AUTENTICAÇÃO DE USUÁRIOS POR MEIO DE REDES SOCIAIS

TIAGO FRAZÃO VALÉRIO

CAIO CANALLI

GUSTAVO

ALEX STOCCO

POR QUE USAR O LOGIN SOCIAL?

USUÁRIO

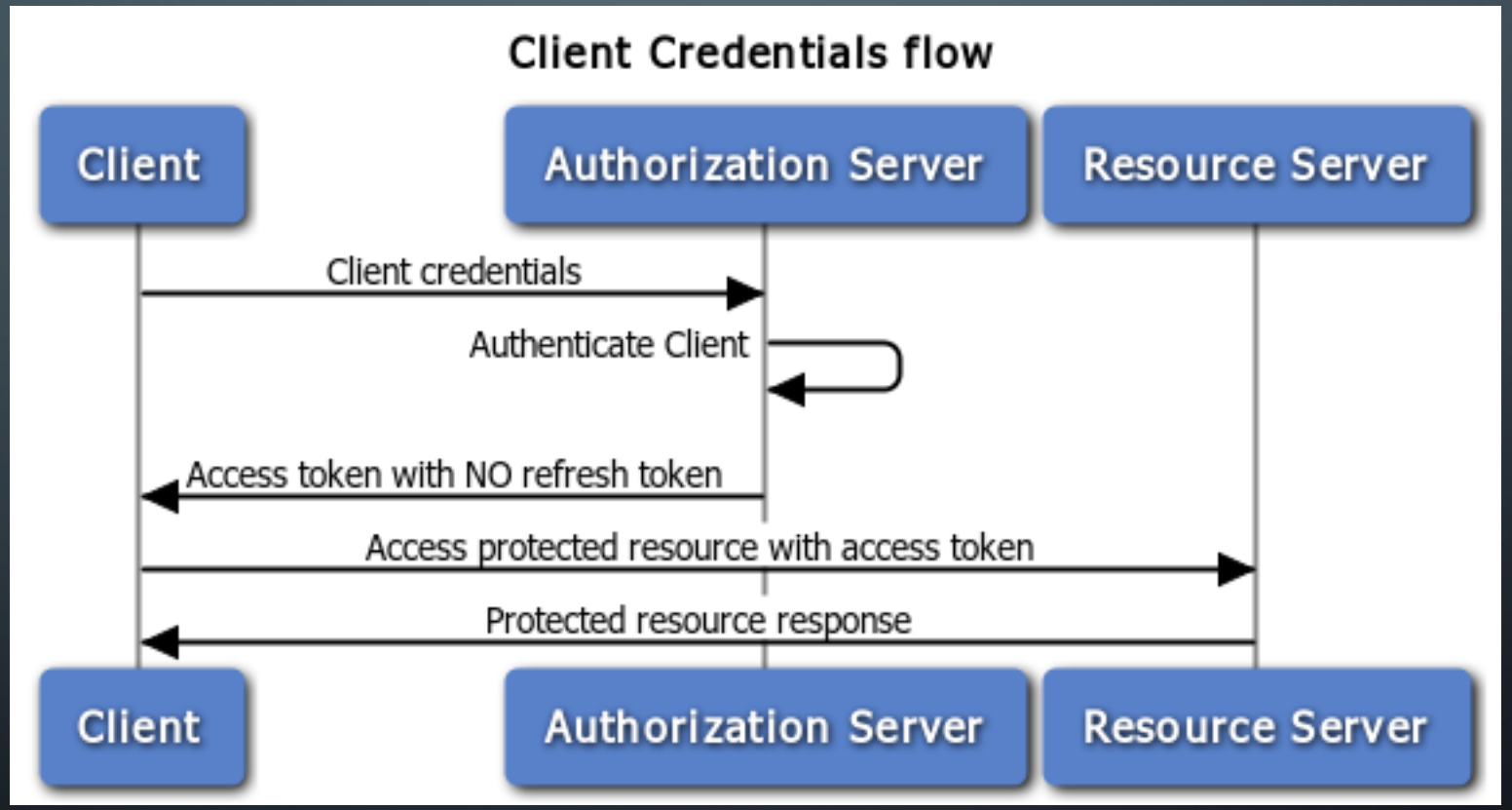
- Acesso rápido
- Elimina a necessidade de preenchimento de formulários
- Utilização de uma única senha

DESENVOLVEDOR

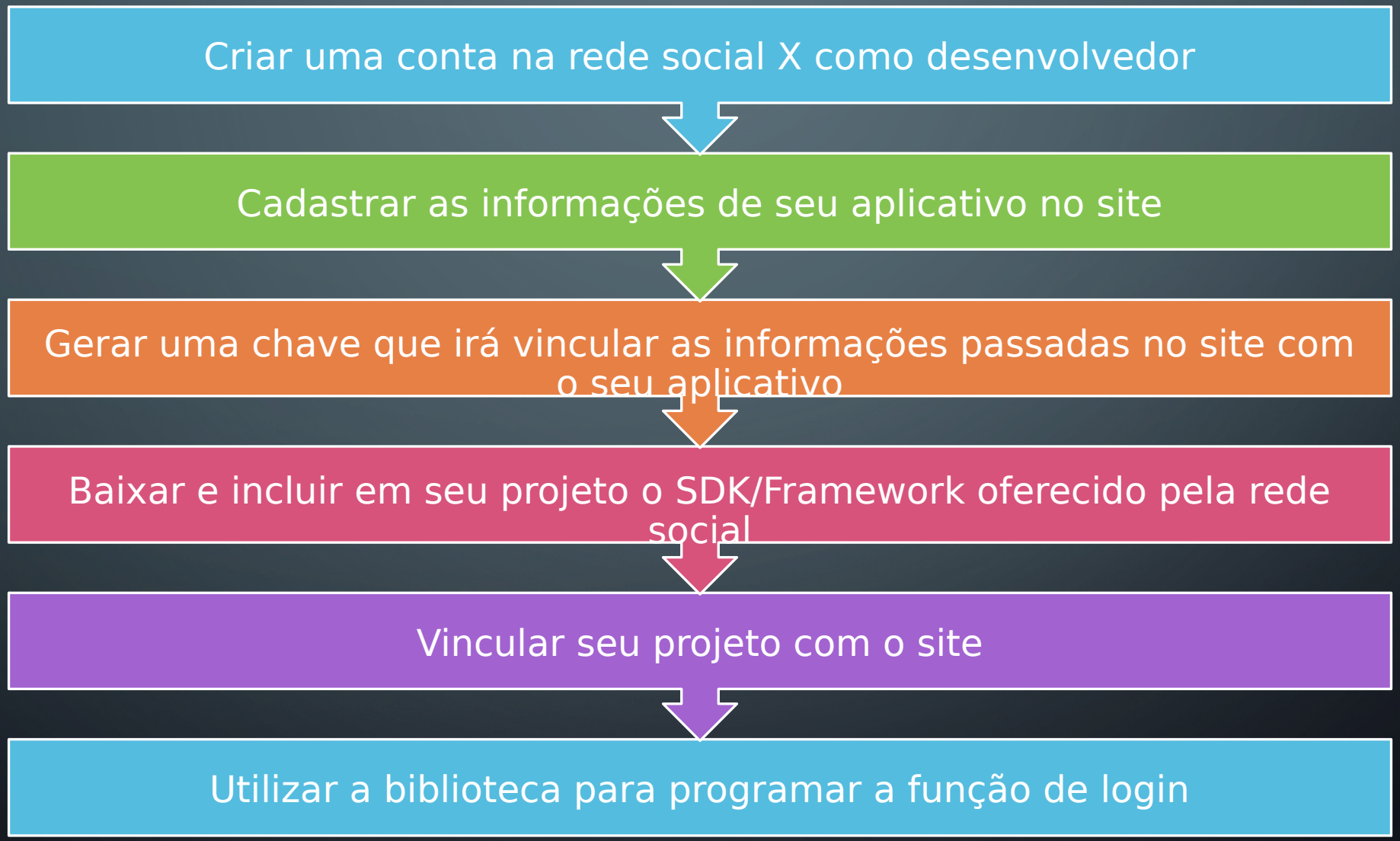
- Aumento na quantidade de cadastros
- Mais veracidade nas informações
- Acesso ao perfil do usuário
- Diminuição da quantidade de cadastros duplicados
- A segurança do login é feita por outra empresa

OAUTH

- Protocolo padrão de autorização

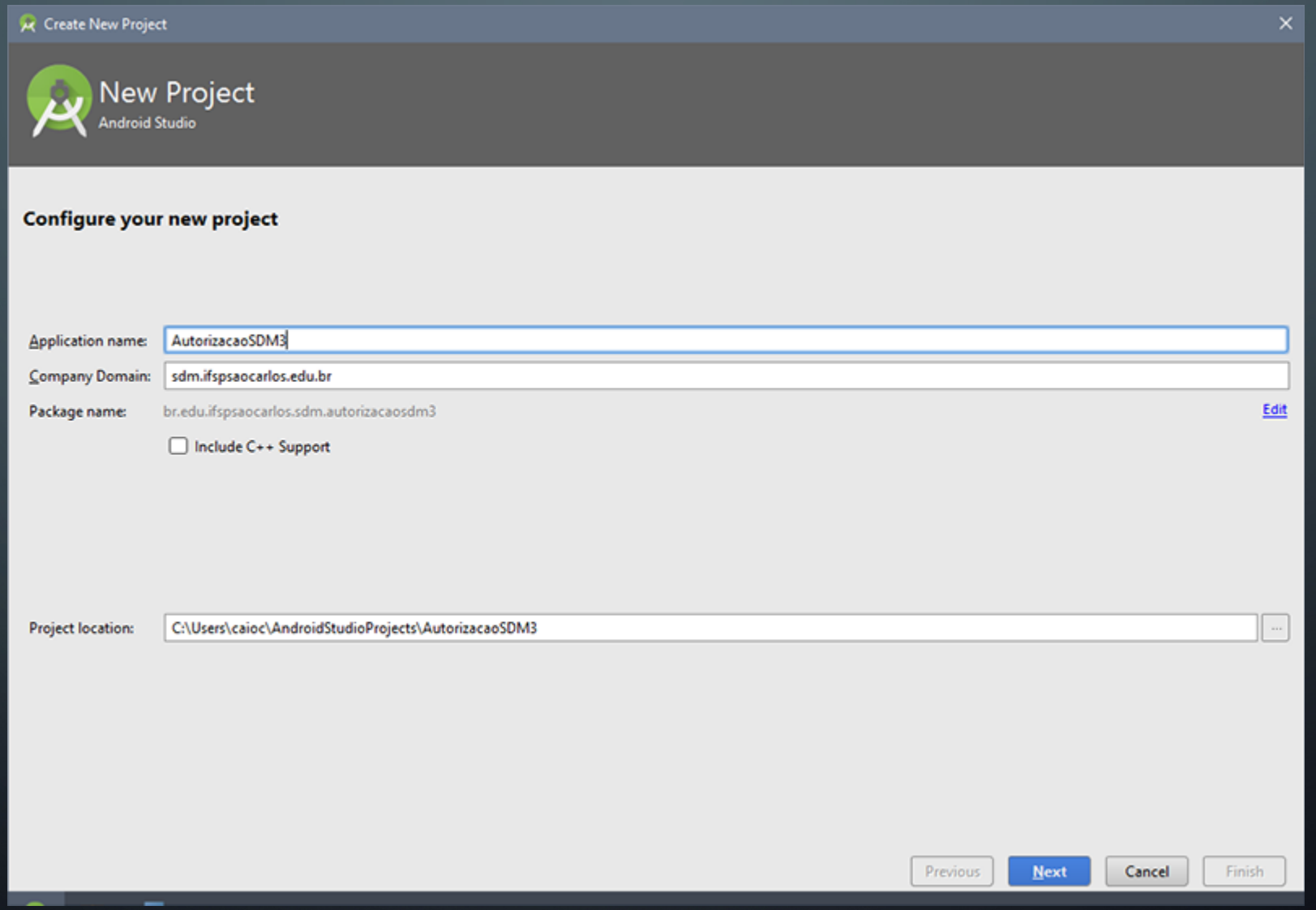


RESUMO DO PASSO-A-PASSO

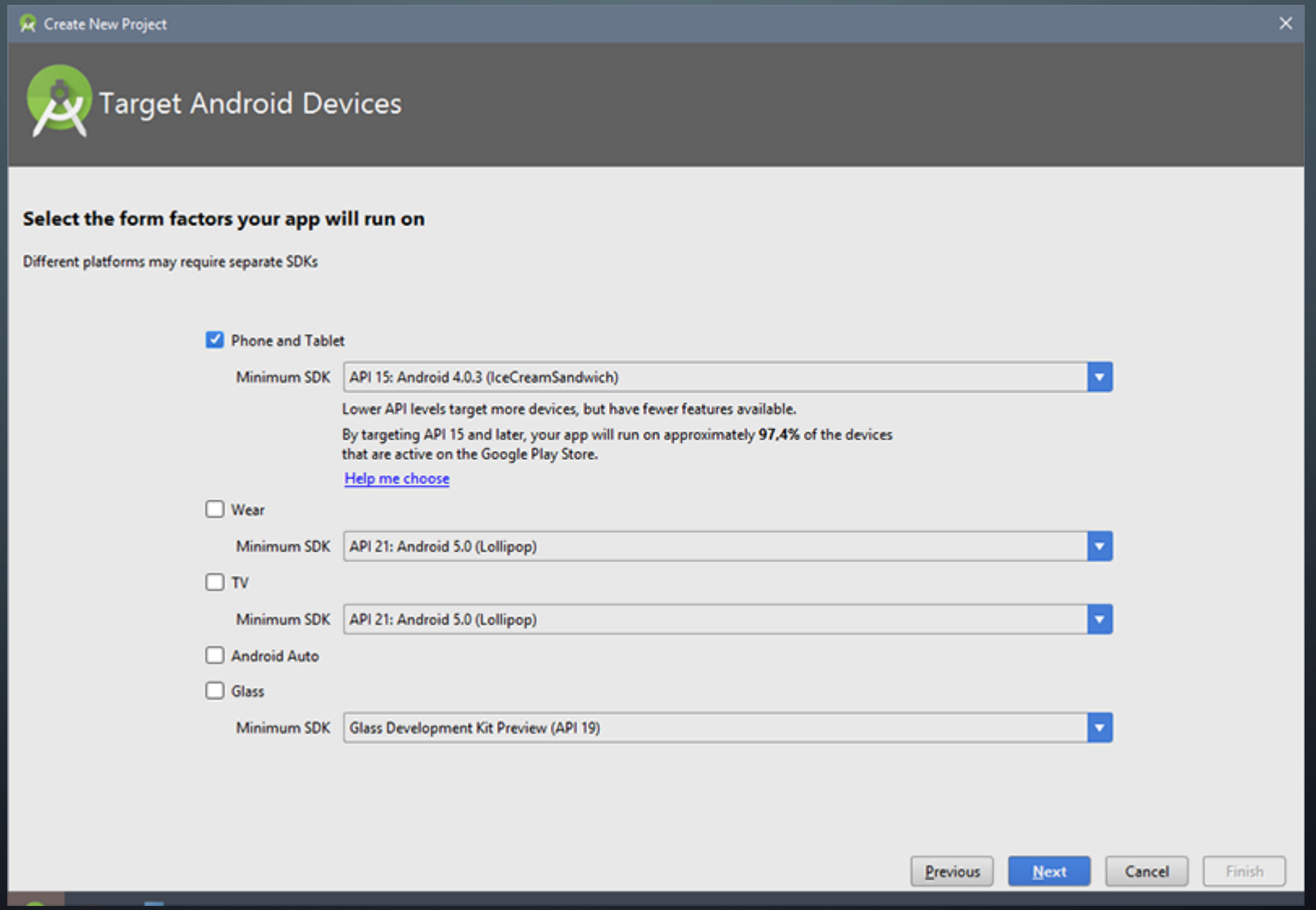


FACEBOOK LOGIN

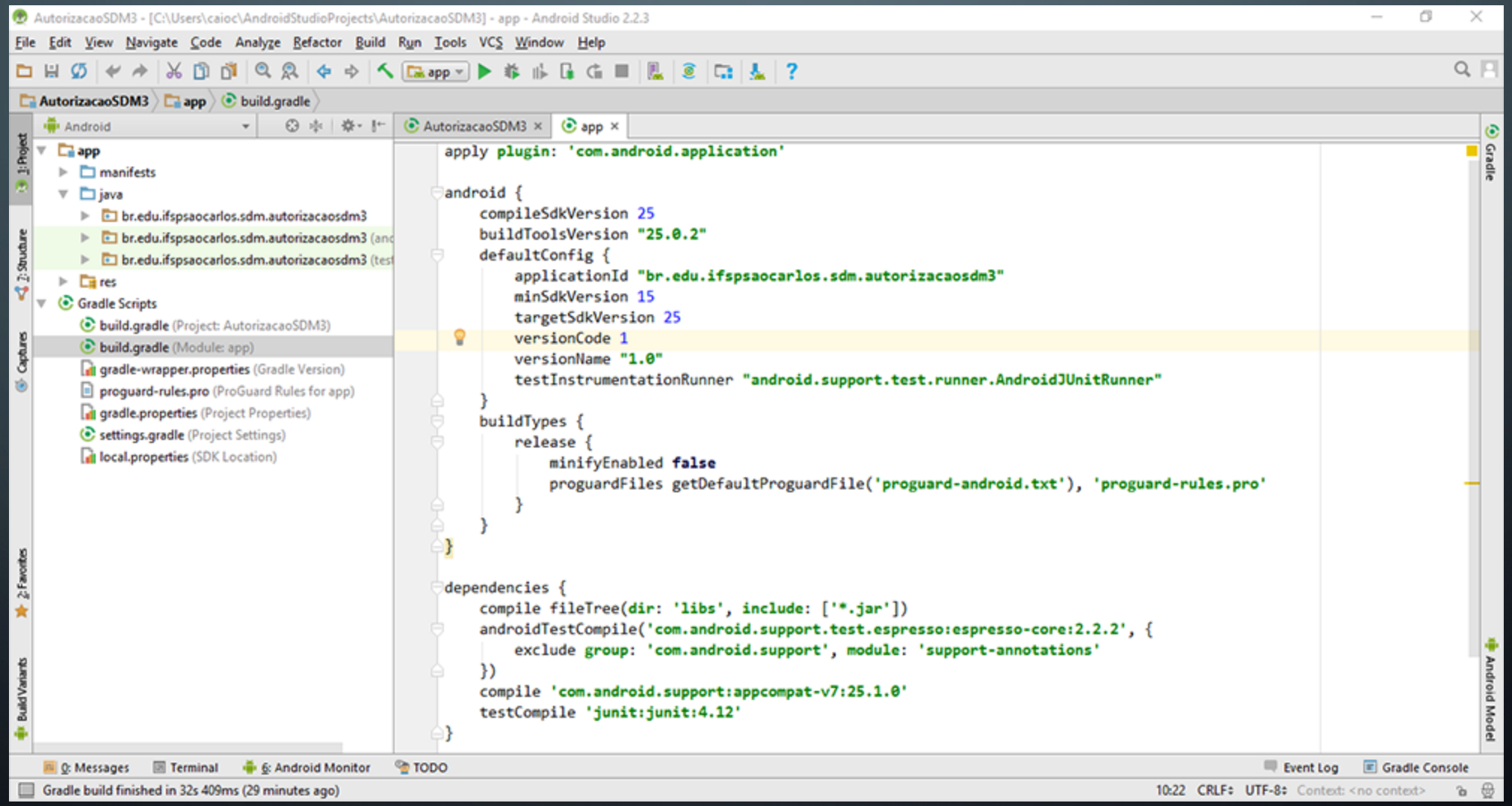
Criando o projeto



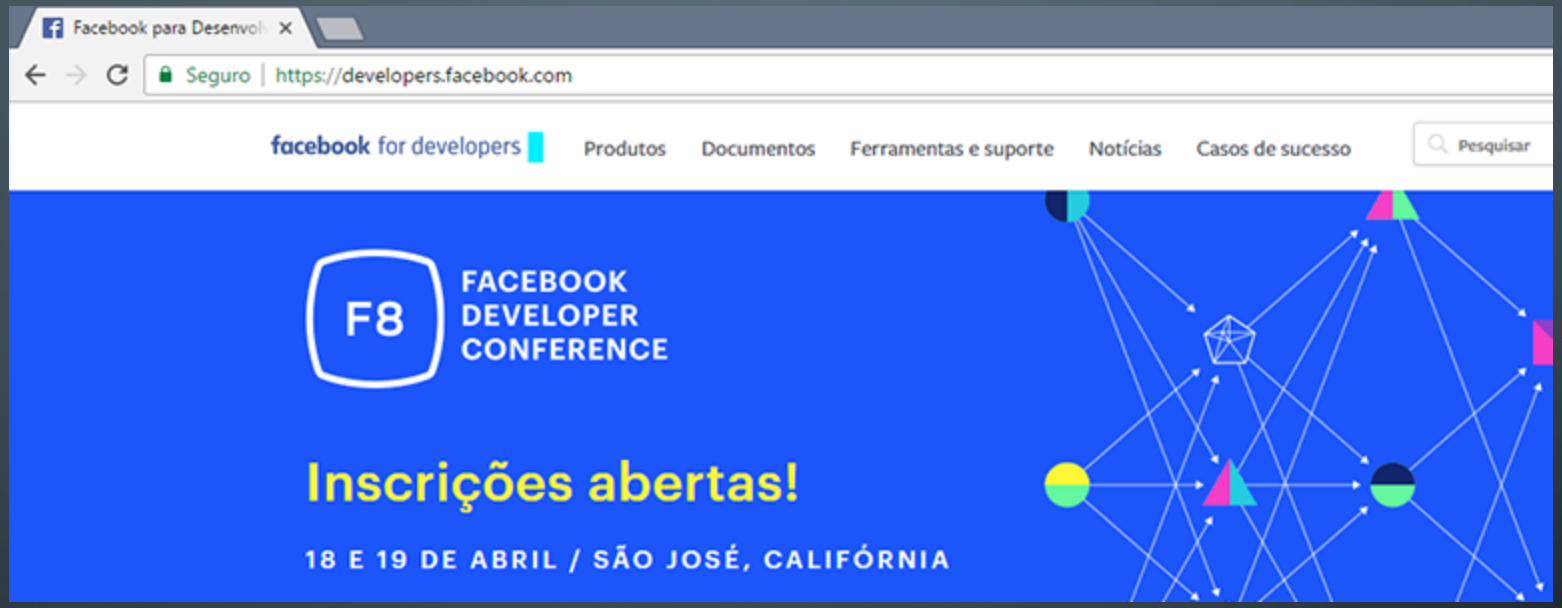
API Level


Create New Project
Target Android Devices
Select the form factors your app will run on
Different platforms may require separate SDKs
Phone and Tablet
Minimum SDK
API 15: Android 4.0.3 (IceCreamSandwich)
Lower API levels target more devices, but have fewer features available.
By targeting API 15 and later, your app will run on approximately 97,4% of the devices that are active on the Google Play Store.
Help me choose
Wear
Minimum SDK
API 21: Android 5.0 (Lollipop)
TV
Minimum SDK
API 21: Android 5.0 (Lollipop)
Android Auto
Glass
Minimum SDK
Glass Development Kit Preview (API 19)
Previous
Next
Cancel
Finish

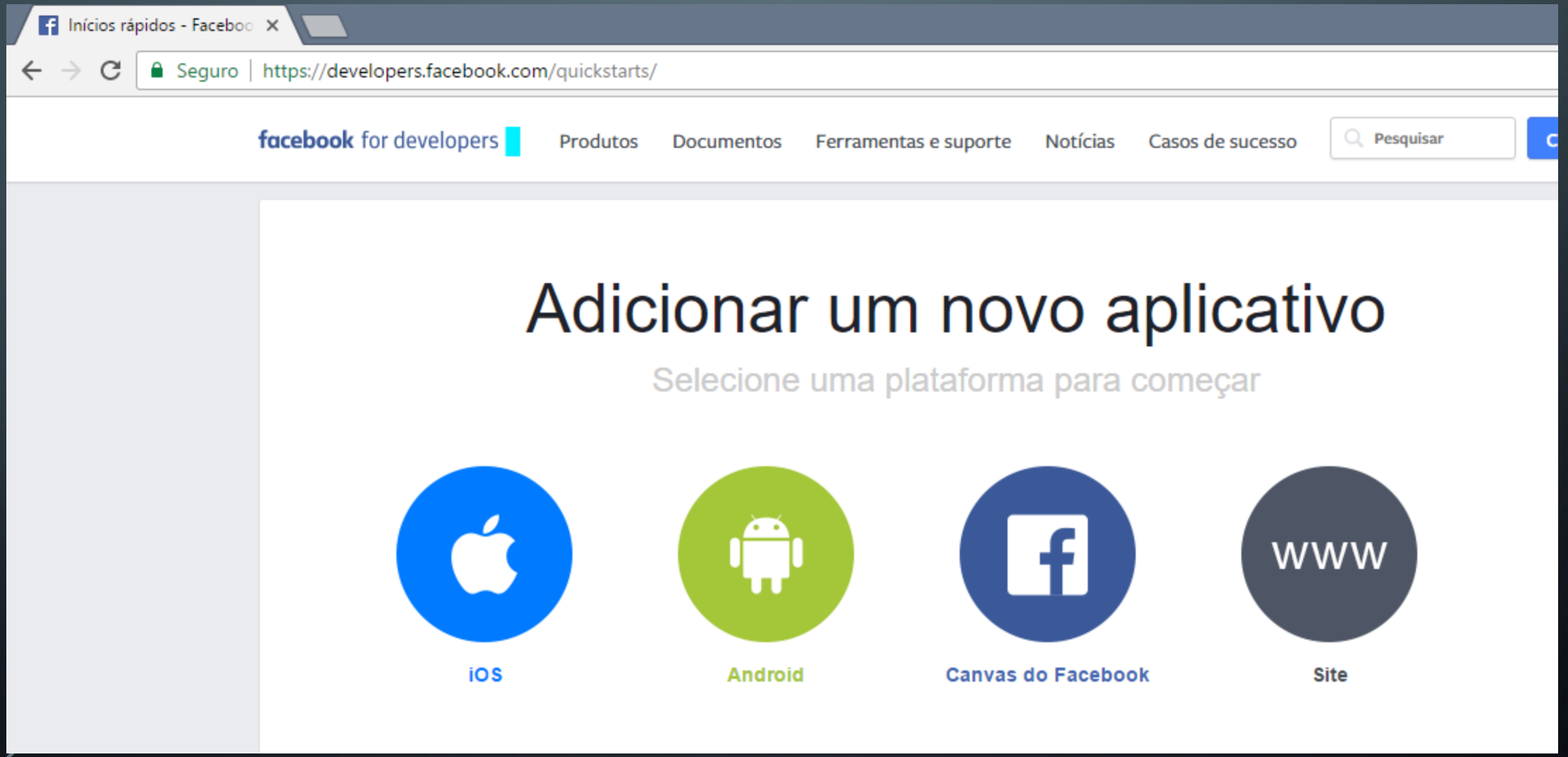
Versão mínima do SDK



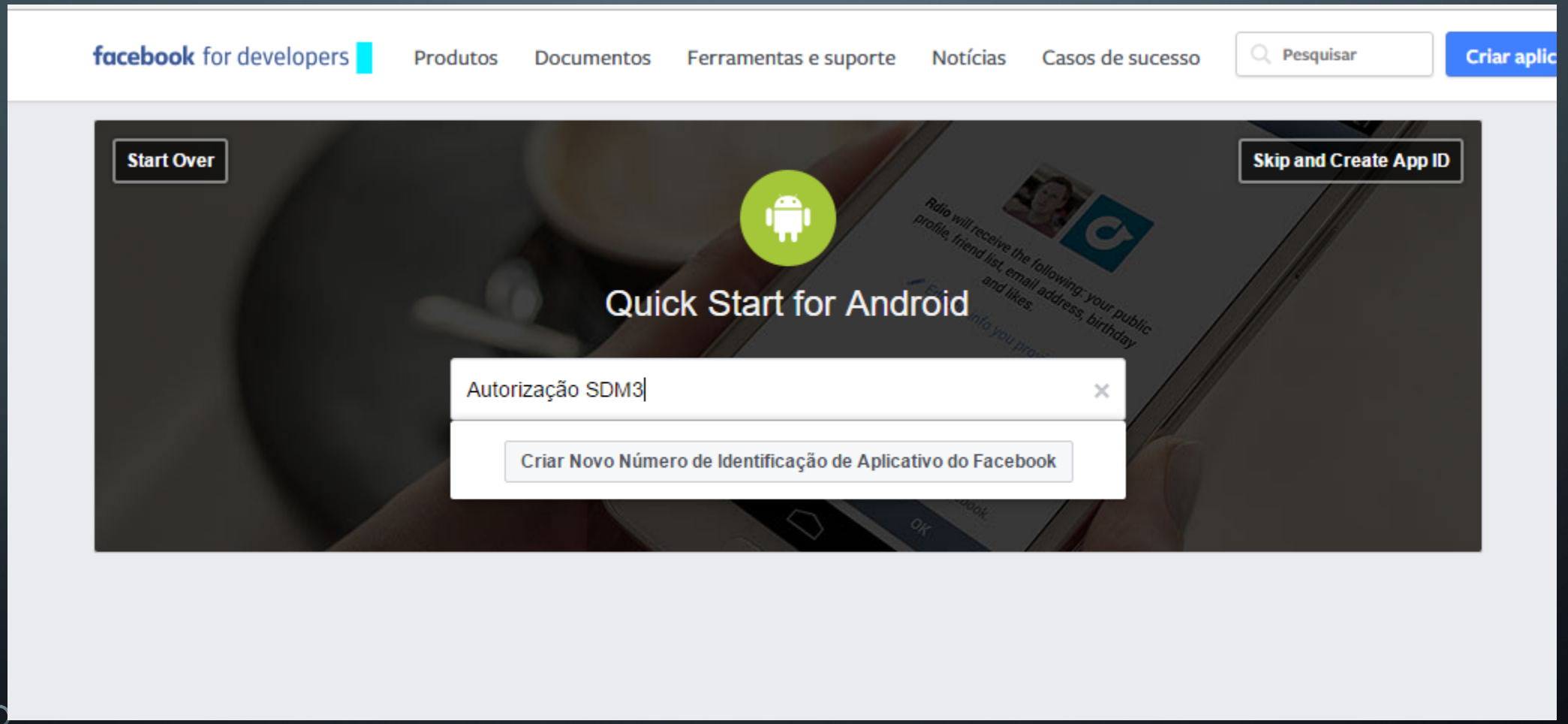
Facebook for developers



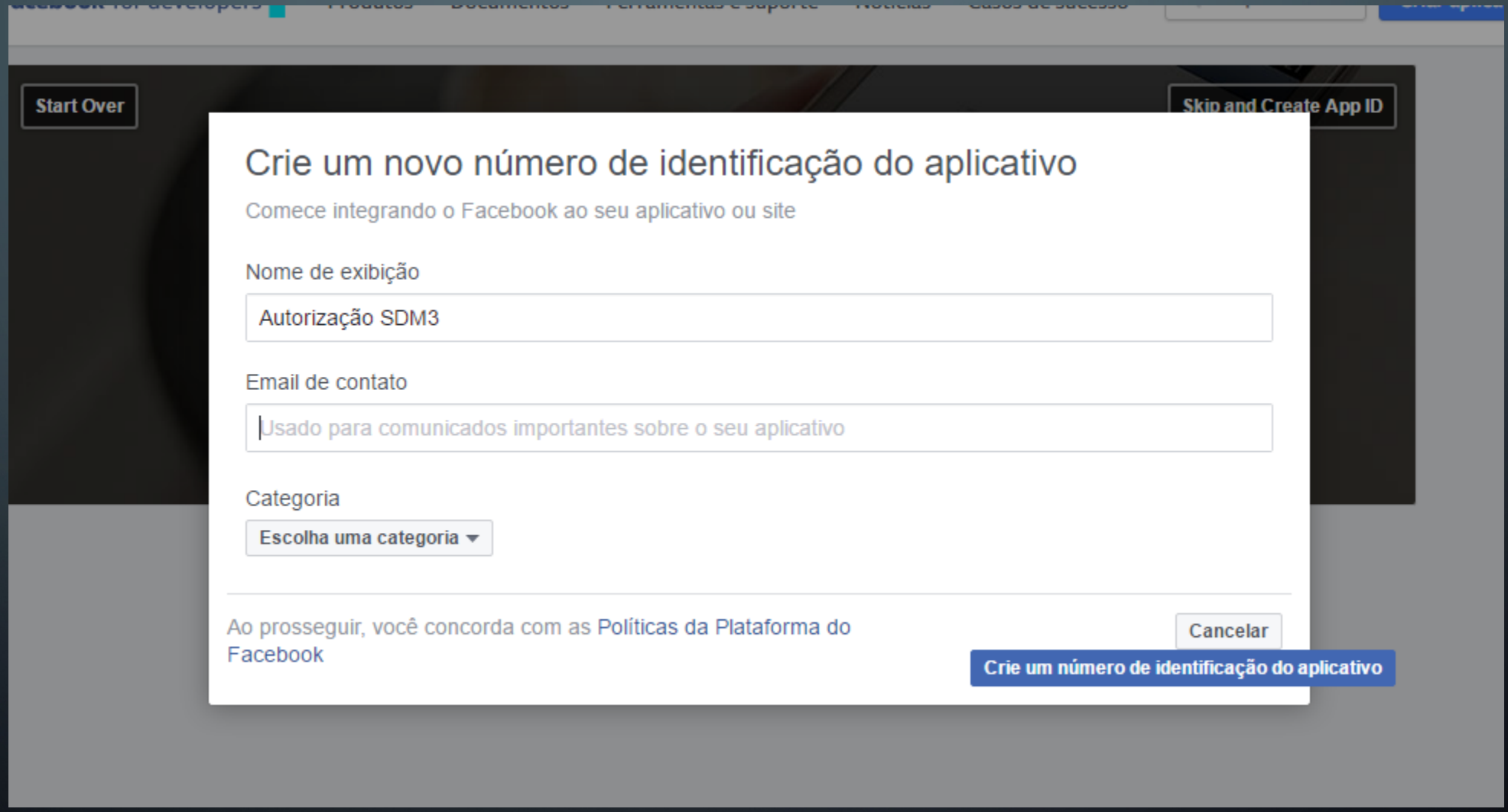
Quickstarts



Nome do Aplicativo



Informações básicas



Importanto o SDK

Import SDK — Add SDK — App Info — Key Hashes — App Events — Finished

Add Facebook SDK to Your Project

To use Facebook SDK in a project, add it as a build dependency and import it. If you are starting a new project, follow all the steps below. To add Facebook SDK to an existing project, start with step 3.

1. Go to **Android Studio | New Project | Minimum SDK**
2. Select "**API 15: Android 4.0.3**" or higher and create your new project.
3. In your project, open

your_app | Gradle Scripts | build.gradle

4. Add the Maven Central Repository to **build.gradle** before **dependencies:**

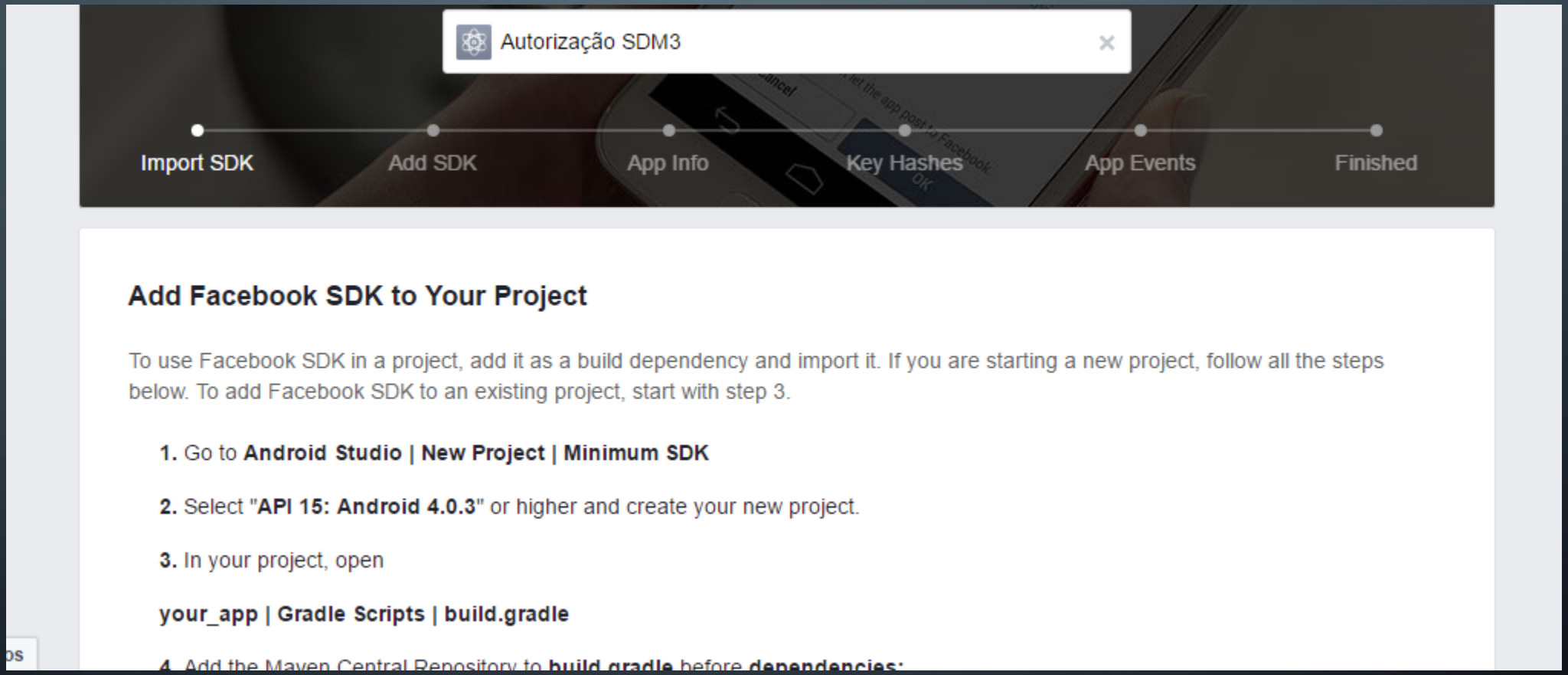
```
repositories {
        mavenCentral()
    }
```

5. Add **compile 'com.facebook.android:facebook-android-sdk:[4,5)'** to your **build.gradle** dependencies.
6. Build your project.
7. Import Facebook SDK into your app:

```
import com.facebook.FacebookSdk;
```

entários

Reforçando


Autorização SDM3
Import SDK
Add SDK
App Info
Key Hashes
App Events
Finished
Add Facebook SDK to Your Project
To use Facebook SDK in a project, add it as a build dependency and import it. If you are starting a new project, follow all the steps below. To add Facebook SDK to an existing project, start with step 3.
1. Go to Android Studio | New Project | Minimum SDK
2. Select "API 15: Android 4.0.3" or higher and create your new project.
3. In your project, open
your_app | Gradle Scripts | build.gradle

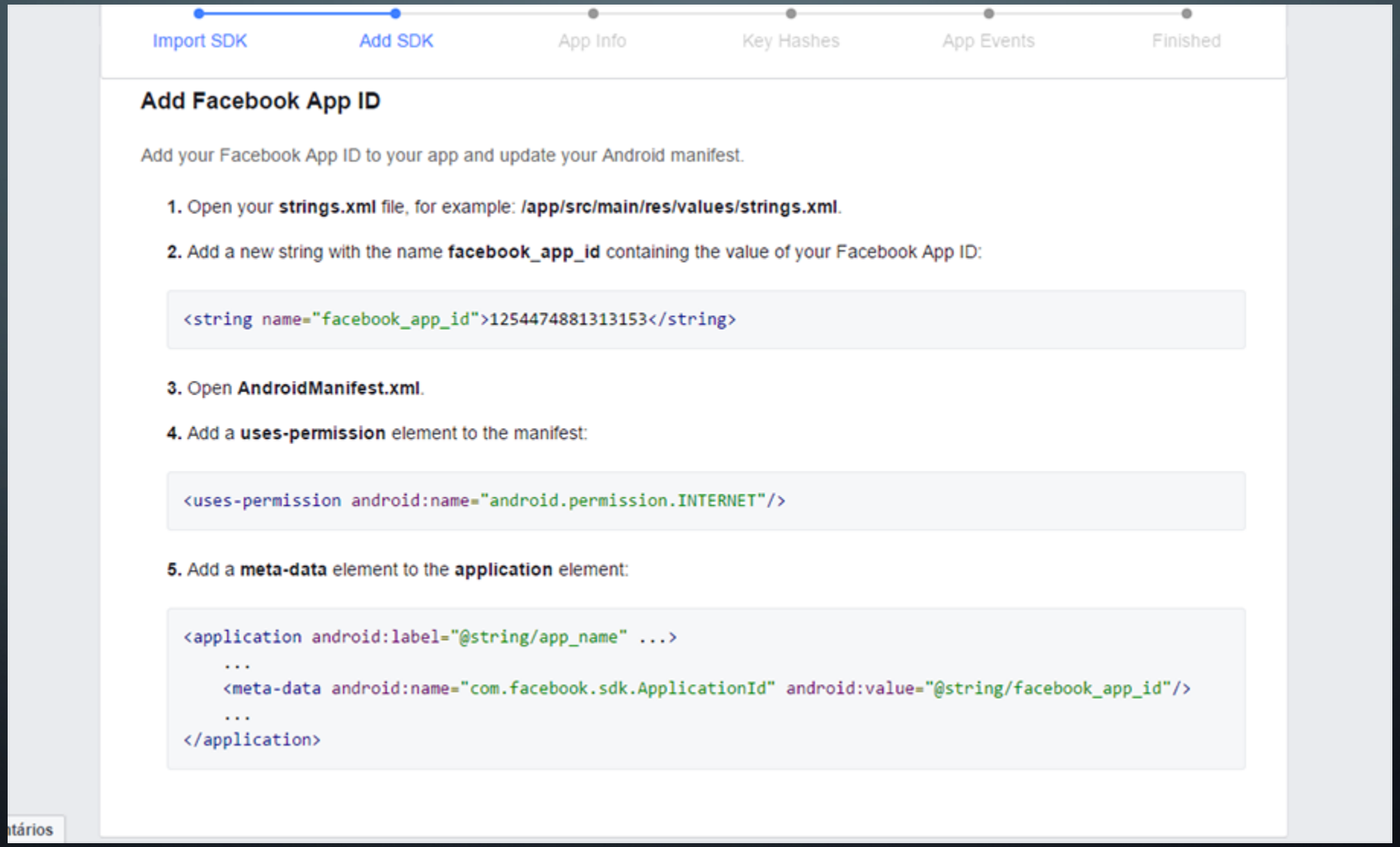
Configurando o SDK

AutorizacaoSDM3 × app ×

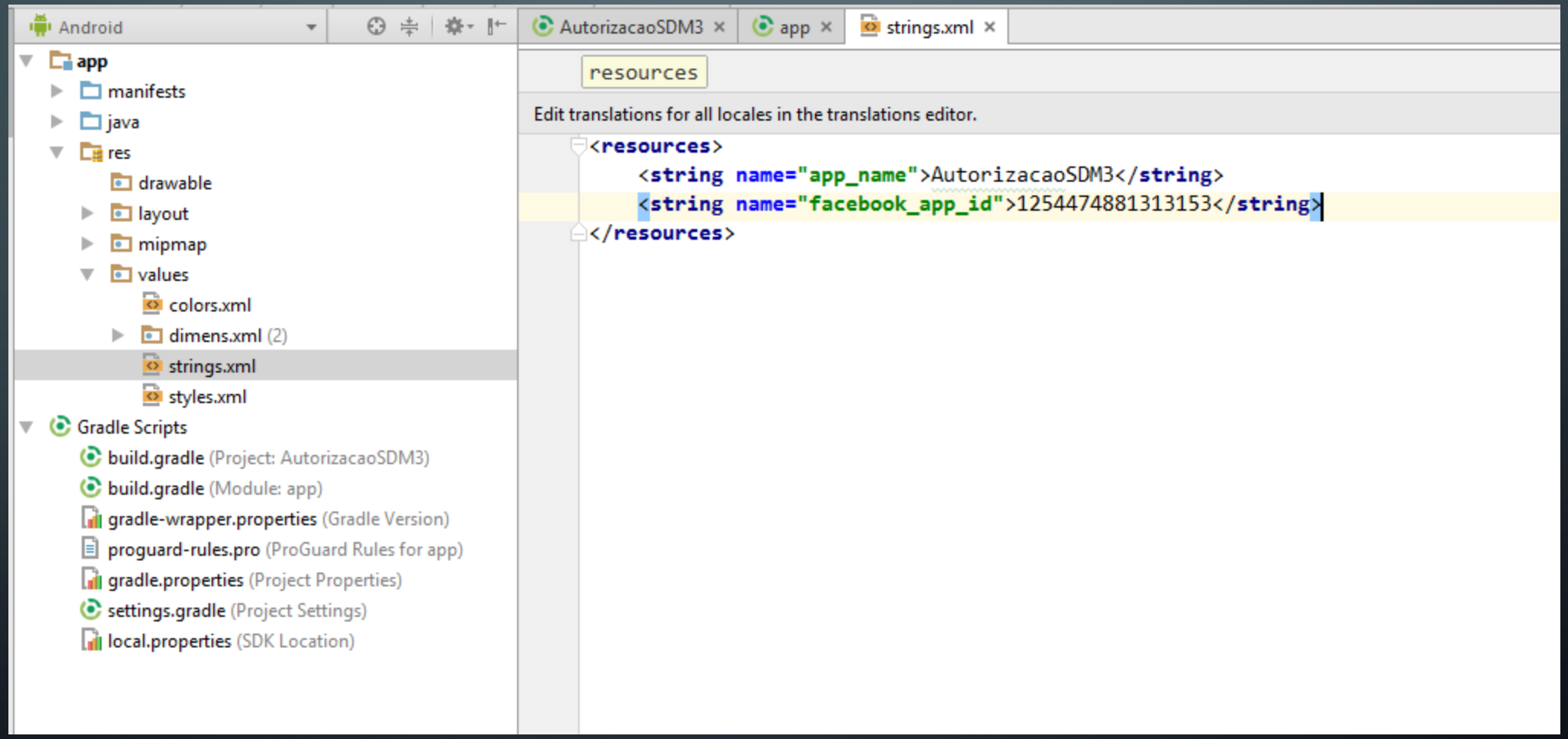
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
        applicationId "br.edu.ifspsaocarlos.sdm.autorizacaosdm3"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.0'
    compile 'com.facebook.android:facebook-android-sdk:[4,5)'

    testCompile 'junit:junit:4.12'
}
```

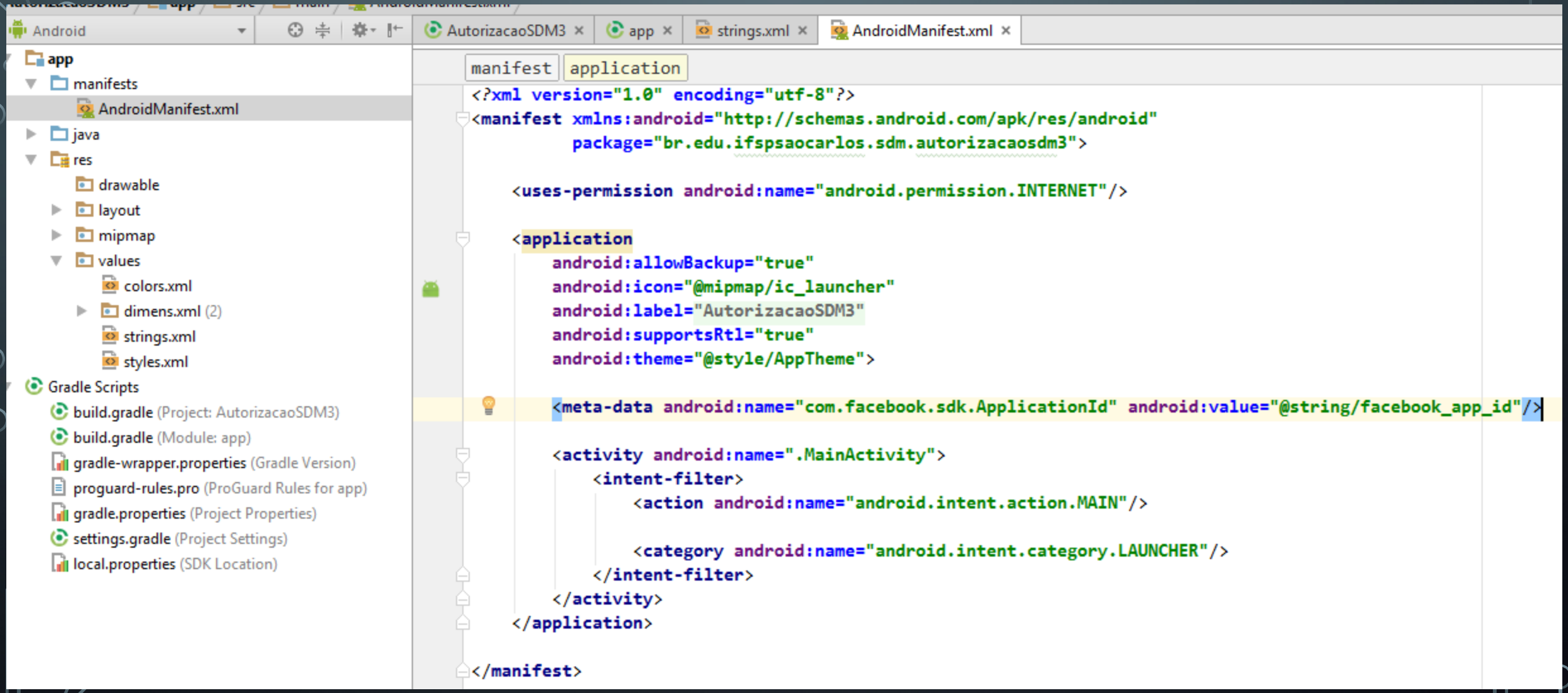
Configurando algumas coisas



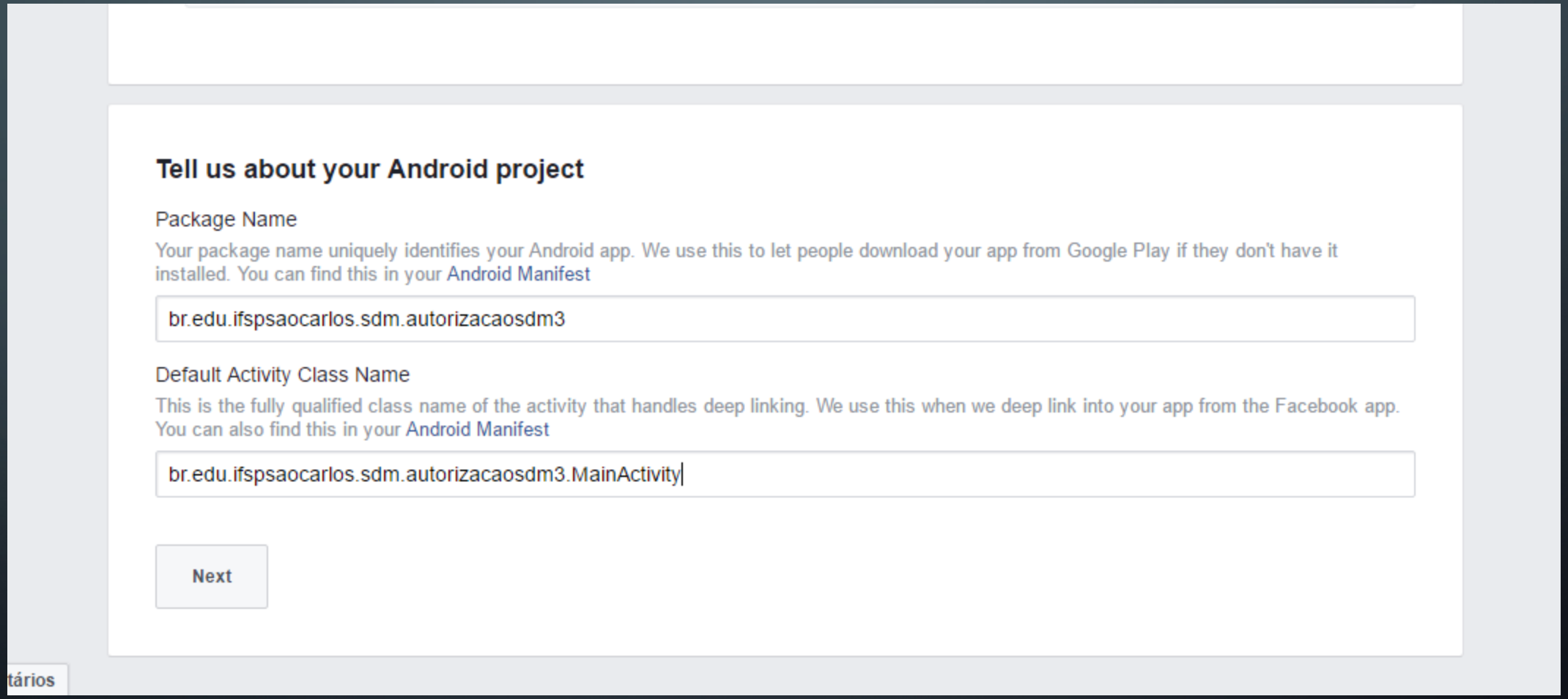
O arquivo strings.xml



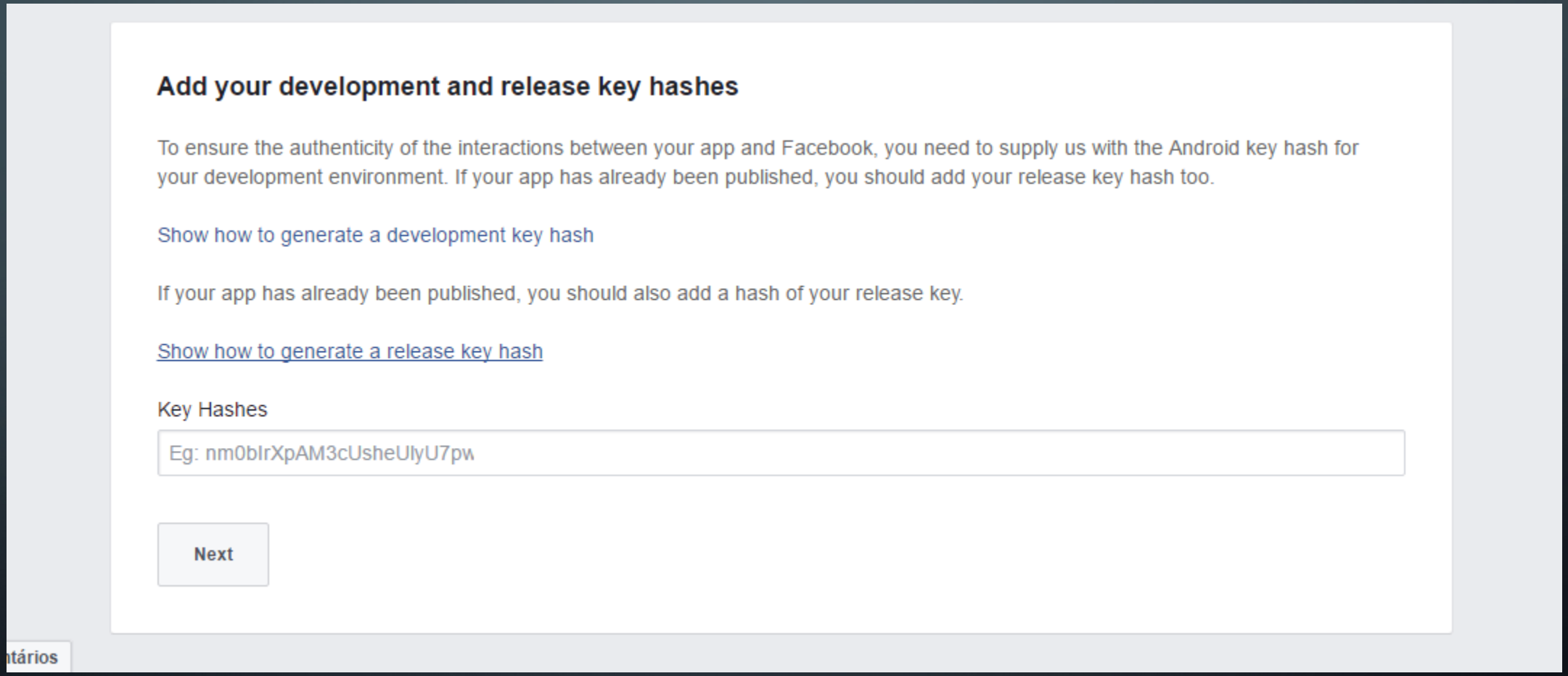
O arquivo AndroidManifest.xml



Nome do pacote e Activity principal



Adicionar a chave do aplicativo



Como gerar e um exemplo de chave

```
keytool -exportcert -alias androiddebugkey -keystore ~/.android/debug.keystore | openssl sha1 -binary | openss
```

On Windows, run this command:

```
keytool -exportcert -alias androiddebugkey -keystore %HOMEPATH%\.android\debug.keystore | openssl sha1 -binary
```

This command will generate a 28-character key hash unique to your development environment. Copy and paste it into the field below. You will need to provide a development key hash for the development environment of each person who works on your app.

If your app has already been published, you should also add a hash of your release key.

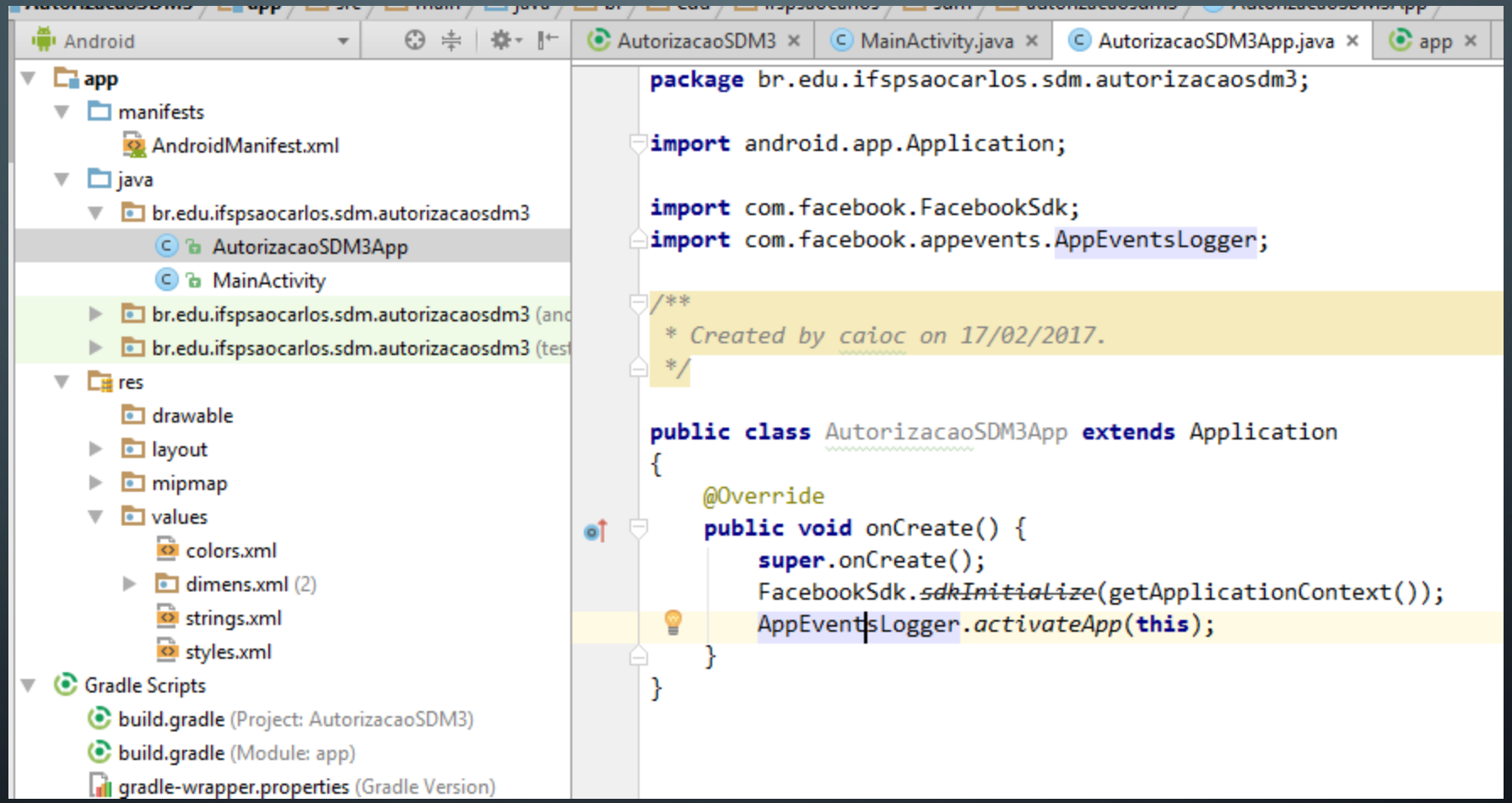
Show how to generate a release key hash

Key Hashes

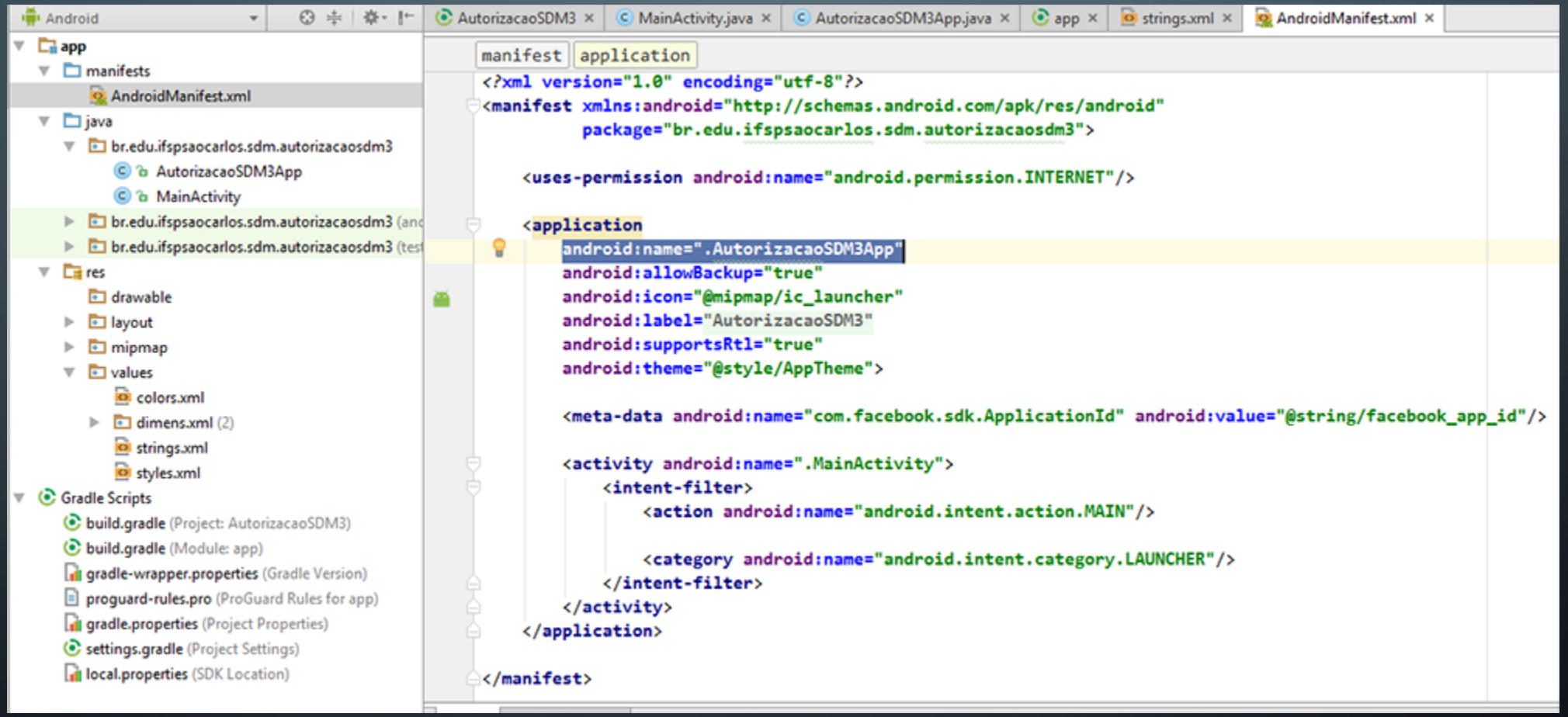
18gO9G2V439XD78HEgL66mkTm28= ×

Next

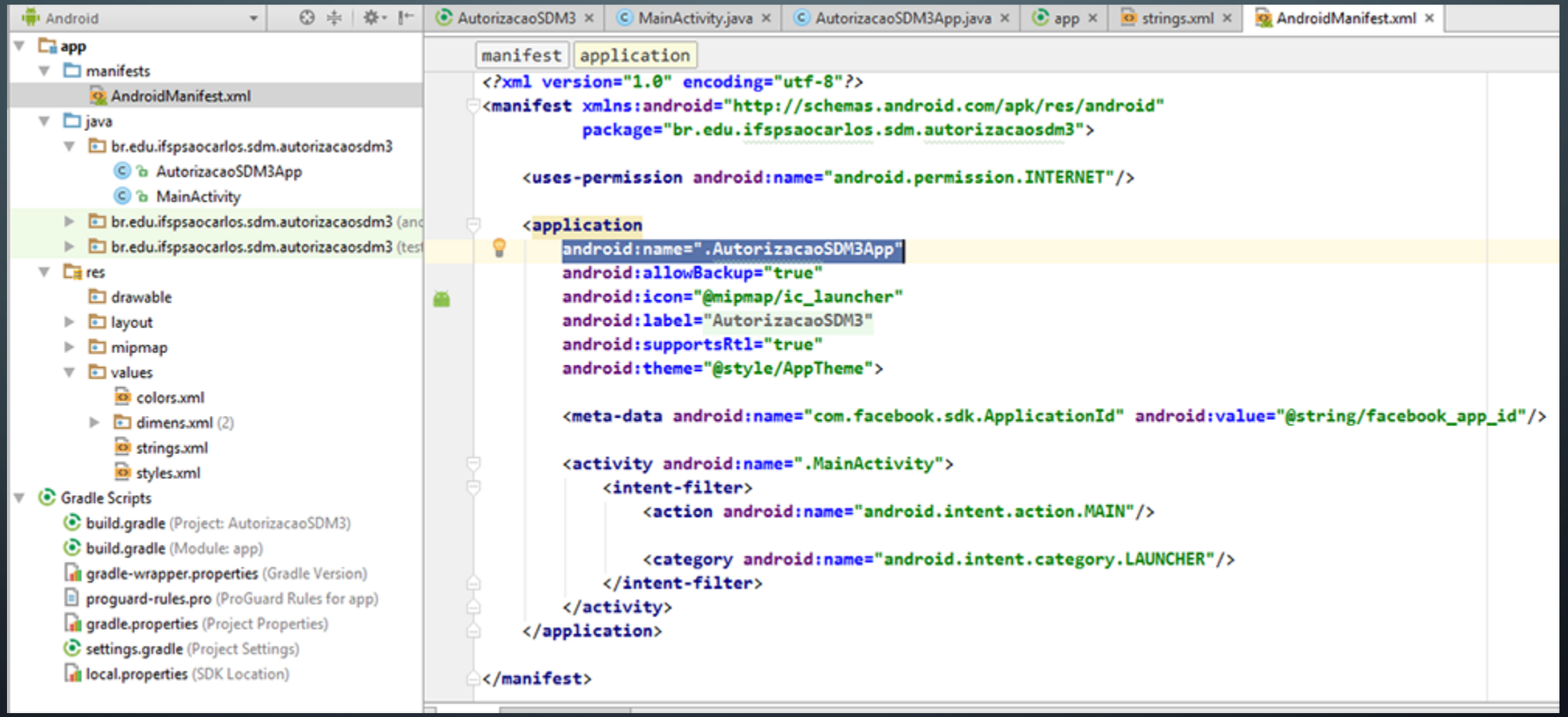
Inicializar o Facebook SDK



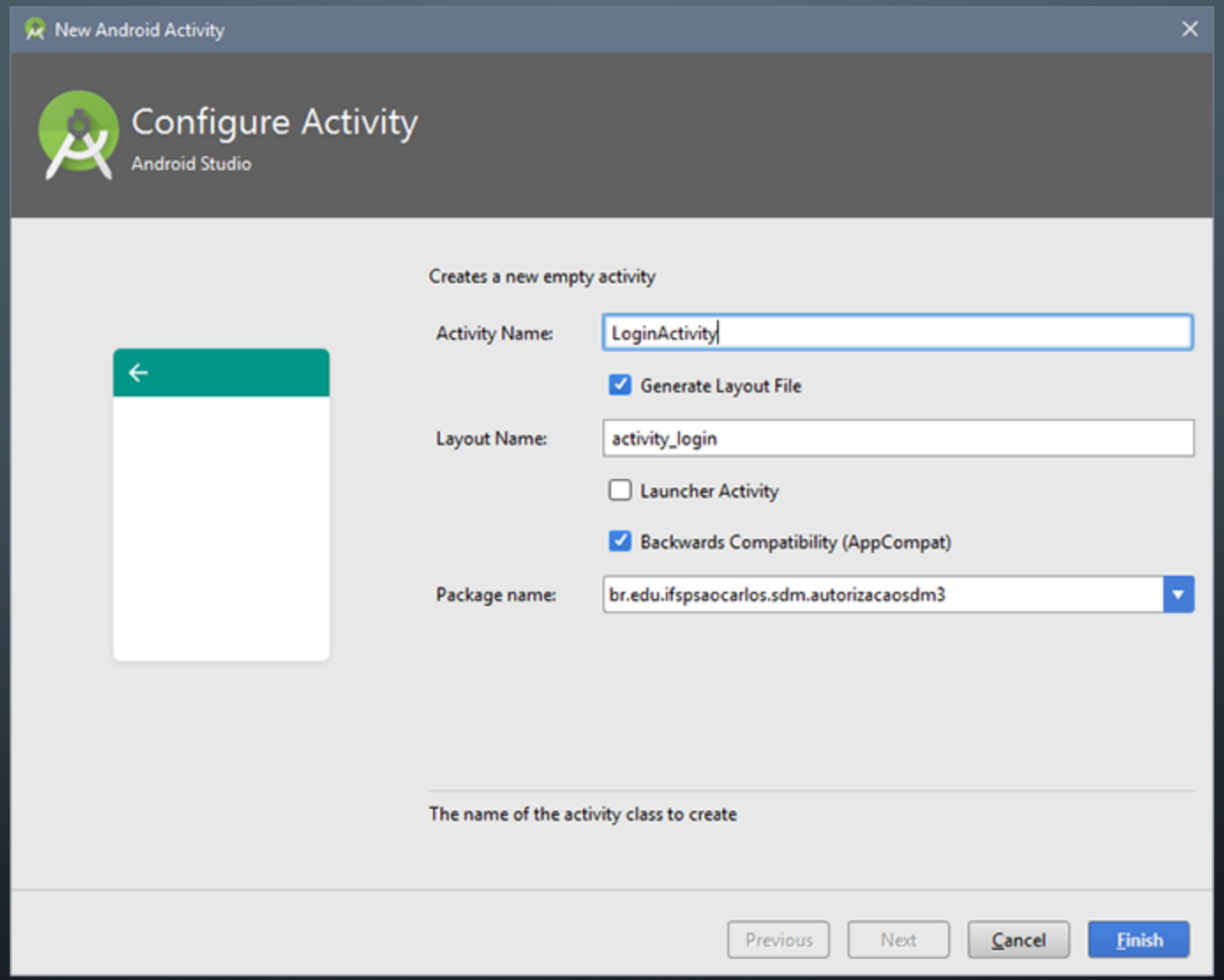
Configurar a classe que inicia o SDK



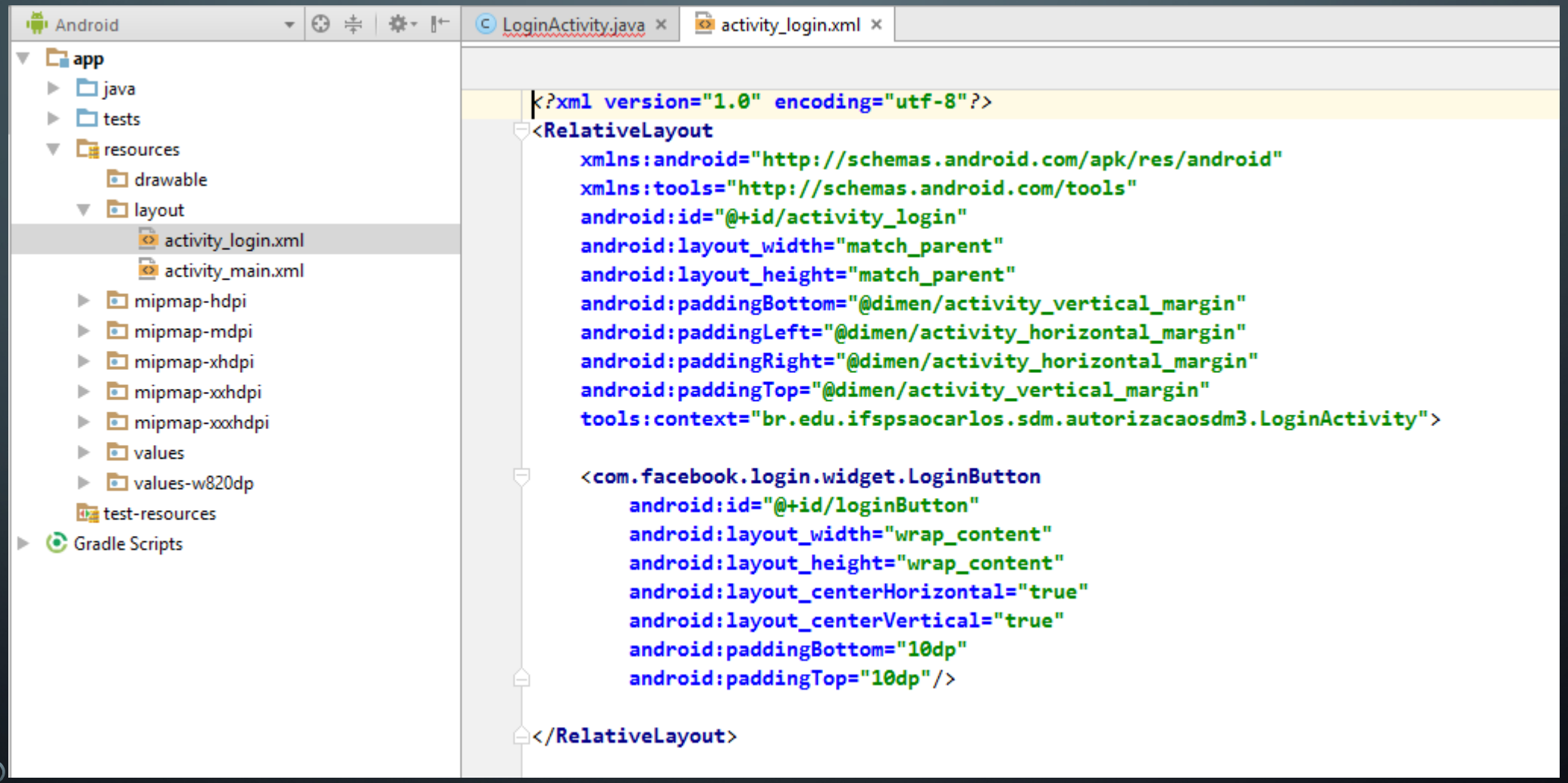
Configurar a classe que inicia o SDK



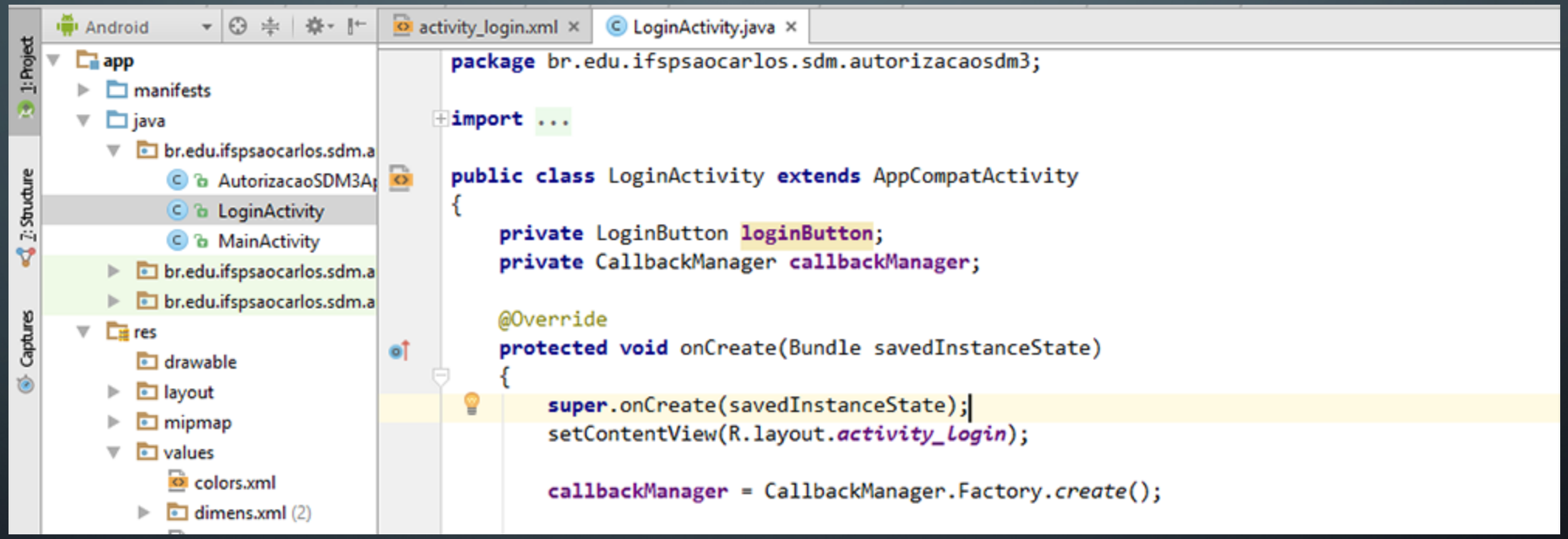
Nova Activity: LoginActivity



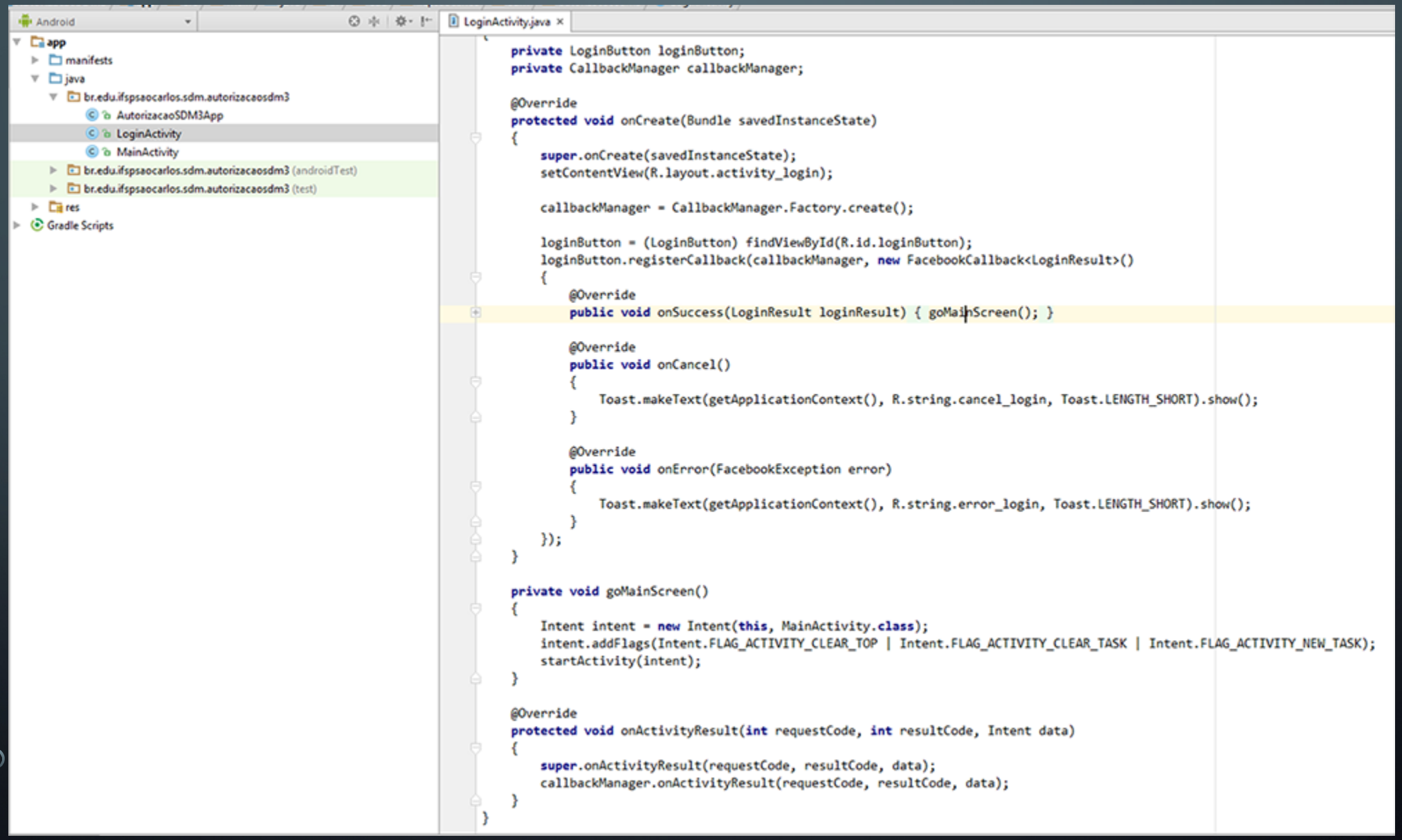
O layout de LoginActivity



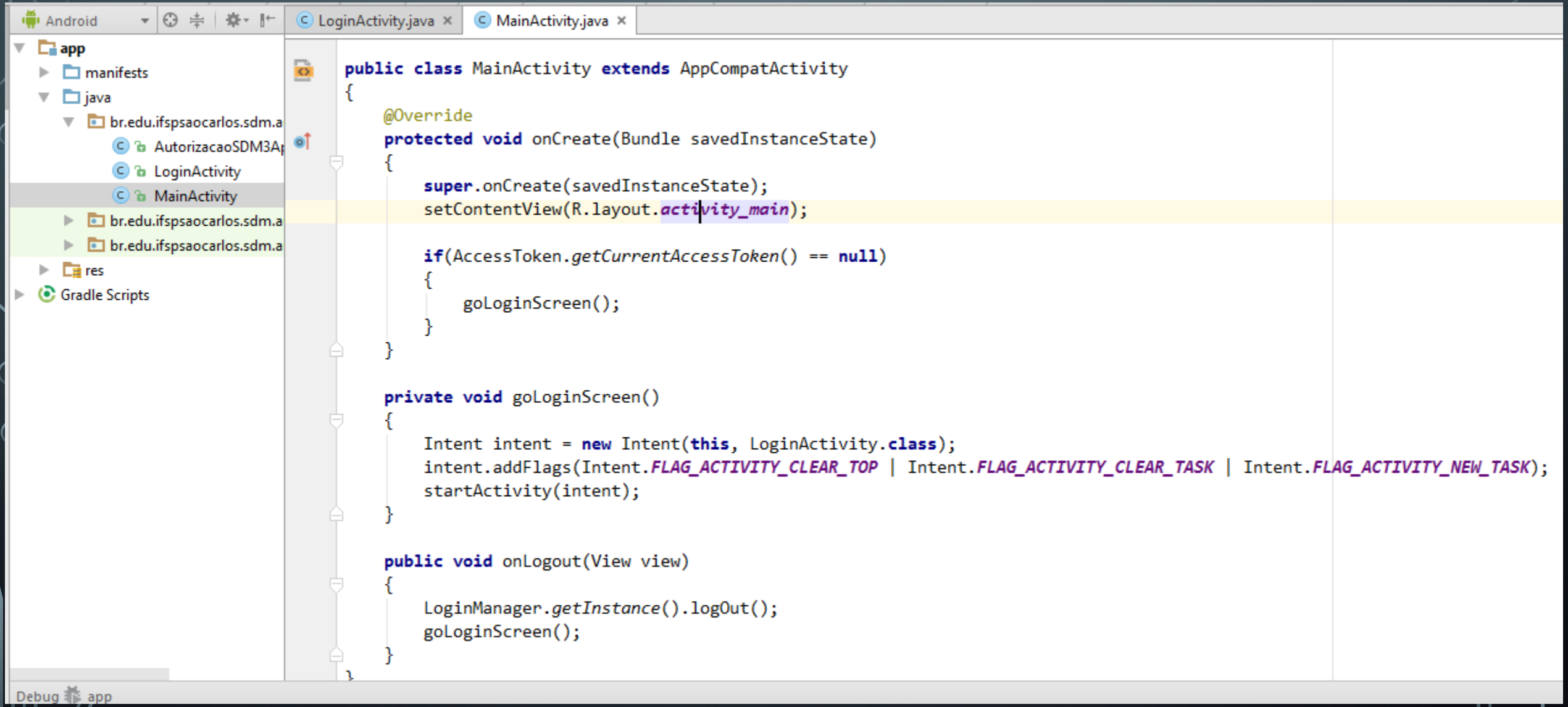
Configurando o login



Configurando o login



Finalizando a configuração



GOOGLE SIGN-IN

Tutorial de Implementação do Google Sign-In

Pré-requisitos:

- Versão mínima Android 2.3 (API 9)
- Emulador AVD Android 4.2 (API 17) Google Play Services 9.8.0
- Instalar SDK Manager > Extras > Google Repository

Novo Projeto Android Studio:

- Empty Activity
- Obter arquivo de configuração:
  https://developers.google.com/identity/sign-in/android/

Add Google Sign-In to Your Android App

Configure Google Sign-In:

```
// Configure sign-in to request the user's ID, email address, and basic profile. ID and
// basic profile are included in DEFAULT_SIGN_IN.
GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestEmail()
    .build();

// Build a GoogleApiClient with access to GoogleSignIn.API and the options above.
mGoogleApiClient = new GoogleApiClient.Builder(this)
    .enableAutoManage(this, this)
    .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
    .build();
```

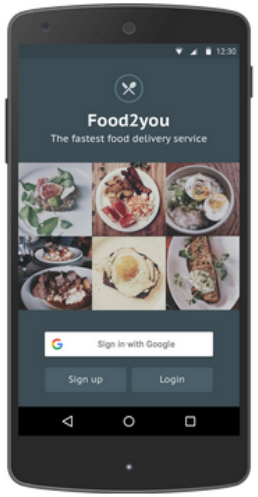
Then, when the sign-in button is clicked, start the sign-in intent:

```
Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
startActivityForResult(signInIntent, RC_SIGN_IN);
```

The user is prompted to select a Google account to sign in with. If you requested scopes beyond `profile`, `email`, and `openid`, the user is also prompted to grant access to the requested resources.

Finally, handle the activity result:

```
@Override public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Result returned from launching the Intent from
    //   GoogleSignInApi.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        if (result.isSuccess()) {
            GoogleSignInAccount acct = result.getSignInAccount();
            // Get account information
            mFullName = acct.getDisplayName();
            mEmail = acct.getEmail();
        }
    }
}
```



Ready to integrate Google Sign-In into your app?

GET STARTED

Try Sign-In for Android

Use our Android sample app to see how Sign-In works, or add Sign-In to your existing app.

Required: The latest versions of Android Studio and Google Play Services.

1 Get the project

If this is your first time using a Google services sample, check out the google-services repository.

```
$ git clone https://github.com/googlesamples/google-services.git
```

Open Android Studio.

Select **File > Open**, browse to where you cloned the `google-services` repository, and open `google-services/android/signin`.

2 Get a configuration file

To use the sample, you need to provide some additional information to get a configuration file and finish setting up your project. Use the package name `com.google.samples.quickstart.signin` for the sample.

After you complete the registration, download the `google-services.json` file to add to your project.

GET A CONFIGURATION FILE

https://developers.google.com/mobile/add?platform=android&cntapi=signin&cnturl=https:%2F%2Fdevelopers.google.c

Pesquisar

Google Developers

Pesquisa

Todos os produtos

Enable Google services for your app

gustavo.rocha.saocarlos@gmail.com
Sign in as another user

- ✓ Android
- **Create or choose app**
- Choose services
- Download config files

Firebase
Powerful new tools from Google for mobile developers.
LEARN MORE →

Create or choose an app

App name

Android package name

CONTINUE TO
Choose and configure services →

https://developers.google.com/mobile/add?platform=android&cntapi=signin&cnturl=https:%2F%2Fdevelopers.google.c

Pesquisar

Google Developers

Pesquisa

Todos os produtos

Enable Google services for your app

gustavo.rocha.saocarlos@gmail.com
Sign in as another user

- ✓ Android
- Create or choose app
- Choose services
- Download config files

Firebase
Powerful new tools from Google for mobile developers.
LEARN MORE →

Create or choose an app

App name

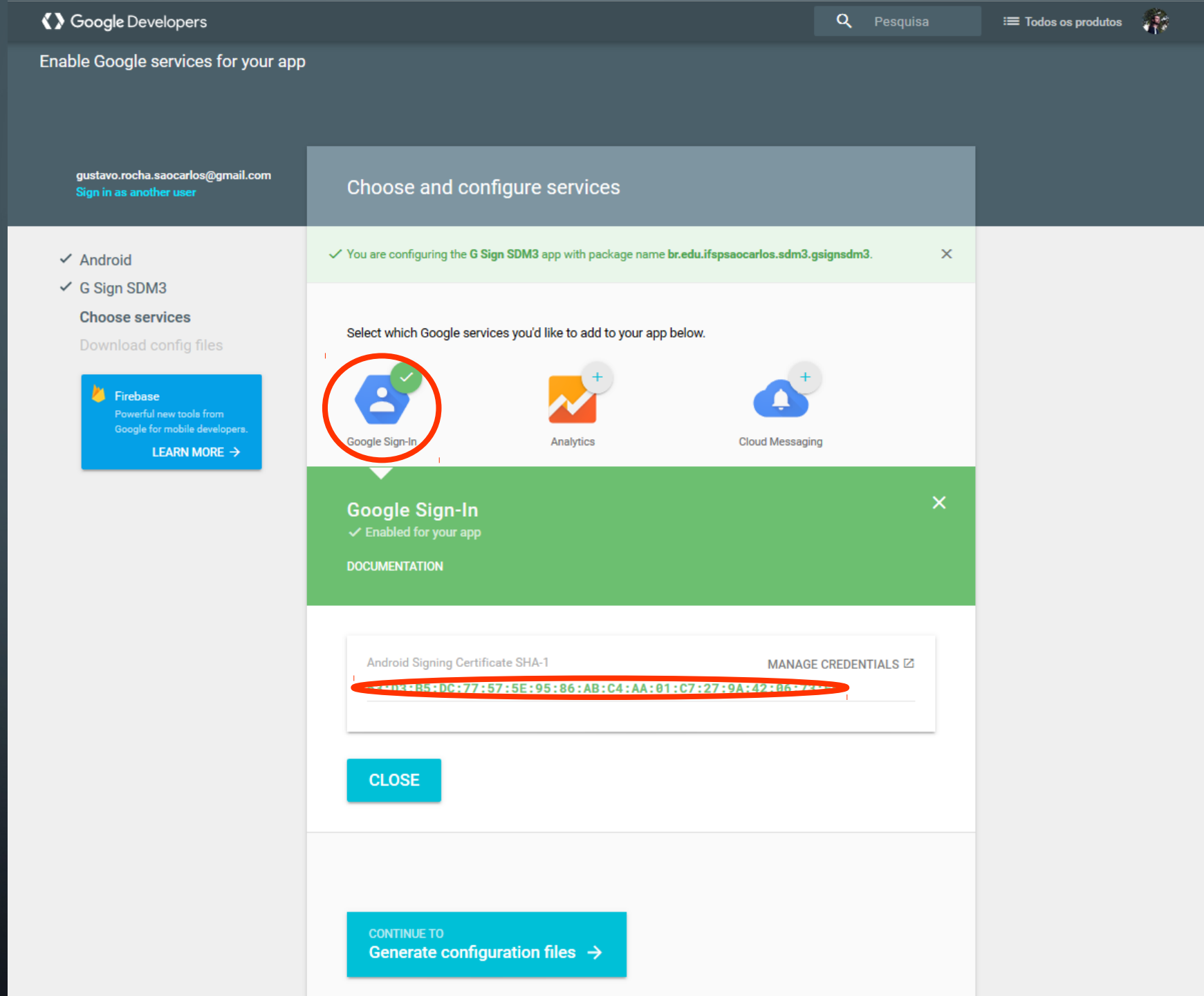
G Sign SDM3

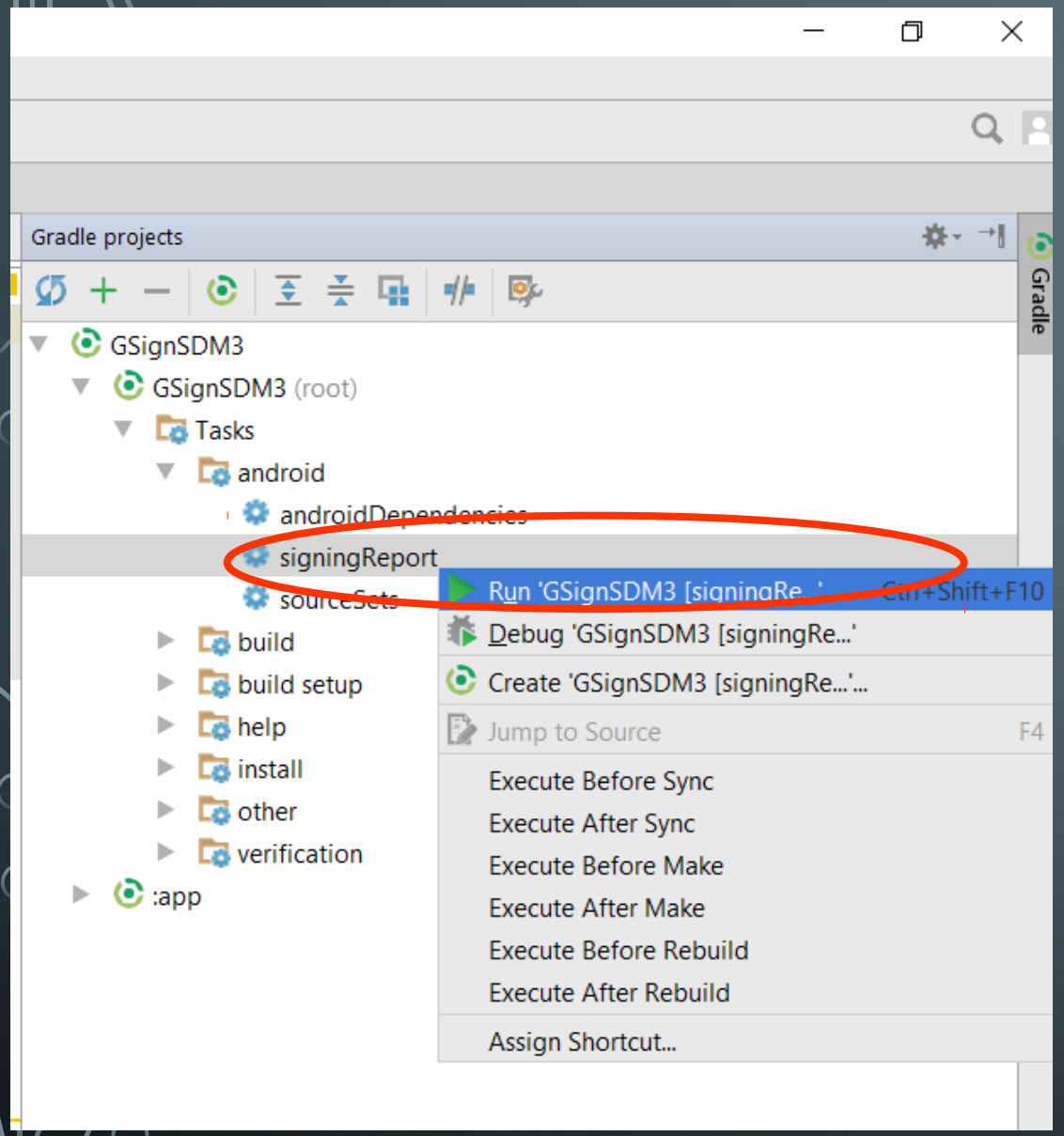
Services will be added to your existing project in the Google Developers Console.

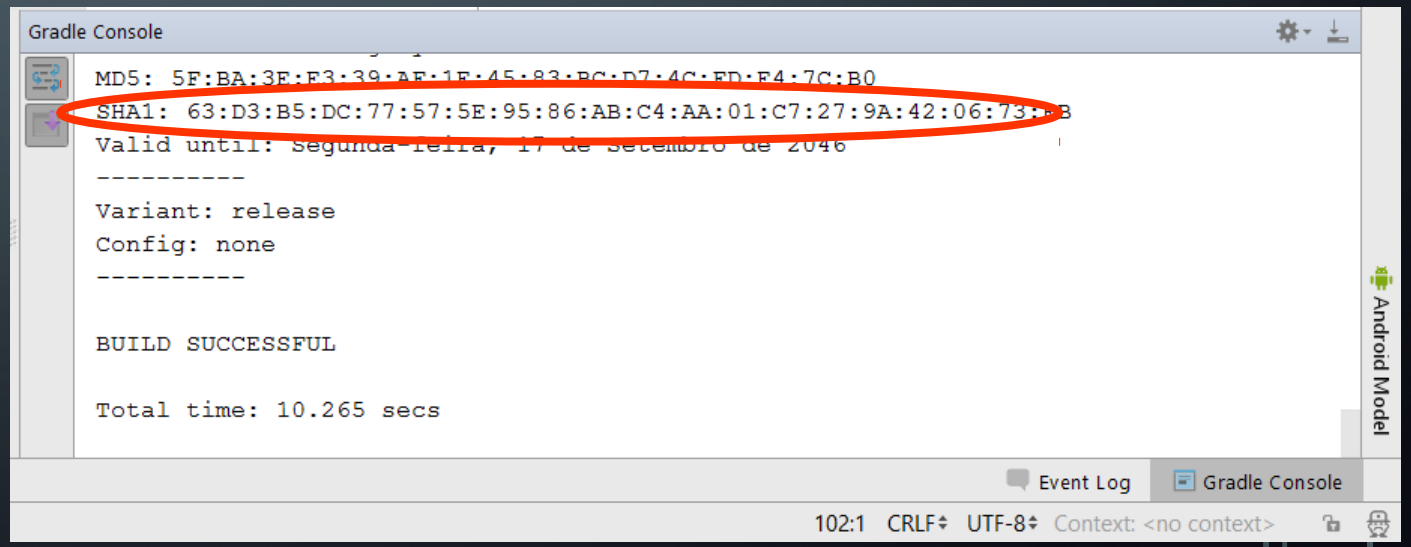
Android package name

br.edu.ifspsaocarlos.sdm3.gsignsdm3

CONTINUE TO
Choose and configure services →

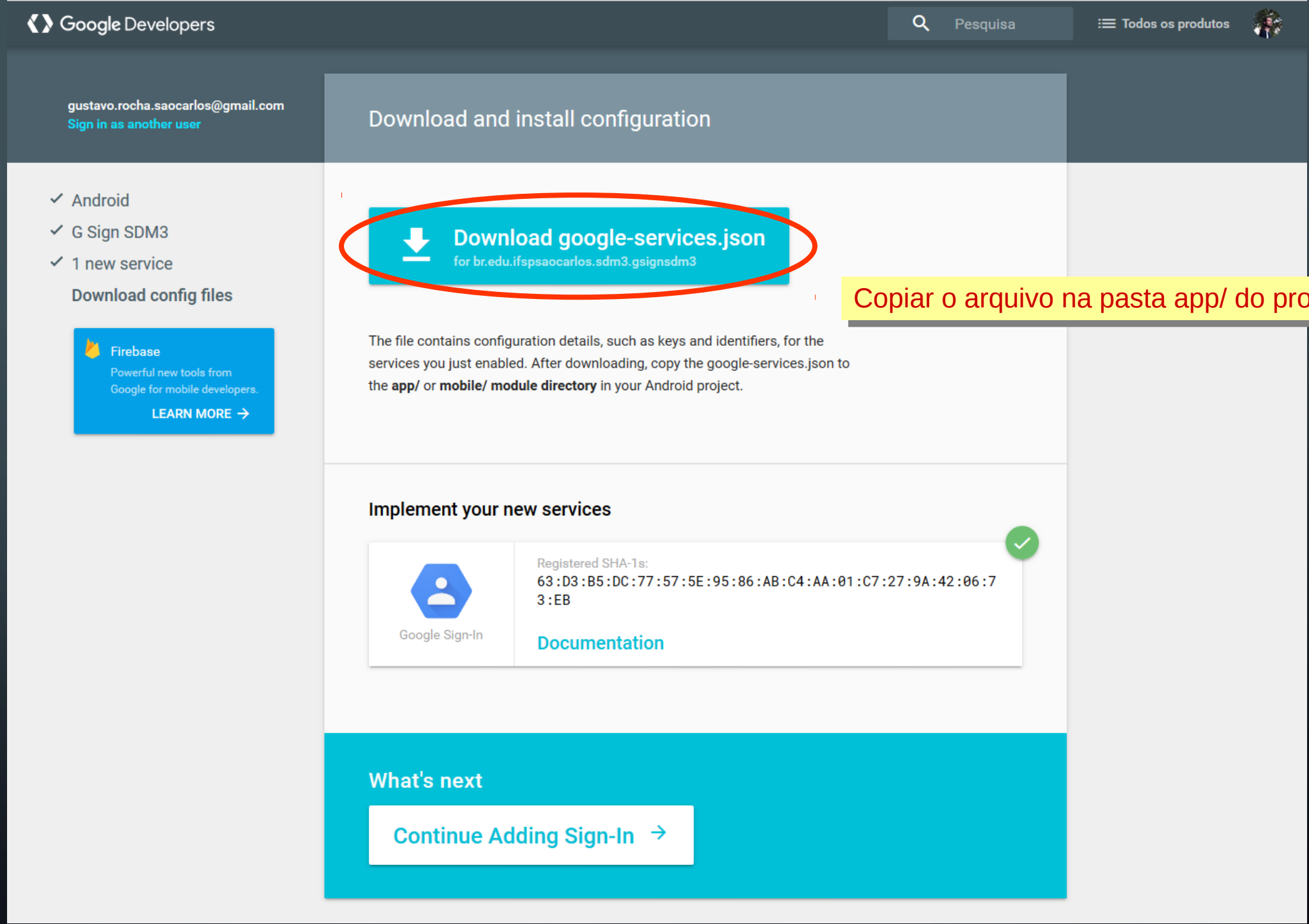

Google Developers
Pesquisa
Todos os produtos
Enable Google services for your app
gustavo.rocha.saocarlos@gmail.com
Sign in as another user
Choose and configure services
Android
G Sign SDM3
Choose services
Download config files
Firebase
Powerful new tools from Google for mobile developers.
LEARN MORE
You are configuring the G Sign SDM3 app with package name br.edu.ifspsaocarlos.sdm3.gsignsdm3.
Select which Google services you'd like to add to your app below.
Google Sign-In
Analytics
Cloud Messaging
Google Sign-In
Enabled for your app
DOCUMENTATION
Android Signing Certificate SHA-1
MANAGE CREDENTIALS
:B5:DC:77:57:5E:95:86:AB:C4:AA:01:C7:27:9A:42:06:
CLOSE
CONTINUE TO
Generate configuration files


Gradle projects
GSignSDM3
GSignSDM3 (root)
Tasks
android
androidDependencies
signingReport
sourceSets
build
build setup
help
install
other
verification
:app
Run 'GSignSDM3 [signingRe...' Ctrl+Shift+F10
Debug 'GSignSDM3 [signingRe...'
Create 'GSignSDM3 [signingRe...'...
Jump to Source F4
Execute Before Sync
Execute After Sync
Execute Before Make
Execute After Make
Execute Before Rebuild
Execute After Rebuild
Assign Shortcut...
Gradle


Gradle Console
MD5: 5F:BA:3E:F3:39:AF:1F:45:83:BC:D7:4C:FD:F4:7C:B0
SHA1: 63:D3:B5:DC:77:57:5E:95:86:AB:C4:AA:01:C7:27:9A:42:06:73:
Valid until: Segunda-feira, 17 de Setembro de 2046

Variant: release
Config: none

BUILD SUCCESSFUL
Total time: 10.265 secs
Event Log
Gradle Console
102:1 CRLF UTF-8 Context: <no context>
Android Model


Google Developers
Pesquisa
Todos os produtos
gustavo.rocha.saocarlos@gmail.com
Sign in as another user
Download and install configuration
✓ Android
✓ G Sign SDM3
✓ 1 new service
Download config files
Firebase
Powerful new tools from Google for mobile developers.
LEARN MORE →
Download google-services.json
for br.edu.ifspsaocarlos.sdm3.gsignsdm3
Copiar o arquivo na pasta app/ do projeto
The file contains configuration details, such as keys and identifiers, for the services you just enabled. After downloading, copy the google-services.json to the app/ or mobile/ module directory in your Android project.
Implement your new services
Google Sign-In
Registered SHA-1s:
63:D3:B5:DC:77:57:5E:95:86:AB:C4:AA:01:C7:27:9A:42:06:73:EB
Documentation
What's next
Continue Adding Sign-In →

build.gradle (módulo de projeto)

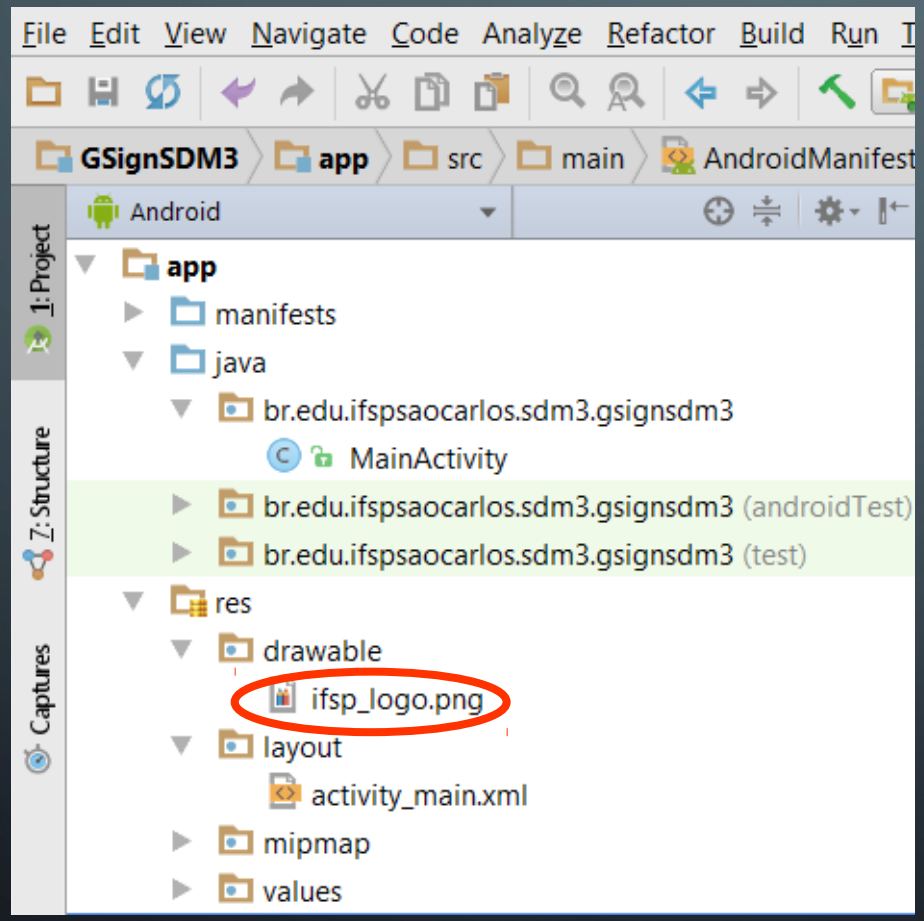
```
dependencies {
    classpath 'com.android.tools.build:gradle:2.2.3'
    classpath 'com.google.gms:google-services:3.0.0'

    // NOTE: Do not place your application dependencies here; they belong
    // in the individual module build.gradle files
}
```

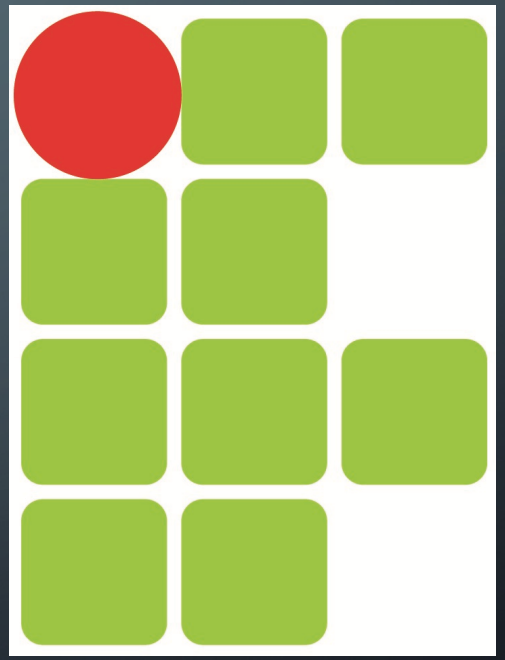
build.gradle (módulo de app)

picasso serve para renderizar a imagem de perfil do usuário

```
dependencies {
    compile 'com.squareup.picasso:picasso:2.5.2'
    compile 'com.google.android.gms:play-services-auth:9.8.0'
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-
core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-
annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.1'
    testCompile 'junit:junit:4.12'
}

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

```xml
<resources>
    <string name="app_name">G Sign SDM3</string>
    <string name="signed_in_fmt">Signed in as: %s</string>
    <string name="signed_out">Signed out</string>
    <string name="sign_out">Sign Out</string>
</resources>
```





```xml
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"        android:orientation="vertical"
    android:layout_width="match_parent"    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="br.edu.ifspsaocarlos.sdm3.gsignsdm3.MainActivity">

    <ImageView android:id="@+id/imageView"  android:src="@drawable/ifsp_logo"
        android:layout_width="200sp"        android:layout_height="200sp"
        android:layout_gravity="center"     android:layout_marginBottom="30sp"/>

    <com.google.android.gms.common.SignInButton    android:id="@+id/sign_in_button"
        android:layout_width="wrap_content"        android:layout_height="wrap_content"
        android:layout_gravity="center"/>

    <TextView android:id="@+id/status"
        android:layout_width="match_parent"   android:layout_height="wrap_content"
        android:text="@string/signed_out"     android:textColor="@android:color/black"
        android:textSize="18sp"               android:gravity="center"
        android:layout_marginTop="20sp"/>

    <LinearLayout android:id="@+id/sign_out"
        android:layout_width="match_parent"  android:layout_height="match_parent"
        android:orientation="horizontal"     android:paddingLeft="16dp"
        android:visibility="gone"            android:paddingRight="16dp"
        tools:visibility="visible">

        <Button android:id="@+id/sign_out_button"
            android:layout_width="match_parent"      android:layout_height="wrap_content"
            android:text="@string/sign_out"          android:layout_gravity="bottom" />

    </LinearLayout>
</LinearLayout>
```

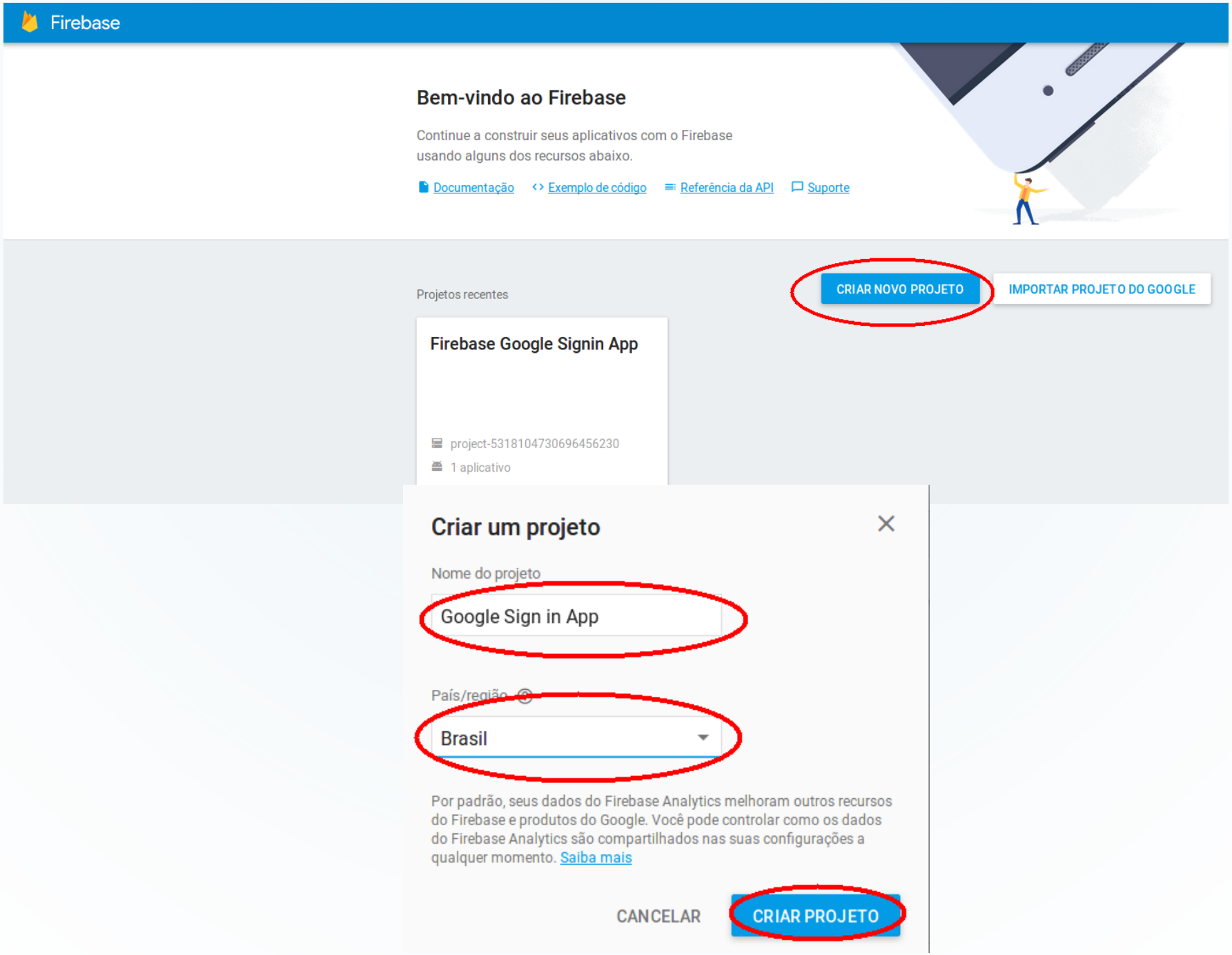
```
package br.edu.ifspsaocarlos.sdm3.gsignsdm3;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
import com.squareup.picasso.Picasso;
```

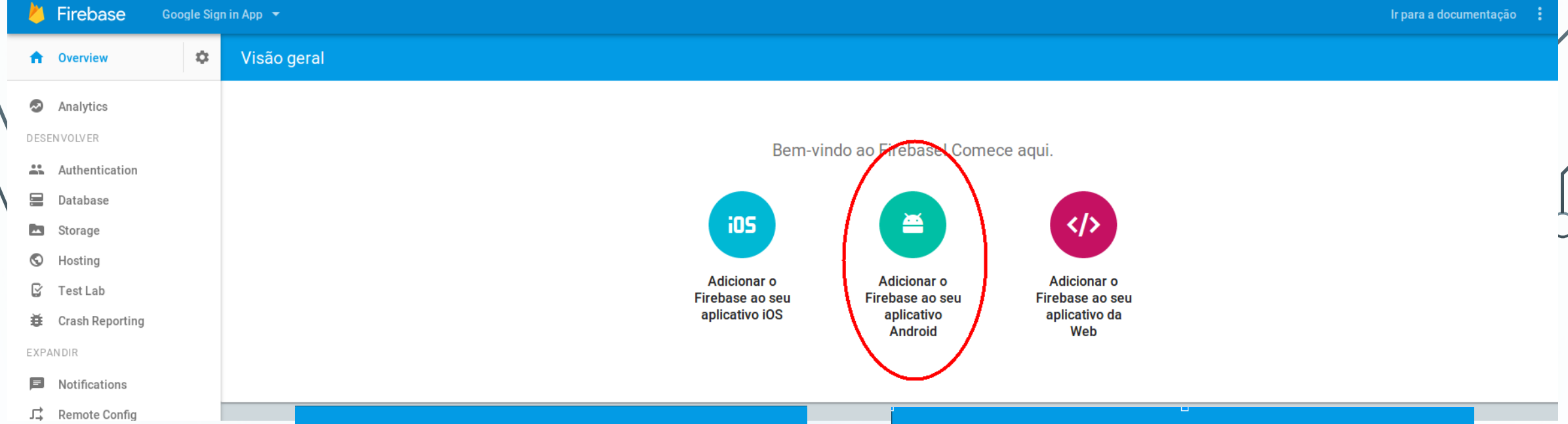
```
public class MainActivity extends AppCompatActivity implements
        GoogleApiClient.OnConnectionFailedListener,
        View.OnClickListener {

    private GoogleApiClient mGoogleApiClient;
    private static final String TAG = "SignInActivity";
    private static final int RC_SIGN_IN = 9001;
    private TextView mStatusTextView;
    private ImageView mStatusImageView;

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

    mStatusImageView = (ImageView) findViewById(R.id.imageView);
    mStatusTextView = (TextView) findViewById(R.id.status);
    findViewById(R.id.sign_in_button).setOnClickListener(this);
    findViewById(R.id.sign_out_button).setOnClickListener(this);

    GoogleSignInOptions gso = new
       GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
       .build();

    mGoogleApiClient = new GoogleApiClient.Builder(this)
       .enableAutoManage(this /* FragmentActivity */, this /* OnConnectionFailedListener */)
       .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
       .build();

    SignInButton signInButton = (SignInButton) findViewById(R.id.sign_in_button);
    signInButton.setSize(SignInButton.SIZE_WIDE);
}
```

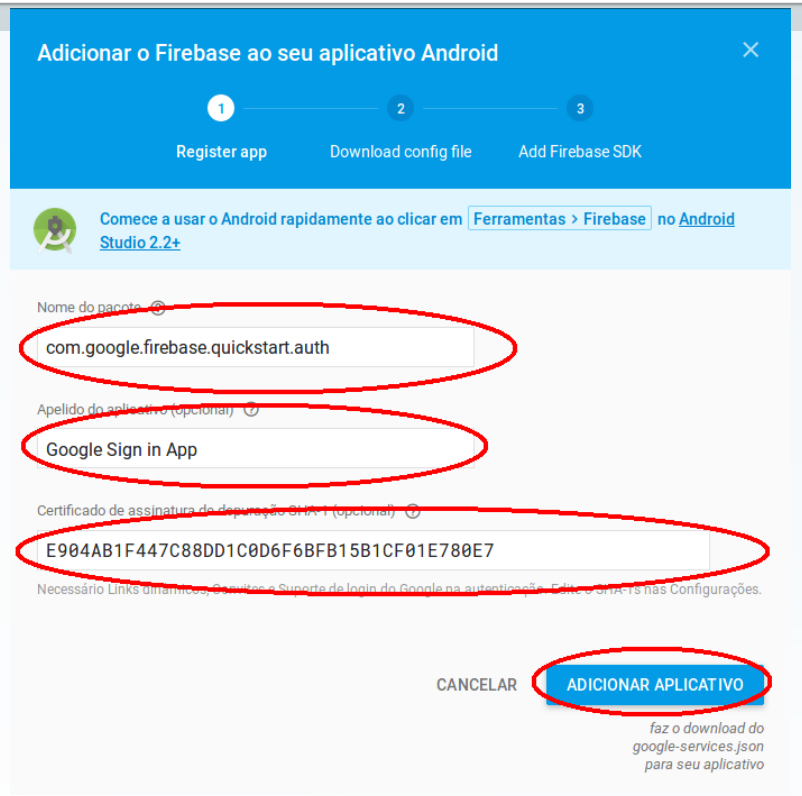
```
@Override
public void onConnectionFailed(ConnectionResult connectionResult) {
    Log.d(TAG, "onConnectionFailed:" + connectionResult);
}
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.sign_in_button:
            signIn();
            break;
        case R.id.sign_out_button:
            signOut();
            break;
    }
}
private void signIn() {
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
    startActivityForResult(signInIntent, RC_SIGN_IN);
}
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RC_SIGN_IN) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        handleSignInResult(result);
    }
}
private void signOut() {
    Auth.GoogleSignInApi.signOut(mGoogleApiClient).setResultCallback(
            new ResultCallback<Status>() {
                @Override
                public void onResult(Status status) {
                    updateUI(false);
                }
            });
}
```

```
private void handleSignInResult(GoogleSignInResult result) {
    Log.d(TAG, "handleSignInResult:" + result.isSuccess());
    if (result.isSuccess()) {
        GoogleSignInAccount acct = result.getSignInAccount();
        mStatusTextView.setText(getString(R.string.signed_in_fmt, acct.getDisplayName()));
        String urlFoto = acct.getPhotoUrl().toString();
        Picasso.with(MainActivity.this).load(urlFoto).resize(100,100).into(mStatusImageView);
        updateUI(true);
    } else {
        updateUI(false);
    }
}

private void updateUI(boolean signedIn) {
    if (signedIn) {
        findViewById(R.id.sign_in_button).setVisibility(View.GONE);
        findViewById(R.id.sign_out).setVisibility(View.VISIBLE);

    } else {
        mStatusTextView.setText(R.string.signed_out);
        mStatusImageView.setImageResource(R.drawable.ifsp_logo);

        findViewById(R.id.sign_in_button).setVisibility(View.VISIBLE);
        findViewById(R.id.sign_out).setVisibility(View.GONE);
    }
}

}
```

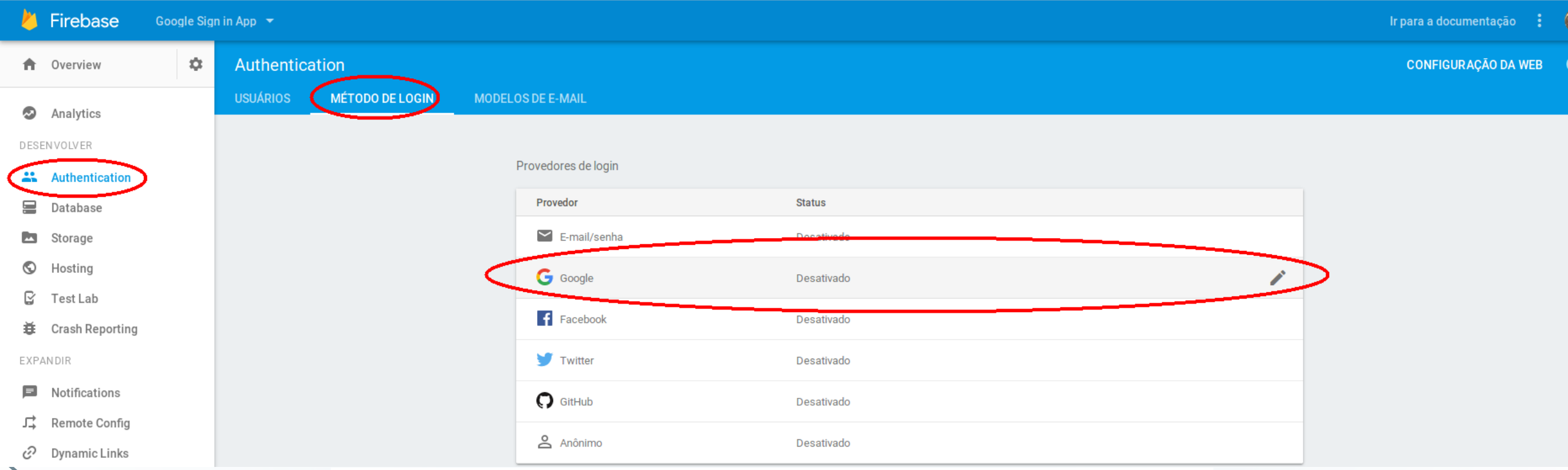
GOOGLE SIGN-IN COM FIREBASE

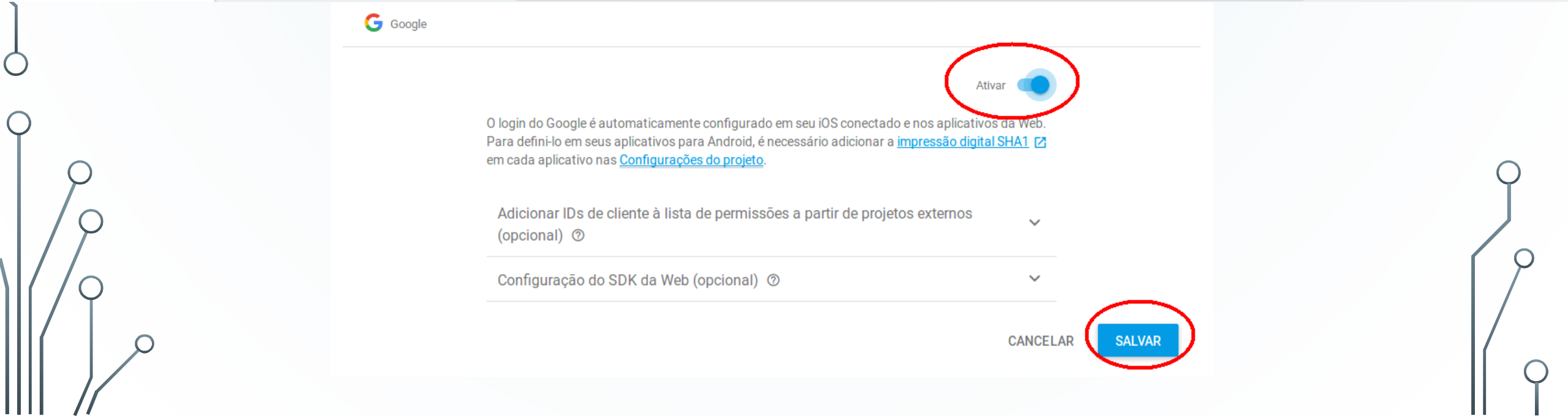

Firebase
Bem-vindo ao Firebase
Continue a construir seus aplicativos com o Firebase usando alguns dos recursos abaixo.
Documentação
Exemplo de código
Referência da API
Suporte
Projetos recentes
CRIAR NOVO PROJETO
IMPORTAR PROJETO DO GOOGLE
Firebase Google Signin App
project-5318104730696456230
1 aplicativo
Criar um projeto
Nome do projeto
Google Sign in App
País/região
Brasil
Por padrão, seus dados do Firebase Analytics melhoram outros recursos do Firebase e produtos do Google. Você pode controlar como os dados do Firebase Analytics são compartilhados nas suas configurações a qualquer momento. Saiba mais
CANCELAR
CRIAR PROJETO


Firebase
Google Sign in App
Ir para a documentação
Overview
Visão geral
Analytics
DESENVOLVER
Authentication
Database
Storage
Hosting
Test Lab
Crash Reporting
EXPANDIR
Notifications
Remote Config
Bem-vindo ao Firebase! Comece aqui.
iOS
Adicionar o Firebase ao seu aplicativo iOS
Adicionar o Firebase ao seu aplicativo Android
Adicionar o Firebase ao seu aplicativo da Web


Adicionar o Firebase ao seu aplicativo Android
1
2
3
Register app
Download config file
Add Firebase SDK
Comece a usar o Android rapidamente ao clicar em Ferramentas > Firebase no Android Studio 2.2+
Nome do pacote
com.google.firebase.quickstart.auth
Apelido do aplicativo (opcional)
Google Sign in App
Certificado de assinatura de depuração SHA-1 (opcional)
E904AB1F447C88DD1C0D6F6BFB15B1CF01E780E7
Necessário Links dinâmicos, Convites e Suporte de login do Google na autenticação. Edite o SHA-1s nas Configurações.
CANCELAR
ADICIONAR APLICATIVO
faz o download do google-services.json para seu aplicativo


Adicionar o Firebase ao seu aplicativo Android
1
2
3
Inserir detalhes do app
Copiar arquivo de configuração
Adicionar a build.gradle
buildscript {
dependencies {
// Add this line
classpath 'com.google.gms:google-services:3.0.0'
}
}
2. build.gradle do nível do aplicativo (<project>/<app-module>/build.gradle):
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
inclui o Firebase Analytics por padrão
3. Finalmente, pressione "Sincronizar agora" na barra que aparece no IDE:
Gradle files have changed sir
Sync now
FINALIZAR


Firebase
Google Sign in App
Ir para a documentação
Overview
Analytics
DESENVOLVER
Authentication
Database
Storage
Hosting
Test Lab
Crash Reporting
EXPANDIR
Notifications
Remote Config
Dynamic Links
Authentication
CONFIGURAÇÃO DA WEB
USUÁRIOS
MÉTODO DE LOGIN
MODELOS DE E-MAIL
Provedores de login
Provedor
Status
E-mail/senha
Google
Desativado
Facebook
Desativado
Twitter
Desativado
GitHub
Desativado
Anônimo
Desativado


Google
Ativar
O login do Google é automaticamente configurado em seu iOS conectado e nos aplicativos da Web.
Para defini-lo em seus aplicativos para Android, é necessário adicionar a impressão digital SHA1
em cada aplicativo nas Configurações do projeto.
Adicionar IDs de cliente à lista de permissões a partir de projetos externos (opcional)
Configuração do SDK da Web (opcional)
CANCELAR
SALVAR

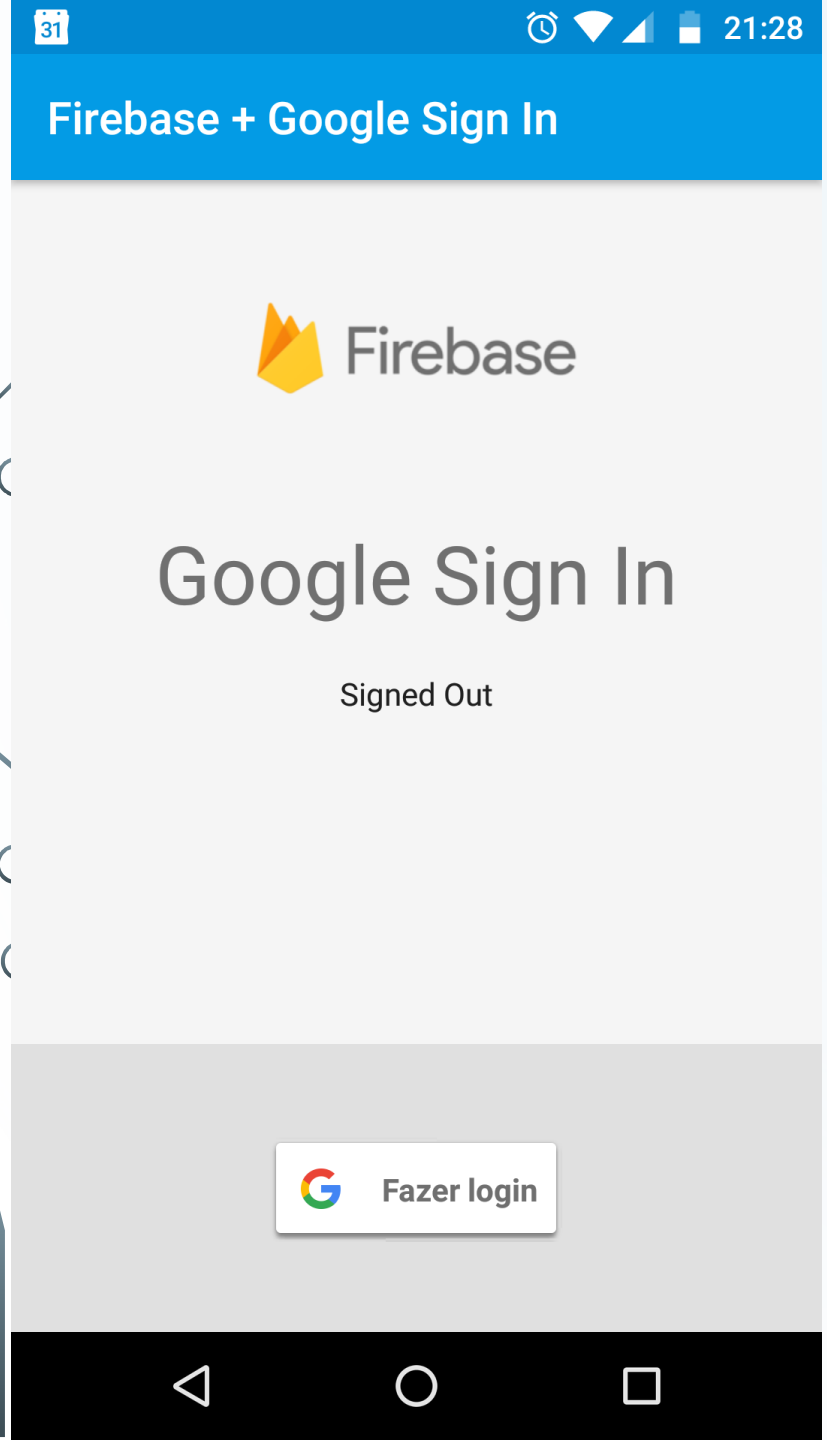
```
apply plugin: 'com.android.application'

android {
  // ...
}

dependencies {
  // ...
  compile 'com.google.firebase:firebase-core:9.6.1'
}

// ADD THIS AT THE BOTTOM
apply plugin: 'com.google.gms.google-services'
```

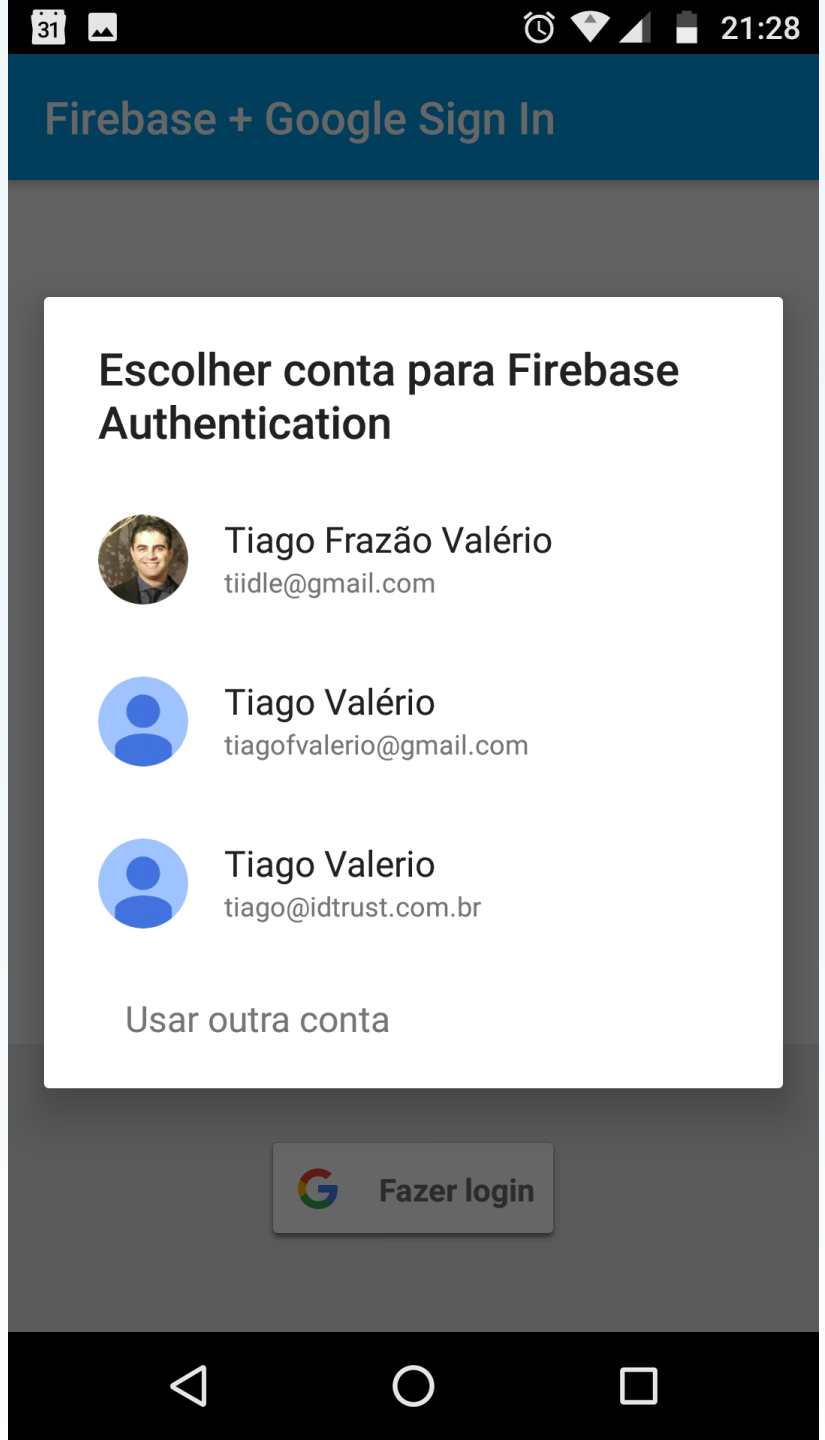
Linha de dependência do Gradle	Serviço
com.google.firebase:firebase-core:9.6.1	Analytics
com.google.firebase:firebase-database:9.6.1	Realtime Database
com.google.firebase:firebase-storage:9.6.1	Storage
com.google.firebase:firebase-crash:9.6.1	Crash Reporting
com.google.firebase:firebase-auth:9.6.1	Authentication
com.google.firebase:firebase-messaging:9.6.1	Cloud Messaging / Notifications
com.google.firebase:firebase-config:9.6.1	Remote Config
com.google.firebase:firebase-invites:9.6.1	Invites / Dynamic Links
com.google.firebase:firebase-ads:9.6.1	AdMob
com.google.android.gms:play-services-appindexing:9.6.1	App Indexing

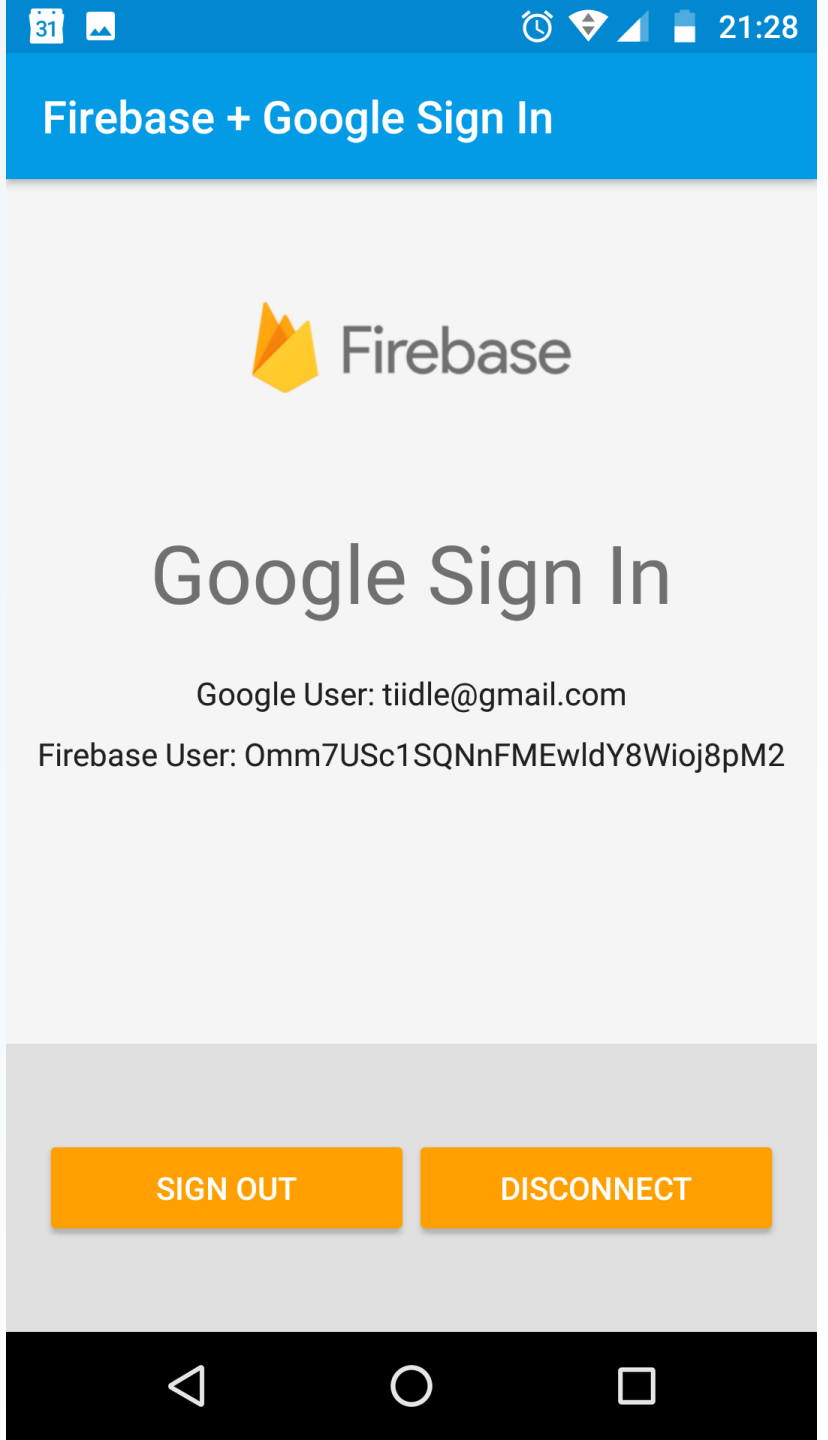
```
// [START declare_auth]
private FirebaseAuth mAuth;
// [END declare_auth]
```

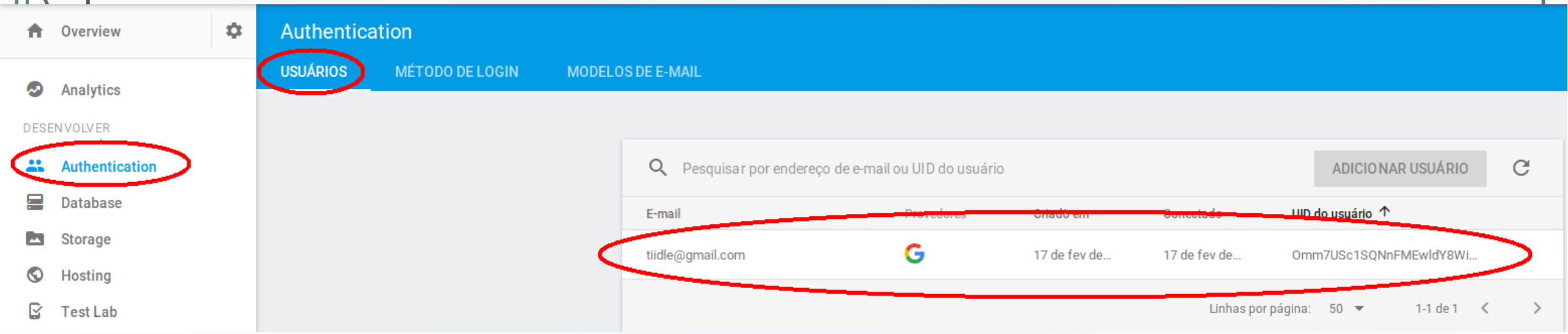
```
// [START initialize_auth]
mAuth = FirebaseAuth.getInstance();
// [END initialize_auth]
```

```
// [START auth_with_google]
private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
    Log.d(TAG, "firebaseAuthWithGoogle:" + acct.getId());
    // [START_EXCLUDE silent]
    showProgressDialog();
    // [END_EXCLUDE]

    AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);
    mAuth.signInWithCredential(credential)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    Log.d(TAG, "signInWithCredential:onComplete:" + task.isSuccessful());

                    // If sign in fails, display a message to the user. If sign in succeeds
                    // the auth state listener will be notified and logic to handle the
                    // signed in user can be handled in the listener.
                    if (!task.isSuccessful()) {
                        Log.w(TAG, "signInWithCredential", task.getException());
                        Toast.makeText(GoogleSignInActivity.this, "Authentication failed.",
                                Toast.LENGTH_SHORT).show();
                    }
                    // [START_EXCLUDE]
                    hideProgressDialog();
                    // [END_EXCLUDE]
                }
            });
}
// [END auth_with_google]
```


21:28
Firebase + Google Sign In
Firebase
Google Sign In
Signed Out
Fazer login


21:28
Firebase + Google Sign In
Escolher conta para Firebase Authentication
Tiago Frazão Valério
tiidle@gmail.com
Tiago Valério
tiagofvalerio@gmail.com
Tiago Valerio
tiago@idtrust.com.br
Usar outra conta
Fazer login


21:28
Firebase + Google Sign In
Firebase
Google Sign In
Google User: tiidle@gmail.com
Firebase User: Omm7USc1SQNnFMEwldY8Wioj8pM2
SIGN OUT
DISCONNECT


Overview
Analytics
DESENVOLVER
Authentication
Database
Storage
Hosting
Test Lab
Authentication
USUÁRIOS
MÉTODO DE LOGIN
MODELOS DE E-MAIL
Pesquisar por endereço de e-mail ou UID do usuário
ADICIONAR USUÁRIO
E-mail
Provedores
Criado em
Conectado
UID do usuário
tiidle@gmail.com
17 de fev de...
17 de fev de...
Omm7USc1SQNnFMEwldY8Wi...
Linhas por página: 50
1-1 de 1

DÚVIDAS?