

# FIREBASE-PYTHON TUTORIAL

## 1.-Introducción

---

### **¿Qué es Firebase?**

R:

Es una plataforma digital que se utiliza para facilitar el desarrollo de aplicaciones web o móviles de una forma efectiva, rápida y sencilla, la cual es utilizada por sus diversas funciones como una técnica de Marketing Digital para aumentar la base de usuarios y generar mayores beneficios económicos.

Su principal objetivo, es mejorar el rendimiento de las apps mediante la implementación de diversas funcionalidades que van a hacer de la aplicación en cuestión, mucho más manejable, segura y de fácil acceso para los usuarios.

### **¿Qué funcionalidades provee?**

R:

Son diferentes y con distintas utilidades:

- Real time data base: nos ayudan, dándonos una base de datos en tiempo real.
- Autenticación: se ejecutan para identificar a los usuarios mediante el email o las redes sociales.
- Nube de almacenamiento: almacenamiento y envío de archivos a la escala de Google.
- Hosting: se usa para publicar nuestra página web.
- Remote config: se emplea para modificar ciertos aspectos de nuestra app sin la necesidad de actualizar la misma.
- Test lab: es para probar la aplicación antes de publicarla.
- Crash reporting: se utiliza para reportar errores de la aplicación.

## 2.- Tutorial

---

### 2.1.- Para comenzar:

Este tutorial no abarca todas las funcionalidades de Firebase, se enfocará principalmente en su módulo de base de datos en tiempo real.

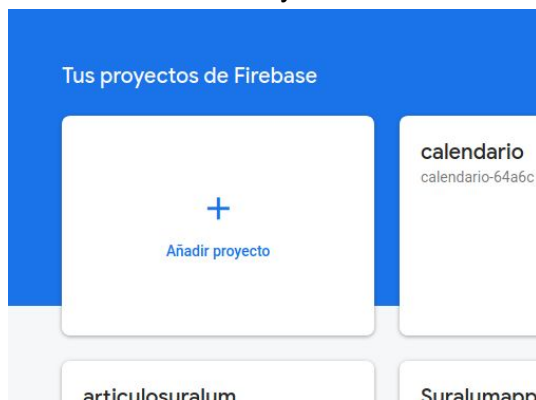
Librerías:

- pip3 install pyrebase

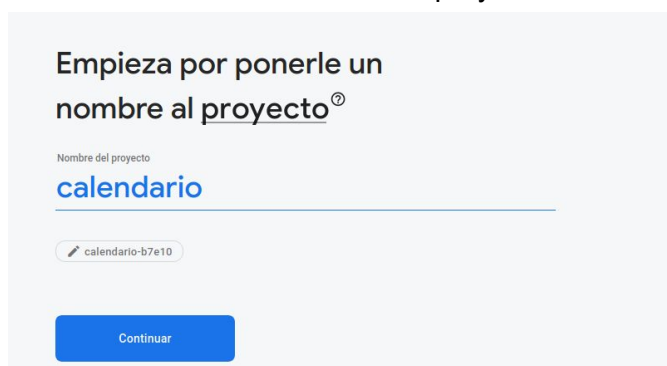
### Crear base de datos en Firebase:

<https://console.firebase.google.com/u/0/>

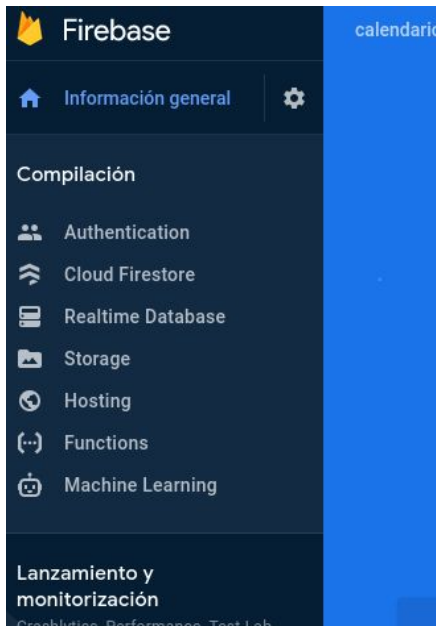
#### 1.Click en Añadir Proyecto



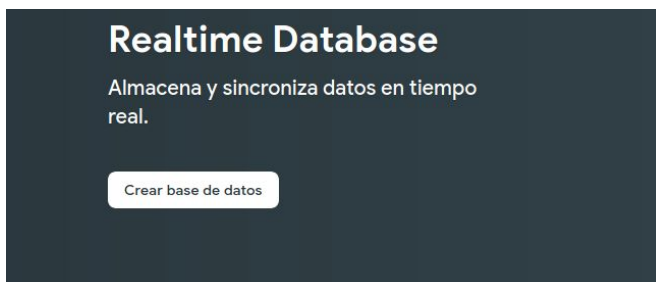
#### 2.-Ponerle un nombre a nuestro proyecto



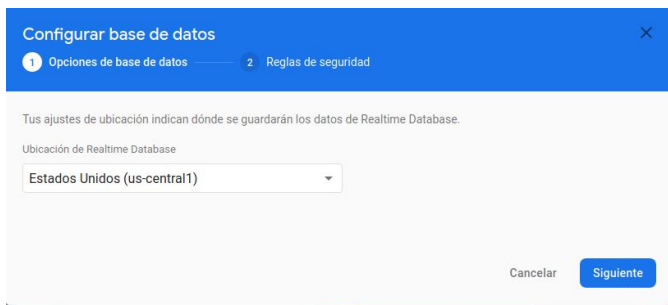
#### 3.- Ir a "RealTime Database"



#### 4.- Crear una nueva base de datos



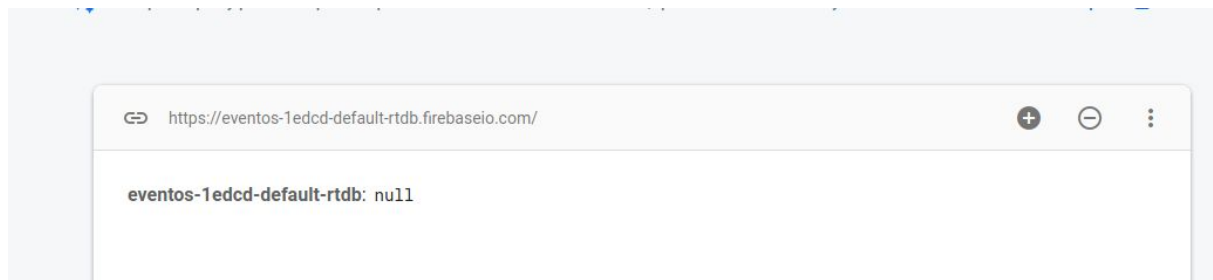
#### 5.- Seleccionamos un servidor



#### 6.- Seleccionamos el modo de prueba



Y Listo, hemos terminado la configuración básica de Firebase-RealTime DataBase al terminar lo pasos anterior veremos esto



## Código de prueba

De la sección anterior podemos obtener los datos para databaseURL , authDomain es el nombre de nuestra base de datos más `".firebaseapp.com"`

```
test_firebase.py > ...
1  import pyrebase
2
3  config = {
4      "apiKey": "apiKey",
5      "authDomain": "eventos.firebaseio.com",
6      "databaseURL": "https://eventos-1edcd-default-rtdb.firebaseio.com/",
7      "storageBucket": "local"
8  }
9
10 firebase = pyrebase.initialize_app(config)
11
12 db = firebase.database()
13 data = {"Evento": "Seminario", "D": 3, "M": 2, "Y": 2020}
14 db.child("users").push(data)
15
16
17
18
```

Al ejecutar este código:

```
onathan@Jaeger:~/Escritorio/Test$ python3 test_firebase.py
```

Podemos ver que nuestro evento se encuentra agregado a nuestra base de datos:



## Aplicando Firebase en el Desarrollo de nuestro bot de slack “Sam”

### Código principal:

#### 1.- Agregar eventos

Declaramos la configuración y conexión a la base de datos:

```
#creamos conexión a firebase
config = {
    "apiKey": "apiKey",
    "authDomain": "eventos.firebaseio.com",
    "databaseURL": "https://eventos-1edcd-default-rtdb.firebaseio.com/",
    "storageBucket": "local"
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()
##### CONEXIÓN A RABBIT MQ #####
```

Si se el mensaje comienza con “[evento]” este envía el mensaje al “buzón” rabbitmq de “eventos”;

```
if text.startswith("[evento]"):
    channel.basic_publish(exchange='sam', routing_key="eventos", body=text)
```

#### 2.-Ver eventos:

Para ver los eventos del mes se agregó a slack el comando “/eventosmes” por lo que se debe crear un endpoint para las consultas entrantes al puerto. Este comando invoca la función *eventosmes()* la cual manda una consulta a la base de datos ubicada en *Firebase*.

```
@app.route('/eventosmes', methods=['POST'])
def eventosmes():
    mes = str(date.today().month)
    eventos= db.child(mes).get()
    if isinstance(eventos.each(), list):
        for user in eventos.each():
            a=str(user.val())
            b= a.split()
            em = "Titulo: "+b[3][1:-2]+"", Día: "+ b[1][:-1]
            channel.basic_publish(exchange='sam', routing_key="publicar_slack", body=em)
    else:
        print("No existen eventos ingresados este mes")
    return Response(),200
```

## Módulo eventos:

Declaramos la configuración y conexión a la base de datos:

```
#creamos conexión a firebase
config = {
    "apiKey": "apiKey",
    "authDomain": "eventos.firebaseio.com",
    "databaseURL": "https://eventos-1edcd-default-rtdb.firebaseio.com/",
    "storageBucket": "local"
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()
##### CONEXIÓN A RABBIT MQ #####
```

El proceso se comunica con la base de datos y hace un “push” de la data.

```
#se inicia
def callback(ch, method, properties, body):
    print(body)
    if str(body).startswith("b'[evento]"):
        q = str(body)[10:]
        eve = q.split()
        T = eve[0][2:]
        M = int(eve[1][2:])
        D = int(eve[2][2:-1])
        data = {'Día':D, 'Título':T}
        db.child(M).push(data)
        body= "Evento ingresado"
        ##### PUBLICA EL RESULTADO COMO EVENTO EN RABBITMQ #####
        channel.basic_publish(exchange='sam', routing_key="publicar_slack", body=body)
```

