Reporte Técnico Reto Final

Sismógrafo IoT

Implementación de internet de las cosas
Grupo 501

Docentes:

M. Eng. Diego Hoyos Robles

M. Eng. Héctor Alberto Gutiérrez Ibarra

M. Marco Luciano Islas Olavarrieta

Gustavo Betancourt Mazomenos | A01252832

Hermosillo, Sonora a 29 de Noviembre de 2022

Problemática

El proyecto que se presenta en este reporte es la implementación de un Sismógrafo con funcionalidades de Internet de las Cosas para conectividad entre dispositivos, bases de datos, y servicios de notificación, todo esto con la función de poder monitorear y visualizar a tiempo real actividad sísmica en donde se encuentren los Sismógrafos.

Las funciones y requisitos que se tuvieron en consideración para la realización del sismógrafo fueron:

- El sismógrafo debe de mandar una señal a los otros sismógrafos al detectar un sismo que no sea un falso positivo, para que los otros dispositivos prendan su led verde por 20 segundos, los cuales pueden ser apagados antes al presionar el botón en el sismógrafo.
- Al detectar un Sismo que no sea un falso positivo, el sismógrafo prendera su led rojo por 5 segundos a partir de la última punta del sismo, y mandara una notificación al usuario con el tiempo de duración y magnitud máxima del sismo.
- Todos los sismógrafos deben de mandar sus mediciones de temperatura y humedad cada 30 segundos al servidor a través del internet.
- El servidor tiene una visualización para ver los registros de temperatura de los sismógrafos, y también los eventos sísmicos con la magnitud máxima, duración del sismo, y el identificador del sismógrafo.
- El sismógrafo tiene un potenciómetro con el cual se puede ajustar la sensibilidad del sismógrafo.

Funcionamiento del Circuito

El circuito que se armo para crear el sismógrafo tiene los siguientes componentes:

- NodeMCU Esp32
- Sensor dht11 de humedad y temperatura
- potenciómetro de 10k ohm
- Sensor MPU6050 acelerómetro y giroscopio de 3 ejes
- Botón estándar de Arduino
- Leds verde y rojo

- Resistencias de 10k ohms
- Protoboard

Las conexiones se muestran en el siguiente diagrama:

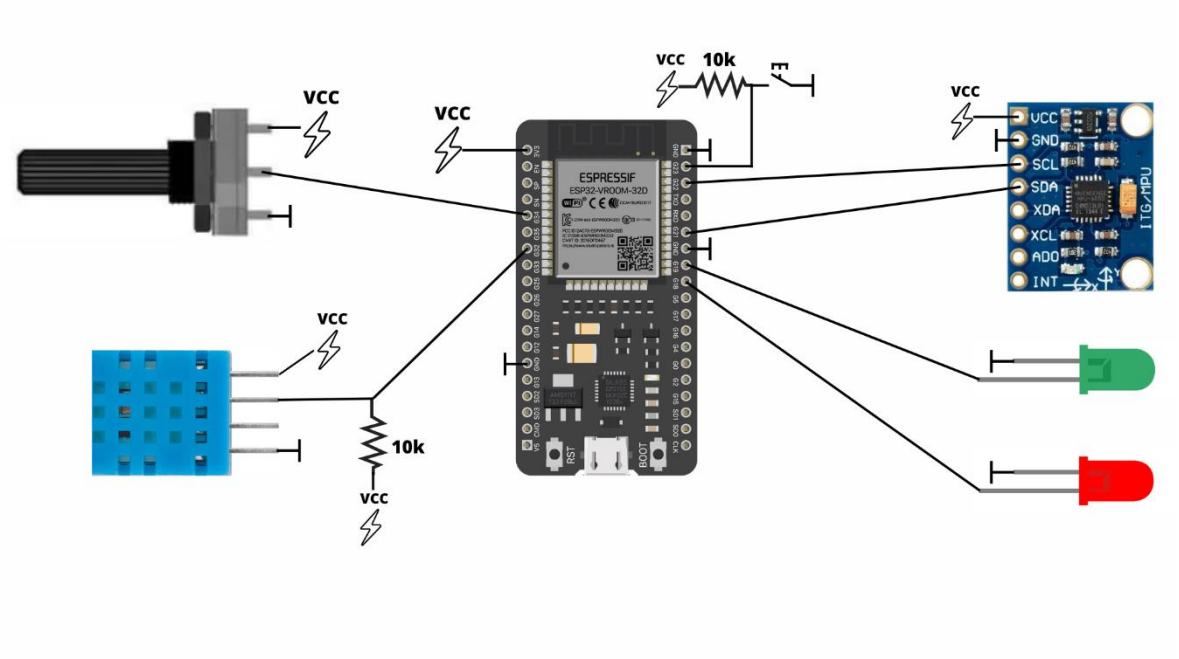


Diagrama de sismógrafo

Estos componentes son los necesarios para poder tener el sismógrafo definido en la problemática, empezando por el esp32, el cual es una placa de desarrollo en la cual se programo para controlar los demás componentes, y adicionalmente tiene el módulo de WiFi, para poder conectarse al internet y así tener el aspecto de IoT.

Después seguimos a los sensores que se utilizan para poder obtener los dato, con el principal siendo el MPU6050 a través del protocolo I2C, con el cual usamos su capacidad de acelerómetro para tener los datos en gravedades y convertirlos en magnitud y así detectar sismos, y también se tiene el dht11, con el cual obtenemos datos de las condiciones ambientales de la humedad y temperatura.

En cuanto a inputs manuales se tiene el potenciómetro, con el cual se ajusta los valores que nos da al girarlo y lo usamos para poder ajustar la sensibilidad del sismógrafo, y también se tiene el botón el cual sirve para apagar el led verde.

Finalmente se tiene los dos leds como outputs para el usuario, con el rojo diciéndonos que hubo un sismo en el sismógrafo, y el verde que hubo un sismo en otro sismógrafo.

Funcionamiento del Código

Para que este proyecto se pudiera realizar, tenemos el desarrollo divido en 3 bloques grandes, siendo estos la parte del Arduino para los sensores, conexión a internet y transmisión de datos, la de las notificaciones y comunicación entre sismógrafos con IFTTT y MQTTX, y la parte del servidor y visualización de datos con MYSQL, PHP y Grafana.

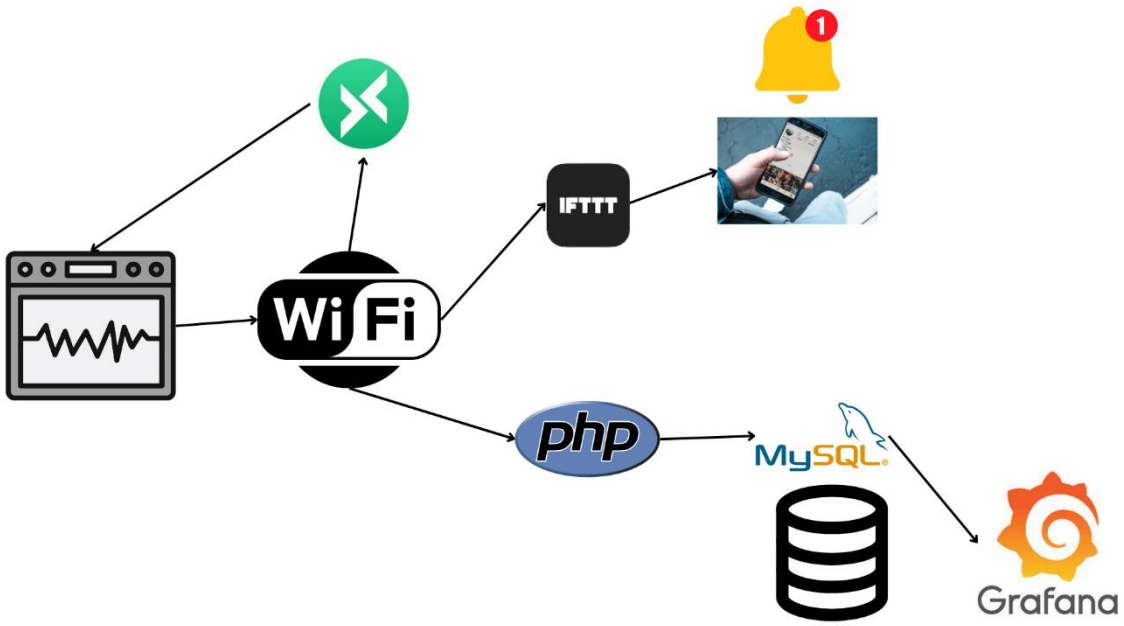


Diagrama con conexiones hechas del sismógrafo a través del internet

Arduino:

En la parte de Arduino, se implemento la lectura de los sensores de dht11 y el acelerómetro con los protocolos de comunicación I2C, el uso de Inputs con botones y el potenciómetro, los leds como outputs y el módulo de WiFi para la comunicación con internet y la funcionalidad de los componentes IoT en el proyecto.

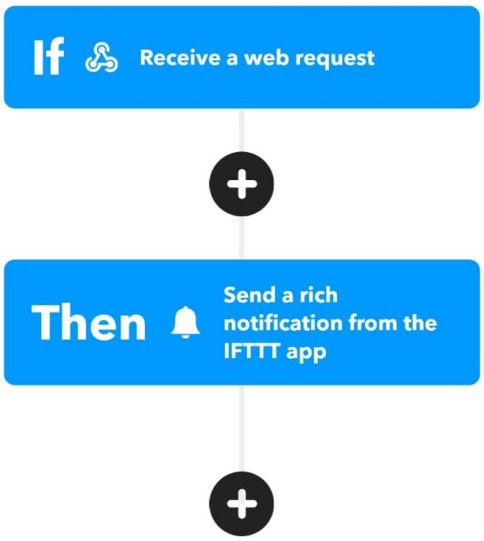
El código también tiene el uso de los tickers y técnicas para poder contar el tiempo y evitar que el código sea frenado al momento de querer tomar este a consideración para la transmisión de datos o duración de sismos.

Notificaciones y comunicación entre sismógrafos:

Para poder implementar las notificaciones al usuario, usamos la plataforma IFTTT, con la cual creamos el evento del sismo el cual se activa y manda la notificación con los datos al usar el URL con la llave proporcionada por IFTTT en el código de Arduino.

```
// Funcion que manda la notificacion a IFTTT con los datos del sismo
void sendIFTTTnotif(unsigned long TotalTime, unsigned long Max){
  HTTPClient http;
  String serverPath = "https://maker.ifttt.com/trigger/test/with/key/_ZwFpyqfgLuZ4WuCEnM2A?value1="; // URL de IFTTT pra mandar mensaje
  //  Se le agrega al path los valores que recibe y muestra la notificacion
  serverPath += TotalTime/1000.00;
  serverPath += "&value2=";
  serverPath += Max;
  http.begin(serverPath.c_str());
  http.end();
}
```

Función para mandar notificaciones por IFTTT



Aplicación de IFTTT con el proceso para mandar la aplicación

También se utilizo el protocolo MQTTX, con el cual tenemos canales a los cuales se suscriben los sismógrafos y desde los cuales reciben la señal entre ellos al detectar un sismo.

```
// Se verifica si sigue con el sismo despues de medio segundo para saber si es falso positivo
if(millis()-timer >=500 && Vsismo == true){
  TotalTime = (millis()-timerSis)-500;
  // Se detecta sismo
  if(TotalTime >= 1000){ // Sismo
    digitalWrite(RED_LED, HIGH);
    Serial.println("Sismo");
    String text = "1";
    client.publish("TC1004B/Javier/sismo",text.c_str()); // se publica a mqttx para prender los leds verdes de los otros esp´s
    client.publish("TC1004B/Alejandro/sismo",text.c_str()); // se publica a mqttx para prender los leds verdes de los otros esp´s
    sendIFTTTnotif(TotalTime, Max); // Se manda la notificacion a IFTTT
    sendSismo(TotalTime);
    led5seconds.once(5, turnoffLedR); // Se apaga el led rojo despues de 5 segundos del sismo
  }
  Vsismo = false; // Se declara el sismo como falso
 }
}
```

Aquí se muestra que al detectar el sismo este manda un 1 a los canales de los otros sismógrafos.

```
// Se establece que al recibir un 1 en su canal de mqttx, este prenda su led verde
void callback(char* topic, byte* payload, unsigned int length){
 if(strcmp(topic, "TC1004B/Gustavo/sismo") == 0){
  if(payload[0] == '1'){
   digitalWrite(GREEN_LED, HIGH);
  }
 }
}
```

Aquí se ve el código para recibir el 1 del canal de MQTTX

Servidor y visualización de datos

Para el servidor usamos una base de datos en MYSQL a través de Free MYSQL hosting, donde tenemos las tablas con los datos guardadas, y tenemos un servidor en infinityapps, el cual se vinculó con el MYSQL y tiene los archivos PHP para realizar cambios a la base de datos, y finalmente se tiene Grafana, el cual se conecta a MYSQL y nos da la visualización de los datos.

Código de PHP para establecer conexión con el servidor de MYSQL

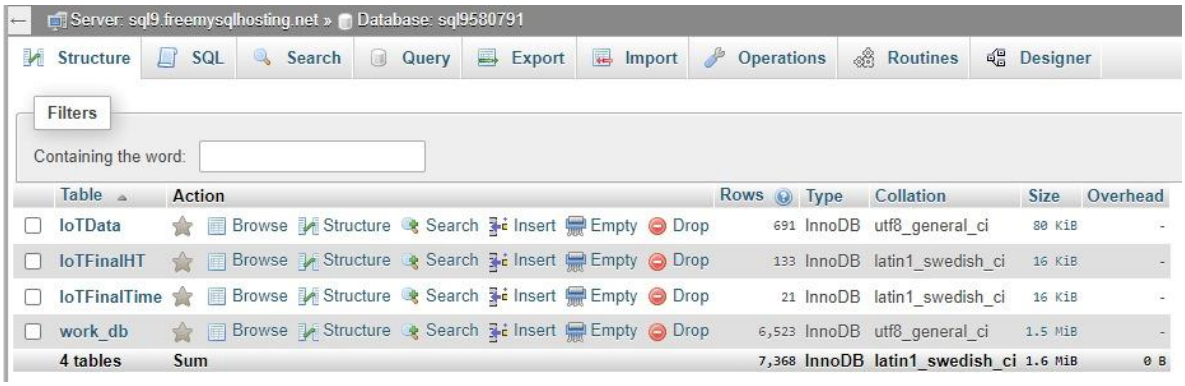
```php
    if($_SERVER['REQUEST_METHOD'] == 'POST') {
        $sql = "INSERT INTO IoTFinalHT(Modulo,temperatura,humedad) VALUES
        (:m, :t, :h)";
        $stmt = $pdo->prepare($sql);
        $stmt->bindValue(':m', $_POST['m']);
        $stmt->bindValue(':t', $_POST['t']);
        $stmt->bindValue(':h', $_POST['h']);
        $stmt->execute();
        $idPost = $pdo->lastInsertId();
        if($idPost) {
            header("HTTP/1.1 200 OK");
            echo json_encode($idPost);
            exit;
        }
    }
?>
```

Porción del archivo PHP para insertar temperatura y humedad en el servidor de MYSQL

```php
    if($_SERVER['REQUEST_METHOD'] == 'POST') {
        $sql = "INSERT INTO IoTFinalTime(Modulo, duracionSismo) VALUES
        (:m,:ma,:ds)";
        $stmt = $pdo->prepare($sql);
        $stmt->bindValue(':m', $_POST['m']);
        $stmt->bindValue(':ds', $_POST['ds']);
        $stmt->execute();
        $idPost = $pdo->lastInsertId();
        if($idPost) {
            header("HTTP/1.1 200 OK");
            echo json_encode($idPost);
            exit;
        }
    }
```

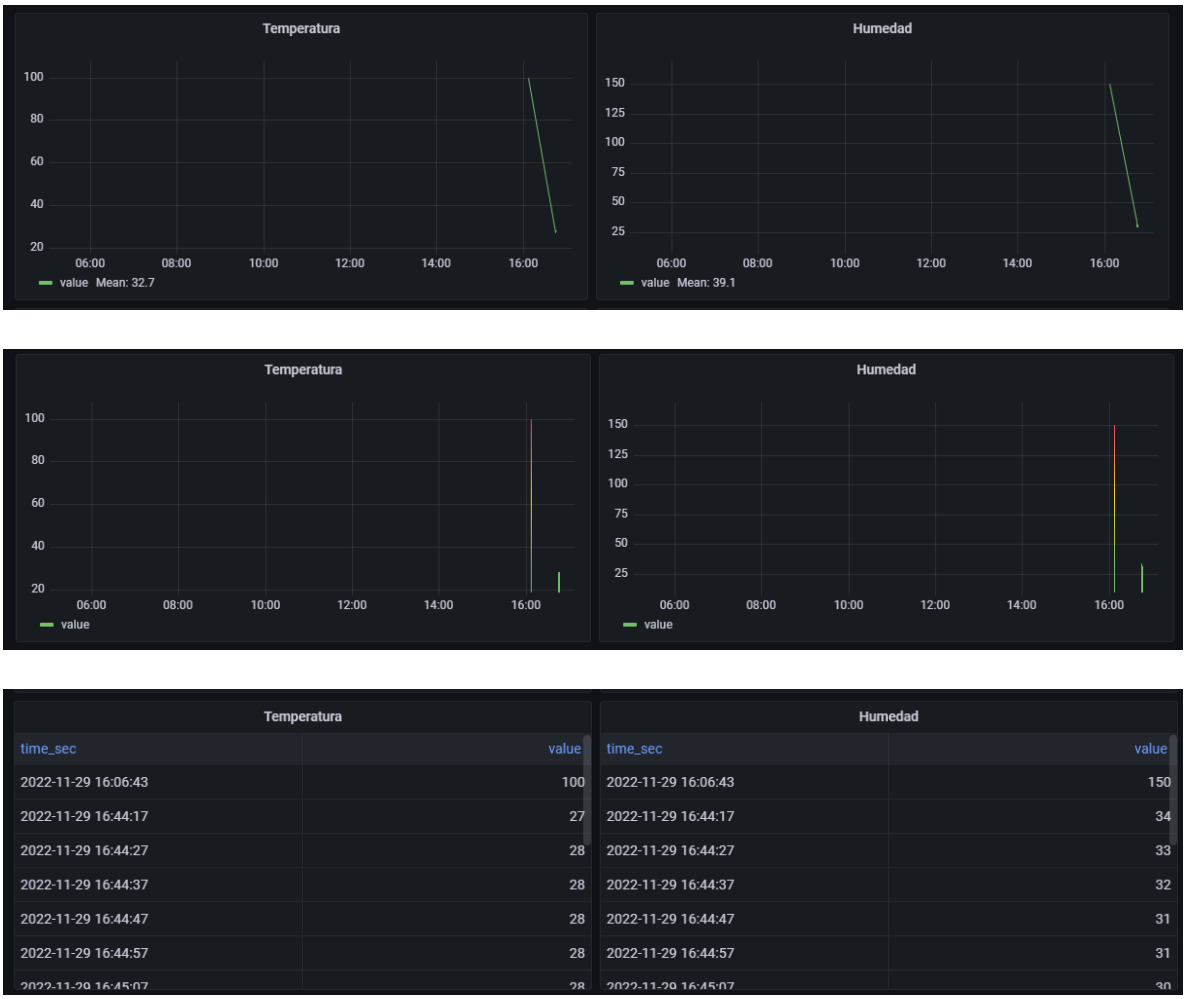
Porción del archivo PHP para insertar sismos en el servidor de MYSQL

Los archivos PHP estan hechos para que al momento de mandar un request a nuestro servidor por un URL, usando el POST recibiendo datos y e insertándolos a la base datos con queries de MYSQL.



Base de datos utilizada

Grafana se conecta a la base de datos y como el PHP, este usa queries de MYSQL para mostrar los datos que sean registrados por los sismógrafos en su base de datos, sin necesidad de que este realice cambios.



Dashboard en Grafana con la visualización de datos

Conclusión

Como conclusión a este reporte, puedo decir que este proyecto fue muy enriquecedor con los temas que tocamos, ya que vimos temas muy importantes en el aspecto de desarrollo de software y de internet de las cosas, aprendiendo sobre Arduino, conexión a internet, bases de datos a través de MYSQL, servidores y api’s, y de como poder diseñar un producto con todos estos elementos.

Este reto ha sido de mis favoritos en la carrera ya que vimos muchos temas de manera detallada y clara, con lo cual pudimos realizar el reto con todos los requisitos necesarios, dándonos un reto interesante y entretenido para poder desarrollar nuestras habilidades de diseño e implementación de software con un sistema a través de internet.