

# Configuración endpoint en servidor

Implementación servicio POST Registro dispositivos y cuentas instagram

El servicio se implementó con Heroku y NodeJS

URL POST

<https://secret-beyond-94805.herokuapp.com/registrar-usuario>

datos de entrada

- ✓ Token
- ✓ id\_usuario\_instagram

//////////

## Configuración servicio en Heroku con nodeJS

```
var express = require('express');
```

```
var app = express();
```

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

```
var bodyParser = require("body-parser");
```

```
app.use(bodyParser.json()); //soporte para codificar json
```

```
app.use(bodyParser.urlencoded({ extended: true })); //soporte para decodificar las url
```

```
var firebase = require("firebase");
```

```
firebase.initializeApp({
```

```
  serviceAccount: "miuser",
```

```
  databaseURL: "urlDatabase"
```

```
});
```

```
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');
});

//Service POST
//https://secret-beyond-94805.herokuapp.com/regarstrar-usuario
//token
//regarstrar-usuario

var tokenDevicesURI_RegisterUser = "regarstrar-usuario";

app.post('/') + tokenDevicesURI_RegisterUser, function(request, response){
  var token = request.body.token;
  var id_usuario_instagram = request.body.id_usuario_instagram;

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

  var path = tokenDevices.toString(); //https://petagram-9e8df.firebaseio.com/regarstrar-
  usuario/-KfX73UtlpfgtOiwC2CF
  var pathSplit = path.split(tokenDevicesURI_RegisterUser + "/");
  var idAutoGenerado = pathSplit[1];

  var respuesta = generarRespuestaRegisterUser(db, idAutoGenerado);

  response.setHeader("Content-Type", "application/json");
  response.send(JSON.stringify(respuesta)); //imprime en json pero formato string
  //response.send(request.body.token);

});

```

```

function generarRespuestaRegisterUser(db, idAutoGenerado ){
    var respuesta = {};
    var usuario = "";
    var ref = db.ref("registrar-usuario");
    ref.on("child_added",function(snapshot, prevChildKey){
        usuario = snapshot.val();
        respuesta = {
            id: idAutoGenerado,
            token: usuario.token,
            id_usuario_instagram: usuario.id_usuario_instagram
        }

    });

    return respuesta;
}

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

```

//////////

Para finalizar se realiza despliegue del servicio a Heroku.

```

git add .
git commit -m "Implementando servicio POST "
git push heroku master

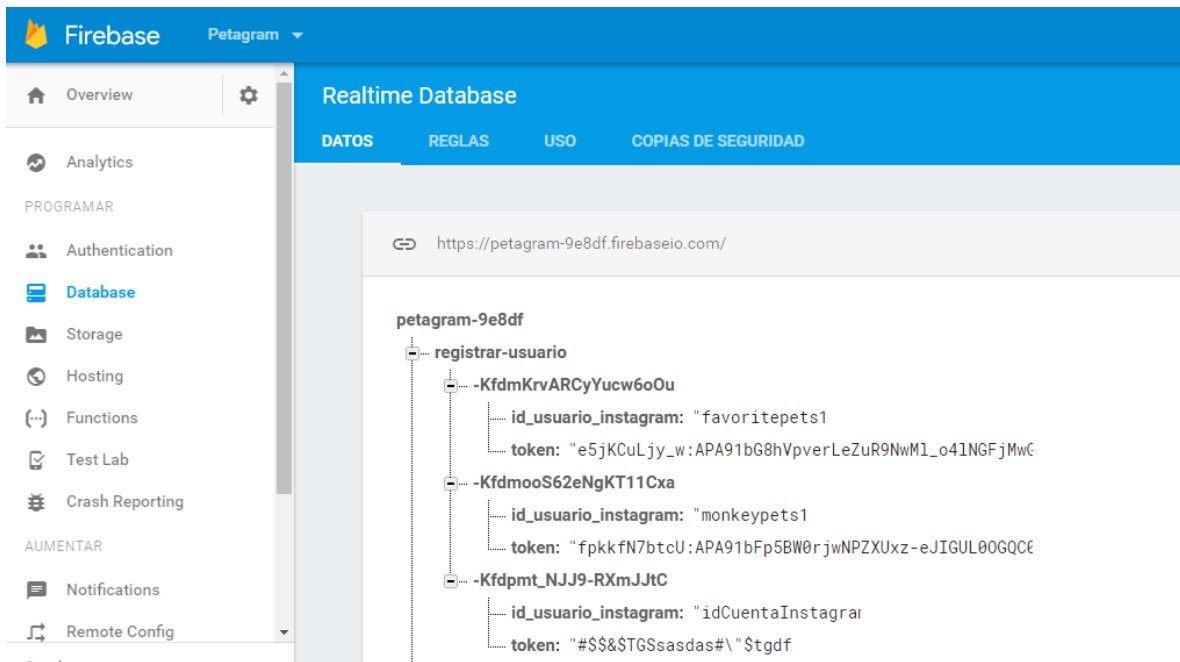
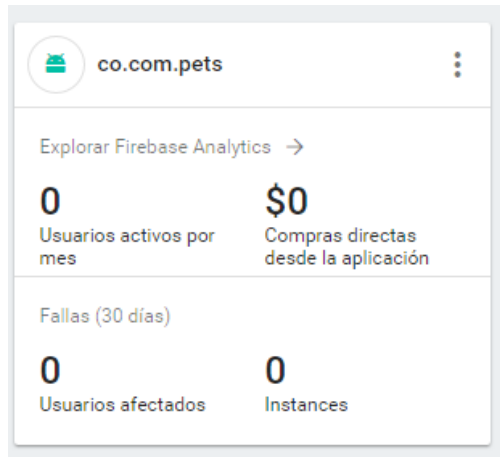
```

## Almacenamiento datos

Se hace uso de la Base de datos en **FIREBASE**

### Pasos


1. Se realiza registro del paquete
  - a. co.com.pets



Consumiendo servicio POST en heroku

Resultado

## ► RegistrarInstagram


POST  https://secret-beyond-94805.herokuapp.com/registrar-usuario

Authorization Headers (1) Body ● Pre-request Script Tests

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

<input checked="" type="checkbox"/>	token	###\$TGSsasd\$#"stgdfg"
<input checked="" type="checkbox"/>	id_usuario_instagram	idCuentaInstagram
	key	value

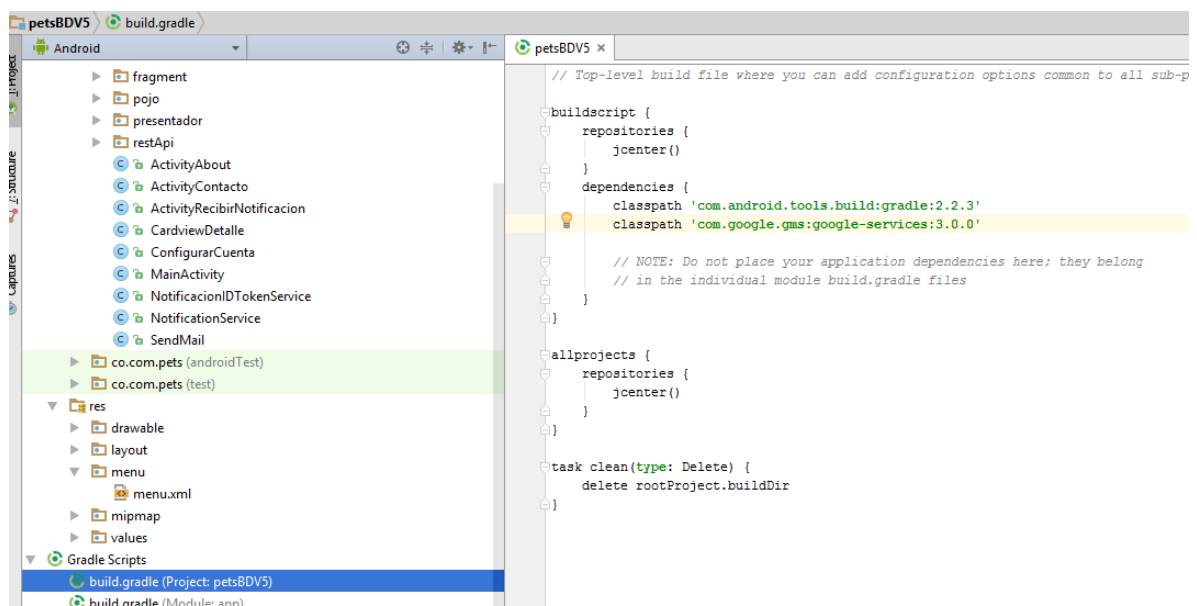
Body Cookies Headers (8) Tests

Pretty Raw Preview JSON 

```
1 {
2   "id": "-Kfdpmt_NJJ9-RXmJJtC",
3   "token": "###$TGSsasd$#"stgdfg",
4   "id_usuario_instagram": "idCuentaInstagram"
5 }
```

## Implementado la app en proyecto android

### 1. configuración librerías

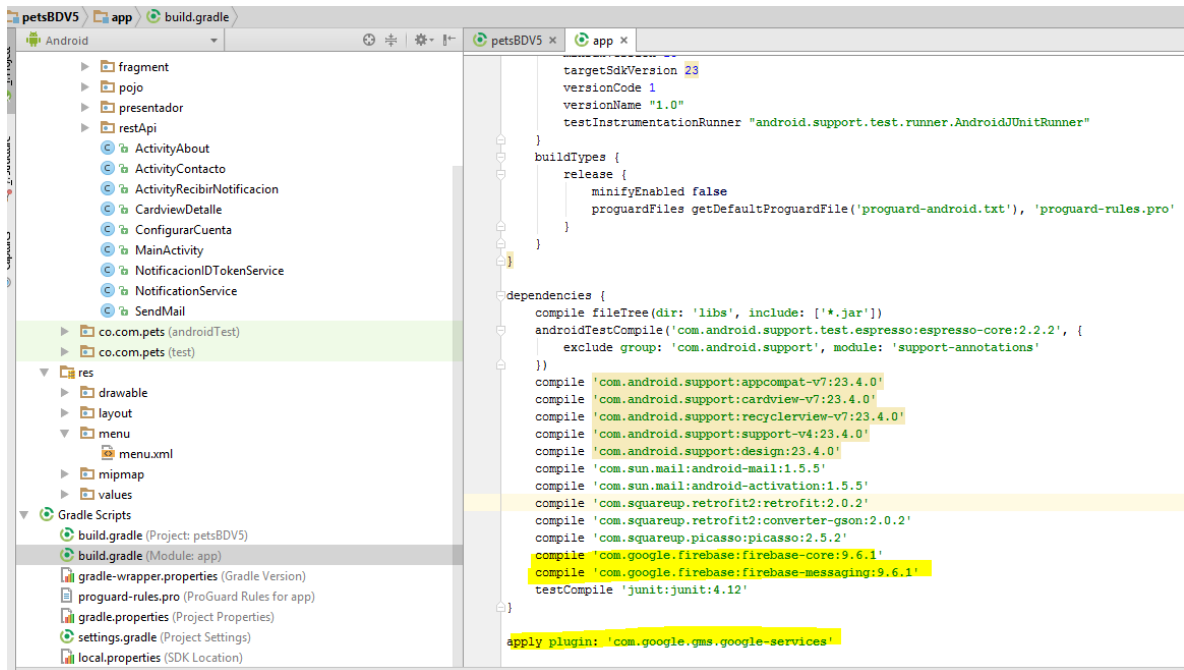
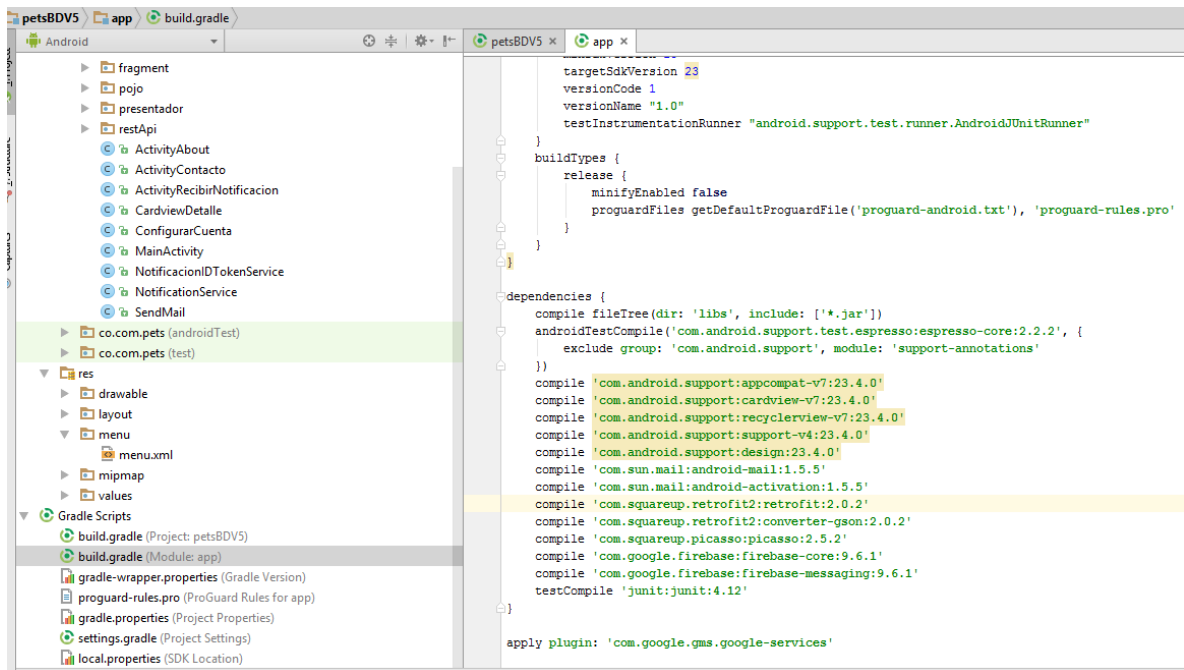


The screenshot shows the Android Studio interface. On the left, the 'Project' tab displays the project structure for 'petsBDV5'. The 'Gradle Scripts' folder is expanded, showing 'build.gradle (Project: petsBDV5)' selected. On the right, the 'build.gradle' file is open, showing the following configuration:

```
// Top-level build file where you can add configuration options common to all sub-p
buildscript {
    repositories {
        jcenter()
    }
    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
}

allprojects {
    repositories {
        jcenter()
    }
}

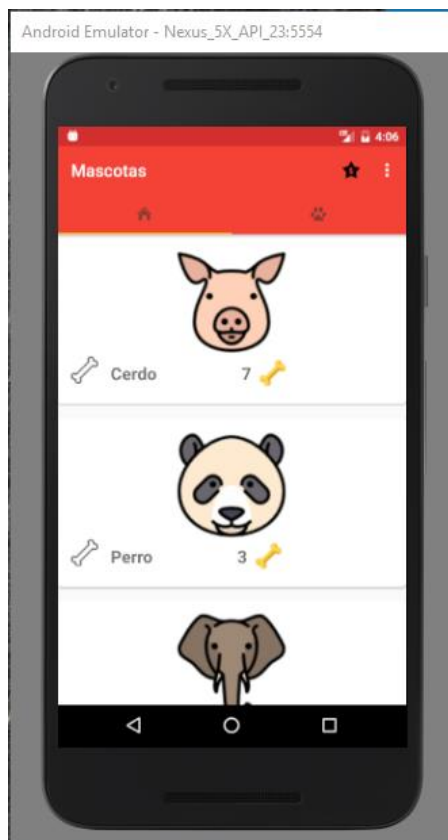
task clean(type: Delete) {
    delete rootProject.buildDir
}
```



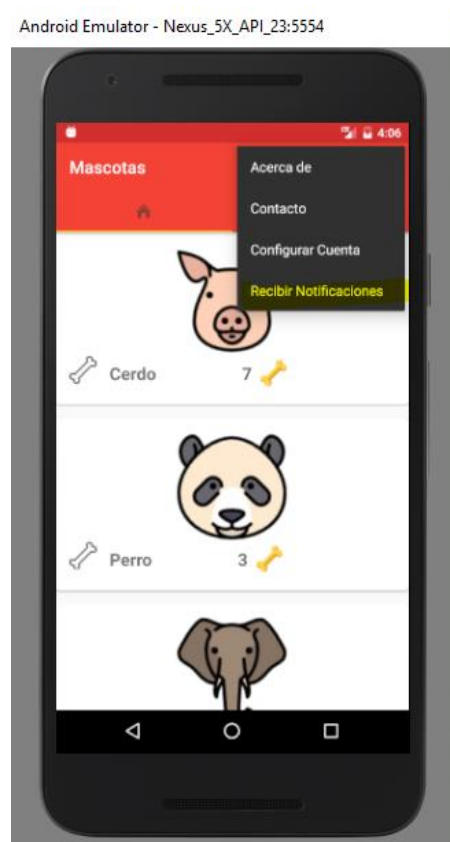
## Resultado aplicación

Registro Dispositivo y cuenta instagram

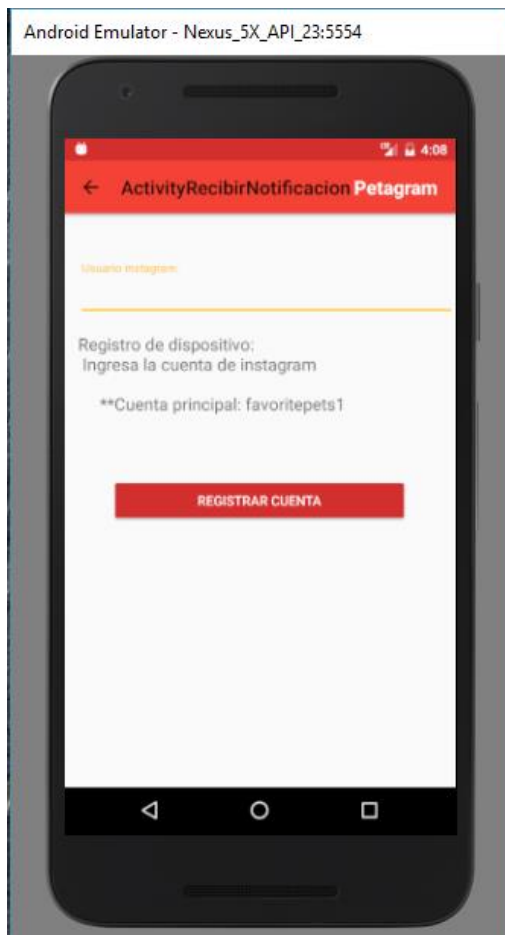
Aplicación



Menú



## Pantalla recibir notificación



## Cuentas registradas en BD

