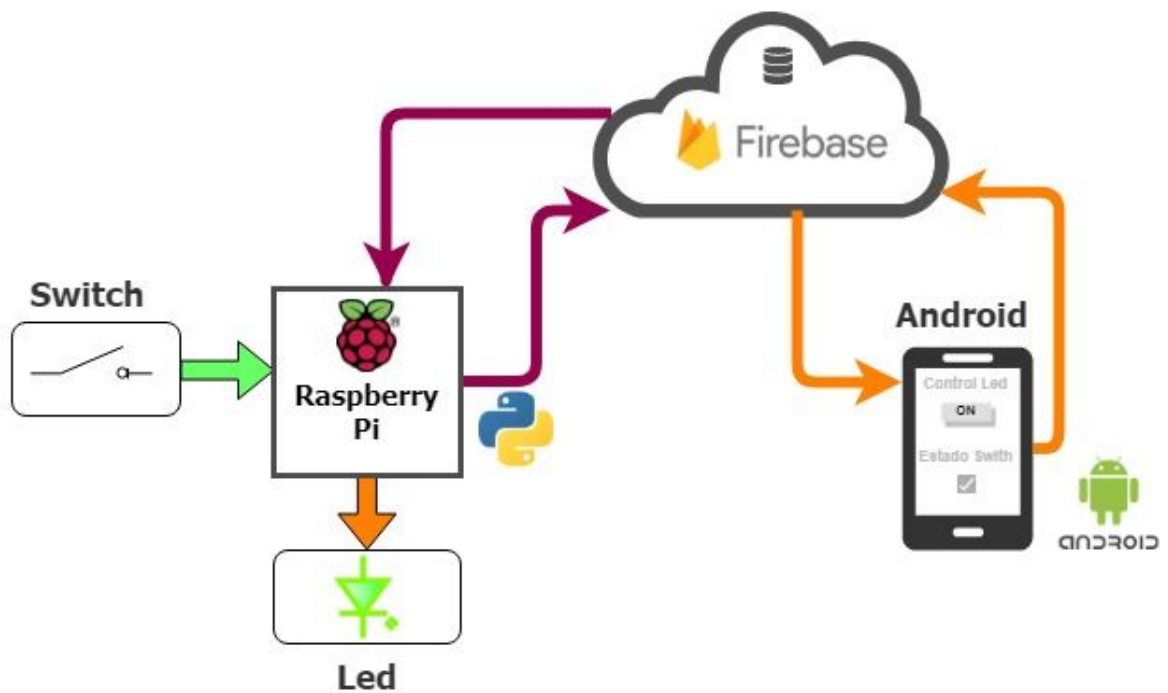
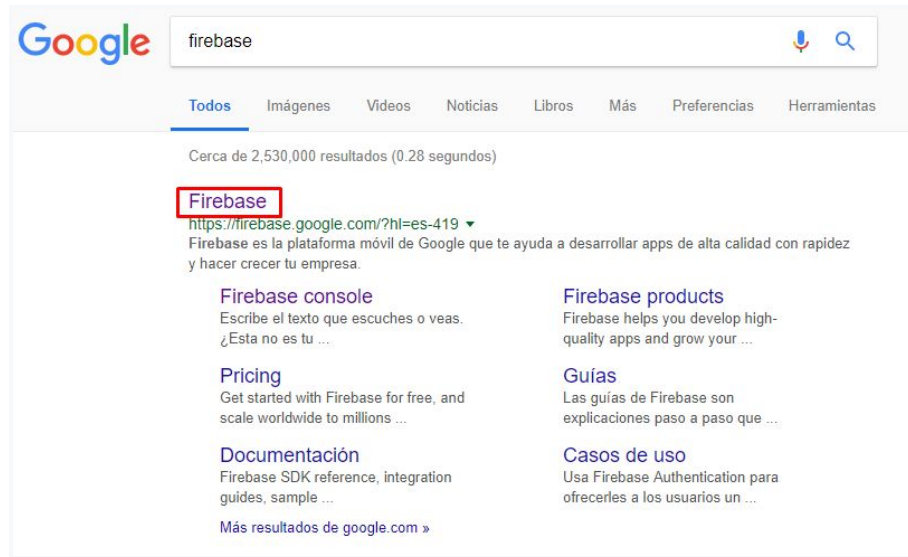


# RASPBERRY PI IOT CON PYTHON Y FIREBASE



# 1. CREAR UN PROYECTO EN FIREBASE

## Pasos para crear al proyecto



## Te damos la bienvenida a Firebase

Herramientas de Google para programar grandes apps,  
interactuar con tus usuarios y ganar más a través de los  
anuncios móviles.

[Más información](#) [Documentación](#) [Asistencia](#)

Proyectos recientes



Agregar proyecto



Explorar un proyecto de  
demostración

testled

testled-eb6bb

### Agregar un proyecto

Nombre del proyecto

RPI-DEMO

+ +

**Sugerencia:** Los proyectos  
llevan las apps a distintas  
plataformas [?](#)

ID del proyecto [?](#)

rpi-demo-e24dd

País/Región [?](#)

Colombia

En la configuración predeterminada, tus datos de Analytics mejorarán otras funciones de  
Firebase y otros productos de Google. Puedes controlar cómo se comparten tus datos  
de Analytics en cualquier momento desde la configuración. [Más información](#)

CANCELAR

CREAR PROYECTO



Creando tu proyecto...

RPI-DEMO

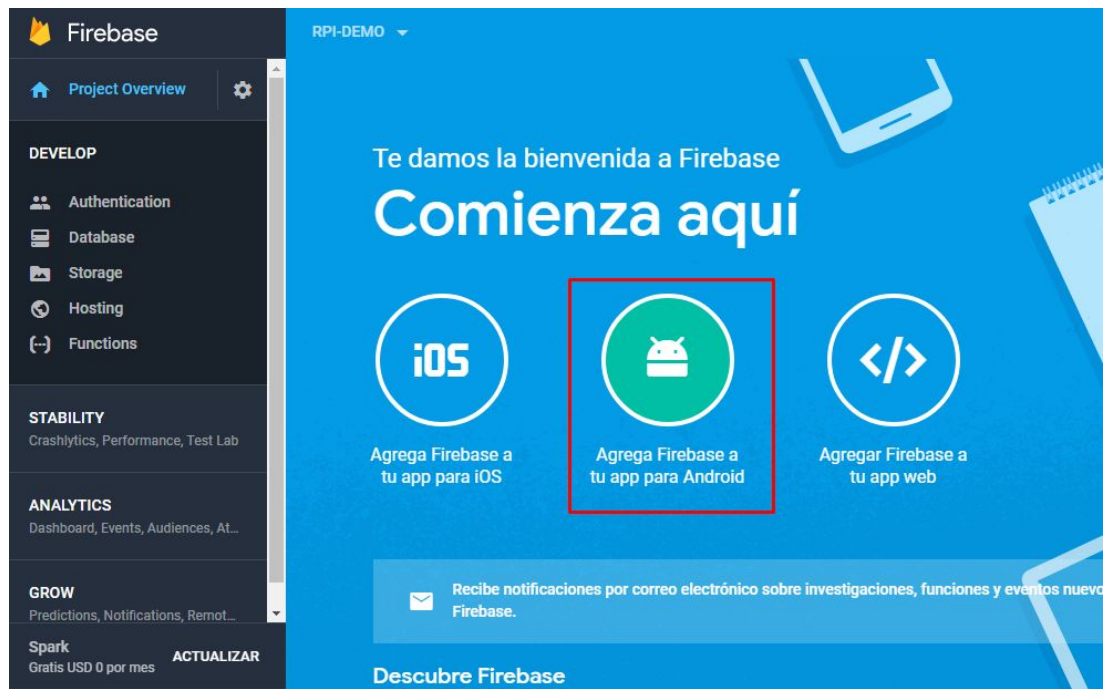


RPI-DEMO



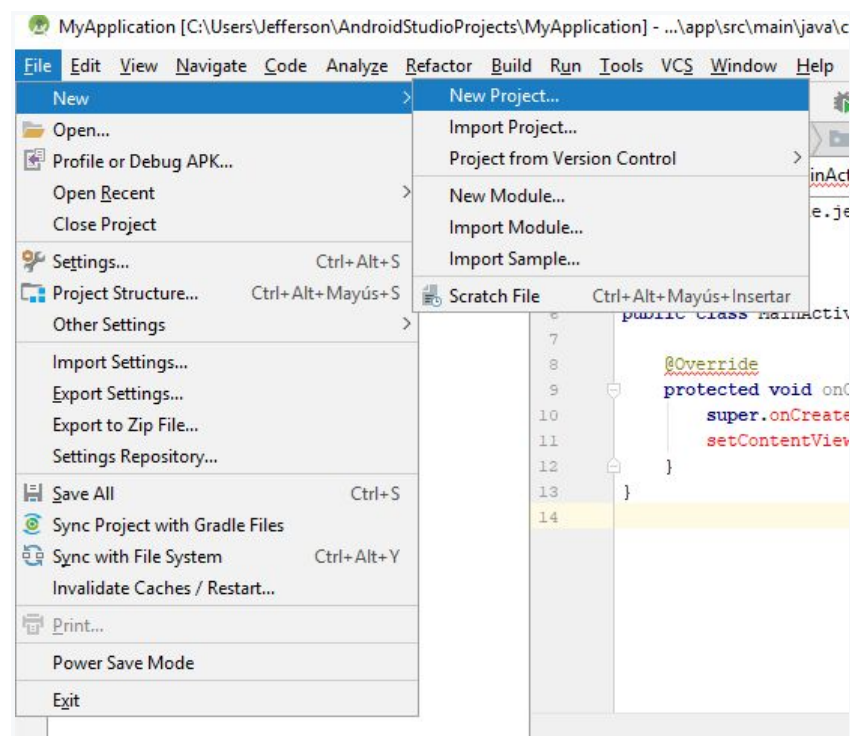
Tu proyecto nuevo está listo

CONTINUAR



Antes de seleccionar la opción de agregar “*Firebase a to app Android*”, vamos a crear un proyecto en el Android Studio

## 2. CREAR LA APLICACIÓN ANDROID



File Edit View Refactor Build Run Tools VCS Window Help

Create New Project

## Create Android Project

**Application name**  
RPI

**Company domain**  
rpi.app.com

**Project location**  
C:\Users\Jefferson\Documents\Android\RPI

**Package name**  
com.app.rpi.rpi

☐ Include C++ support  
☐ Include Kotlin support

Previous Next Cancel Finish

Create New Project

## Target Android Devices

**Select the form factors and minimum SDK**  
Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**  
API 15: Android 4.0.3 (IceCreamSandwich)  
By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)  
☐ Include Android Instant App support

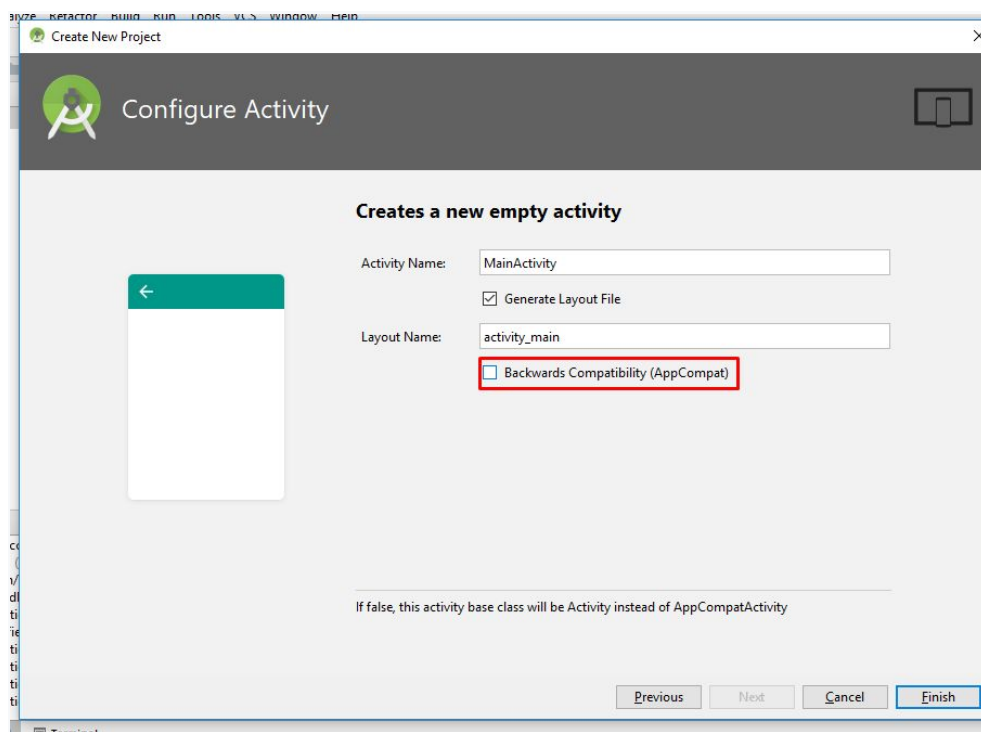
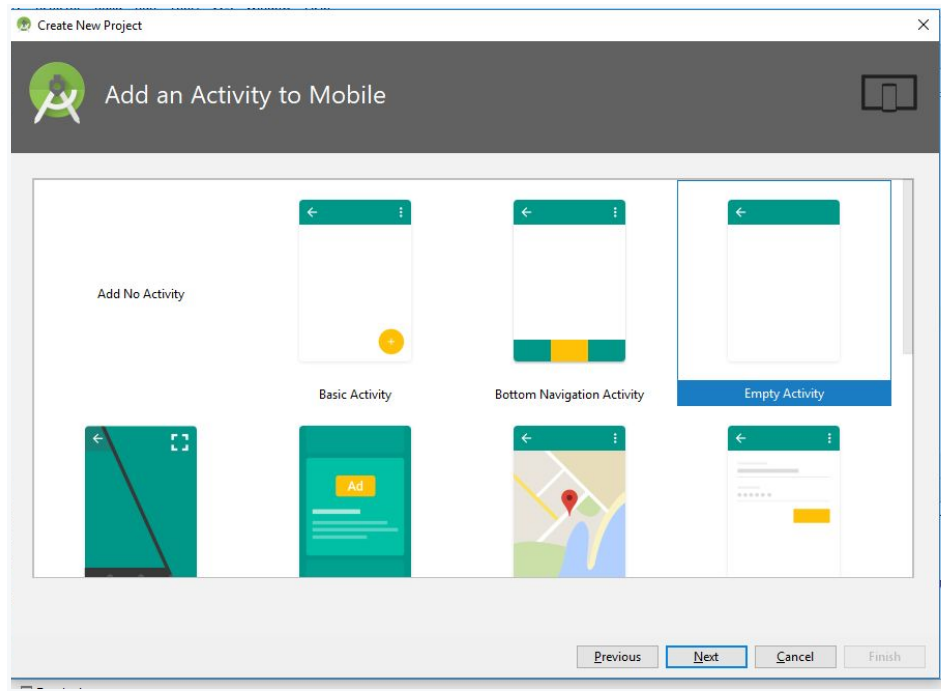
☐ **Wear**  
API 21: Android 5.0 (Lollipop)

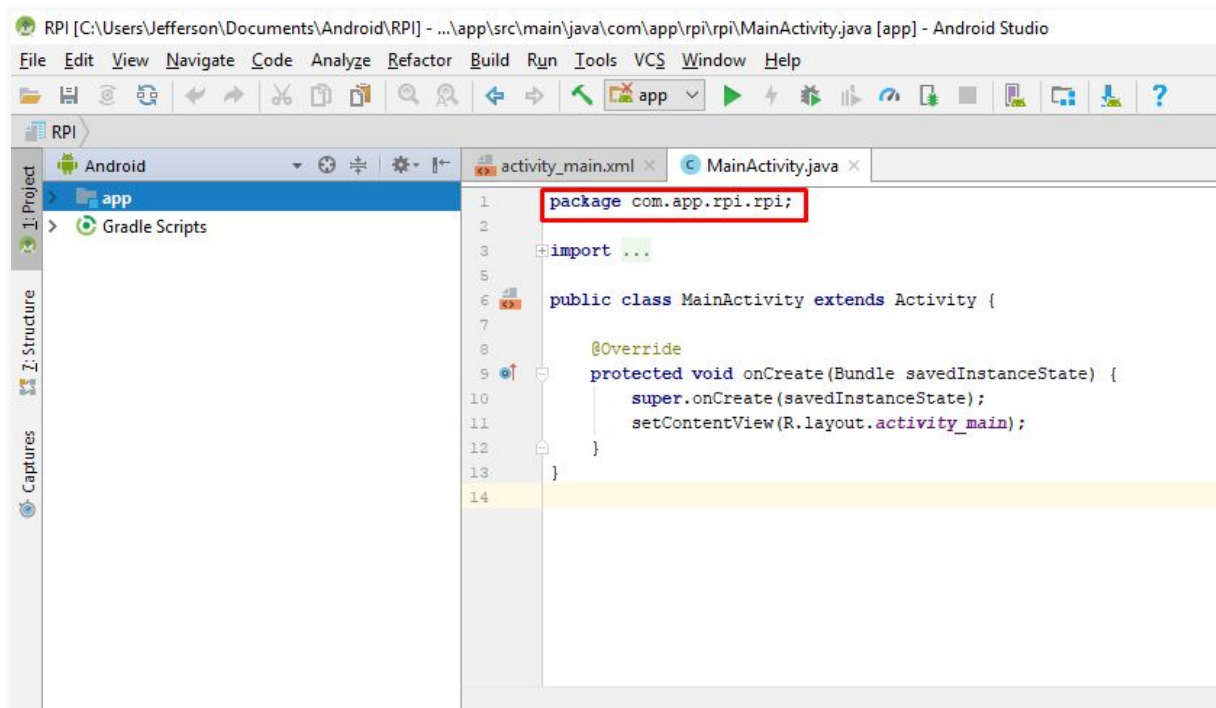
☐ **TV**  
API 21: Android 5.0 (Lollipop)

☐ **Android Auto**

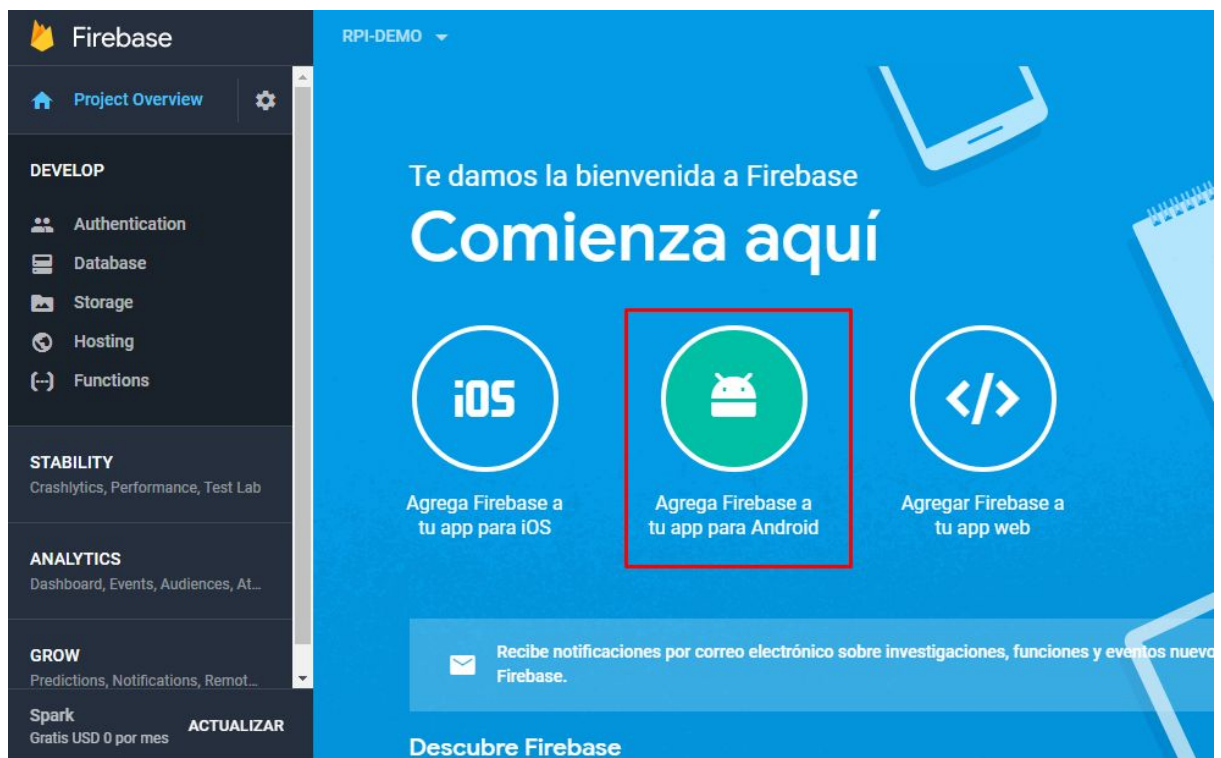
☐ **Android Things**  
API 24: Android 7.0 (Nougat)

Previous Next Cancel Finish





Ya que hemos creado el proyecto Android, tomamos el nombre del package, y volvemos al proyecto de firebase y seleccionamos “Agregar firebase a tu app Android”





Copiamos el nombre del paquete, como lo indica la imagen.

1

2

3

Registrar app

Descargar archivo de configuración

Agregar el SDK de Firebase

Nombre de paquete de Android ⓘ

com.app.rpi.rpi

Sobrenombre de la app (opcional) ⓘ

RPI

Certificado de firma SHA-1 de depuración (opcional) ⓘ

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Obligatoria para Dynamic Links, Invites y la asistencia con un número telefónico o el Acceso con Google en Auth. Puedes editar la clave SHA-1s en Configuración.

CANCELAR

REGISTRAR APP

en el proyecto RPI-DEMO

## Agrega Firebase a tu app para Android

1

2

3

Registrar app

Descargar archivo de configuración

Agregar el SDK de Firebase

### Instrucciones de Android Studio

Alternativas: [Unity](#), [C++](#)

1.


Descargar google-services.json

2.

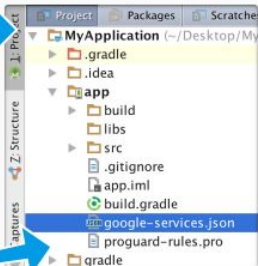
Cambia a la vista Proyecto de Android Studio para ver el directorio raíz de tu proyecto.

3.

Coloca el archivo google-services.json que acabas de descargar en el directorio raíz del módulo de tu app de Android.



google-services.json



El siguiente paso es descargar el archivo `google-service.json`, y arrastrarlo y soltarlo en la carpeta `app`, del proyecto en el Android Studio.

Ahora agregamos el SDK, al proyecto, como lo indica la imagen.

## Agrega Firebase a tu app para Android

1

2

3

Registrar app

Descargar archivo de configuración

Agregar el SDK de Firebase

### Instrucciones de Gradle

Alternativas: [Unity](#) [C++](#)

El complemento de los servicios de Google para [Gradle](#) carga el archivo `google-services.json` que acabas de descargar. Para poder usar el complemento, debes modificar los archivos `build.gradle`.

1. `build.gradle` de proyecto (`<project>/build.gradle`):

```
buildscript {
  dependencies {
    // Add this line
    classpath 'com.google.gms:google-services:3.2.0'
  }
}
```

## Agrega Firebase a tu app para Android

1

2

3

Registrar app

Descargar archivo de configuración

Agregar el SDK de Firebase

```
// Add this line
classpath 'com.google.gms:google-services:3.2.0'
}
```

2. `build.gradle` de aplicación (`<project>/<app-module>/build.gradle`):

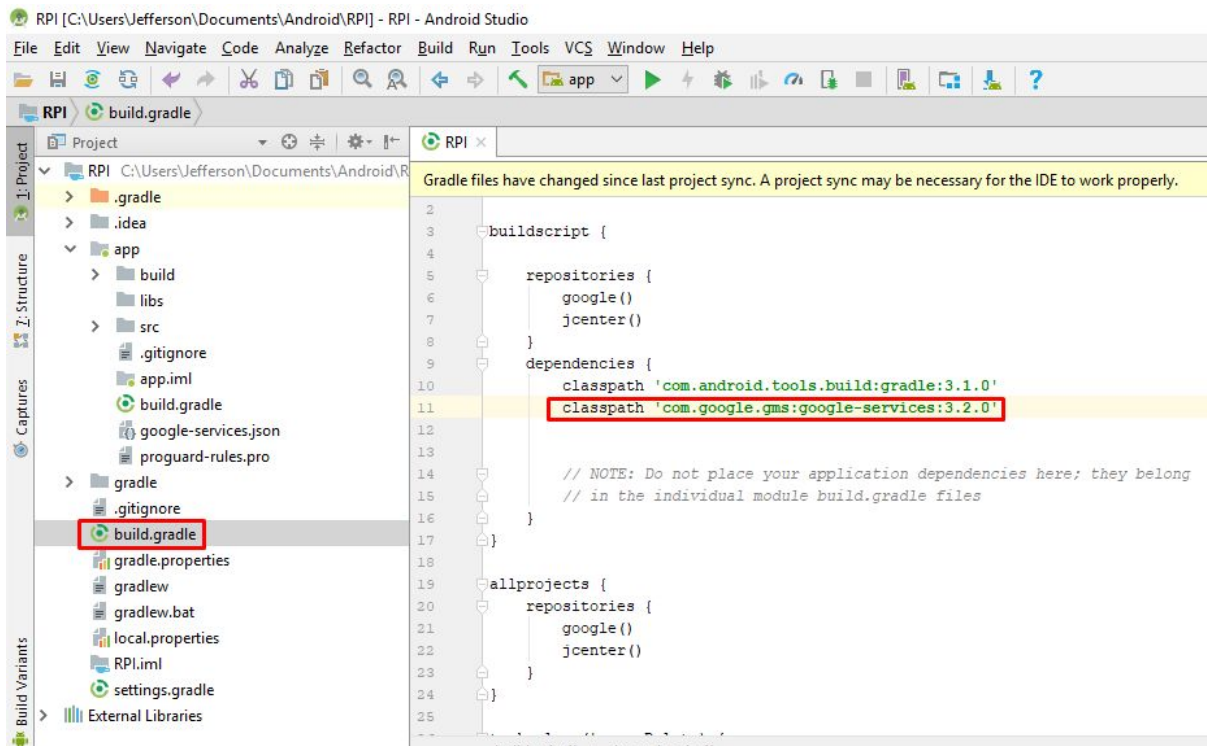
```
dependencies {
  // Add this line
  compile 'com.google.firebase:firebase-core:12.0.0'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

*incluye Analytics en la configuración predeterminada* ⓘ

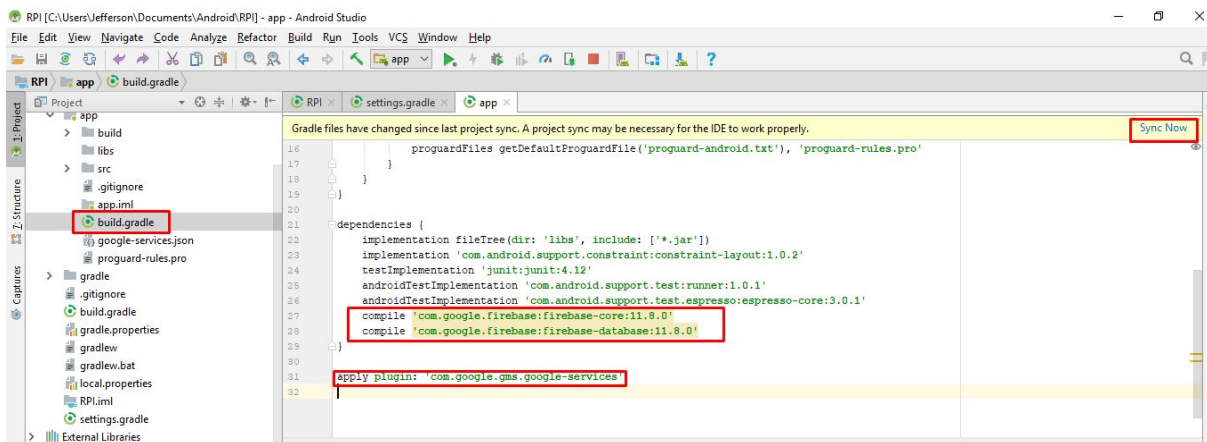
3. Por último, presiona **Sincronizar ahora** en la barra que aparece en el entorno IDE:

Gradle files have changed since last sync. [Sync now](#)

FINALIZAR



`classpath 'com.google.gms:google-services:3.2.0'`



`compile 'com.google.firebase:firebase-core:11.8.0'`

`compile 'com.google.firebase:firebase-database:11.8.0'`

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

Luego de esto, vamos a establecer la conexión con la base de datos.

<https://firebase.google.com/docs/database/android/start/>

## Write to your database

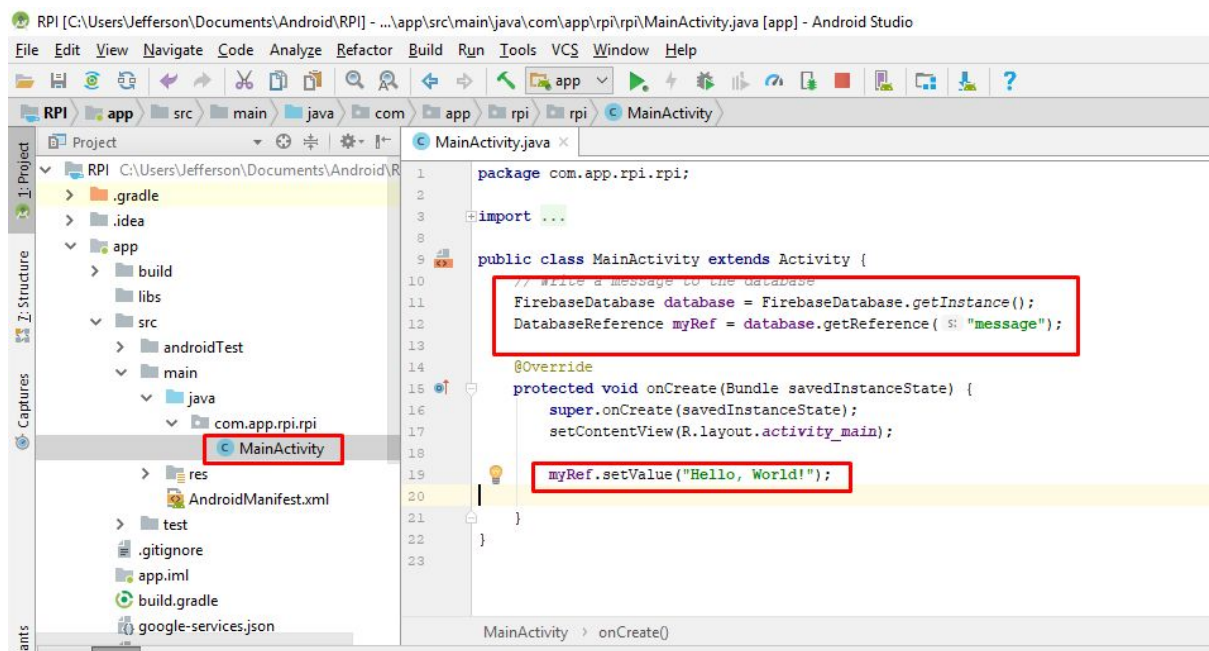
Retrieve an instance of your database using `getInstance()` and reference the location you want to write to.

```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");

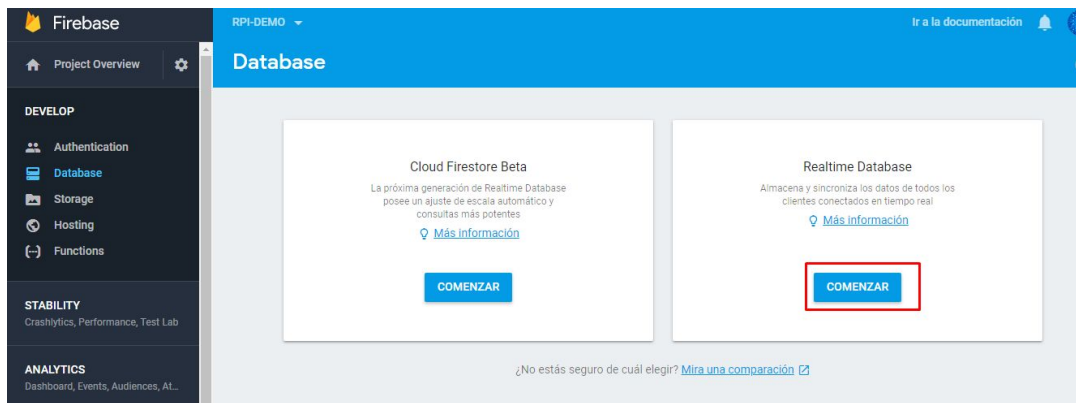
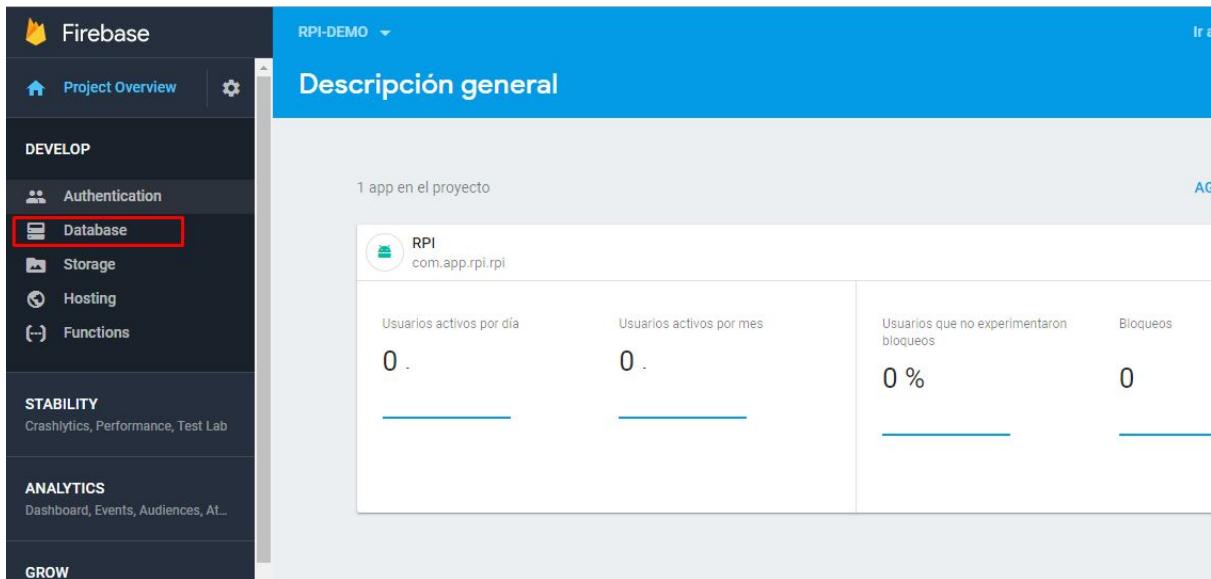
myRef.setValue("Hello, World!");
```

You can save a range of data types to the database this way, including Java objects. When you save an object the responses from any getters will be saved as children of this location.

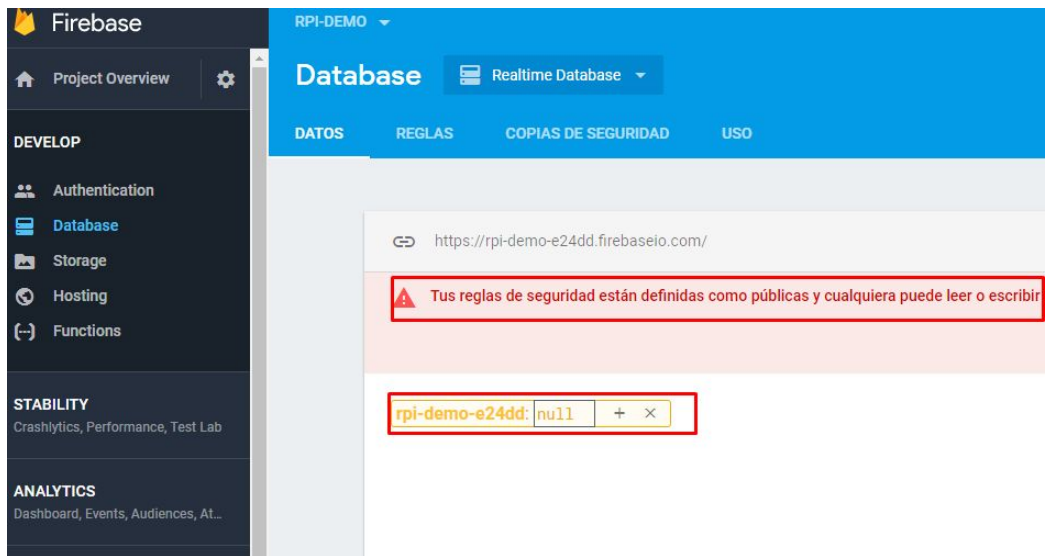
El siguiente código es para probar que estamos conectados correctamente a la base de datos de Firebase



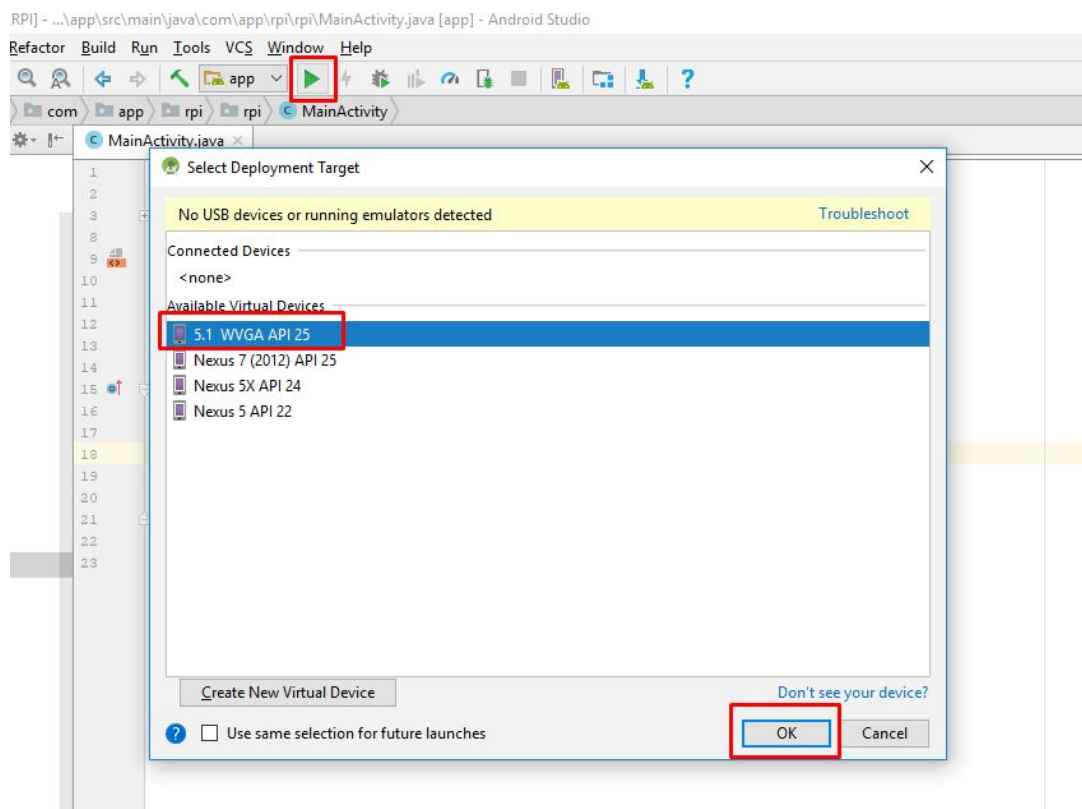
Antes de ejecutar este código, vamos a firebase y hacemos la configuración inicial de la base de datos.

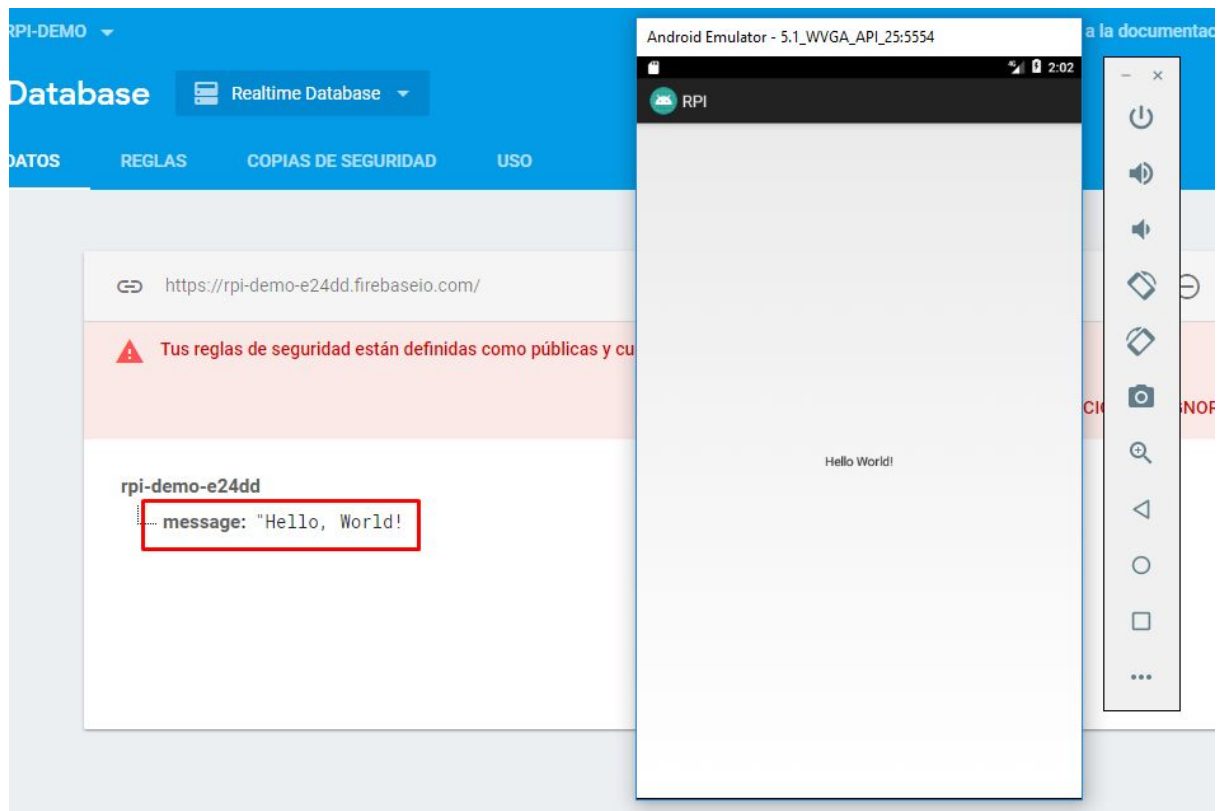




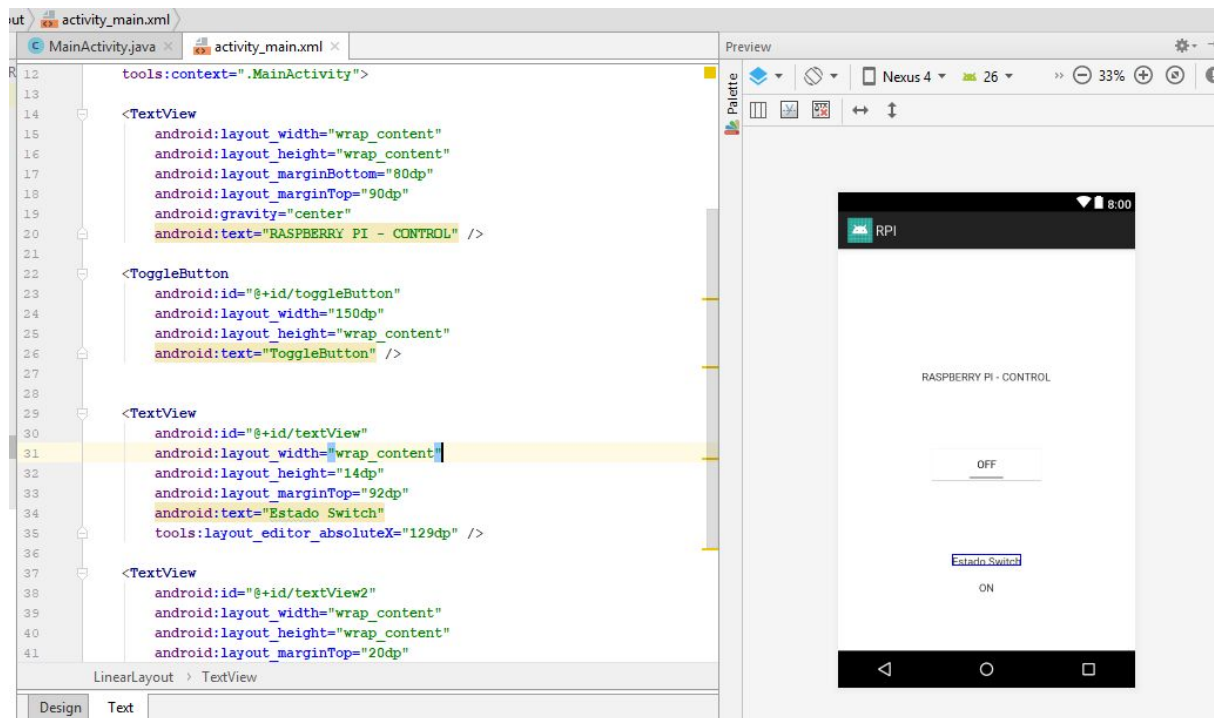


Ahora si volvemos al Android Studio, y Ejecutamos el proyecto.





Una vez se lanza la aplicación, vemos en Firebase que nos agregó message: "Hello World!"  
Qué fue lo que le seteamos en el código.



En el siguiente enlace puede ver el código completo de la aplicación.


[https://github.com/rpi-jefer/rpio\\_iot\\_android\\_python](https://github.com/rpi-jefer/rpio_iot_android_python)

### 3. DESCARGAR LAS CREDENCIALES DE FIREBASE PARA LA CONEXIÓN CON PYTHON

Primero, tenemos que ir a Firebase y descargar las credenciales, como se muestra a continuación.



Firestore

Project Overview 

DEVELOP

- Authentication
- Database
- Storage
- Hosting
- Functions

STABILITY

Crashlytics, Performance, Test Lab

ANALYTICS

Dashboard, Events, Audiences, AI...

GROW

Predictions, Notifications, Remote...

Spark

Gratis USD 0 por mes ACTUALIZAR

RPI-DEMO

Ir a la documentación

## Configuración

GENERAL CLOUD MESSAGING VINCULACIÓN DE LA CUENTA CUENTAS DE SERVICIO PRIVACIDAD DE LOS DATOS

[Administrar todas las cuentas de servicio](#)

**Firebase Admin SDK**

Informes de fallos

Credenciales heredadas

Database Secrets

Firebase Admin SDK

Tu cuenta de servicio de Firebase puede usarse para autenticar automáticamente varias funciones de Firebase, como Database, Storage y Auth, a través del SDK de Admin unificado. [Más información](#)

Cuenta de servicio de Firebase

firebase-adminsdk-ux17c@rpi-demo-e24dd.iam.gserviceaccount.com

Fragmento de configuración de Admin SDK

☐ Node.js ☐ Java ☒ Python ☐ Go

```
import firebase_admin
from firebase_admin import credentials

cred = credentials.Certificate("path/to/serviceAccountKey.json")
```

Cuenta de servicio de Firebase

firebase-adminsdk-ux17c@rpi-demo-e24dd.iam.gserviceaccount.com

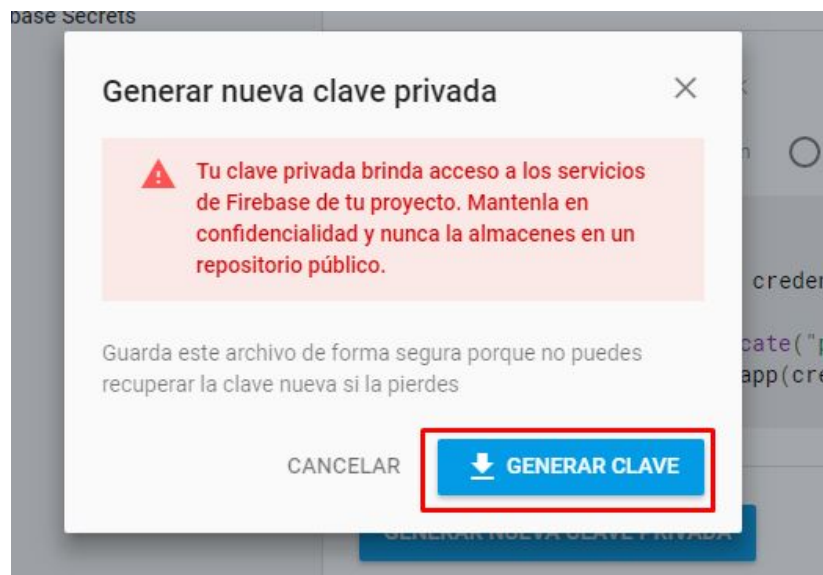
Fragmento de configuración de Admin SDK

☐ Node.js ☐ Java ☒ Python ☐ Go

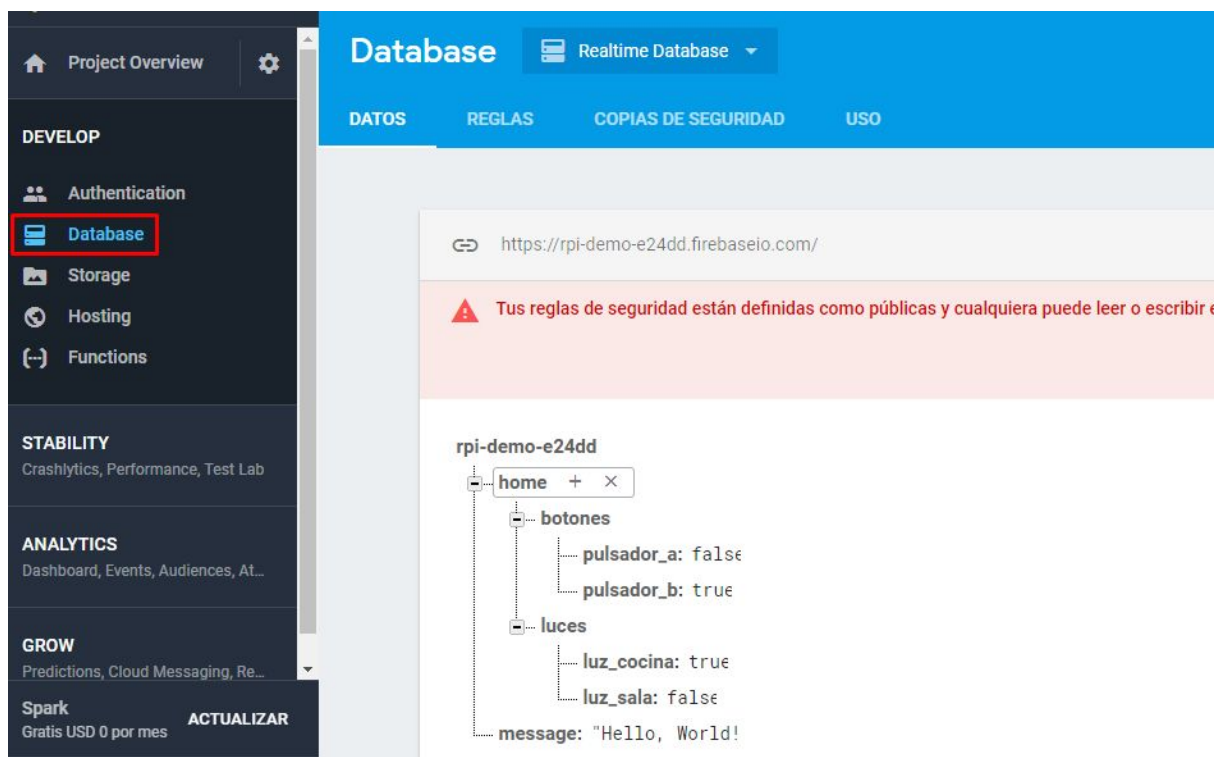
```
import firebase_admin
from firebase_admin import credentials

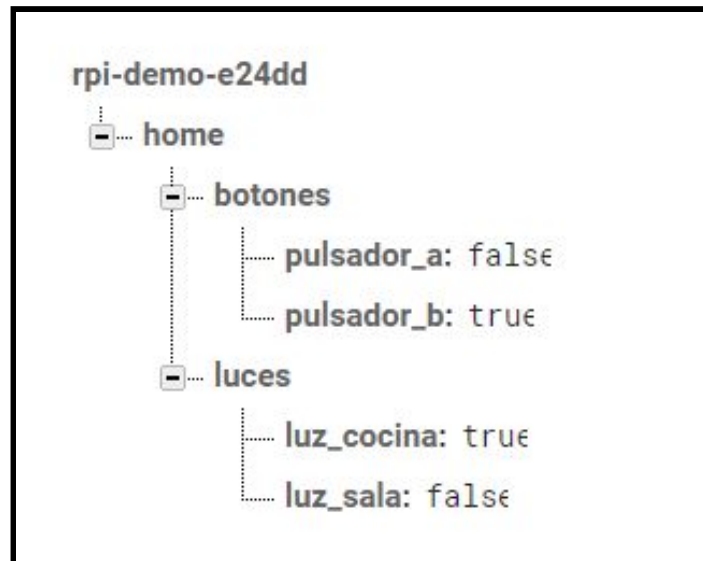
cred = credentials.Certificate("path/to/serviceAccountKey.json")
firebase_admin.initialize_app(cred)
```

**GENERAR NUEVA CLAVE PRIVADA**



Ahora vamos a la base de datos en Firebase y creamos el siguiente árbol





```
{
  "home" : {
    "botones" : {
      "pulsador_a" : false,
      "pulsador_b" : true
    },
    "luces" : {
      "luz_cocina" : true,
      "luz_sala" : false
    }
  }
}
```

## 4. INSTALAR LO NECESARIO EN LA RASPBERRY PI

### ACTUALIZAR LOS REPOSITORIOS

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

<http://www.linuxhispano.net/2013/05/03/diferencia-entre-apt-get-update-y-apt-get-upgrade/>

### INSTALAR PIP

```
$ sudo wget https://bootstrap.pypa.io/get-pip.py
```

```
$ sudo python get-pip.py
```

### INSTALAR PYTHON DEV

```
$ sudo apt-get install python-dev
```

## INSTALAR FIREBASE ADMIN

```
$ sudo pip install firebase-admin
```

## INSTALAR GPIO ZERO

```
$ sudo apt install python-gpiozero
```

## CÓDIGO PRELIMINAR PARA TESTEAR FIREBASE

```
$ sudo nano test_fire.py
```

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

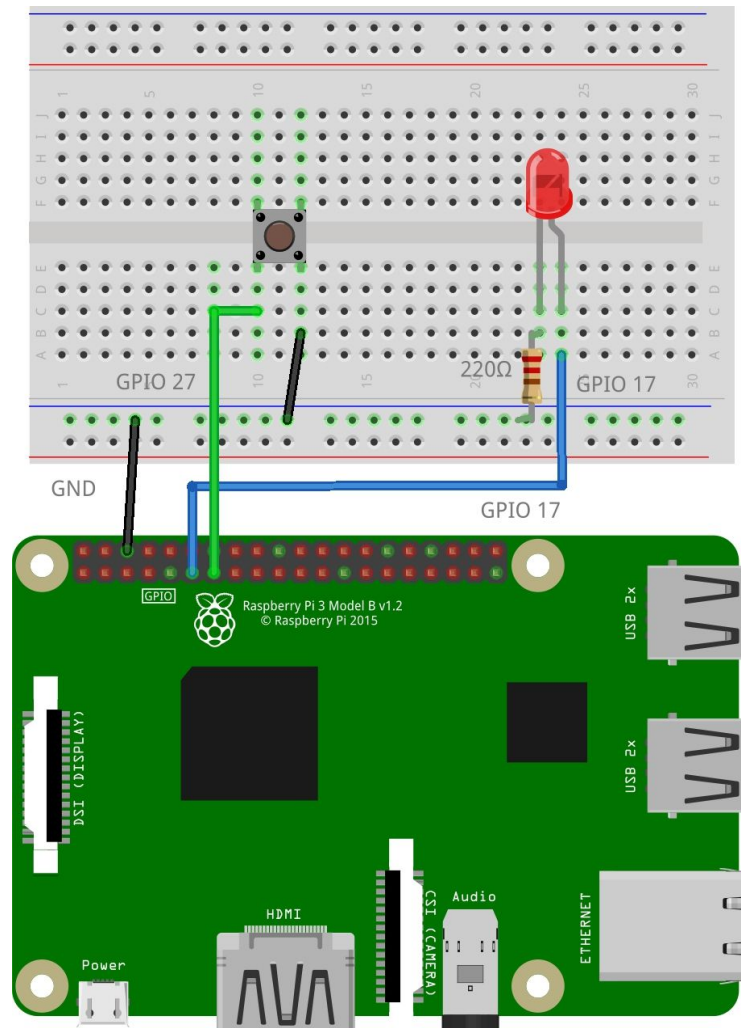
cred = credentials.Certificate('/home/pi/cred.json')

# Initialize the app with a service account, granting admin privileges
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://tu-proyecto.firebaseio.com/'
})

ref = db.reference('demo')
print(ref.get())
print ('Ok !')
```

```
$ sudo python test_fire.py
```

## 5. PLANO ESQUEMÁTICO



## 5. CREAR CÓDIGO PYTHON

```
$ mkdir iot
```

```
$ cd iot
```

```
$ sudo nano main.py
```

Ver el código en el siguiente enlace

[https://github.com/rpi-jefer/rpio\\_iot\\_android\\_python/blob/master/python/main.py](https://github.com/rpi-jefer/rpio_iot_android_python/blob/master/python/main.py)

```

#!/usr/bin/python
# -*- coding: utf-8 -*-
# autor: Jefferson Rivera
# Abril de 2018
# email: riverajefer@gmail.com

import sys
from time import sleep

import signal
from gpiozero import LED, Button
from threading import Thread
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

LED = LED(17)
BUTTON = Button(27)

PAHT_CRED = '/home/pi/iot/cred.json'
URL_DB = 'https://rpi-demo-e24dd.firebaseio.com/'
REF_HOME = 'home'
REF_LUCES = 'luces'
REF_BOTONES = 'botones'
REF_LUZ_SALA = 'luz_sala'
REF_PULSADOR_A = 'pulsador_a'

class IOT():

    def __init__(self):
        cred = credentials.Certificate(PAHT_CRED)
        firebase_admin.initialize_app(cred, {
            'databaseURL': URL_DB
        })

        self.refHome = db.reference(REF_HOME)

        #self.estructuraInicialDB() # solo ejecutar la primera vez

        self.refLuces = self.refHome.child(REF_LUCES)
        self.refLuzSala = self.refLuces.child(REF_LUZ_SALA)

        self.refBotones = self.refHome.child(REF_BOTONES)
        self.refPulsadorA = self.refBotones.child(REF_PULSADOR_A)

    def estructuraInicialDB(self):
        self.refHome.set({
            'luces': {
                'luz_sala': True,
                'luz_cocina': True
            },
            'botones': {
                'pulsador_a': True,
                'pulsador_b': True
            }
        })

```

```

    }
}))

def ledControlGPIO(self, estado):
    if estado:
        LED.on()
        print('LED ON')
    else:
        LED.off()
        print('LED OFF')

def lucesStart(self):

    E, i = [], 0

    estado_anterior = self.refLuzSala.get()
    self.ledControlGPIO(estado_anterior)

    E.append(estado_anterior)

    while True:
        estado_actual = self.refLuzSala.get()
        E.append(estado_actual)

        if E[i] != E[-1]:
            self.ledControlGPIO(estado_actual)

        del E[0]
        i = i + 1
        sleep(0.4)

def pulsador_on(self):
    print('Pulsador On')
    self.refPulsadorA.set(True)

def pulsador_off(self):
    print('Pulsador Off')
    self.refPulsadorA.set(False)

def botonesStart(self):
    print('Start btn !')
    BUTTON.when_pressed = self.pulsador_on
    BUTTON.when_released = self.pulsador_off

print ('START !')
iot = IOT()

subproceso_led = Thread(target=iot.lucesStart)
subproceso_led.daemon = True
subproceso_led.start()

subproceso_btn = Thread(target=iot.botonesStart)
subproceso_btn.daemon = True

```

```
subproceso_btn.start()  
signal.pause()
```

```
$ sudo python main.py
```

Código en GitHub

[https://github.com/rpi-jefer/rpio\\_iot\\_android\\_python/blob/master/python/main.py](https://github.com/rpi-jefer/rpio_iot_android_python/blob/master/python/main.py)

## 5. EJECUTAR EL SCRIPT AL ARRANCAR EL SISTEMA OPERATIVO.

Instalar Supervisor

```
$ sudo apt-get install supervisor
```

Crear el archivo de configuración.

```
$ sudo nano /etc/supervisor/conf.d/iot.conf
```

Agregamos:

```
[program:iot]  
command=sudo python /home/pi/iot/main.py  
autostart=true  
autorestart=true  
stderr_logfile=/var/log/iot.err.log  
stdout_logfile=/var/log/iot.out.log
```

Damos permiso de ejecución al archivo main.py

```
$ sudo chmod 755 /home/pi/iot/main.py
```

Ahora reiniciamos

```
$ sudo reboot
```



Y con esto ya tenemos la aplicación ejecutándose al iniciar la Raspberry Pi.

@autor: Jefferson Rivera  
[riverajefer@gmail.com](mailto:riverajefer@gmail.com)  
Bogotá Abril de 2018