Manual TÉcnico

Grupo 5: George Gaibor, Karla Aviles, Andrés Torres, Melanie Cruz, Leonardo Ortiz

Escuela Superior Politécnica del Litoral

Programación de Sistemas Telemáticos



1. Resumen

Se construyó un prototipo capaz de registrar los sólidos totales disueltos en una muestra de agua, para así poder clasificar el agua, identificando la aptitud para el consumo. Se usó un modulo WiFi ESP8266 y un sensor TDS, el módulo WiFi se encargó de enviar los datos por la red y el sensor se encargó de medir las impurezas en el agua. Se realizó la conexión a través de la red WiFi con una base de datos en Firebase, esta recolectó los datos y los envió a una página web, la cual mostró las gráficas junto con el nivel de aptitud de consumo del agua. El usuario podrá supervisar constantemente el estado del agua, y así conocer cuando su agua se encuentra en un estado aceptable para el consumo.

1. Descripción

El consumo de aguas que no cuentan con un rango de purificación moderado, puede ser muy perjudicial para la salud, muchas veces se consume agua que no se encuentra en un estado consumible, a pesar de que no se logran percibir las impurezas, estos componentes no deseados pueden provocar danos a largo plazo, sin que el consumidor lo note hasta presentar síntomas. Dado esto, es necesario la revisión constante del agua que consumimos, conociendo así si esta puede ser ingerida sin generar cambios negativos en nuestro organismo. La constante supervisión del agua se puede realizar incluso en el propio filtro de nuestros hogares, dado que este al necesitar un mantenimiento, si este no se realiza a menudo, el agua que ingerimos podría llegar a presentar contaminantes, exceso de minerales o presencia de bacterias.

1. Objetivos

**Objetivo general**

* Desarrollar un prototipo capaz de determinar la aptitud para el consumo del agua, midiendo los sólidos totales disueltos en la misma, haciendo uso de un sensor TDS y mostrando los datos en una aplicación web.

**Objetivos específicos**

* Registrar en una base de datos, la cantidad de solidos totales disueltos en el agua, medidos en tiempo real.
* Categorizar los sólidos totales disueltos en el agua, estimando así su calidad y aptitud para el consumo humano.
* Crear una aplicación web, en la cual se muestren los registros generados, empleando gráficas para una mejor visualización de los datos a través del tiempo.

1. Solución

Se analizó la problemática, y se generó una solución factible y asequible, la cual consistió en el desarrollo de un prototipo que logre realizar mediciones, en tiempo real, de los solidos totales disueltos en el agua, enviando los datos recogidos y mostrándolos en una pagina web a manera de gráficas, de esta manera el usuario tenga la facilidad de acceder a sus datos en cualquier momento y poder analizar el estado del agua en análisis.

1. Prototipo

Se construyó un sistema capaz de medir los sólidos totales disueltos en el agua y que logró enviarlos a una base de datos en Firebase, luego se creó una pagina web que mostró las graficas con los datos registrados previamente por el dispositivo. Para poder enviar los datos por Wifi a la base de datos se usó un módulo ESP8266, el cual permitió la conectividad entre el sensor y la base de datos. De esta manera el usuario ha sido capaz de supervisar constantemente el estado del agua en análisis, para así conocer si esta es apta para el consumo o no.

1. Recursos

Recursos de hardware y de software

**Hardware**

* Modulo WiFi NodeMCU ESP8266
* TDS Meter V1.0

**Software**

* Firebase
* Arduino IDE

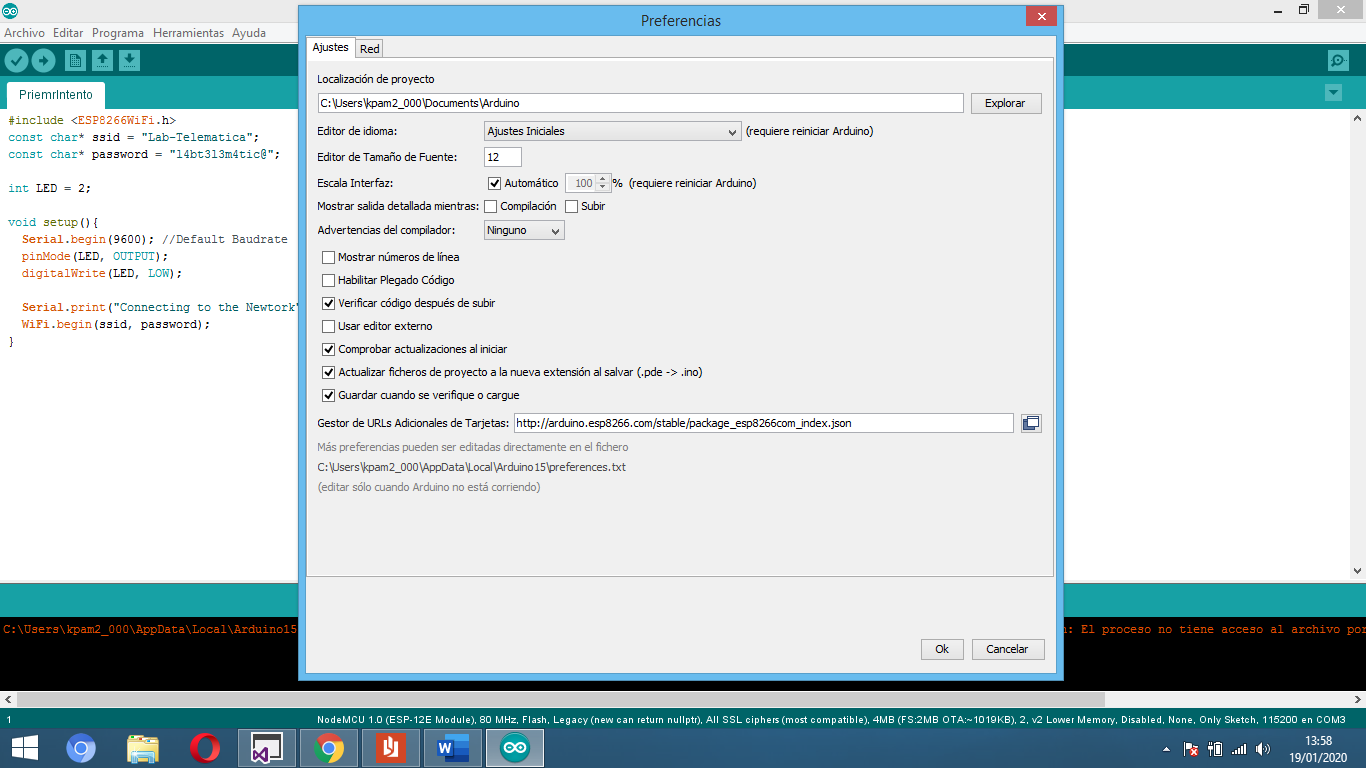
1. Implementación del prototipo

Explicación paso a paso de la implementación del proyecto

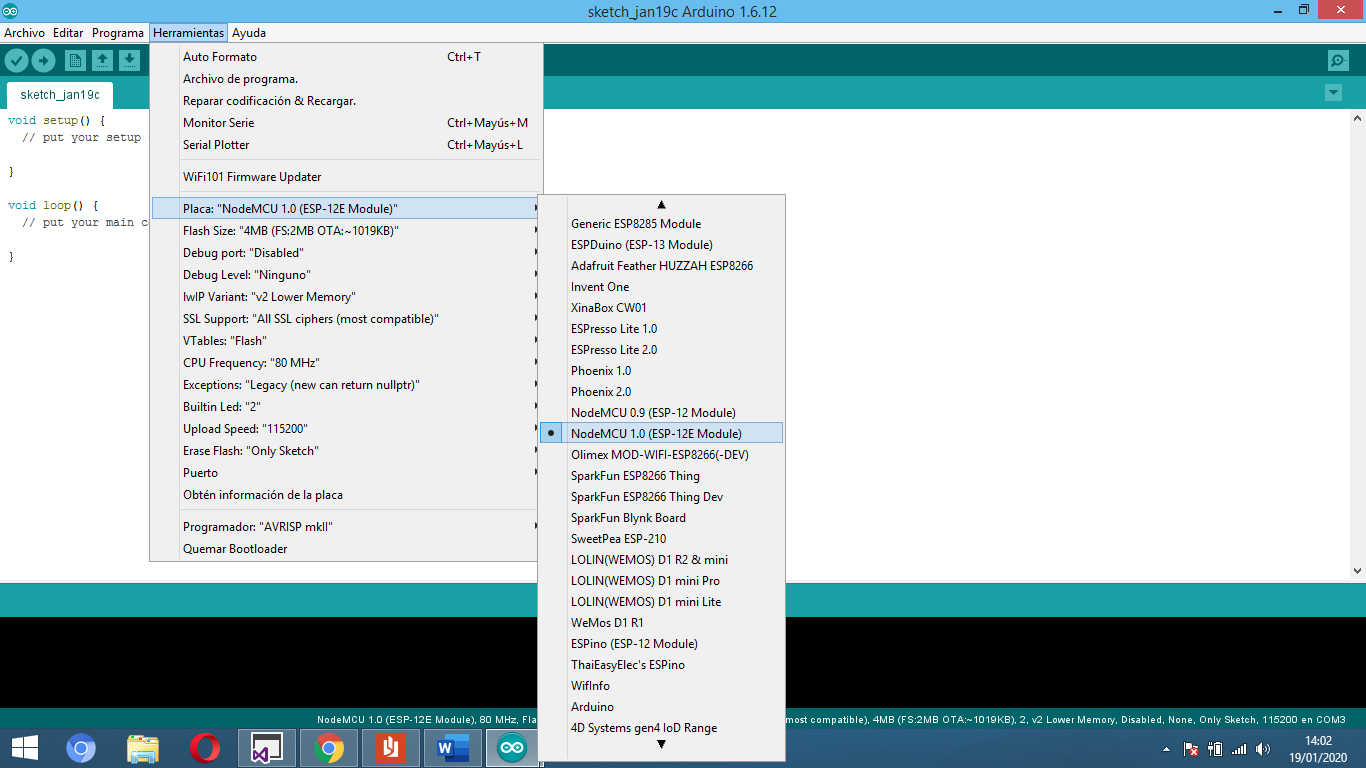
Diagramas de diseño del proyecto, diagrama de circuito, diagrama del modelo entidad-relación, diagrama de casos UML, diagrama de despliegue

Descripción de los campos, tipos de datos creados en la base de datos, y el código SQL Explicación del código fuente completo desarrollado con los comentarios correspondientes

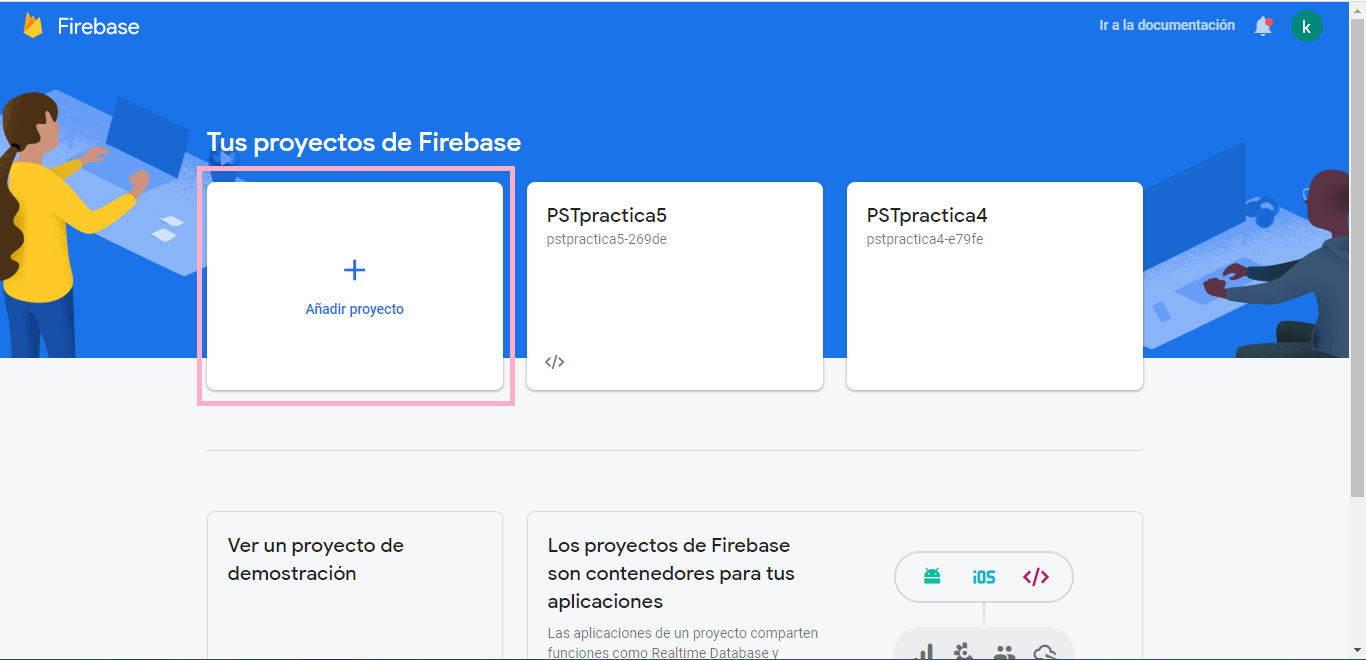
1. Conectar el módulo WiFi al sensor TDS (EXPLICAR LOS PINES Y PONER UNA FOTO DE LA CONEXIÓN)
2. Conectar el módulo WiFi a la computadora y abrir *Archivo – Preferencias*:



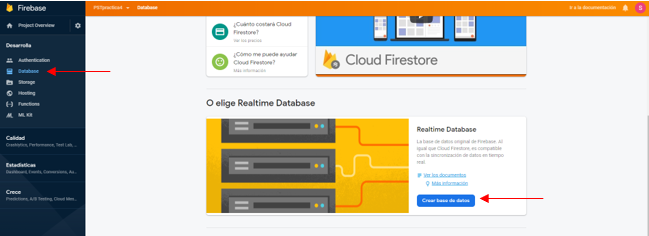
1. En *Gestor de URLs Adicionales de Tarjetas,* copiar el siguiente link: ***http://arduino.esp8266.com/stable/package\_esp8266com\_index.json***
2. Luego de que todos los paquetes se hayan descargado, en la pantalla principal del IDE, seleccionamos *Herramientas-Placas-NodeMCU 1.0 (ESP-12E Module)*



1. Ingresar a su cuenta de Google, ir a: <https://firebase.google.com/>. En la esquina superior derecha seleccionar *Ir a Consola.* Crea un proyecto con el nombre que desee.



1. Una vez creado el proyecto, dirigirse a *DataBase* y luego a *Crear Base de Datos.*





1. Abrir el IDE de Arduino y copiar el siguiente código:

#include <ArduinoJson.h>

#include <FirebaseArduino.h>

#include <ESP8266WiFi.h>

**#define FIREBASE\_HOST "pryt-pst-g5.firebaseio.com"**

**#define FIREBASE\_AUTH "YxwA6o8DQY5jMTicwmpLRgSv5HeQnop8bf4pnX9x"**

**#define WIFI\_SSID "Lab-Telematica"**

**#define WIFI\_PASSWORD "l4bt3l3m4tic@"**

//definiciones para el sensor TDS

#define TdsSensorPin A0

#define VREF 5.0 // analog reference voltage(Volt) of the ADC

#define SCOUNT 30 // sum of sample point

int analogBuffer[SCOUNT]; // store the analog value in the array, read from ADC

int analogBufferTemp[SCOUNT];

int analogBufferIndex = 0,copyIndex = 0;

float averageVoltage = 0,tdsValue = 0,temperature = 25;

int c = 0;

void setup()

{

Serial.begin(9600);

delay(2000);

Serial.println('\n');

wifiConnect();

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);

delay(10);

//setup sensor TDS

pinMode(TdsSensorPin,INPUT);

}

void loop()

{

static unsigned long analogSampleTimepoint = millis();

if(millis()-analogSampleTimepoint > 40U) //every 40 milliseconds,read the analog value from the ADC

{

analogSampleTimepoint = millis();

analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin); //read the analog value and store into the buffer

analogBufferIndex++;

if(analogBufferIndex == SCOUNT)

analogBufferIndex = 0;

}

static unsigned long printTimepoint = millis();

if(millis()-printTimepoint > 800U)

{

printTimepoint = millis();

for(copyIndex=0;copyIndex<SCOUNT;copyIndex++)

analogBufferTemp[copyIndex]= analogBuffer[copyIndex];

averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) \* (float)VREF / 1024.0; // read the analog value more stable by the median filtering algorithm, and convert to voltage value

float compensationCoefficient=1.0+0.02\*(temperature-25.0); //temperature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02\*(fTP-25.0));

float compensationVolatge=averageVoltage/compensationCoefficient; //temperature compensation

tdsValue=(133.42\*compensationVolatge\*compensationVolatge\*compensationVolatge - 255.86\*compensationVolatge\*compensationVolatge + 857.39\*compensationVolatge)\*0.5; //convert voltage value to tds value

//Serial.print("voltage:");

//Serial.print(averageVoltage,2);

//Serial.print("V ");

Serial.print("TDS Value:");

Serial.print(tdsValue,0);

Serial.println("ppm");

Firebase.setFloat("ppm/"+String(c)+"/valor",tdsValue);

c++;

if (Firebase.failed()) {

Serial.print("pushing tds value failed:");

Serial.println(Firebase.error());

return;

}

Serial.print("pushed: tds value");

}

delay(10);

if(WiFi.status() != WL\_CONNECTED)

{

wifiConnect();

}

delay(100);

}

void wifiConnect()

{

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); // Connect to the network

Serial.print("Connecting to ");

Serial.print(WIFI\_SSID); Serial.println(" ...");

int teller = 0;

while (WiFi.status() != WL\_CONNECTED)

{ // Wait for the Wi-Fi to connect

delay(1000);

Serial.print(++teller); Serial.print(' ');

}

Serial.println('\n');

Serial.println("Connection established!");

Serial.print("IP address:\t");

Serial.println(WiFi.localIP()); // Send the IP address of the ESP8266 to the computer

}

int getMedianNum(int bArray[], int iFilterLen)

{

int bTab[iFilterLen];

for (byte i = 0; i<iFilterLen; i++)

bTab[i] = bArray[i];

int i, j, bTemp;

for (j = 0; j < iFilterLen - 1; j++)

{

for (i = 0; i < iFilterLen - j - 1; i++)

{

if (bTab[i] > bTab[i + 1])

{

bTemp = bTab[i];

bTab[i] = bTab[i + 1];

bTab[i + 1] = bTemp;

}

}

}

if ((iFilterLen & 1) > 0)

bTemp = bTab[(iFilterLen - 1) / 2];

else

bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;

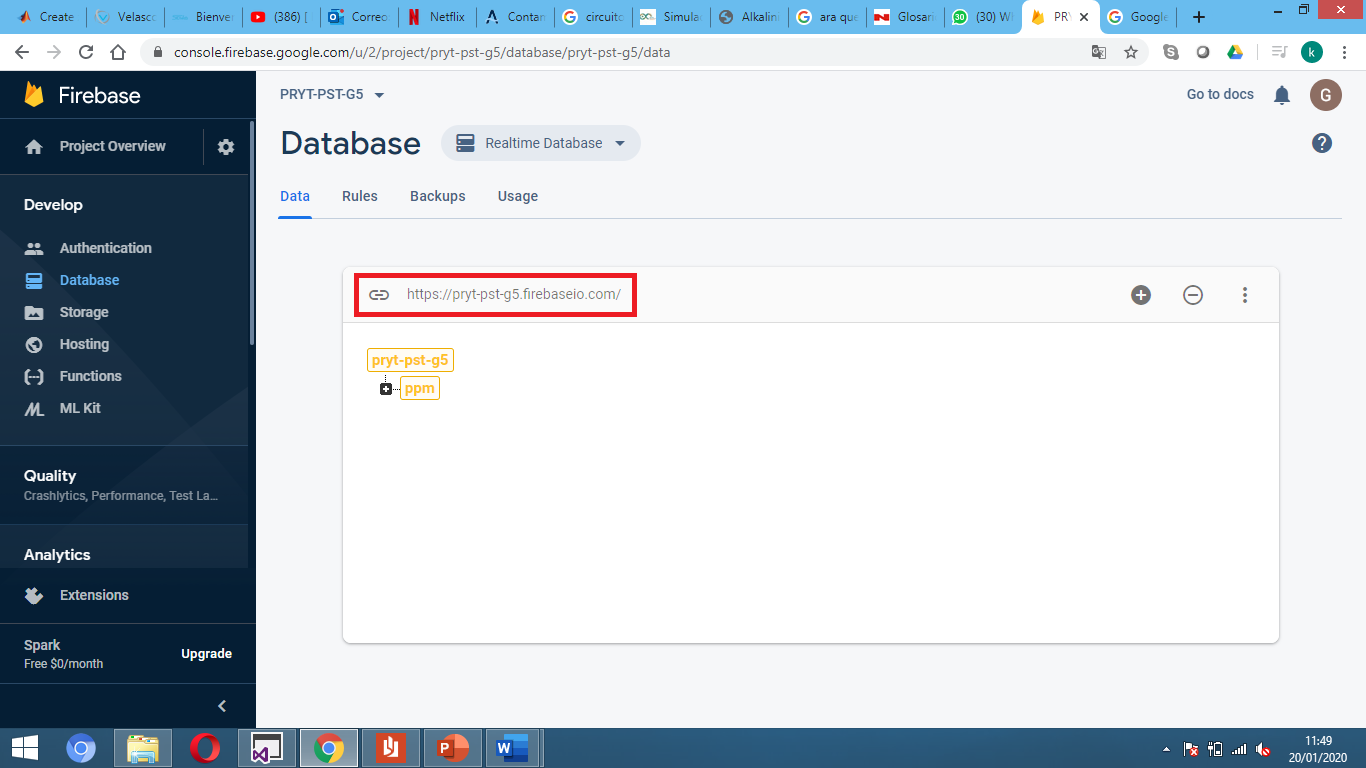
return bTemp;

}

1. Se deben corregir las líneas de código en letras rojas, para poder realizar la conexión a la base de datos que ya se generó. Las dos primeras líneas que se deben corregir, se muestran a continuación:

**ACTUAL: #define FIREBASE\_HOST "pryt-pst-g5.firebaseio.com"**

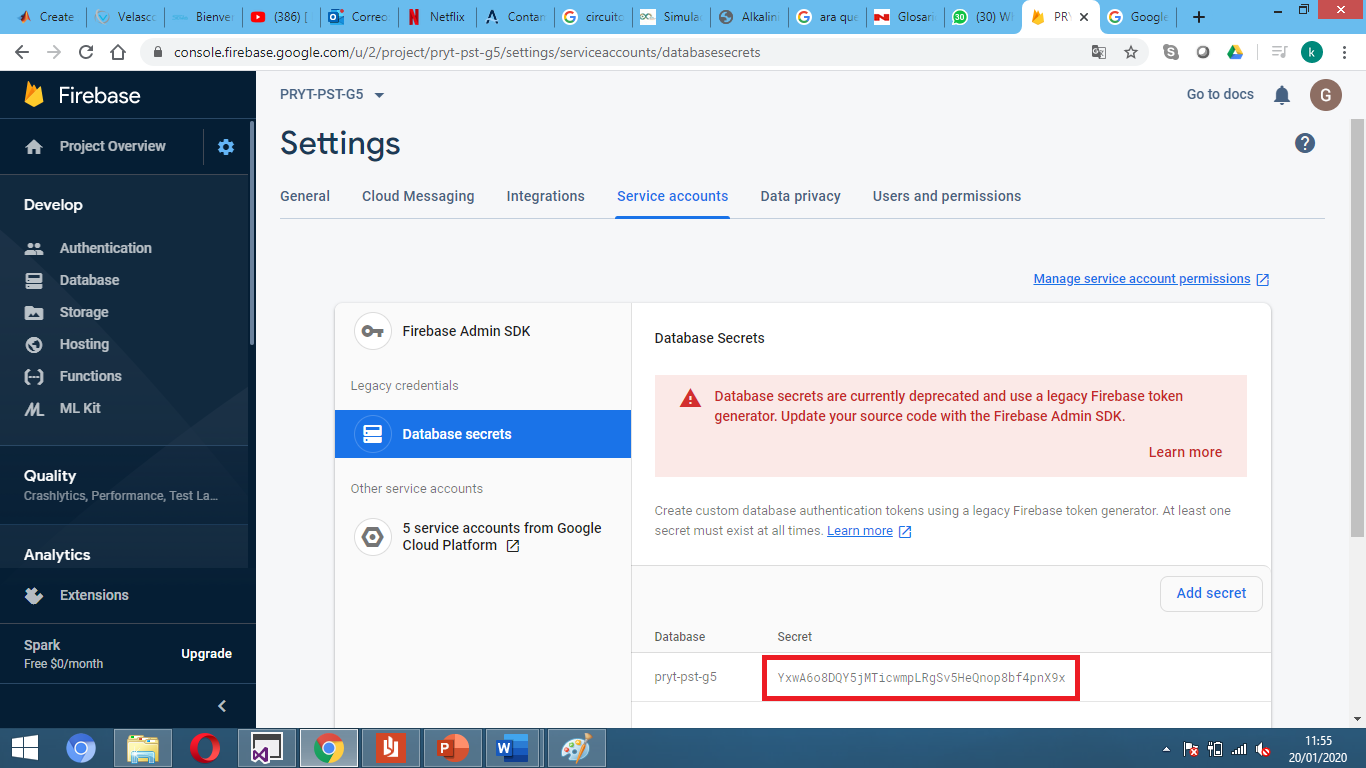
**CORRECCIÓN:** entre las comillas se debe pegar el link que se encuentra en la base de datos.

****

**ACTUAL: #define FIREBASE\_AUTH "YxwA6o8DQY5jMTicwmpLRgSv5HeQnop8bf4pnX9x"**

**CORRECCIÓN:** entre las comillas se debe pegar el

En la esquina superior izquierda, dar click en la rueda de Ajustes, seleccionar ***Project Settings – Service Accounts – DataBase secrets*** y copiar la clave de autenticación

****

**ACTUAL: #define WIFI\_SSID "Lab-Telematica"**

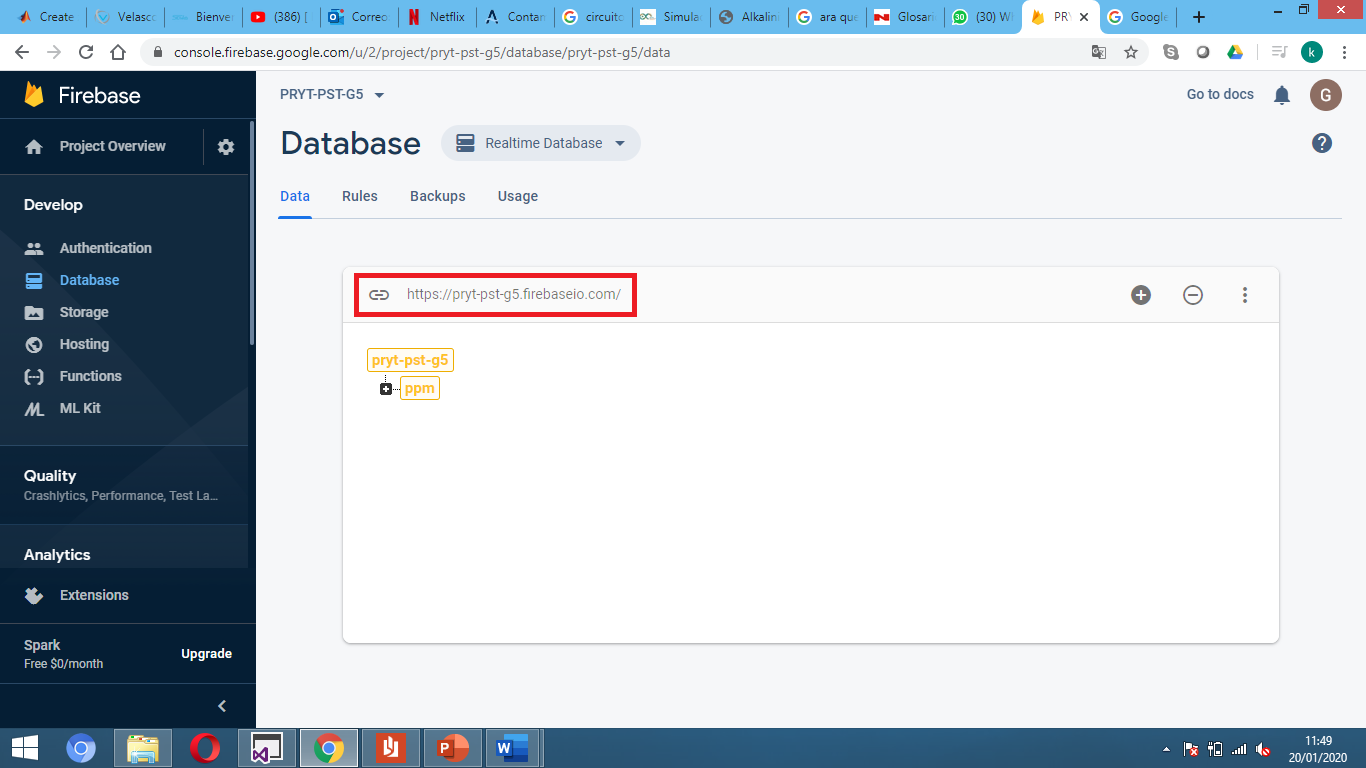
**ACTUAL: #define WIFI\_PASSWORD "l4bt3l3m4tic@"**

**CORREGIR:** Su usuario y contraseña de su red WiFi en la que se encuentre.

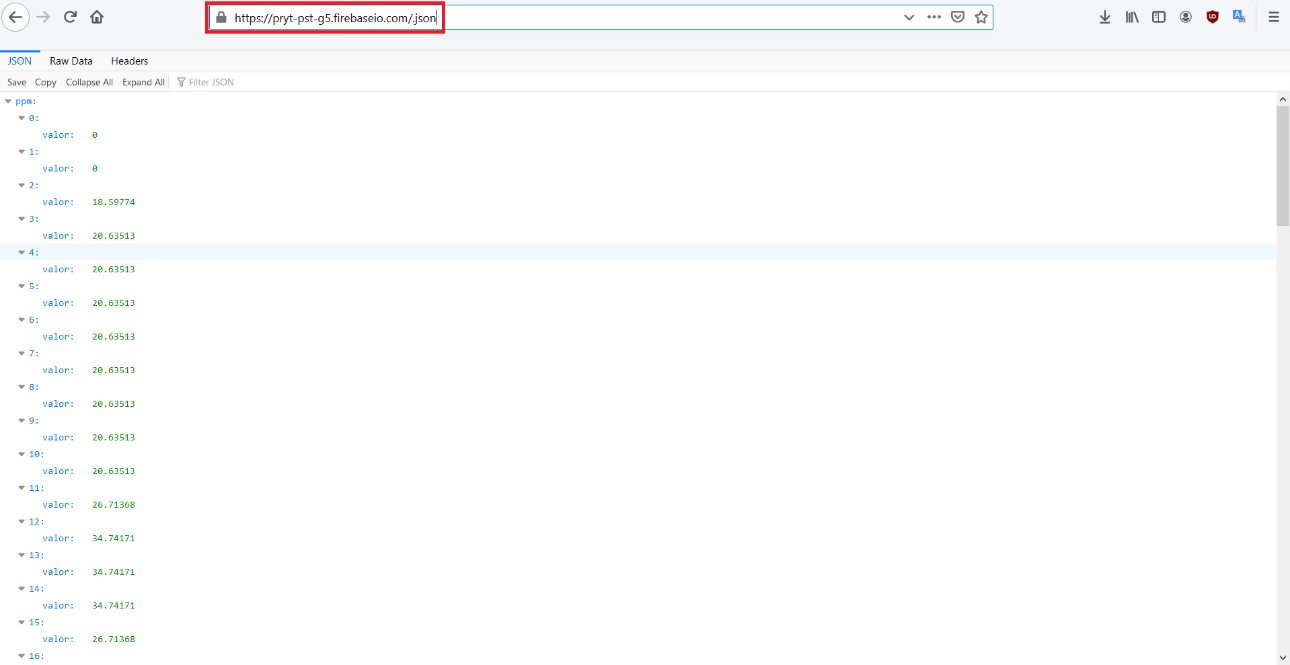
1. Una vez modificado el código, se debe corregir una librería:

Dirigirse a ***C:\Users\user\Documents\Arduino\libraries\firebase-arduino-0.3\src\FirebaseHttpClient.h,*** abrirlo con bloc de notas.

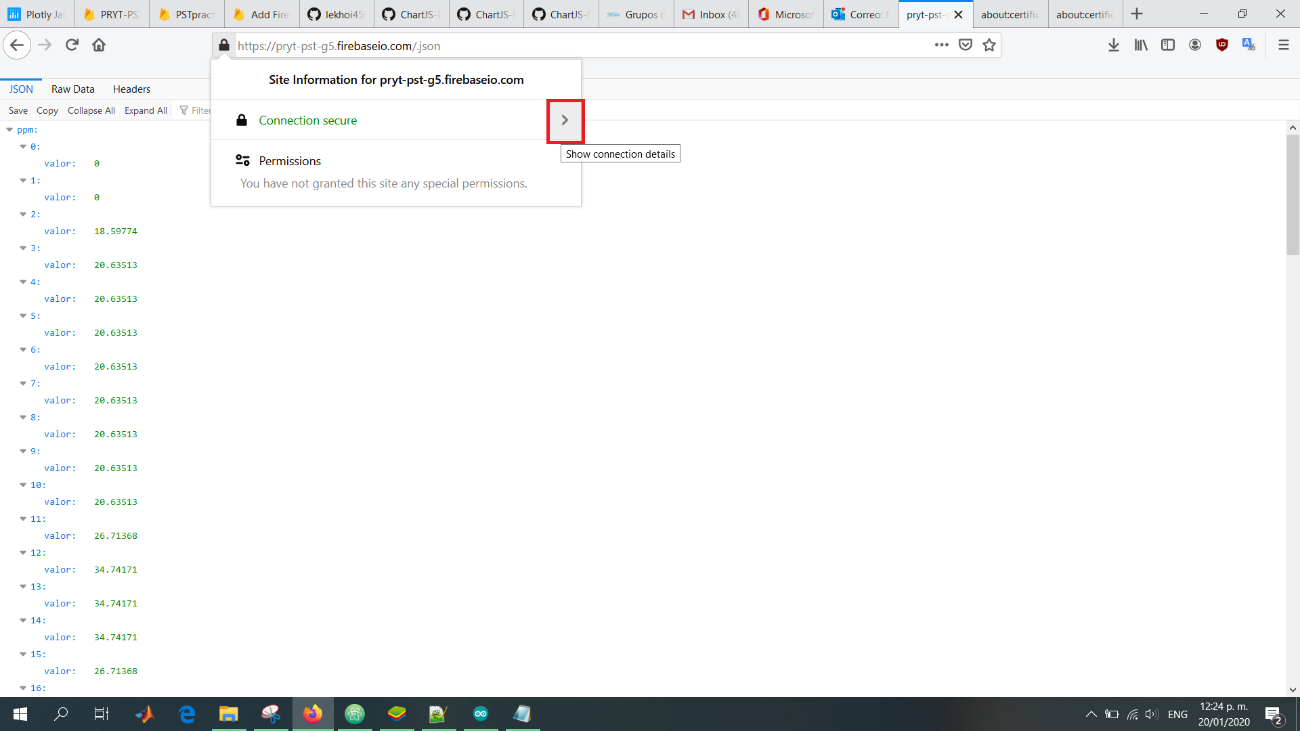
1. En la base de datos, copiar el siguiente link:

****

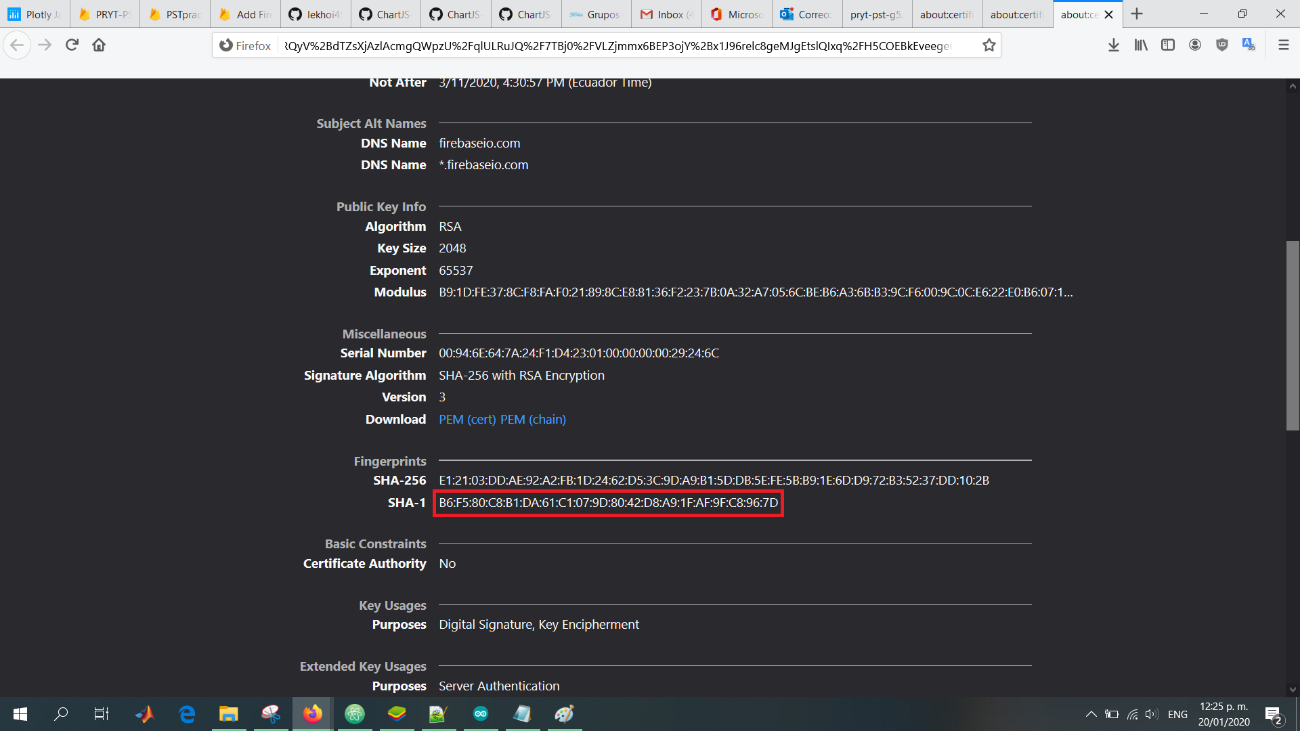
1. Abrir **FIREFOX**, en una nueva pestana pegar el link y anadir “.json” al final del mismo, presionar enter y esperar unos segundos. Aparecerá de la siguiente forma:



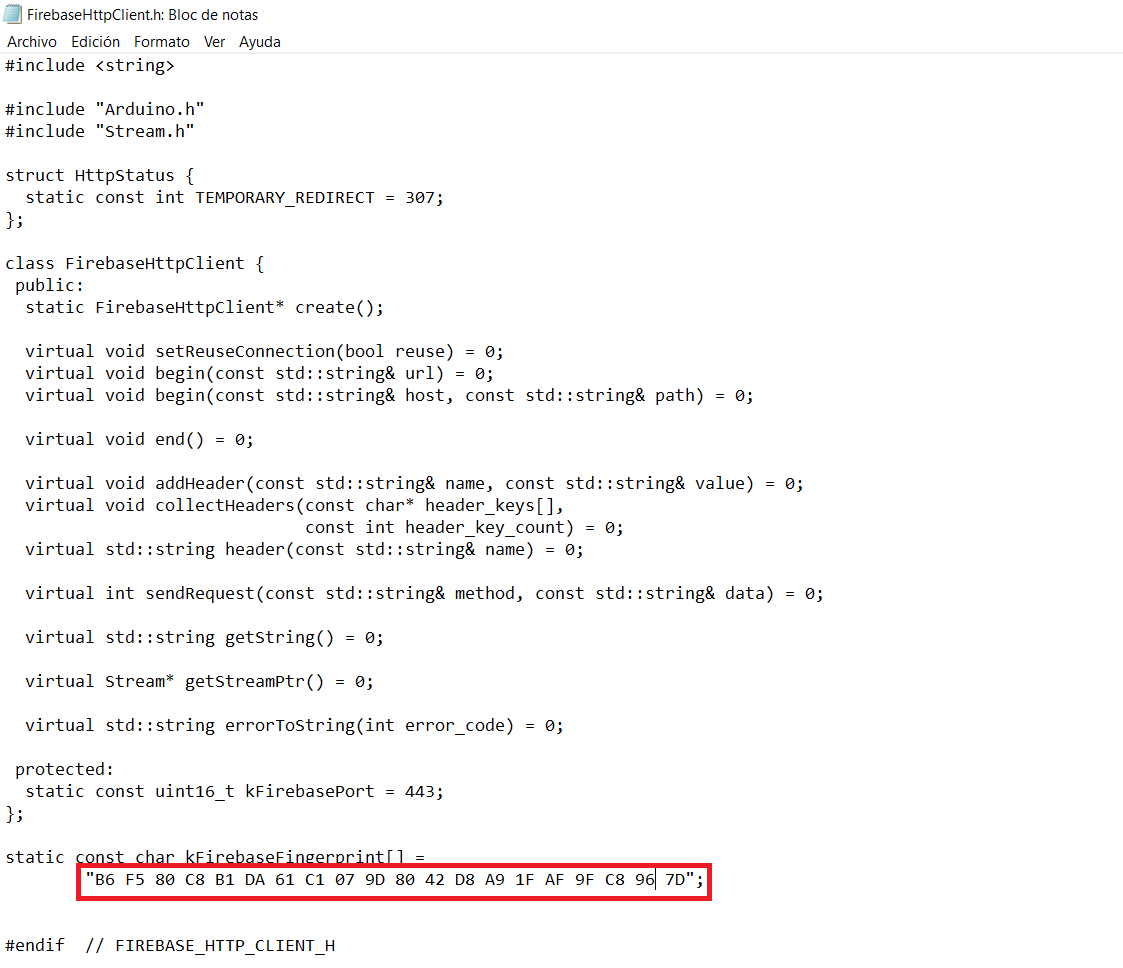
1. Dar click en el candado mostrado al lado del link y en la pestaña mostrada a continuación, al lado de ***Connection Secure*** y luego en ***More Information.***



1. En la pestana emergente seleccionar ***View Certificate*** y copiar el siguiente ID.



1. Pegarlo en el bloc de notas que se abrió en el paso 9, dentro de las comillas en el recuadro marcado.



1. Empezar a tomar datos, estos se enviarán a la base de datos.

**Creación de Sitio Web**

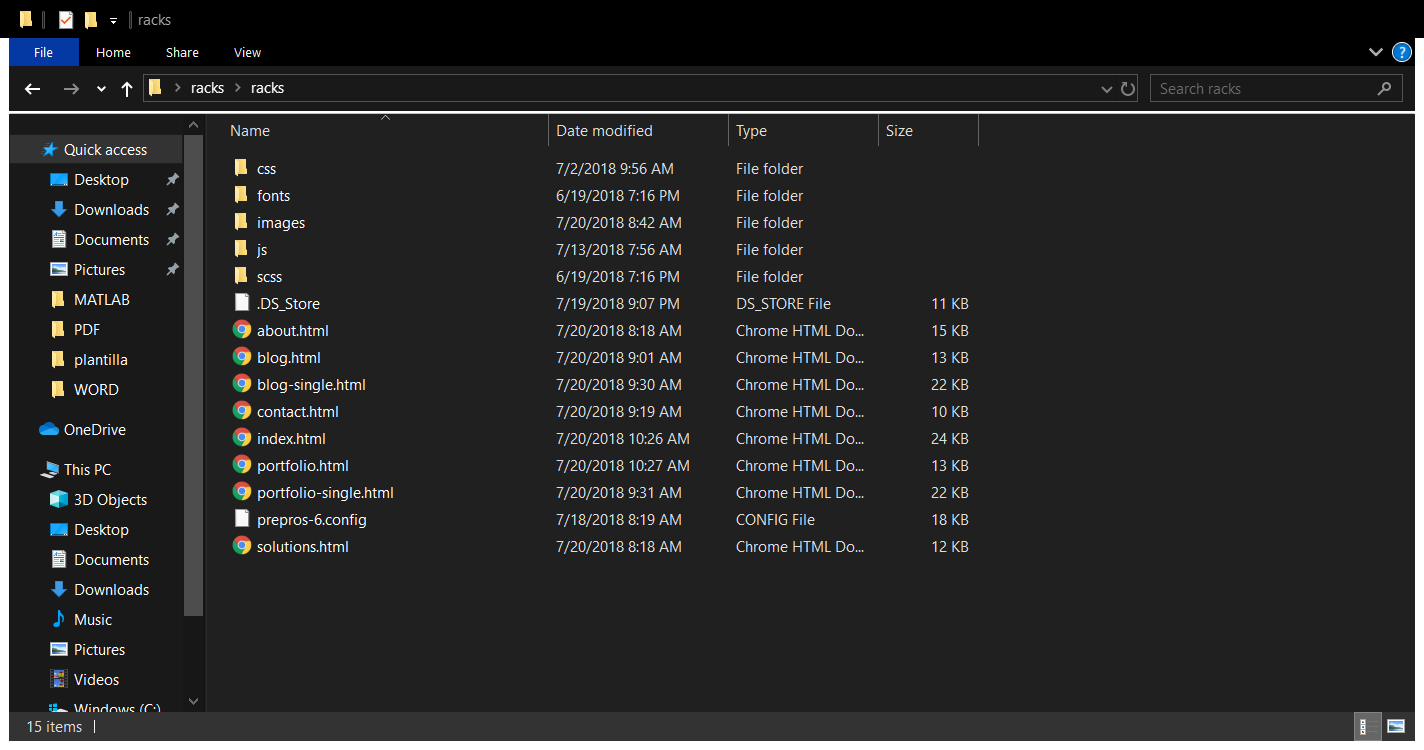
Se necesita:

* NotePad++
* Plantilla de diseño HTML de <https://mex.000webhost.com/plantillas/software/racks>

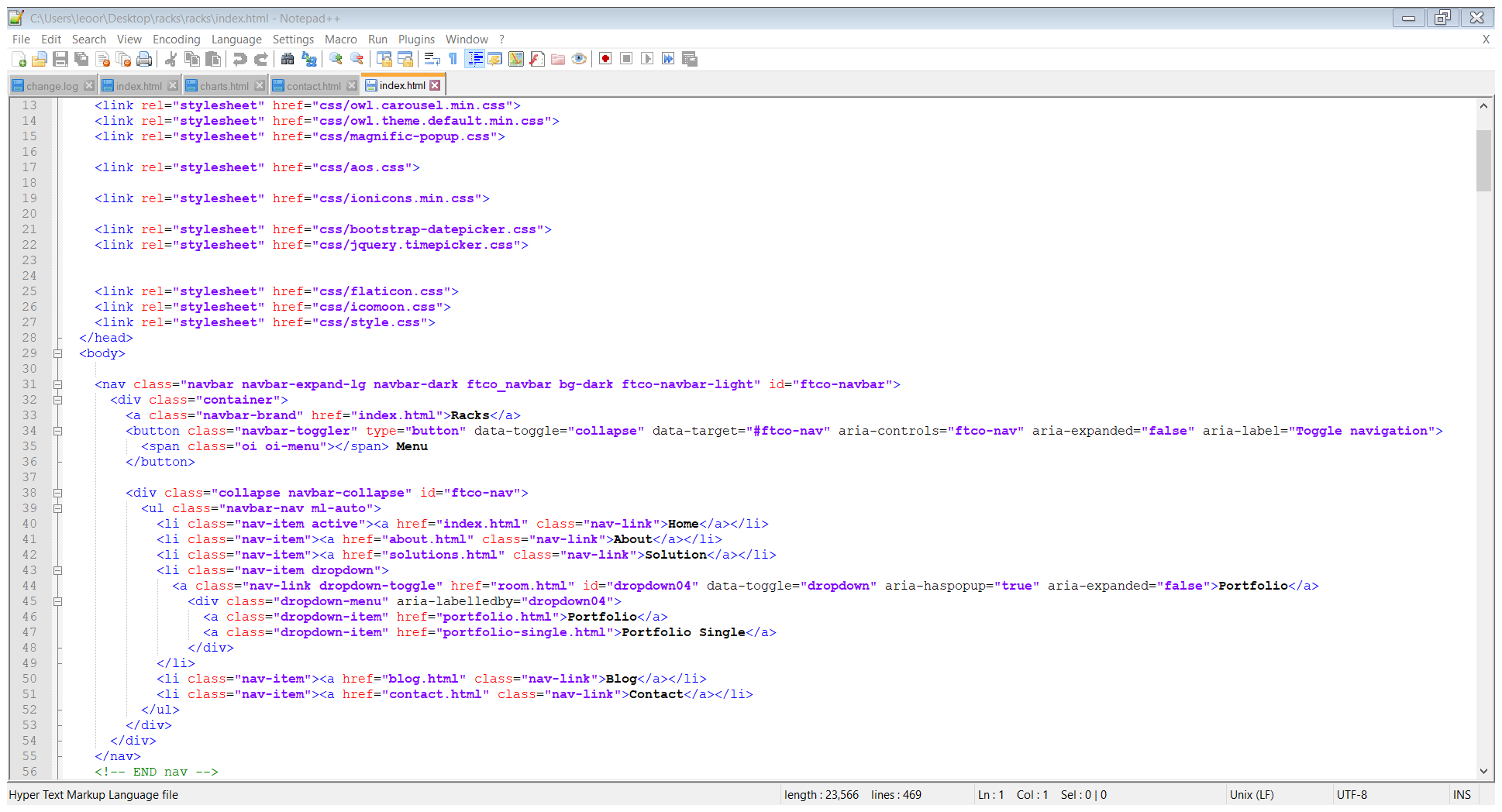
1. Descomprimimos la carpeta y eliminamos los archivos que no usaremos y la carpeta de portfolio la usaremos para charts.

A screenshot of a computer screen

Description automatically generated



1. Abrimos el archivo de index con Notepad++ y editamos los nombres de nuestro menú, eliminamos todo lo que no vayamos a usar. También cambiaremos el link del video de la página para agregar uno sobre el proyecto. Copiamos el botón que es usado para el link de video y lo usamos para crear un botón que direccione a la página de charts.



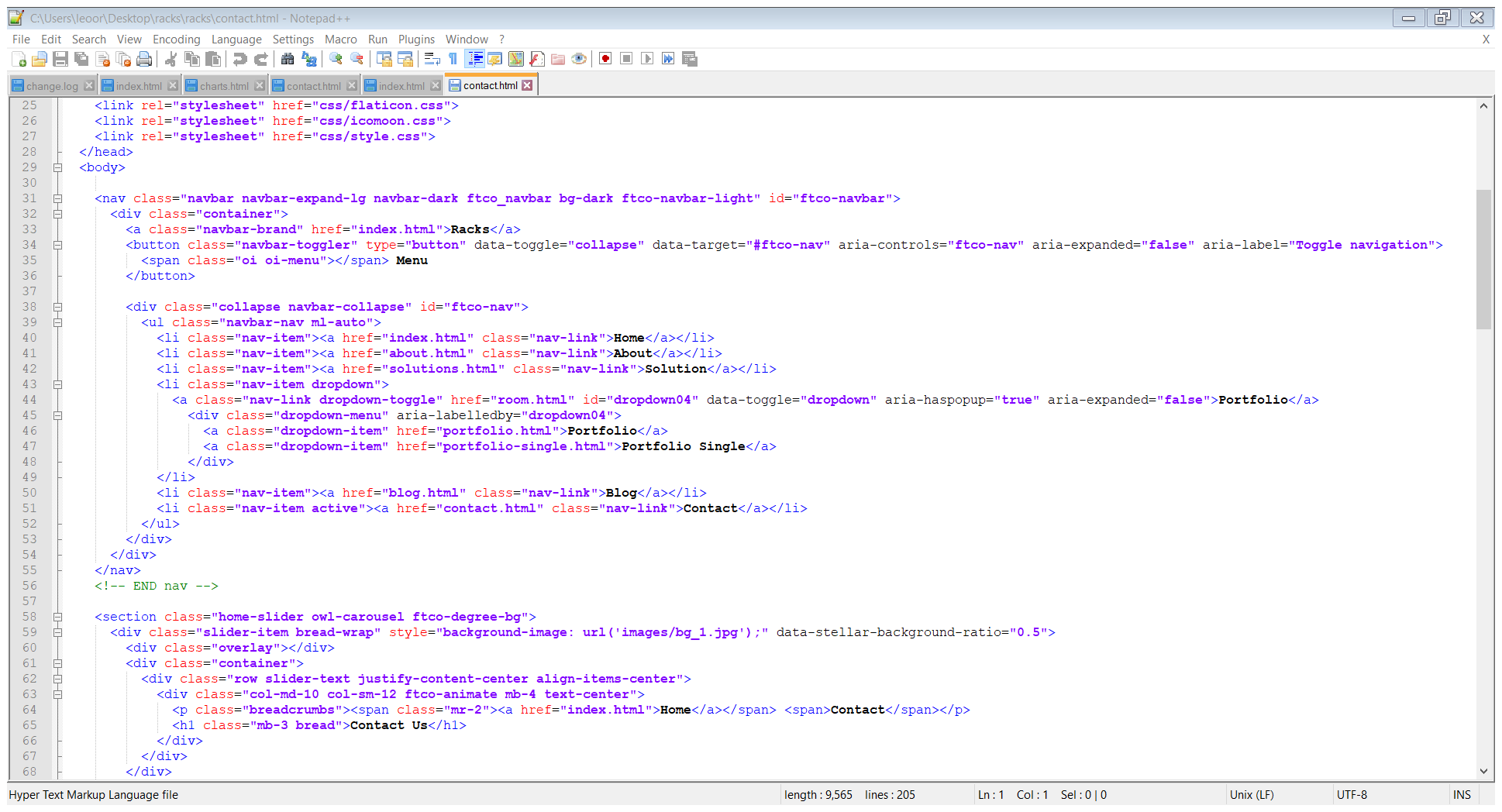
A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. Abrimos la carpeta de Contact y modificamos las mismas etiquetas que modificamos a index, eliminamos lo que no vamos a usar y agregamos la información de contacto en el contenido de la página.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. Repetimos el proceso de cambiar las etiquetas del menú así como eliminar lo que no usaremos(será necesario repetir este proceso en todas las pestañas del menú del sitio web) agregamos el diseño del blog que nos brinda la plantilla a esta pestaña para poder agregar información sobre el proyecto.

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

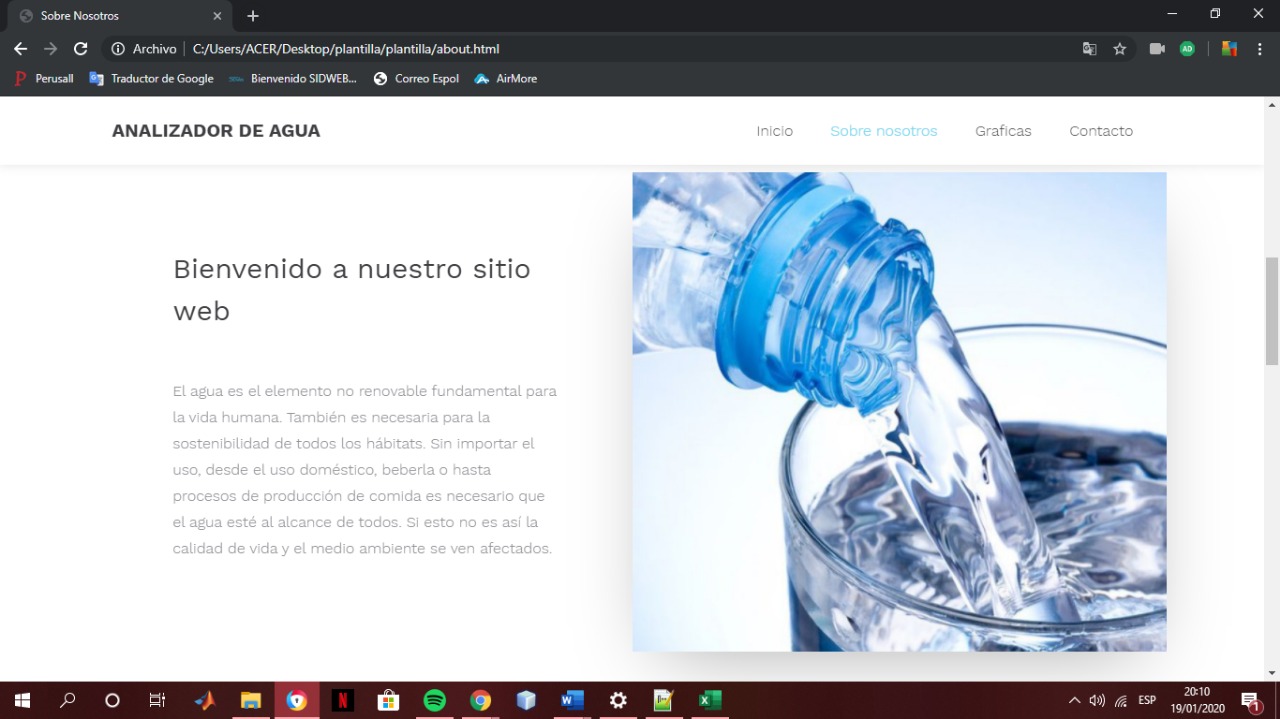
Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a cell phone

Description automatically generated

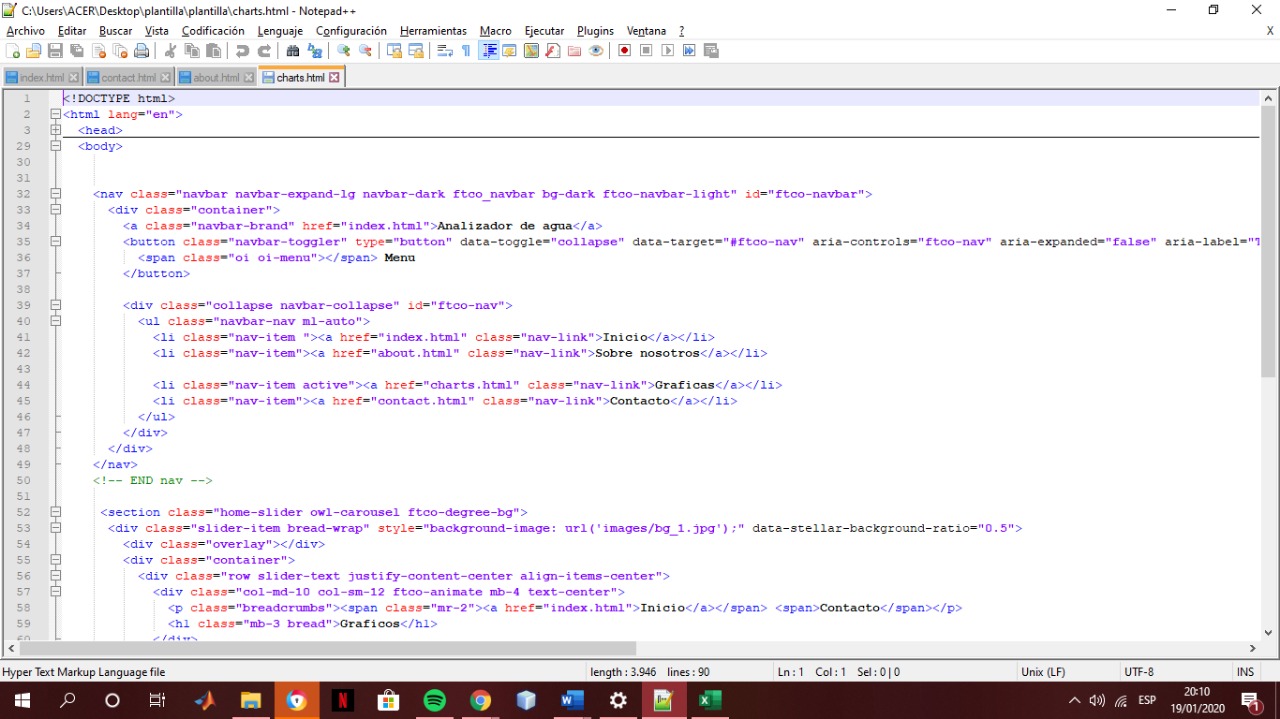
A screenshot of a cell phone

Description automatically generated

1. Eliminamos el contenido de la página de portfolio y agregamos el diseño de las gráficas y la conectamos con la información de nuestra base de datos de firebase.

A screenshot of a social media post

Description automatically generated



A screenshot of a computer

Description automatically generated

1. Empezar a tomar datos y a visualizarlos en el sitio web.
2. Diagramas del proyecto

A picture containing screenshot

Description automatically generated

Ilustración 1: Diagrama entidad-relación

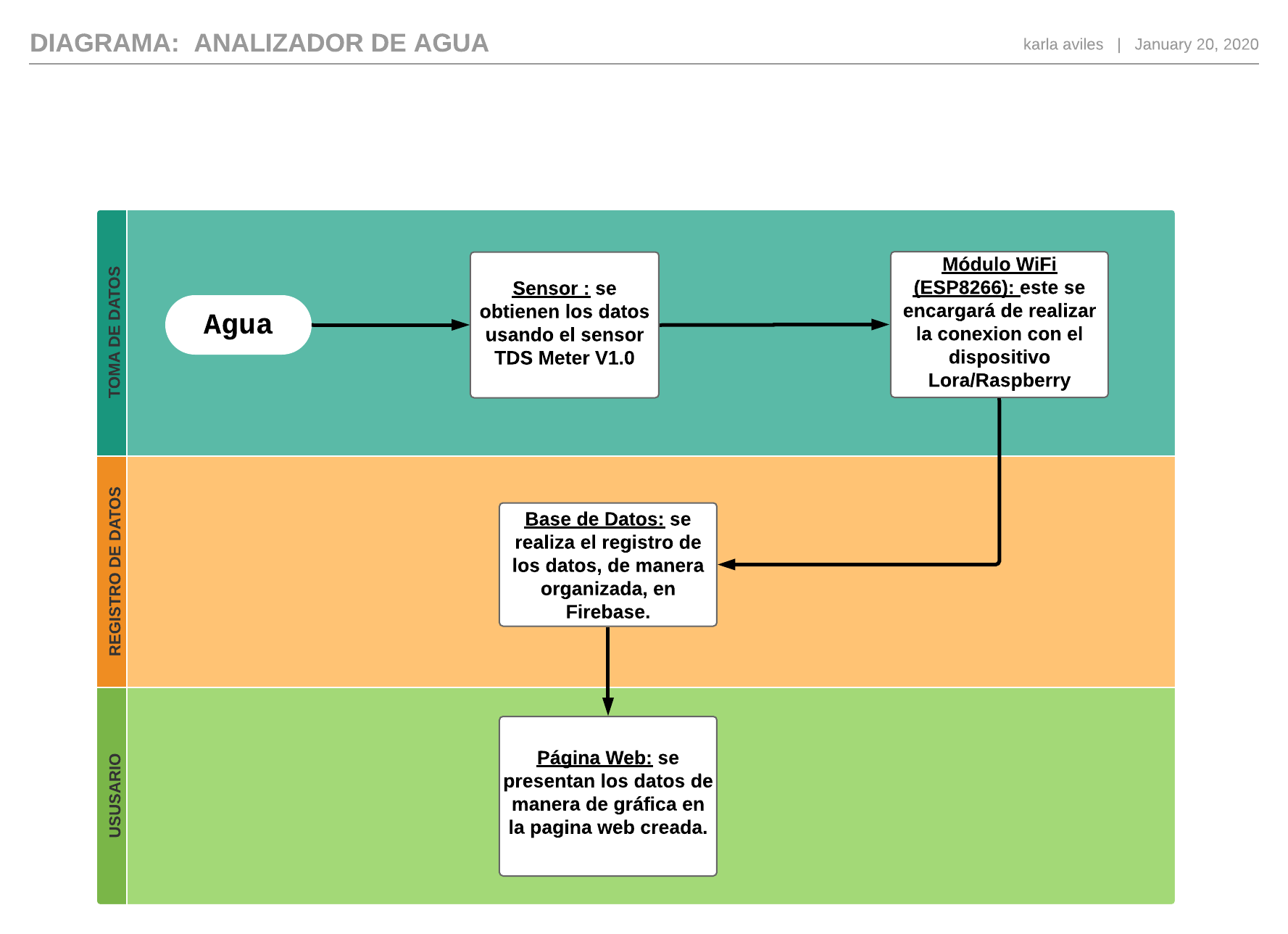


Ilustración 2: Diagrama del proyecto

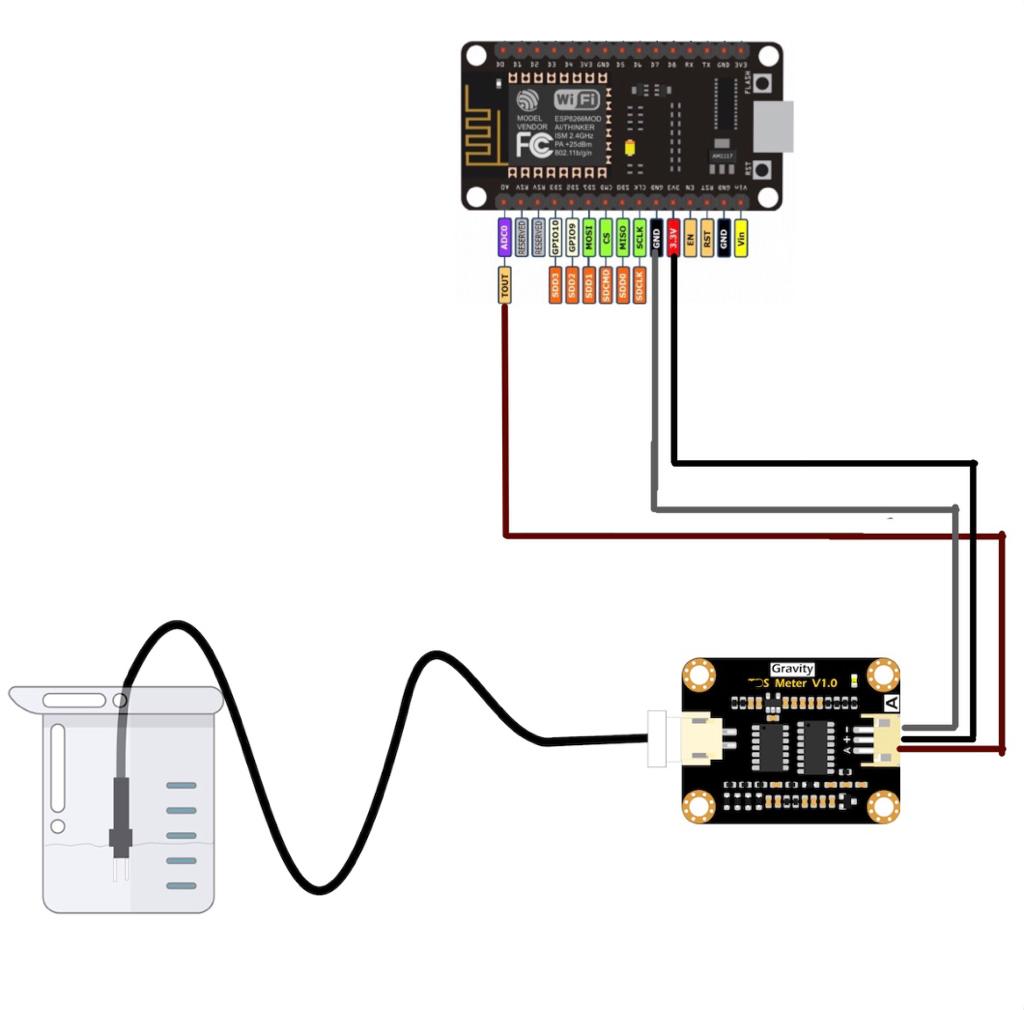


Ilustración 3: Diagrama del circuito

1. Presupuesto

* Modulo WiFi NodeMCU ESP8266 – $12
* TDS Meter V1.0 - $29

1. Conclusiones

* Se logró construir un prototipo, empleando como hardware, un módulo WiFi y un sensor TDS, realizando la conexión adecuada con una base de datos en Firebase, la cual logró recopilar los datos medidos en tiempo real.
* A través de ciertos rangos establecidos, se logró clasificar el agua, tomando en consideración los sólidos totales disueltos en la misma, determinando así la aptitud para el consumo del líquido.
* Se registraron, en una página web, los datos recolectados con un sensor TDS, estos fueron mostrados a manera de gráfica, lo cual facilitó el entendimiento de los datos al usuario, pudiendo fácilmente notar la calidad del agua en análisis.

1. Referencias bibliográficas

Arduino. (s.f.). *Arduino*. Obtenido de https://www.arduino.cc/en/Tutorial/HomePage?from=Main.Tutorials

DFRobot. (s.f.). *DF Robot*. Obtenido de https://www.dfrobot.com/product-1662.html

Google. (3 de Diciembre de 2019). *Firebase*. Obtenido de https://firebase.google.com/docs/web/setup

Oram, B. (2014). *Water Research Center*. Obtenido de https://water-research.net/index.php/water-treatment/tools/total-dissolved-solids