ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"DESARROLLO DE UN PROTOTIPO DE ALARMA SÍSMICA DE BAJO COSTO CON HERRAMIENTAS DE CÓDIGO ABIERTO"

INFORME DE MATERIA INTEGRADORA

Previa a la obtención del Título de:

INGENIERO EN TELEMÁTICA

GUILLERMO ANDRES BERMEO VELASQUEZ

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTOS

Agradezco a Dios, a mis abuelos Vicente y Teresa, a mis tías Luz y Rosario, a mis profesores Jorge, Ignacio, Patricia, Néstor y Gabriel. En especial a mi compañera del alma Victoria.

Guillermo Bermeo

DEDICATORIA

A mi ñaño, wiki, charo y luz.

Guillermo Bermeo

TRIBUNAL DE EVALUACIÓN

Ing. Gabriel Astudillo, MSIG.	**Ing. Néstor Arreaga, MG.**
PROFESOR EVALUADOR	PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

_____

Guillermo Andrés Bermeo Velásquez

RESUMEN

En este documento se describe una solución para el monitoreo de aceleraciones del suelo del edificio 15A de la FIEC para emitir una alarma durante evento sísmico, así mismo, se registran en un historial local y en un servidor externo para un acceso a las lecturas más detallado.

La primera fase corresponde a una aplicación web adaptiva que permite monitorear la información del sistema y visualizarlo desde cualquier ordenador. La segunda fase consiste en la creación de una infraestructura servidor-cliente, tal que, el cliente envía la información de cada estación de cada nodo y esta lo almacena localmente y luego mediante un cronjob se envían los logs al servidor; el servidor también se encarga de generar las alertas y enviarlas a sus respectivos destinatarios. La tercera fase corresponde a la implementación de un sistema de monitoreo de aceleraciones del suelo utilizando *Arduino* para que recolecte datos y en caso de cumplir las condiciones programadas activar una sirena para alertar a los usuarios de que existe un evento sísmico en proceso.

Para realizar este proyecto se utilizaron varios lenguajes de programación tales como: C, shell, python, javascript, html, entre otros, por lo que para comunicarse entre sí se utilizó MQTT que consiste en un standard ISO basado en un protocolo publicación-suscriptor TCP, diseñado para conexiones remotas donde el ancho de banda está restringido. Mediante el uso de un bróker basado en comunicación TCP se gestiona las publicaciones y los suscriptores. Para generar las alertas de *SMS* se utiliza un shield GSM conectado directamente al arduino para notificar de la actividad a los usuarios.

GLOSARIO DE TERMINOS

HTML:	HyperText Markup Language, lenguaje de marcado para la elaboración de sitios web.
CRON JOB:	Tarea programada a ejecutarse en intervalos regulares.
SHELL:	Lenguaje de programación diseñado para operar en Unix.
LOG:	Historial de las eventos y operación que se realizan en un sistema.
TCP:	Protocolo de control de transmisión de datos.
NUBE:	Propuesta tecnológica que permite ofrecer servicios a través de la red.
SMS:	Servicio de mensajes cortos que son usados para comunicación entre dispositivos móviles.
API:	Interfaz de programación de aplicaciones.
HTTP:	Hypertext Transfer Protocol.
PYTHON:	Lenguaje de programación de alto nivel multiproposito.
MQTT:	Message Queue Telemetry Transport, un protocolo de comunicación Maquina a Maquina diseñado para Internet de las cosas .
FRAMEWORK:	Infraestructura digital, estructura conceptual definido, contiene módulos concretos de software.

ÍNDICE GENERAL

CAPÍTULO 1

1. PLANTEAMIENTO DEL PROBLEMA.

La Facultad de Ingeniería en Electricidad y Computación (FIEC) cuenta con varios edificios de oficinas, aulas y laboratorios, los cuales se usan para ofrecer un ambiente para que los alumnos, docentes y empleados ejerzan sus actividades diarias de aprendizaje y enseñanza.

El uso de la infraestructura es esencial para el desarrollo de las actividades y representa en un problema de seguridad grave que no existan alarmas para eventos sísmicos que permitan a los usuarios de las edificaciones salir hacia los puntos de reunión. El perfil costanero ecuatoriano al estar dentro del “círculo de fuego” siempre ha registrado una elevada actividad sísmica, motivo por el cual se dictan capacitaciones y realizan simulacros para evitar el menor número de afectados en el caso de eventos sísmicos de gran escala.

Para que los estudiantes, profesores y trabajadores puedan laborar con normalidad las aceleraciones del suelo deben ser nulas y por consiguiente no representar una amenaza para la integridad de los usuarios de la estructura.

El seguir las instrucciones del manual para eventos sísmicos se dificulta en medio del acontecimiento, ya que se podría no percibir de correctamente lo que está sucediendo o sobredimensionar el suceso e infundir pánico en los demás.

1.1 Descripción del problema

Se necesita de un equipo que monitoree las aceleraciones del suelo en el edificio 15A de la FIEC, así mismo en los demás edificios correspondientes a la facultad, teniendo en cuenta que para cada edificio se necesitaría hacer pruebas para determinar el sitio adecuado para la colocación de la alarmas y sensores.

De esta manera se puede tener lecturas más precisas y confiables de las aceleraciones del suelo.

Adicional, a pesar de que el equipo opera independientemente para ejecutar alertas, se debe configurar cada uno para permitirle el acceso a la red de la universidad con el fin de enviar los datos hacia el servidor en el cual se registran los valores medidos por el acelerómetro.

Para llegar a la implementación debe tener en cuenta lo siguiente:

- Se debe contar con un módulo que permita medir las aceleraciones del suelo.
- Implementar un módulo de alarmas y GSM para enviar a los usuarios del edificio en caso de alguna incidencia.
- El sistema debe contar con un servidor que este monitoreando los sensores y los equipos en caso de una desconexión.
- Se debe implementar una aplicación web adaptativa para que un operador pueda observar con facilidad las lecturas de los sensores de aceleración.

1.2 Justificación del problema

Debido a los eventos sísmicos ocurridos en el país desde el año 2016, han surgido replicas y nuevos sismos sobre todo en el litoral ecuatoriano, y dado que es mandatorio el uso de los edificios en la facultad para realizar las actividades académicas y administrativas diarias.

Este proyecto ayudará a prevenir y salvaguardar la integridad de los estudiantes, profesores y trabajadores mediante un sistema de monitoreo de la aceleración del suelo, el cual a través una interfaz web se muestra la información de los sensores de aceleración del Edificio 15A de Facultad de Ingeniería en Electricidad y Computación.

Para garantizar el buen funcionamiento de los equipos, éstos deben tener sus respectivos periféricos conectados y estar ubicados en una zona ideal con conexión a la red y lejos de la calle, al menos a 100 metros, para evitar que el ruido de vehículos pesados al transitar afecte a las lecturas del acelerómetro. Además, para que estos equipos puedan estar siempre conectados, los dispositivos de comunicación deben estar operativos a cada momento.

Así, cuando el equipo detecte que ha ocurrido un cambio en la aceleración del suelo, éste notifique de lo ocurrido a los usuarios del edificio en el cual este localizado.

1.3 Objetivos

1.3.1 Objetivo general

Implementar un prototipo de monitor de alarma para eventos sísmicos. Un servidor permite conocer las mediciones del sensor, utilizando hardware y software libre.

1.3.2 Objetivos específicos.

- Desarrollar un sistema cliente-servidor para el monitoreo de las aceleraciones del suelo en el edificio 15A.
- Implementar una interfaz para la administración del sistema en el cual, puedan ver la información de los equipos que se encuentran monitoreando.
- Implementar un módulo electrónico que cense la aceleración del suelo para su posterior almacenamiento en logs y en una base de datos.
- Implementar un módulo que permita realizar alertas vía SMS para notificar de eventos sísmicos a los usuarios.
- Implementar un módulo que permita una alerta sonora a los usuarios del edificio para notificarles de que un evento sísmico de alto riesgo está en curso.

1.4 Limitaciones y alcance del sistema

- Dado que es necesario al menos tres equipos de monitoreo censando las aceleraciones y separados uno del otro, no es posible realizar una triangulación y así determinar e informar con precisión el hipocentro, ni el epicentro, ni la magnitud del sismo.
- Los muestreos de las aceleraciones del suelo deben tomar como máximo 200 milisegundos.
- Este prototipo fue orientado para alertar a los usuarios del edificio 15A de la FIEC por lo cual el consumo energético de los mismos no han sido prioridad.
- El sistema está diseñado para el uso en el edifico 15A sin embargo puede ser usado en cualquier otro edificio que cumpla con las características requeridas para su correcto funcionamiento.
- Este prototipo no es una alarma temprana en caso de eventos sísmicos debido a que para que se considere como tal, se deben ubicar los equipos a lo largo de la falla geológica.

CAPÍTULO 2

2. MARCO TEÓRICO.

2.1 HTML (Hyper Text Markup Language)

Es un lenguaje en el cual se desarrollan las páginas web. Es un estándar de hipertexto, esto quiere decir que permite escribir texto de forma estructurada para la definición de contenido de una página web como texto, imágenes, videos entre otros. Un documento HTML no solo se compone de texto, sino también de imágenes, sonido, videos, y demás, como resultado puede obtenerse un documento de tipo multimedia.

HTML se escribe en forma de etiquetas rodeadas de corchetes tal que también puede describir la apariencia y flujo de una página web. Además, puede tener referencia a un tipo de programa llamado script y que puede afectar el comportamiento de los browsers y otros intérpretes de código HTML. [1]

2.2 CRONJOB

Es una tarea programada que se ejecuta a través del demonio “cron”, esta tarea se ejecuta a intervalos regulares (por ejemplo, cada minuto, hora, día o semana), Cron es impulsado por un crond, un archivo de configuración que específica el comando shell para ejecutarse periódicamente a una hora específica. [2]

2.3 SHELL

Es un término en UNIX que hace referencia a la interacción entre el usuario y el sistema operativo, es una capa de programación que entiende y ejecuta los comandos ingresados por el usuario, en algunos sistemas también es conocido como INTERPRETE DE COMANDOS. [3]

2.4 LOG

Un log es un archivo en el cual puedes encontrar un historial de los eventos que han sido ejecutado en el sistema operativo, aplicaciones y servicios. Estos archivos son guardados en texto plano para facilitar su lectura. [4]

2.5 TCP

Es un protocolo de control de transmisión seguro orientado a conexión, provee de un transporte fiable ejecutados entre pares de sistemas finales, usando el servicio de la capa de red provista por el protocolo IP. [5]

2.6 API

Un Interfaz de Programación de Aplicaciones es un conjunto de rutinas, protocolos y herramientas usadas para crear aplicaciones de software, básicamente explica como los componentes de un software deben interactuar y son ampliamente usadas para programas interfaces graficas de usuario. [6]

2.7 NUBE

Es un término general que hace referencia servicios provistos a través de internet, estos servicio pueden ser de uso público, privados o híbridos, y se clasifican de acuerdo a sus beneficios en: SaaS(Software como Servicio), PaaS(Plataforma como Servicio) e IaaS(Infraestructura como Servicio). [7]

2.8 ARDUINO YÚN

Arduino Yún es un dispositivo diseñado para proyectos de equipos conectados, generalmente, para el internet de las cosas ya que combina el poder de Linux y la sencillez de Arduino. Arduino Yún es una placa para prototipos que contiene un microcontrolador Atmega32u4 y el Atheros AR9331. Atheros es un procesador que soporta una distribución de Linux basado en OpenWRT llamado Linino OS, el equipo contiene embebido un puerto Ethernet RJ45, soporte WiFi, USB tipo-A, socket para micro SD y un cristal oscilador de 16MHz.

Yún se distingue de otros dispositivos al tener la habilidad de comunicarse con la distribución de Linux embebida, ofreciendo una poderosa computadora conectada a la red con las facilidades que presta Arduino, adicionalmente con el uso de comandos de Linux tales como cURL, puede desarrollarse shells y scripts de Python para interacciones robustas. [8]

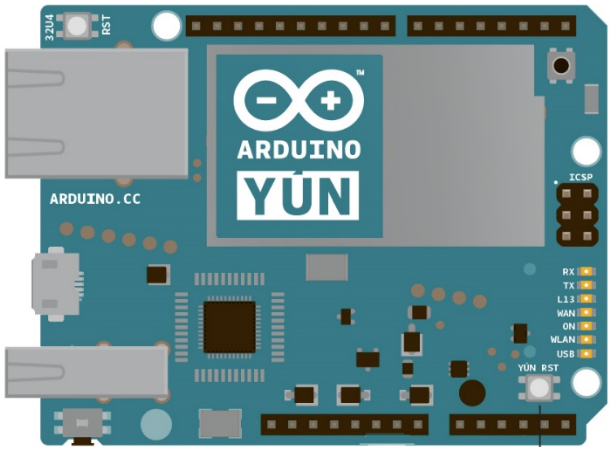


Figura 2.1: Placa Arduino YUN.

2.9 MÓDULO MPU6050

El MPU6050 es un sensor digital que incluye un acelerómetro y un giroscopio. Este sensor permite 6 grados de libertad y cuenta con una resolución de 16bits lo cual divide el rango en 65536 fracciones y se aplican a cada eje (X, Y, Z). [9]

Características:

- Tensión de alimentación: 2.37 a 3.46V
- Giroscopio con sensibilidad de ±250, ±500, ±1000, y ±2000dps
- Rango de temperatura: 0-50 ° C error de ± 2 ° C
- Acelerómetro con sensibilidad de ±2g, ±4g, ±8g y ±16g
- Interfaz: Digital.

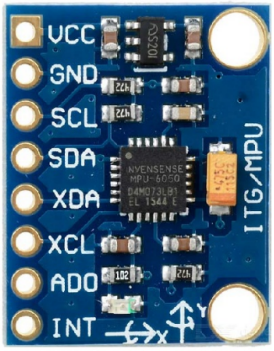


Figura 2.2: Módulo MPU6050

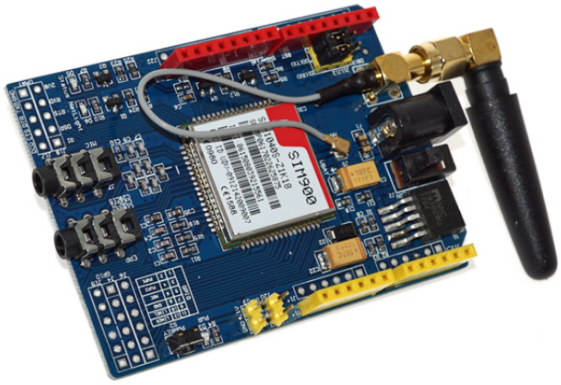
2.10 SHIELD GSM 900 ARDUINO

Es un módulo que permite el envío de SMS, realizar y recibir llamadas.

Su al arduino es a través del puerto serial, entre sus características tenemos: [10]

- Requiere placa *Arduino*
- Voltaje de operación: 5v 2A

- Bandas: 850 MHz, 900 MHz, 1800 MHz, 1900 MHz
- Tipo de SIM: SIM



2.11 NODE-RED

Es una herramienta de programación para unir dispositivos físicos mediante el uso de código, es una aplicación basada en un navegador que permite crear rutinas de manera muy sencilla y rapida. [11]

CAPÍTULO 3

3 IMPLEMENTACIÓN Y DESARROLLO.

La solución al problema descrito se divide en tres partes; La primera fase corresponde a una aplicación web adaptiva que permite monitorear la información del sistema y visualizarlo desde cualquier ordenador. La segunda fase consiste en la creación de una infraestructura servidor-cliente, tal que, el cliente envía la información de cada estación de cada nodo y esta lo almacena localmente y luego mediante un cronjob se envían los logs al servidor; el servidor también se encarga de generar las alertas y enviarlas a sus respectivos destinatarios.

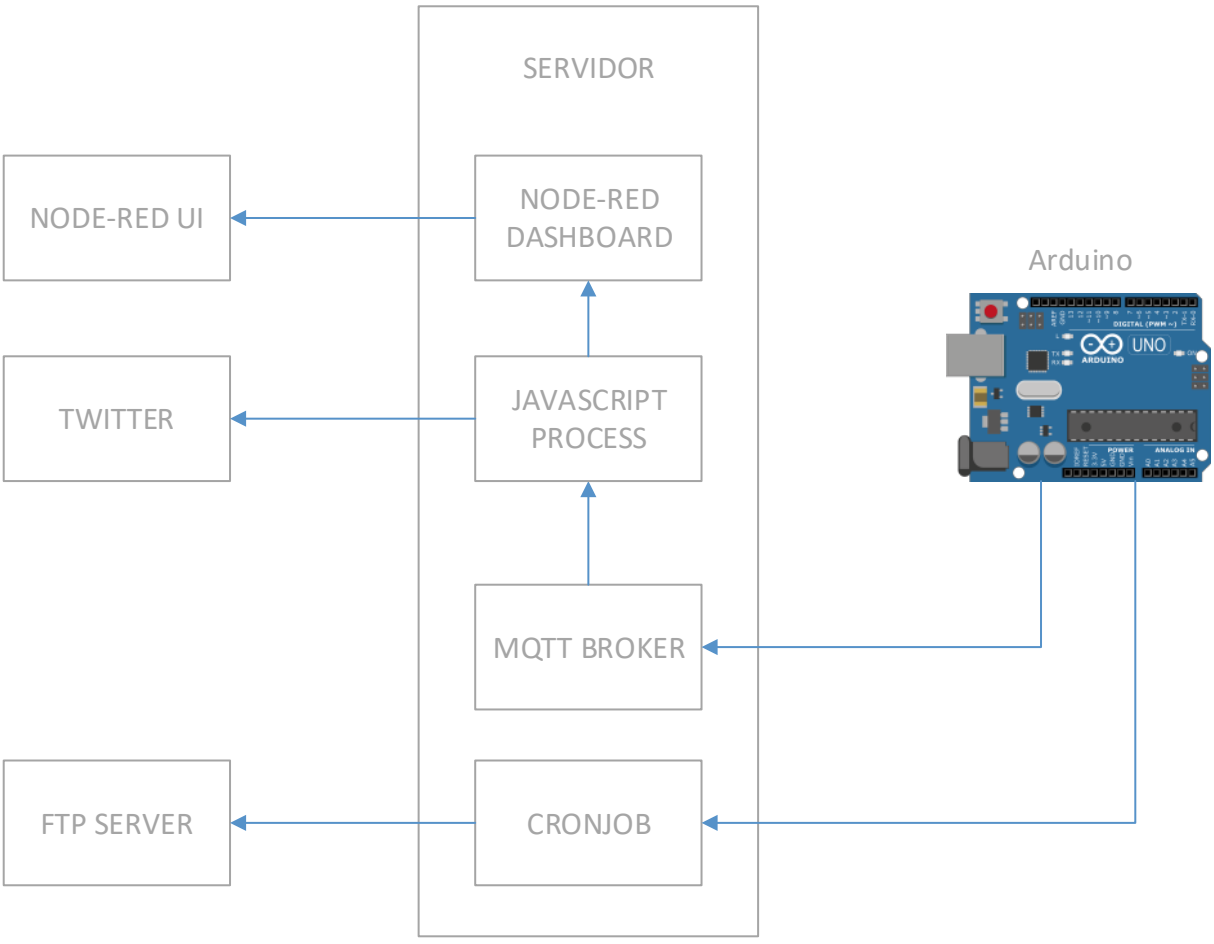


Figura 3.01: Diagrama de bloques fase uno y dos.

La tercera fase corresponde a la implementación de un sistema de monitoreo de aceleraciones utilizando *Arduino* para la recolección datos, en caso de cumplir las condiciones programadas activar una alarma para alertar a los usuarios de que existe un evento sísmico en proceso.

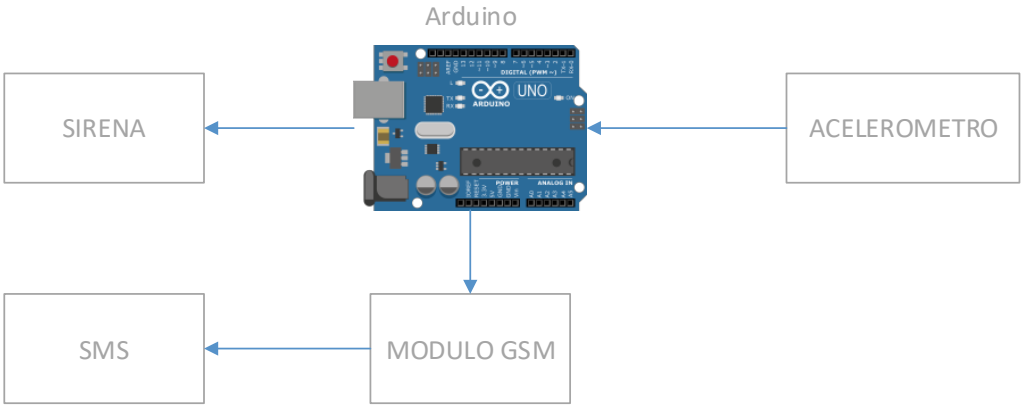


Figura 3.02: Diagrama de bloques fase tres.

3.1 Aplicación Web Adaptiva

El diseño de la aplicación web se lo realizó mediante la programación con los lenguajes HTML y JAVASCRIPT utilizando un framework llamado *node-red* basado en node.js para crear de una manera más sencilla un sitio web. Para la manipulación de los datos de la base desde la interfaz web, y validaciones se usó JavaScript.

3.1.1 Ingreso al sistema

Como se muestra en la figura 3.4 la interfaz para el control y modificación de la rutina del sistema se hace a través de la ip del servidor web y el puerto 1880

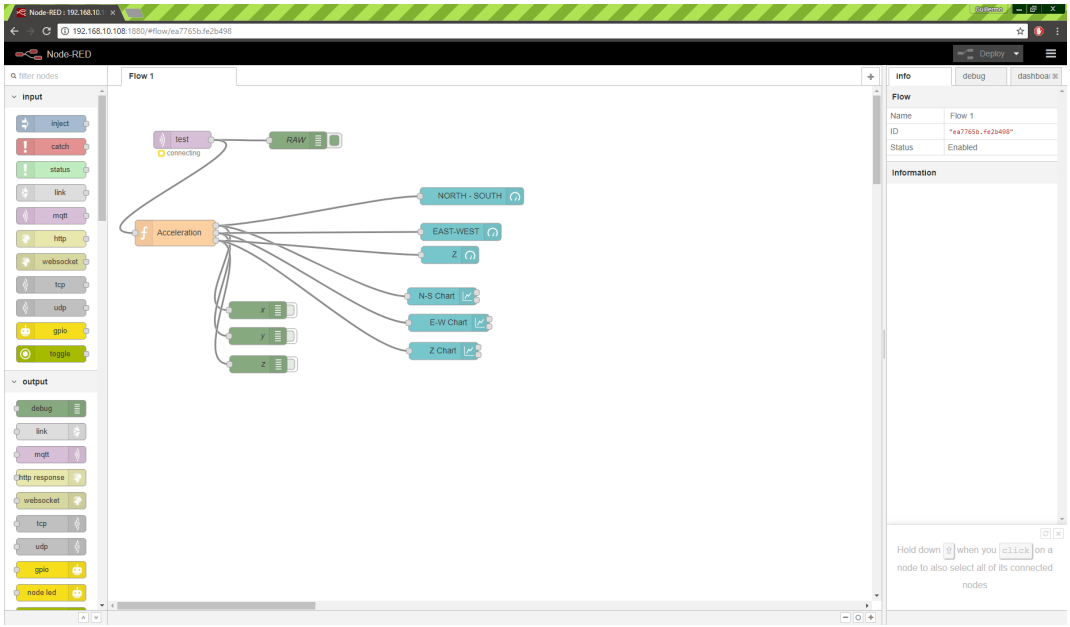


Figura 3.03: Pantalla de ingreso al sistema Node-red.

3.1.2 Bloques

Los bloques utilizados se detallan a continuación:

- **MQTT**

Se conecta a un Bróker de MQTT y se subscribe a los mensajes dado un Tema específico.

- **FUNCTION**
  Ejecuta un código JavaScript usando los datos del mensaje recibido.
- **DEBUG**
  Muestra las propiedades del mensaje recibido en la barra de Debug y también el historial de mensajes, por lo general muestra lo contenido en *msg.payload*
- **GAUGE**
  Añade un widget de tipo medidor a la interfaz de usuario.
- **CHART**
  Añade un Widget de tipo Grafico, que rotula los valores enviados.
- **TWITTER**
  Permite enviar un Tweet una vez configurada la cuenta de usuario.

3.1.3 Monitoreo

Esta ventana fue implementada para mostrar un historial gráfico y en tiempo real de las aceleraciones medidas por el sensor.

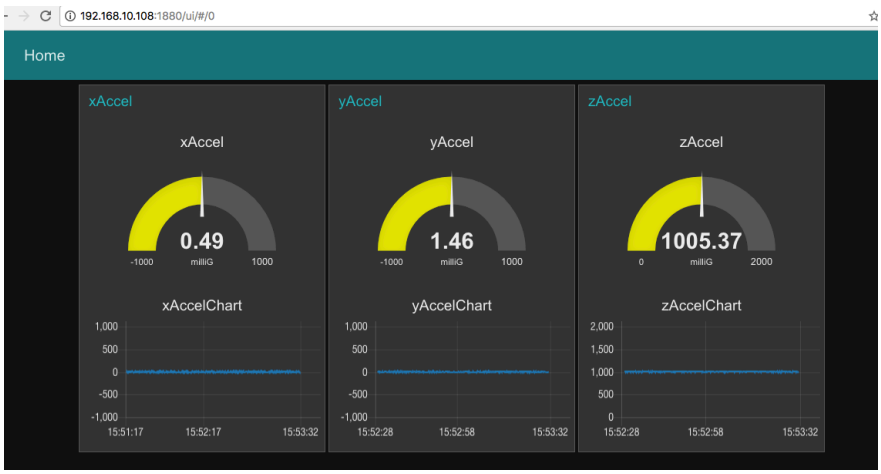


Figura 3.04: Pantalla de Monitoreo

La figura 3.5 presenta la gráfica de aceleraciones en tiempo real e historial de aceleraciones del suelo del edificio, para implementar las gráficas se utilizó la paleta de dashboard-ui:

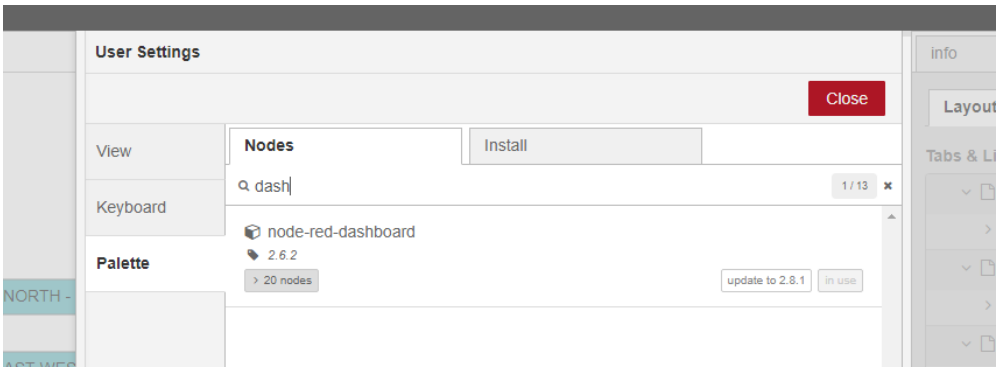


Figura 3.05: Paletas de Node-Red

3.2 Servidor

El servidor puede ser creado en cualquier sistema operativo disponible, para este proyecto se usó MAC OS High Sierra.

3.2.1 Preparación del servidor

El servidor está basado en node.js, que trabaja en múltiples sistemas operativos, en nuestro caso se usó OS High Sierra, necesita tener instalado Homebrew instalado, para lo cual ejecutaremos:

```
Instalar y configurar Homebrew:

brew install node

Luego descargamos e instalamos el paquete de node-red:

 curl "https://nodejs.org/dist/latest/node-${VERSION:-$(wget -qO-
 https://nodejs.org/dist/latest/ | sed -nE 's|.*>node-
 (.*)\.pkg</a>.*|\1|p')}.pkg" > "$HOME/Downloads/node-latest.pkg" &&
 sudo installer -store -pkg "$HOME/Downloads/node-latest.pkg" -target
 "/"

 Una vez instalado, levantamos el servidor con el comando:

 node-red
```

3.2.2 Instalando el Bróker de MQTT

El bróker MQTT hace las veces de intermediario entre los usuarios y los proveedores de contenido, filtrando mediante temas.

```
Instalando el broker Mosquitto, despues de haber instalado el
homebrew previamente.

brew install mosquitto

Una vez instalado el broker , ejecutamos lo siguiente para
configurarlo y especificar que queremos usar el siguiente archivo:
/usr/local/sbin/mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
```

```
brew install mosquitto

Para suscribirnos a temas usarmos el siguiente comando:
mosquitto_sub -V mqttv311 -t test –d

-V version de mosquitto a usarse
-t tema a suscribirse
-d permitir debug
```

3.2.3 Script notificación vía Twitter

Para realizar la notificación vía twitter necesitas configurar la cuenta en el servidor:

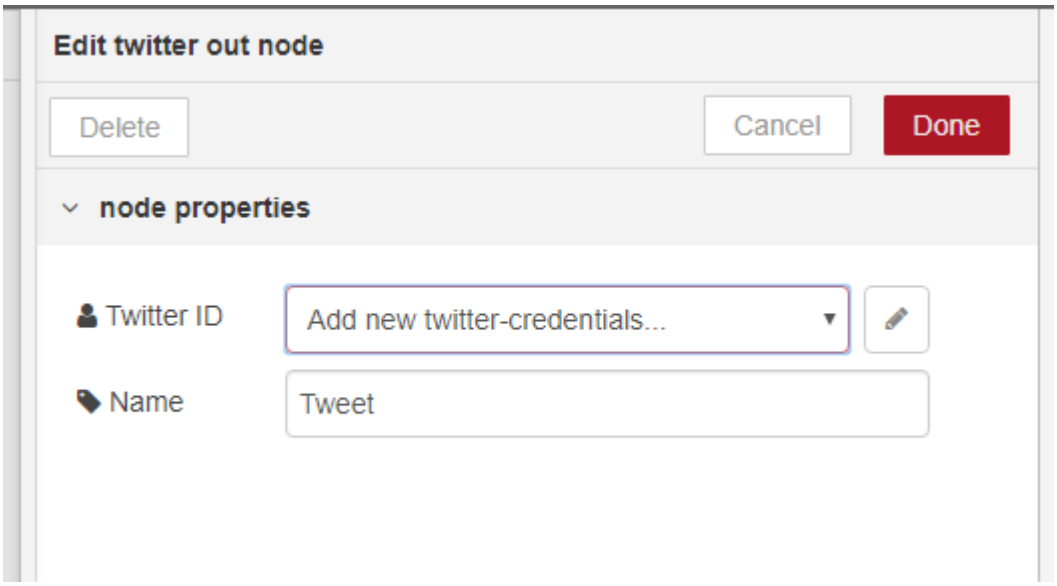


Figura 3.06: Nodo Twitter, Configuración de Credenciales

Una vez configurada las credenciales se procedió a configurar en el nodo de función el mensaje a ser enviado:

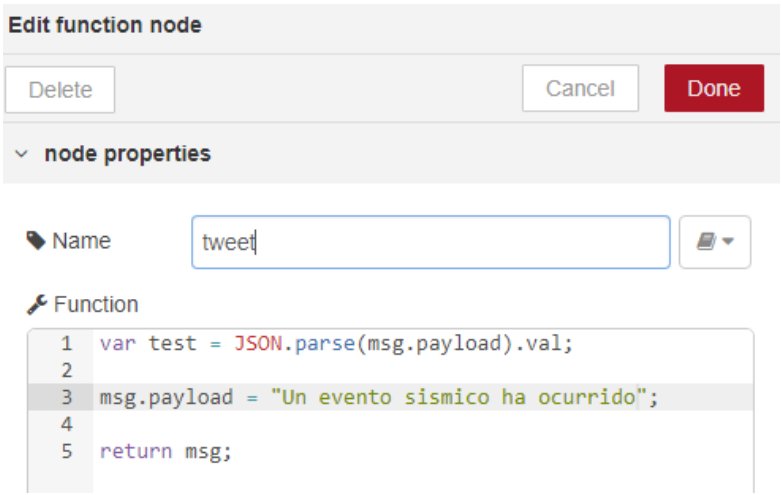


Figura 3.07: Nodo Twitter, Configuración de Credenciales

3.2.4 Script de conexión y copiado de archivos vía FTP

Para realizar un respaldo total de los datos almacenados en la tarjeta micro SD se ejecutan los siguientes scripts que comprimirán el directorio donde se guardan los registros y luego se transferirán al servidor FTP.

```
#!/bin/sh

tar -zcvf /mnt/logs.tar.gz /mnt/sda1
chmod 777 /sda1/mnt/logs.tar.gz
```

Figura 3.08: Script para comprimir el directorio sda1

```
#!/bin/sh
HOST='192.168.20.2'
USER='admin'
PASSWD='admin'
FILE='/mnt/logs.tar.gz'

ftp -n $HOST <<END_SCRIPT
quote USER $USER
quote PASS $PASSWD
put $FILE
quit
END_SCRIPT
exit 0
```

Figura 3.09: Script para copiar el archivo comprimido vía FTP

3.2.5 Configuración del CRONJOB

Para realizar la tarea de respaldar los archivos de manera automática se creó una tarea programada modificando el archivo crontab.

bEjecutamos el comando:
 # crontab –e
Agregamos las siguientes entradas a la lista de tareas programadas

```
5 0 * * * /mnt/compress.sh
6 0 * * * /mnt/ftpCopy.sh
```

Figura 3.10: Script para copiar el archivo comprimido vía FTP

La primera entrada se ejecutará todos los días 5 minutos después de la media noche, y la segunda 1 minuto después, de esta manera se comprime el directorio y luego se envía al servidor FTP.

3.3 Acelerómetro y Modulo GSM

Se utiliza un acelerómetro MPU6050 vcc y gnd los cuales son conectados a los pines de alimentación 5v y gnd, y los pines SCL , SDA y AD0 a los pines de SCL, SDA y GND en ese orden.
A su vez el módulo GSM debe conectarse del pin TX al RX del arduino y del RX al TX del arduino.

Para obtener la aceleración se ejecutaron las siguientes instrucciones:

```
void recordAccelRegisters() {
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x3B); //Starting register for Accel Readings
  Wire.endTransmission();
  Wire.requestFrom(0b1101000, 6); //Request Accel Registers (3B - 40)
  while (Wire.available() < 6);
  accelX = Wire.read() << 8 | Wire.read(); //Store first two bytes into
accelX
  accelY = Wire.read() << 8 | Wire.read(); //Store middle two bytes into
accelY
  accelZ = Wire.read() << 8 | Wire.read(); //Store last two bytes into
accelZ
  processAccelData();
}
```

Al ejecutar dichas este código se guardan en las variables accelZ, accelY y accelZ los valores de aceleración del sensor, estos a su vez deben ser transformados a milliGs para lo cual ejecutamos la siguiente función:

```
void processAccelData() {
  gForceX = (accelX * 1000) / 16384.0; //raw data to milliG's
  gForceY = (accelY * 1000) / 16384.0; //raw data to milliG's
  gForceZ = (accelZ * 1000) / 16384.0; //raw data to milliG's
}
```

Una vez validados los datos recibidos si la aceleración es superior a 300 milliGs, se envía una notificación vía SMS usando el módulo GSM.

Usando la función sendSMS(); la cual ejecuta las siguientes instrucciones:

```
  SIM900.print("AT+CMGF=1\r");
  delay(100);
  SIM900.println("AT + CMGS = \"+593996766171\"");
  delay(100);
  SIM900.println("Un evento sismico ha ocurrido");
  delay(100);
  SIM900.println((char)26);
  delay(100);
  SIM900.println();
  delay(5000)
```

CAPÍTULO 4

4. PRUEBAS Y RESULTADOS.

Se desarrolló un sistema de monitoreo de eventos sísmicos que permite alertar a los usuarios del edificio 15A de la FIEC en caso de eventos sísmico de considerable intensidad para que los mismo puedan abandonar el edificio y evitar víctimas fatales.

El sistema es capaz de obtener la información que se necesita monitorear, alertar de manera automática además de enviarla al servidor para ser analizada. El servidor recibe los datos enviados por el nodo, lo almacena y realiza las rutinas de notificaciones través de redes sociales.

4.1 Pruebas de comunicación

Para comprobar la comunicación entre el servidor y los dispositivos clientes se muestran los paquetes recibidos del proveedor de contenido a través del bróker MQTT.

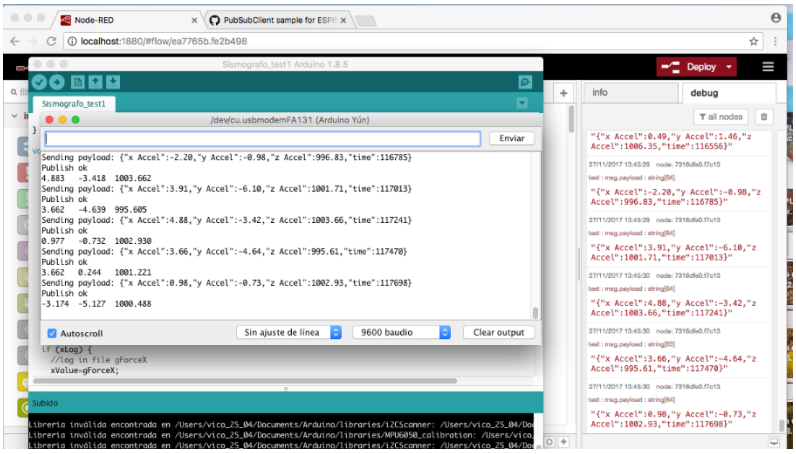


Figura 4.1: Inicialización del servicio en el cliente

4.2 Pruebas de aceleraciones, comunicación con el bróker y notificaciones.

4.2.1 Prueba de aceleraciones

Para la prueba de aceleraciones se usó una mesa a la cual se le aplicaban fuerzas externas para simular sismos con el fin de evaluar el modulo.

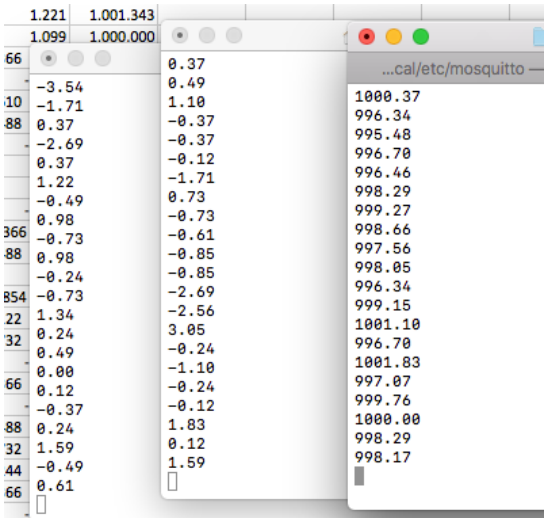


Figura 4.2: Lecturas del acelerómetro sin aceleraciones

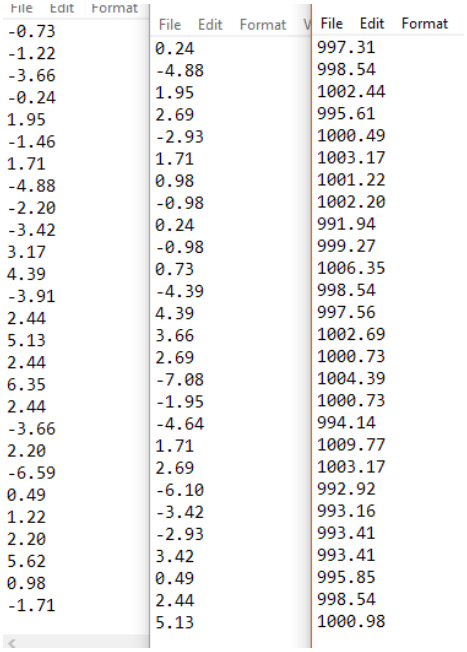


Figura 4.3: Lecturas del acelerómetro con aceleraciones

En la figura 4.4 llega un nuevo paquete, en el cual se observa que el estado del mouse es NO, lo que significa que el periférico ha sido desconectado.

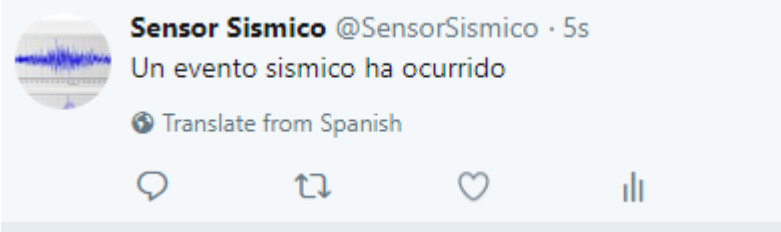


Figura 4.4: Alerta vía Twitter generada por el servidor

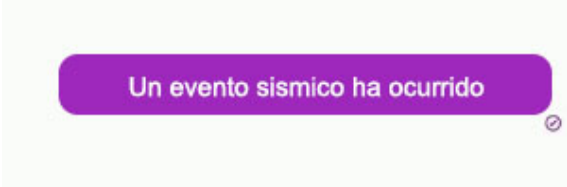


Figura 4.5: Alerta de mensaje de texto generada por el módulo GSM

En la figura 4.4 observamos la alerta via twitter que se originó por un aumento en la aceleración y en la figura 4.5 se encuentra la alerta de mensaje de texto generada por el módulo GSM.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Se logró implementar un sistema cliente-servidor para el monitoreo de aceleraciones del suelo usando lenguajes de programación tales JavaScript, HTML, ANSI C, entre otros.
2. Haciendo uso de la herramienta Node-Red y de Arduino se desarrolló una interfaz para monitorear las aceleraciones de las estructuras en tiempo real.
3. Con el uso de una plataforma de desarrollo de hardware se implementó un módulo electrónico que cense la aceleración del suelo y este a su vez sea guardado en un log de manera local.
4. Se desarrolló un script que permite enviar alertas SMS y vía twitter cuando ocurran eventos sísmicos.
5. Fue posible implementar una solución para el problema planteado anteriormente de manera eficaz, a un costo muy bajo y con herramientas de código abierto.

Recomendaciones

1. Al momento de encender el dispositivo este se auto calibrará y corregirá las inclinaciones del suelo, se debe esperar aproximadamente 5 minutos para que el sistema se estabilice.
2. El equipo puede estar conectado de manera simultánea a una red inalámbrica y alámbrica, de esta se asegura la redundancia a nivel de red.

3. Usar usuario y contraseña para los temas que es enviaran a través de MQTT en el caso de proyectos en desarrollo, así se asegurara que la información no sea recibida por usuarios no autorizados.
4. Ubicar el equipo a nivel del suelo, ya que, si es colocado en las plantas superiores, el periodo de oscilación cambiará y tendrá mediciones menos exactas.
5. Debido a que el sensor usado es proclive al ruido se aconseja ubicarlo en lugares alejados de las vibraciones ocasionadas por vehículos pesados, tales como aplanadoras y camiones.

BIBLIOGRAFÍA

[1] R. Data, «Introduction to HTML, » Refnes Data, 1999. [En línea]. Disponible: http://www.w3schools.com/html/html_intro.asp. [Último acceso: 9 ENERO 2016].

[2] Burak Guzel, « Scheduling Tasks with Cron Jobs» 26 Enero 2010. [En línea]. Disponible: https://code.tutsplus.com/tutorials/scheduling-tasks-with-cron-jobs--net-8800 [Último acceso: 10 Enero 2018].

[3] Margaret Rouse, « shell » Julio 2006. [En línea]. Disponible: http://searchdatacenter.techtarget.com/definition/shell [Último acceso: 12 Enero 2018].

[4] Juergen Haas, « An Introduction To Linux Log Files» 16 Marzo 2017. [En línea]. Disponible: https://www.lifewire.com/introduction-to-linux-log-files-2192233 [Último acceso: 12 Enero 2018].

[5] Gorry Fairhurst, « Transmission Control Protocol (TCP)» 17 Diciembre 2003. [En línea]. Disponible: http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/tcp.html [Último acceso: 08 Enero 2018].

[6] Vangie Beal, « API - application program interface» Julio 2012. [En línea]. Disponible: https://www.webopedia.com/TERM/A/API.html. [Último acceso: 14 Enero 2018].

[7] Margaret Rouse, «cloud computing» Julio 2017. [En línea]. Disponible: http://searchcloudcomputing.techtarget.com/definition/cloud-computing. [Último acceso: 10 Enero 2018].

[8] Arduino, «Arduino Yún,» [En línea]. Disponible: https://store.arduino.cc/usa/arduino-yun. [Último acceso: 17 Enero 2018].

[9] Orlando, « MPU6050 Arduino, Acelerómetro y Giroscopio» 2011. [En línea]. Disponible: https://hetpro-store.com/TUTORIALES/modulo-acelerometro-y-giroscopio-mpu6050-i2c-twi/. [Último acceso: 10 Enero 2018].

[10] Sara Santos, « Guide to SIM900 GSM GPRS Shield with Arduino,» 14 Septiembre 2017. [En línea]. Disponible: https://randomnerdtutorials.com/sim900-gsm-gprs-shield-arduino/ [Último acceso: 05 Enero 2018].

[11] Node-red, «Node-red,» 2017. [En línea]. Disponible: https://nodered.org/. [Último acceso: 02 Enero 2018].

ANEXOS

alarmaSismica.ino