



JAVA

Histórico e principais características Android/Java

Cubatão – SP

1º Semestre de 2020

Adalto Luis Alves Pereira Filho

Rayanne Rayssa de Albuquerque Gonçalves

CTII - 418

**Resolução de questões solicitadas
pela docente à disciplina de Projeto
de Sistemas lecionada no Instituto
Federal de Educação, Ciências e
Tecnologia de São Paulo – Campus
Cubatão.**

Orientador: Prof. Mauricio Neves Asenjo

Cubatão – SP

1º Semestre de 2020

O sistema operacional Android surgiu 5 anos antes do primeiro smartphone através da empresa Android Inc, em 2003. O público alvo da empresa era câmeras digitais, porém, resolveram focar seu sistema inovador (Open Source, baseado no Kernel Linux) para um público mais amplo, o mercado mobile.

Com uma interface simples, o objetivo foi gerar um sistema gratuito, funcional e integrado a vários instrumentos que possibilitasse simplicidade para os desenvolvedores.

Com a promessa de um sistema flexível e atualizável, a Google Mobile Division surgiu em 2005 após o Google ter adquirido o Android Inc.

Em 2007 marcas como Samsug, Sony, HTC e Qualcomm fundaram a Open Handset Alliance, com o objetivo de criar uma plataforma de código aberto para smartphones, resultando no primeiro Android comercial lançado em 22 de outubro de 2008.

Todas as versões do Android estão em ordem alfabética e são nomes de doces, com exceção das versões Beta e Alpha.

Disponíveis em smartphones, tablets, netbooks, smartbooks, câmeras digitais. Mp3 player, telefones fixos, Smart tv's entre outros, Atualmente, o sistema operacional mais utilizado é o Android e devido ao fato de ser código aberto, o Android é a base de inúmeras criações e sistemas customizados espalhadas pela internet.

Graças as semelhanças entre a API do Android e a API do Java, o Android pode utilizar Java para desenvolvimento de aplicativos sem gerar complicações para desenvolvedores acostumados a trabalhar com Java.

Tabela 1 - Distribuição dos Androids Existentes

Tabela apresenta as versões do Androids, seus respectivos codinomes, datas de lançamento e a porcentagem de dispositivos que acessaram a Google Play Store em um intervalo de 7 dias encerrado em 22/05/2020.

Versão	Codinome	Data de lançamento	Nível API	Distribuição
11.0	<i>Android 11 (R)</i>	8 de setembro de 2020 ^[15]	30	
10.0	<i>Android 10 (Q)</i>	3 de setembro de 2019	29	
9.0	<i>Pie</i>	6 de agosto de 2018	28	10,4%
8.1	<i>Oreo</i>	5 de dezembro de 2017	27	15,4%
8.0	<i>Oreo</i>	21 de agosto de 2017	26	12,9%
7.1	<i>Nougat</i>	5 de dezembro de 2016	25	7,8%
7.0		22 de agosto de 2016	24	11,4%
6.0	<i>Marshmallow</i>	5 de outubro de 2015	23	16,9
5.1	<i>Lollipop</i>	10 de março de 2015	22	11,5%
5.0-5.0.2		12 de novembro de 2014	21	3,0%
4.4W-4.4W.2	<i>Android Wear (KitKat)</i>	18 de março de 2014	20	Desconhecido
4.4	<i>KitKat</i>	31 de outubro de 2013	19	6,9%
4.3	<i>Jelly Bean</i>	24 de julho de 2013	18	0,5%
4.2.x		13 de novembro de 2012	17	1,5%
4.1.x		9 de julho de 2012	16	1,2%
4.0.3-4.0.4	<i>Ice Cream Sandwich</i>	16 de dezembro de 2011	15	0,3%
4.0.1-4.0.2		19 de outubro de 2011	14	<0,1%
3.2-3.2.6	<i>Honeycomb</i>	15 de julho de 2011	13	<0,1%
3.1		10 de maio de 2011	12	<0,1%
3.0		22 de fevereiro de 2011	11	<0,1%
2.3.3-2.3.7	<i>Gingerbread</i>	9 de fevereiro de 2011	10	0,3%
2.3		06 de dezembro de 2010	9	<0,1%
2.2-2.2.3	<i>Froyo</i>	20 de maio de 2010	8	<0,1%
2.1	<i>Eclair</i>	11 de janeiro de 2010	7	<0,1%
2.0.1		03 de dezembro de 2009	6	<0,1%
2.0		26 de outubro de 2009	5	<0,1%
1.6	<i>Donut</i>	15 de setembro de 2009	4	<0,1%
1.5	<i>Cupcake</i>	30 de abril de 2009	3	<0,1%
1.1	<i>Beta*</i>	9 de fevereiro de 2009	2	<0,1%
1.0	<i>Alpha*</i>	23 de setembro de 2008	1	<0,1%

Fonte: Histórico das versões do Android disponível em Wikipédia.



JAVA

Principais IDEs para desenvolvimento JAVA

Cubatão – SP

1º Semestre de 2020

Adalto Luis Alves Pereira Filho

Rayanne Rayssa de Albuquerque Gonçalves

CTII - 418

Pesquisa solicitada pela docente à disciplina de Projeto de Sistemas lecionada no Instituto Federal de Educação, Ciências e Tecnologia de São Paulo – Campus Cubatão.

Orientador: Prof. Mauricio Neves Asenjo

Cubatão – SP

1º Semestre de 2020

1. IDEs mais populares de serem utilizadas no JAVA

O Java possui inumeras IDEs. Podemos citar:

1.1. Eclipse



1.2. NetBeans



1.3. IntelliJ IDEA



1.4. BlueJ



1.5. JCreator



2. IDE escolhida

Os fatores que levaram o grupo Java escolher a IDE NetBeans: gratuidade e familiaridade com a IDE.

3. Processo de dowlond e instalação da IDE escolhida

Acessando o site da NetBeans (<https://netbeans.org/>) é possível encontrar o botão "Download":



A página com as últimas versões da NetBeans e seus respectivos downloads é aberta a partir do botão anterior:

Apache NetBeans Releases

Apache NetBeans is released four times a year. For details, see [full release schedule](#).



Our annual May/June release is a long-term support (LTS) release that benefits from our [NetCAT community testing process](#), and remains available and supported for a year. Our other quarterly releases provide early access to new features, which are tested and consolidated in the subsequent LTS release.

Apache NetBeans 12 feature update 1 (NB 12.1)

Latest version of the IDE, released on September 1, 2020.

Features

Download

O grupo optou pela versão do dia 1 de setembro de 2020.

Ao clicar no botão “Download” somos redirecionados para uma página que contém as versões do NetBeans 12.1 disponíveis para os sistemas operacionais: Windows; Linux e Mac.

O grupo optou em instalar a versão para Windows:

Downloading Apache NetBeans 12.1

Apache NetBeans 12.1 was released on September 1, 2020. See [Apache NetBeans 12.1 Features](#) for a full list of features.

Apache NetBeans 12.1 is available for download from your closest Apache mirror.

- Binaries: [netbeans-12.1-bin.zip](#) (SHA-512, PGP ASC)
- Installers:
 - [Apache-NetBeans-12.1-bin-windows-x64.exe](#) (SHA-512, PGP ASC)
 - [Apache-NetBeans-12.1-bin-linux-x64.sh](#) (SHA-512, PGP ASC)
 - [Apache-NetBeans-12.1-bin-macosx.dmg](#) (SHA-512, PGP ASC)
- Source: [netbeans-12.1-source.zip](#) (SHA-512, PGP ASC)
- Javadoc for this release is available at <https://bits.netbeans.org/12.1/javadoc>

Officially, it is important that you **verify the integrity** of the downloaded files using the PGP signatures (.asc file) or a hash (.sha512 files). The PGP keys used to sign this release are available [here](#).

Apache NetBeans can also be installed as a self-contained [snap package](#) on Linux.

[Deployment platforms](#)

[Building from source](#)

[Community approval](#)

[Earlier releases](#)

Ao clicar no link da versão do Windows 64, somos redirecionados para sugestões de links para o download. Optamos pelo segundo link:



COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

[Projects](#) ▾

[People](#) ▾

[Community](#) ▾

[License](#) ▾

[Sponsors](#) ▾

We suggest the following mirror site for your download:

<http://mirror.nbrtelecom.com.br/apache/netbeans/netbeans/12.1/Apache-NetBeans-12.1-bin-windows-x64.exe>

Other mirror sites are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (.asc file) or a hash (.md5 or .sha* file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

HTTP

<http://ftp.unicamp.br/pub/apache/netbeans/netbeans/12.1/Apache-NetBeans-12.1-bin-windows-x64.exe>

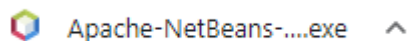
<http://mirror.nbrtelecom.com.br/apache/netbeans/netbeans/12.1/Apache-NetBeans-12.1-bin-windows-x64.exe>

BACKUP SITES

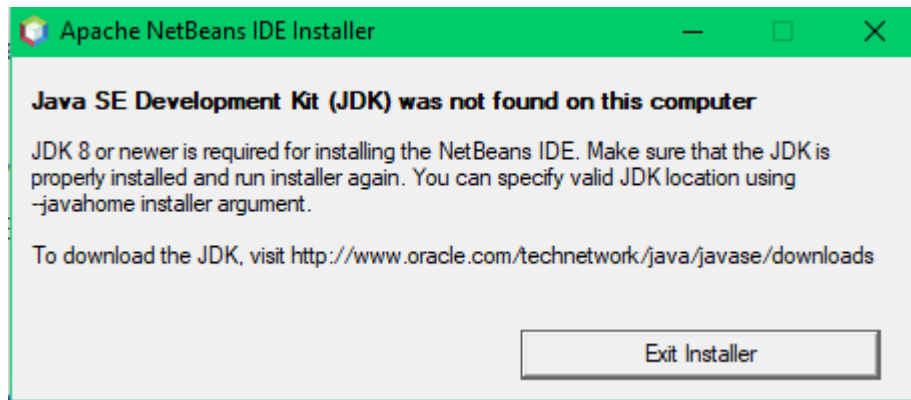
Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

<https://downloads.apache.org/netbeans/netbeans/12.1/Apache-NetBeans-12.1-bin-windows-x64.exe>

O link gera o download de um arquivo executável de 375MB que baixa através do navegador:



Ao abrir o arquivo tivemos o seguinte erro:



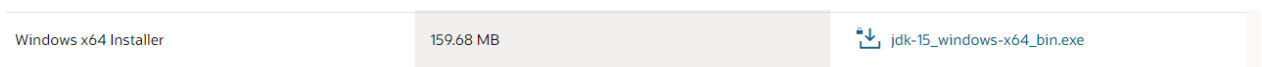
Para resolver, temos que entrar no link (<https://www.oracle.com/java/technologies/javase-downloads.html>) e efetuar a instalação do **Java SE Downloads**.

Escolhemos a 1º opção:

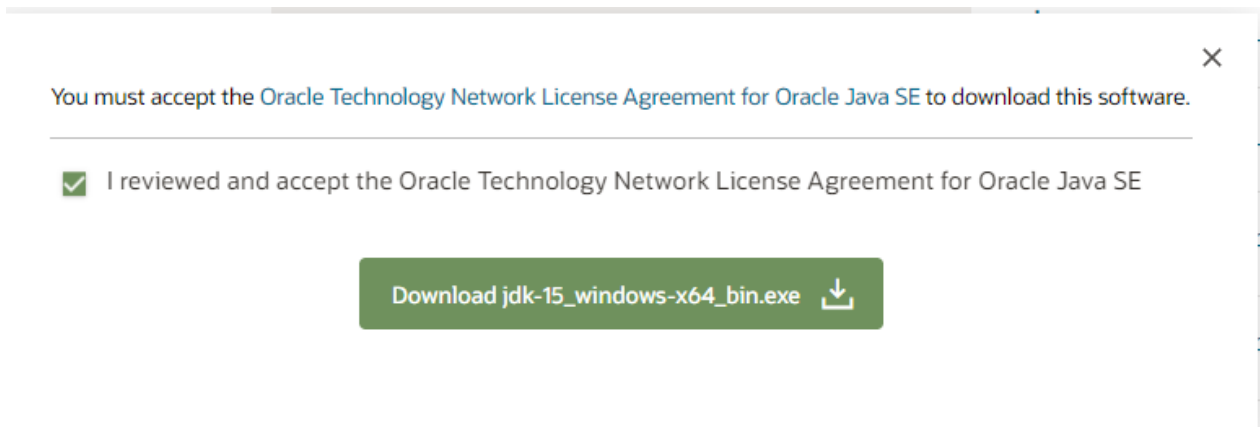


A primeira opção redireciona para uma página que apresenta diversos instaladores que deverão ser escolhidos conforme o sistema operacional da máquina que será utilizada.

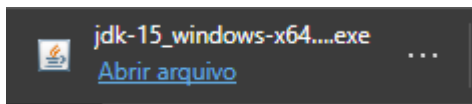
Escolhemos a versão para Windows:



É necessário aceitar os termos de uso e clicar no botão "Download jdk_windows-x64_bin.exe" :



Um arquivo de 160MB começará a baixar no navegador. Quando terminar de baixar ficará assim:

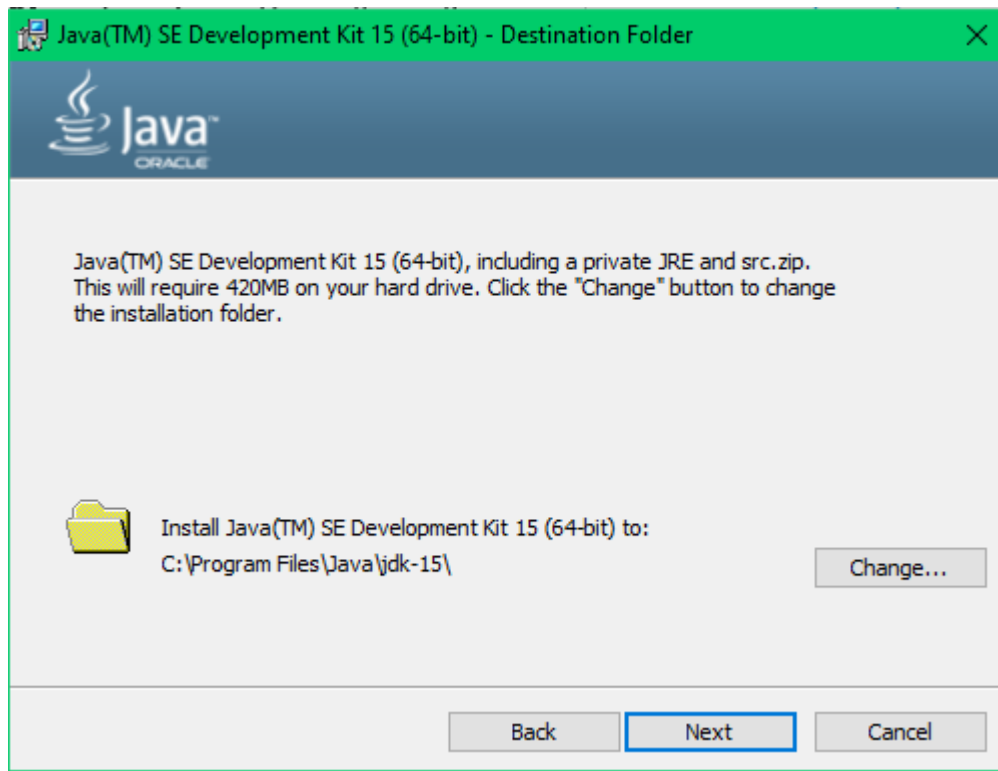


Abra o arquivo e quando carregar essa tela:



Clique em "Next >".

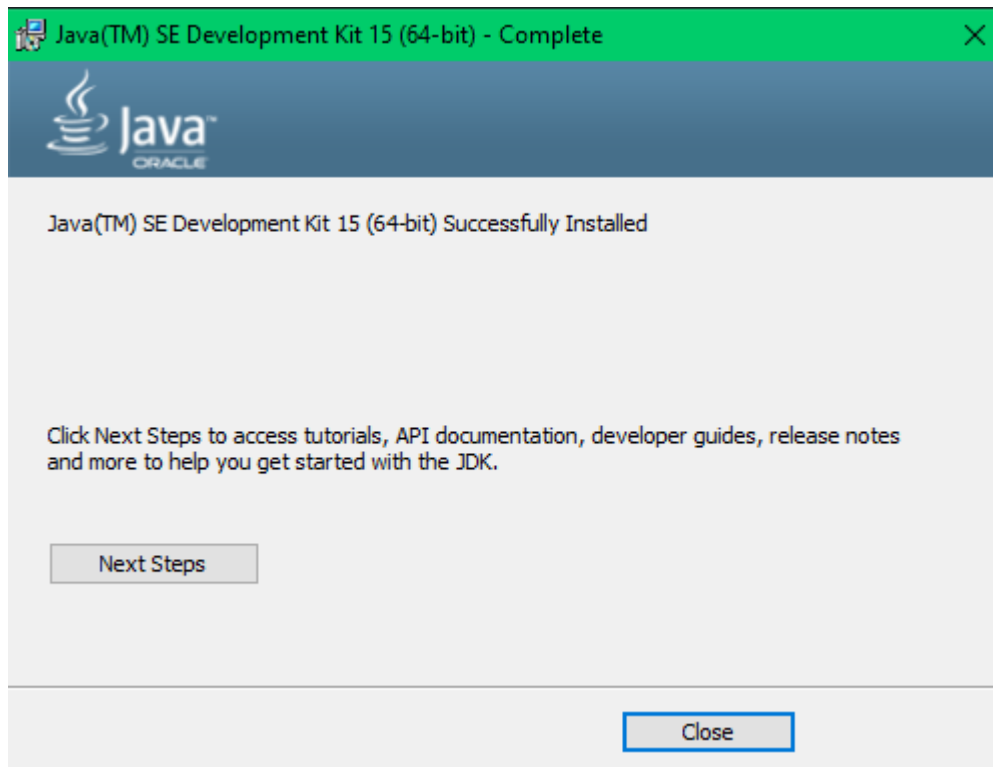
Depois clique novamente em "Next >"



Espere baixar.

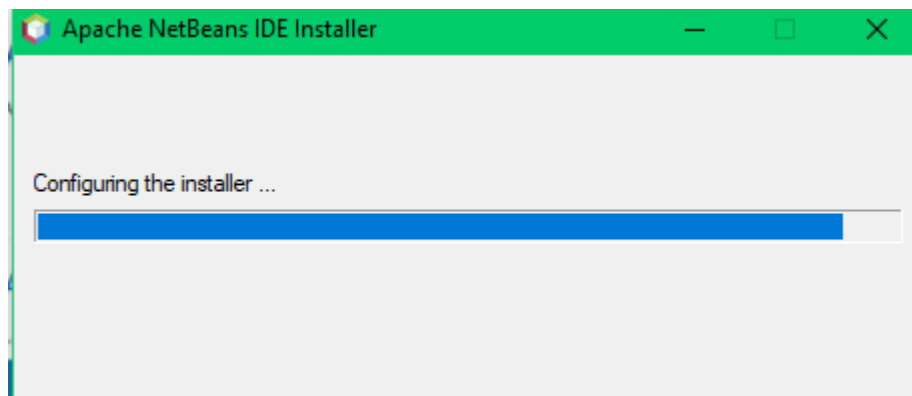


Ao terminar de carregar, teremos esta tela:

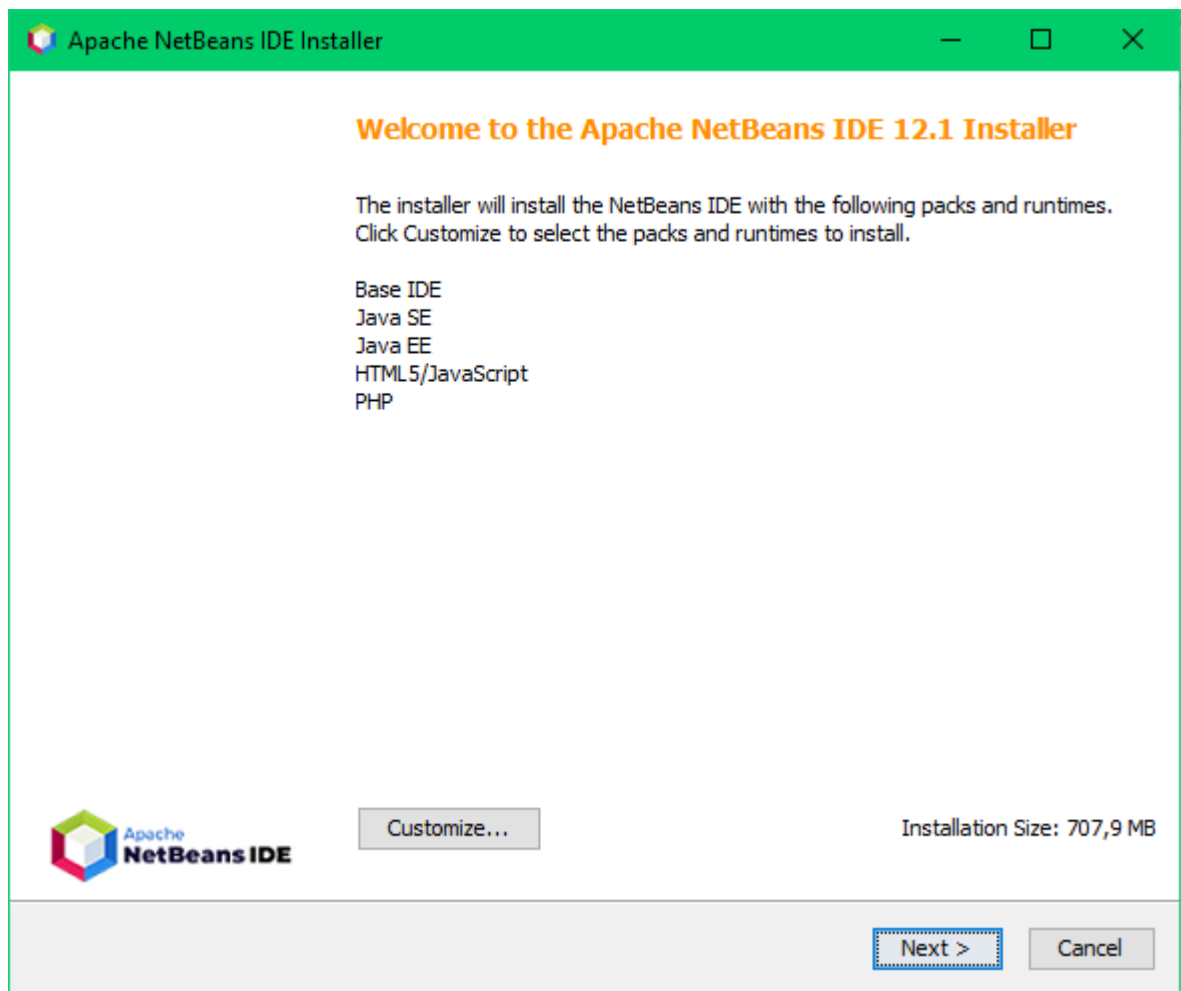


Clique em Close.

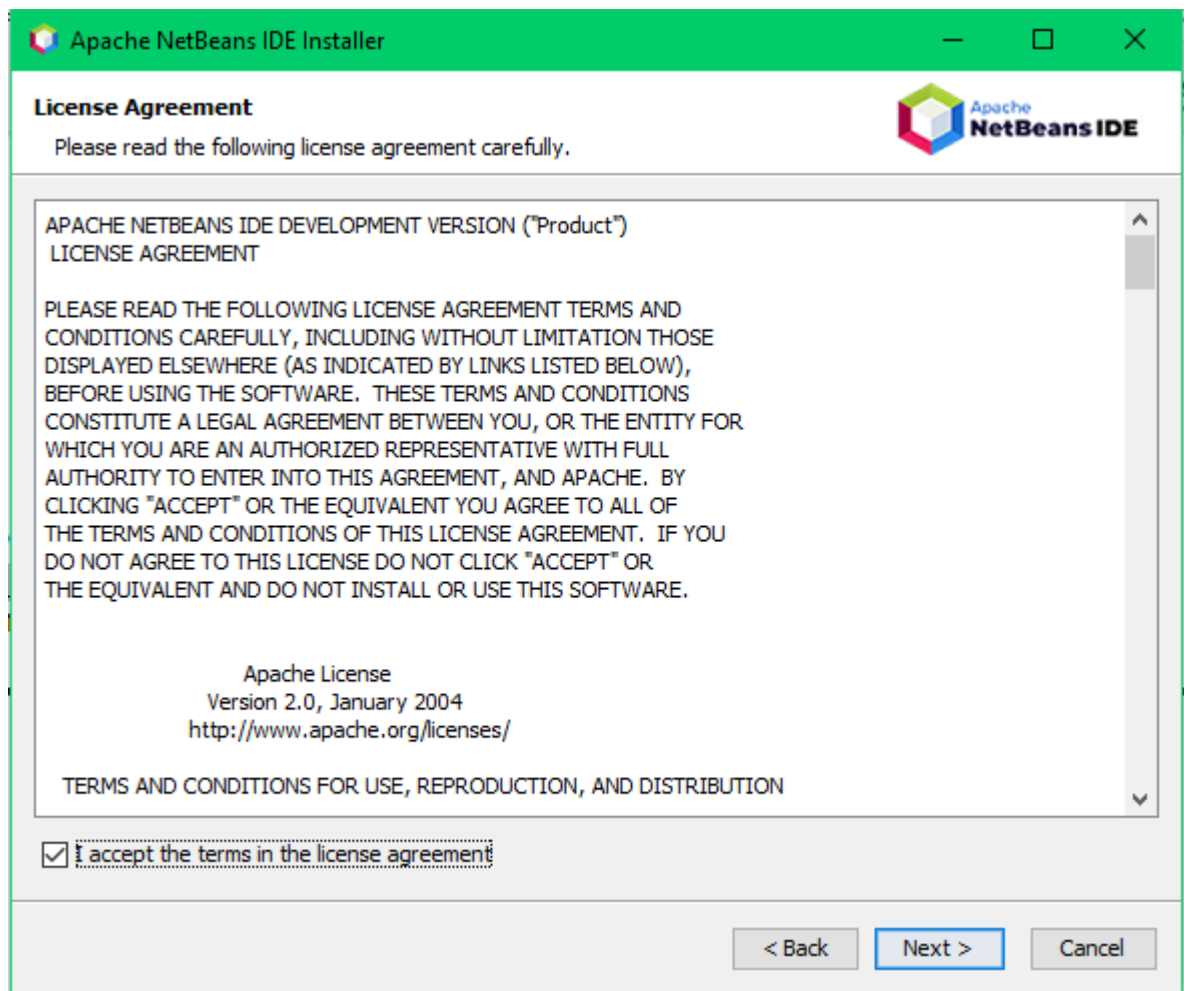
Aparecerá uma tela configurando a instalação. Aguarde.



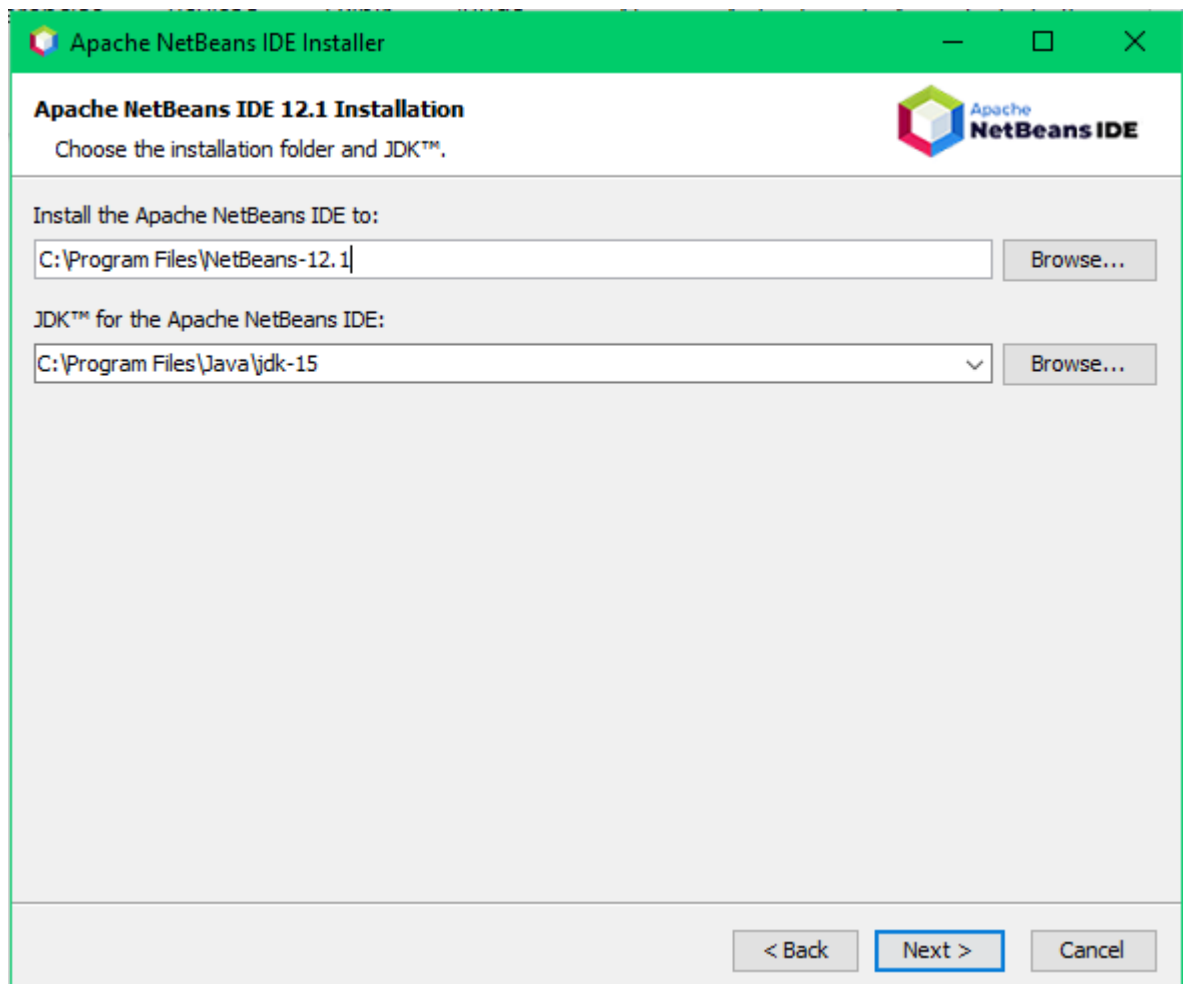
Depois de aguardar a configuração da instalação, aparecerá uma tela que pede para customizar os pacotes que serão instalados. O grupo optou em instalar todos que estão sendo sugeridos. Podemos customizar e/ou clicar no botão "Next >":



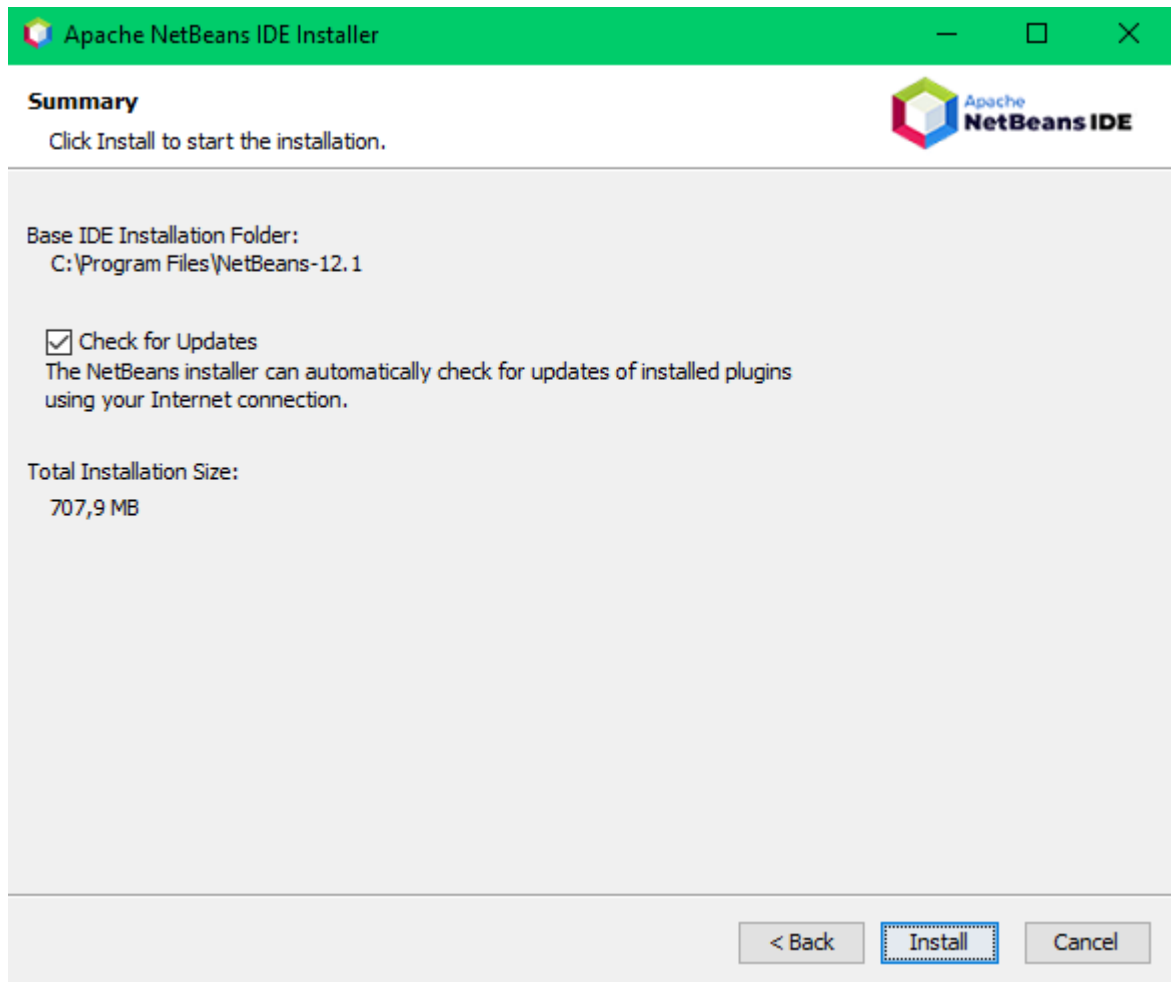
Para prosseguir é necessário aceitar a licença. Marque a caixa “I accept the terms in the license agreement” e clique no botão “Next >”:



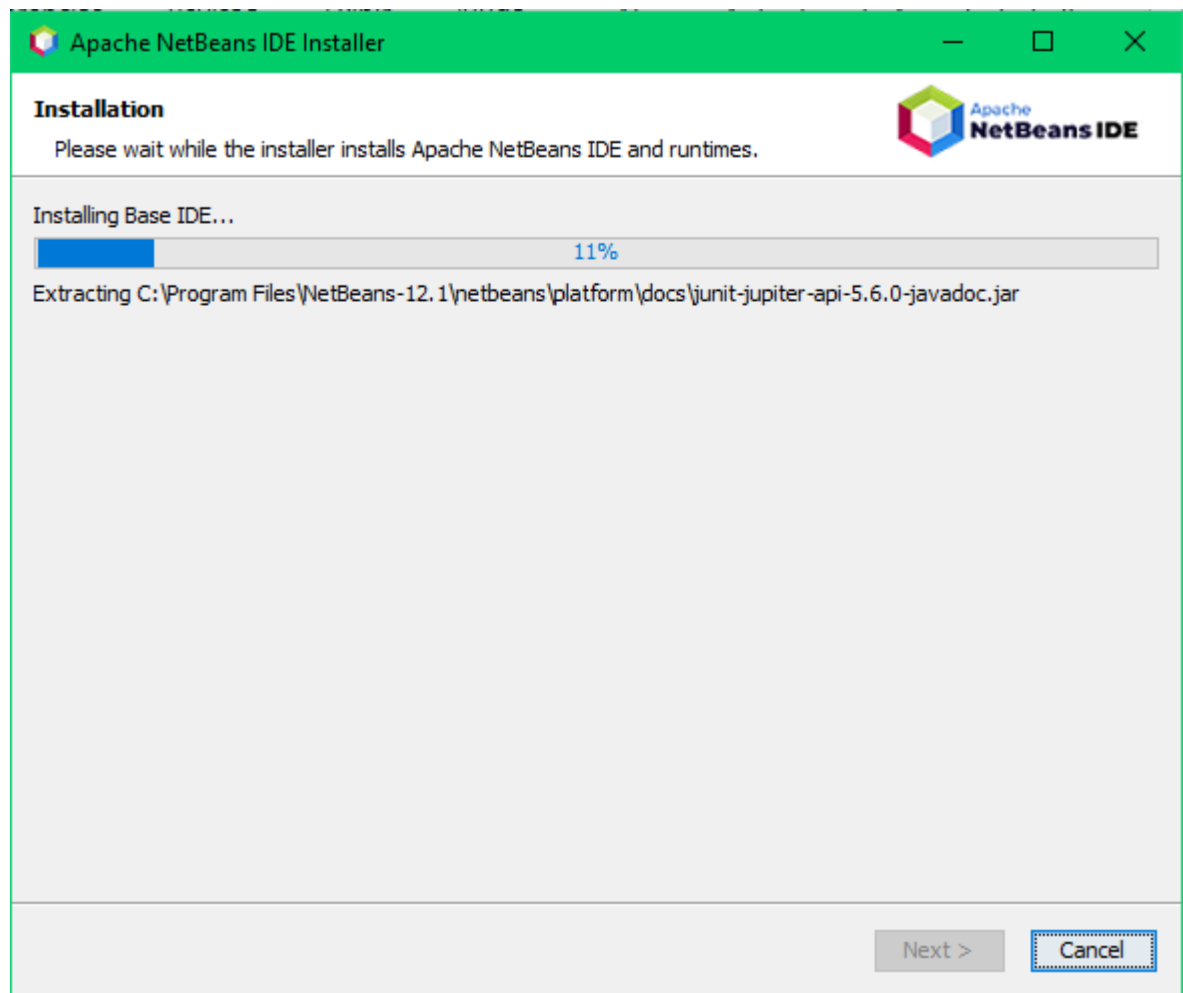
Na próxima etapa precisamos definir onde ocorrerá a instalação. Optamos em seguir a sugestão e clicar em "Next >".



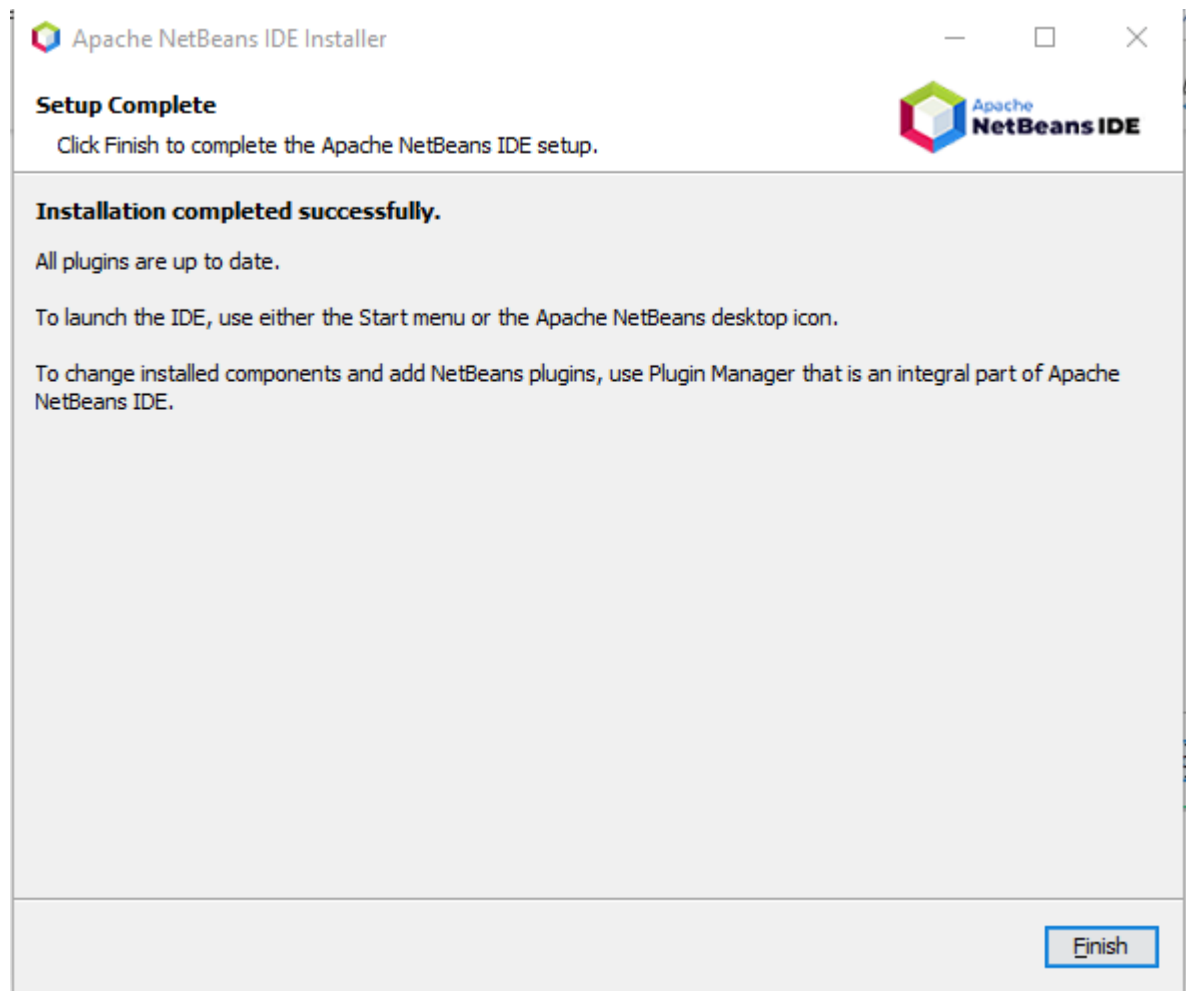
Nesta proxima tela apenas clicamos no botão "Install":



>” : É necessário aguardar a instalação. Depois de concluída, clique no botão “Next

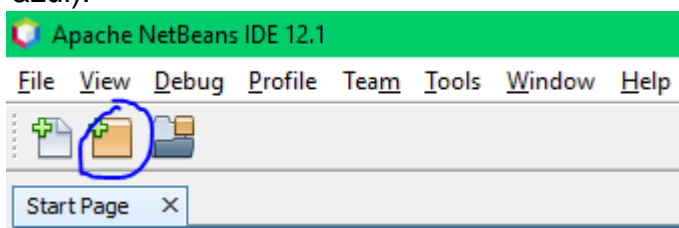


Ao terminar a instalação é só clicar no botão “Finish”:

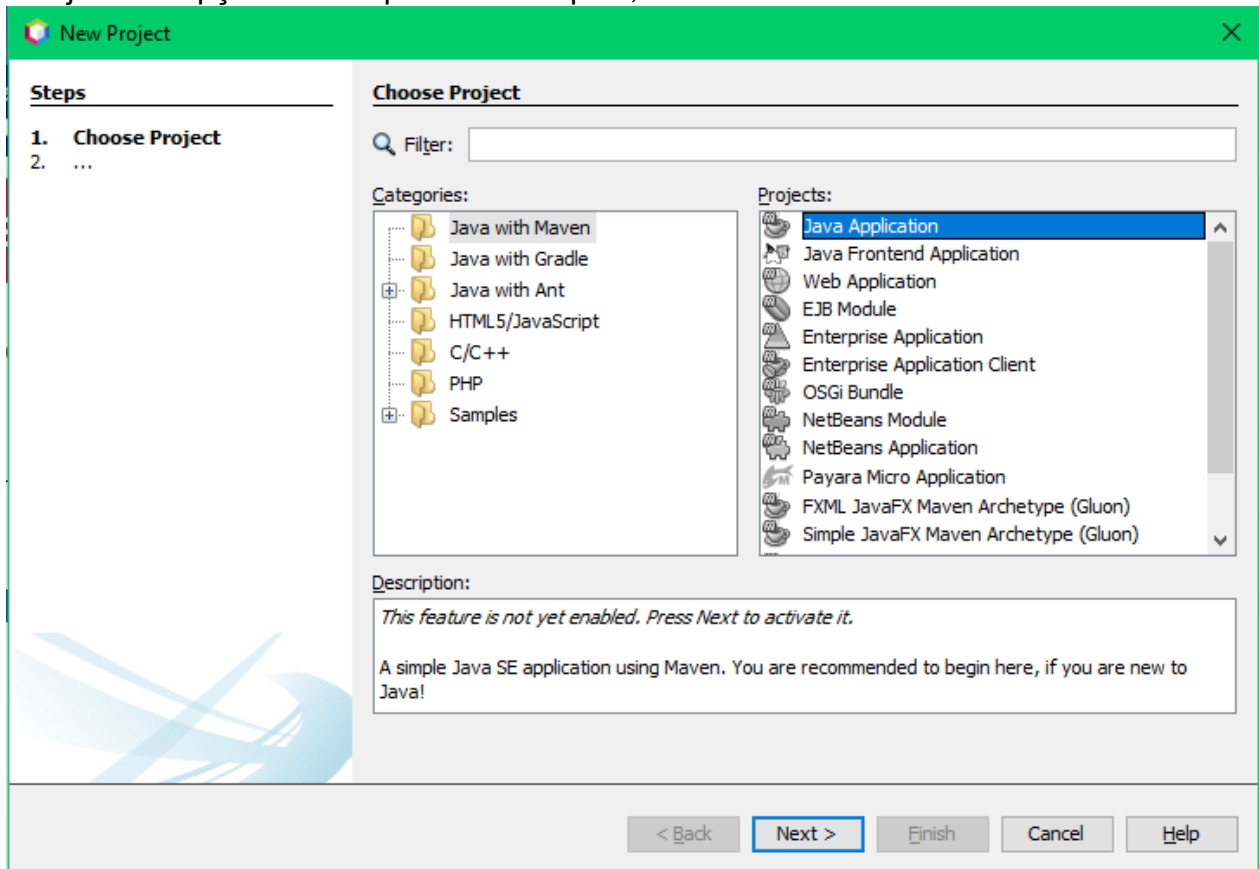


4. Hello Word

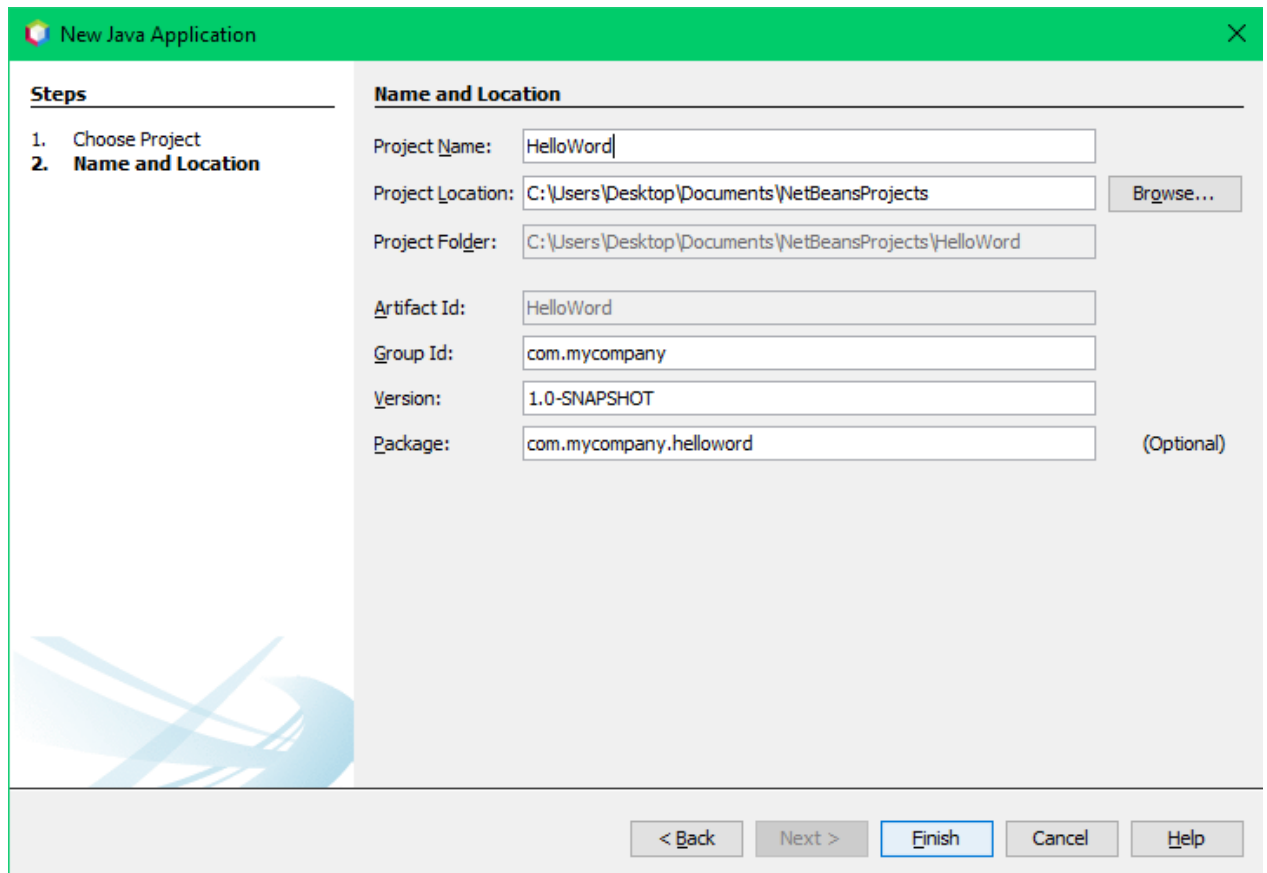
Para criar um novo projeto deveremos clicar no botão "New Project" (indicado no circulo azul):



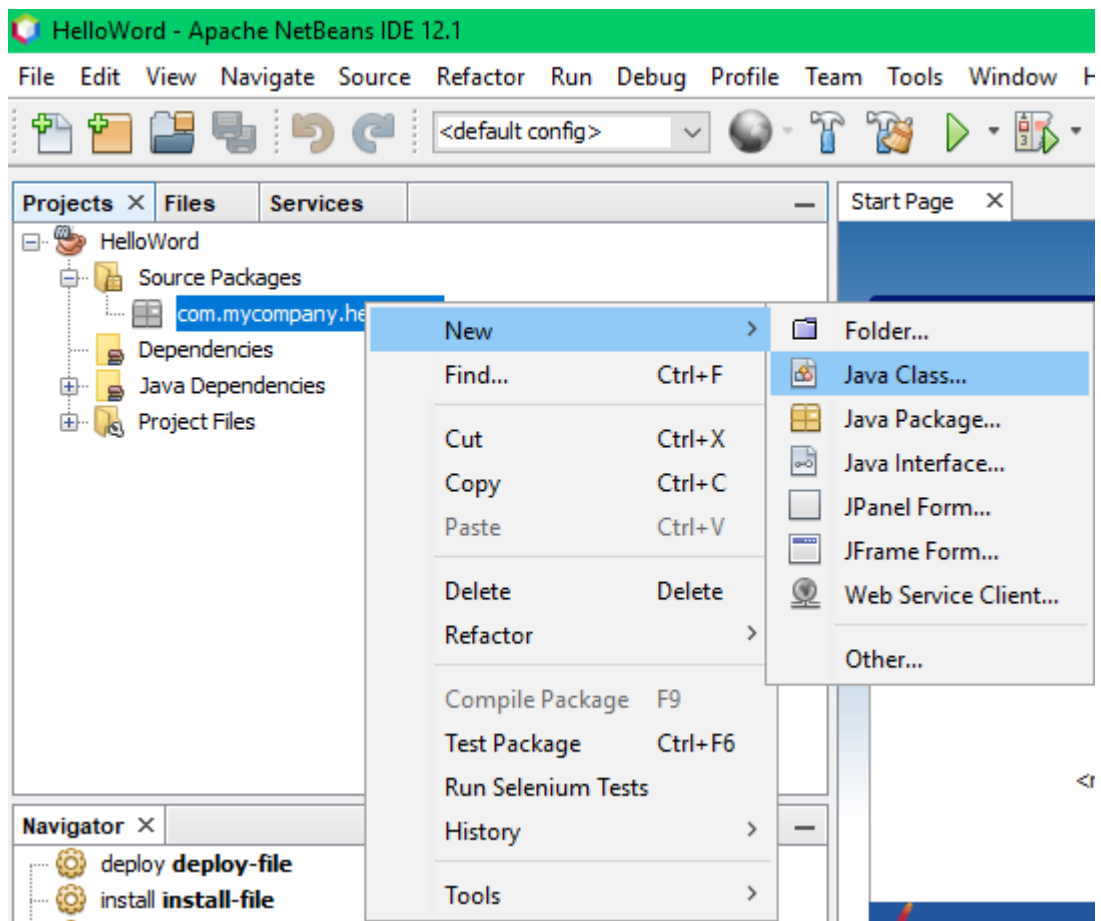
Agora deveremos selecionar na lista “Categories” a opção Java e na lista “Projects” a opção “Java Application”. Depois, deveremos clicar no botão “Next>”:



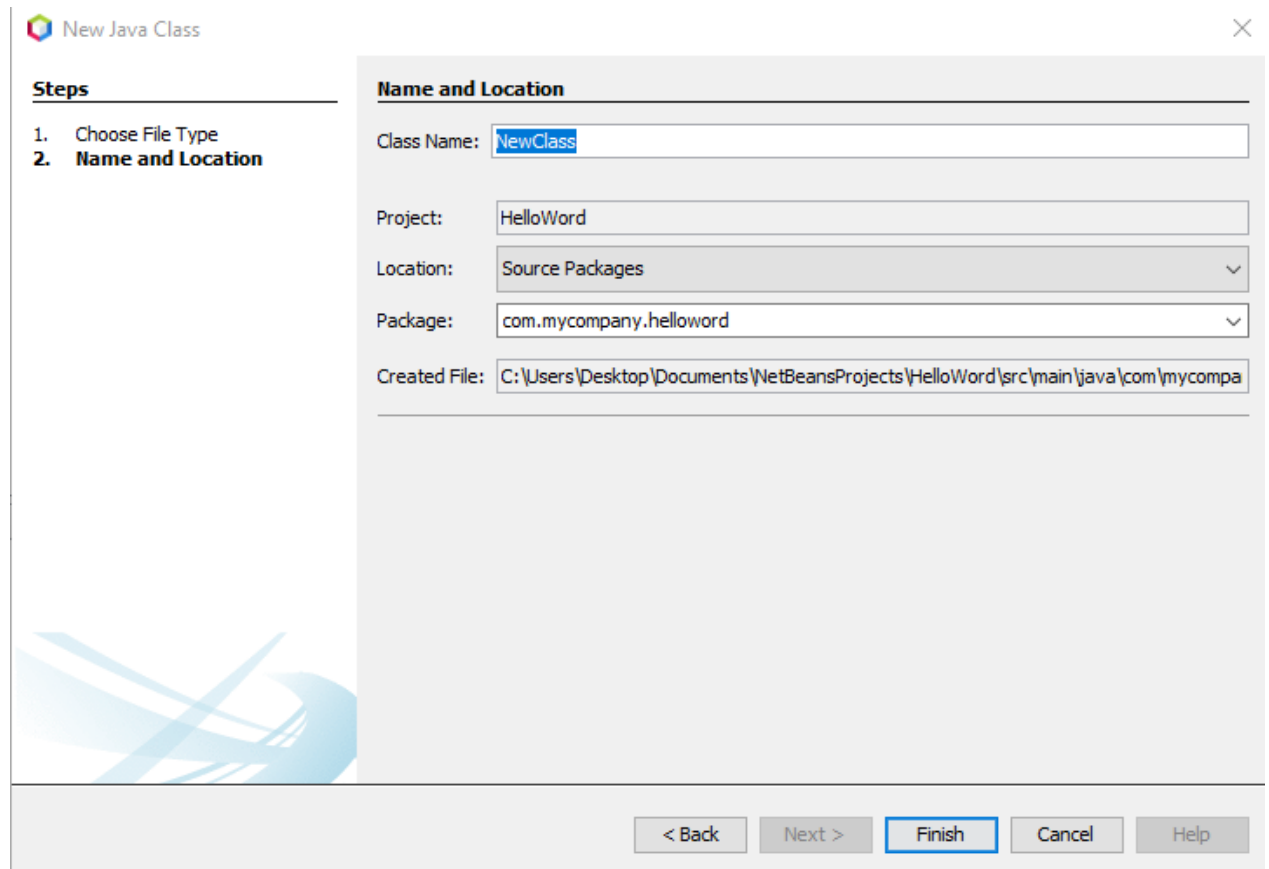
Nesta etapa podemos alterar o nome e local do projeto. No campo “Project Name” colocamos HelloWorld e optamos na permanência do local sugerido. Após decidir/acatar o nome do projeto e sua localização, deveremos clicar no botão “Finish”:



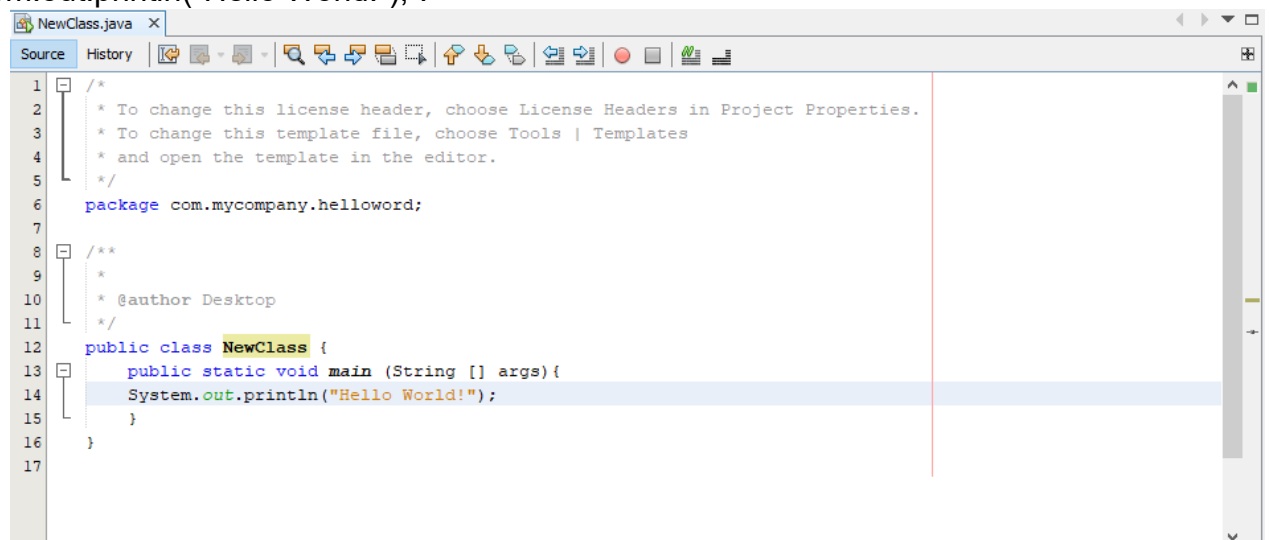
É necessário dar um duplo clique no projeto HelloWord (que tem como símbolo uma xícara de café). Em sequência devemos dar mais um duplo clique, desta vez no pacote “Source Packages”. Clique com o botão direito no pacote que apareceu, neste caso se chama “com.mycompany.helloworld”, selecione “New” e pois clique em “Java Class”:



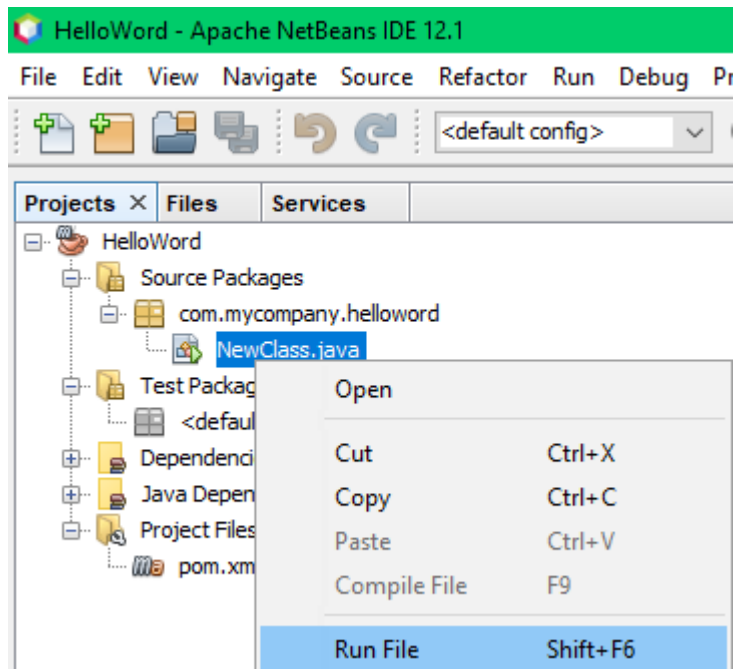
Para criar uma nova classe deveremos apenas escolher um nome através do campo "Class Name" e clicar no botão "Finish":



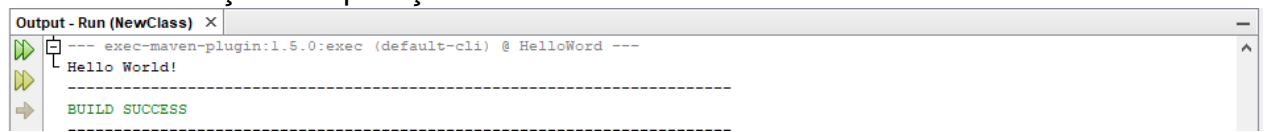
No código fonte, na parte “public class NewClass”, lembrando que “NewClass” é o nome que demos para nossa classe (public class mudará conforme o nome que a classe receber), deveremos criar nosso método principal o “public static void main (String [] args)”. Entre chaves, deveremos escrever o comando Java “System.out.println(“Hello World!”);”:



Para executar a aplicação deveremos clicar com o botão direito na classe “NewClass.java” e depois selecionar a opção “Run file”:



Esta é a execução da aplicação:





INSTITUTO FEDERAL

São Paulo

Câmpus Cubatão

JAVA

Primeira Aplicação

Cubatão – SP
2º Semestre de 2020

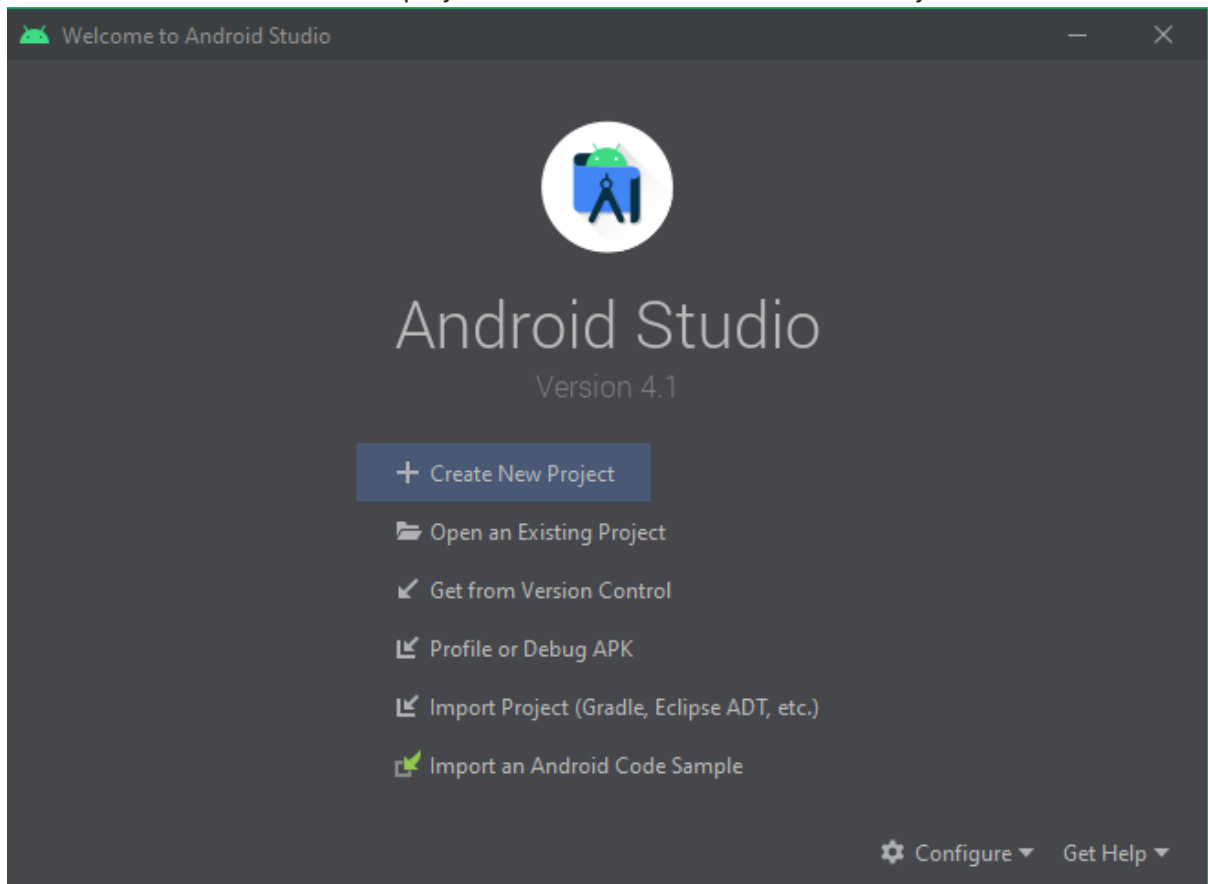
Adalto Luis Alves Pereira Filho
Rayanne Rayssa de Albuquerque Gonçalves
CTII - 418

**Aplicação solicitada pelo
docente à disciplina de Projeto de
Sistemas lecionada no Instituto
Federal de Educação, Ciências e
Tecnologia de São Paulo – Campus
Cubatão.**

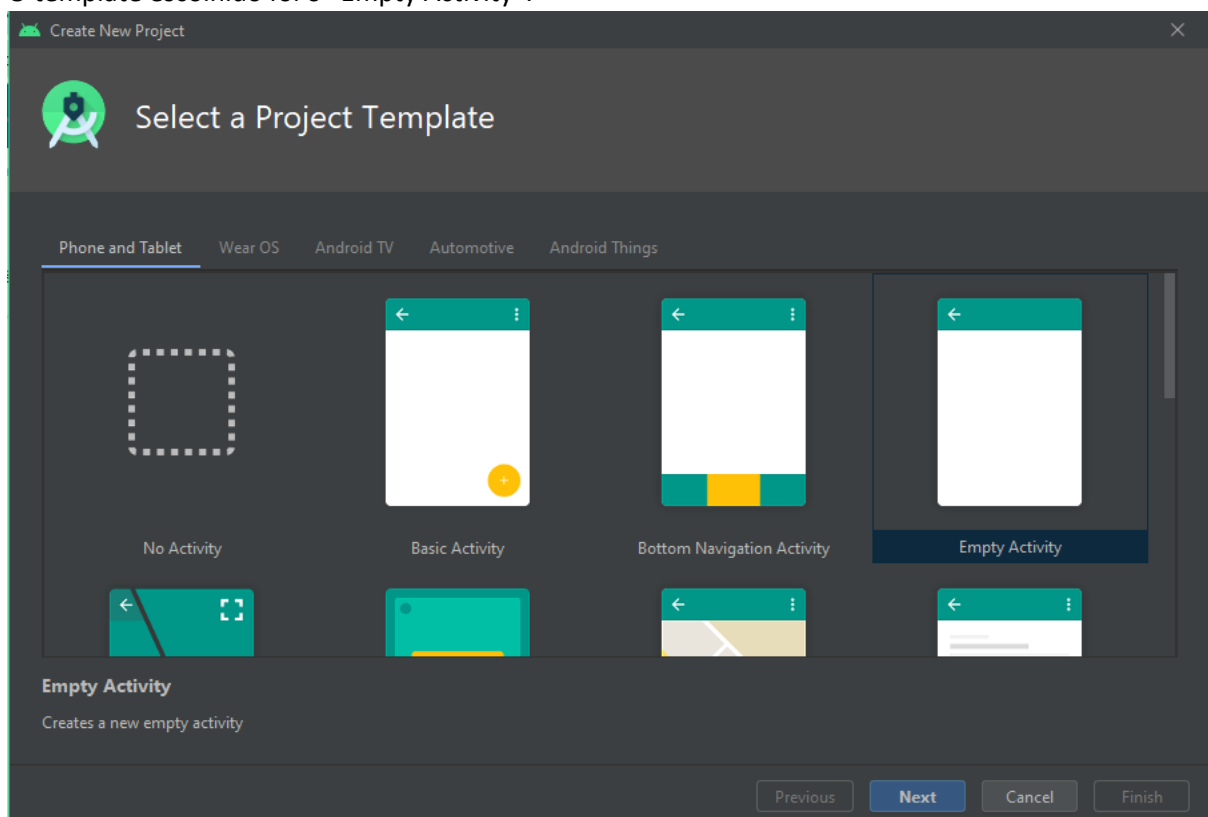
Orientador: Prof. Mauricio Neves Asenjo

Cubatão – SP
2° Semestre de 2020

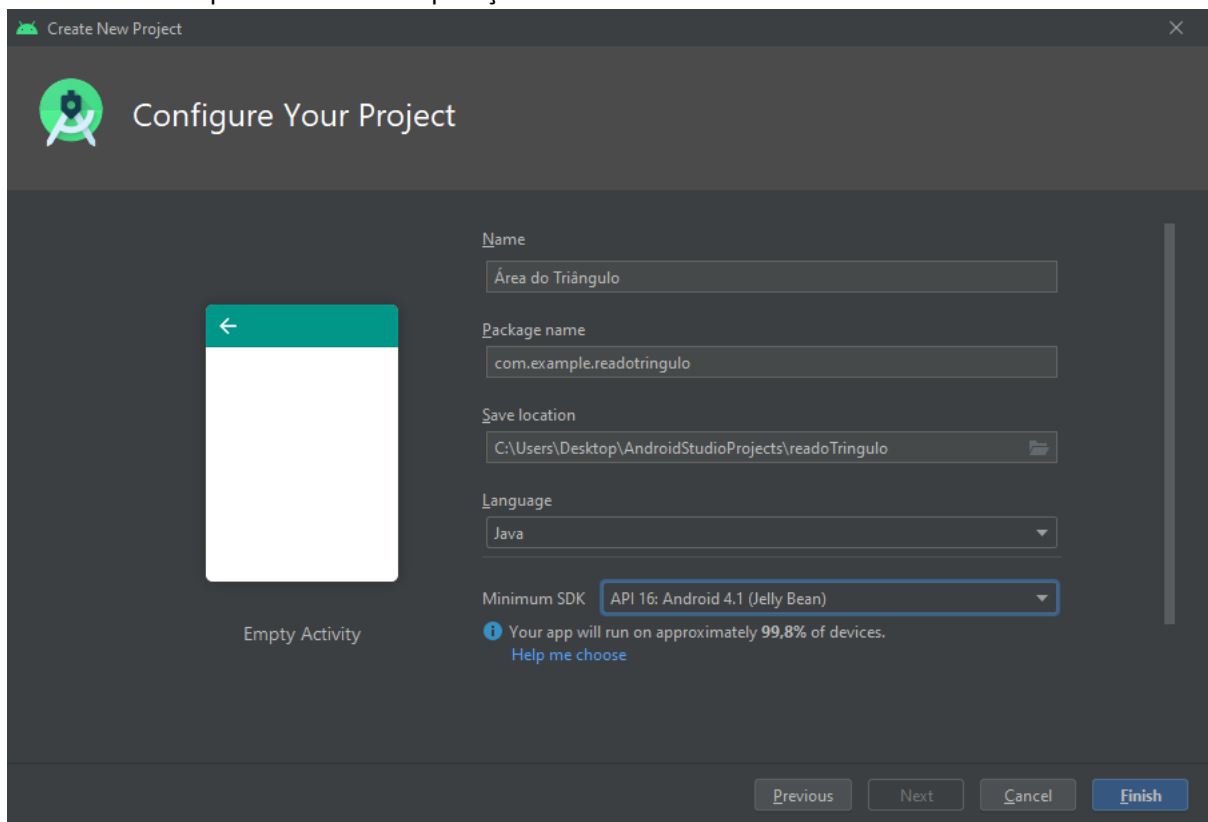
Primeiro devemos criar um novo projeto clicando no botão “Create New Project”:



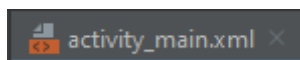
O template escolhido foi o “Empty Activity”:



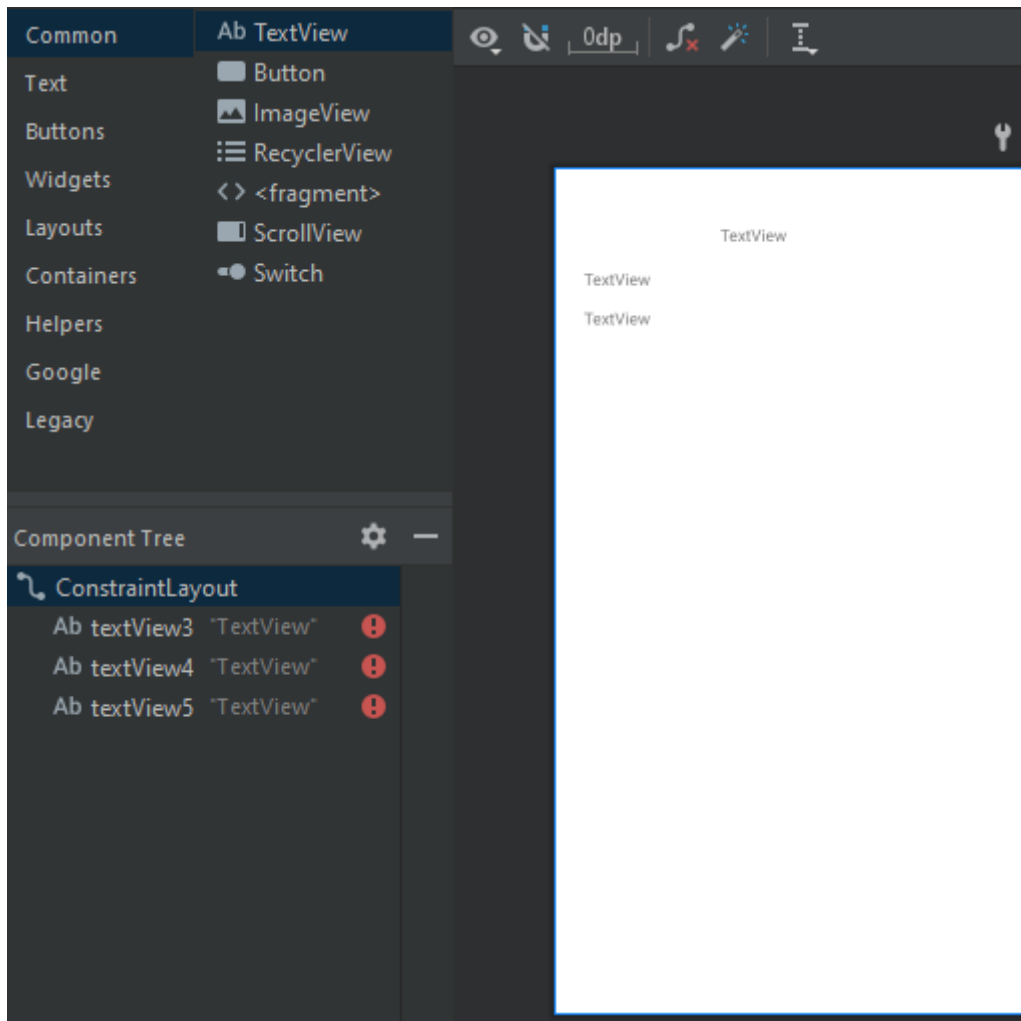
Nessa parte nomeamos o projeto para "Área do Triângulo", mas daria para nomear o pacote, selecionar o local onde será salvo, escolher a linguagem (que no caso do grupo é Java) e escolhemos a versão mínima para rodar nossa aplicação:



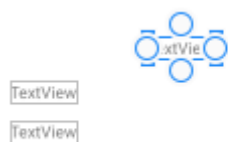
Na aba com a extensão "xml" iremos alterar a parte visual do projeto:



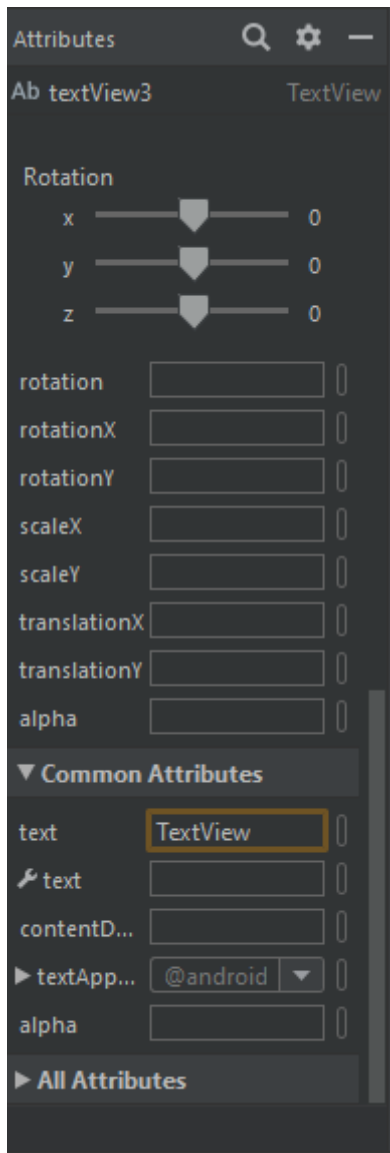
Para colocar texto na parte visual do projeto podemos arrastar a opção "TextView" para a parte branca:



Para alterar os textos das “textView”s é necessário clicar em cima de uma delas:



Ir na parte em “Attributes”, descer até a opção “Common Attributes” e Trocar os textos que estão nas caixas “text”:



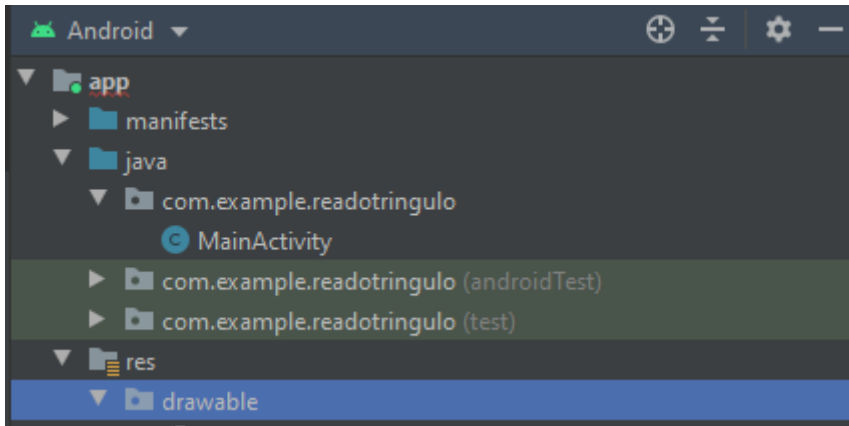
Depois de alterar os 3 ficará assim:

Calcular área do triângulo

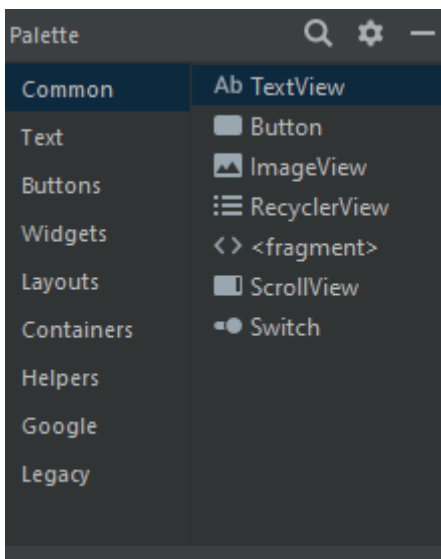
Altura:

Base:

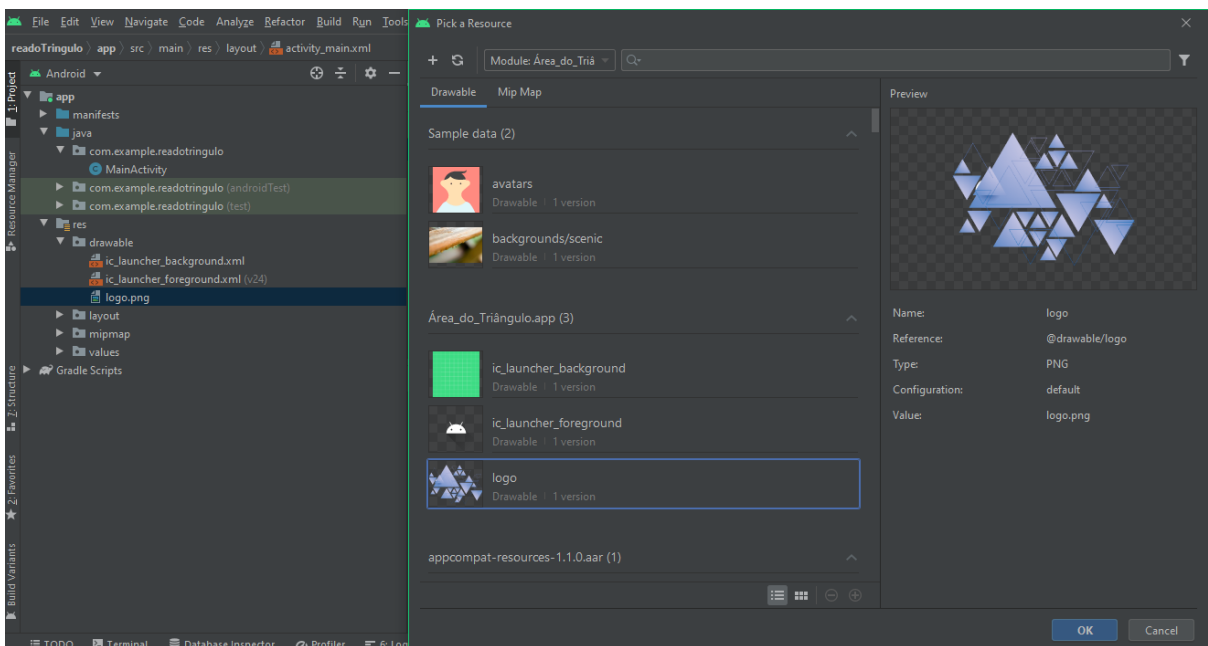
Para colocar um plano de fundo é necessário ter a imagem baixada no computador. Devemos copiar o arquivo da imagem e colar no “drawable”:



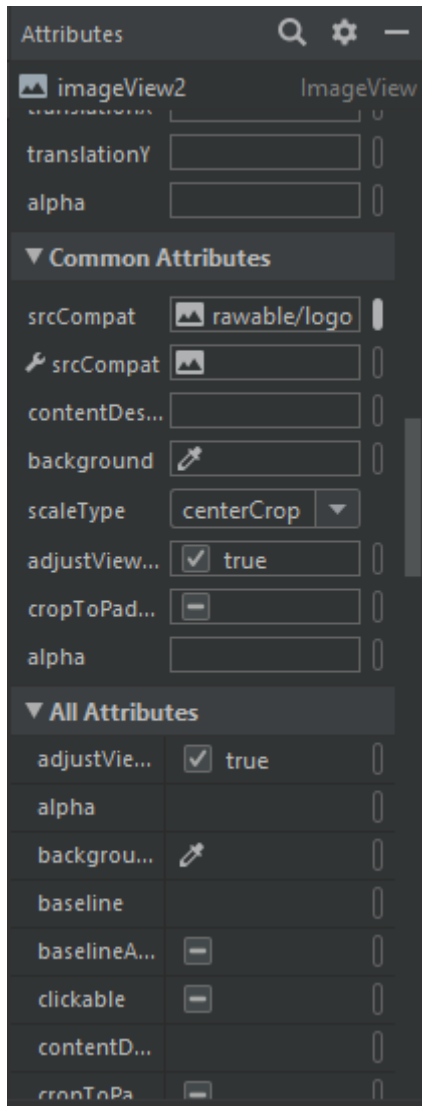
Depois vamos arrastar o “ImageView” para nosso layout:



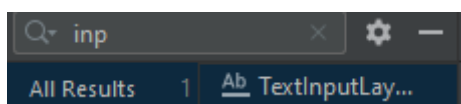
Após aparecer esta tela, clique em “OK”:



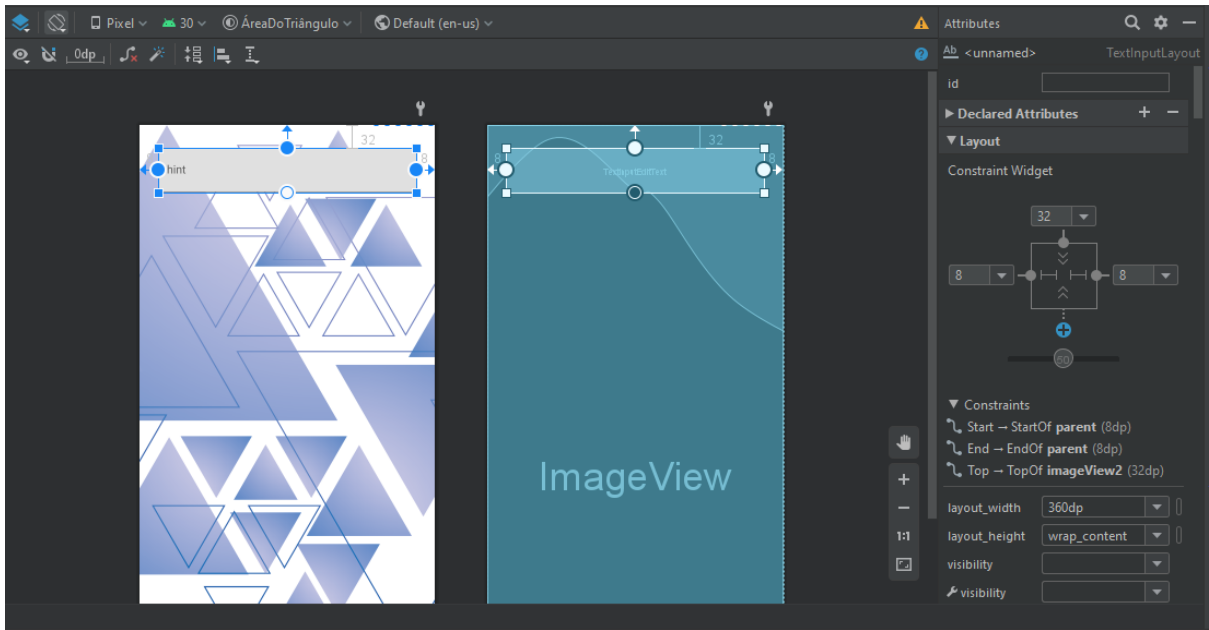
Para dimensionar o plano de fundo utilizamos os recursos da parte “Common Attributes”:
“scaleType” na opção “centerCrop” e “adjustViewBounds” na opção “true”. Na parte “All Attributes”
colocamos o recurso “adjustViewBounds” na opção “true”:



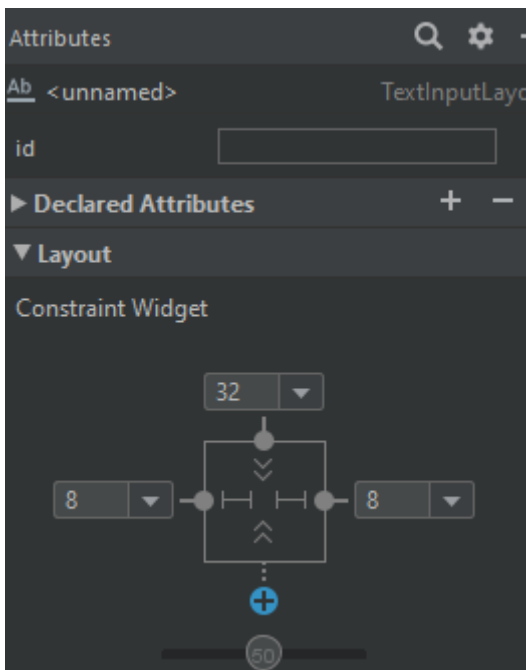
Ao pesquisar “inp” temos a opção “TextInputLayout”. Devemos arrastar para o nosso layout:



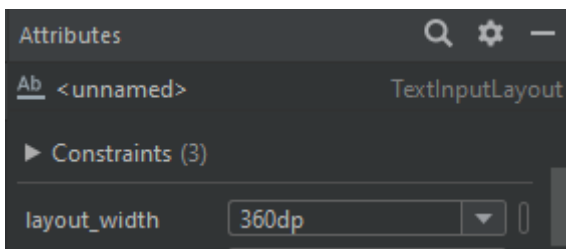
Agora é necessário arrastar a bolinha da esquerda para a margem esquerda, a bolinha da margem direita para a direita e a bolinha de cima para cima. Depois vamos na parte de “Attributes”.



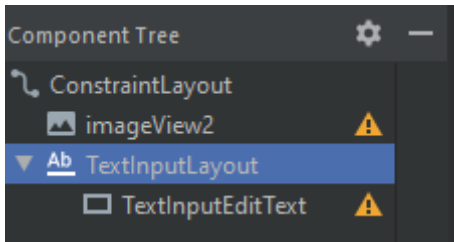
Na parte de "Attributes" em "Layout" alteraremos para esses valores:



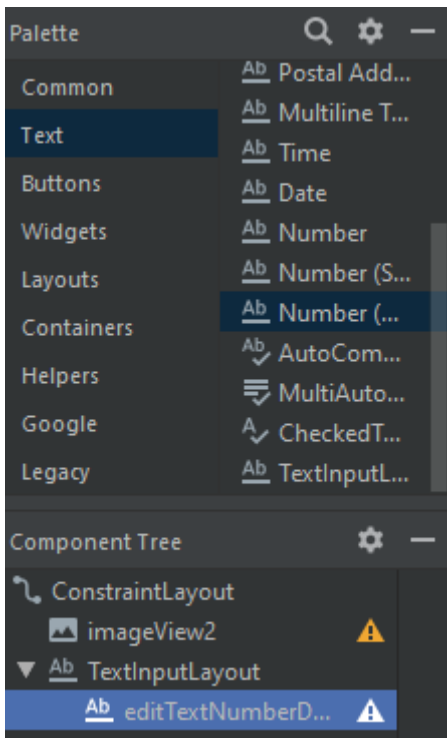
Ainda na parte de "Attributes" porém em "Constraints", alteramos o valor de "layout_width" para 360dp:



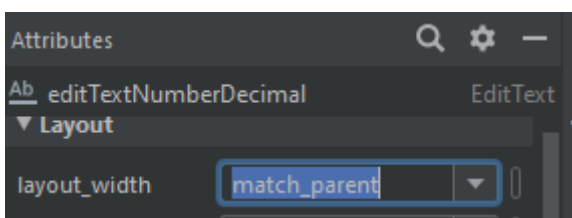
Deletamos o padrão de texto "TextInputEditText":



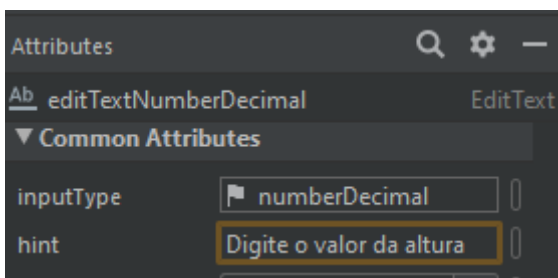
Arrastamos a opção “Number (decimal)” para o “TextInputLayout”:



Para preencher todo o layout de texto é necessário selecionar a opção “match_parent” no “Layout”:

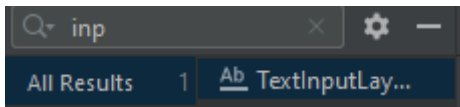


Colocamos o texto “Digite o valor da altura” na parte “hint” de “Common Attributes” do Attributes para guiar o que o usuário deverá digitar:

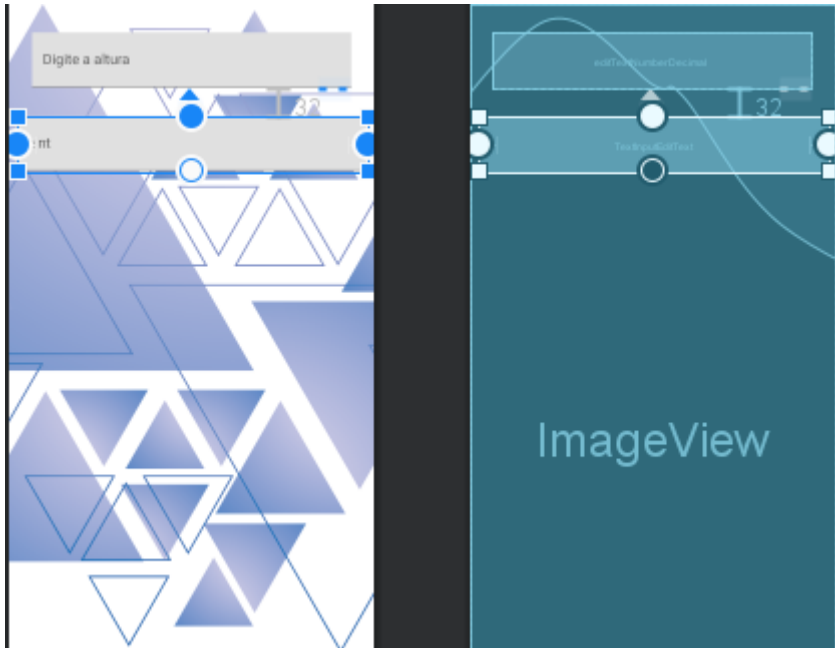


Agora repetiremos o processo para criar um campo onde o usuário deverá digitar o valor da base do triângulo.

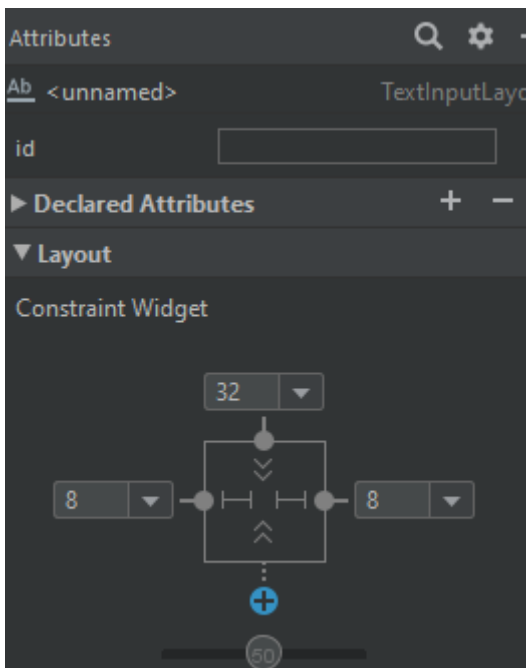
Ao pesquisar “inp” temos a opção “TextInputLayout”. Devemos arrastar para o nosso layout:



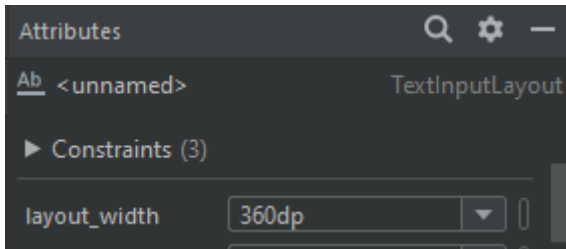
Agora é necessário arrastar a bolinha da esquerda para a margem esquerda, a bolinha da margem direita para a direita e a bolinha de cima para cima. Depois vamos na parte de “Attributes”.



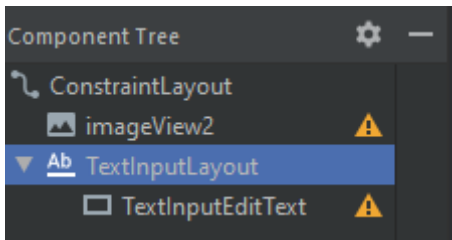
Na parte de “Attributes” em “Layout” alteraremos para esses valores:



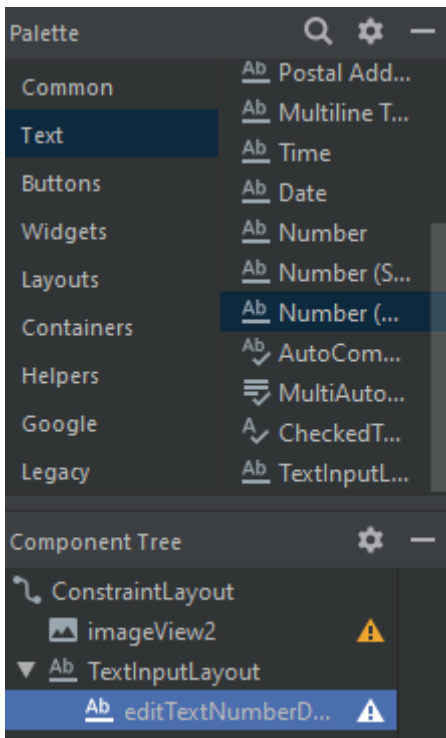
Ainda na parte de “Attributes” porém em “Constraints”, alteramos o valor de “layout_width” para 360dp:



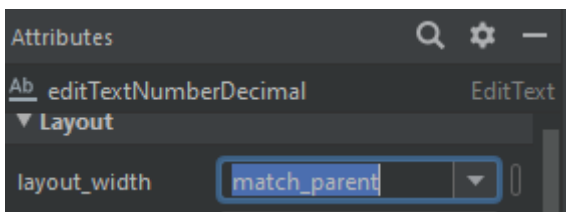
Deletamos o padrão de texto “TextInputEditText”:



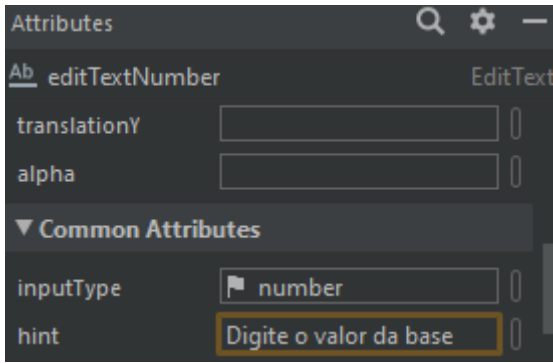
Arrastamos a opção “Number (decimal)” para o “TextInputLayout”:



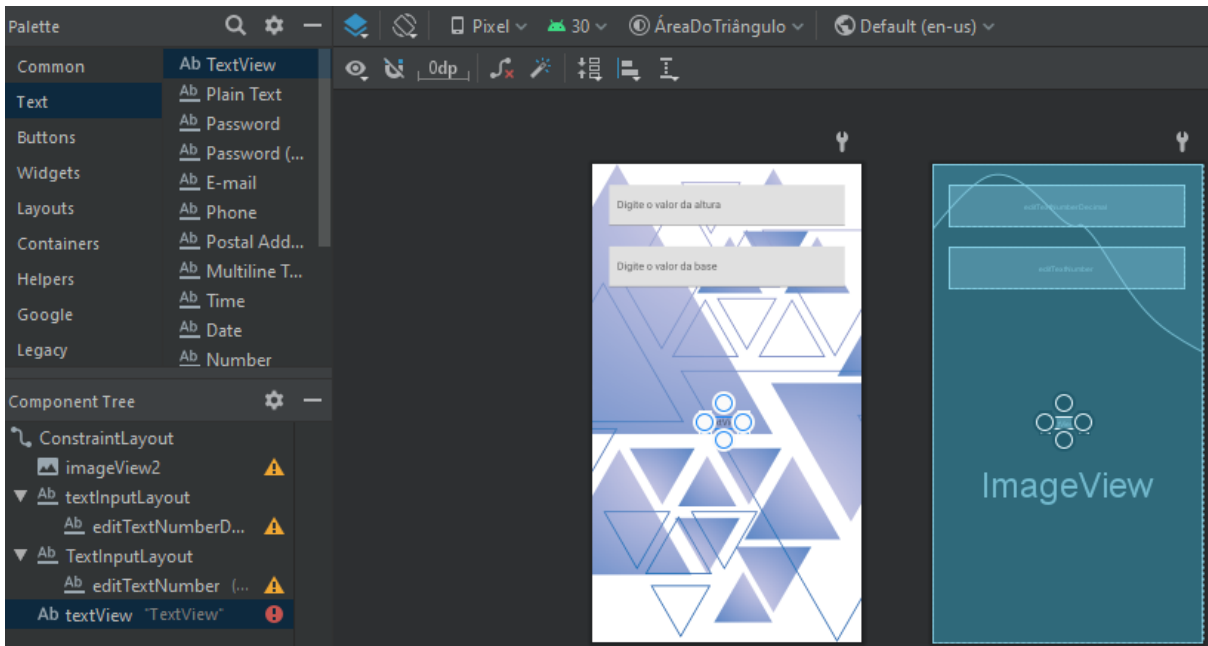
Para preencher todo o layout de texto é necessário selecionar a opção “match_parent” no “Layout”:



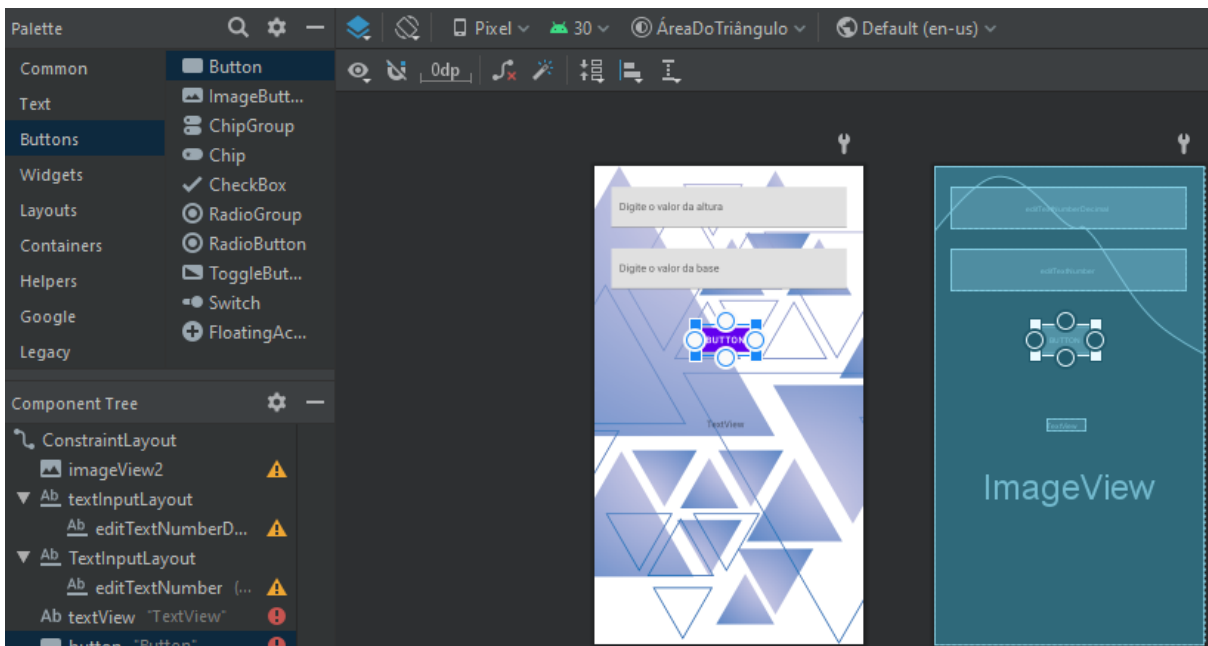
Colocamos o texto “Digite o valor da base” na parte “hint” de “Common Attributes” do Attributes para guiar o que o usuário deverá digitar:



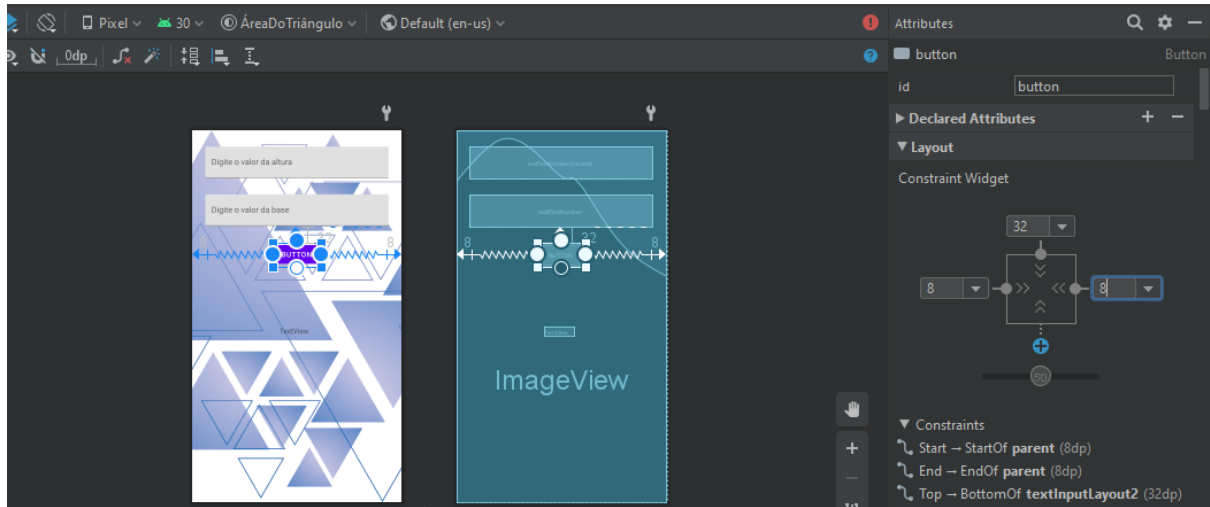
Colocamos um “TextView” para informar o resultado ao usuário:



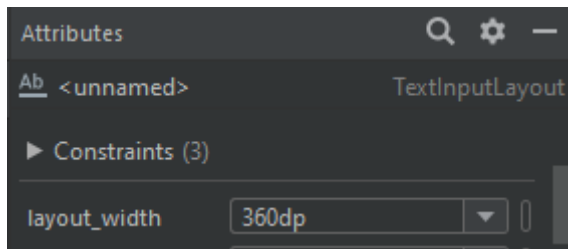
Colocamos um botão (Button) que ficará responsável por “startar” todas as ações:



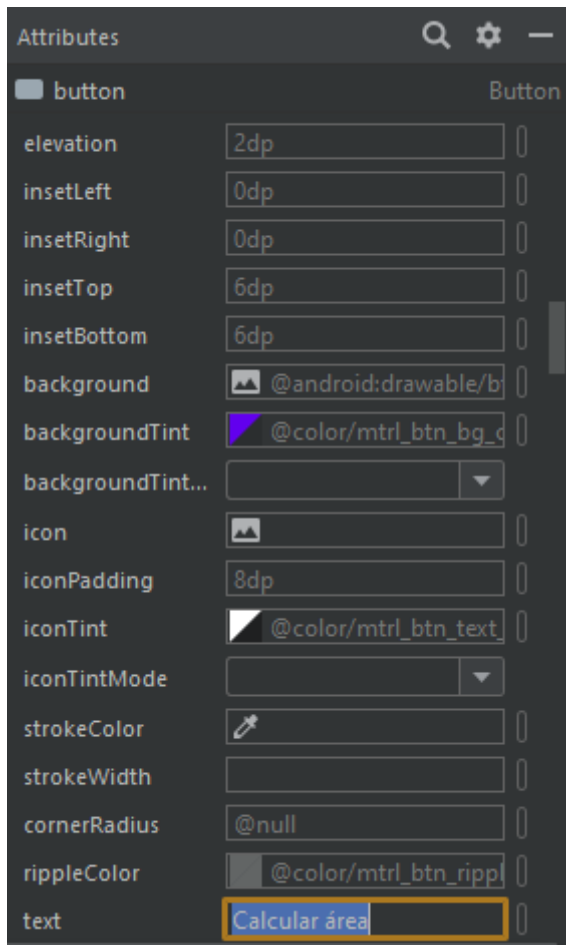
Ligamos a bolinha da esquerda para a margem esquerda, a bolinha da margem direita para a direita e a bolinha de cima para cima. Depois vamos na parte de “Attributes” e alteramos os valores como apresentado a baixo:



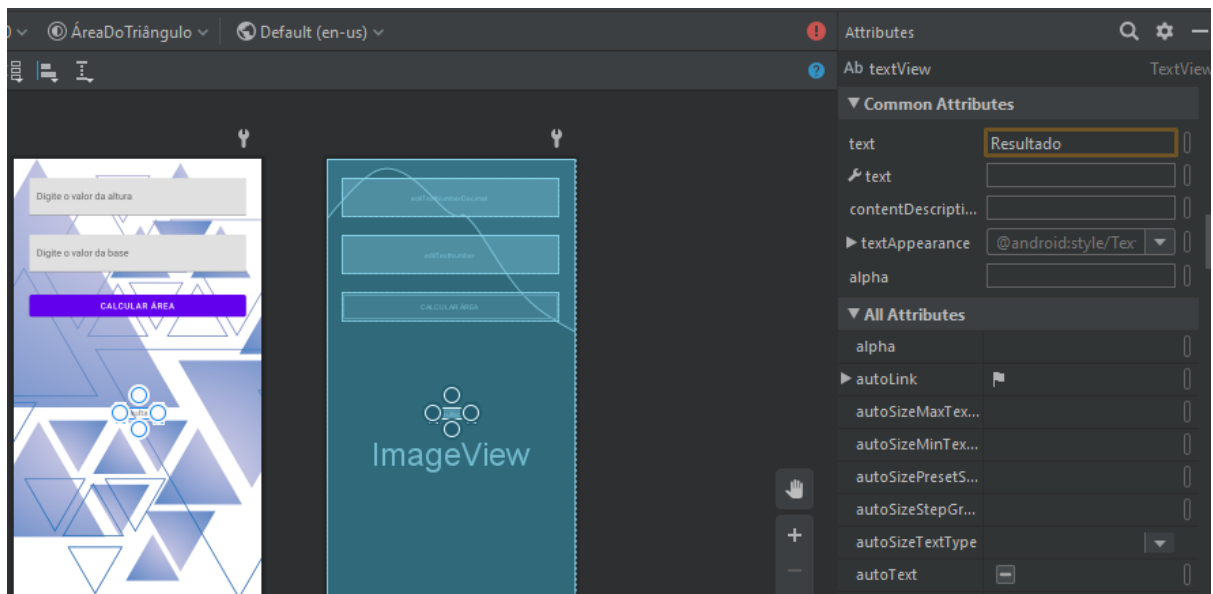
Ainda na parte de “Attributes” porém em “Constraints”, alteramos o valor de “layout_width” para 360dp:



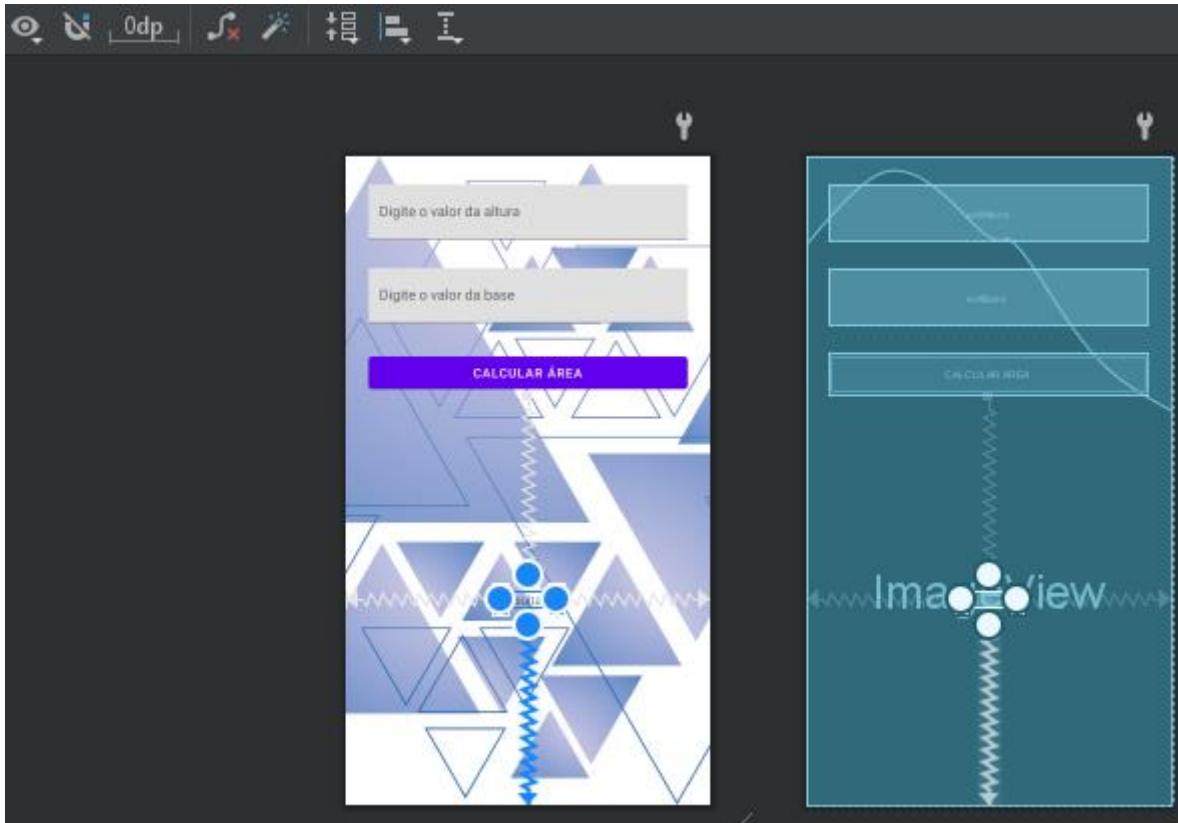
Em “Common Attributes” alteramos “text” para “Calcular área”:



Clicamos no “textView” e alteramos o “Common Attributes” a parte “text” para “Resultado”:

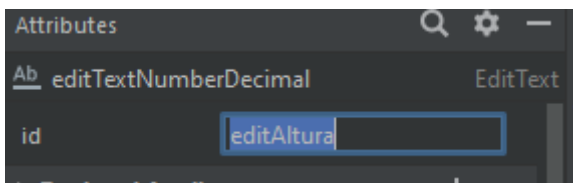


Ligamos a bolinha da esquerda para a margem esquerda, a bolinha da margem direita para a direita e a bolinha de cima para cima:

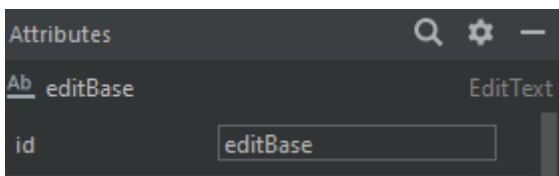


Alteraremos os nomes para que facilite a identificação dos componentes na parte de código.

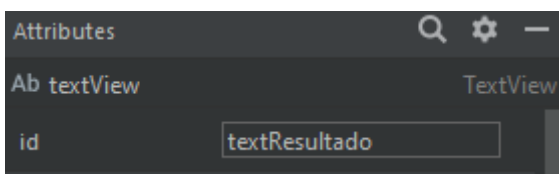
Clicamos no primeiro TextInputLayout. Em “Attributes” alteramos o “id” para “editAltura”:



Clicamos no segundo TextInputLayout. Em “Attributes” alteramos o “id” para “editBase”:



Clicamos no TextView. Em “Attributes” alteramos o “id” para “textResultado”:





JAVA
Segunda Aplicação

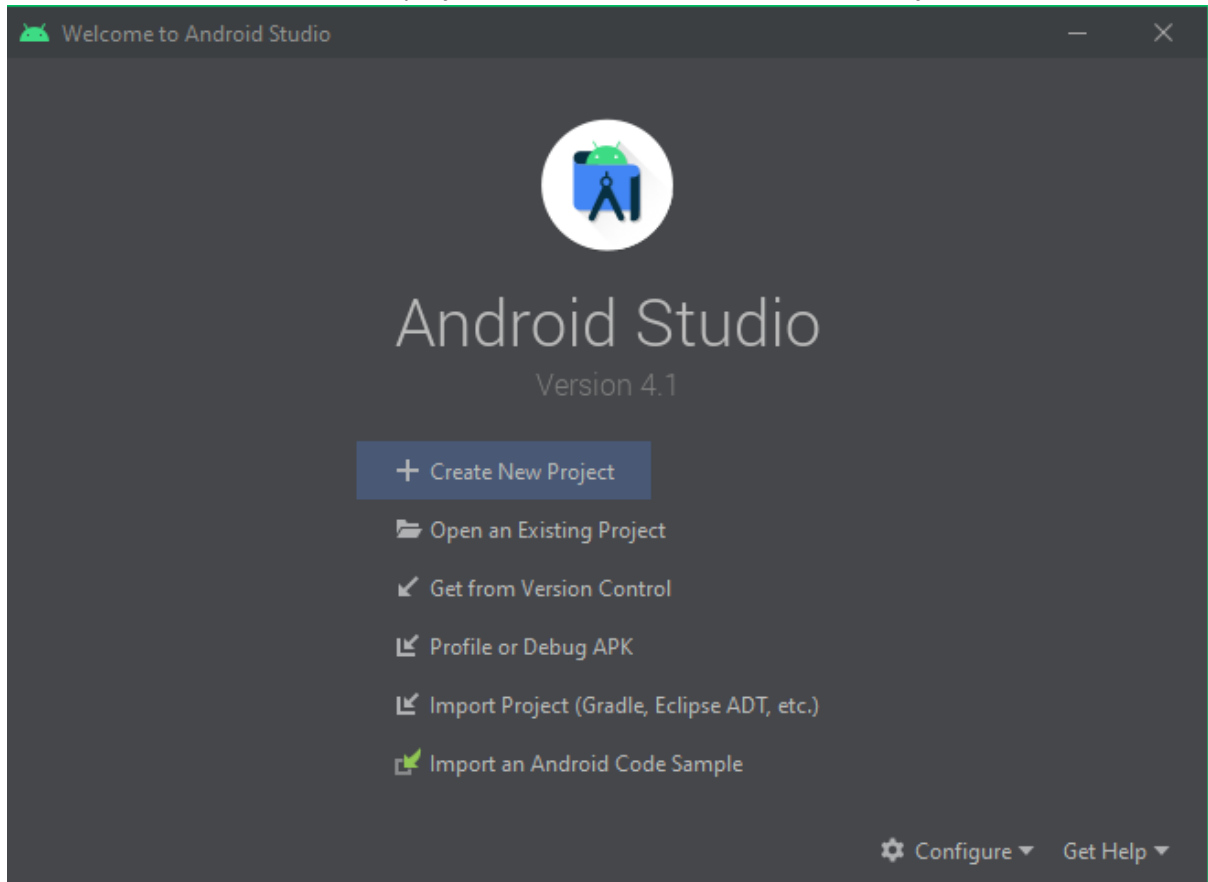
Adalto Luis Alves Pereira Filho
Rayanne Rayssa de Albuquerque Gonçalves
CTII - 418

**Aplicação solicitada pelo
docente à disciplina de Projeto de
Sistemas lecionada no Instituto
Federal de Educação, Ciências e
Tecnologia de São Paulo – Campus
Cubatão.**

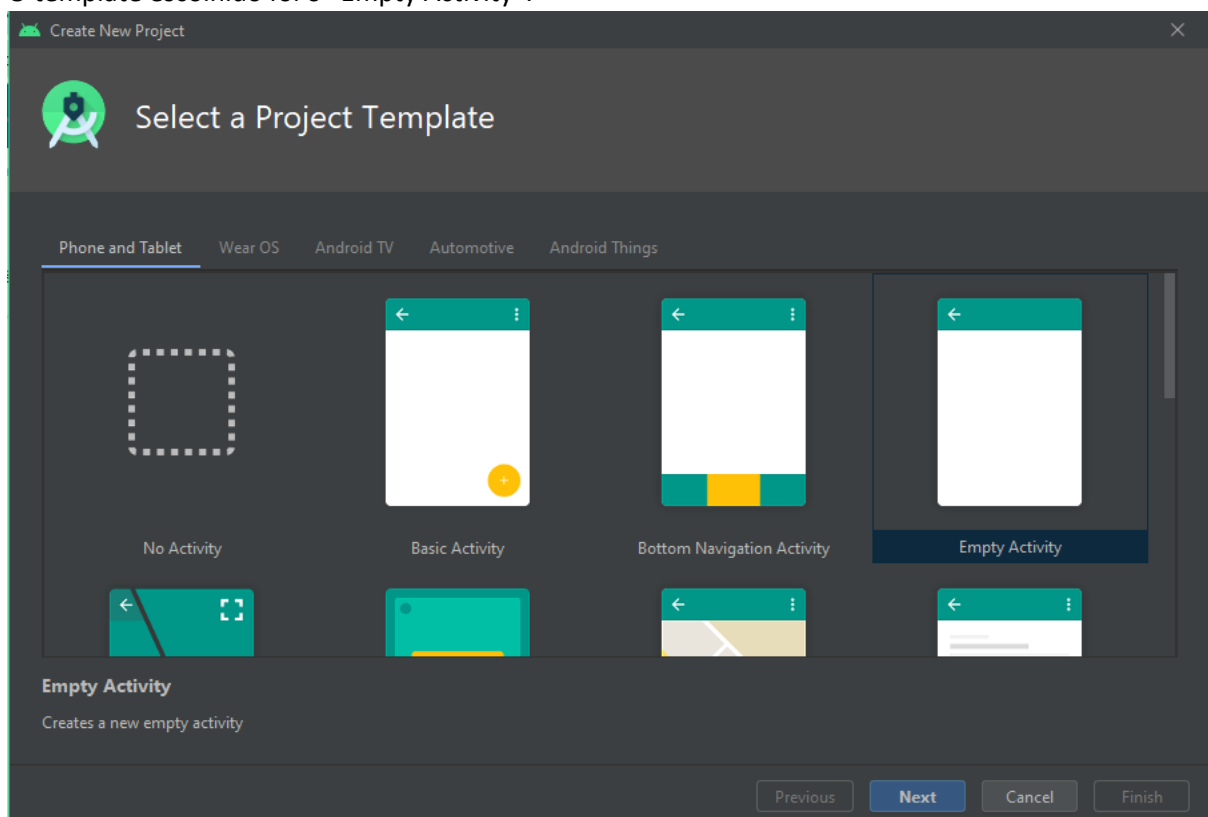
Orientador: Prof. Mauricio Neves Asenjo

Cubatão – SP
2° Semestre de 2020

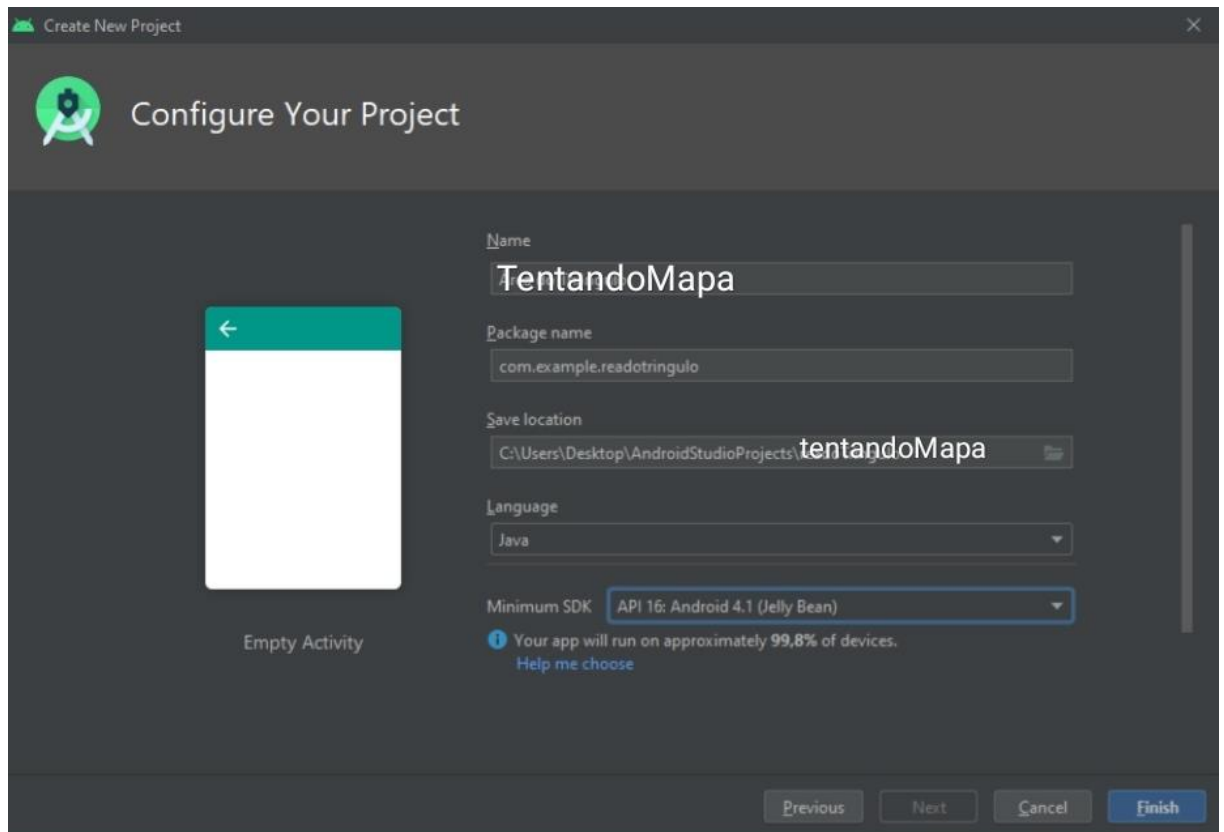
Primeiro devemos criar um novo projeto clicando no botão “Create New Project”:



O template escolhido foi o “Empty Activity”:



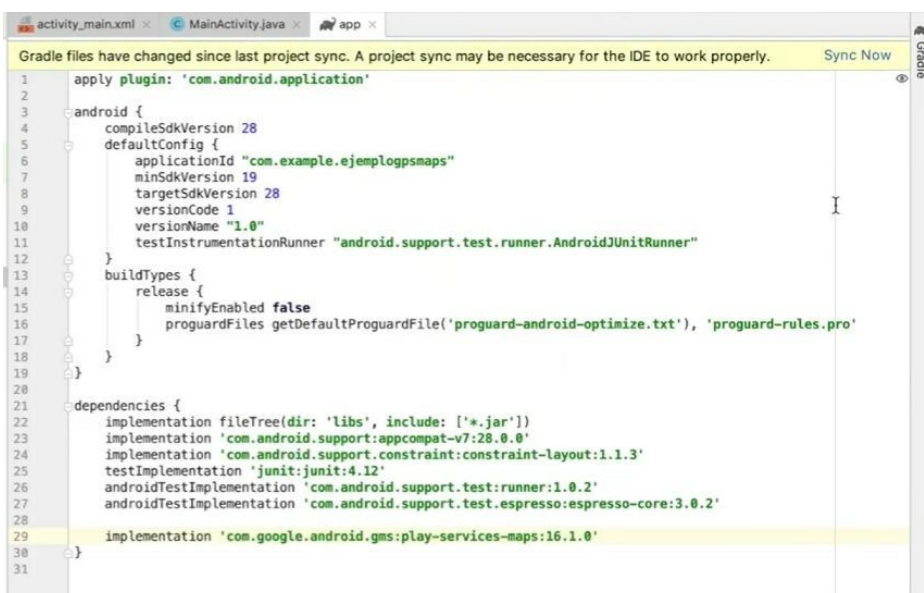
Nessa parte nomeamos o projeto para "TentandoMapa", mas daria para nomear o pacote, selecionar o local onde será salvo, escolher a linguagem (que no caso do grupo é Java) e escolhemos a versão mínima para rodar nossa aplicação:



Agora vamos importar a biblioteca:

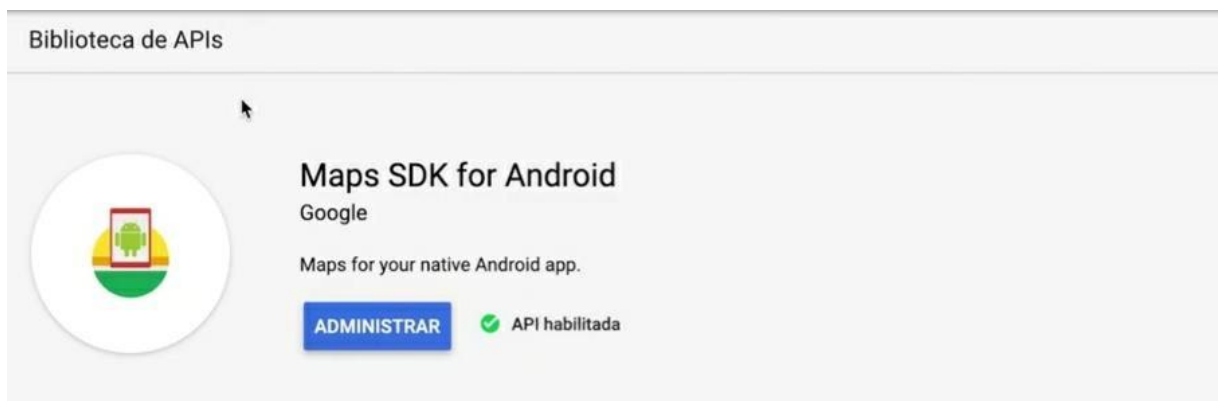
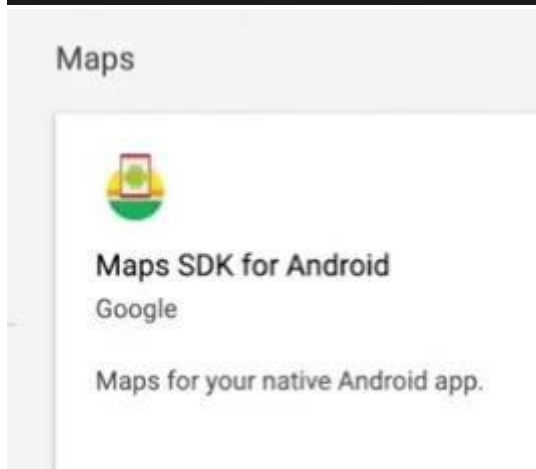
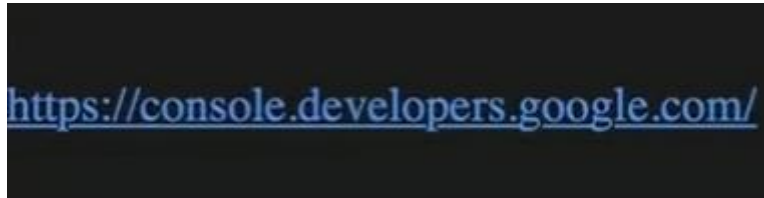
```
implementation 'com.google.android.gms:play-services-maps:16.1.0'
```

Na classe "build.gradle (Module:app)":



Depois deveremos clicar em “sync” para sincronizar/atualizar o projeto.

Agora é necessário dar permissões no “AndroidManifest.xml” e colocar q chave de acesso do Google APIs que pode ser gerará através do link:



Tipo

APIs y servicios

Última actualización

9/1/19 14:32

Categoría

Maps

Nombre del servicio

maps-android-backend.googleapis.com

Visión general

Add maps based on Google Maps data to your Android application with the Maps SDK for Android. The SDK automatically handles access to Google Maps servers, map display and response to user gestures such as clicks and drags.

Información sobre Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

Google APIs Pruebas Maps

API APIs y servicios

Credenciales

Credenciales Pantalla de consentimiento de OAuth Verificación de dominio

Crear credenciales Eliminar

Crea credenciales para acceder a tus API activadas. Para obtener más información, consulta la documentación de autenticación.

Claves de API

<input type="checkbox"/>	Nombre	Fecha de creación	Restricciones	Clave
<input type="checkbox"/>	Clave de API 1	23 feb. 2019	Ninguna	AlzaSyB-adBmGNxqfDqy11HDA1-xUbSJS2J1Y

Android

app

- manifests
 - AndroidManifest.xml
- java
 - com.example.ejemplogpsmaps
 - MainActivity
 - com.example.ejemplogpsmaps (androidTest)
 - com.example.ejemplogpsmaps (test)
 - generatedJava
 - res
- Gradle Scripts
 - build.gradle (Project: EjemploGPSMaps)
 - build.gradle (Module: app)
 - gradle-wrapper.properties (Gradle Version)
 - proguard-rules.pro (ProGuard Rules for app)
 - gradle.properties (Project Properties)
 - settings.gradle (Project Settings)
 - local.properties (SDK Location)

```

5 <uses-permission android:name="android.permission.INTERNET" />
6 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
7
8 <application
9     android:allowBackup="true"
10    android:icon="@mipmap/ic_launcher"
11    android:label="@string/app_name"
12    android:roundIcon="@mipmap/ic_launcher_round"
13    android:supportRtl="true"
14    android:theme="@style/AppTheme">
15    <activity android:name=".MainActivity">
16        <intent-filter>
17            <action android:name="android.intent.action.MAIN" />
18
19            <category android:name="android.intent.category.LAUNCHER" />
20        </intent-filter>
21    </activity>
22
23    <meta-data
24        android:name="com.google.android.gms.version"
25        android:value="@integer/google_play_services_version" />
26    <meta-data
27        android:name="com.google.android.maps.v2.API_KEY"
28        android:value="AlzaSyB-adBmGNxqfDqy11HDA1-xUbSJS2J1Y" />
29    </meta-data>
30 </application>

```

Criaremos a classe "Localizacion", dentro do pacote que contém a classe "MainActivity", que herdará "LocationListener"

activity_main.xml MainActivity.java AndroidManifest.xml Localizacion.java

```

1 package com.example.ejemplogpsmaps;
2
3 import android.location.Location;
4 import android.location.LocationListener;
5 import android.os.Bundle;
6 import android.widget.TextView;
7
8 public class Localizacion implements LocationListener {
9
10    MainActivity mainActivity;
11    TextView tvMensaje;
12
13    public MainActivity getMainActivity() {
14        return mainActivity;
15    }
16
17    public void setMainActivity(MainActivity mainActivity, TextView tvMensaje) {
18        this.mainActivity = mainActivity;
19        this.tvMensaje = tvMensaje;
20    }

```



```

@Override
public void onLocationChanged(Location location) {
    // Este metodo se ejecuta cuando el GPS recibe nuevas coordenadas
    String texto = "Mi ubicación es: \n"
        + "Latitud = " + location.getLatitude() + "\n"
        + "Longitud = " + location.getLongitude();

    tvMensaje.setText(texto);

    mapa(location.getLatitude(), location.getLongitude());
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    switch (status) {
        case LocationProvider.AVAILABLE:
            Log.d( tag: "debug", msg: "LocationProvider.AVAILABLE");
            break;
        case LocationProvider.OUT_OF_SERVICE:
            Log.d( tag: "debug", msg: "LocationProvider.OUT_OF_SERVICE");
            break;
        case LocationProvider.TEMPORARILY_UNAVAILABLE:
            Log.d( tag: "debug", msg: "LocationProvider.TEMPORARILY_UNAVAILABLE");
            break;
    }
}

@Override
public void onProviderEnabled(String provider) {
    tvMensaje.setText("GPS Activado");
}

@Override
public void onProviderDisabled(String provider) {
    tvMensaje.setText("GPS Desactivado");
}

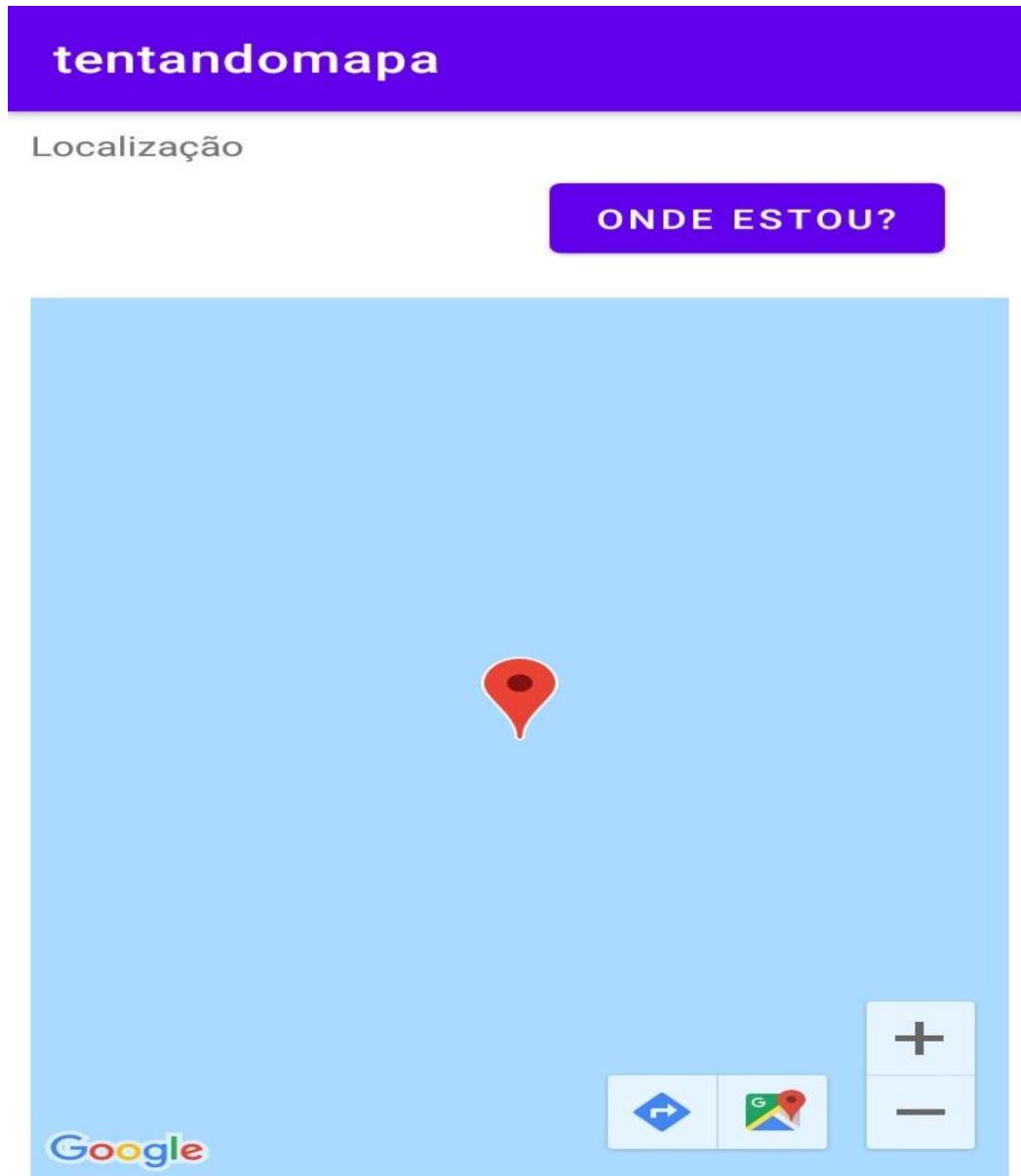
```

Localizacion > onProviderDisabled()

Criaremos a classe "FragmentMaps", dentro do pacote que contém a classe "MainActivity":

```
activity_main.xml x MainActivity.java x AndroidManifest.xml x Localizacion.java x FragmentMaps.java x
6 public class FragmentMaps extends SupportMapFragment implements OnMapReadyCallback {
7
8     double lat, lon;
9
10    public FragmentMaps() { }
11
12
13    @Override
14    public View onCreateView(LayoutInflater inflater, ViewGroup viewGroup, Bundle bundle) {
15        View rootView = super.onCreateView(inflater, viewGroup, bundle);
16
17        if(getArguments() != null) {
18            this.lat = getArguments().getDouble( key: "lat");
19            this.lon = getArguments().getDouble( key: "lon");
20        }
21
22        getMapAsync( onMapReadyCallback: this);
23
24        return rootView;
25    }
26
27    @Override
28    public void onMapReady(GoogleMap googleMap) {
29        LatLng latLng = new LatLng(lat, lon);
30
31        float zoom = 17;
32
33        googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, zoom));
34
35        googleMap.getUiSettings().setZoomControlsEnabled(true);
36
37        googleMap.addMarker(new MarkerOptions().position(latLng));
38
39        UiSettings settings = googleMap.getUiSettings();
40
41        settings.setZoomControlsEnabled(true);
42    }
43 }
44
```

Modificaremos o Layout para que fique como abaixo:



Contém TextView com id "TvMensaje", Fragment com id "fragmentMaps" e um Button com id Button.

Agora alteraremos o comportamento da classe "MainActivity" para que fique conforme o código abaixo:

```

2
3
4 import ...
5
6 public class MainActivity extends AppCompatActivity {
7
8     TextView tvMensaje;
9
10    // Minimo tiempo para updates en Milisegundos
11    private static final long MIN_TIME = 10000; // 10 segundos
12
13    @Override
14    protected void onCreate(Bundle savedInstanceState) {
15        super.onCreate(savedInstanceState);
16        setContentView(R.layout.activity_main);
17
18        tvMensaje = findViewById(R.id.tvMensaje);
19
20        if(ActivityResultCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
21           != PackageManager.PERMISSION_GRANTED
22           && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION)
23           != PackageManager.PERMISSION_GRANTED) {
24
25            ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION,}, requestCode: 1000);
26
27        } else {
28            iniciarLocalizacion();
29        }
30    }
31
32    private void iniciarLocalizacion() {
33        LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
34
35        Localizacion local = new Localizacion();
36
37        local.setMainActivity( mainActivity: this, tvMensaje);
38
39        final boolean gpsEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
40        if(!gpsEnabled) {
41            Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
42            startActivity(intent);
43        }
44    }
45
46    MainActivity -> onRequestPermissionsResult()

```

activity_main.xml x MainActivity.java x AndroidManifest.xml x Localizacion.java x FragmentMaps.java x

```

    }
    }

    private void iniciarLocalizacion() {
        LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

        Localizacion local = new Localizacion();

        local.setMainActivity( mainActivity: this, tvMensaje);

        final boolean gpsEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
        if(!gpsEnabled) {
            Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            startActivity(intent);
        }

        if(ActivityResultCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
           != PackageManager.PERMISSION_GRANTED
           && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION)
           != PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION,}, requestCode: 1000);
            return;
        }

        locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, MIN_TIME, minDistance: 0, local);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, MIN_TIME, minDistance: 0, local);

        tvMensaje.setText("Localizacion agregada");
    }

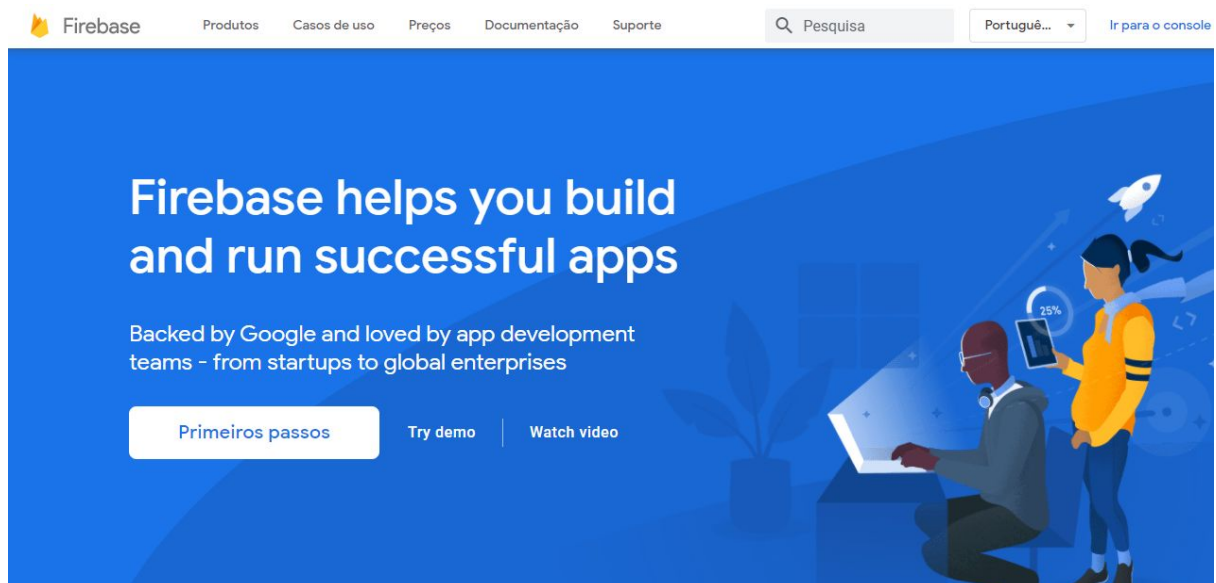
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
        if(requestCode == 1000) {
            if(grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                iniciarLocalizacion();
                return;
            }
        }
    }
}

```



Adauto Luis Alves Pereira Filho - Nome
Rayanne Rayssa de Albuquerque Gonçalves - Nome
JAVA - Grupo
CTII 418 - Turma
App04 - PJS

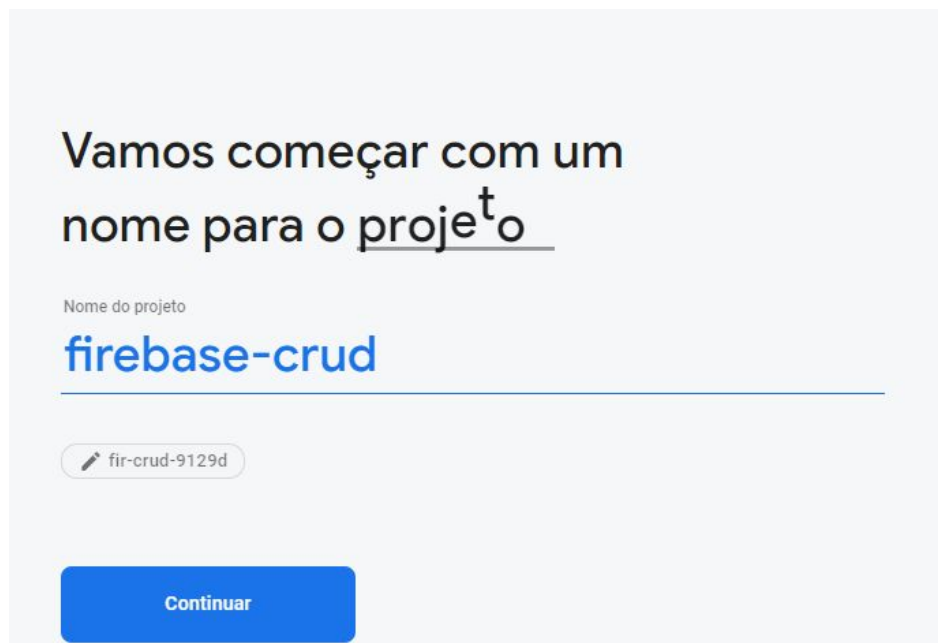
Para que possamos acessar a plataforma do FireBase (<https://firebase.google.com/?hl=pt-br>) devemos ter uma conta Gmail que esteja aberta.



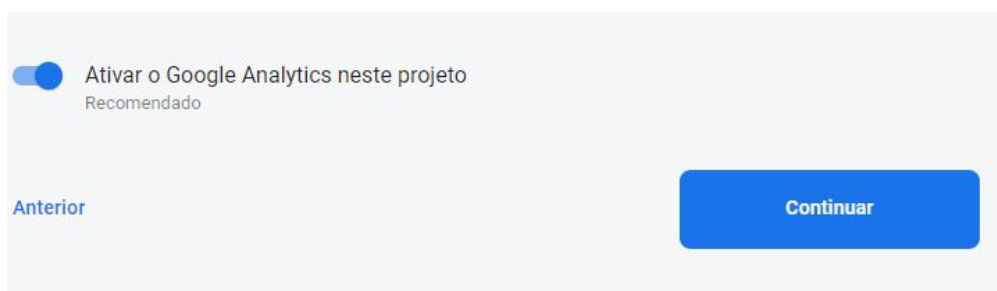
Para criar um projeto é necessário clicar no botão “Primeiros passos”. Depois precisamos clicar no botão “Adicionar Projeto”:



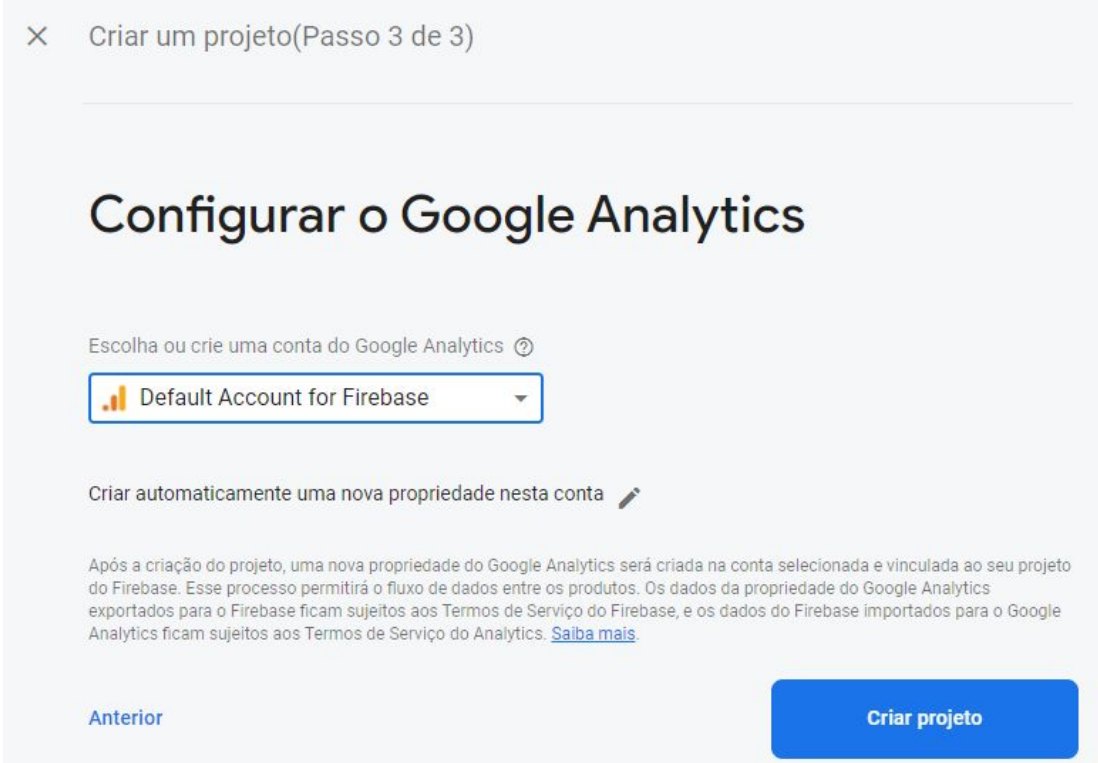
Na próxima etapa é necessário nomear o projeto. Escolhemos o nome “firebase-crud” e clicamos no botão “Continuar”:



O próximo passo explica o que é o Google Analytics e sua importância para os projetos. Devemos garantir que a opção “Ativar o Google Analytics neste projeto” esteja ativa. Feito isso, clicamos no botão “Continuar”:




A próxima etapa pede para escolher ou criar uma conta do Google Analytics. Escolhemos a opção de criar conta "Default Account for Firebase" e clicamos no botão "Criar Projeto":



✕ Criar um projeto(Passo 3 de 3)

Configurar o Google Analytics

Escolha ou crie uma conta do Google Analytics ⓘ

 Default Account for Firebase ▾

Criar automaticamente uma nova propriedade nesta conta ✎

Após a criação do projeto, uma nova propriedade do Google Analytics será criada na conta selecionada e vinculada ao seu projeto do Firebase. Esse processo permitirá o fluxo de dados entre os produtos. Os dados da propriedade do Google Analytics exportados para o Firebase ficam sujeitos aos Termos de Serviço do Firebase, e os dados do Firebase importados para o Google Analytics ficam sujeitos aos Termos de Serviço do Analytics. [Saiba mais](#).

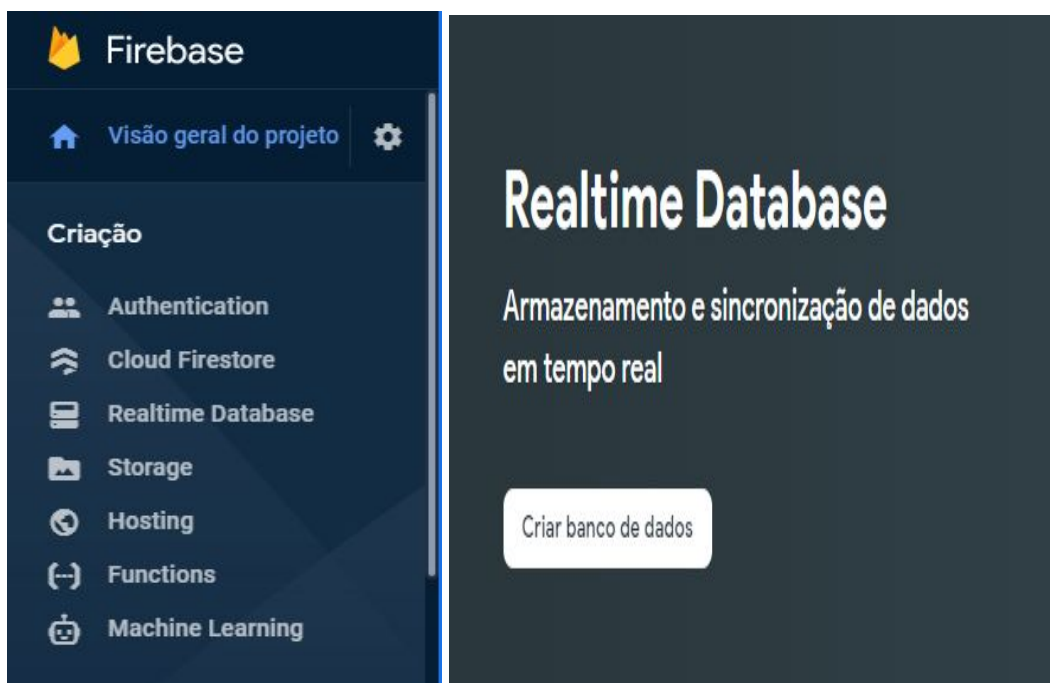
[Anterior](#) Criar projeto

Devemos clicar no botão "Continuar":

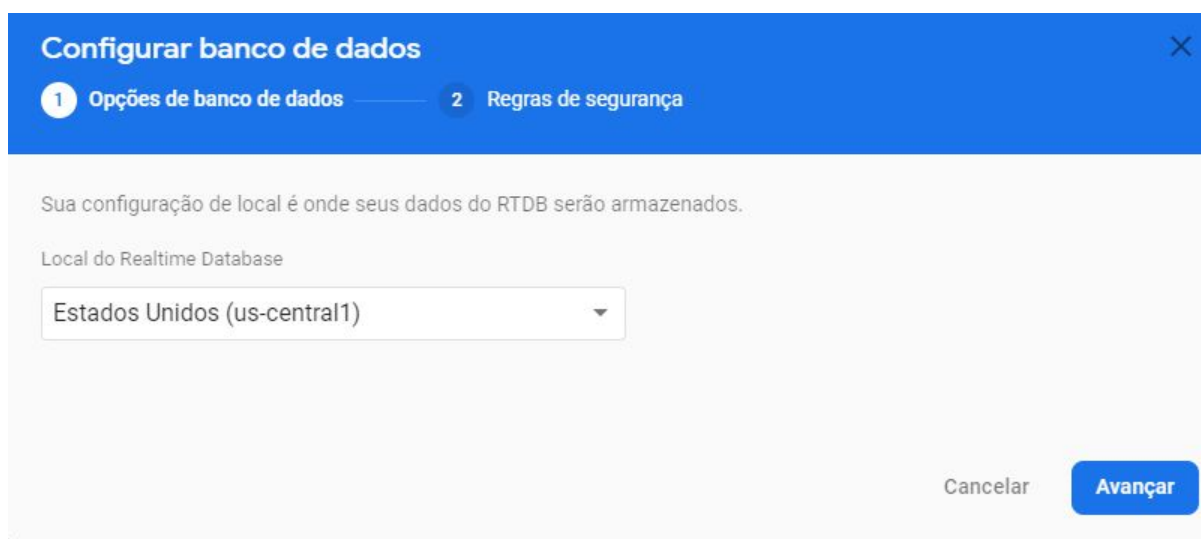


Assim teremos acesso ao painel de controle do projeto.

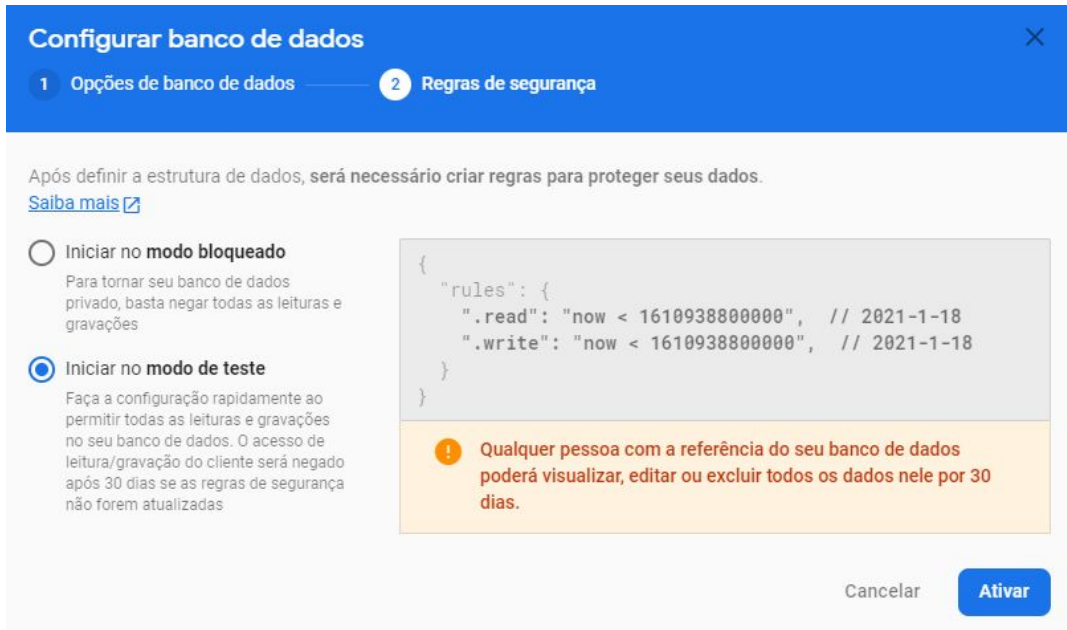
Clicando em "Criação" teremos "Realtime Database" que nos levará ao botão "Criar banco de dados":



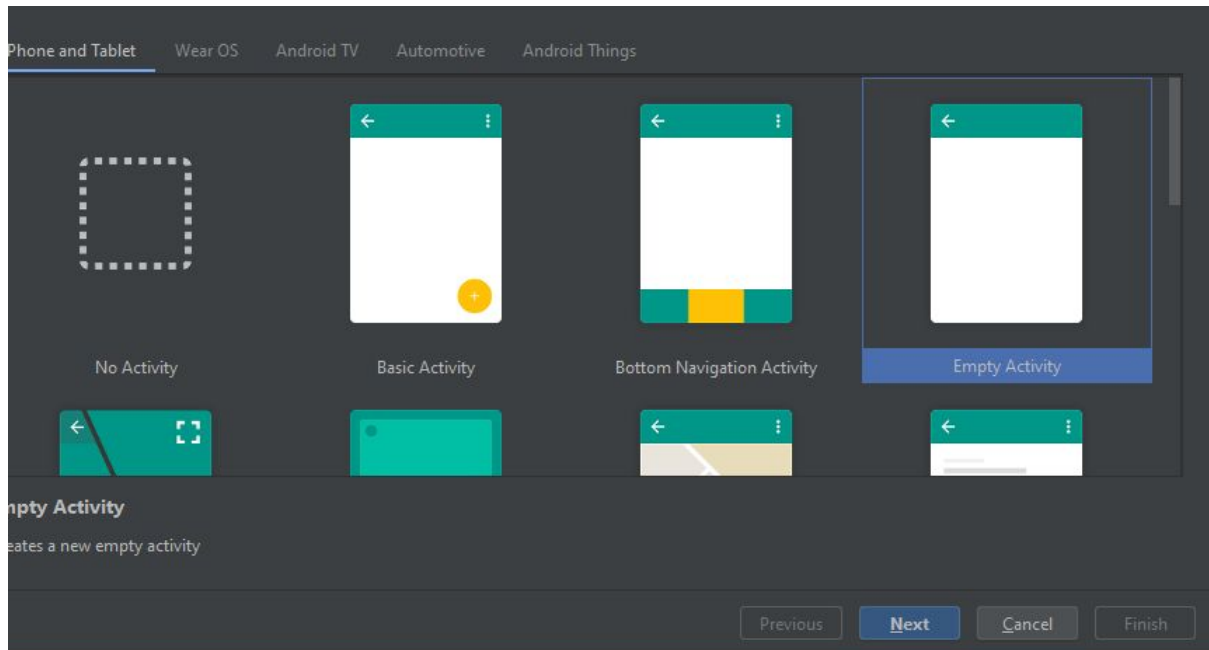
Em “configurar banco de dados” optamos a opção “Estados Unidos (us-central1)” e clicamos no botão “Avançar”:



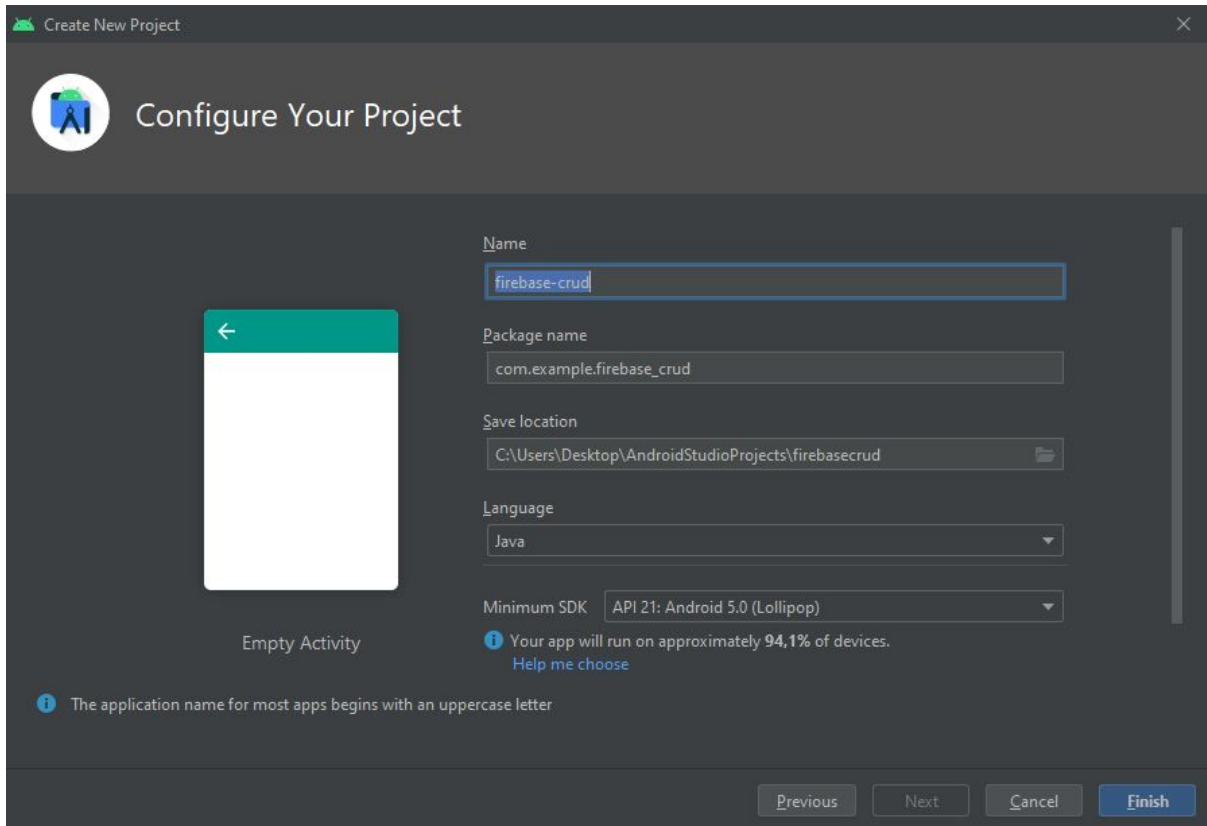
Na parte “Regras de segurança” optamos em “iniciar modo de teste” e clicamos no botão “Ativar”:



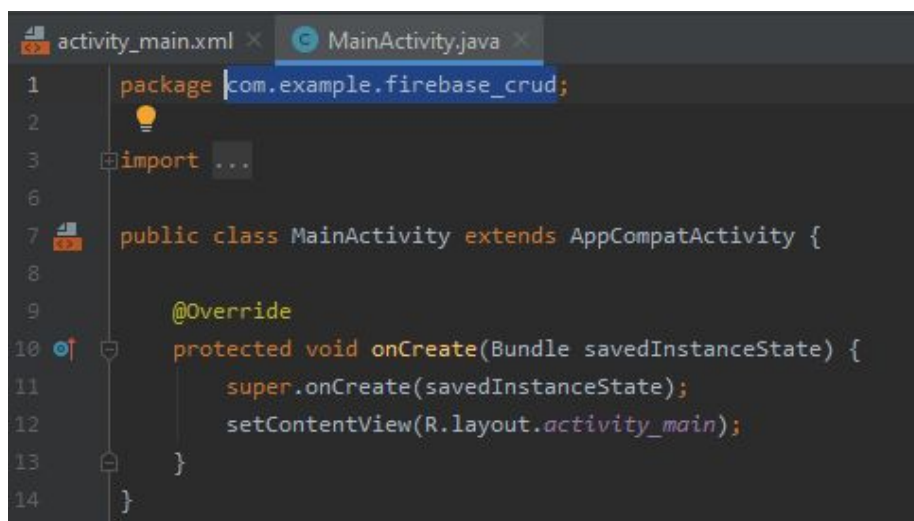
Como não temos nenhum registro/dado, ainda, o nosso banco de dados está null (vazio). Para adicionar o Android Studio ao Firebase devemos criar o nosso projeto no Android Studio. Opção “Empty Activity”, clique no botão “Next”:



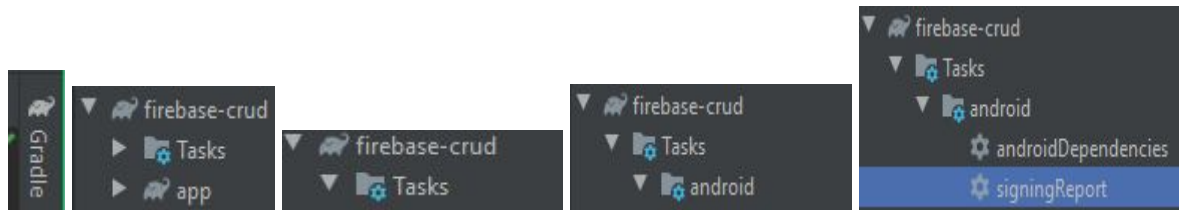
Nomeamos o projeto para “firebase-crud” e optamos pelas configurações abaixo, e clicamos no botão “Finish”:



Uma vez criado o projeto, devemos copiar o nome do pacote que contém o MainActivity.java, que no caso é “com.example.firebase_crud”:



Optamos em copiar também o certificado de assinatura da depuração SHA-1. Para isso, é necessário clicar em “Gradle”, “firebase-crud”, “Tasks”, “Android” e “signingReport” respectivamente:



Na parte inferior do projeto aparecerá algumas séries. Copiamos a “SHA1. Somente as séries, não devendo copiar a parte ”SHA1: “:

```
Variant: debug
Config: debug
Store: C:\Users\Desktop\.android\debug.keystore
Alias: AndroidDebugKey
MD5: E6:59:F5:C9:19:34:02:63:BD:CA:1E:AB:15:44:0D:6B
SHA1: D2:C2:D0:3D:CF:6A:37:E3:DB:91:EE:BB:3D:10:94:0F:21:62:26:D6
SHA-256: 6D:8C:2A:D8:C0:BA:AC:BF:05:D6:9B:2A:48:B1:E3:F2:4C:28:D9:75:35:95:49:FD:25:DC:EE:F7:A1:0C:54:7A
Valid until: Segunda-feira, 28 de Novembro de 2050
-----
```

Depois de Copiar os 2 itens acima, vamos retornar ao site do Firebase, clicar em “Visão geral do projeto”:



Devemos clicar na opção “Android”:



Para registrar o App colocaremos o nome do pacote no primeiro campo de texto e a sequência no último campo e clicaremos no botão “Registrar app”:

1 Registrar app

Nome do pacote do Android [?]

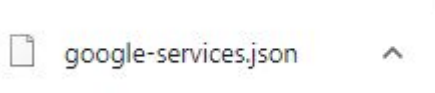
Apelido do app (opcional) [?]

Certificado de assinatura de depuração SHA-1 (opcional) [?]

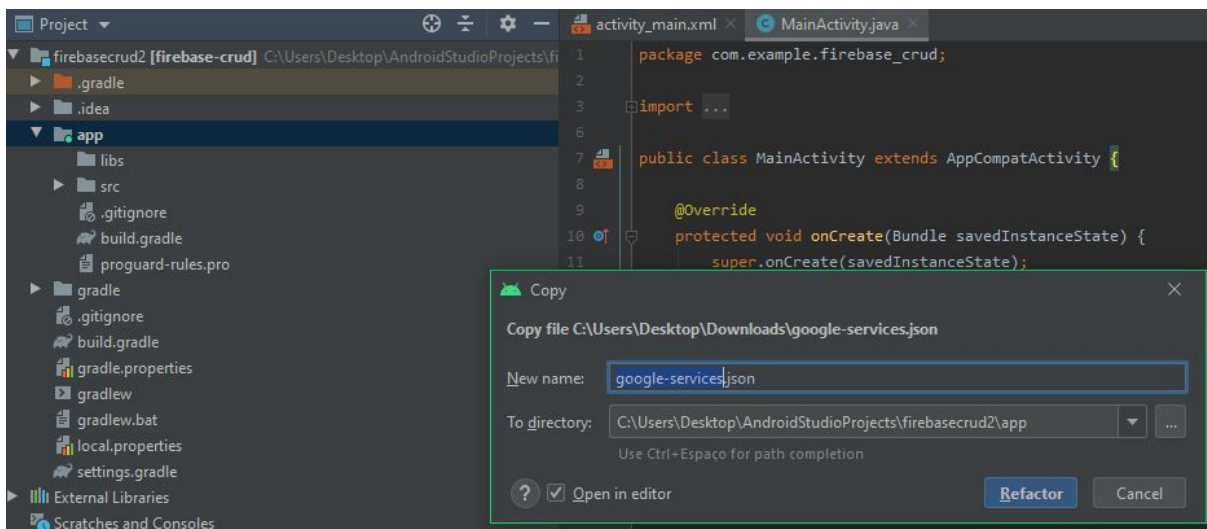
Necessário para o Dynamic Links, para o Invites e para o Login do Google ou para receber suporte por telefone no Auth. Edite o SHA-1 nas configurações.

Registrar app

Agora deveremos baixar o arquivo “google-services.json”, clicando no botão “Fazer download de google-services.json”:



Precisamos abrir o projeto do Android Studio e mover o arquivo baixado para o diretório raiz do módulo do app Android. Para fazer isso é necessário entrar no local que o arquivo (google-services.json) foi baixado, copiar o arquivo, abrir o Android Studio, alterar a opção de Android para Projeto, clicar em “firebasecrud2”, em seguida clicar na opção “App” e colar o arquivo lá, ficando assim clicamos no botão “Refactor”:

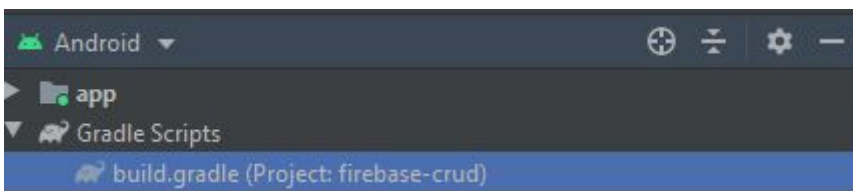


Voltaremos para o site do Firebase e clicaremos no botão "Próxima" para adicionar o SDK do Firebase. Devemos clicar na opção dentro do círculo para copiar uma linha que iremos colar no projeto do Visual Studio:

build.gradle no nível do projeto (<project>/build.gradle):

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.4'
    }
}
```

Devemos voltar para a opção "Android", clicar em "Gradle Scripts" e em seguida clicar em "build.gradle":



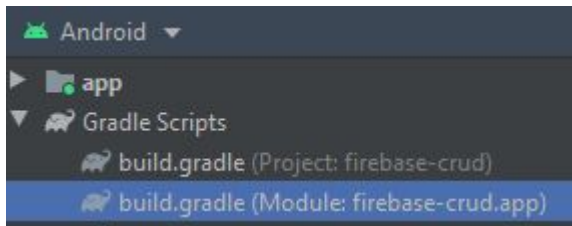
Colaremos dentro de "dependencies":

```
dependencies {
    classpath "com.android.tools.build:gradle:4.1.1"
    classpath 'com.google.gms:google-services:4.3.4'
```

Retornamos para a página do Firebase para clicar no local onde marcamos um círculo e copiar linha de código para o projeto do Visual Studio:

```
// Add the dependency for the Firebase SDK for Google Analytics
// When using the BoM, don't specify versions in Firebase dependencies
implementation 'com.google.firebase:firebase-analytics'
```

Retornando ao projeto deveremos clicar em "build.gradle (Module: firebase-crud.app)" para colar dentro de "dependencies":



```
dependencies {  
  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'com.google.android.material:material:1.2.1'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    implementation 'com.google.firebase:firebase-analytics'  
}
```

Retornamos para a página do Firebase para clicar no local onde marcamos um círculo e copiar linha de código para o projeto do Visual Studio:

```
apply plugin: 'com.android.application'  
// Add this line
```

No projeto, colaremos antes de “android {”:

```
apply plugin: 'com.android.application'
```

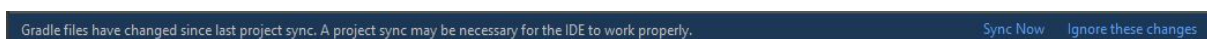
Retornamos para a página do Firebase para clicar no local onde marcamos um círculo e copiar linha de código para o projeto do Visual Studio:

```
apply plugin: 'com.google.gms.google-services'
```

Colaremos abaixo da última linha de código colada:

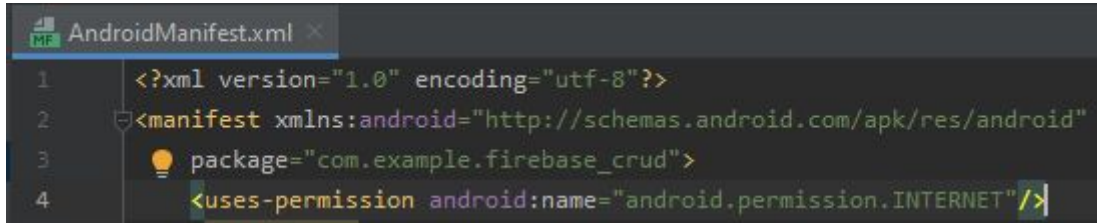
```
apply plugin: 'com.android.application'  
// Add the following line:  
apply plugin: 'com.google.gms.google-services' // Google Services plugin  
android {
```

Agora precisamos sincronizar. Clicar na opção “Sync now”:



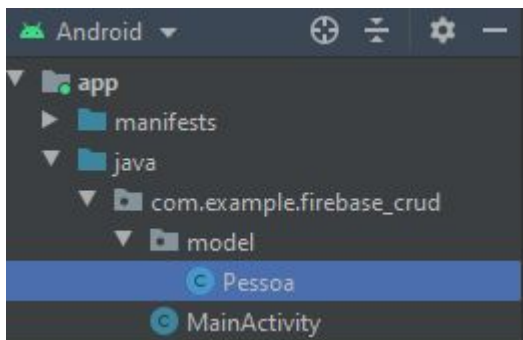
Retornamos para a página do Firebase para clicar no botão "Próxima" e na página que é aberta clicaremos no botão “Continuar no console”.

Voltando para o projeto no Android Studio, vamos abrir o “AndroidManifest.xml” para adicionar a permissão de internet:

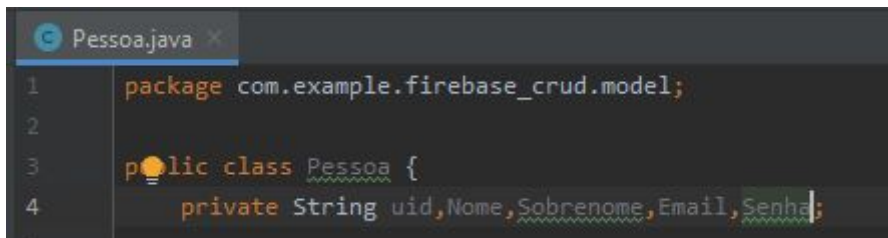


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.firebase_crud">
    <uses-permission android:name="android.permission.INTERNET"/>
```

Agora vamos criar um pacote e dentro desse pacote devemos criar a classe "Pessoa":



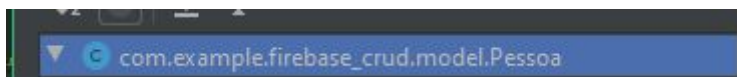
Vamos declarar 5 variáveis, sendo elas de tipo String: uid (responsável pelo ID das pessoas que iremos trabalhar), Nome, Sobrenome, Email e Senha.



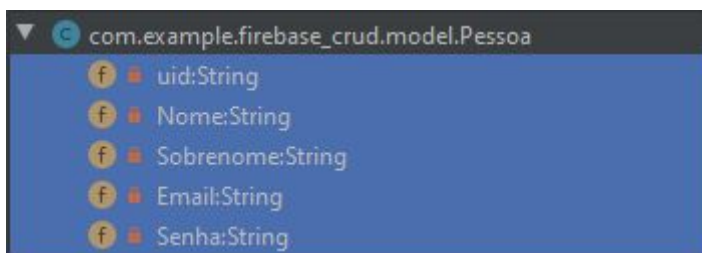
```
package com.example.firebase_crud.model;

public class Pessoa {
    private String uid, Nome, Sobrenome, Email, Senha;
```

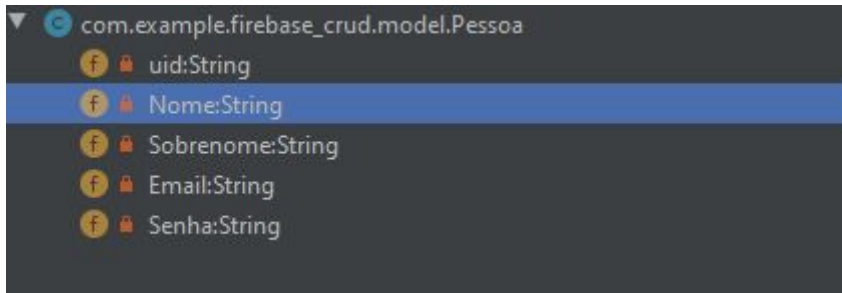
Agora vamos gerar um construtor, através do comando Alt+Insert, selecionamos “com.example.firebase_crud.model.Pessoa” e clicaremos no botão “Ok”:



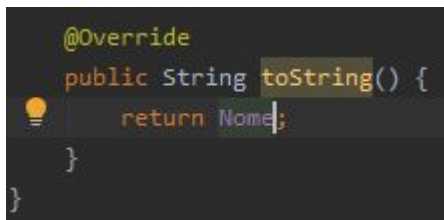
Agora vamos gerar Getters e Setters, através do comando Alt+Insert, selecionamos todas as opções que estão em azul na imagem abaixo, pressionando o botão Ctrl, e clicaremos no botão “OK”:



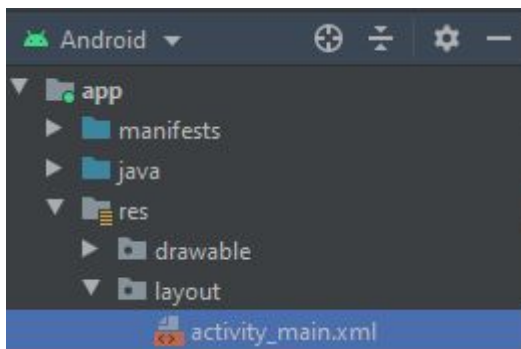
Agora vamos gerar “toString()”, através do comando Alt+Insert, selecionamos apenas a opção que está em azul na imagem abaixo e clicaremos no botão “OK”:



No “return” da “toString()” criada deveremos apagar o conteúdo e digitar “Nome”:



Agora devemos abrir o “layout”:



Selecionaremos a opção “Split” para alterar o código do layout e vê como está ficando:



Apagaremos “androidx.constraintlayout.widget.ConstraintLayout” e no lugar colocaremos “LinearLayout”. Adicionaremos “ android:orientation="vertical" ” no LinearLayout e “ android:padding="12dp" ”. Após isso, apagaremos toda a TextView:


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="12dp"
  android:orientation="vertical"
  tools:context=".MainActivity">
  |
</LinearLayout>
```

Adicionaremos TextInputLayout e TextInputEditText para receber os dados que o usuário deverão digitar. Serão 4, Nome, Sobrenome, Email e Senha:

```
<com.google.android.material.textfield.TextInputLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:hint="Nome">
  <com.google.android.material.textfield.TextInputEditText
    android:id="@+id/txt_nomePessoa"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:paddingTop="12dp"/>
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:hint="Sobrenome">
  <com.google.android.material.textfield.TextInputEditText
    android:id="@+id/txt_sobrenomePessoa"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:paddingTop="12dp"/>
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email" >
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/txt_emailPessoa"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textEmailAddress"
        android:paddingTop="12dp" />
</com.google.android.material.textfield.TextInputLayout>
```

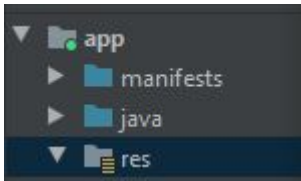
```
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Senha" >
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/txt_SenhaPessoa"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword"
        android:paddingTop="12dp" />
</com.google.android.material.textfield.TextInputLayout>
```

Para sinalizar a ListView, deveremos adicionar uma TextView em cima da ListView, que servirá para visualizar os usuários que realizaram cadastro no nosso App:

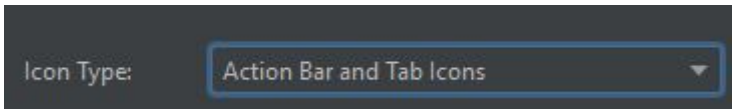
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Lista de Pessoas"
    android:textSize="30dp"
    android:layout_margin="30dp"
    android:layout_gravity="center_horizontal"/>

<ListView
    android:id="@+id/lv_dadosPessoais"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

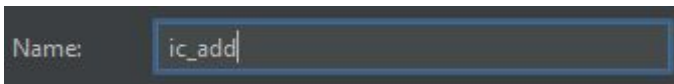
Iremos criar um pequeno menu. Para isso é necessário clicar com o botão direito do mouse no "res":



Ir na opção “new” e clicar em “Image Asset”. Na lista de opções do “Icon Type” selecionamos “Action Bar ant Tab Icons”:



Alteramos o nome para “ic_add”:



Em “Clip Art” escolhemos um ícone que remete ao ato de adicionar:

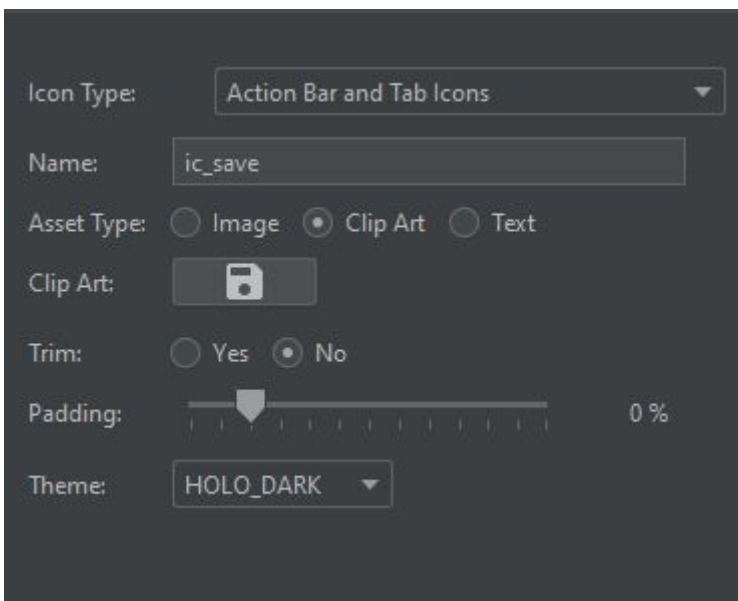


Alteramos o tema para “HOLO_DARK”:



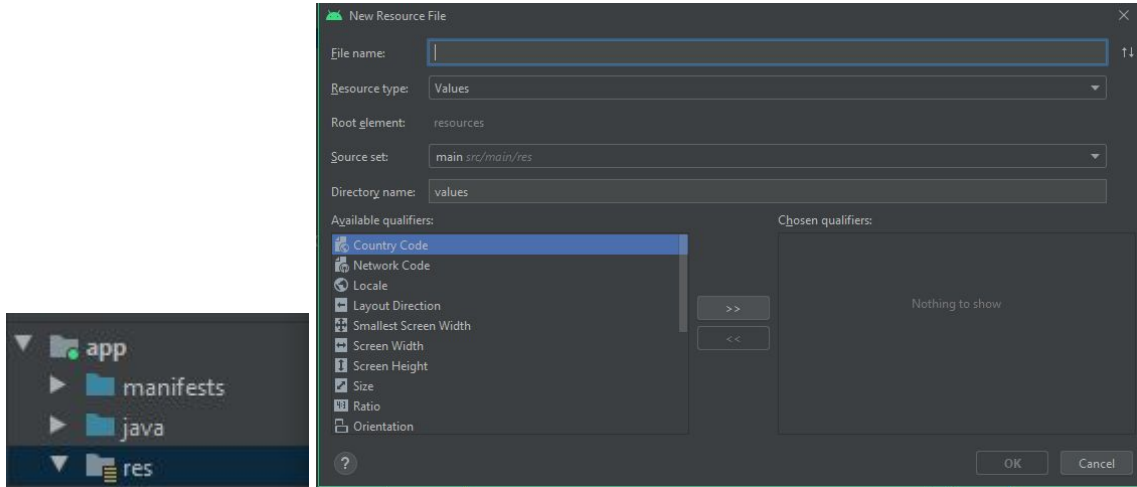
Clicamos no botão “Next” e depois no botão “Finish”

Depois de ter criado o primeiro ícone vamos repetir o processo para criar o segundo:

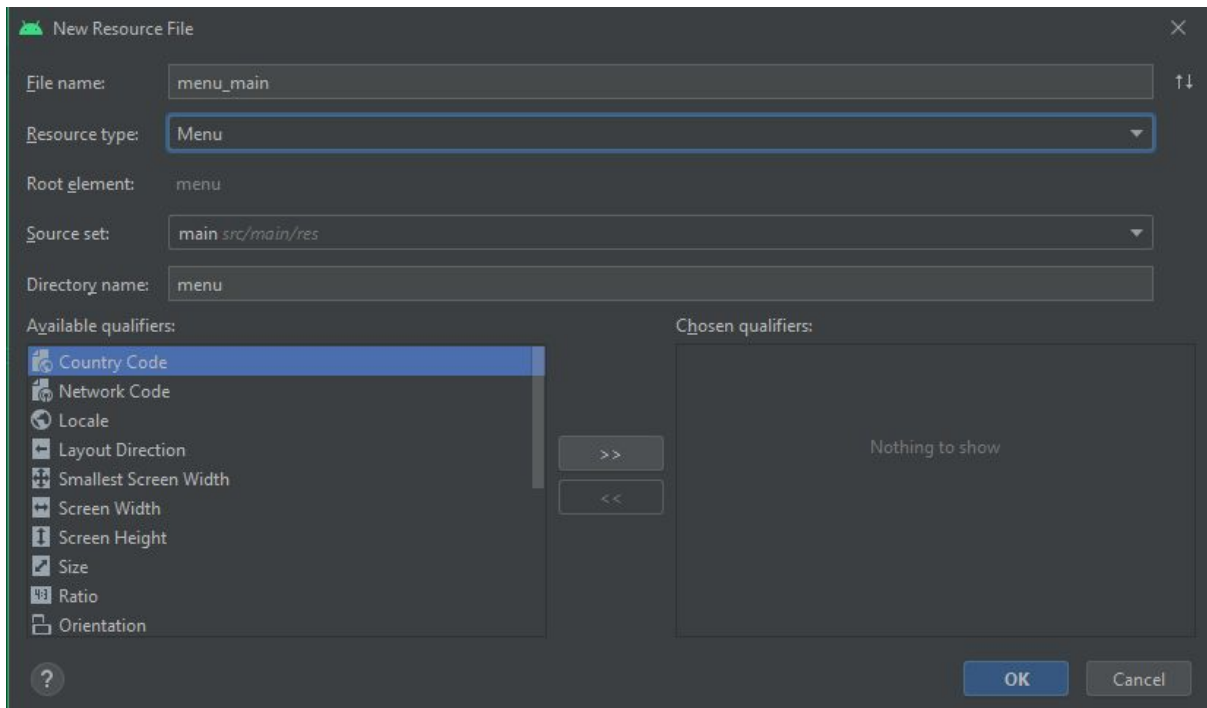


Agora criaremos o último ícone repetindo os processos anteriores, porém o “Name” será “ic_delete” e o “Clip Art” será alguma imagem de lixeira.

Para adicionar esses ícones no menu deveremos clicar em “res” depois em “new” e em seguida na opção “New Resource File”:



No campo “Resource type” escolhemos a opção “Menu”. Em “File name” nomeamos para “menu_main” e clicamos no botão “OK”:



Selecionaremos a opção “Split” para alterar o código do layout(menu_main.xml) e vê como está ficando:



Agora vamos adicionar o ícone de adicionar com o seguinte código:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/icon_add"
        android:icon="@drawable/ic_add"
        android:title="Edit done"
        app:showAsAction="always" />
</menu>
```

Para adicionar o ícone de salvar:

```
<item
    android:id="@+id/icon_salve"
    android:icon="@drawable/ic_save"
    android:title="Edit done"
    app:showAsAction="always" />
```

Para adicionar o ícone de deletar:

```
<item
    android:id="@+id/icon_delete"
    android:icon="@drawable/ic_delete"
    android:title="Edit done"
    app:showAsAction="always" />
```

Quando o usuário clicar no ícone de adicionar o app deverá gerar um novo registro, quando clicar em salvar o app deverá salvar os dados e quando o botão de deletar for clicado deverá deletar registro.

Agora vamos implementar no “build.gradle (Module: firebase-crud.app)”:

```
Gradle Scripts
  build.gradle (Project: firebase-crud)
  build.gradle (Module: firebase-crud.ap
```

Na parte de “dependencies” deveremos adicionar a linha de código abaixo:

```
//noinspection GradleCompatible,GradleCompatible
implementation 'com.android.support:design:28.0.0'
```

E em baixo da linha anteriormente adicionada, colocamos essa outra linha:

```
implementation 'com.google.firebase:firebase-database:17.0.0'
```

Agora precisamos sincronizar. Clicar na opção “Sync now”:

Na parte de MainActivity vamos implementar os ícones (que serão botões) e colocar mensagens quando eles forem utilizados no app. Para isso adicione o código abaixo:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    //switch
    switch (item.getItemId()){
        case R.id.icon_add:
            Toast.makeText( context: this, text: "Adicionado", Toast.LENGTH_SHORT).show();
            break;
        case R.id.icon_salve:
            Toast.makeText( context: this, text: "Salvo", Toast.LENGTH_SHORT).show();
            break;
        case R.id.icon_delete:
            Toast.makeText( context: this, text: "Deletado", Toast.LENGTH_SHORT).show();
            break;
        default:break;
    }
}
```

Agora vamos capturar o que o usuário digitou.

Devemos adicionar essas 2 últimas linhas de código :

```
public class MainActivity extends AppCompatActivity {

    EditText nome, sobrenome, email, senha;
    ListView dadosPessoais;
}
```

E dentro do “protected void onCreate(Bundle savedInstanceState)” fazemos a relação abaixo:

```
nome=findViewById(R.id.txt_nomePessoa);
sobrenome=findViewById(R.id.txt_sobrenomePessoa);
email=findViewById(R.id.txt_emailPessoa);
senha=findViewById(R.id.txt_SenhaPessoa);
dadosPessoais=findViewById(R.id.lv_dadosPessoais);
```

Para validar os campos vamos para o método “onOptionsItemSelected(Menu menu item)” e digitaremos o código da “String” e o “If”, conforme a foto abaixo:

```
public boolean onOptionsItemSelected(MenuItem item) {

    String nome1=nome.getText().toString(),
        sobrenome1=sobrenome.getText().toString(),
        email1=email.getText().toString(),
        senha1=senha.getText().toString();

    switch (item.getItemId()){
        case R.id.icon_add:
            if (nome1.equals("") || sobrenome1.equals("") || email1.equals("") || senha1.equals("")){
                validacao();
            }
    }
}
```

Clicaremos na lâmpada vermelha e criaremos o método “validação”.

Criaremos alguns “if” porque se algum campo de dados for vazio aparecerá mensagem de erro:

```
private void validacao() {
    String nome1=nome.getText().toString(),
        sobrenome1=sobrenome.getText().toString(),
        email1=email.getText().toString(),
        senha1=senha.getText().toString();

    if (nome1.equals("")){
        nome.setError("Campo Obrigatório");
    }else if (sobrenome1.equals("")){
        sobrenome.setError("Campo Obrigatório");
    }else if (email1.equals("")){
        email.setError("Campo Obrigatório");
    }else if (senha1.equals("")){
        senha.setError("Campo Obrigatório");
    }
}
```

Devemos voltar para onde chamamos o método "Validacao" e colocamos "}" e "{" no "else":

```
public boolean onOptionsItemSelected(MenuItem item) {  
  
    String nome1=nome.getText().toString(),  
           sobrenome1=sobrenome.getText().toString(),  
           email1=email.getText().toString(),  
           senha1=senha.getText().toString();  
  
    switch (item.getItemId()){  
        case R.id.icon_add:  
            if (nome1.equals("") || sobrenome1.equals("") || email1.equals("") || senha1.equals("")){  
                validacao();  
            }else {  
                Toast.makeText( context: this, text: "Adicionado", Toast.LENGTH_SHORT).show();  
            }  
    }  
}
```

Dentro do "else" chamaremos o método "LimparCampos" e criaremos ele a partir da lâmpada vermelha de erro que aparecerá:

```
Validacao();  
}else {  
    Toast.makeText( context: this, text: "Adicionado", Toast.LENGTH_SHORT).show();  
    LimparCampos();  
}
```

O método "LimparCampos" deverá ficar assim:

```
private void LimparCampos() {  
    nome.setText("");  
    sobrenome.setText("");  
    email.setText("");  
    senha.setText("");  
}
```

Agora vamos adicionar códigos que irão nos permitir trabalhar com o Firebase:

```
public class MainActivity extends AppCompatActivity {  
  
    EditText nome,sobrenome,email,senha;  
    ListView dadospessoais;  
  
    FirebaseDatabase firebaseDatabase;  
    DatabaseReference databaseReference;  
}
```

Vamos adicionar e criar (através da lâmpada vermelha de erro) o método "IniciarFireBase":


```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    nome=findViewById(R.id.txt_nomePessoa);
    sobrenome=findViewById(R.id.txt_sobrenomePessoa);
    email=findViewById(R.id.txt_emailPessoa);
    senha=findViewById(R.id.txt_SenhaPessoa);
    dadospessoais=findViewById(R.id.lv_dadosPessoais);

    IniciarFirebase();
}
```

O método deverá ficar assim:

```
private void IniciarFirebase() {
    FirebaseApp.initializeApp(this);
    firebaseDatabase=FirebaseDatabase.getInstance();
    firebaseDatabase.setPersistenceEnabled(true);
    databaseReference=firebaseDatabase.getReference();
}
```

Agora vamos criar uma instância da classe Pessoa, um objeto. E trabalharemos com o nosso objeto:

```
public boolean onOptionsItemSelected(MenuItem item) {

    String nome1=nome.getText().toString(),
    sobrenome1=sobrenome.getText().toString(),
    email1=email.getText().toString(),
    senha1=senha.getText().toString();

    switch (item.getItemId()){
        case R.id.icon_add:
            if (nome1.equals("") || sobrenome1.equals("") || email1.equals("") || senha1.equals("")){
                Validacao();
            }else {
                Pessoa p = new Pessoa();
                p.setUid(UUID.randomUUID().toString());
                p.setNome(nome1);
                p.setSobrenome(sobrenome1);
                p.setEmail(email1);
                p.setSenha(senha1);
                Toast.makeText(context, this, text: "Adicionado", Toast.LENGTH_SHORT).show();
                LimparCampos();
            }
        }
    }
```

Agora vamos referenciar nosso Database:

```
public boolean onOptionsItemSelected(MenuItem item) {

    String nome1=nome.getText().toString(),
    sobrenome1=sobrenome.getText().toString(),
    email1=email.getText().toString(),
    senha1=senha.getText().toString();

    switch (item.getItemId()){
        case R.id.icon_add:
            if (nome1.equals("") || sobrenome1.equals("") || email1.equals("") || senha1.equals(""))
                Validacao();
            }else {
                Pessoa p = new Pessoa();
                p.setUid(UUID.randomUUID().toString());
                p.setNome(nome1);
                p.setSobrenome(sobrenome1);
                p.setEmail(email1);
                p.setSenha(senha1);
                databaseReference.child("Pessoa").child(p.getUid()).setValue(p);
                Toast.makeText(context, this, "Adicionado", Toast.LENGTH_SHORT).show();
                LimparCampos();
            }
    }
}
```

Adicionaremos essas duas ultimas linhas de código ao começo da classe MainActivity:

```
public class MainActivity extends AppCompatActivity {

    private List<Pessoa> pessoaList = new ArrayList<Pessoa>();
    ArrayAdapter<Pessoa> pessoaAdapter;
```

Criaremos o método "ListarDados" (com a ajuda da lâmpada vermelha) abaixo da chamada do método "IniciarFirebase":

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    nome=findViewById(R.id.txt_nomePessoa);
    sobrenome=findViewById(R.id.txt_sobrenomePessoa);
    email=findViewById(R.id.txt_emailPessoa);
    senha=findViewById(R.id.txt_SenhaPessoa);
    dadospessoais=findViewById(R.id.Lv_dadosPessoais);

    IniciarFireBase();
    ListarDados();
}
```

Com o método criado, vamos fazer referência ao Database, adicionar os valores e criar outros métodos:

```
private void ListarDados() {
    databaseReference.child("Pessoa").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {

        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}
```

A classe deverá ficar assim:

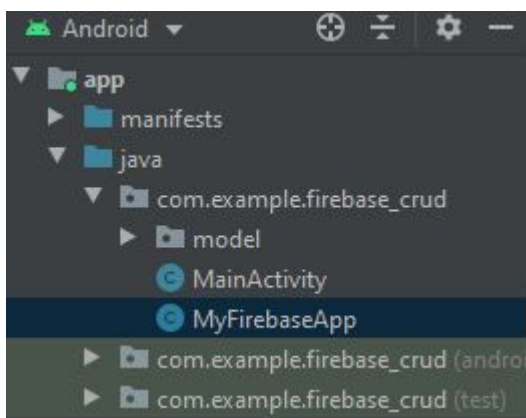
```
private void ListarDados() {
    databaseReference.child("Pessoa").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            pessoaList.clear();
            for (DataSnapshot objSnapshot : dataSnapshot.getChildren()){
                Pessoa p =objSnapshot.getValue(Pessoa.class);
                pessoaList.add(p);

                pessoaArrayAdapter=new ArrayAdapter<Pessoa>( context: MainActivity.this, android.R.layout.simple_list_item_1,pessoaList);
                dadospessoais.setAdapter(pessoaArrayAdapter);
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}
```

Agora criaremos uma classe chamada “MyFirebaseApp” que ficará dentro do pacote “com.example.firebase_crud”:



Alteraremos a declaração da classe. Dentro dela criaremos o método “onCreate”.

Importaremos a classe FirebaseDatabase:

```
MyFirebaseApp.java x
1 package com.example.firebase_crud;
2
3 import com.google.firebase.database.FirebaseDatabase;
4
5 public class MyFirebaseApp extends android.app.Application{
6
7     @Override
8     public void onCreate() {
9
10        super.onCreate();
11        FirebaseDatabase.getInstance().setPersistenceEnabled(true);
12    }
13
14 }
15
```

Para fazer com que a classe “MyFirebaseApp” seja executada primeiro devemos ir

no  na parte de “Application” devemos adicionar:

```
android:name=".MyFirebaseApp"
```

Retornando para a classe “MainActivity.java” vamos declarar a variável abaixo:

```
public class MainActivity extends AppCompatActivity {

    private List<Pessoa> pessoalist = new ArrayList<>();
    ArrayAdapter<Pessoa> pessoaArrayAdapter;

    EditText nome, sobrenome, email, senha;
    ListView dadospessoais;

    FirebaseDatabase firebaseDatabase;
    DatabaseReference databaseReference;

    Pessoa pessoaSelecionada;
```


Dentro do `protected void onCreate(Bundle savedInstanceState) {` colocaremos

```
dadosPessoais.setOnItemClickListener(new AdapterView.OnItemClickListener(){
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        pessoaSelecionada=(Pessoa)parent.getItemAtPosition(position);
        nome.setText(pessoaSelecionada.getNome());
        sobrenome.setText(pessoaSelecionada.getSobrenome());
        email.setText(pessoaSelecionada.getEmail());
        senha.setText(pessoaSelecionada.getSenha());
    }
});
```

abaixo do método `ListarDados();`.

Dentro de `public boolean onOptionsItemSelected(MenuItem item) {` abaixo de

`case R.id.icon_salve:` criaremos um objeto da classe "Pessoa", instanciamos e limpamos os campos:

```
case R.id.icon_salve:
    Pessoa p = new Pessoa();
    p.setUid(pessoaSelecionada.getUid());
    p.setNome(nome.getText().toString().trim());
    p.setSobrenome(sobrenome.getText().toString().trim());
    p.setEmail(email.getText().toString().trim());
    p.setSenha(senha.getText().toString().trim());
    Toast.makeText(context this, text "Salvo", Toast.LENGTH_SHORT).show();
    databaseReference.child("Pessoa").child(p.getUid()).setValue(p);
    LimparCampos();
    break;
```

Abaixo de `case R.id.icon_delete:` colocaremos códigos de tal modo que fique igual a imagem abaixo:

```
case R.id.icon_delete:
    Pessoa pd = new Pessoa();
    po.setUid(pessoaSelecionada.getUid());
    databaseReference.child("Pessoa").child(po.getUid()).removeValue();
    Toast.makeText(context this, text "Deletado", Toast.LENGTH_SHORT).show();
    LimparCampos();
    break;
default:break;
```