PROPUESTA DE PROYECTO CAPSTON 2021

Sistema de monitoreo para la detección de actividad sísmica y prevención de accidentes dentro de casas habitación.

Autores:

Dr. Genaro Hernández Valdez

M. en C. Mario Alberto Ramírez Reyna (Representante)

Ing. Víctor Hugo Caro Martínez

Contenido

1. Resumen

En este documento se presenta la propuesta de un proyecto tecnológico para el apoyo en el monitoreo de la salud estructural y prevención de daños colaterales de casas habitación que experimenten movimientos y vibraciones debidos a actividades sísmicas. El sistema de monitoreo está compuesto por una red inalámbrica de nodos que recolectan información relacionada a la actividad sísmica y la transmiten a un centro de control (servidor). Los nodos de la red inalámbrica se componen de sensores y actuadores que permiten detectar movimientos y vibraciones de la estructura, fugas de gas, aumento de temperatura y humedad de la casa habitación. Con los datos recolectados se obtienen métricas que permiten apoyar en la evaluación de la salud estructural de la vivienda. Además, dependiendo del grado de las vibraciones y movimiento de la estructura monitoreada así como el nivel de gas, temperatura y humedad, el centro de control puede ordenar a los nodos que activen a los actuadores para cerrar válvulas de gas y agua con la finalidad de prevenir accidentes colaterales dentro de la casa habitación. El desarrollo de este proyecto tiene como finalidad apoyar en el mejoramiento de la seguridad en viviendas que se encuentren en zonas sísmicas de riesgo medio o alto; de esta manera, se contribuye en el Objetivo Mundial 11 de las Naciones Unidas (Ciudades y Comunidades Sostenibles) [1].

2. Introducción

Las redes inalámbricas para aplicaciones del Internet de las Cosas (IoT) se componen de nodos que integran a un número grande de sensores y actuadores de pequeñas dimensiones físicas. Dichos nodos tienen capacidades de cómputo limitada y se pueden intercomunicar entre ellos o con un centro de control a través de una red inalámbrica [2]. La programación y electrónica de los nodos, sensores, actuadores y del centro de control se debe configurar de forma adecuada para trabajar de manera efectiva y para adaptarse dinámicamente para cumplir con las tareas requeridas por una determinada aplicación. En este proyecto se pretende implementar una red inalámbrica de sensores que apoye tanto en la evaluación del posible daño estructural como en la prevención de accidentes en casas habitación (o departamentos) debido a movimientos telúricos

(sismos). Este es un tópico importante en nuestro país ya que es bien conocido, que nos encontramos dentro de una zona conocida como el “cinturón de fuego” (en la que también se encuentran Japón y Chile, entre otros países) caracterizada por la alta actividad sísmica [3]. El sistema que se pretende desarrollar se enfoca en recolectar información relevante correspondiente a una actividad sísmica con el propósito de transmitirla a un centro de control, el cual la procesa para realizar acciones al respecto. Por tratarse de una aplicación en construcciones reales, es importante colocar los sensores y actuadores en lugares estratégicos dentro de la vivienda. En específico, se recomienda colocar los giroscopios y acelerómetros en los extremos de la vivienda para medir los desplazamientos verticales, mientras que los detectores de gas, temperatura y humedad se recomienda colocarlos en la cocina y baños de la vivienda [2].

3. Objetivo General

Diseñar e implementar el prototipo de una red inalámbrica de sensores y actuadores para la detección de actividad sísmica que permita el apoyo en la evaluación de la salud estructural y en la prevención de accidentes dentro de los hogares ubicados en zonas sísmicas.

4. Objetivos Particulares

1. Construir un bloque de adquisición de información para medir la salida de un acelerómetro y sensor de vibración para la detección de actividad sísmica.
2. Construir un bloque de adquisición de información para medir la salida de un sensor de gas LP y de temperatura y humedad para la detección de incendios.
3. Diseñar e implementar un bloque de conexión inalámbrica local para la comunicación de los nodos de la red con un centro de control.
4. Construir un bloque de activación para los actuadores que permitan controlar el suministro de agua y gas.
5. Construir un bloque para la transmisión de información por intercambio de mensajes entre los diferentes nodos con un servidor central.
6. Diseñar e implementar un servidor donde se concentre la información adquirida por los nodos de la red.

7. Diseñar e implementar una interfaz gráfica donde se presente de manera clara la información generada por los sensores.

5. Justificación

Para fines de diseño antisísmico, el territorio mexicano está dividido en 4 zonas sísmicas (ver figura 1), cada una con sus respectivas peculiaridades [4]:

- La zona A (Riesgo Bajo), no se han reportado sismos durante los últimos 80 años y no se esperan aceleraciones del suelo mayores al 10% de la aceleración de la gravedad a causa de temblores.
- La zona B y C (Riesgo Medio), no se presentan sismos frecuentemente y se esperan aceleraciones del suelo a causa de temblores de alrededor del 70% de la aceleración gravedad.
- La zona D (Riesgo Alto), se reportan sismos muy frecuentes y la aceleración del suelo a causa de sismos suele sobrepasar el 70% de la aceleración gravedad.

Figura 1. Mapa de Vulnerabilidad ante sismos en México, extraído de [4].

Para el contexto en el que trabajaremos, se pretende que este sistema funcione en edificaciones que se encuentran en la zona del Valle de México (aunque su uso no se limita a esta zona ya que puede adecuarse para que opere en cualquier ciudad del

mundo). Es importante destacar que a pesar de que la Ciudad de México (CDMX) se encuentra en la zona B las condiciones del subsuelo del Valle de México (ver figura 2) hacen que la CDMX sea catalogada como una zona de alta actividad sísmica [5].

Figura 2. Mapa de Tipos de Suelo en la CDMX, extraído de [5].

Históricamente, desde 1957 hasta 2020 se han suscitado 8 eventos sísmicos que han dejado a la CDMX con grandes daños [6], cada uno dejando un panorama desolador en la vida de los ciudadanos de la CDMX.

Es importante mencionar que los daños que sufren los habitantes de la CDMX a sus viviendas no solamente son daños estructurales que son causados por los movimientos agresivos producidos durante el sismo, sino que también llegan a suceder daños secundarios. De acuerdo con [7] durante el sismo del 23 de junio del 2020 el C5 tuvo reporte de 8 caídas de postes de energía eléctrica, cuatro figas de gas en departamentos, 3 incendios y caídas de cables con corriente eléctrica.

Por tal motivo, el sistema de monitoreo propuesto en este trabajo tiene como finalidad apoyar al usuario en garantizar la seguridad de su vivienda permitiendo cortar el flujo de los servicios que comúnmente podrían causar algún accidente derivado de la actividad sísmica, generalmente incendios por corto circuito o fugas de gas; o inundaciones por fugas de agua internas dentro del departamento.

Por otro lado, este sistema también permitirá tomar información estadística de la actividad sísmica y como ha afectado al inmueble, permitiendo monitorear la intensidad con la que las paredes o soportes experimentan los movimientos y en caso de tener una actividad fuerte, permita al usuario poder realizar un reconocimiento remoto a través de imagen para garantizar que la entrada al inmueble sea segura.

En la sección de metodología se describirá a detalle el funcionamiento de dicho sistema.

6. Metodología

El diagrama a bloques del sistema de monitoreo para la detección de movimiento y vibraciones, reportar posibles daños estructurales y prevenir accidentes en casas habitación durante actividades sísmicas se muestra en la figura 3.

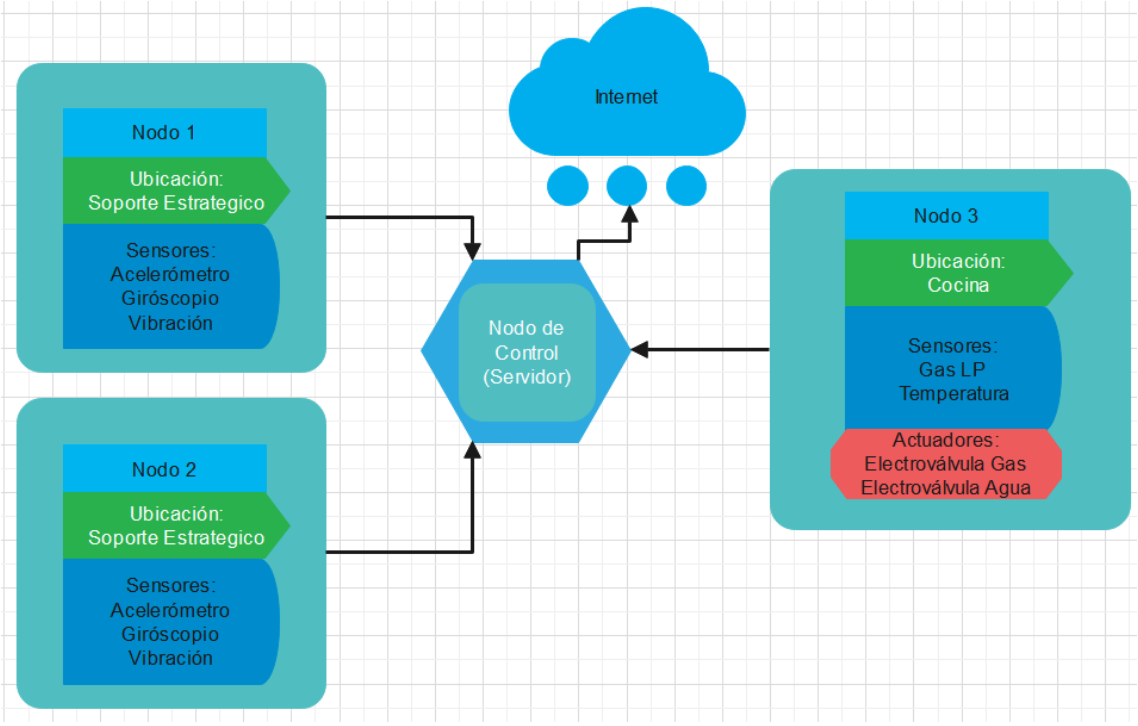

Figura 3. Topología del sistema de monitoreo, creación propia.

Los **nodos N1 y N2** estarán integrados por un acelerómetro con giroscopio y un sensor de vibración. Se ubicarán en los extremos de la vivienda para monitorear los movimientos y vibraciones anómalos sucedidos durante y después del sismo. La figura 4 muestra un diagrama de la tarjeta de desarrollo ESP32CAM con la integración de los sensores de monitoreo.

El **nodo N3** se ubicará en la cocina de la vivienda y se le integrarán los siguientes elementos: un sensor de gas LP, un sensor de temperatura y humedad, además de dos actuadores, una electroválvula para agua y una electroválvula para gas. La figura 5 muestra un diagrama de la tarjeta de desarrollo ESP32CAM con la integración de los actuadores y los sensores de temperatura/humedad y gas LP.

Los **nodos N1, N2** y **N3** del sistema de monitoreo que se muestra en la figura 3, están representados físicamente por tarjetas de desarrollo ESP32CAM, las cuales se pueden programar para conectarse a una red inalámbrica y comunicarse con el nodo de control (servidor).

El **nodo de control** que se muestra en la figura 3, corresponde a un nodo con capacidad de conexión a Internet (Raspberry Pi 4) utilizado para procesar las lecturas de las magnitudes físicas recolectadas por los sensores. La comunicación entre el servidor y los nodos se realizará a través de un agente mediante la suscripción y publicación en diferentes tópicos (por ejemplo, se puede utilizar el agente público MQTT). Además, este sensor permitirá brindar alertas luminosas (a través del led flash de la ESPC32CAM) en caso de que esté sucediendo algún sismo o exista posibilidad de algún otro accidente derivado a causa de este (incendio).

La comunicación entre los nodos y el servidor es bidireccional. Cuando alguno (o ambos) nodos N1 o N2 detecten movimiento y vibración estructural arriba de cierto umbral, o que el nodo *N*3 detecte presencia de gas LP o temperatura arriba de cierto umbral, se lo informa al servidor. El servidor analizará la información y tomará decisiones al respecto; por ejemplo, si el sismo tiene una magnitud mayor a 5 grados en la escala de Richter, instruye a los nodos para que activen su cámara e instruye al nodo N3 para que active los servomotores (para cerrar las válvulas de gas y agua y prevenir posibles accidentes). El servidor procesará la información y permitirá que sea visualizada a través de algún otro dispositivo móvil conectado a la misma red o a la Internet, de esta forma se podrán visualizar las gráficas de las magnitudes físicas que miden los sensores y fotografías del interior de la vivienda, todo esto como apoyo para evaluar el posible daño estructural causado por la actividad sísmica y decidir si es seguro regresar a la vivienda y reactivar los servicios de gas y agua, o si es necesario solicitar a personal especializado una evaluación más detallada y profesional de la salud estructural de la vivienda.

La programación de los diferentes sensores y la comunicación entre el centro de control y los diferentes nodos se realizarán con ayuda de lenguajes de alto nivel tales como Python y C++. Para la comunicación entre el centro de control y los nodos se utilizará un agente MQTT y las gráficas y fotografías en la Internet se presentarán con ayuda de la aplicación NODE-RED y GRAFANA.

7. Material

La lista de material requerido para realizar el presente proyecto se lista a continuación:

- 1 computadora de bajo costo (Raspberry Pi 4)
- 3 tarjetas de desarrollo ESP32CAM
- 2 acelerómetros con giroscopio, resolución de 16 bits (MPU6058).
- 2 sensores de vibración (SW-420).
- 1 sensor de gas LP (MQ6)
- 1 sensor de temperatura y humedad (DHT11)
- 3 conectores de carga USB
- 1 servomotor para válvula de Gas (SG90)
- 1 servomotor para válvula de Agua (SG90)
- 1 Fuente externa de 12V
- 3 Fuentes externas de 5V

8. Elementos del Sistema

En esta sección se describen cada uno de los elementos que conforman el sistema, se detallan los circuitos, el material y los programas que describen su funcionamiento.

8.1. Nodos N1 y N2.

La labor de los Nodos N1 y N2 es realizar el sensado de las alteraciones sísmicas, dígase vibraciones, cambios en la aceleración de las paredes del edificio, cambios en las inclinaciones de las paredes o posiciones. Esto se realiza a través de los módulos SW-420 y GY-521.

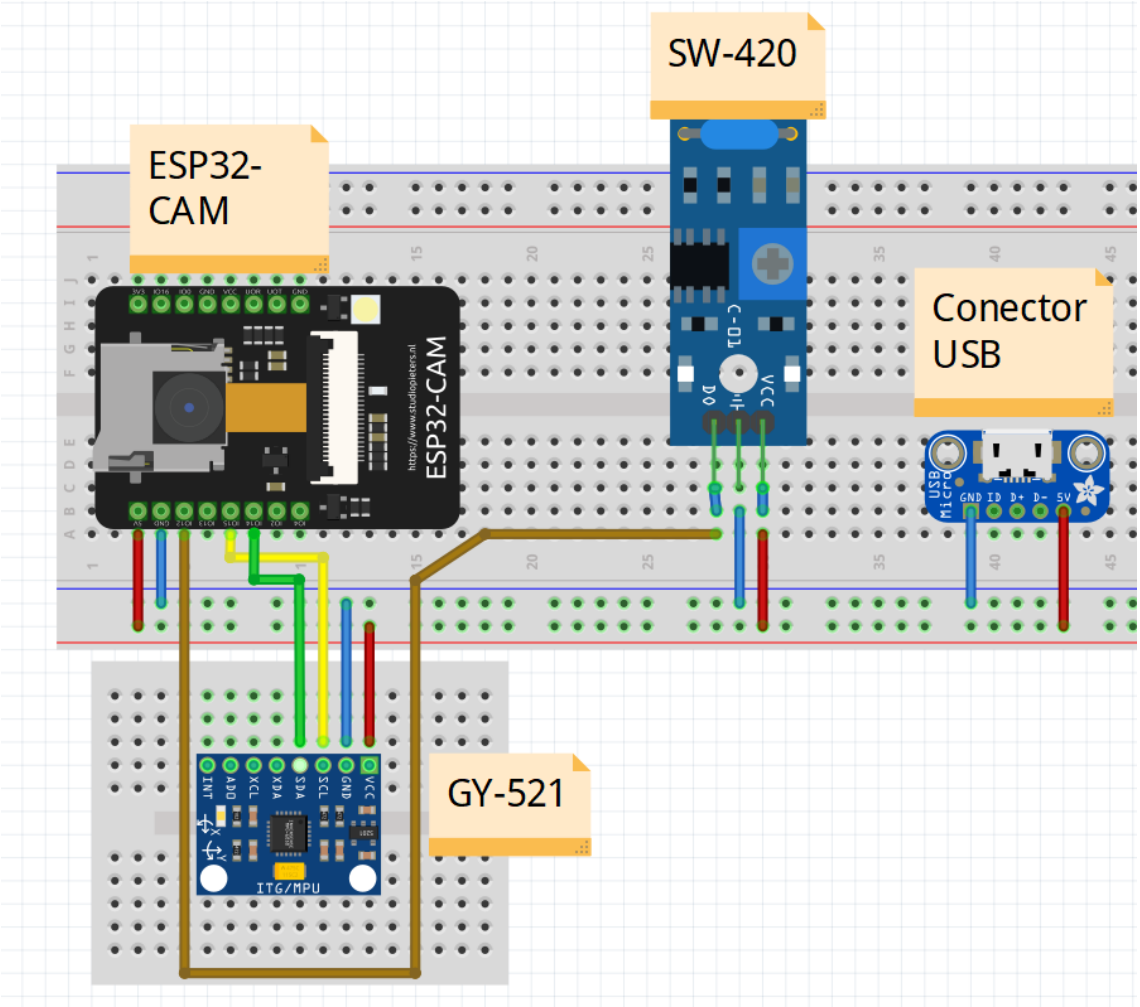

Figura 4. Diagrama del circuito de integración de la ESP32-CAM con los sensores de monitoreo (Nodos N1 y N2), creación propia.

8.1.1. Descripción del Circuito.

En la Figura 4 se aprecia el diagrama de conexiones de los Nodos N1 y N2. Estos consisten en los siguientes elementos:

- 1 SW-420
- 1 GY-521
- 1 conector de Carga USB
- 1 ESP32CAM

Cada uno de los módulos se alimenta a 5 Volts a través del conector USB. El SW-420 se conecta directamente a la GPIO12 mientras que el GY-521 se conecta a través de los puertos GPIO15 para el SCL y GPIO14 para el SDA para establecer la comunicación I2C, además cuentan con capacidad de conexión WiFi.

8.1.2. Descripción del Funcionamiento del Nodo.

Los nodos N1 y N2 son un sistema que permita la captura de la vibración, aceleración, velocidad, rotación e inclinación de las paredes sobre las cuales se coloquen. Inicialmente el sistema cuenta con una velocidad de muestreo de 60 segundos. Al detectar una vibración a través del SW-420 este disparará un proceso de interrupción que modificará la velocidad de muestreo a 6 segundos entre muestras, de esta manera al detectar un evento sísmico se tendrá una mejor resolución de la información a capturar.

Por otro lado, utilizando el módulo GY-521 utilizando el giroscopio y acelerómetro se obtiene la información de la aceleración y velocidad experimentada por la pared o trabe, además de también capturar la información de inclinación y rotación en caso de que hayan sucedido.

Toda la información capturada se envía de manera periódica (de acuerdo con la velocidad de muestreo) hacía el Nodo de Control (Servidor) utilizando el protocolo de comunicación MQTT, donde será procesada y se tomarán las decisiones pertinentes acerca de que si se deben cerrar o no las llaves de paso.

8.1.3. Código Fuente

A continuación, se presenta el código fuente que establece el funcionamiento de estos Nodos.

```
#include <I2Cdev.h>
#include <MPU6050.h>
#include <Wire.h>
#include <EspMQTTClient.h>

// inicialización de puertos
#define SCL 15
#define SDA 14
#define SIGNAL_PIN 12
#define led_FLASH 4

EspMQTTClient client(
  "P_Capstone", // nombre de la red
  "PCapstone1", // contraseña
  "192.168.0.102",  // dirección del broker
  "",   // omitir
  "",   // omitir
  "Nodos N1_N2" // nombre del cliente
);

//Inicialización de variables
const int mpuAddress = 0x68;  //Puede ser 0x68 o 0x69
MPU6050 mpu(mpuAddress);

int16_t ax, ay, az;
int16_t gx, gy, gz;
int vibra = 0;
long tiempo_prev, dt;
float girosc_ang_x, girosc_ang_y;
float girosc_ang_x_prev, girosc_ang_y_prev;
```

```
float accel_ang_x, accel_ang_y;

float acc_x, acc_y, acc_z;
float vel_x, vel_y, vel_z;
float rot_x, rot_y;
float inc_x, inc_y;

char dataString[8];
String salida;
int periodo = 60000;
unsigned long TiempoAhora = 0;

const float accScale = 2.0 * 9.81 / 32768.0;
const float gyroScale = 250.0 / 32768.0;

void printTab()
{
  Serial.print(F("\t"));
}

void act_info()
{

  // Se calculan métricas
  dt = millis() - tiempo_prev;
  tiempo_prev = millis();

  girosc_ang_x = (gx / 131) * dt / 1000.0 + girosc_ang_x_prev;
  girosc_ang_y = (gy / 131) * dt / 1000.0 + girosc_ang_y_prev;
  girosc_ang_x_prev = girosc_ang_x;
  girosc_ang_y_prev = girosc_ang_y;
  accel_ang_x = atan(ax / sqrt(pow(ay, 2) + pow(az, 2))) * (180.0 /
3.14);
  accel_ang_y = atan(ay / sqrt(pow(ax, 2) + pow(az, 2))) * (180.0 /
3.14);

  rot_x = girosc_ang_x;
  rot_y = girosc_ang_y;
  inc_x = accel_ang_x;
  inc_y = accel_ang_y;
  acc_x = ax * accScale;
  acc_y = ay * accScale;
  acc_z = az * accScale;
  vel_x = gx * gyroScale;
  vel_y = gy * gyroScale;
  vel_z = gz * gyroScale;

  if (!digitalRead(SIGNAL_PIN))
    vibra = 1;
  else
    vibra = 0;

  salida = String("{\"vibra\": \"" + String(vibra) +
                  "\", \"ax\": \"" + String(acc_x) + "\", \"ay\": \"" +
String(acc_y) + "\", \"az\": \"" + String(acc_z) +
                  "\", \"gx\": \"" + String(vel_x) + "\", \"gy\": \"" +
String(vel_y) + "\", \"gz\": \"" + String(vel_z) +
                  "\", \"rx\": \"" + String(rot_x) + "\", \"ry\": \"" +
String(rot_y) +
                  "\", \"ix\": \"" + String(inc_x) + "\", \"iy\": \"" +
String(inc_y) + "\"}");
```

```
  Serial.println(salida);
  client.publish("capstone/nodo_mpu6050", salida);
}

void onConnectionEstablished() {

  client.subscribe("capstone/nodo_actuadores", [] (const String &
payload) {
    Serial.println(payload);
  });

  client.publish("capstone/nodo_mpu6050", "Conectado");
}

void onMessageReceived(const String& topic, const String& message) {
  Serial.println(topic + ": " + message);
}

//diseño de interrupción para cambio de velocidad de muestreo
void IRAM_ATTR ISR() {
  if (!digitalRead(SIGNAL_PIN))
  {
    periodo = 60000;
    digitalWrite(led_FLASH, LOW);
  }
  else
  {
    periodo = 6000;
    digitalWrite(led_FLASH, HIGH);
  }
}

void setup()
{
  pinMode(led_FLASH, OUTPUT);
  attachInterrupt(SIGNAL_PIN, ISR, CHANGE);
  Serial.begin(115200);
  Serial.println("init serial");
  delay(5000);
  Wire.begin(SDA, SCL);
  Serial.println("init wire");
  delay(5000);
  mpu.initialize();
  Serial.println(mpu.testConnection() ? F("IMU iniciado correctamente")
: F("Error al iniciar IMU"));
  delay(5000);

}
void loop()
{
  client.loop();
  if (millis() > TiempoAhora + periodo)
  {
    TiempoAhora = millis();
    mpu.getAcceleration(&ax, &ay, &az);
    mpu.getRotation(&gx, &gy, &gz);
    act_info();
  }
}
```

8.2. Nodo N3

La labor del Nodo N3 es realizar el monitoreo de los niveles de gas LP, temperatura y humedad de la vivienda. Estos sensores se colocarán en la cocina de la vivienda. Esto se realiza a través de los módulos MQ6 y DHT11. El Nodo N3, también incluye el control de los servomotores que permiten la apertura y cierre de los suministros de gas y agua de la vivienda.

8.2.1. Descripción del Circuito.

En la Figura 5 se aprecia el diagrama de conexiones del Nodo N3. Estos consisten en los siguientes elementos:

- 1 sensor de gas LP MQ6
- 1 sensor de temperatura y humedad DHT11
- 2 servomotores (SG90)
- 1 ESP32CAM

Los sensores, los servomotores y la tarjeta ESPC32CAM se alimentan con 5 Volts. La salida digital (DO) del sensor de gas LP MQ6 se conecta directamente a la GPIO12, la salida digital del sensor DHT11 se conecta directamente al puerto GPIO14, mientras que los servomotores para controlar las válvulas de gas y agua se conectan a los puertos GPIO13 y GPIO15, respectivamente. Además, la ESP32CAM cuenta con capacidad de conexión WiFi.

8.2.2. Descripción del Funcionamiento del Nodo.

El Nodo N3 es un sistema que permita la captura del nivel de gas LP, la temperatura y humedad del lugar donde se coloquen (normalmente en la cocina de la vivienda). Inicialmente el sistema cuenta con una velocidad de muestreo de 60 segundos. Cuando el sensor MQ6 detecta la presencia de gas LP o el sensor DHT11 detecta una temperatura mayor a un cierto umbral (por ejemplo 50 grados Celsius) el programa disparará un proceso de interrupción que modificará la velocidad de muestreo a 6 segundos entre muestras, de esta manera al detectar un evento anómalo se tendrá una mejor resolución de la información a capturar. La válvula de gas se cierra si se detecta presencia de gas LP o si la temperatura es mayor a un cierto umbral (por ejemplo, 50°C). La válvula de agua se cierra si se detecta un nivel de humedad mayor a un cierto umbral (por ejemplo, 50%). Las válvulas de gas y agua también se pueden controlar de forma remota por el servidor. Para esto, el sistema se subscribe a un agente (broker) público en los temas "capstone/actuador_gas" y "capstone/actuador_agua". Si se recibe el mensaje "true" la válvula se abre, si se recibe el mensaje "false", la válvula se cierra.

Toda la información capturada se envía de manera periódica (de acuerdo con la velocidad de muestreo) hacía el Nodo de Control (Servidor) utilizando el protocolo de comunicación MQTT, donde será procesada y se tomarán las decisiones pertinentes acerca de que si se deben volver a abrir no las llaves de paso.

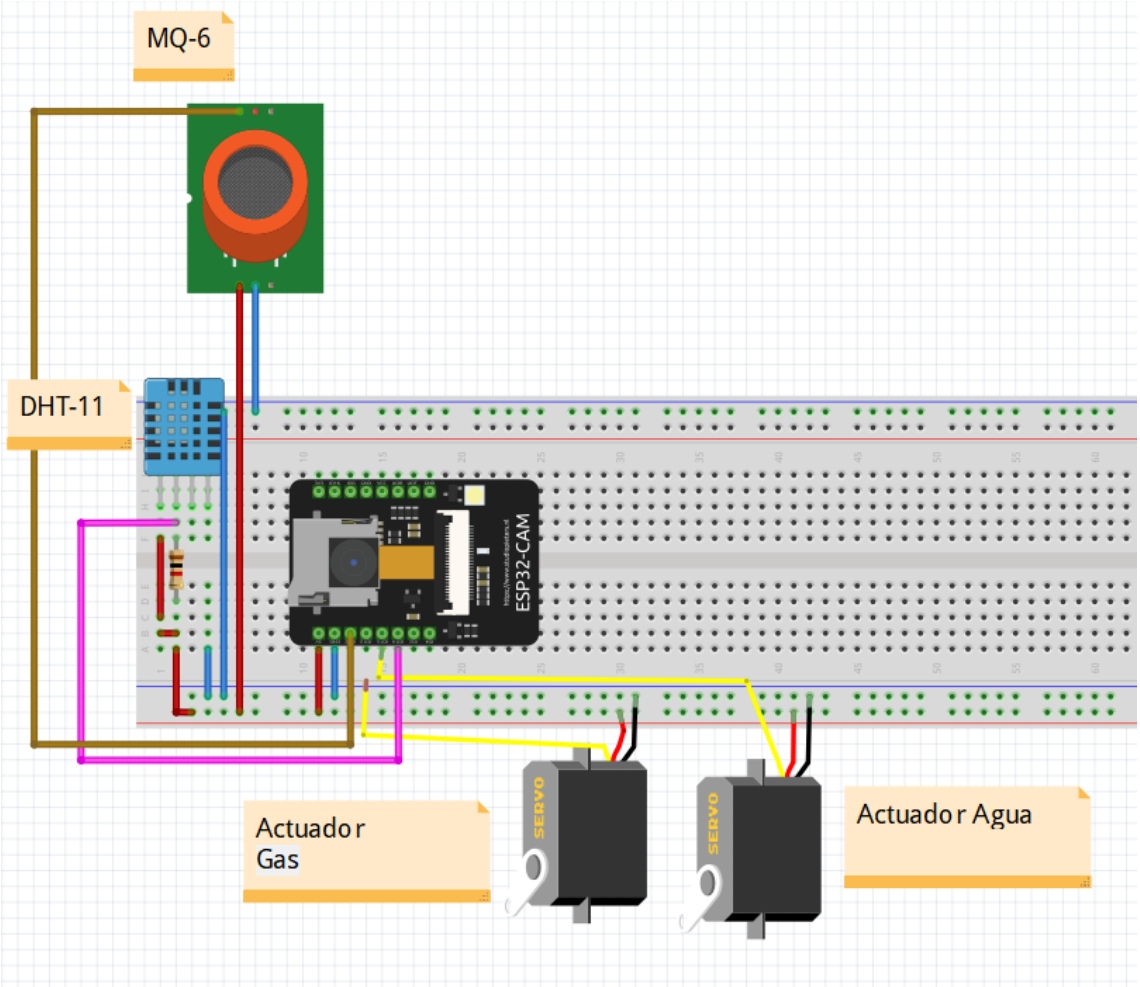

Figura 5. Diagrama del circuito de integración de la ESP32-CAM con los sensores de monitoreo y actuadores (Nodo N3), creación propia.

8.2.3. Código Fuente

A continuación, se presenta el código fuente que establece el funcionamiento del Nodo N3.

```
#include <DHT.h>
#include <EspMQTTClient.h>
#include <ESP32Servo.h>

#define DHTPIN 14   // Definimos el GPIO donde se conecta el sensor DHT11 (SIGNAL)
#define DHTTYPE DHT11    // Tipo de sensor de temperatura (DHT11 o DHT22)
#define MQ6PIN 12          // GPIO donde se conecta la salida digital del sensor MQ-6
#define LED_FLASH 4        // Se enciende en presencia de gas
#define SERVO_GAS_PIN 13  // GPIO para controlar al servomotor de GAS (SG90)
#define SERVO_AGUA_PIN 15 // GPIO para controlar al servomotor de AGUA (SG90)

int periodo = 60000; //Velocidad de muestreo por omisión
```

```
unsigned long tiempoactual = 0;
int edo_servo_gas;                // 1 abierto, 0 cerrado
int edo_servo_agua;               // 1 abierto, 0 cerrado
int nivel_sensor_gas;             // 1 sin fuga de gas, 0 existe fuga de gas
int nivel_sensor_humedad;  // 1 humedad baja, 0 humedad alta

Servo servo_gas;
Servo servo_agua;

EspMQTTClient client(
  "P_Capstone",      // Nombre de la red
  "PCapstone1",      // Contraseña
  "192.172.0.102",    // Dirección del broker
  "",             // MQTTUsername Can be omitted if not needed
  "",             // MQTTPassword Can be omitted if not needed
  "Nodo N3",      // Client name that uniquely identify your device
  1883            // The MQTT port, default to 1883. this line can be omitted
);

DHT dht(DHTPIN, DHTTYPE);    // Inicializamos el sensor DHT11
String salida;

void onConnectionEstablished() {
  client.subscribe("capstone/actuador_gas", [] (const String & payload)
  {
    Serial.println(payload);
    if ((payload.equals("true")) && (edo_servo_gas == 0) && (nivel_sensor_gas == 1))
    {
      Serial.println("TRUE: Abre la válvula de gas");
      edo_servo_gas = 1;
      servo_gas.write(0);
    }
    if ((payload.equals("false")) && (edo_servo_gas == 1))
    {
      Serial.println("FALSE: Cierra la válvula de gas");
      edo_servo_gas = 0;
      servo_gas.write(180);
    }
    if ((edo_servo_gas == 1) && (nivel_sensor_gas == 0))
    {
      Serial.println("GAS: Cierra la válvula de gas");
      edo_servo_gas = 0;
      servo_gas.write(180);
    }
  });

  client.subscribe("capstone/actuador_agua", [] (const String & payload)
```

```
  {
    Serial.println(payload);
    if          ((payload.equals("true"))          &&          (edo_servo_agua          ==          0))
//&&(nivel_sensor_humedad==1))
    {
      Serial.println("TRUE: Abre la válvula de agua");
      edo_servo_agua = 1;
      servo_agua.write(0);
    }
    if ((payload.equals("false")) && (edo_servo_agua == 1))
    {
      Serial.println("FALSE: Cierra la válvula de agua");
      edo_servo_agua = 0;
      servo_agua.write(180);
    }
    if ((edo_servo_agua == 1) && (nivel_sensor_humedad == 0))
    {
      Serial.println("HUMEDAD: Cierra la válvula de agua");
      edo_servo_agua = 0;
      servo_agua.write(180);
    }
  });
client.publish("capstone/humedad_temperatura_niveldegas_niveldehumedad_estado
servogas_estadoservoagua", "Conectado al MQTT");
}

void onMessageReceived(const String& topic, const String& message) {
  Serial.println(topic + ": " + message);
}

// Interrupción para cambiar la velocidad con la que se leen los sensores
// y para CERRAR la válvula de gas en caso de que exita fuga de gas.
void IRAM_ATTR ISR() {
  if (digitalRead(MQ6PIN)) {
    periodo = 60000; // Los sensores se leen cada 60 segundos
    digitalWrite(LED_FLASH, LOW);
  }
  else {
    periodo = 6000; // Los sensores se leen cada 6 segundos
    digitalWrite(LED_FLASH, HIGH);
    if (edo_servo_gas == 1)
    {
      Serial.println("INTERRUPCIÓN PARA CERRAR la válvula de gas");
      edo_servo_gas = 0;
      servo_gas.write(180);
      //  delay(1500);
    }
```

```
  }
}

void setup() {
  pinMode(MQ6PIN, INPUT); //Se configura como pin de entrada
  pinMode(LED_FLASH, OUTPUT);
  pinMode(SERVO_GAS_PIN, OUTPUT);
  pinMode(SERVO_AGUA_PIN, OUTPUT);
  Serial.begin(115200);// Se inicializa la comunicación serial

  ESP32PWM::allocateTimer(0);
  ESP32PWM::allocateTimer(1);
  ESP32PWM::allocateTimer(2);
  ESP32PWM::allocateTimer(3);

  servo_gas.setPeriodHertz(50);    // Estandard 50 Hz
  servo_agua.setPeriodHertz(50);   // Estandard 50 Hz
  servo_gas.attach(SERVO_GAS_PIN, 1000, 2000);
  servo_agua.attach(SERVO_AGUA_PIN, 1000, 2000);

  servo_gas.write(0);   edo_servo_gas = 1;
  servo_agua.write(0); edo_servo_agua = 1;
  Serial. println("Servomotores abiertos");
  delay(3000);
  nivel_sensor_gas = 1;

  attachInterrupt(MQ6PIN, ISR, CHANGE);
  client.enableDebuggingMessages(); // Permite visualizar los mensajes que envía MQTT
  Serial.println(F("Prueba de conexión del sensor DHT11"));
  dht.begin();// Se inicializa el sensor DHT11
}

void loop() {
  // Esperamos "periodo" segundos entre mediciones
  if (millis() > tiempoactual + periodo)
  {
    tiempoactual = millis();
    nivel_sensor_gas = digitalRead(MQ6PIN); //lee el nivel digital de gasLP
    Serial.print ("Nivel digital de gas LP: ");
    Serial.println(nivel_sensor_gas);
    client.loop();
    float h = dht.readHumidity();   // Se lee la humedad relativa (en porcentaje)
    float t = dht.readTemperature();// Se lee la temperatura en grados centígrados
    if (isnan(h) || isnan(t)) {     // Comprueba lectura del sensor DHT11
      Serial.println("Error al realizar la lectura del sensor DHT11");
      return;
    }
```

```
  if ((t > 50.0) && (edo_servo_gas == 1))
  {
   Serial.println("TEMPERATURA ELEVADA: Cierra la válvula de gas");
   servo_gas.write(180);
   edo_servo_gas = 0;
  }

  if ((h >= 50.0) && (edo_servo_agua == 1)) // Detecta hummedad alta
  {
   Serial.println("HUMEDAD ALTA: Cierra la válvula de AGUA");
   servo_agua.write(180);
   edo_servo_agua = 0;
   nivel_sensor_humedad = 0;
  }
  if ((h < 50.0)) nivel_sensor_humedad = 1;

  salida = String("{\"HUMEDAD\": \" " + String(h) + "\", \"TEMPERATURA\": \"" +
String(t) + "\",\"NIVELDEGAS\": \"" + String(nivel_sensor_gas) +
"\",\"NIVELDEHUMEDAD\": \"" + String(nivel_sensor_humedad) +
"\",\"ESTADOSERVOGAS\": \"" + String(edo_servo_gas) + "\",\"ESTADOSERVOAGUA\":
\"" + String(edo_servo_agua) + "\"}");
  Serial.println(salida);

client.publish("capstone/humedad_temperatura_niveldegas_niveldehumedad_estado
servogas_estadoservoagua", salida);
 }
 else client.loop();
}
```

8.3. Nodo de Control

La labor del Nodo de Control es recopilar la información generada por los Nodos N1, N2 y N3, almacenarla para su análisis posterior y la toma de decisiones del sistema en general.

Este proceso se lleva a cabo utilizando la Raspeberry Pi 4 como servidor de Node-Red, agente público MQTT y servidor de Base de Datos (por ejemplo, Maria DB). La Raspberry Pi 4 se encuentra conectada dentro de una red de infraestructura donde comparte información con los Nodos 1, 2 y 3, esta información se pasa a través de mensajes utilizando el protocolo MQTT (por sus siglas en inglés Message Queue Telemetry Transport). Estos mensajes son recopilados desde el Agente MQTT utilizando Node-Red leyendo la información que producen los nodos en sus respectivos tópicos. Esta información (como se puede apreciar en la descripción de los nodos) se transmite en forma de JSON, por lo cual es posible realizar un “parser” que permite convertirlos en texto plano y desplegarlos en la interfaz gráfica (Figura 6).

Figura 6. Interfaz gráfica para el desplegado de información y control de los actuadores de forma manual.

La información recopilada utilizando los nodos de la Figura 7 se almacena en una base de datos utilizando los bloques de la Figura 8.

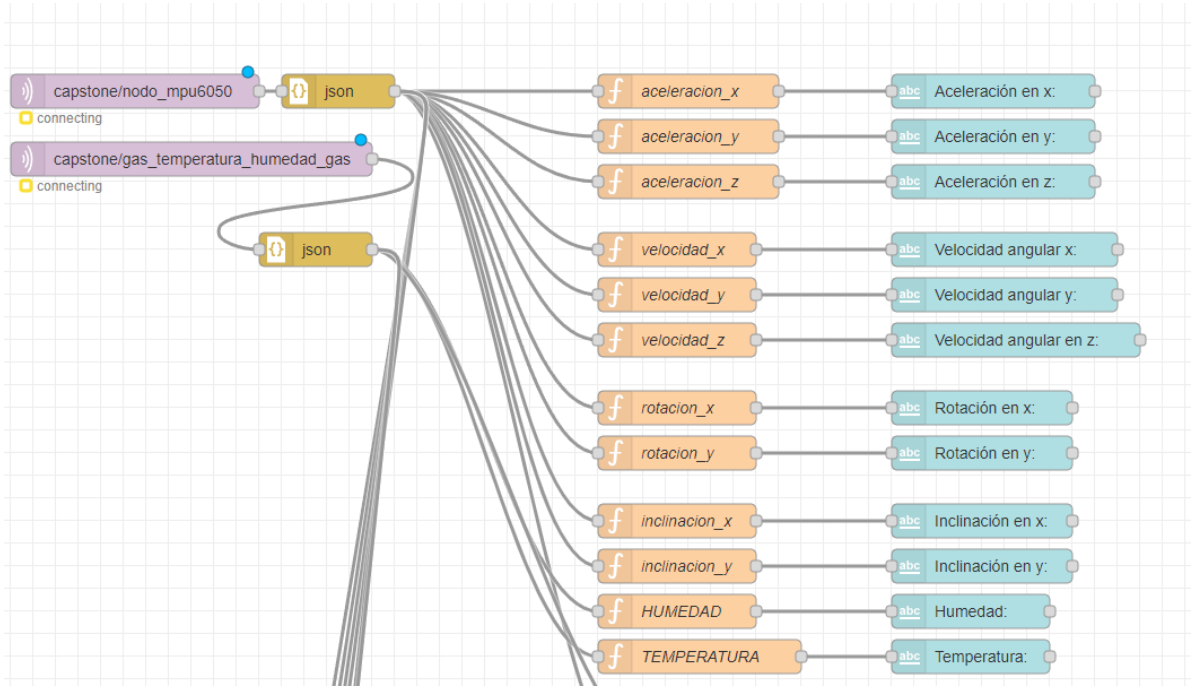

Figura 7. Etapa de adquisición de información del diagrama a bloques del nodo de control realizado en node-red.

Al mismo tiempo, dentro de los JSON se verifica la información de los niveles de gas, temperatura, humedad y de vibración, de esta manera en caso de que exista alguna fuga o se esté produciendo un evento sísmico, los actuadores que controlan las llaves de paso de agua y de gas se cierren de manera automática. Para esto, se implementó una etapa lógica que permite verificar la información proveniente de los nodos para generar

mensajes que les indiquen a los actuadores que realicen la función de apertura o de cerrado.

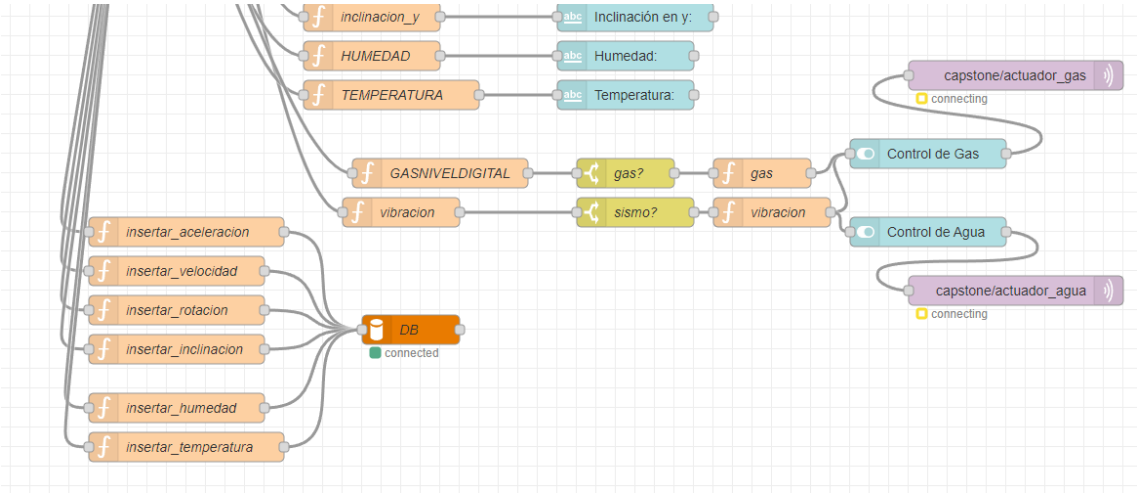

Figura 8. Etapa manejo de información a base de datos y control de actuadores del diagrama a bloques del nodo de control realizado en node-red.

Para verificar el código y observar cada una de las funciones que se generaron, referirse al repositorio de GITHUB que contiene el JSON del diagrama de Node-Red en la siguiente liga: https://github.com/mramirezr/ProyectoCapstone/tree/main/nodo_control.

8.4. Base de Datos

La base de datos se desplegó en MariaDB utilizando como servidor la Raspberry Pi4, en ella se tiene una base de datos llamada “proyecto capstone”, esta base de datos guarda las siguientes tablas.

Tabla 1. Tablas contenidas dentro de la base de datos “proyecto_capstone”

Base de datos: proyecto_capstone
aceleracion
humedad
inclinacion
rotacion
temperatura
velocidad

A su vez, cada una de las tablas contiene la siguiente información (tablas 2 a la 7).

Tabla 2. Contenido de la tabla "aceleración"

COLUMNA	FECHAHORA (DATETIME)	AX (FLOAT)	AY (FLOAT)	AZ (FLOAT)
METADATO	Fecha y hora de la medición realizada de la aceleración angular.	Aceleración Angular en x.	Aceleración Angular en y.	Aceleración Angular en z

Tabla 3. Contenido de la tabla "humedad"

COLUMNA	FECHAHORA (DATETIME)	HUM (FLOAT)
METADATO	Fecha y hora de la medición realizada de la humedad.	Humedad ambiental captada por el dht11.

Tabla 4. Contenido de la tabla "inclinación"

COLUMNA	FECHAHORA (DATETIME)	AX (FLOAT)	AY (FLOAT)
METADATO	Fecha y hora de la medición realizada de la inclinación angular.	Inclinación Angular en x.	Inclinación Angular en y.

Tabla 5. Contenido de la tabla "rotación"

COLUMNA	FECHAHORA (DATETIME)	AX (FLOAT)	AY (FLOAT)
METADATO	Fecha y hora de la medición realizada de la rotación angular.	Rotación Angular en x.	Rotación Angular en y.

Tabla 6. Contenido de la tabla "temperatura"

COLUMNA	FECHAHORA (DATETIME)	TEMP (FLOAT)
METADATO	Fecha y hora de la medición realizada de la temperatura.	Temperatura captada por el dht11.

Tabla 7. Contenido de la tabla "velocidad"

COLUMNA	FECHAHORA (DATETIME)	GX (FLOAT)	GY (FLOAT)	GZ (FLOAT)
METADATO	Fecha y hora de la medición realizada de la velocidad angular.	Velocidad Angular en x.	Velocidad Angular en y.	Velocidad Angular en z

La información fue capturada en la base de datos utilizando el nodo de control en Node-Red a través de un interfaz de conexión provista por los bloques del sistema.

La información captura se utiliza para generar reportes interactivos utilizando una interfaz de concentración de información en Grafana.

8.5. Interfaz de concentración de Información.

En Grafana se diseñó una interfaz gráfica que permite reportar la información estadística de la intensidad de los movimientos sísmicos y la información acerca del estado de la vivienda, esta información es de suma importancia ya que en trabajos a futuro se puede utilizar para el diseño de modelos matemáticos que permitan mejorar el diseño estructural de las viviendas en zonas sísmicas. En las Figuras 9 y 10 se aprecia la interfaz gráfica de concentración de información que despliega de manera clara, precisa y amigable la información recabada por el sistema, de esta manera se pueden generar reportes temporales para un futuro análisis.

NOTA: Para verificar el funcionamiento del prototipo se construyó una mesa vibratoria que simula sismos, la frecuencia de oscilación se puede controlar y puede ser variable. Para las pruebas se utilizó una frecuencia de oscilación en el eje x que va de 0 a 3 Hz. Además, se utilizó un encendedor para activar el sensor de gas LP.

Figura 9. Interfaz Gráfica para el reporte de información: Velocidad Angular y Aceleración; Rotación e Inclinación.

Figura 10. Interfaz Gráfica para el reporte de información: Temperatura y Humedad.

9. Trabajo a Futuro

Como trabajo a futuro se propone establecer un enlace entre el servidor y el canal del Sistema de Alerta Sísmica Nacional para alertar con anticipación a las personas de la vivienda y éstas puedan salir con calma antes de que inicie el sismo (es bien sabido que en ocasiones no se activa la alerta por fallas técnicas). También se propone que la red inalámbrica sea tipo ad-hoc o basada en el concepto de radio cognoscitivo para solventar posibles fallas de las redes inalámbricas o móviles comerciales (WiFi o Celular) [2]. Adicionalmente, se contemple el uso de sensores embebidos o sensores de fibra óptica y el uso de cámaras de video para visualizar el estado de trabes de carga y muros principales de la vivienda.

10. Referencias.

[1] UN, 2015. Transforming Our World: the 2030 Agenda for Sustainable Development. A/RES/70/1. New York, US.

[2] Hassel A. Alcalá-Garrido, Mario E. Rivero-Angeles, Eleazar Aguirre-Anaya, Felipe A. Cruz-Pérez, Sandra L. Castellanos-López, and Genaro Hernández-Valdez, "Performance Analysis of a Wireless Sensor Network with Cognitive Radio Capabilities in Structural Health Monitoring Applications: A Discrete Model," International Journal of Distributed Sensor Networks (Special Collection on Structural Health Monitoring for Mechanical Structures Using Multi-Sensor Data), no. 5, vol, 14, pp. 1-16, 2018.

[3] López, A., C. I. Álvarez y E. Villarreal. 2017. Migración de fuentes sísmicas a lo largo del Cinturón de Fuego del Pacífico. La Granja: Revista de Ciencias de la Vida. Vol. 25(1):5-15. pISSN:1390-3799; eISSN:1390-8596.

[4] Gobierno de México, 2021. *Sismología de México*. [online] Sgm.gob.mx. Disponible en: <https://www.sgm.gob.mx/Web/MuseoVirtual/Riesgos-geologicos/Sismologia-de-Mexico.html> [Última visita 21 octubre 2021].

[5] CIRES, 2021. *Tipos de suelo en el Distrito Federal y Zona Metropolitana*. [online] Blogcires.mx. Disponible en: <https://blogcires.mx/tag/tipos-de-suelo-en-el-distrito-federal-y-zona-metropolitana/> [Última visita 21 octubre 2021].

[6] PAOT, 2021. *Panorama de Riesgo Sísmico*. [online] Paot.org.mx. Disponible en: <http://www.paot.org.mx/micrositios/riesgo-sismico/antecedentes.html> [Último acceso 21 octubre 2021].

[7] Rincón, S., 2021. CDMX reporta 14 daños en inmuebles tras sismo. [online] Forbes México. Disponible en: <https://www.forbes.com.mx/noticias-cdmx-reporta-14-danos-en-inmuebles-tras-sismo-de-esta-manana/> [Último acceso 21 octubre 2021].