

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO – *CAMPUS* SÃO CARLOS

ESPECIALIZAÇÃO *LATO SENSU* EM
DESENVOLVIMENTO DE SISTEMAS PARA DISPOSITIVOS MÓVEIS

Autorização de usuários por meio de Redes Sociais

São Carlos – SP
2017

LISTA DE FIGURAS

| | |
|----------------|----|
| Figura 1..... | 3 |
| Figura 2..... | 4 |
| Figura 3..... | 4 |
| Figura 4..... | 5 |
| Figura 5..... | 5 |
| Figura 6..... | 6 |
| Figura 7..... | 7 |
| Figura 8..... | 7 |
| Figura 9..... | 8 |
| Figura 10..... | 9 |
| Figura 11..... | 10 |
| Figura 12..... | 11 |
| Figura 13..... | 12 |

SUMÁRIO

| | | |
|-----|--|----|
| 1 | Login social via Twitter | 3 |
| 1.1 | Registrando seu aplicativo | 3 |
| 1.2 | Registrando-se no Fabric..... | 4 |
| 1.3 | Instalando o plugin do Fabric no Android Studio..... | 5 |
| 1.4 | Usando o plug-in Fabric..... | 8 |
| 1.5 | Definindo um layout..... | 14 |
| 1.6 | Programando o botão de login | 15 |
| 2 | ANEXO | 16 |

1 Login social via Twitter

Para realizar a conexão com esta rede social é disponibilizado para os desenvolvedores o framework Fabric, o qual contém bibliotecas e uma estrutura de tutoriais e instaladores que facilitam a criação do vínculo entre o seu aplicativo e o Twitter.

Para seguir este guia o desenvolvedor deverá ter disponível o Software Android Studio e uma conta ativa no Twitter e Fabric.

1.1 Registrando seu aplicativo

Como primeiro passo devemos registrar o nosso aplicativo no sistema do Twitter, para isso utilizaremos o site <https://apps.twitter.com/>. Após entrar com o seu usuário do Twitter será apresentado uma tela (Figura 1) onde você poderá iniciar o registro de seu projeto.

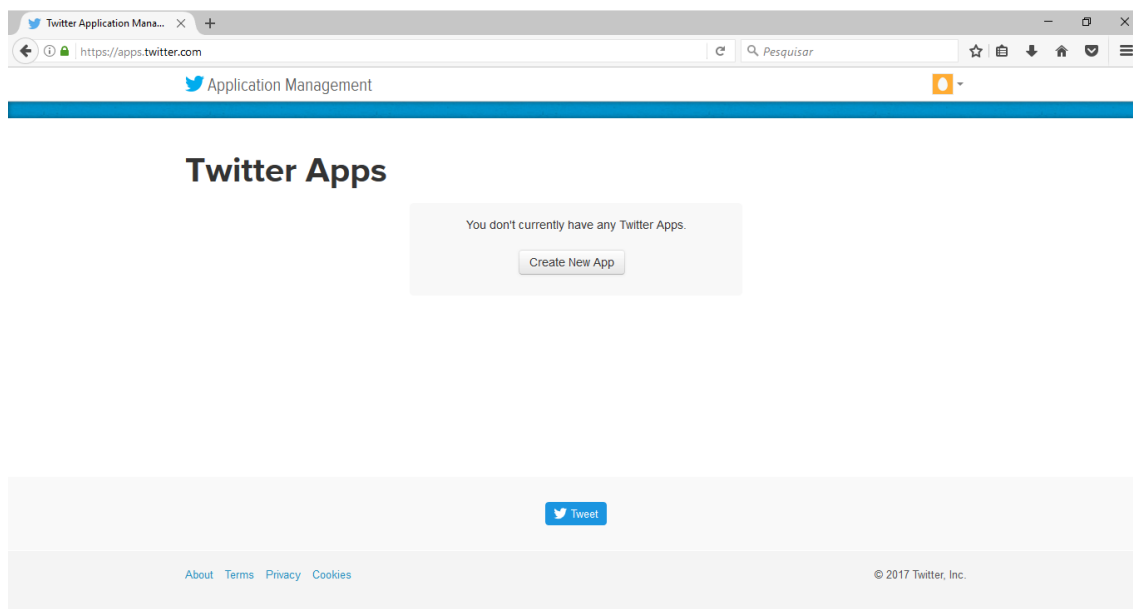


Figura 1

Para registrar o seu aplicativo, basta acionar a opção “*Create New App*” e preencher as informações conforme a Figura 2.

As informações obrigatórias de preenchimento para seguir este tutorial são o nome do aplicativo (“*Name*”), descrição do aplicativo (“*Description*”) e um *website* (“*WebSite*”). Neste ultimo, caso você não possua nenhum site, você poderá colocar uma informação fictícia como <http://exemplo.com>.

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Figura 2

Após registrar o aplicativo, você poderá escolher a aba “Keys and Access Tokens” para verificar os códigos “Consumer Key” e “Consumer Secret” gerados por este cadastro (ver Figura 3). Estas informações serão utilizadas posteriormente para vincular o nosso projeto com este cadastro.

T Sign SDM3

Details Settings **Keys and Access Tokens** Permissions

Application Settings
Keep the “Consumer Secret” a secret. This key should never be human-readable in your application.

Consumer Key (API Key) **erPcLkzQlplug6Gtcm5OZkhzY**

Consumer Secret (API Secret) **ICGskJpON9sPpwUvulCbbPZlrhxTeVmfj0c9ggyHvp5VTAsBI5**

Access Level Read and write (modify app permissions)

Owner SteakStocco

Owner ID 102943788

Application Actions

Regenerate Consumer Key and Secret Change App Permissions

Figura 3

1.2 Registrando-se no Fabric

Como mencionado anteriormente, faremos o uso do framework Fabric, e para isso precisaremos criar uma conta.

O cadastro poderá ser realizado informando seus dados na opção “GET FABRIC” do site <http://fabric.io>. (Figura 4)

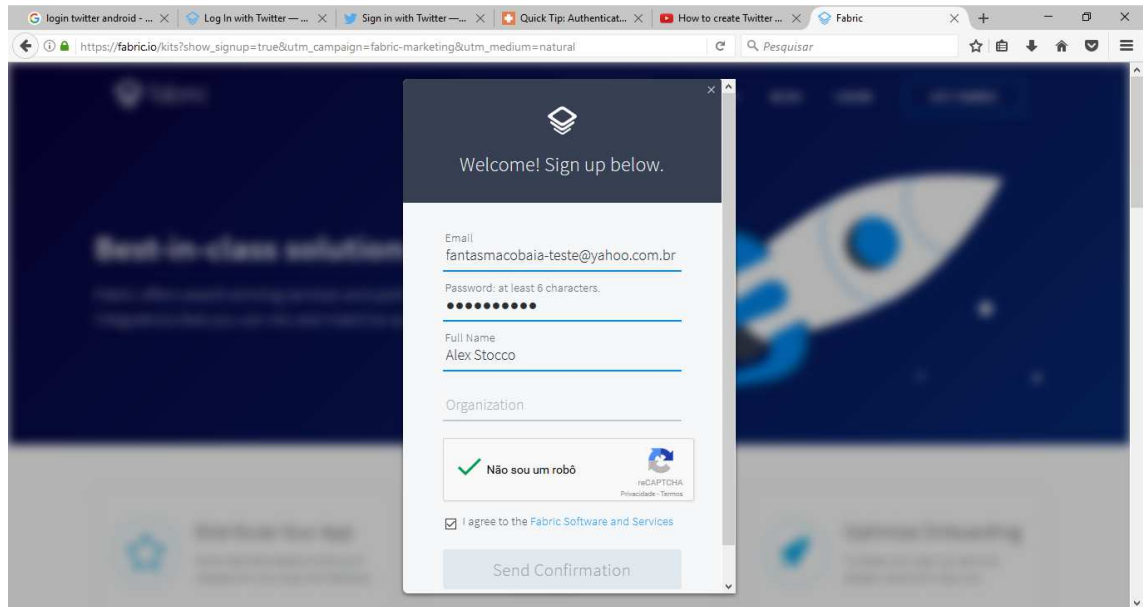


Figura 4

1.3 Instalando o plugin do Fabric no Android Studio

A instalação do Fabric poderá ser realizada diretamente pelo Android Studio seguindo os seguintes passos:

1. Abra o Android Studio e escolha o menu File/Settings (Figura 5);

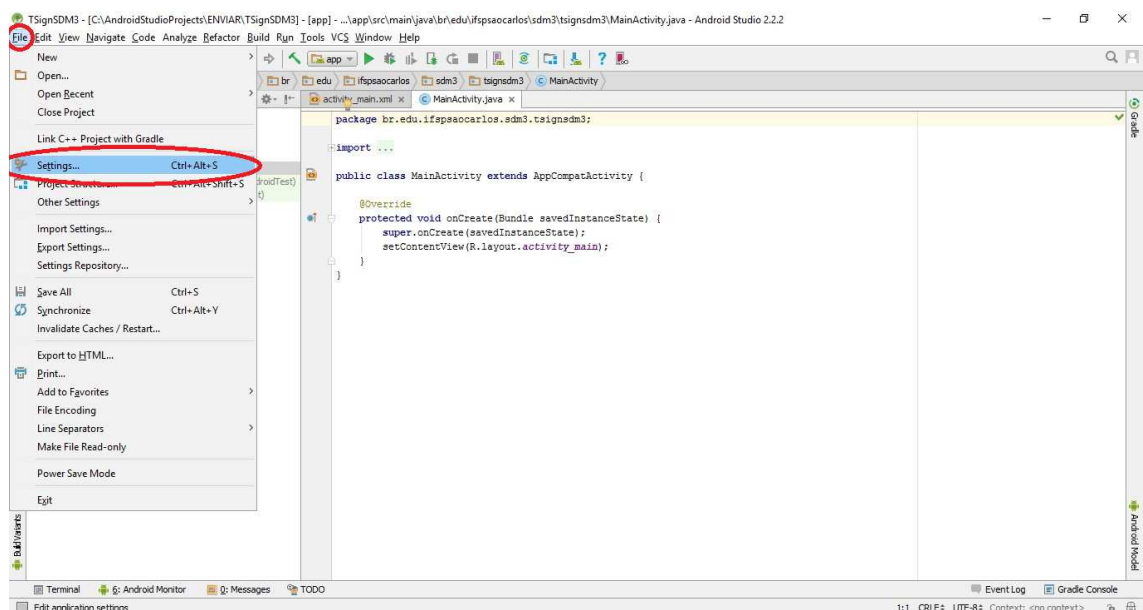


Figura 5

2. Na janela de *Settings* escolha o item “*Plugins*” e depois o botão “*Browse repositories...*” (Figura 6);

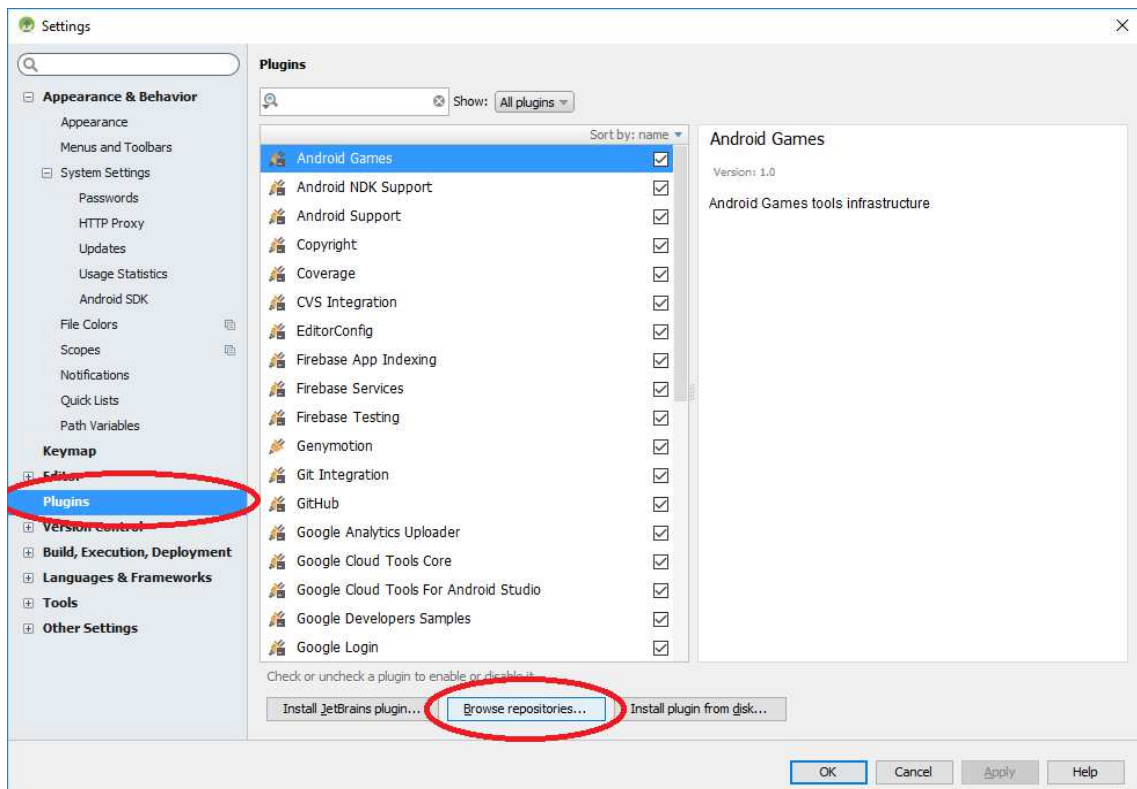


Figura 6

3. Na lista de plug-ins apresentada escolha a opção “*Fabric for Android Studio*” e depois clique em “*Install*” (Figura 7). Após realizar a instalação o Android Studio irá reinicializar automaticamente e liberará o uso do plug-in Fabric conforme mostrado na Figura 8.

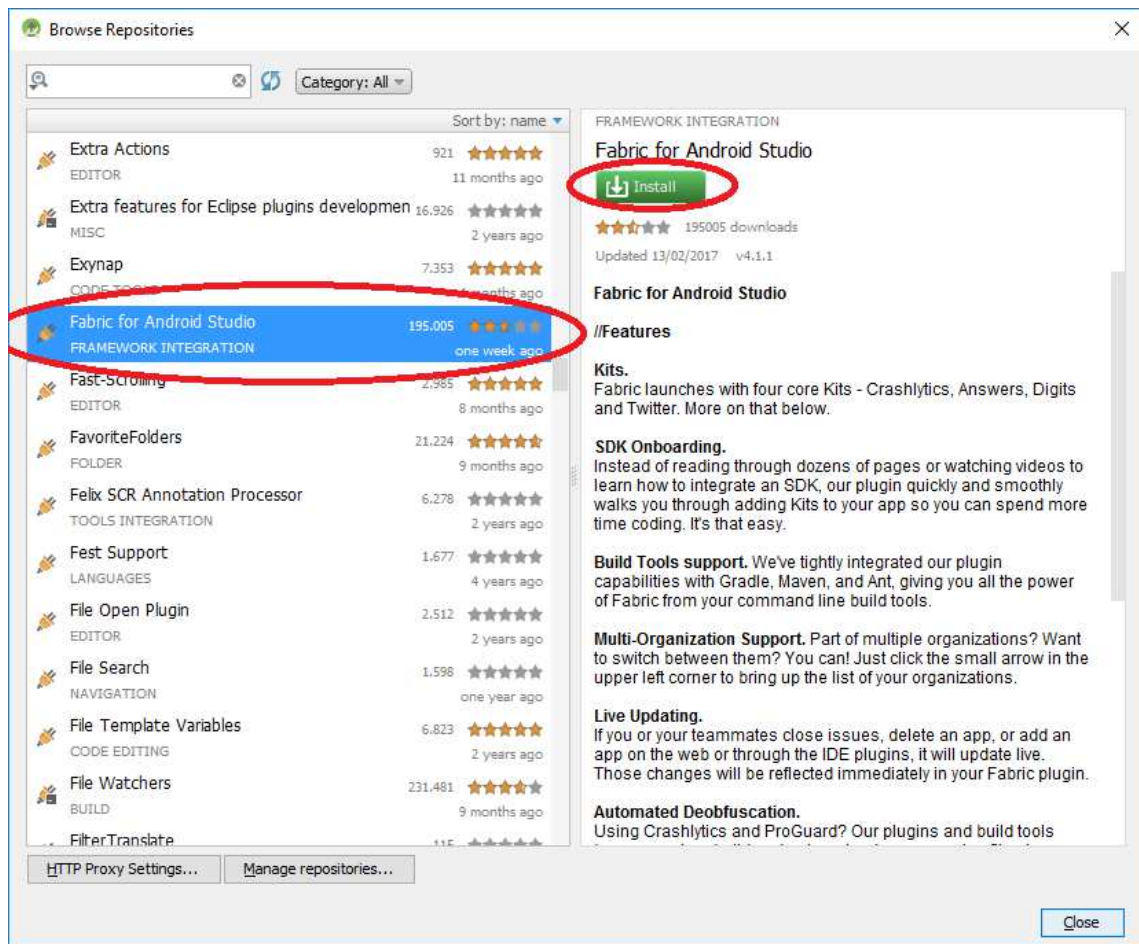


Figura 7

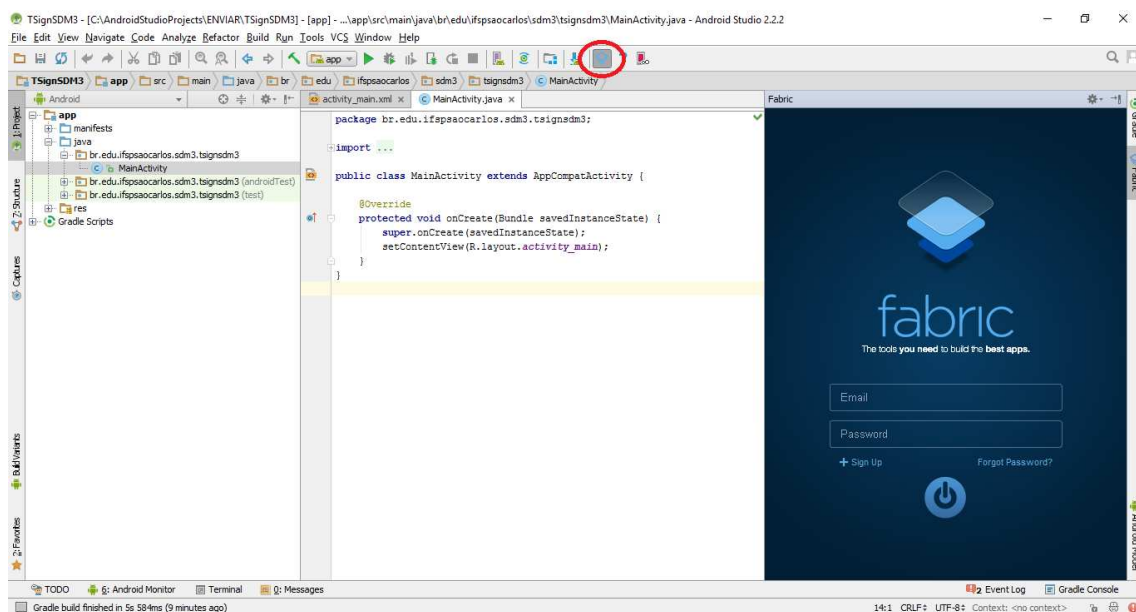


Figura 8

1.4 Usando o plug-in Fabric

Para utilizar o plug-in Fabric basta aciona-lo na barra de tarefas do Android Studio, conforme Figura 8 e inserir seu login e senha criados anteriormente no capítulo “Registrando-se no Fabric”.

Após o login você será guiado até uma janela mostrando alguns recursos disponíveis no Fabric. Escolha a opção “Twitter” como mostrado na Figura 9.

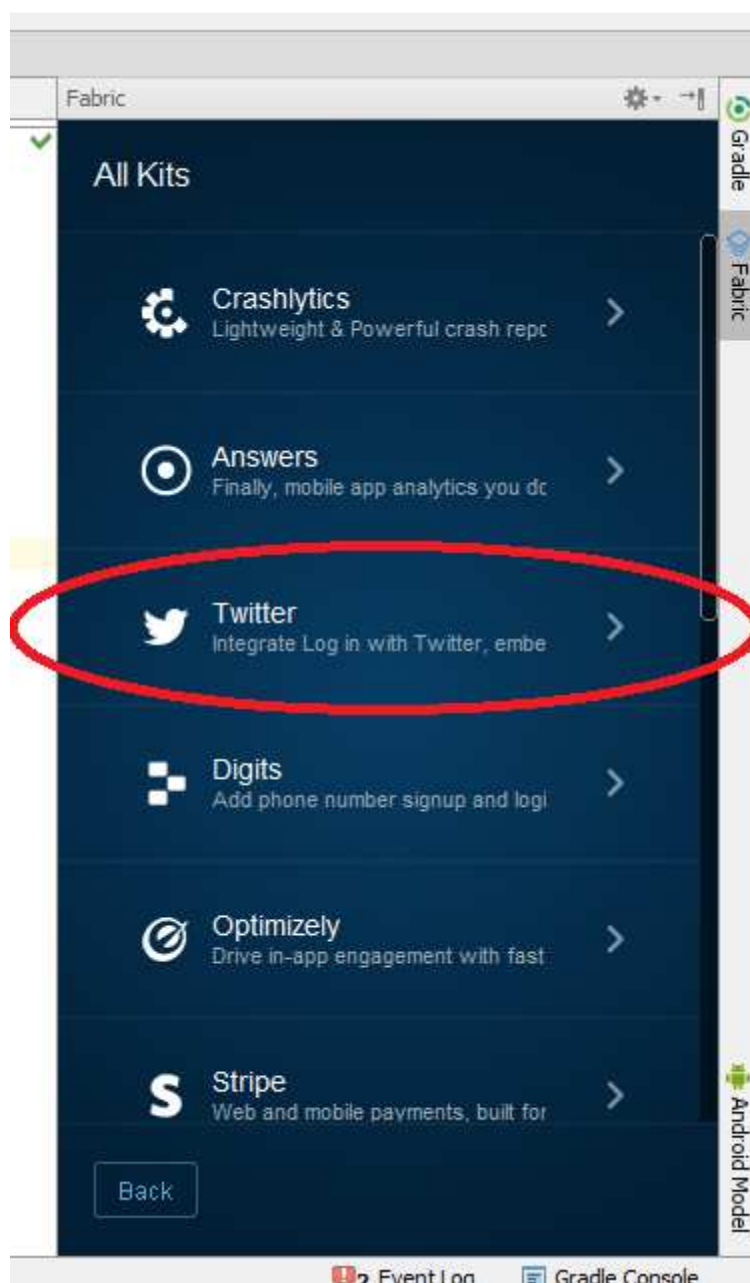


Figura 9

Na próxima tela apresentada escolha a opção “Install” e concorde com os termos descritos (Figura 10 e Figura 11).

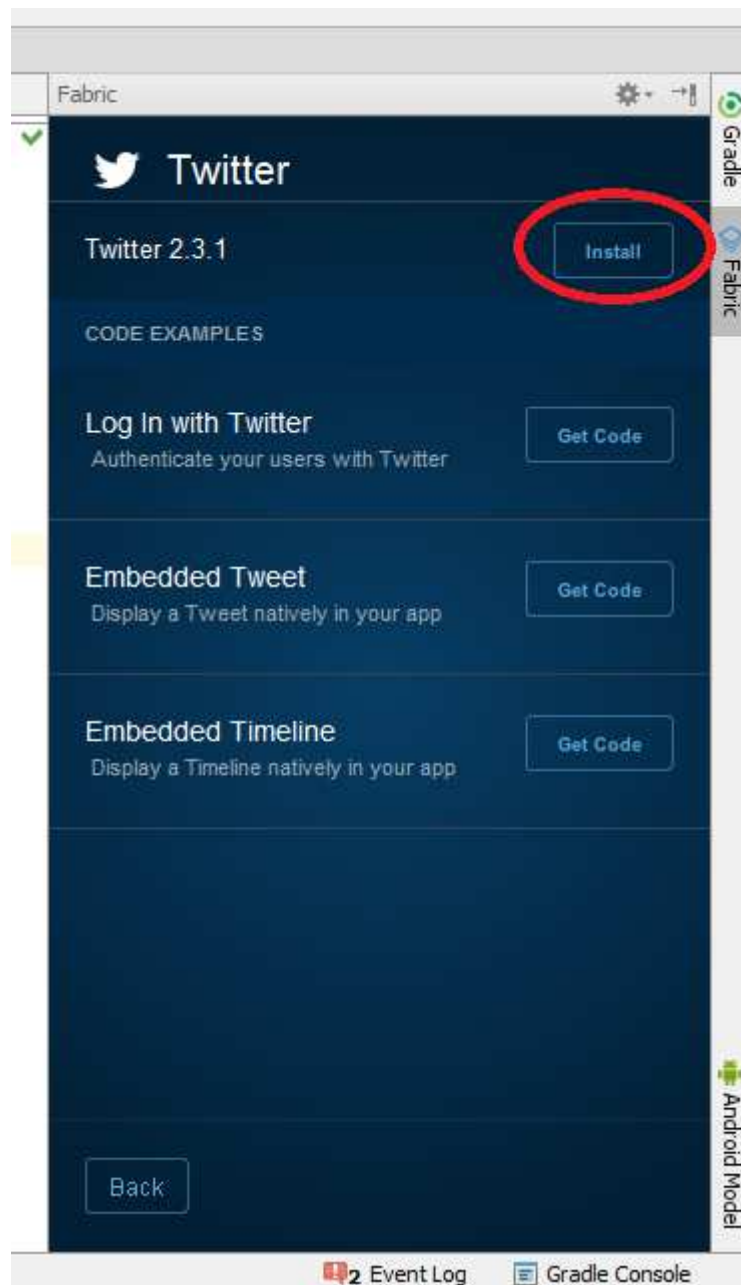


Figura 10

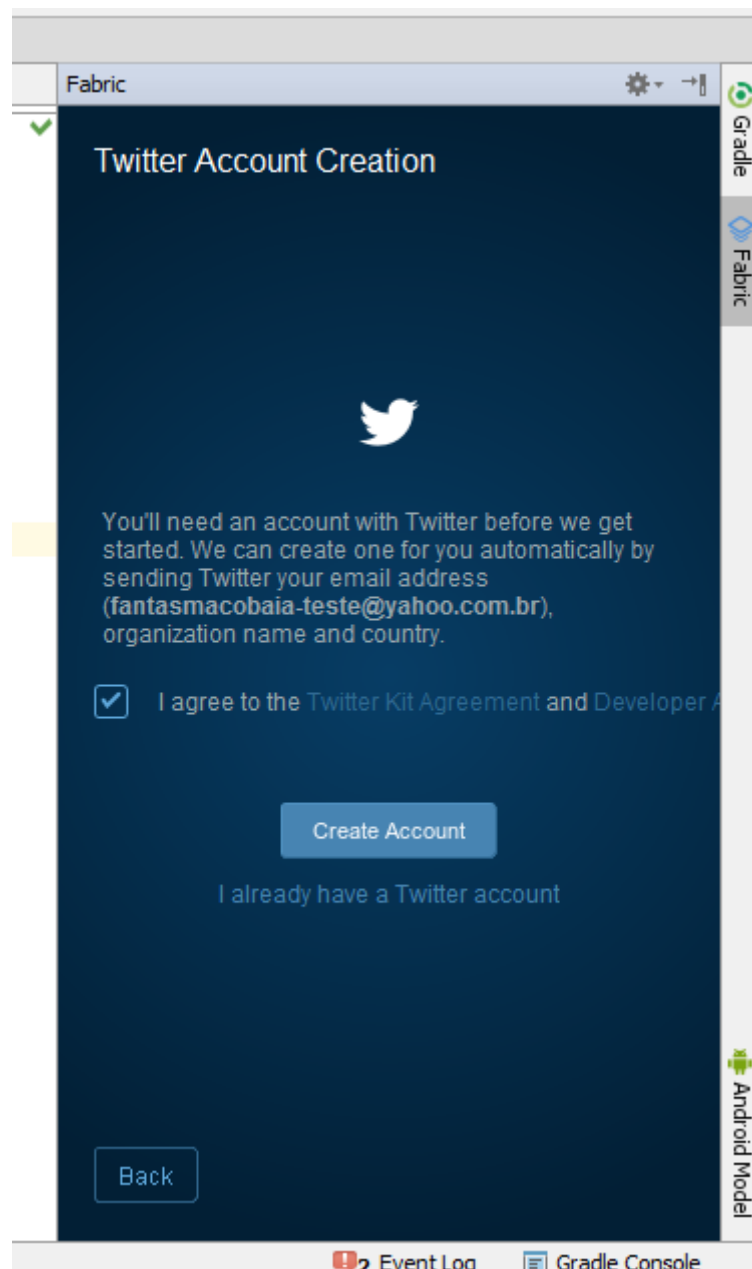


Figura 11

Concordando com os termos de uso você será redirecionado para a janela mostrada na Figura 12.

É nesta tela onde você irá vincular o seu projeto com o cadastro efetuado no site do Twitter, por isso, informe o *Consumer Key* e *Consumer Secret* que geramos no capítulo “Registrando seu aplicativo”.

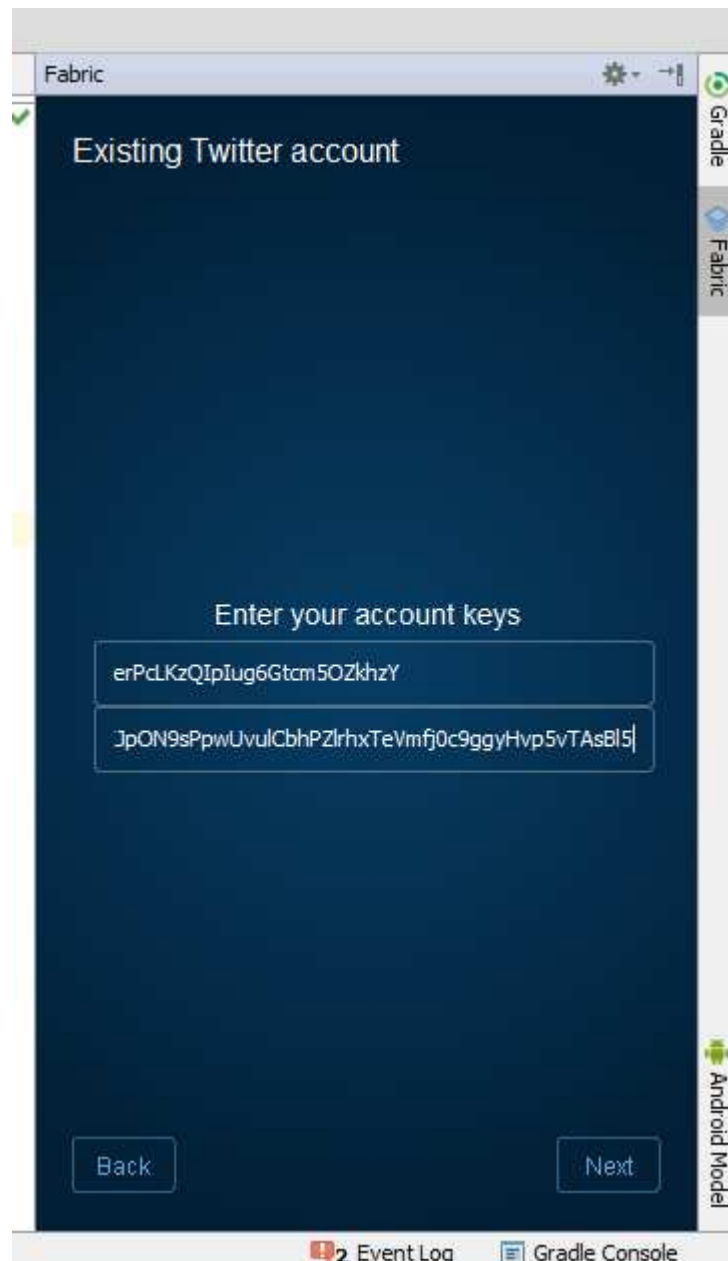


Figura 12

Para finalizar o próprio plug-in irá lhe mostrar os códigos que deverão ser inseridos em seu projeto para permitir a utilização do login social do Twitter (Figura 13). Você poderá inserir estas alterações automaticamente apenas clicando em “Apply”.

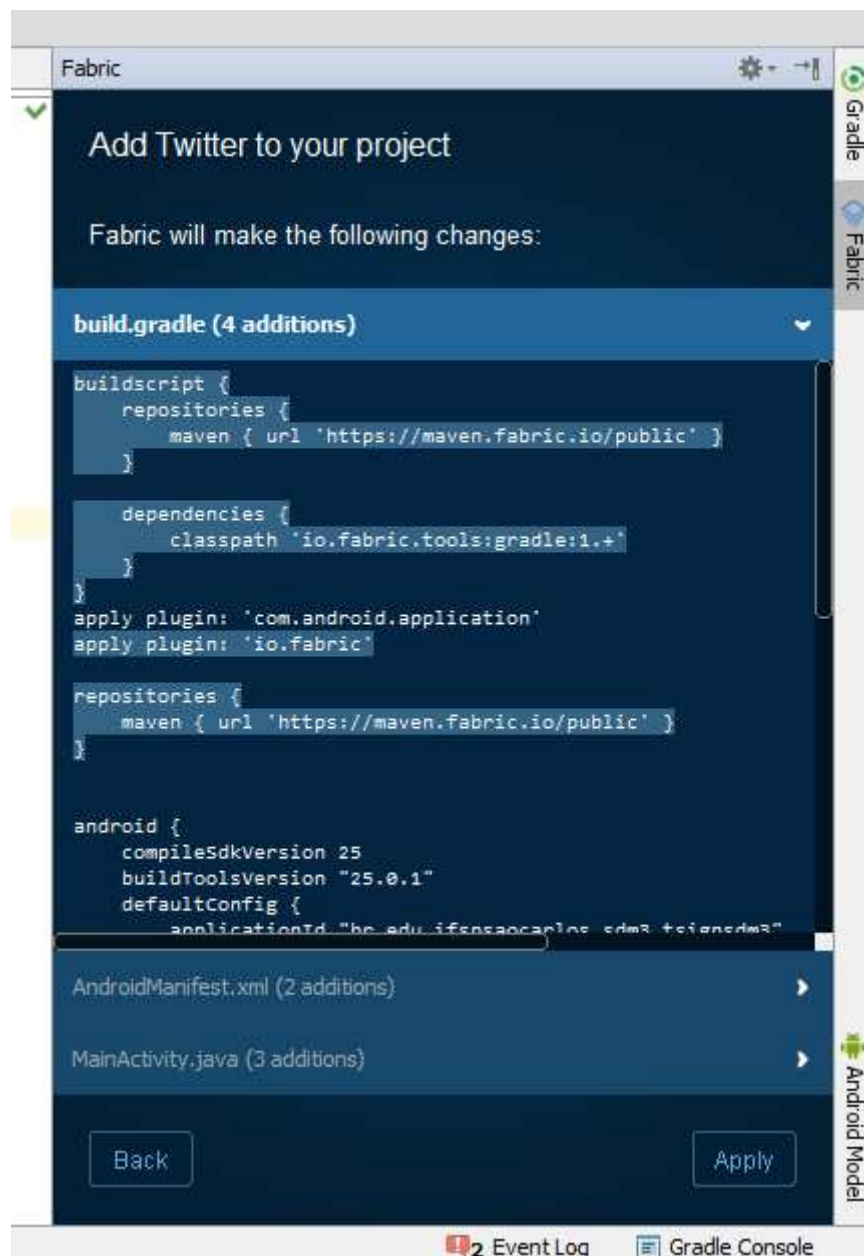


Figura 13

Abaixo, em amarelo, você poderá observar as alterações que serão efetuadas pelo plug-in.

| build.gradle |
|--|
| <pre> buildscript { repositories { maven { url 'https://maven.fabric.io/public' } } dependencies { classpath 'io.fabric.tools:gradle:1.+' } } apply plugin: 'com.android.application' apply plugin: 'io.fabric' repositories { maven { url 'https://maven.fabric.io/public' } } </pre> |

```

}

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.1"
    defaultConfig {
        applicationId "br.edu.ifpsaocarlos.sdm3.tsignsdm3"
        minSdkVersion 22
        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-v7:25.0.1'
    testCompile 'junit:junit:4.12'
    compile('com.twitter.sdk.android:twitter:2.3.1@aar') {
        transitive = true;
    }
}

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.edu.ifpsaocarlos.sdm3.tsignsdm3">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <meta-data
            android:name="io.fabric.ApiKey"
            android:value="0ee3648d02306ebf04ef640459d939d77641b3c2" />
        </application>

        <uses-permission android:name="android.permission.INTERNET" />
    </manifest>

```

MainActivity.java

```

package br.edu.ifpsaocarlos.sdm3.tsignsdm3;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import com.twitter.sdk.android.Twitter;
import com.twitter.sdk.android.core.TwitterAuthConfig;
import io.fabric.sdk.android.Fabric;

public class MainActivity extends AppCompatActivity {

    // Note: Your consumer key and secret should be obfuscated in your source code before shipping.
    private static final String TWITTER_KEY = "erPcLKzQlplug6Gtcm5OZkhzY";
    private static final String TWITTER_SECRET = "iCGsKJpON9sPpwUvuICbhPZlrhxTeVmfj0c9ggyHvp5vTAsBI5";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TwitterAuthConfig authConfig = new TwitterAuthConfig(TWITTER_KEY, TWITTER_SECRET);
        Fabric.with(this, new Twitter(authConfig));
        setContentView(R.layout.activity_main);
    }
}

```

1.5 Definindo um layout

Para testarmos a aplicação iremos criar um layout simples contendo apenas o botão padrão do Twitter para login e dois campos de texto para exibir o nome e o ID do usuário após o login.

Abaixo você poderá ver o código na íntegra.

| activity_main.xml |
|--|
| <pre> <?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools" android:id="@+id/activity_main" 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.ifpsaocarlos.sdm3.tsignsdm3.MainActivity"> <com.twitter.sdk.android.core.identity.TwitterLoginButton android:id="@+id/twitter_login_button" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_centerInParent="true"/> <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:id="@+id/userName" android:textSize="20sp" android:layout_alignParentTop="true" android:layout_alignParentLeft="true" /> <TextView </pre> |

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/userId"
        android:textSize="20sp"
        android:layout_below="@+id/userName"
    />
</RelativeLayout>

```

1.6 Programando o botão de login

Para finalizarmos iremos inserir a programação no botão de login para que este se conecte com o sistema de validação do Twiter e retorne se o usuário está autenticado ou não.

Abaixo você poderá observar as alterações que foram realizadas na classe MainActivity.java.

```

MainActivity.java

public class MainActivity extends AppCompatActivity {

    // Note: Your consumer key and secret should be obfuscated in your source code before shipping.
    private static final String TWITTER_KEY = "erPcLKzQlplug6Gtcm5OZkhzY";
    private static final String TWITTER_SECRET = "iCGsKJpON9sPpwUvulCbhPZlrhxTeVmfi0c9ggyHvp5vTAsBI5";
    private TwitterLoginButton loginButton;
    private TextView status;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        TwitterAuthConfig authConfig = new TwitterAuthConfig(TWITTER_KEY, TWITTER_SECRET);
        Fabric.with(this, new Twitter(authConfig));

        setContentView(R.layout.activity_main);

        loginButton = (TwitterLoginButton) findViewById(R.id.twitter_login_button);
        loginButton.setCallback(new Callback<TwitterSession>() {
            public void success(Result<TwitterSession> result) {
                TwitterSession session = result.data;
                TextView userName = (TextView) findViewById(R.id.userName);
                userName.setText("USUÁRIO: " + session.getUserName());
                TextView userId = (TextView) findViewById(R.id.userId);
                userId.setText("ID: " + session.getUserId());
            }
            public void failure(TwitterException exception) {
                Log.d("T Sign SDM3", "Erro ao efetuar login", exception);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        loginButton.onActivityResult(requestCode, resultCode, data);
    }
}

```


2 ANEXO

O código fonte deste projeto está disponível no servidor do GitHub através do endereço https://github.com/AlexStocco/Twitter_Sign_SDM3/.