



Introducción al Internet de las Cosas (IoT)





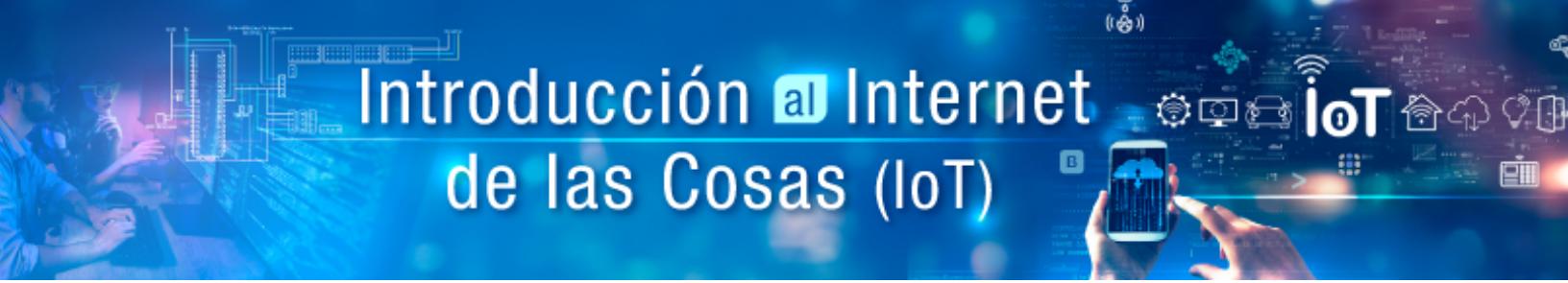
Guía Trabajo Final

Conexión de sistemas de nube realtime database firebase

En esta guía, se creará una aplicación web en Firebase para mostrar las lecturas de los sensores guardadas en Firebase Realtime Database. La página web de lecturas del sensor está protegida con autenticación por medio de correo electrónico y contraseña.

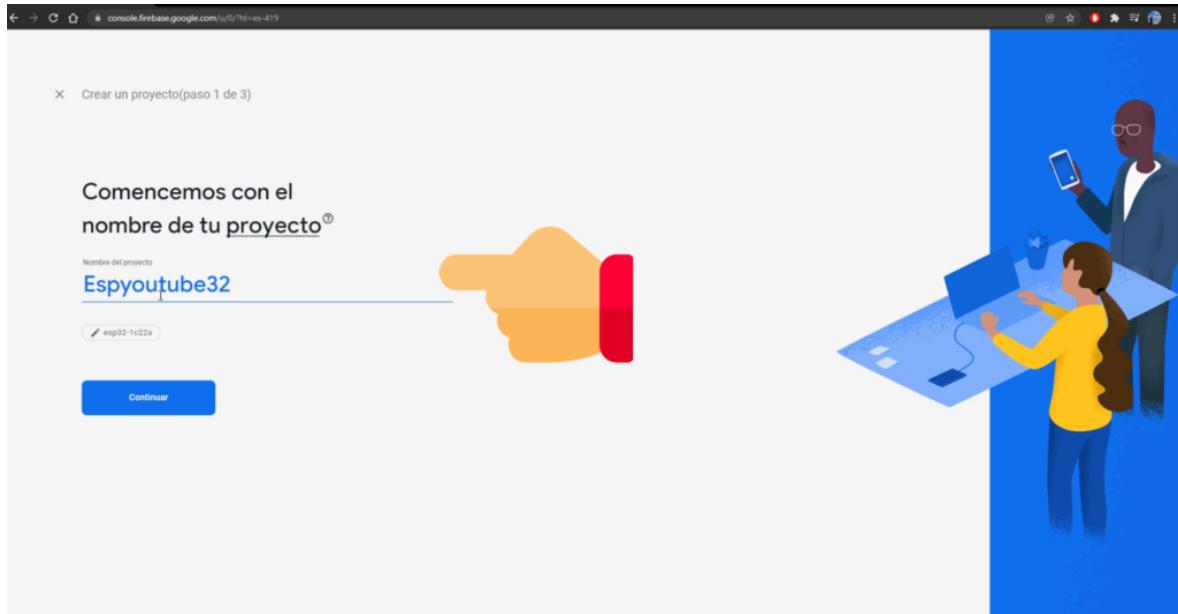
El siguiente diagrama muestra el proyecto a grandes rasgos que vamos a construir



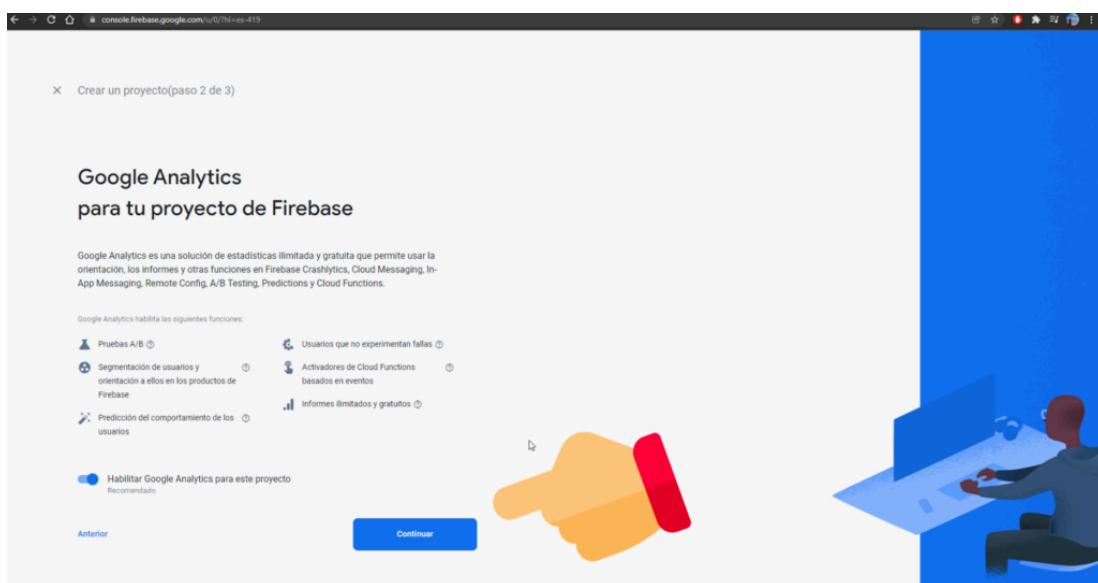


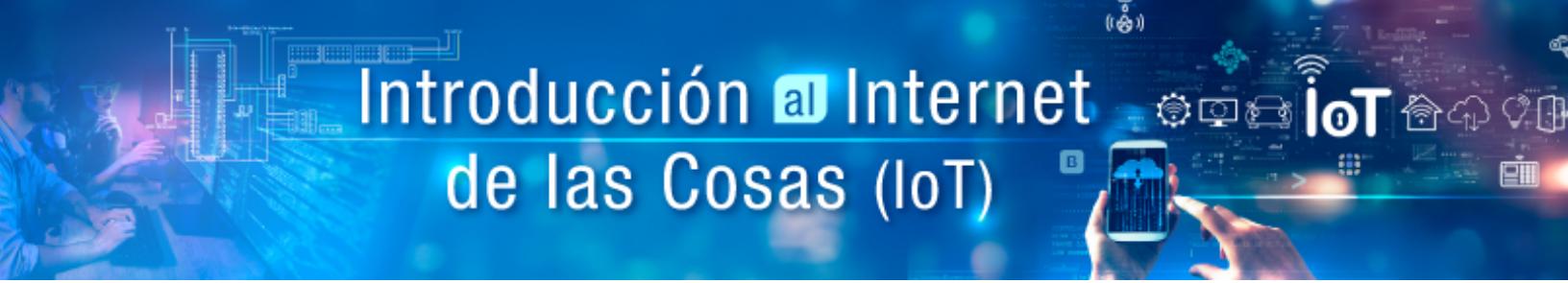
Instrucciones:

Lo primero que debemos hacer es ingresar a Firebase Console y crear un nuevo proyecto.



Le marcamos continuar en el proceso





Generamos un nombre para la configuración de Google Analytics, en este caso puse Esp32Youtube

× Crear un proyecto(paso 3 de 3)

Configurar Google Analytics

Elige o crea una cuenta de Google Analytics ⓘ

Esp32Youtube

Ubicación de Analytics ⓘ

Estados Unidos

Configuración de uso compartido de datos y Condiciones de Google Analytics

Usar la configuración predeterminada para compartir datos de Google Analytics [Más información](#)

- ✗ Compartir tus datos de Analytics con Google para mejorar nuestros productos y servicios
- ✓ Compartir tus datos de Analytics con Google para habilitar las comparativas
- ✓ Compartir tus datos de estadísticas con Google para habilitar la asistencia técnica
- ✓ Compartir tus datos de Analytics con los especialistas en Cuentas de Google

 Acepto las [Condiciones de Google Analytics](#).

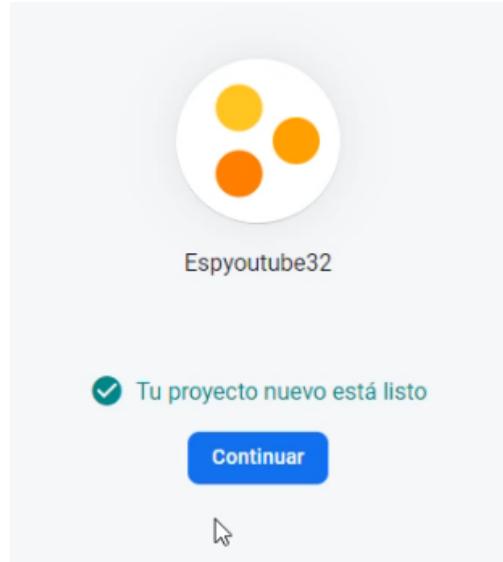
Cuando se cree el proyecto, también se creará una nueva propiedad de Google Analytics. Este vínculo permitirá el flujo de datos entre los productos. Los datos que se exportan de tu Firebase están sujetos a las Condiciones del Servicio de Firebase, mientras que los datos de Google Analytics están sujetos a las Condiciones del Servicio de Google Analytics. [Obtén más información](#).

[Anterior](#) [Crear proyecto](#)

Luego firebase construye el proyecto con los parámetros establecidos



Introducción al Internet de las Cosas (IoT)



Ingresamos en la opción web

The screenshot shows the Firebase console interface. On the left, a sidebar lists services: Compilación (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Lanzamiento y supervisión (Crashlytics, Performance, Test Lab, App Distribution), and Analytics (Dashboard, Realtime, Events, Conversions, Audience, Extensions). The main area displays the project 'Espyoutube32'. It includes a 'Comienza por agregar Firebase a tu app' section with icons for iOS+, Android, Web, and a hand pointing to the 'Web' icon. Below this, there's a section about real-time data synchronization and two detailed cards: 'Authentication' (Autentica y administra usuarios) and 'Cloud Firestore' (Actualizaciones en tiempo real, consultas poderosas y ajuste de escala automático).

Introducción al Internet de las Cosas (IoT)

Le indicamos en la opción de configuración **Firebase Hosting** y le ponemos un nombre, en este caso **Sensor App**

X Agrega Firebase a tu app web

1 Registrar app

Sobrenombre de la app [?](#)

Sensor App

Además, configura **Firebase Hosting** para esta app. [Más información](#)

También puedes configurarlo más adelante. Puedes comenzar a usarlo sin costo en cualquier momento.

esp32-1c22a (Aún no hay implementaciones) ▾

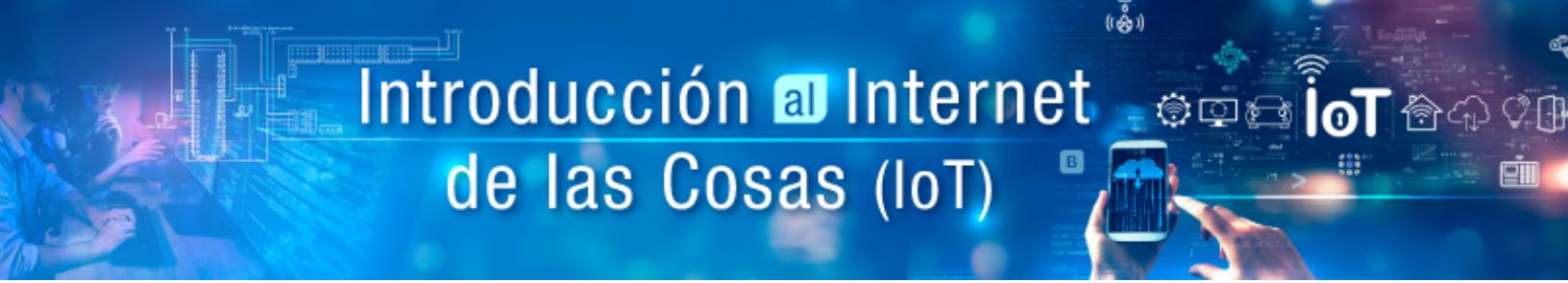
Registrar app

2 Agrega el SDK de Firebase

3 Instala Firebase CLI

4 Realiza la implementación en Firebase Hosting

💡 Nos muestra el siguiente fragmento de código, recomiendo **guardarlo en un blog de notas**, por que lo necesitaremos mas adelante.



Introducción al Internet de las Cosas (IoT)

Si ya usas [npm](#) y un agrupador de módulos como [Webpack](#) o [Rollup](#), puedes ejecutar el siguiente comando para instalar la versión más reciente del SDK:

```
$ npm install firebase
```



Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieras usar.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyCUKw9FsZeuzEXuoLUuwhe4cNoTNAQ-gY",
  authDomain: "esp32-1c22a.firebaseio.com",
  projectId: "esp32-1c22a",
  storageBucket: "esp32-1c22a.appspot.com",
  messagingSenderId: "593098947458",
  appId: "1:593098947458:web:8a5abddb8f41f4bfe1ed9b",
  measurementId: "G-87V1XN68NJ"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```



Nota: Esta opción utiliza el [SDK de JavaScript modular](#), que proporciona un tamaño reducido del SDK.

Obtén más información sobre Firebase para la Web: [primeros pasos](#), [referencia de la API del SDK web](#) y [muestras](#)

Siguiente



Le damos siguientes, siguiente y siguiente hasta finalizar.



× Agrega Firebase a tu app web

- 1 Registrar app
- 2 Agrega el SDK de Firebase
- 3 Instala Firebase CLI
- 4 Realiza la implementación en Firebase Hosting

Puedes realizar la implementación ahora o [más adelante](#). Para hacerlo ahora mismo, abre una ventana de la terminal y, luego, navega al directorio raíz de tu app web o crea uno.

[Acceder a Google](#)

```
$ firebase login
```



Inicia el proyecto

Ejecuta el siguiente comando en el directorio raíz de tu app:

```
$ firebase init
```



Cuando estés listo, implementa tu app web

Ubica los archivos estáticos (p. ej., HTML, CSS y JS) en el directorio de implementación de tu app (el directorio predeterminado es "public"). Luego, ejecuta este comando desde el directorio raíz de la app:

```
$ firebase deploy
```



Después de la implementación, consulta tu app en [esp32-1c22a.web.app](#).

¿Necesitas ayuda? Consulta [los documentos de Hosting](#).

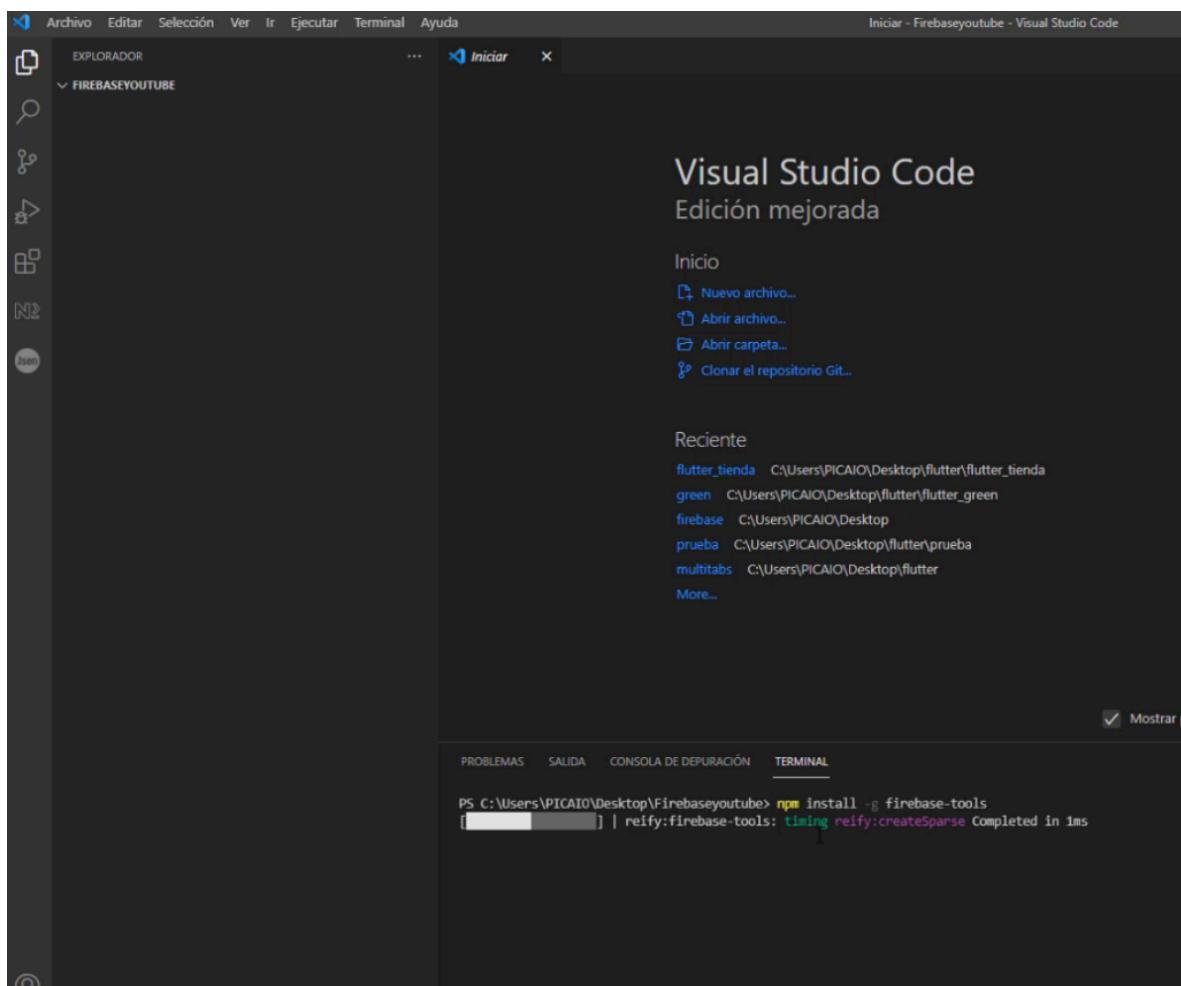
[Ir a la consola](#)

Introducción al Internet de las Cosas (IoT)

debemos instalar el Firebase CLI, para continuar con el proceso, el siguiente es el comando:

```
npm install -g firebase-tools
```

En mi caso, cree una carpeta en el escritorio que se llama FirebaseYoutube y en la terminar ingreso en la carpeta, luego doy al comando para instalar el Firebase CLI, como lo muestra la imagen.



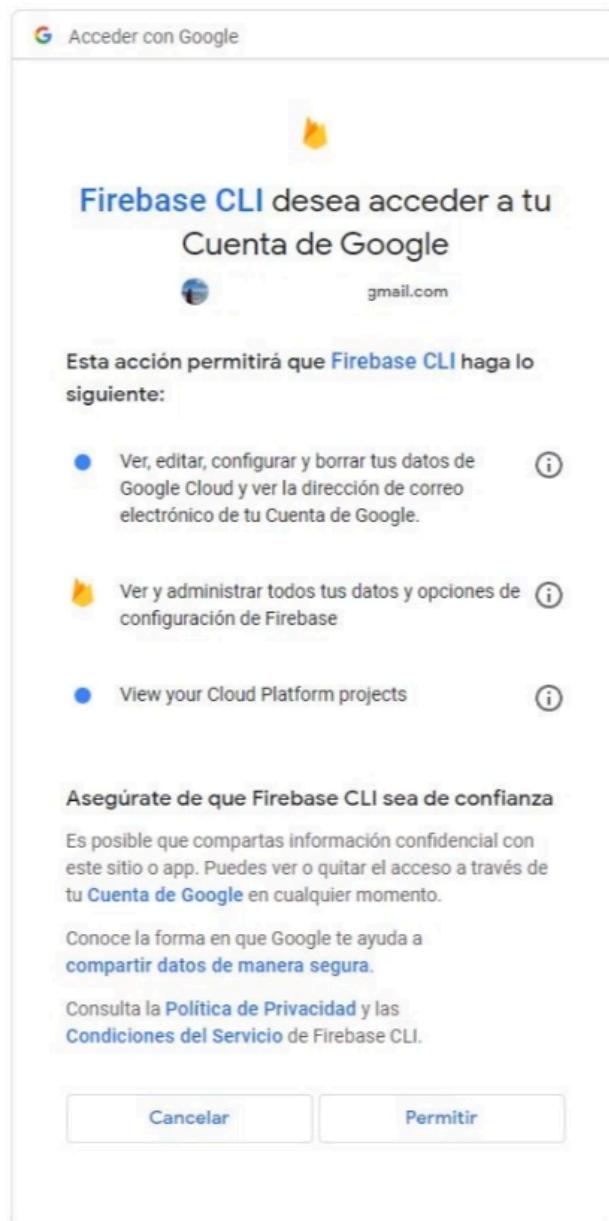
Ahora se ingresa el comando de firebase login

```
PS C:\Users\PICAIO\Desktop\Firebaseyoutube> firebase login
```



firebase login

Y aparece una ventana del navegador, solicitando ingreso a la cuenta de gmail que se utilizó para el proceso de firebase



Posteriormente saldrá el mensaje de que el login a sido exitoso

Introducción al Internet de las Cosas (IoT)

Woohoo!

Firebase CLI Login Successful

You are logged in to the Firebase Command-Line Interface. You can immediately close this window and continue using the CLI.

`firebase init`

Cuando ingresamos el comando de **firebase init**, nos muestra la pregunta, a la cual marcamos con una “y” y luego **ENTER**

Luego aparece la siguiente configuración, con la **Barra Espaciadora**, seleccionamos las siguientes opciones y luego **ENTER**

Introducción al Internet de las Cosas (IoT)

```
C:\Users\PICAO\Desktop\Firebaseyoutube
```

```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features,
d <enter> to proceed)
(*) Realtime Database: Configure a security rules file for Realtime Database and (optionally) prov
( ) Firestore: Configure security rules and indexes files for Firestore
( ) Functions: Configure a Cloud Functions directory and its files
>(*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
( ) Hosting: Set up GitHub Action deploys
( ) Storage: Configure a security rules file for Cloud Storage
( ) Emulators: Set up local emulators for Firebase products
(Move up and down to reveal more choices)
```

Luego seleccionamos la opcion de Use an existing project.

```
? Please select an option: (Use arrow keys)
> Use an existing project
Create a new project
Add Firebase to an existing Google Cloud Platform project
Don't set up a default project
```

Aparecerá el proyecto que creamos anteriormente

```
? Please select an option: Use an existing project
? Select a default Firebase project for this directory: (Use arrow keys)
> esp32-1c22a (Espyoutu32)
```

Luego pregunta si la inicializamos, le marcamos con la "y" y posteriormente seleccionamos el us-central1

```
? It seems like you haven't initialized Realtime Database in your project yet. Do you want to set it up? Yes
? Please choose the location for your default Realtime Database instance: (Use arrow keys)
> us-central1
europe-west1
asia-southeast1
```

A esta opción simplemente presionamos **ENTER**

```
? What file should be used for Realtime Database Security Rules? database.rules.json
```

A la siguiente opciones le marcamos con "n" para decirle que no

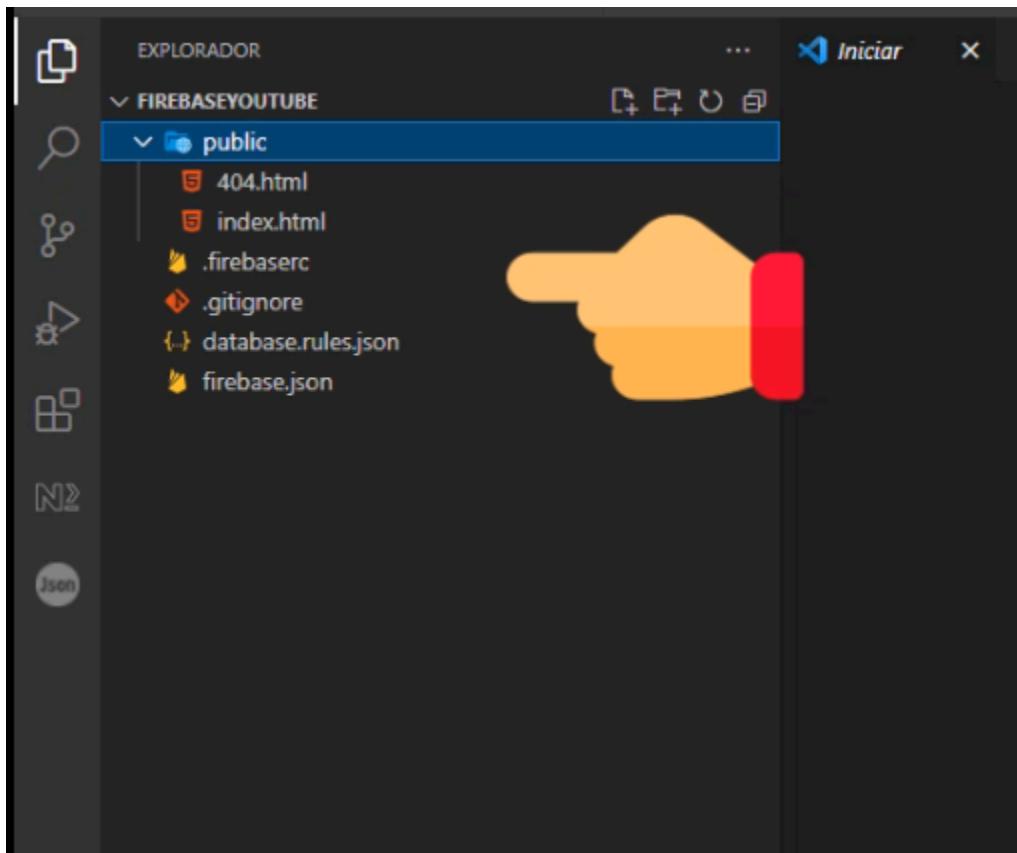
```
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N) n
```

Las siguientes son las opciones que se dejaron marcadas en los anteriores pasos.

Introducción al Internet de las Cosas (IoT)

```
? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? No
? Set up automatic builds and deploys with GitHub? No
```

Cuando terminamos el proceso, el crea por defecto los siguientes archivos.



Ahora damos el comando firebase deploy, para que lo despliegue a produccion en el hosting de firebase

```
PS C:\Users\PICAIO\Desktop\Firebaseyoutube> firebase deploy
```

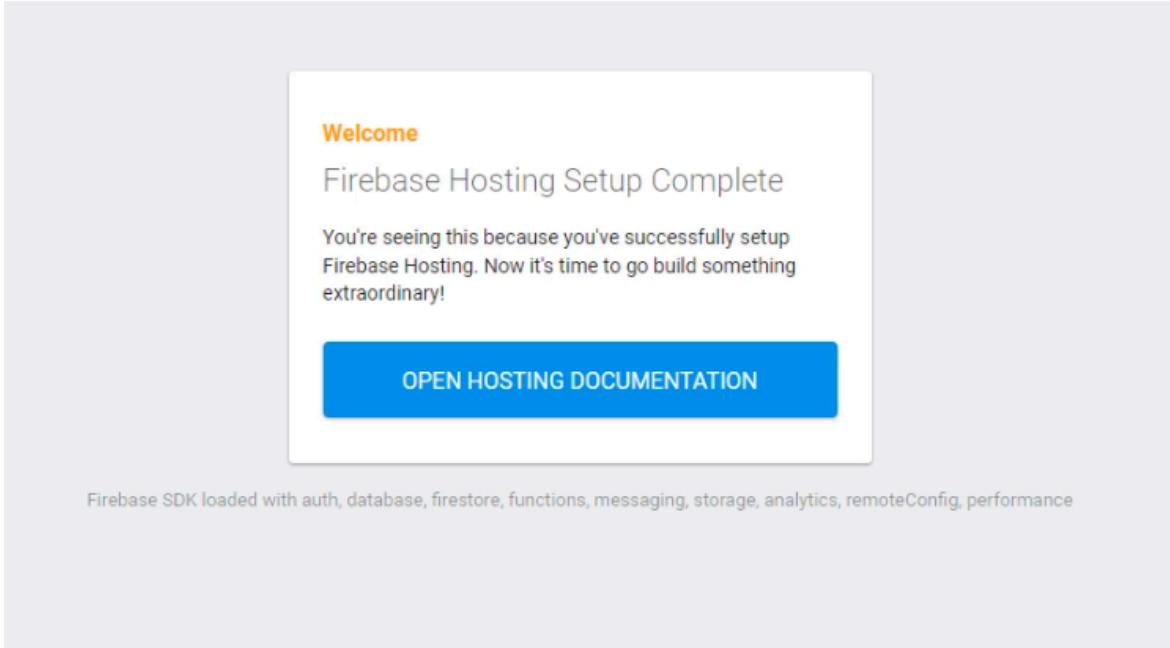
```
firebase deploy
```

Luego del deploy el genera un link en la consola donde podemos ingresar

```
Project Console: https://console.firebaseio.google.com/project/esp32-1c22a/overview
Hosting URL: https://esp32-1c22a.web.app
PS C:\Users\PICAIO\Desktop\Firebaseyoutube>
```

Introducción al Internet de las Cosas (IoT)

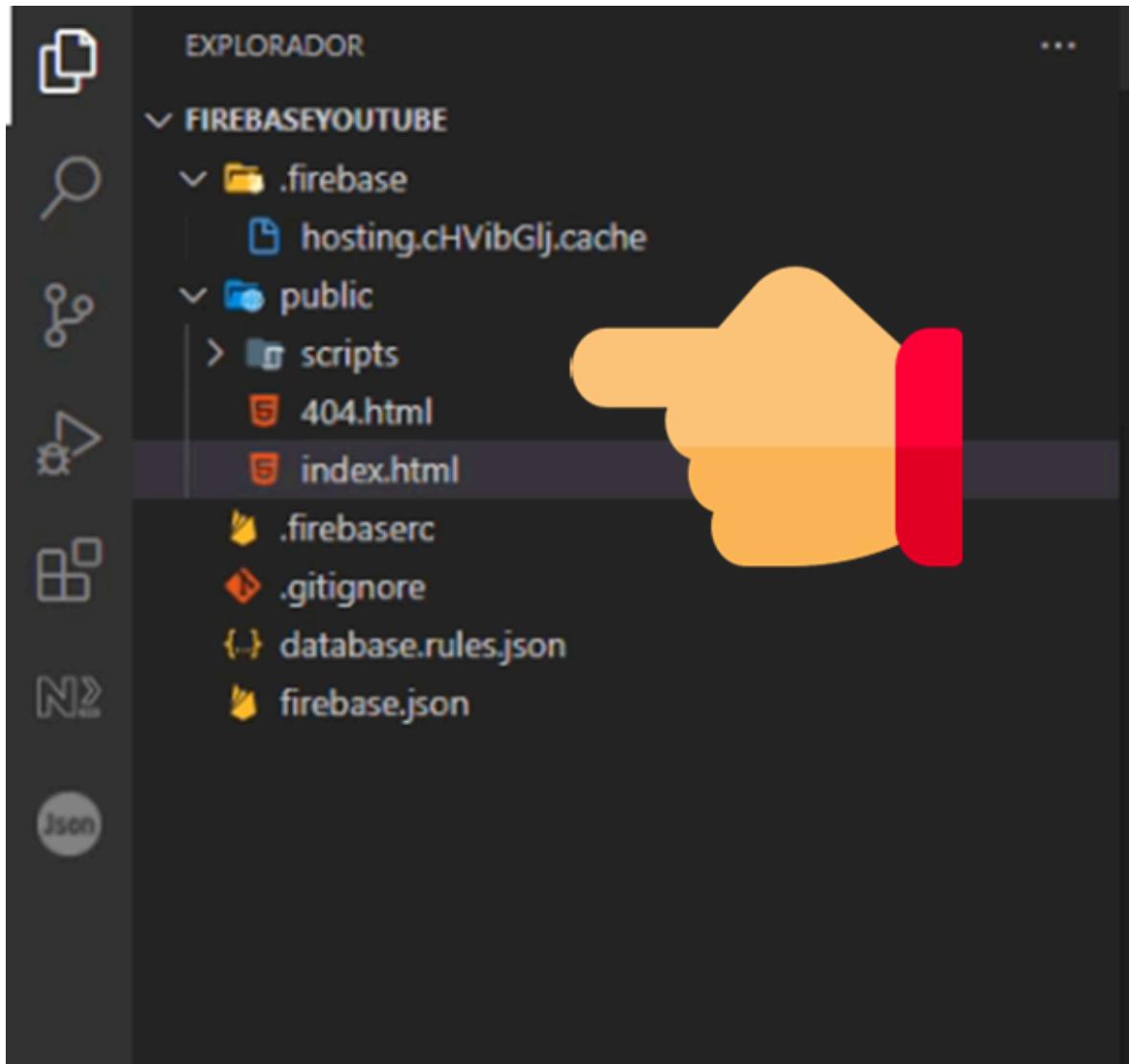
Y nos muestra la página por defecto que el crea.



Ahora si, vamos a subir nuestro código en la pagina, para darle un poco de logica, latoneria y pintura a nuestro proyecto. Debemos descargar los archivos en el siguiente [Github](#) y reemplazamos los archivos de **index.html**

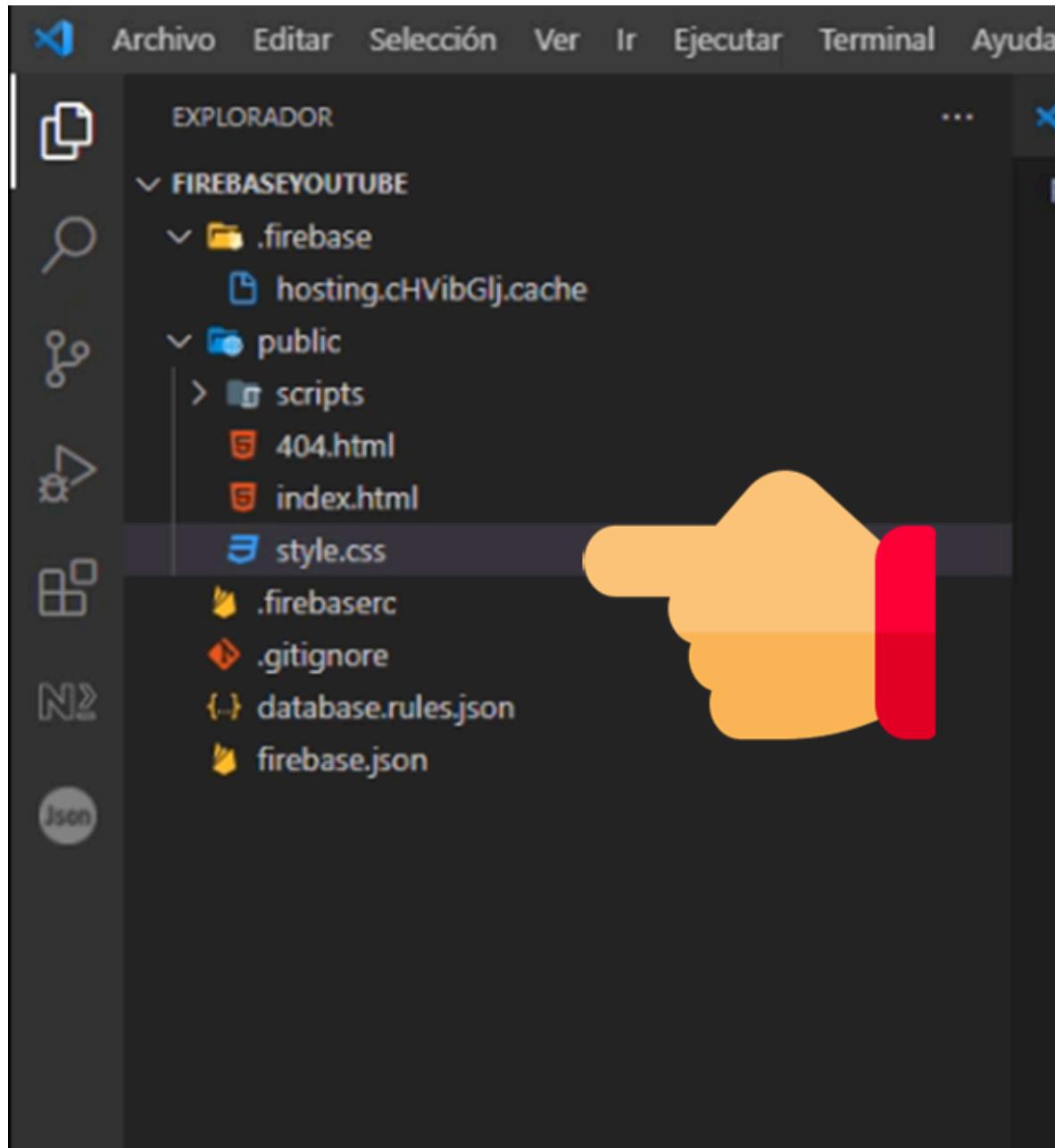
Luego copiamos la carpeta de **scripts**, dentro de la carpeta

Introducción al Internet de las Cosas (IoT)



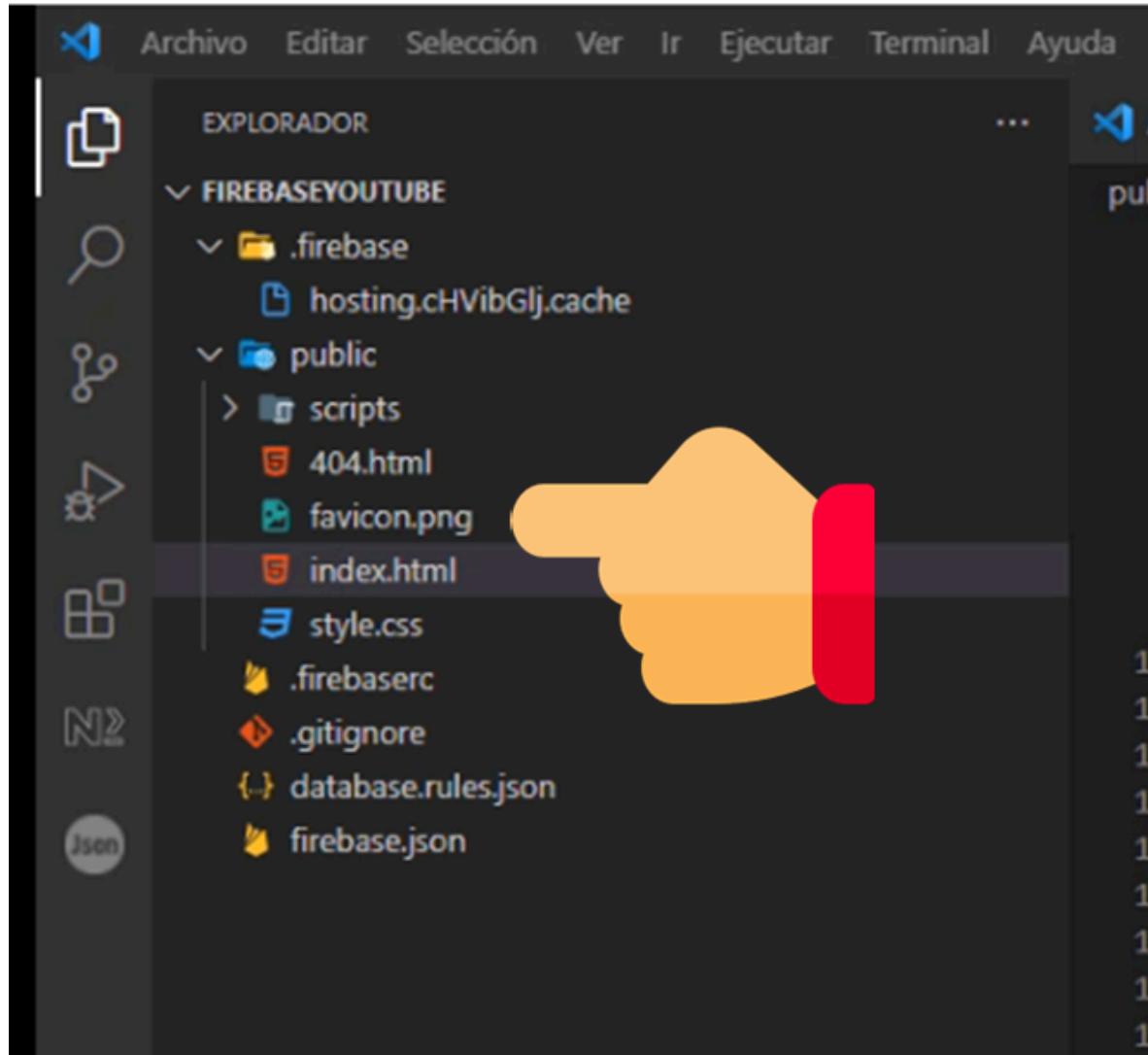
Y luego el archivos de style.css

Introducción al Internet de las Cosas (IoT)



Y por último la imagen png del favicon

Introducción al Internet de las Cosas (IoT)



Ahora, recuerda ¿el fragmento de código que les conté que debiamos guardar en un blog de notas que generaba el firebase?, pues aqui debemos reemplazarlo.

Introducción al Internet de las Cosas (IoT)

```
<!-- include only the Firebase features as you need -->
<script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-auth.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.8.1.firebaseio.js"></script>

<script>
// REPLACE WITH YOUR web app's Firebase configuration
const firebaseConfig = {
    apiKey: "AIzaSyBWoGwsF5CWN1v_fwL6RGXIkyYJ52uka38",
    authDomain: "esp32-60310.firebaseio.com",
    projectId: "esp32-60310",
    storageBucket: "esp32-60310.appspot.com",
    messagingSenderId: "526802567025",
    appId: "1:526802567025:web:c134d8148e2e2dcbe3c0d0",
    measurementId: "G-SZQ7L32LHV"
};

// Initialize firebase
firebase.initializeApp(firebaseConfig);

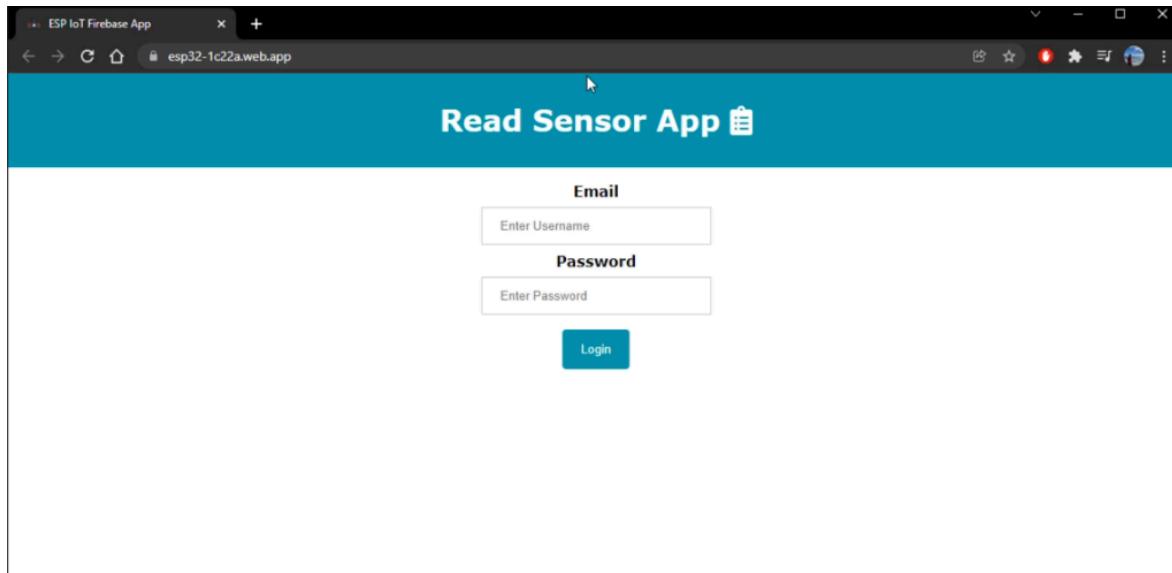
// Make auth and database references
const auth = firebase.auth();
const db = firebase.database();
```



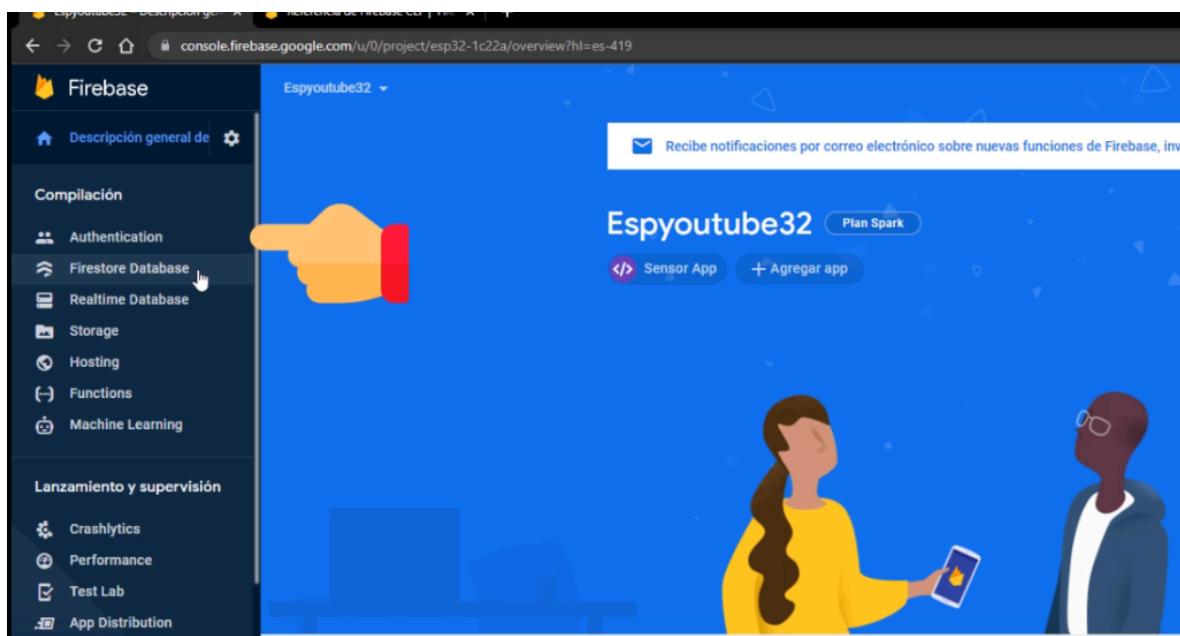
Ya que acomodamos los archivos, nuevamente le indicamos el **firebase deploy**, para subir nuevamente los archivos al hosting.

```
PS C:\Users\PICAIO\Desktop\Firebaseyoutube> firebase deploy
```

Si subimos correctamente los archivos, en el link que genera nuevamente, ya nos debería mostrar la siguiente interfaz.



Ahora el siguiente paso, es crear un usuario y una contraseña en firebase para poder ingresar. Entonces volvemos nuevamente a firebase y le damos en la opción **Authentication**



Introducción al Internet de las Cosas (IoT)

Le damos a **comenzar** y posteriormente mostrara la siguiente imagen, debemos escoger la opcion de correo electronico y contraseña

Proveedores de acceso

Agrega tu primer método de acceso y comienza a utilizar Firebase Auth

Proveedores nativos

Correo electrónico/contraseña

Teléfono

Anónimo

Proveedores adicionales

Google

Facebook

Play Juegos

Game Center

Apple

Github

Microsoft

Twitter

Yahoo

Le marcamos habilitar

Proveedores de acceso

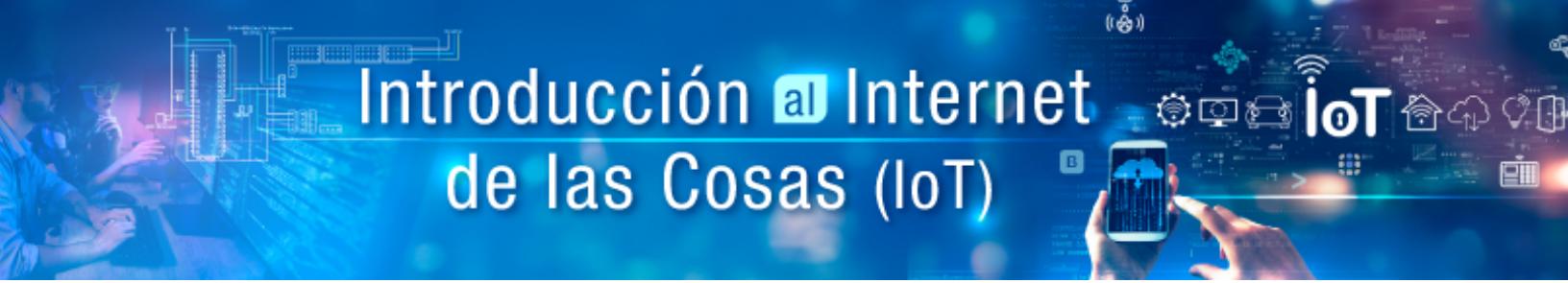
Correo electrónico/contraseña Habilitar

Permite que los usuarios se registren con su dirección de correo electrónico y contraseña. Nuestros SDK también proporcionan verificación de la dirección de correo electrónico, recuperación de contraseñas y primitivas de cambio de dirección de correo. [Más información](#)

Vínculo del correo electrónico (acceso sin contraseña) Habilitar

Cancelar Guardar

Y luego vamos a la pestaña de Users, para crear nuestro cliente



Introducción al Internet de las Cosas (IoT)

Authentication

Users Sign-in method Templates Usage

Crea prototipos y haz pruebas

Y creamos nuestro usuario, con un correo y contraseña

Identificador Proveedores Fecha de creación Fecha de acceso UID de usuario

① Agregar usuario

② Correo electrónico
③ Contraseña

Agrega un usuario con correo electrónico y contraseña

Cancelar Agregar usuario

Este proyecto todavía no tiene usuarios



Con esas credenciales ya podemos probar si ingresa a nuestro dashboard

Read Sensor App

Email

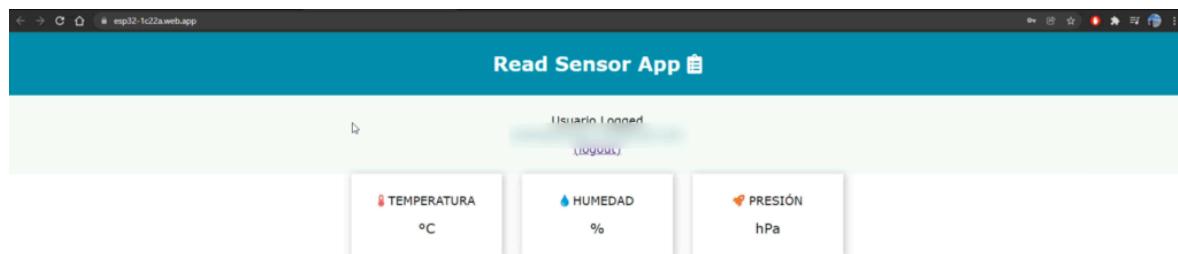
Password

.....

Login



Ahora ya ingresara a la siguiente pantalla



The screenshot shows a web browser window titled "Read Sensor App". The header bar is blue with the title "Read Sensor App". Below the header, there is a message "Usuario Logged (jorgejorge)". Three cards are displayed below the message:

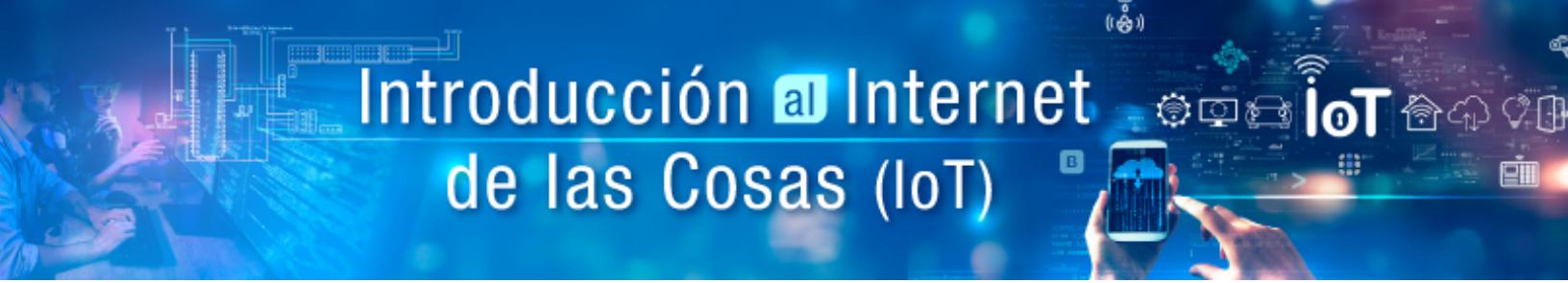
- TEMPERATURA °C
- HUMEDAD %
- PRESIÓN hPa

Finalmente debemos generar una ultima configuracion en firebase, para eso vamos a la opcion **Realtime Database**



The screenshot shows the Firebase console interface. On the left, a sidebar lists services: Descripción general de Firebase, Compilación (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), and Lanzamiento y supervisión (Crashlytics, Performance, Test Lab, App Distribution). A large yellow hand cursor points to the "Realtime Database" link in the "Compilación" section. The main content area is titled "Authentication" and shows tabs for Users, Sign-in method, Templates, and Usage. A modal window is open over the "Users" tab, prompting the user to "Crear un prototipo y haz pruebas de extracción". The modal includes a search bar, an "Identificador" input field containing "andresarango.es@gmail.c...", and a "Continuar" button at the bottom.

Le damos crear una base de datos y se nos muestra el siguiente modal, en el cual seleccionamos la opcion que aparece en la imagen



Reglas de seguridad de Realtime Database

Cuando definas la estructura de los datos, deberás crear reglas para protegerlos.
[Más información](#)

Comenzar en modo bloqueado
De forma predeterminada, tus datos son privados. El acceso de lectura/escritura de los clientes solo se otorgará como se indica en tus reglas de seguridad.

Comenzar en modo de prueba
Para permitir una configuración rápida, los datos se abren de forma predeterminada. Sin embargo, debes actualizar las reglas de seguridad en un plazo de 30 días a fin de habilitar el acceso de lectura/escritura a largo plazo para los clientes.

```
{  
  "rules": {  
    ".read": "now < 1646370000000", // 2022-3-4  
    ".write": "now < 1646370000000", // 2022-3-4  
  }  
}
```

! Las reglas de seguridad predeterminadas del modo de prueba permiten que cualquier usuario con acceso a tu referencia de base de datos pueda ver, editar y borrar todos los datos durante los siguientes 30 días.

Cancelar Habilitar

Vamos a la opción **Reglas**, pegamos las reglas que aparece en la imagen y le damos publicar

Datos Reglas Copias de seguridad Uso

1 Editar reglas Supervisar reglas 3

Cambios sin publicar | Publicar Descartar Zona de pruebas de reglas

```
1 // These rules grant access to a node matching the authenticated  
2 // user's ID from the Firebase auth token  
3 {  
  "rules": {  
    "UsersData": {  
      "Suid": {  
        ".read": "$uid === auth.uid",  
        ".write": "$uid === auth.uid"  
      }  
    }  
  }  
}
```

2

```
// These rules grant access to a node matching the authenticated  
// user's ID from the Firebase auth token  
{
```



Introducción al Internet de las Cosas (IoT)

```
"rules": {
  "UsersData": {
    "$uid": {
      ".read": "$uid === auth.uid",
      ".write": "$uid === auth.uid"
    }
  }
}
```

Ahora le damos en el engranaje en la parte superior y copiamos la clave de API, ya que la vamos a necesitar para el programa del ESP32

The screenshot shows the 'Configuración de proyecto' (Project Configuration) screen in the Firebase console. On the left, there's a sidebar with project settings like 'Descripción general de', 'Compilación', 'Lanzamiento y supervisión', and 'Analytics'. The main area has tabs for 'General', 'Cloud Messaging', 'Integraciones', 'Cuentas de servicio', 'Privacidad de los datos', 'Usuarios y permisos', and 'Verificación de aplicaciones (BETA)'. Under 'General', the 'Tu proyecto' section shows details: Nombre del proyecto (Espyoutube32), ID del proyecto (esp32-1c22a), Número de proyecto (593098947458), Ubicación predeterminada de los recursos de GCP (Aún sin seleccionar), and Clave de API web (AizaSyGJkW9Fz/zeuZjXuOLUuwheIcNe1NAQ-gY). Below this is the 'Entorno' section, which is currently set to 'Sin especificar'. At the bottom is the 'Configuración pública' section.

En el programa del ESP32, que ya descargamos cambiamos los datos necesarios que ya hemos creado anteriormente

Introducción al Internet de las Cosas (IoT)

firebase Arduino 1.8.10

Archivo Editar Programa Herramientas Ayuda



```
firebase §

// Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

// Insert your network credentials
#define WIFI_SSID "REPLACE_WITH_YOUR_SSID"
#define WIFI_PASSWORD "REPLACE_WITH_YOUR_PASSWORD"

// Insert Firebase project API Key
#define API_KEY "REPLACE_WITH_YOUR_PROJECT_API_KEY"

// Insert Authorized Email and Corresponding Password
#define USER_EMAIL "REPLACE_WITH_THE_USER_EMAIL"
#define USER_PASSWORD "REPLACE_WITH_THE_USER_PASSWORD"

// Insert RTDB URLdefine the RTDB URL
#define DATABASE_URL "REPLACE_WITH_YOUR_DATABASE_URL"

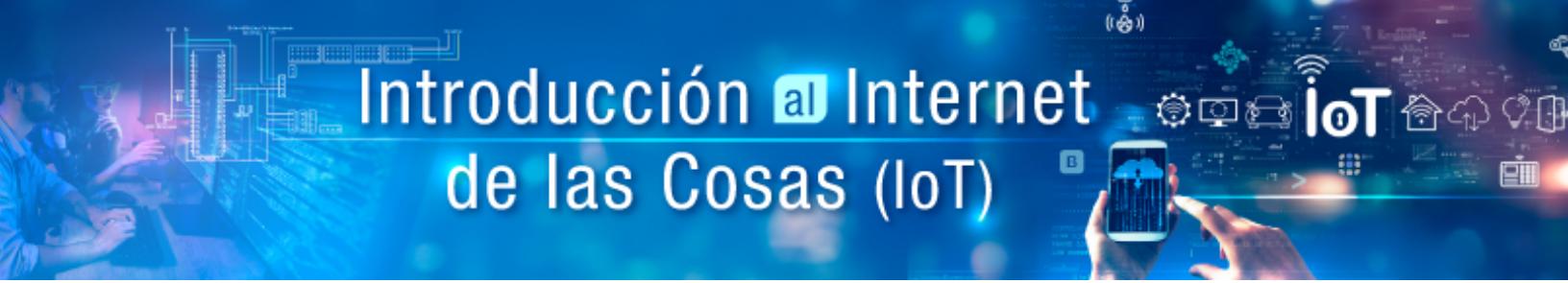
// Define Firebase objects
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// Variable to save USER UID
String uid;

// Variables to save database paths
```



Para la **DATABASE_URL**, debemos ingresar a la siguiente opción y copiar la url que nos muestra en la imagen.



The screenshot shows the Firebase Realtime Database console. The left sidebar has a dark theme with white text. The 'Realtime Database' option is highlighted with a red box. The main area is titled 'Realtime Database' and shows the URL 'https://esp32-1c22a-default.firebaseio.com/'. Below the URL is a note: 'Ubicación de la base de datos: Estados Unidos (us-central1)'. There are tabs for 'Datos', 'Reglas', 'Copias de seguridad', and 'Uso'. A note at the top says: 'Protege tus recursos de Realtime Database contra los abusos, como fraudes de facturación o suplantación de identidad.' and a link to 'Configurar la Verificación de aplicaciones'.

Finalmente, le damos programar al ESP32 y deberia generar el envio de datos desde la placa a firebase, y poder ver los datos desde la página web

The left screenshot shows a web browser window titled 'Read Sensor App'. It displays three sensor readings: 'TEMPERATURA' (0.20 °C), 'HUMEDAD' (0.40 %), and 'PRESIÓN' (0.60 hPa). The right screenshot shows a terminal window with log output from the ESP32. The logs show successful writes to the Firebase database under the path '/UsersData/mGGU3py3mMOPN7p1Z7cuiQkcfDj2'. The logs include:

```
Token info: type = id token, status = ready
Getting User UID
User UID: mGGU3py3mMOPN7p1Z7cuiQkcfDj2
Writing value: 0.20 on the following path: /UsersData/mGGU3py3mMOPN7p1Z7cuiQkcfDj2/temperature
PASSED
PATH: /UsersData/mGGU3py3mMOPN7p1Z7cuiQkcfDj2/temperature
TYPE: float
Writing value: 0.40 on the following path: /UsersData/mGGU3py3mMOPN7p1Z7cuiQkcfDj2/humidity
PASSED
PATH: /UsersData/mGGU3py3mMOPN7p1Z7cuiQkcfDj2/humidity
TYPE: float
Writing value: 0.60 on the following path: /UsersData/mGGU3py3mMOPN7p1Z7cuiQkcfDj2/pressure
PASSED
PATH: /UsersData/mGGU3py3mMOPN7p1Z7cuiQkcfDj2/pressure
TYPE: float
```