Hoy en día los sistemas IoT se encuentran en todo lugar y crecen de manera exponencial y son vitales para el desarrollo de la 4.0.

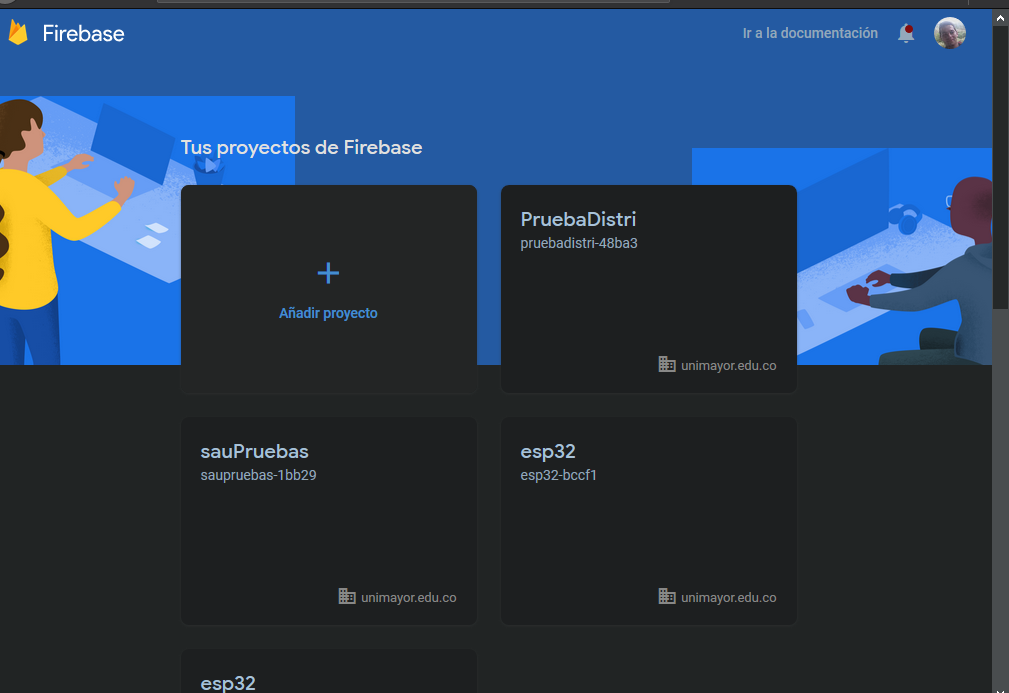
Por esto, en este proyecto pretende mostrar paso a paso, como podemos llevar datos en tiempo real a la nube haciendo uso del potente Esp32 y al igual consultar estos datos desde una aplicación web realizada para la consulta de los datos con node.js y algo de HTML y CSS.

**Requerimientos**

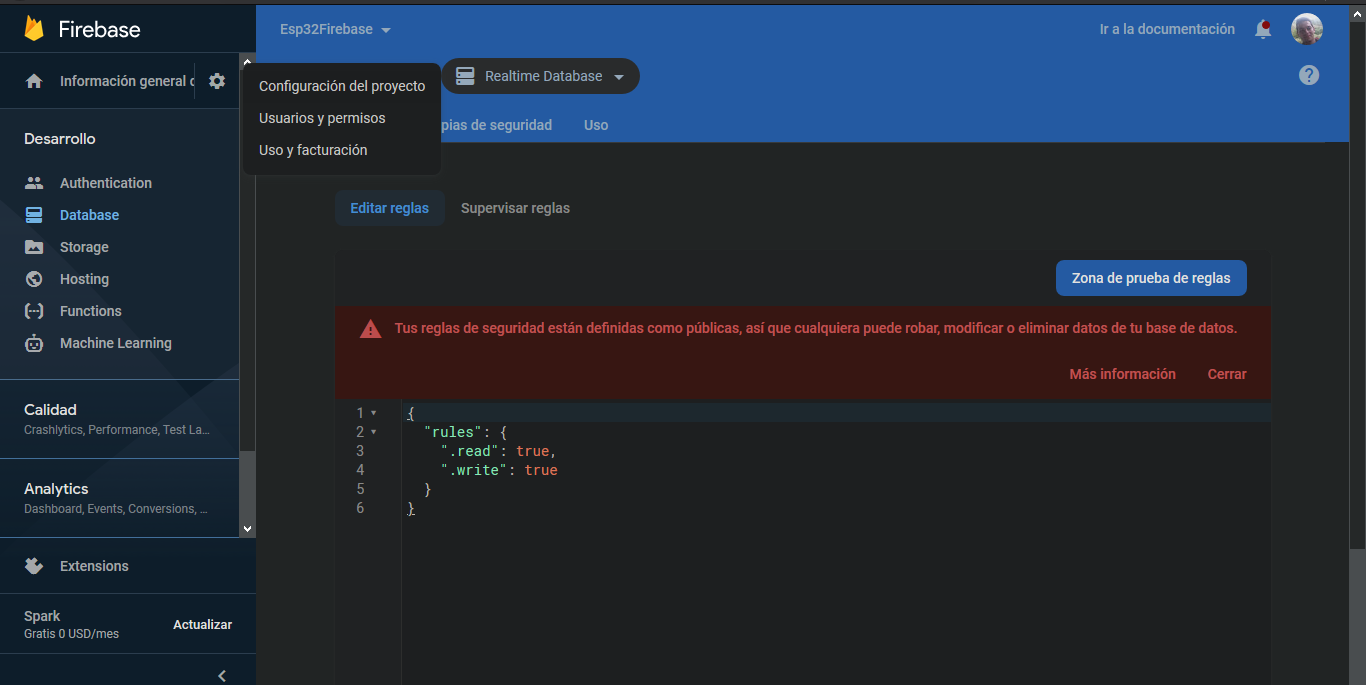
* Esp32 .
* Cuenta Gmail.
* VSCode0
* Extensión Platformio o el IDE de Arduino.
* Node.js

**Configuración del proyecto Firebase.**

Inicialmente, debemos ir a firebase y crear una cuenta, la cual debe ser autenticada con nuestro GmaIl. Cuando ya tenemos la cuenta vamos a la consola para poder crear un nuevo proyecto, crearlo es muy sencillo, solo sigues los pasos y listo.



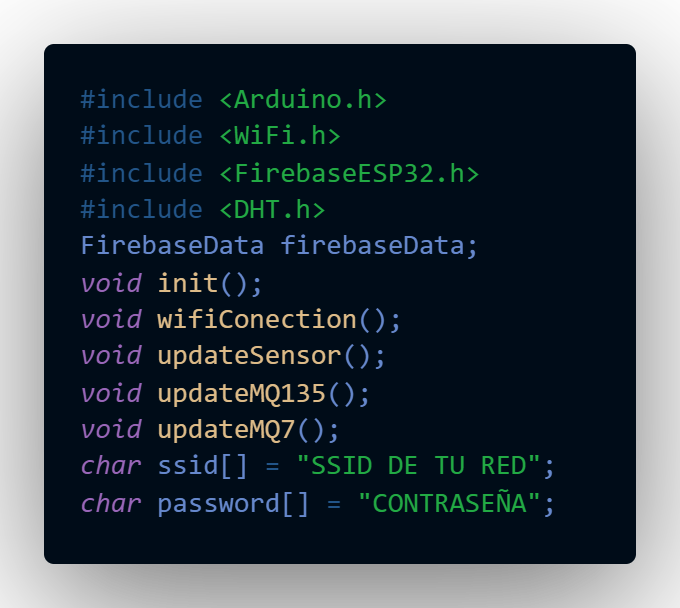
Esperamos a que nuestro proyecto esté listo y creamos una Realtime Database y como estamos en un proyecto de prueba, nuestras reglas pueden ser para modo de prueba. “Si tu proyecto sale al cliente debes configurar estas reglas para que no cualquiera pueda leer y escribir en tu base de datos”



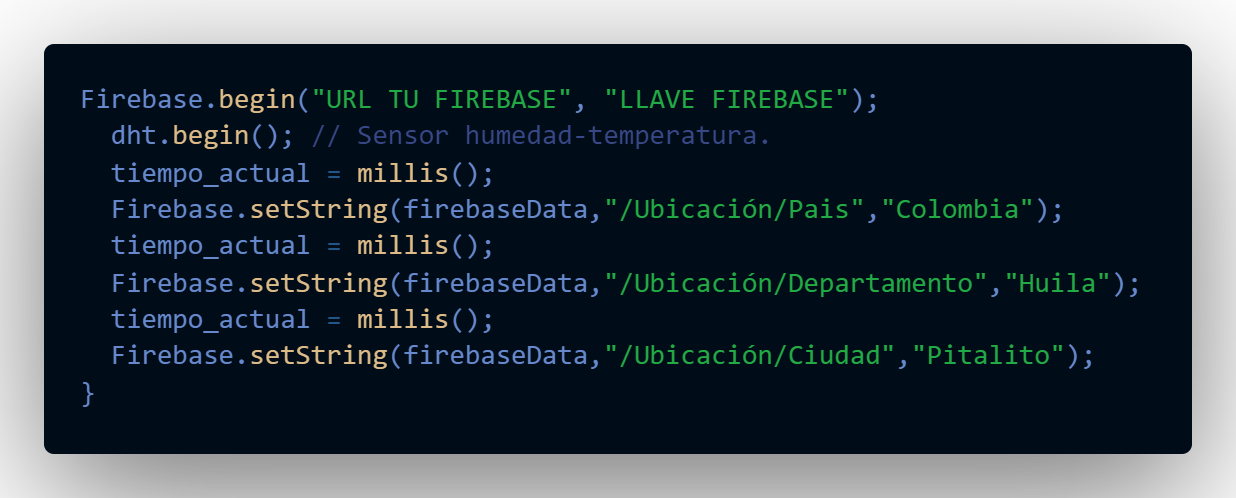
Posterior a esto, nuestra base de datos no tendrá ningún dato y no es necesario empezar a crear el árbol ya que este lo empezamos a crear con nuestro Esp32.

**Configuración Esp32.**

Para nuestro Esp32 debemos tener la librería FirebaseEsp32 <https://github.com/mobizt/Firebase-ESP32> , Wifi y en este caso a DHT ya que usaremos un sensor de temperatura DHT11. Estas librerías deben ser incluidas en el proyecto, así como el SSID Y PASSWORD de su red Wfi.



Luego debemos configurar la conexión a nuestra base de datos con la librería Firebase con la que tendremos acceso a la base de datos para poder escribir sobre ella.



**Codificar la aplicación web.**

Como requisito debemos tener instalado node.js, teniendo en cuenta el requerimiento creamos una carpeta, se agrega a VSCode e iniciamos una terminal en la que ingresaremos los comandos para iniciar el proyecto con: npm init -y.

Luego de tener el proyecto de node.js debemos instalar las dependencias y los módulos necesarios y requeridos para el proyecto:

npm i express express-handlebars morgan firebase-admin

De igual manera requerimos del módulo nodemon: npm I nodemon -D, que nos permite reiniciar el servidor cuando guardamos algo nuevo en el proyecto. Para poder usarlo debemos crear un script en el archivo package.js

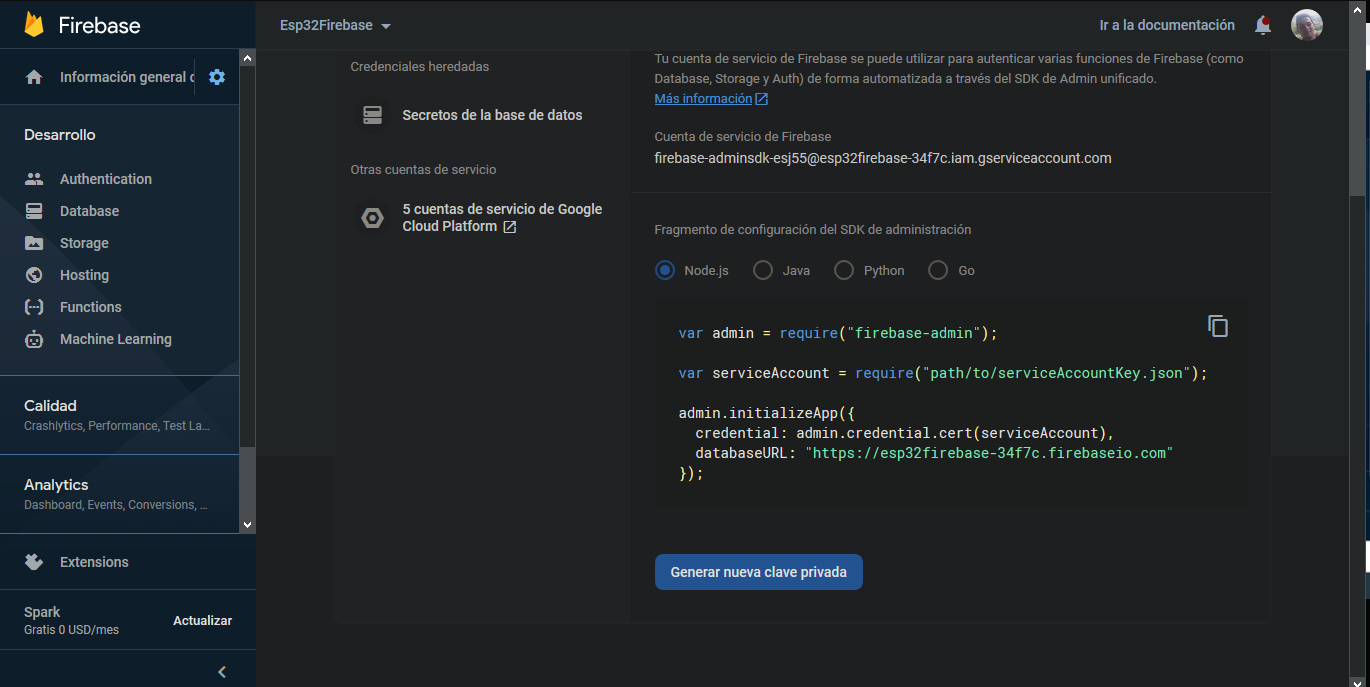
|  |
| --- |
| "scripts": { |
|  |

"dev": "nodemon src/index.js"

En nuestro index.js requerimos agregar la URL de nuestro proyecto firebase, esta URL la encontraras en tu Realtime DataBase.



Teniendo la conexión a la URL, vamos a configuraciones de nuestro proyecto en firebase para generar una nueva llave privada, la llave es un archivo JSON, lo descargamos y lo agregamos a la carpeta del proyecto de node.js.



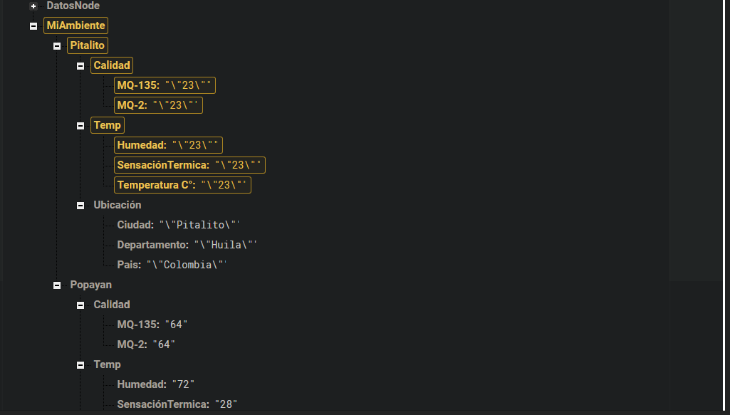
Posteriormente debemos tener en cuenta que se debe crear la variable de entorno para brindar más seguridad a nuestro proyecto, para realizar este paso, creamos un archivo README.MD en el que agrega la ruta que nos proporciona firebase y debemos especificar la ruta del archivo JSON:

$env:GOOGLE\_APPLICATION\_CREDENTIALS="C:\Users\username\Downloads\service-account-file.json"

Con los pasos anteriores ya hemos configurado las credenciales de la base de datos, para poder escribir y leer sobre ella, lo demás será código HTML-CSS Y JAVASCRIPT.

En las imágenes posteriores se observa como leer los datos de nuestra base de datos con el método post. Y finalmente una donde puede observar como los datos son actualizados en tiempo real por el Esp32.





Si quieres conocer el código completo del proyecto, tanto de la aplicación web y del Esp32 visita el repositorio <https://github.com/crojastriana3/Esp32Firebase>

Tu imaginación es el límite.