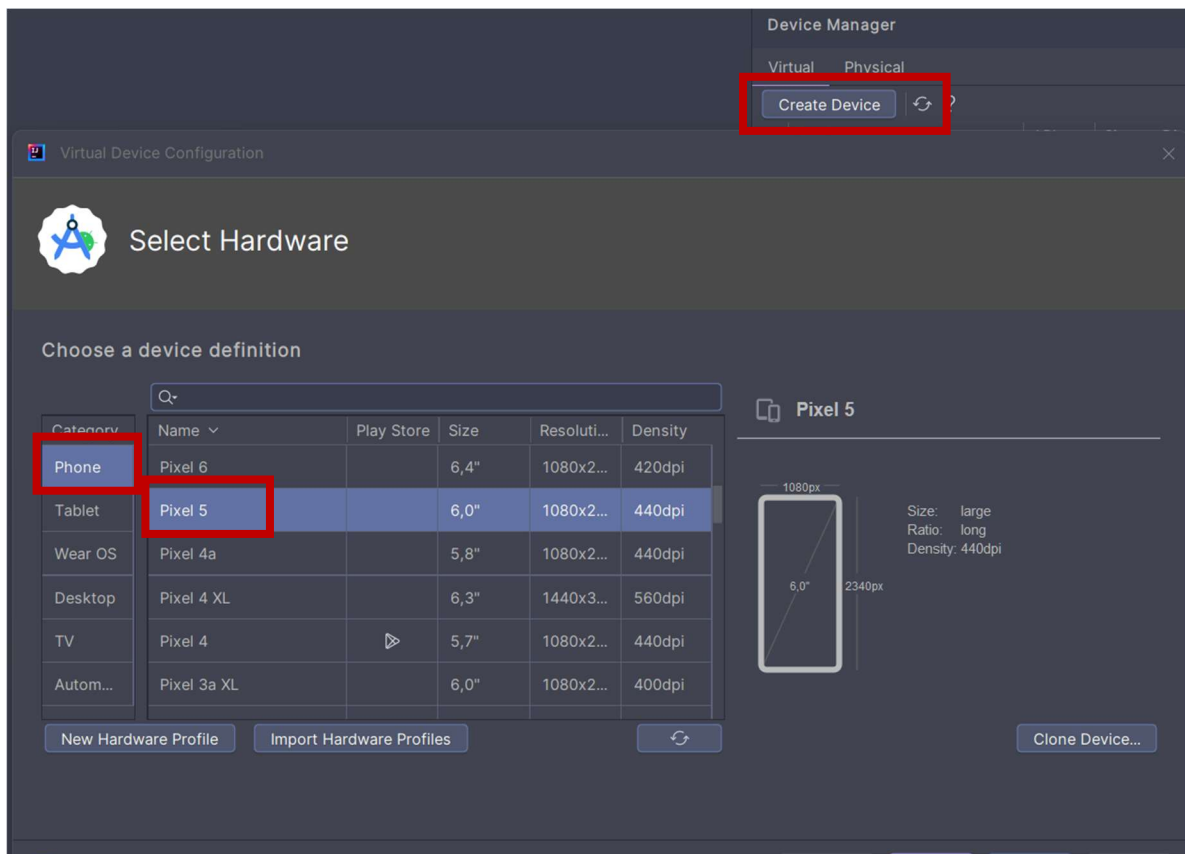
Selecione a opção Configurations > **Android App**



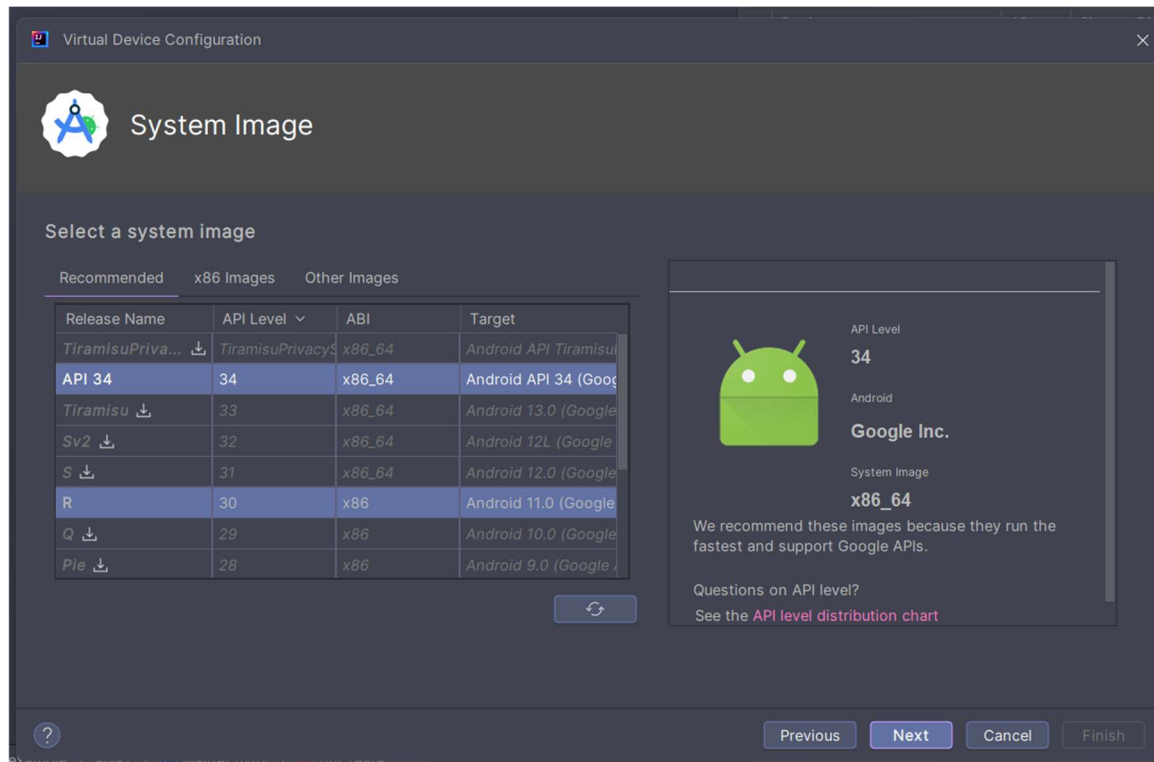
2. Devices



Create Device > Phone > **Pixel 5** > Next



Fazer download de **API 34** > Next



Agora é só rodar o projeto.