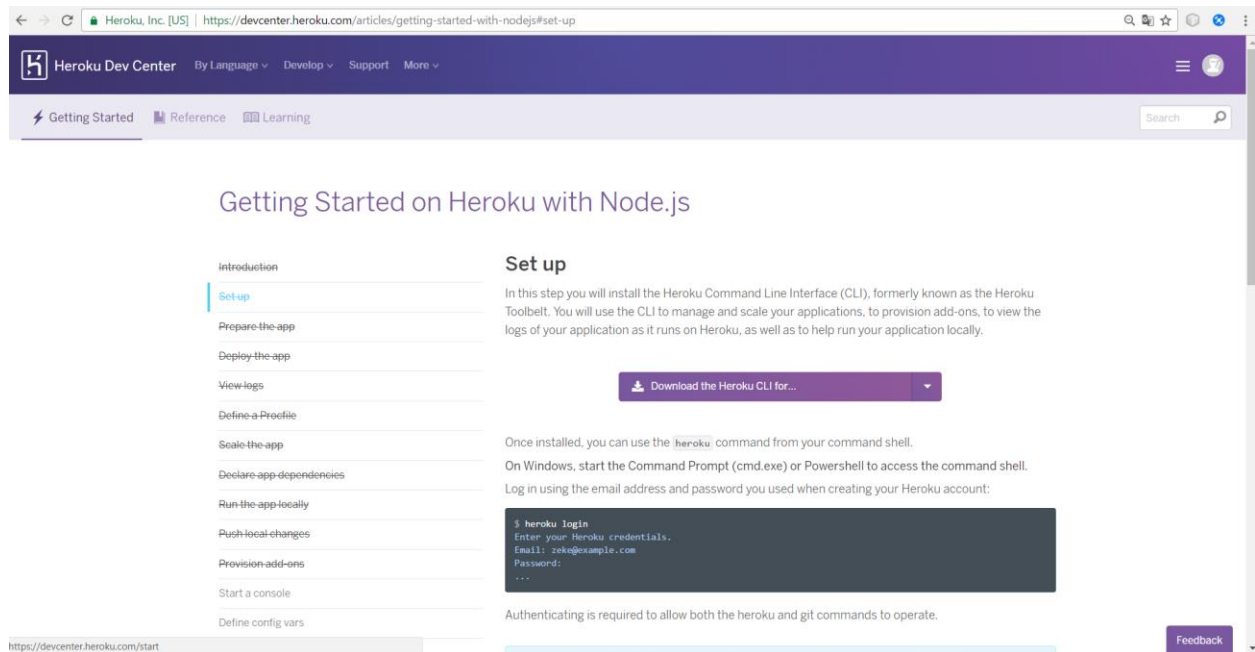


Configurando mi propio endpoint en mi Servidor

Configuración de Heroku (Node.js)

Seguir todos los pasos de la página:

<https://devcenter.heroku.com/articles/getting-started-with-nodejs#introduction>



Instalar las librerías:

- npm install --save --save-exact body-parser
- npm install --save --save-exact firebase

Configurar el archivo con las librerías y métodos necesarios: index.js

```
var express = require('express');
var app = express();
var bodyParser = require('body-parser')
var firebase = require("firebase");

app.set('port', (process.env.PORT || 5000));

app.use(bodyParser.json()); //soporte para codificar
app.use(bodyParser.urlencoded({ extended: true})); //soporte para decodificar mi url

firebase.initializeApp({
  serviceAccount: "PetagramModulo4-dd376205182f.json",
  databaseURL: "https://petagrammodulo4.firebaseio.com"
});

app.use(express.static(__dirname + '/public'));

// views is directory for all template files
app.set('views', __dirname + '/views');
app.set('view engine', 'ejs');

app.get('/android', function(request, response) {
  response.render('pages/index');
});

var tokenDeviceURI = "token-device";
app.post('/' + tokenDeviceURI, function(request, response) {
  //response.send(request.body.token);

  var token = request.body.token;
  var usuariolInstagram = request.body.usuariolInstagram;

  var db = firebase.database();
  var tokenDevices = db.ref(tokenDeviceURI).push();
  tokenDevices.set({
    token: token,
    usuariolInstagram: usuariolInstagram
  });

  var path = tokenDevices.toString();
  var pathSplit = path.split(tokenDeviceURI + "/");
  var idAutoGenerado = pathSplit[1];

  var respuesta = generarRespuestaAToken(db, idAutoGenerado);
  response.setHeader("Content-type", "application/json");
  response.send(JSON.stringify(respuesta));

  response.send(request.body.token);
```

```

});

function generarRespuestaAToken(db, idAutoGenerado){

    var respuesta = {};
    var usuario = "";
    var ref = db.ref(tokenDeviceURI);
    ref.on("child_added", function(snapshot, preChildKey){
        usuario = snapshot.val();
        respuesta = {
            id: idAutoGenerado,
            token: usuario.token,
            usuariolInstagram: usuario.usuariolInstagram
        };
    })

    return respuesta;
}

app.listen(app.get('port'), function() {
    console.log('Node app is running on port', app.get('port'));
});

```

Confirmar los cambios

git add .

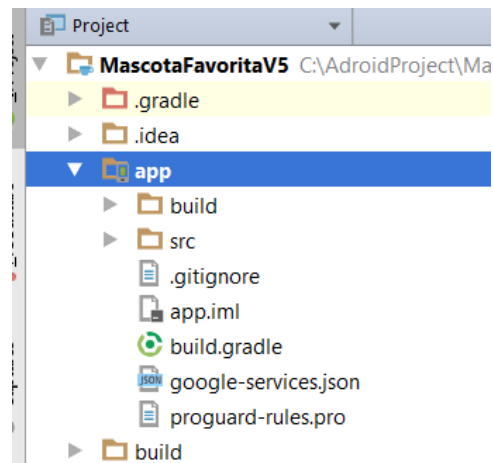
git commit -m "Path /implementado el repositorio v4"

git push heroku master

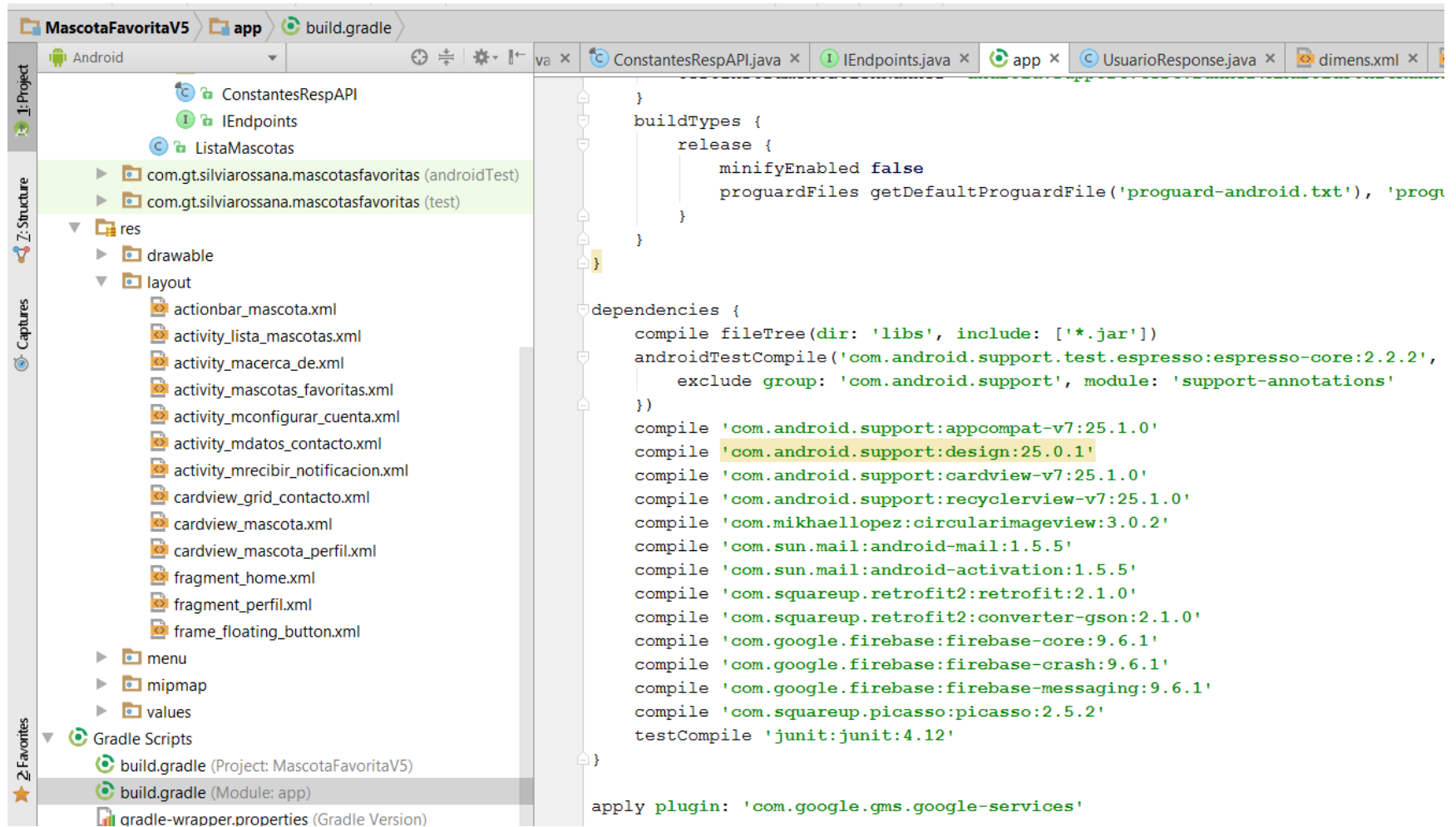
heroku open android

Configuración y Código Principal para la Aplicación

- Registrar el id de la aplicación en google, crear el SHA 1 y pegar el archivo de configuración de servicio que genera google en el proyecto: **google-service.json**



- Agregar las librerías correspondientes de google-services y firebase en gladle:





MascotaFavoritaV5 build.gradle

Android

1: Project

2: Structure

Captures

3: Favorites

ConstantesRespAPI

IEndpoints

ListaMascotas

com.gt.silviarossana.mascotasfavoritas (androidTest)

com.gt.silviarossana.mascotasfavoritas (test)

res

drawable

layout

actionbar_mascota.xml

activity_lista_mascotas.xml

activity_macerca_de.xml

activity_mascotas_favoritas.xml

activity_mconfigurar_cuenta.xml

activity_mdatos_contacto.xml

activity_mrecibir_notificacion.xml

cardview_grid_contacto.xml

cardview_mascota.xml

cardview_mascota_perfil.xml

fragment_home.xml

fragment_perfil.xml

frame_floating_button.xml

menu

mipmap

values

Gradle Scripts

build.gradle (Project: MascotaFavoritaV5)

va x

ConstantesRespAPI.java x

IEndpoints.java x

app x

MascotaFavoritaV5 x

Usuari...

```
// Top-level build file where you can add configuration options common to all sub-projects/modules

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.2.2'
        classpath 'com.google.gms:google-services:3.0.0'

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

allprojects {
    repositories {
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

- Agregar los permisos y servicios necesarios a:

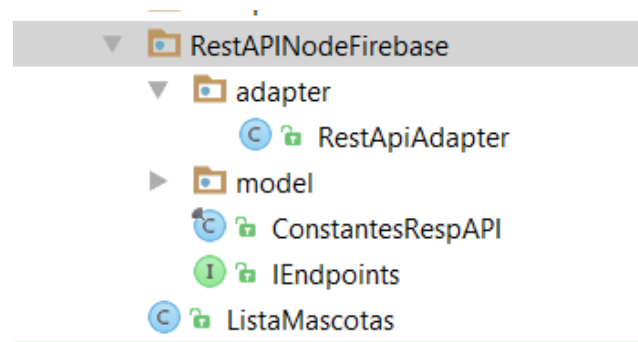
[AndroidManifest.xml](#)

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

<service
    android:name=".db.NotificationIDTokenService" android:exported="true">
    <intent-filter>
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT" />
    </intent-filter>
</service>

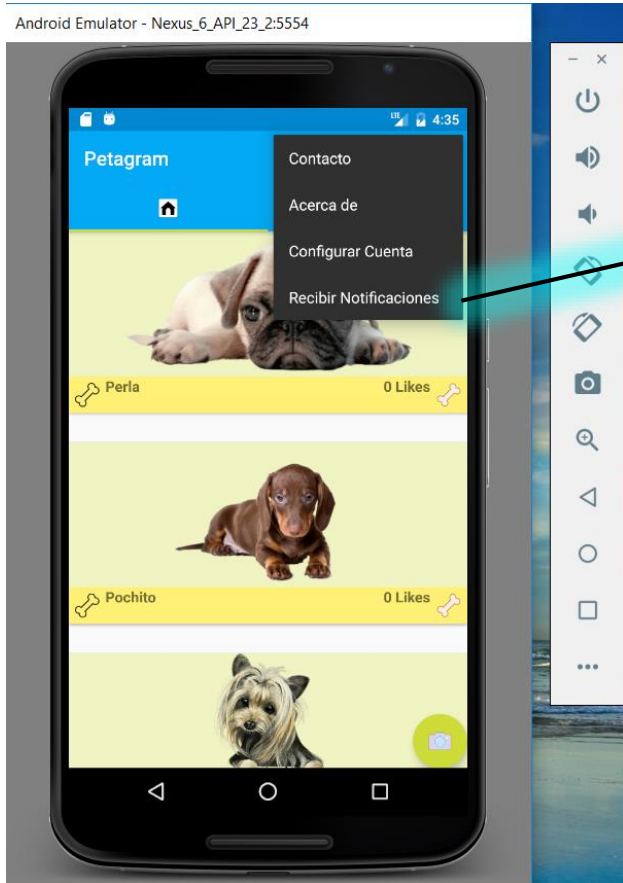
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

Las clases de conexión del API REST:

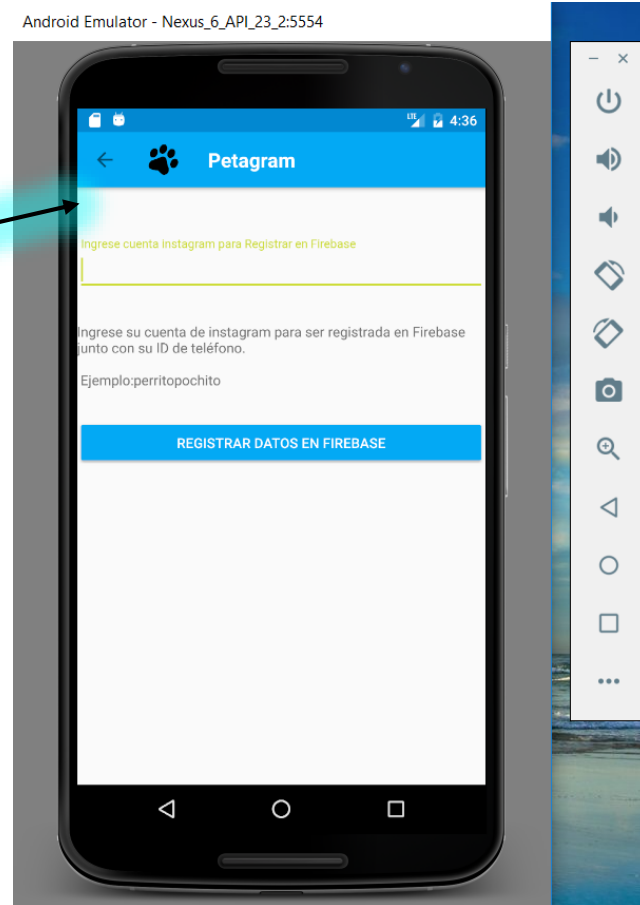


IMÁGENES DE LA APLICACIÓN

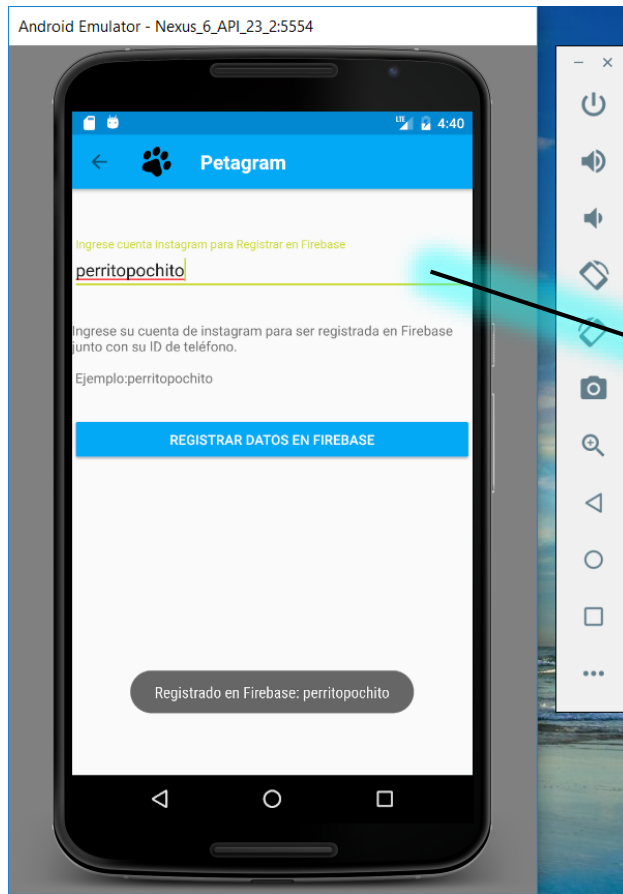
PÁGINA PRINCIPAL



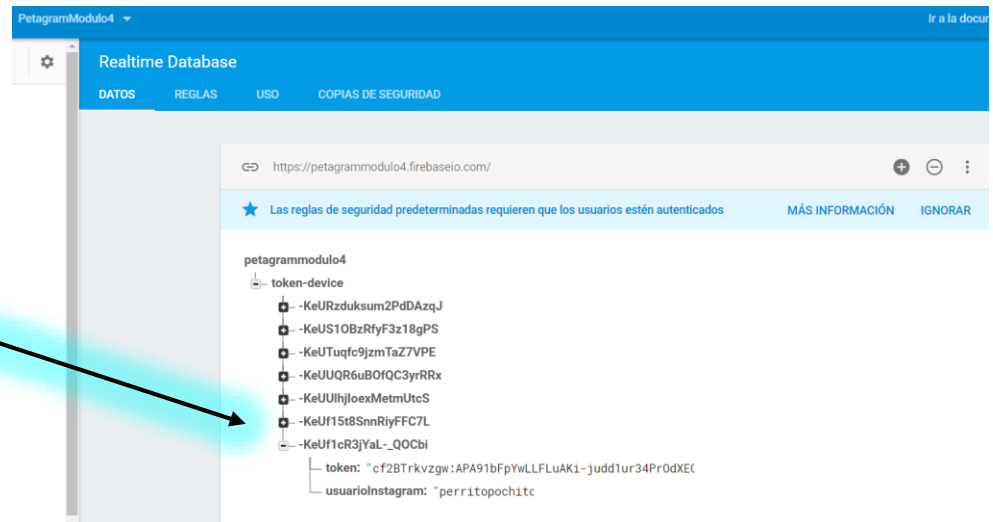
REGISTRAR EN FIREBASE ID TELEFONO Y USUARIO DE INSTAGRAMA



REGISTRAR EN FIREBASE ID TELEFONO Y USUARIO DE INSTAGRAMA

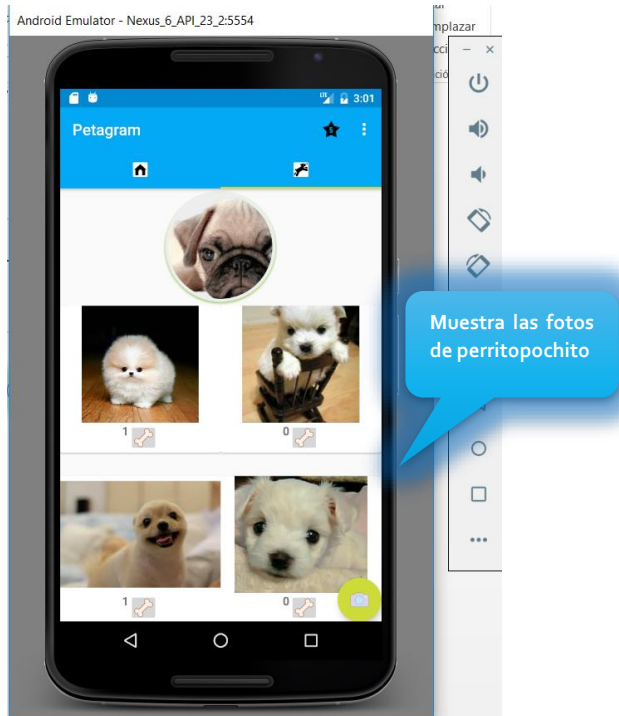


BASE DE DATOS DE GOOGLE-FIREBASE EN TIEMPO REAL

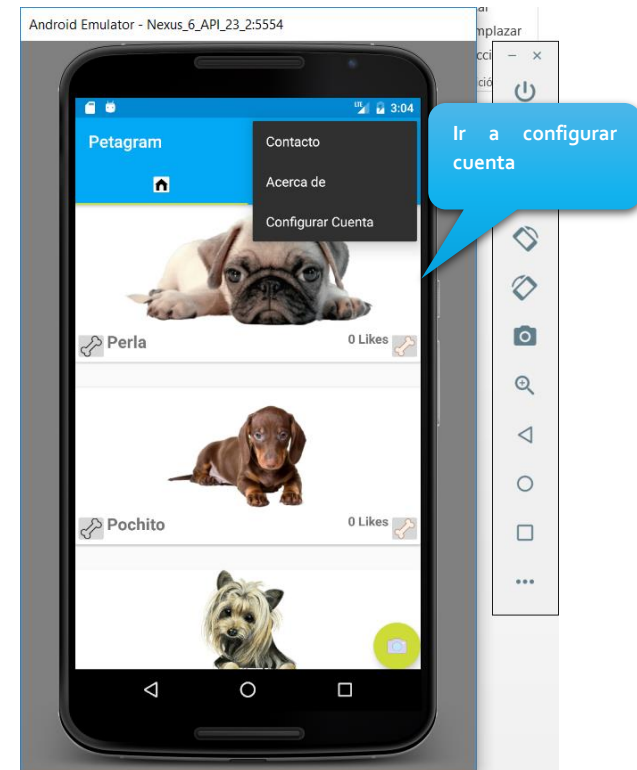


PÁGINA PRINCIPAL FRAGMENT PERFIL

<https://www.instagram.com/perritopochito/>



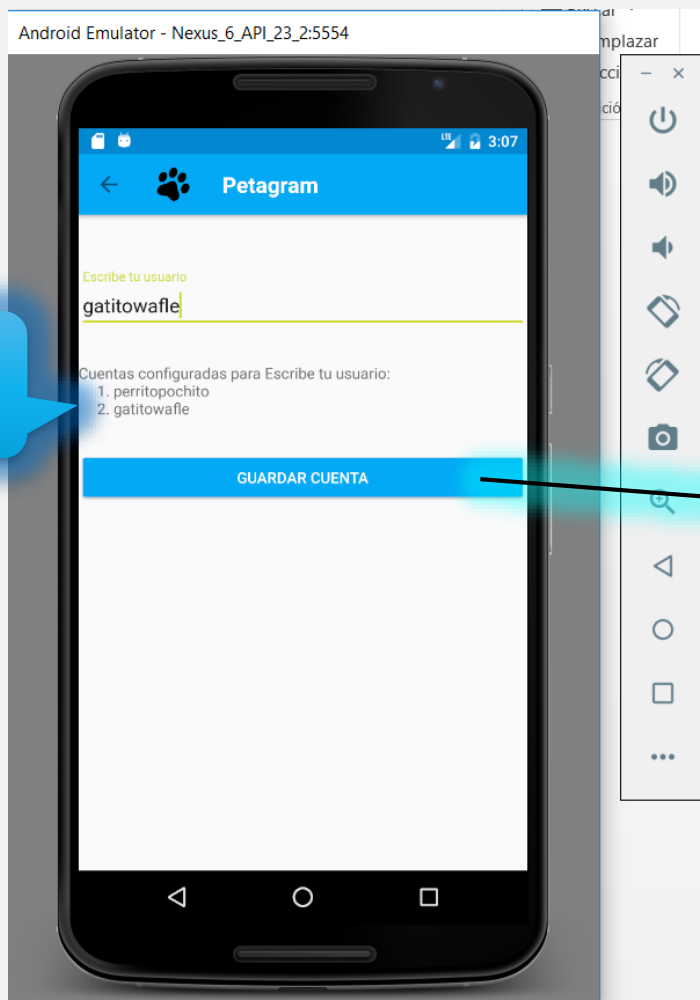
PÁGINA PRINCIPAL – CONFIGURAR CUENTA



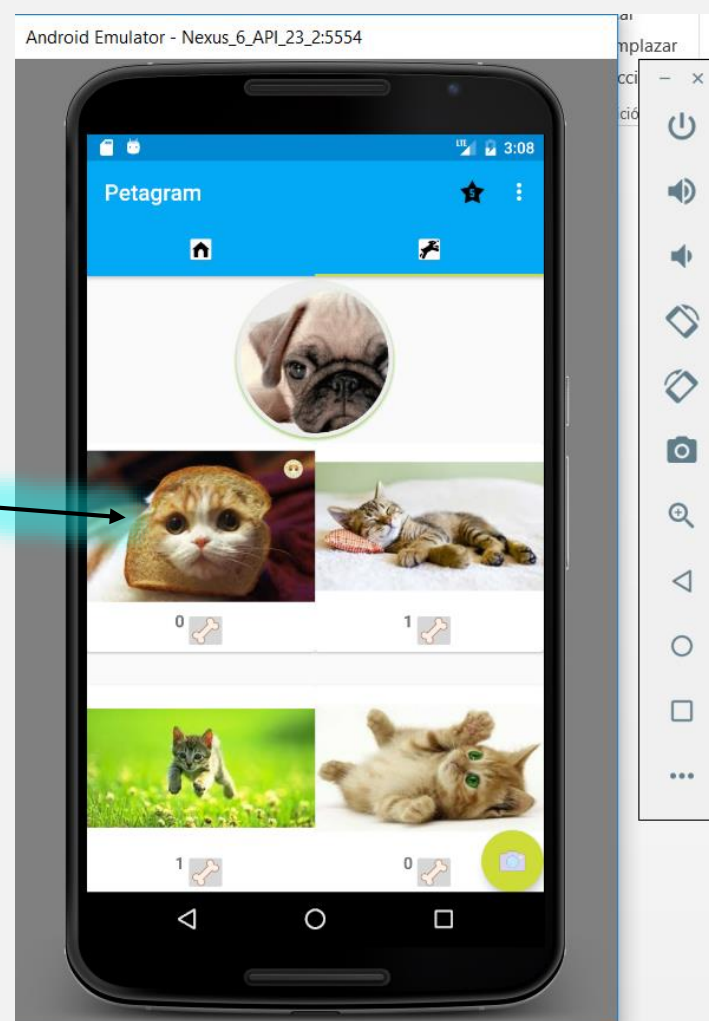
PÁGINA CONFIGURAR CUENTA

<https://www.instagram.com/gatitowafle/>

A estos nombres de cuentas tiene permisos la aplicación



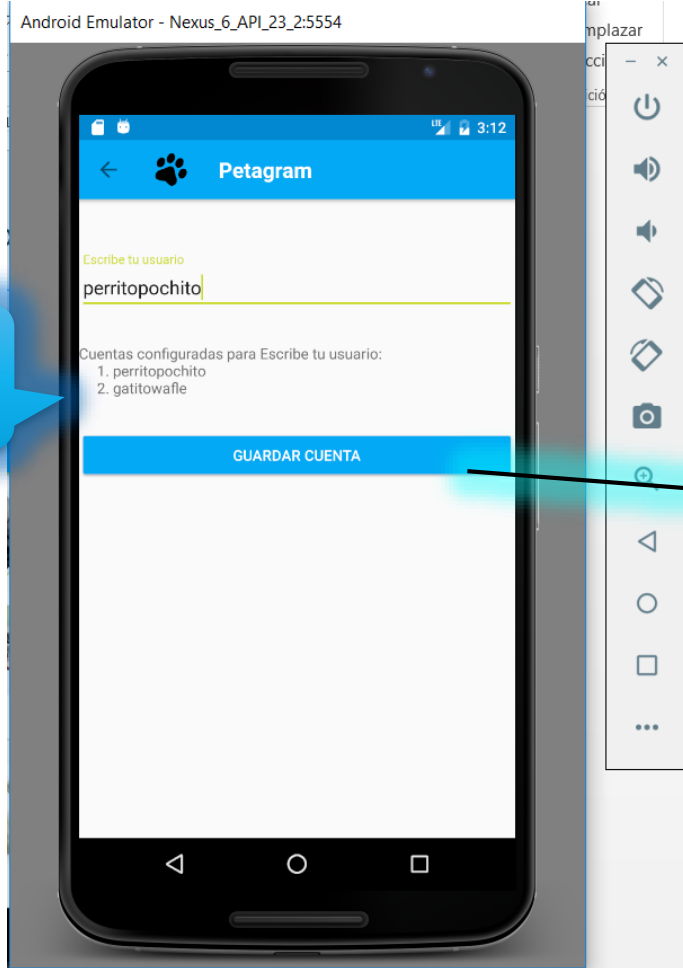
CAMBIO DE FOTOS EN INSTAGRAM DE GATITOWAFLE



PÁGINA PRINCIPAL FRAGMENT PERFIL

<https://www.instagram.com/perritopochito/>

A estos nombres de cuentas tiene permisos la aplicación



CAMBIO DE FOTOS EN INSTAGRAM DE PERRITPOCHITO

