

### Tarea 3: Configurando mi propio Endpoint en mi Servidor

Se realizó la configuración de firebase y se agregó el proyecto.

#### Es un placer verte de nuevo por Firebase

Sigue usando Firebase para crear tus aplicaciones con algunos de los recursos que encontrarás a continuación.

[Documentación](#) <> [Código de ejemplo](#) = [Referencia de la API](#) [Asistencia](#)

Tus proyectos que usan Firebase

**PetCare**  
  
petcare-feb9d.firebaseio.com  
1 aplicación

CREAR N

#### Firebase

PetCare

- Analytics
- DEVELOP
- Authentication
- Database
- Storage
- Hosting
- Test Lab
- Crash Reporting
- GROW
- Notifications
- Remote Config

#### Overview

Aplicaciones móviles PetCare

**mx.octaviocervantes.mypetcare**

Analytics (últimos 30 días)

<b>2</b> Usuarios activos/mes	<b>0 \$</b> Compras en la aplicación
----------------------------------	---

Bloqueos (30 días)

<b>0</b> Usuarios afectados	<b>0</b> Errores
--------------------------------	---------------------

## Realtime Database

DATOS REGLAS USO COPIAS DE SEGURIDAD

<https://petcare-feb9d.firebaseio.com/>

★ Las reglas de seguridad predeterminadas requieren que los usuarios estén autenticados

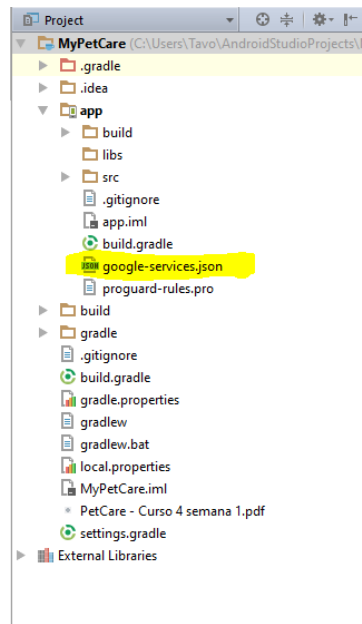
**petcare-feb9d**

- registrar-usuario
  - KVJdBFRVeTKiFddhf58
  - KVJdMRNQAAAnNUxexvzB
  - KVJd\_CADF4cL\_YWPzZV

```

21
22 repositories {
23     jcenter()
24     maven {
25         url "https://maven.java.net/content/groups/public/"
26     }
27 }
28
29 dependencies {
30     compile fileTree(include: ['*.jar'], dir: 'libs')
31     testCompile 'junit:junit:4.12'
32     compile 'com.android.support:appcompat-v7:24.2.0'
33     compile 'com.android.support:design:24.2.0'
34     compile 'com.android.support:recyclerview-v7:24.2.0'
35     compile 'com.android.support:cardview-v7:24.2.0'
36     compile 'com.sun.mail:android-mail:1.5.5'
37     compile 'com.sun.mail:android-activation:1.5.5'
38     compile 'com.android.support:support-v4:24.2.0'
39     compile 'com.mikhaellopez:circularimageview:3.0.2'
40     compile 'com.squareup.retrofit2:retrofit:2.1.0'
41     compile 'com.squareup.retrofit2:converter-gson:2.1.0'
42     compile 'com.squareup.picasso:picasso:2.5.2'
43     compile 'com.google.firebase:firebase-messaging:9.6.1'
44     compile 'com.google.firebase:firebase-core:9.6.1'
45 }
46
47 apply plugin: 'com.google.gms.google-services'
48

```



```

50
51 <activity
52     android:name=".LoginUsuario"
53     android:label="Configurar Cuenta"
54     android:parentActivityName=".MascotasLista" >
55
56     <!-- Versiones anteriores a Android 4.0 -->
57     <meta-data
58         android:name="android.support.PARENT_ACTIVITY"
59         android:value=".MascotasLista" />
60 </activity>
61
62 <service
63     android:name=".notificaciones.NotificationService">
64     <intent-filter>
65         <action android:name="com.google.firebase.MESSAGING_EVENT"/>
66     </intent-filter>
67 </service>
68
69 <service
70     android:name=".notificaciones.NotificationIDService">
71     <intent-filter>
72         <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
73     </intent-filter>
74 </service>

```

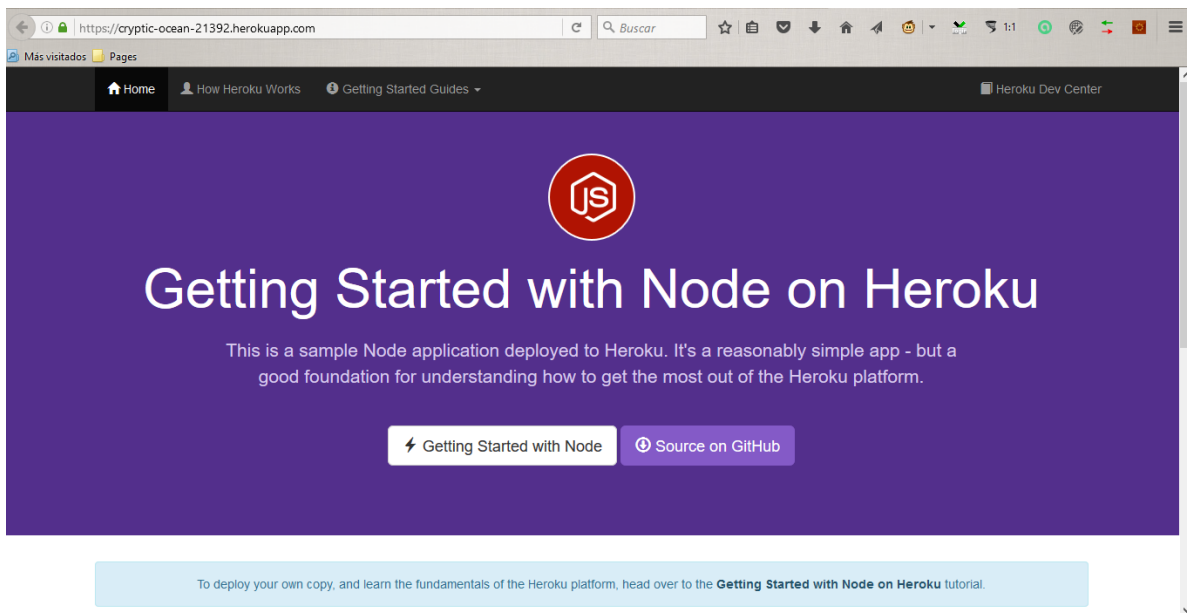
```

1 // Top-level build file where you can add configuration options common to all s
2
3 buildscript {
4     repositories {
5         jcenter()
6     }
7     dependencies {
8         classpath 'com.android.tools.build:gradle:2.1.2'
9         classpath 'com.google.gms:google-services:3.0.0'
10
11         // NOTE: Do not place your application dependencies here; they belong
12         // in the individual module build.gradle files
13     }
14 }
15
16 allprojects {
17     repositories {
18         jcenter()
19     }
20 }
21
22 task clean(type: Delete) {
23     delete rootProject.buildDir
24 }

```

## Configurando server Heroku

```
Nodejs command prompt
remote: ----- Installing binaries
remote:   engines.node (package.json): 5.9.1
remote:   engines.npm (package.json):  unspecified (use default)
remote:   Downloading and installing node 5.9.1...
remote:   Using default npm version: 3.7.3
remote: ----- Restoring cache
remote:   Loading 2 from cacheDirectories (default):
remote:   - node_modules
remote:   - bower_components (not cached - skipping)
remote: ----- Building dependencies
remote:   Installing node modules (package.json)
remote: ----- Caching build
remote:   Clearing previous node cache
remote:   Saving 2 cacheDirectories (default):
remote:   - node_modules
remote:   - bower_components (nothing to cache)
remote: ----- Build succeeded!
remote:   body-parser@1.15.2
remote:   |   ejs@2.4.1
remote:   |   express@4.13.3
remote:   |   firebase@3.5.2
remote: ----- Discovering process types
remote:   Procfile declares types -> web
remote: ----- Compressing...
remote:   Done: 13M
remote: ----- Launching...
remote:   Released v19
remote:   https://cryptic-ocean-21392.herokuapp.com/ deployed to Heroku
remote: Verifying deploy... done.
To https://git.heroku.com/cryptic-ocean-21392.git
4ec068e..e3e73b6 master -> master
```



## Configurando Endpoint “registrar-usuario”

```
25
26 var tokenDevicesURI = "registrar-usuario";
27 app.post('/' + tokenDevicesURI, function(request, response) {
28   var idusuario = request.body.idusuario;
29   var usuario = request.body.usuario;
30   var iddispositivo = request.body.iddispositivo;
31
32   var db = firebase.database();
33   var tokenRegistrar = db.ref(tokenDevicesURI).push();
34   tokenRegistrar.set({
35     iddispositivo: iddispositivo,
36     idusuario: idusuario,
37     usuario: usuario
38   });
39
40   var path = tokenRegistrar.toString();
41   var pathSplit = path.split(tokenDevicesURI + '/');
42   var idAutoGenerado = pathSplit[1];
43
44   var respuesta = generarRespuestaToken(db, idAutoGenerado);
45   response.setHeader("Content-Type", "application/json");
46   response.send(JSON.stringify(respuesta));
47 });
48
49 function generarRespuestaToken(db, idAutoGenerado){
50   var respuesta = {};
51   var usuario = "";
52   var ref = db.ref(tokenDevicesURI);
53   ref.on("child_added", function(snapshot, prevChildKey){
54     usuario = snapshot.val();
55     respuesta = {
56       id: idAutoGenerado,
57       iddispositivo: usuario.iddispositivo,
58       idusuario: usuario.idusuario,
59       usuario: usuario.usuario
60     };
61   });
62 }
63 return respuesta;
64 }
65
```

```
scotasLista.java × RestAPIFirebaseAdapter.java × ConstantesRestAPIFirebase.java ×
package mx.octaviocervantes.mypetcare.restAPI;

public class ConstantesRestAPIFirebase {
    public static final String TOKEN_URL = "https://cryptic-ocean-21392.herokuapp.com/";
    public static final String TOKEN = "registrar-usuario/";
}
```

```
scotasLista.java × RestAPIFirebaseAdapter.java × ConstantesRestAPIFirebase.java ×
package mx.octaviocervantes.mypetcare.restAPI.adapter;

import mx.octaviocervantes.mypetcare.restAPI.ConstantesRestAPIFirebase;
import mx.octaviocervantes.mypetcare.restAPI.EndpointsAPIFirebase;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class RestAPIFirebaseAdapter {

    public EndpointsAPIFirebase establecerConexionRestAPI() {
        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(ConstantesRestAPIFirebase.TOKEN_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build();

        return retrofit.create(EndpointsAPIFirebase.class);
    }
}
```

```

private void recibirNotificaciones(){
    String token = FirebaseInstanceId.getInstance().getToken();

    RestAPIFirebaseAdapter restAPIFirebaseAdapter = new RestAPIFirebaseAdapter();
    EndpointsAPIFirebase endpointsAPIFirebase = restAPIFirebaseAdapter.establecerConexionRestAPI();
    Call<UsuarioFirebaseResponse> usuarioFirebaseResponseCall = endpointsAPIFirebase.registrarTokenId(
        token, idUsuario, sUsuario, idDispositivo);

    usuarioFirebaseResponseCall.enqueue(new Callback<UsuarioFirebaseResponse>() {
        @Override
        public void onResponse(Call<UsuarioFirebaseResponse> call, Response<UsuarioFirebaseResponse> response) {
            UsuarioFirebaseResponse usuarioFirebaseResponse = response.body();
            Toast.makeText(getApplicationContext(), "Información enviada a la base de datos de Firebase",
                Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onFailure(Call<UsuarioFirebaseResponse> call, Throwable t) {
            // Manejar el error
        }
    });
}

```

```

package mx.octaviocervantes.mypetcare.restAPI;

import mx.octaviocervantes.mypetcare.restAPI.model.UsuarioFirebaseResponse;
import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.POST;

public interface EndpointsAPIFirebase {

    @FormUrlEncoded
    @POST(ConstantsRestAPIFirebase.TOKEN)
    Call<UsuarioFirebaseResponse> registrarTokenId(@Field("idusuario") String idUsuario,
        @Field("susuario") String sUsuario,
        @Field("iddispositivo") String idDispositivo);
}

```

```

package mx.octaviocervantes.mypetcare.restAPI.model;

public class UsuarioFirebaseResponse {
    private String id;
    private String idUsuario;
    private String idDispositivo;
    private String sUsuario;

    public UsuarioFirebaseResponse() {
    }

    public UsuarioFirebaseResponse(String id, String idUsuario, String sUsuario, String idDispositivo) {
        this.id = id;
        this.idDispositivo = idDispositivo;
        this.idUsuario = idUsuario;
        this.sUsuario = sUsuario;
    }

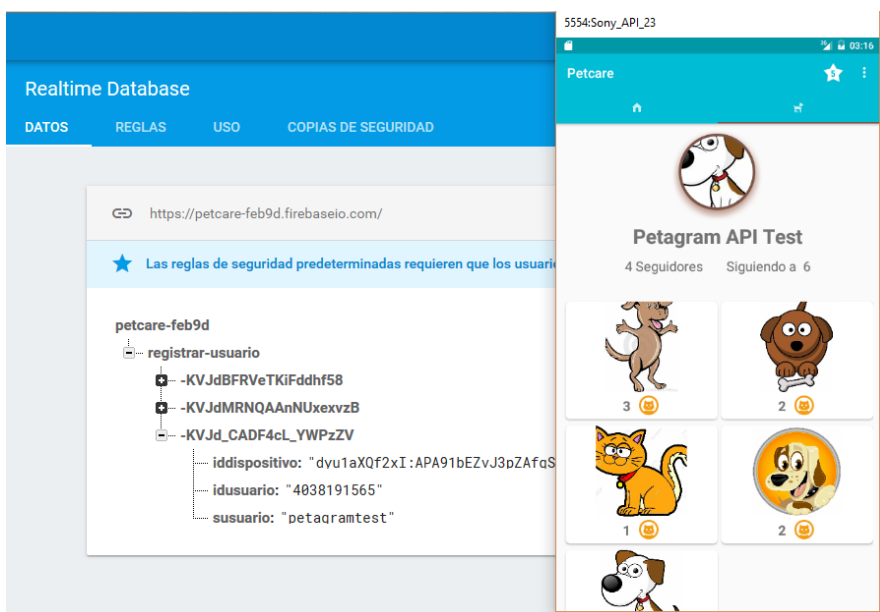
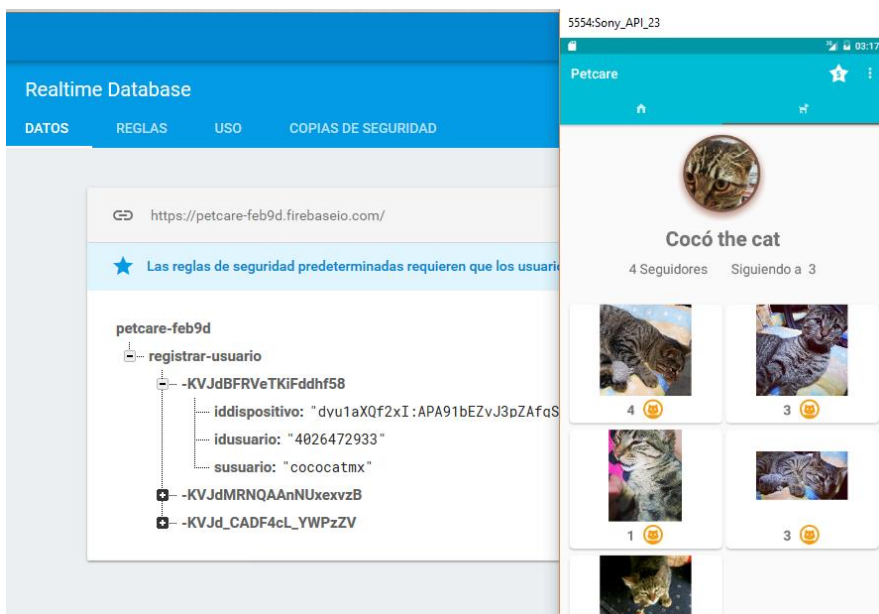
    public String getId() {
        return id;
    }

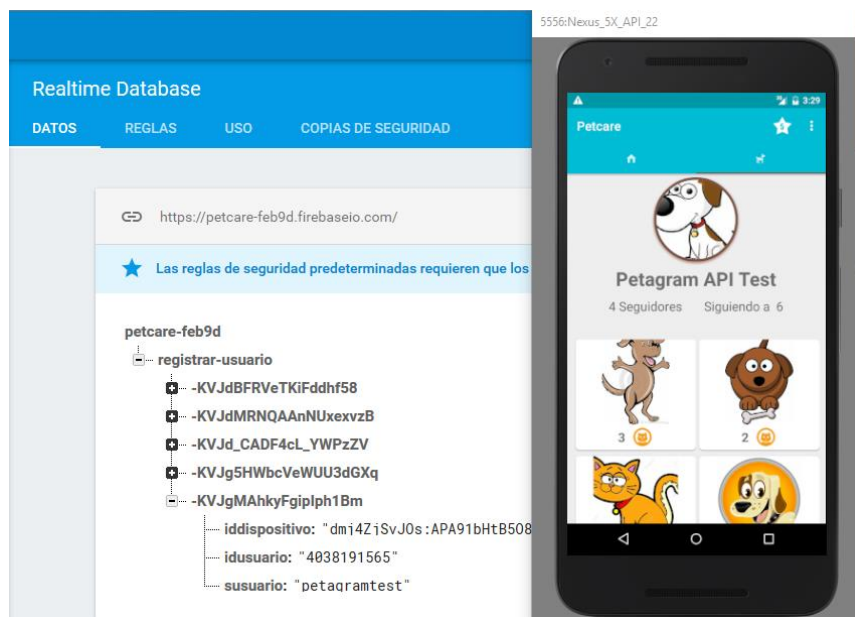
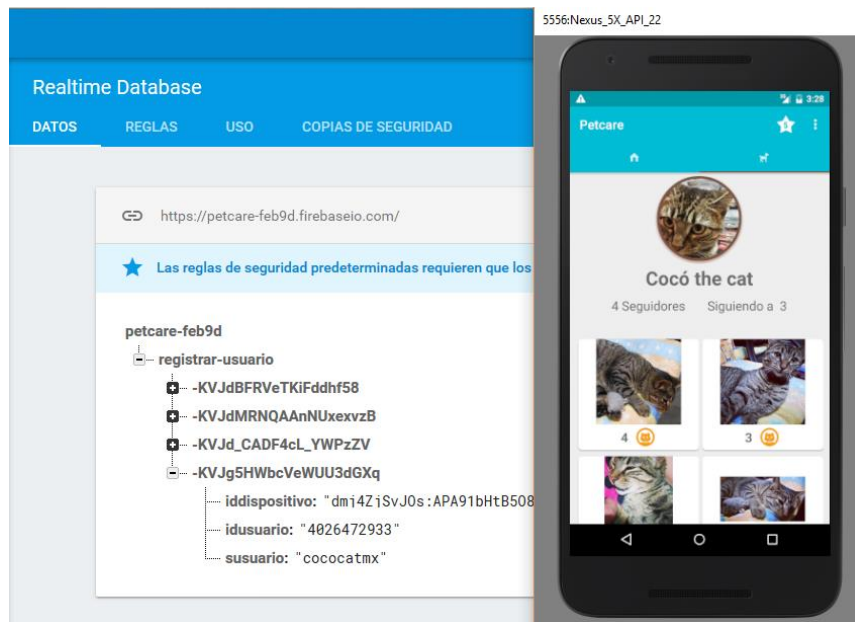
    public void setId(String id) {
        this.id = id;
    }

    public String getIdDispositivo() {
        return idDispositivo;
    }
}

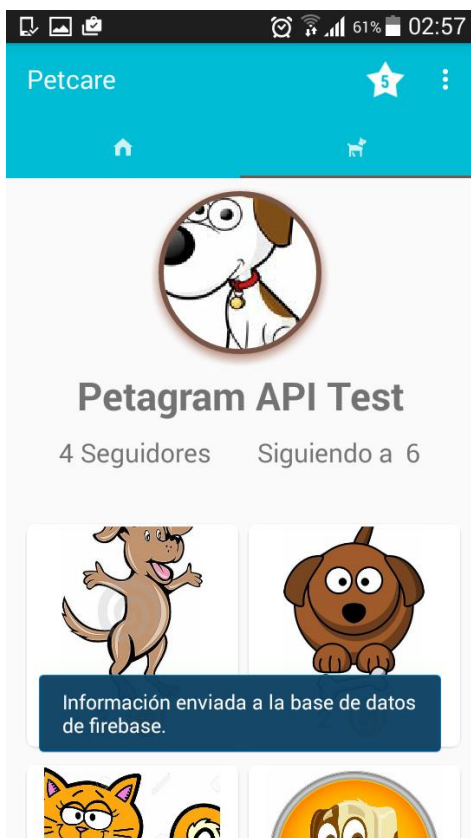
```

Aplicación corriendo en dos emuladores y un celular





Celular Samsung Grand Prime



petcare-feb9d

registrar-usuario

+ -KVJdBFRVeTKiFddhf58

+ -KVJdMRNQAAAnNUxexvzB

+ -KVJd\_CADF4cL\_YWPzZV

+ -KVJg5HWbcVeWUU3dGXq

+ -KVJgMAhkyFgiplph1Bm

-KVJnkxmsDkfFi6eD2j2

iddispositivo: "cwpz1j9X\_iQ:APA91bHwHSkj2ssIURcZJQSMdMSMQCqkS5C..."

idusuario: "4038191565"

susuario: "petagramtest"