

**Desarrollo de un prototipo de un sismógrafo de alerta temprana para escuelas públicas que
incluya el sistema de comunicación inalámbrica Wi-Fi y ubicación GPS**

Asimbaya García, Melissa Noemí y Suárez Criollo, Enver Miguel

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones - Carrera de Ingeniería en Electrónica,
Automatización y Control

Trabajo de titulación, previo a la obtención del título de
Ingeniero en Electrónica en Telecomunicaciones
Ingeniero en Electrónica, Automatización y Control

M.Sc. León Vásquez, Rubén Darío.

3 de septiembre del 2020

URKUND**Document Information**

Analyzed document ESCRITO_ASIMBAYA_SUAREZ.docx (D78523366)
Submitted 9/3/2020 5:24:00 AM
Submitted by León Vásquez Rubén Darío
Submitter email rdleon@espe.edu.ec
Similarity 0%
Analysis address rdleon_espe@analysis.orkund.com

Sources included in the report

RUBÉN D. LEÓN V.
1801654284



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

CARRERAS DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES Y CARRERA DE INGENIERÍA EN
ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**Desarrollo de un prototipo de un sismógrafo de alerta temprana para escuelas públicas que incluya el sistema de comunicación inalámbrica Wi-Fi y ubicación GPS**” fue realizado por los señores **Asimbaya García Melissa Noemí** y **Suárez Criollo Enver Miguel**, el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 3 de septiembre de 2020

M.Sc. León Vásquez, Rubén Darío.

C. C. 1801654284



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERAS DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES Y CARRERA DE INGENIERÍA EN
ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Asimbaya García Melissa Noemí** con cédula de ciudadanía n° 1718410002 y **Suárez Criollo Enver Miguel** con cédula de ciudadanía n° 1720354552, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Desarrollo de un prototipo de un sismógrafo de alerta temprana para escuelas públicas que incluya el sistema de comunicación inalámbrica Wi-Fi y ubicación GPS”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 3 de septiembre de 2020


.....

Asimbaya García Melissa Noemí

C.C.: 1718410002


.....

Suárez Criollo Enver Miguel

C.C.: 1720354552



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERAS DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES Y CARRERA DE INGENIERÍA EN
ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros **Asimbaya García Melissa Noemí** con cédula de ciudadanía n° 1718410002 y **Suárez Criollo Enver Miguel** con cédula de ciudadanía n° 1720354552, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Desarrollo de un prototipo de un sismógrafo de alerta temprana para escuelas públicas que incluya el sistema de comunicación inalámbrica Wi-Fi y ubicación GPS”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 3 de septiembre de 2020


.....

Asimbaya García Melissa Noemí

C.C.: 1718410002


.....

Suárez Criollo Enver Miguel

C.C.: 1720354552

Dedicatoria

A mi familia, quienes han sido un apoyo grande en el transcurso de mi vida, a mis padres Rocío y Santiago quienes con su apoyo, amor y esfuerzo me han permitido llegar a ser una profesional. A todos mis amigos quienes compartieron conmigo mi vida estudiantil y me dieron su apoyo para no rendirme.

Melissa Noemí Asimbaya García

Dedicatoria

A mis Padres y hermano

Dedicado a ustedes.

Enver Miguel Suárez Criollo

Agradecimientos

Agradezco a todas las personas que me apoyaron para que llegue a cumplir mi sueño de ser una profesional, a los docentes quienes me prepararon académicamente a lo largo de la vida universitaria, a mis amigas, a mi familia, a mi compañero de tesis y a mi tutor el Msc. Rubén León quien nos apoyó a lo largo del proyecto de titulación.

Melissa Noemí Asimbaya García

Agradecimientos

Alvaro C., Adri C., Bryan L., Melissa A. Brenda P.,

Aby G., Joss Gon., Joss Gom., Giovanni P.,

Victor Y., Galo C., Ale E., Ale Q., Michu Cal.,

Michu Cri., Thaly M., Ari R. e Ivo U.

sin ustedes no hubiera sido lo mismo.

Enver Miguel Suárez Criollo

Índice de contenidos

Desarrollo de un prototipo de un sismógrafo de alerta temprana para escuelas públicas que incluya el sistema de comunicación inalámbrica Wi-Fi y ubicación GPS	1
Hoja de resultados de la herramienta Urkund.....	2
Certificación trabajo de titulación.....	3
Autoría de responsabilidad.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimientos	8
Agradecimientos	9
Índice de contenidos.....	10
Índice de tablas	15
Índice de figuras.....	16
Índice de segmentos de códigos.....	21
Resumen	22
Abstract.....	23
Capítulo I Planteamiento del problema de investigación.....	24
Antecedentes	24
Justificación e Importancia	25
Alcance del proyecto.....	26
Objetivos	29
<i>Objetivo general</i>	29
<i>Objetivos específicos</i>	29
Capítulo II Fundamento teórico.....	30
Sismología	30

<i>Sismos</i>	30
<i>Ondas sísmicas</i>	31
<i>Características de un sismo</i>	33
Actividad sísmica en Ecuador.....	34
Principios de un sistema de alerta sísmica temprana.....	35
Instrumentación para medición sísmica	37
<i>Sismógrafo</i>	37
<i>Acelerógrafo</i>	39
Microcontroladores	40
<i>Protocolo Inter Integrated Circuit</i>	41
<i>Protocolo Serial Peripheral Interface</i>	42
Protocolo Message Queue Telemetry Transport	43
Módulos Electrónicos	45
<i>Módulo acelerómetro 6050</i>	46
<i>Módulo Micro Secure Digital</i>	47
<i>Módulo de Posicionamiento Global</i>	48
<i>Esp wroom 32</i>	50
Interfaces de Programación.....	52
<i>Entorno de desarrollo integrado Arduino</i>	52
<i>Django</i>	53
<i>Virtualenv</i>	55
<i>VisualCode</i>	55
<i>Pycharm</i>	55
<i>Matplotlib</i>	55
<i>Mysql</i>	56
<i>Base de datos</i>	57

Lenguajes de programación.....	58
<i>C++</i>	58
<i>Javascript</i>	59
<i>Python</i>	59
Capítulo III Desarrollo del hardware del prototipo de sismógrafo para el sistema de monitoreo de sismos.....	61
Hardware del sistema	61
<i>Estructura del Hardware</i>	61
<i>Librerías de los módulos</i>	64
Wire.	64
Time.	65
Tinygps++.	66
File System, Secure Data, Serial Peripheral Interface.	67
Algoritmo detección de eventos sísmicos sta/lta.....	67
<i>Niveles de Umbral</i>	69
<i>Parámetros previos y posteriores al evento</i>	69
<i>Procesamiento del algoritmo sta/lta</i>	69
<i>Desarrollo del algoritmo para la detección de ondas sísmicas</i>	71
Librerías para la comunicación Mqtt y mensajería en formato Json.....	73
<i>Arduinojson</i>	73
<i>Wi-Fi</i>	74
<i>Pubsubclient</i>	74
<i>Desarrollo del código para la comunicación con el servidor</i>	75
Almacenamientos de datos.....	80
Diseño de la placa electrónica.....	81
Capítulo IV Desarrollo del software del sistema de monitoreo de sismos.....	85

Implementación del servidor	85
Servicios alojados en el servidor	86
<i>Implementación broker Mqtt</i>	87
<i>Implementación de node red</i>	89
Adquisición de paquetes de datos por el broker y node red	90
Desarrollo del algoritmo para detectar falsos sismos	91
Automatización de datos.....	94
Implementación de modelos de django para la base de datos	98
<i>Base de datos y su relación con django</i>	100
<i>Implementación de las views en django</i>	100
<i>Organización de las urls en django</i>	101
<i>Implementación de la template base del proyecto en django</i>	102
<i>Incorporación de la librería matplotlib en django</i>	103
<i>Preprocesamiento de datos en django</i>	104
Python y jupyter para la depuración de algoritmos.....	106
Algoritmos para procesar datos de eventos telúricos de base datos conocidas.	106
Estructura de datos json para datos enviados por la estación de monitoreo	108
Filtrado de datos del sismo	109
Algoritmo para detectar la onda primaria.....	117
Algoritmo para detectar y localizar la onda secundaria.....	122
Algoritmo para calcular el tiempo de arribo	130
Algoritmo para cálculos de trilateración	131
Desarrollo del algoritmo de cálculo de magnitud de un sismo.....	134
Implementación de visualización de mapas.....	136
Capítulo V Pruebas y resultados	141

Estación de monitoreo sísmico.....	141
<i>Estabilidad y consumo de energía de la estación de monitoreo sísmico</i>	141
<i>Tiempo de funcionalidad con respaldo de batería</i>	142
<i>Sensibilidad del acelerómetro mpu6050</i>	144
Pruebas de ruido.	144
Pruebas de la estación sismográfica.....	147
<i>Algoritmo sta/lta</i>	147
Servidor web.....	150
<i>Pruebas servidor y página web</i>	150
<i>Instalación de la estación</i>	159
<i>Pruebas de señal Wi-Fi en el lugar de la instalación</i>	160
Pruebas del sistema total de monitoreo sísmico	162
Conclusiones.....	164
Recomendaciones	166
Referencias bibliográficas	168

Índice de tablas

Tabla 1	<i>Características del Módulo MPU6050.....</i>	47
Tabla 2	<i>Características del Módulo GY-GPSMV2.....</i>	50
Tabla 3	<i>Características de la placa de desarrollo ESP32 NodeMcu32.....</i>	52
Tabla 4	<i>Funciones de la librería Wire para el acelerómetro MPU6050.....</i>	64
Tabla 5	<i>Funciones de la librería Time para el GPS-GPS6MV2.....</i>	66
Tabla 6	<i>Funciones de la librería TinyGPS++ para el GPS-GPS6MV2.....</i>	66
Tabla 7	<i>Funciones de las librerías FS, SD y SPI para el módulo Micro SD.....</i>	67
Tabla 8	<i>Valores del Algoritmo STA/LTA.....</i>	72
Tabla 9	<i>Comando de la librería ArduinoJson.....</i>	74
Tabla 10	<i>Función para protocolo MQTT.....</i>	74
Tabla 11	<i>Pruebas de ruido del sensor MPU6050 eje X.....</i>	145
Tabla 12	<i>Pruebas de ruido del sensor MPU6050 eje Y.....</i>	146
Tabla 13	<i>Pruebas de ruido del sensor MPU6050 eje Z.....</i>	147
Tabla 14	<i>Comparación de datos obtenidos.....</i>	158
Tabla 15	<i>Comparación de datos obtenidos por el sistema.....</i>	159
Tabla 16	<i>Datos del eje Z obtenidos.....</i>	162

Índice de figuras

Figura 1 <i>Esquema del Prototipo del Sistema de Monitoreo Sísmico</i>	28
Figura 2 <i>Descripción del origen de un sismo</i>	31
Figura 3 <i>Clasificación de ondas Sísmicas</i>	33
Figura 4 <i>Principios básicos del sistema de alerta sísmica</i>	37
Figura 5 <i>Esquema de elementos de un sismógrafo</i>	38
Figura 6 <i>Distribución de estaciones en la Red de Acelerógrafos en el Ecuador</i>	39
Figura 7 <i>Componentes de un Microcontrolador</i>	40
Figura 8 <i>Comunicación I2C</i>	41
Figura 9 <i>Comunicación SPI</i>	43
Figura 10 <i>Elementos de una comunicación por MQTT</i>	44
Figura 11 <i>Diagrama de las Etapas de comunicación con el protocolo MQTT</i>	45
Figura 12 <i>Distribución de pines y orientación del MPU6050</i>	46
Figura 13 <i>Módulo GY-521</i>	47
Figura 14 <i>Módulo SD Card</i>	48
Figura 15 <i>Neo-6 u-blox 6 GPS</i>	49
Figura 16 <i>Módulo GY-GPS6MV2</i>	49
Figura 17 <i>Distribución de pines del ESP WROOM 32</i>	51
Figura 18 <i>Placa de desarrollo ESP32 NodeMcu-32</i>	51
Figura 19 <i>Interfaz de Arduino IDE</i>	53
Figura 20 <i>Esquema Django</i>	54
Figura 21 <i>Librería Matplotlib</i>	56
Figura 22 <i>Diagrama de bloques prototipo de sismógrafo</i>	61
Figura 23 <i>Diagrama de flujo del Hardware</i>	63

Figura 24 <i>Ejemplo del funcionamiento del Algoritmo de disparo STA/LTA.</i>	70
Figura 25 <i>Ejemplo archivo.txt</i>	81
Figura 26 <i>Esquemático del circuito del Prototipo</i>	82
Figura 27 <i>PCB del prototipo de Sismógrafo.</i>	83
Figura 28 <i>Servidor en ejecución Nginx.</i>	86
Figura 29 <i>Conjuntos de servicios del servidor.</i>	87
Figura 30 <i>Ejecución del broker Mosquito.</i>	88
Figura 31 <i>Características implementadas en Node-Red.</i>	89
Figura 32 <i>Diagrama de flujo del algoritmo de detección de falsos sismos.</i>	94
Figura 33 <i>Adquisición y alojamiento de datos en la memoria del host con Node-Red.</i>	95
Figura 34 <i>Diagrama de flujo del manejo JSON.</i>	97
Figura 35 <i>Diferencia entre el SQL y Django ORM.</i>	98
Figura 36 <i>Models usados para organizar la base de datos.</i>	99
Figura 37 <i>URLs, Views y Template HTML.</i>	102
Figura 38 <i>Aceleración vs Tiempo.</i>	105
Figura 39 <i>Entorno gráfico de Jupyter.</i>	106
Figura 40 <i>Formato del archivo plano .txt de las estación ALIB.</i>	107
Figura 41 <i>Formato del archivo .csv.</i>	108
Figura 42 <i>Estructura JSON del acelerómetro.</i>	109
Figura 43 <i>Algoritmo de tratamiento de los datos.</i>	110
Figura 44 <i>Señal original del sismo del 7 de septiembre del 2018.</i>	111
Figura 45 <i>Espectrograma del sismo del 7 de septiembre del 2018.</i>	112
Figura 46 <i>Señal del sismo y filtración.</i>	113
Figura 47 <i>Espectrograma y filtros.</i>	113

Figura 48 <i>Normalización de la señal.</i>	114
Figura 49 <i>Algoritmo de normalización.</i>	114
Figura 50 <i>Transformada de Hilbert resultante.</i>	115
Figura 51 <i>Envolvente de la señal sísmica.</i>	116
Figura 52 <i>Algoritmo para la detección de la onda P.</i>	118
Figura 53 <i>Resultado de la primera derivada de la envolvente.</i>	119
Figura 54 <i>Segunda derivada de la envolvente.</i>	119
Figura 55 <i>Comparativo de diferentes envolventes.</i>	120
Figura 56 <i>Análisis de la envolvente.</i>	121
Figura 57 <i>Algoritmo para la detección del disparo.</i>	123
Figura 58 <i>Algoritmo para la detección de la onda S.</i>	124
Figura 59 <i>Espectrograma de una señal sísmica.</i>	125
Figura 60 <i>Espectrograma de una señal sísmica filtrada.</i>	125
Figura 61 <i>Densidad espectral promedio de los datos del IGEPN.</i>	126
Figura 62 <i>Umbral de disparo de la onda S.</i>	127
Figura 63 <i>Densidad espectral promedio.</i>	128
Figura 64 <i>Densidad espectral promedio y localización de ondas S.</i>	128
Figura 65 <i>Localización de onda S.</i>	129
Figura 66 <i>Identificación de las onda de cuerpo P y S.</i>	130
Figura 67 <i>Diagrama de flujo para calcular ToA.</i>	131
Figura 68 <i>Cálculos de trilateración.</i>	132
Figura 69 <i>Ubicación geográfica del epicentro.</i>	134
Figura 70 <i>Valores de las amplitudes.</i>	135
Figura 71 <i>Aceleraciones registradas.</i>	136

Figura 72 <i>Vista de la página principal</i>	137
Figura 73 <i>Capa de relieves</i>	138
Figura 74 <i>Diagrama de bloques de la alimentación de la estación</i>	142
Figura 75 <i>Medición del consumo en amperios a la entrada del transformador</i>	142
Figura 76 <i>Medición del consumo en amperios</i>	143
Figura 77 <i>Comportamiento de la batería</i>	143
Figura 78 <i>Datos de ruido pertenecientes</i>	145
Figura 79 <i>Datos de ruido pertenecientes al eje Y</i>	146
Figura 80 <i>Datos de ruido pertenecientes al eje Z</i>	147
Figura 81 <i>Comportamiento de los valores STA/LTA</i>	148
Figura 82 <i>Análisis del algoritmo STA/LTA</i>	149
Figura 83 <i>Inyección de datos hacia la base de datos</i>	151
Figura 84 <i>Datos almacenados en la base de datos MySQL</i>	151
Figura 85 <i>Página de inicio de la página web</i>	152
Figura 86 <i>Página del historial de eventos sísmicos</i>	153
Figura 87 <i>Página de información detallada</i>	153
Figura 88 <i>Datos generada para el evento</i>	154
Figura 89 <i>Ubicación geográfica de estaciones</i>	155
Figura 90 <i>Componentes Z, N y E</i>	156
Figura 91 <i>Cálculo del TOA y detección de la onda P y S</i>	157
Figura 92 <i>Epicentro de un evento registrado</i>	157
Figura 93 <i>Estación 1 instalada</i>	160
Figura 94 <i>Estación 2 instalada</i>	160
Figura 95 <i>Resultados presentados por la app N-PERF</i>	161

Figura 96 <i>Historial de los registros</i>	163
Figura 97 <i>Historial de datos almacenados</i>	163

Índice de segmentos de códigos

Segmento de código 1	<i>Función read_data().</i>	71
Segmento de código 2	<i>Programación del algoritmo STA/LTA en el eje Z.</i>	73
Segmento de código 3	<i>Función Alerta_Evento().</i>	76
Segmento de código 4	<i>Función callback().</i>	77
Segmento de código 5	<i>Función Callback().</i>	78
Segmento de código 6	<i>Función JSON_Z envío de datos de la componente Z.</i>	79
Segmento de código 7	<i>Función Registrar_Evento().</i>	80
Segmento de código 8	<i>Función Fragmentador.</i>	96
Segmento de código 9	<i>Almacenamiento de datos de las estaciones en la DB.</i>	97
Segmento de código 10	<i>Envío de datos cuando al cliente.</i>	100
Segmento de código 11	<i>Función "estacionListar".</i>	101
Segmento de código 12	<i>URLs de Django.</i>	102
Segmento de código 13	<i>Librería matplotlib.</i>	104
Segmento de código 14	<i>Codificación y almacenamiento.</i>	104
Segmento de código 15	<i>Parámetros del mapa.</i>	138
Segmento de código 16	<i>Sentencias Javascript.</i>	139
Segmento de código 17	<i>Incorporación de figuras geométricas.</i>	140

Resumen

El siguiente proyecto presenta el desarrollo e implementación de un prototipo de sismógrafo con comunicación Wi-Fi hacia un servidor web para el monitoreo de eventos sísmicos mediante una página web. El prototipo se divide en dos etapas, Hardware y Software. El Hardware representa la estación sismográfica, que cuenta con varios módulos electrónicos interconectados entre sí para así realizar la detección de señales sísmicas, entre ellos un módulo GPS para su ubicación geográfica, una lectora de SD Card para el almacenamiento de datos, un acelerómetro para la detección de movimientos y una tarjeta ESP32 para la ejecución mediante código del algoritmo STA/LTA, para la detección de eventos sísmicos mediante umbrales, gracias a las características de la tarjeta se realiza la comunicación Wi-Fi con un servidor Web, al cual se enviará la alerta de un evento sísmico, posteriormente los datos correspondientes a la señal detectada para su registro y almacenamiento. Al servidor web se lo ha denominado Software del sistema, este se encargará de almacenar los datos de la señal sísmica detectada por la estación, para su posterior procesamiento mediante cálculos para obtener valores aproximados de la magnitud y profundidad, para finalmente mostrar mediante una página web la ubicación sobre un mapa de la o las estaciones, las gráficas de la señal sísmica y mediante tablas la información del evento registrado, para lograr las características del servidor se usó lenguajes de programación como JavaScript, Django y Python. Finalmente se realiza la instalación de dos estaciones sismográficas en dos sedes de la universidad para sus respectivas pruebas.

Palabras claves:

- **SISMÓGRAFO**
- **ACELERÓMETRO**
- **SERVIDOR WEB**

Abstract

The following project presents the development and implementation of a seismograph prototype with Wi-Fi communication to a web server for monitoring seismic events through a web page. The prototype is divided into two stages, Hardware and Software. The Hardware represents the seismographic station, which has several electronic modules interconnected to perform the detection of seismic signals, including a GPS module for its geographical location, an SD Card reader for data storage, an accelerometer for the movement detection and an ESP32 card for the execution by means of the STA / LTA algorithm code, for the detection of seismic events through thresholds, thanks to the characteristics of the card, Wi-Fi communication is carried out with a Web server, to which it will be sent the alert of a seismic event, subsequently the data corresponding to the detected signal for its registration and storage in a database. The web server has been called System Software, it will be in charge of storing the data of the seismic signal detected by the station, for its subsequent processing through calculations to obtain approximate values of the magnitude and depth, to finally display through a web page the location on a map of the station (s), the graphs of the seismic signal and through tables the information of the registered event. To achieve the characteristics of the server, programming languages such as JavaScript and Python were used. Finally, the installation of two seismographic stations is carried out in two university campuses for their respective tests.

Key words:

- **SEISMOGRAPH**
- **ACCELEROMETER**
- **WEB SERVER**

Capítulo I

Planteamiento del problema de investigación

Antecedentes

Los eventos sísmicos se han convertido en los últimos años en uno de los fenómenos naturales destructivos más frecuentes, y a pesar de grandes esfuerzos de investigación realizados por la comunidad sismológica mundial durante años, aún no se ha logrado avanzar en el estudio de la predicción de terremotos (INPRES, s.f.).

Mayormente cada país cuenta con un organismo que administra una red de sismógrafos que registra y estudia la actividad sísmica regional. El cual tiene como propósito principal la protección de su población y la reducción de daños provocados por un evento sísmico.

Los organismos mas importantes dedicados a la actividad sísmica en América Latina son el Instituto Nacional de Prevención Sísmica (INPRES) en Argentina (INPRES, s.f.), el Centro Sismológico Nacional en Chile (CSN, s.f.) . El Instituto Geofísico de la Escuela Politécnica Nacional (IGEPN) es una de las instituciones más importantes en el Ecuador, dedicada al estudio de los fenómenos sísmicos, a través del RENAC o Red Nacional de sismógrafos lleva a cabo en forma permanente, el control de la actividad sísmica y volcánica del país (IGEPN, s.f.).

Ecuador se encuentra en el denominado Cinturón de Fuego del Pacífico, debido a su ubicación su actividad sísmica es alta, por lo que ha presenciado varios eventos sísmicos entre los cuales se destaca el de 1906, que alcanzó una magnitud de 8.8, siendo este el de mayor magnitud registrado en la historia. El último registrado con una magnitud de 7.8, el 16 de abril del 2016 el cual causó daños significativos tanto materiales como pérdidas de vidas humanas en diferentes partes del país, esto dejó en evidencia ciertos problemas relacionados al diseño sismorresistente en el Ecuador (Martínez, 2016).

En la actualidad se presenta la necesidad del desarrollo de nuevos dispositivos enfocados a la detección de actividad sísmica, principalmente que sean de bajo costo, entre los cuales se puede citar la investigación “Sistema Distribuido de Detección de Sismos Usando una Red de Sensores Inalámbrica para Alerta Temprana”, esta red utiliza sensores COTS (Commercial Off-The-Shelf) los cuales son sensores que se utilizan generalmente en los teléfonos inteligentes ya que se manejan a través de aplicaciones de bajo consumo energético, dichos sensores tienen la capacidad de detectar, analizar y notificar un evento sísmico (Zambrano Vizúete, 2015). Por lo que este sería un estudio clave para la creación de un sismógrafo por medio de acelerómetros de bajo costo que se ofertan en el mercado. Al cual se le pueda adecuar a los requerimientos que presenta la detección de actividad sísmica, así, logrando un avance tecnológico y aportando a la investigación para que más instituciones puedan unirse al estudio de dichos eventos.

Justificación e Importancia

Los fenómenos geológicos que tienen mayor representación y que generan mayor efectos negativos a la población son los sismos y las erupciones volcánicas. El gobierno de nuestro país bajo el cumplimiento de la constitución, debe proteger a la población y disminuir los daños ocasionados por desastres naturales entre ellos los movimientos sísmicos, para lo cual se ha visto en la necesidad de conformar organizaciones cuyo objetivo es monitorear dichos fenómenos y recopilar información, para realizar estudios del comportamiento de estos eventos , a través de la implementación y administración de redes conformadas por acelerómetros y la integración de sistemas de alerta temprana (Moncayo Theurer, 2017).

Si bien en el mercado mundial existen instrumentos apropiados y de alta sensibilidad dedicados a la detección de eventos sísmicos, estos representan una alta inversión, por lo que es de interés disponer de un instrumento moderno, de diseño propio y bajo costo que contribuya en la detección de dichos eventos (Quinde Martínez, 2016). En la actualidad los

sensores que proporcionan información de los movimientos sísmicos son los acelerómetros, su costo se ve directamente reflejado en la precisión de la información que entrega. Por lo que al combinar los acelerómetros con microcontroladores y conectividad a internet, se podría tener un instrumento completo el cual pueda llegar a tener características parecidas al de un sismógrafo (Jaramillo Echeverría, 1994).

Las unidades educativas y universidades públicas se han tratado de ubicar en las denominadas zonas seguras del país, sin embargo, la experiencia establece que se está en exposición permanente a actividad sísmica. Es por esto que es necesario monitorear los sismos, con una red propia de dispositivos que podrían ser instalados en todas las edificaciones de alto interés, y por medio de internet revisar las distintas fluctuaciones de movimientos telúricos producidos en las zonas (Moncayo Theurer, 2017).

En el Ecuador se construyó las denominadas unidades educativas del milenio las cuales cuentan con internet de alta velocidad por fibra óptica. Aprovechando la infraestructura de dichas unidades educativas, se propone un prototipo para desarrollar e instalar un sistema sismográfico (integrado al internet vía WIFI), para uso exclusivo de las unidades educativas.

Alcance del proyecto

La finalidad del trabajo de grado presentado, es el diseño y la construcción de un prototipo de sismógrafo el cual desempeñe principalmente los siguientes roles: monitorear, detectar y registrar la actividad sísmica, de una ubicación geográfica determinada. Dicho dispositivo tiene como finalidad formar una red de sensores inalámbricos para el monitoreo sísmico en las unidades educativas del país, al cual lo denominaremos estación sismográfica.

La estación sismográfica estará formada tanto por hardware como por software, el hardware contará con módulos electrónicos que conectados entre sí nos permitan el monitoreo

y la detección de la actividad sísmica, también se integrará un módulo que permita la conexión hacia la red mediante Wi-Fi.

El software será el medio por el cual se le dará la inteligencia suficiente al sismógrafo para que desempeñe sus roles de manera eficaz, por medio de la programación de algoritmos destinados a dicho propósito. Además del desarrollo de un servidor el cual almacene y procese los datos registrados por la estación sismográfica. También se creará una plataforma web para la visualización de los eventos registrados.

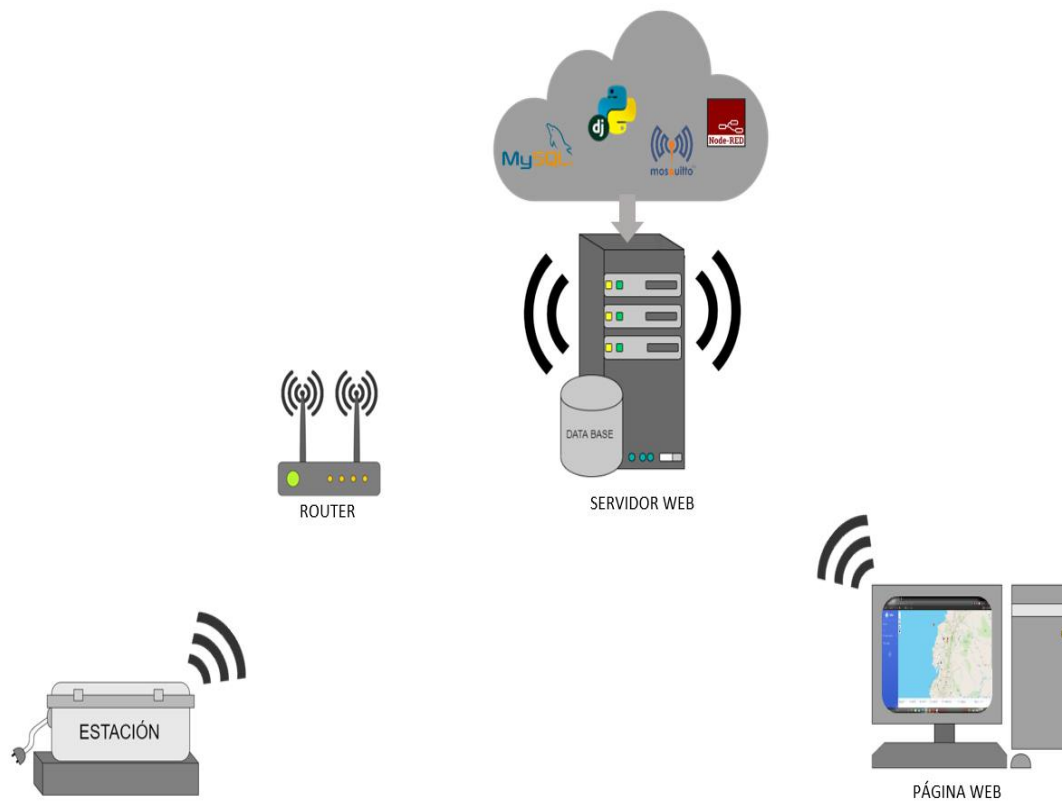
La comunicación entre la estación sismográfica y la plataforma web se realizará de forma inalámbrica mediante Wi-Fi. Utilizando este medio como base para la implementación de protocolos de comunicación utilizados en IoT (Internet de las Cosas). El uso de esta tecnología permitirá una comunicación confiable y segura, con estándares específicos para redes de sensores. Esto permitirá la transferencia de información entre la estación y el servidor.

La plataforma web será diseñada de tal forma que sea capaz de mostrar la información adquirida y procesada para una fácil comprensión por parte del usuario. En dicha plataforma se desplegará gráficas de ubicación de la estación sismográfica sobre un mapa, gráficas de la señal sísmica obtenida tanto en las componentes Norte (N), Este (E) y Z, una tabla con información de los valores de cada componente, también se presentará los valores de la profundidad y magnitud del evento sísmico, incluso se podrá observar una ubicación aproximada del epicentro y el tiempo de arribo del evento del sismo hacia la estación sismográfica.

Una vez finalizado el prototipo, se realizará pruebas del funcionamiento de la estación sismográfica en las instalaciones de la Universidad de las Fuerzas Armadas "ESPE", donde se ubicará un mojón el cual es una esta estructura de concreto destinada para la instalación de este tipo de estaciones, dicho mojón contará con información georreferenciada, permitiendo realizar pruebas con una reducción del error de la ubicación..

Figura 1

Esquema del Prototipo del Sistema de Monitoreo Sísmico.



Objetivos

Objetivo general

- Desarrollo de un prototipo de un sismógrafo de alerta temprana para las escuelas públicas que incluya el sistema de comunicación inalámbrica WIFI y ubicación GPS.

Objetivos específicos

- Desarrollar un prototipo del sismógrafo con comunicación inalámbrica WIFI y ubicación GPS que permita detectar movimientos telúricos y almacenar los datos obtenidos.
- Implementar la comunicación inalámbrica entre el sismógrafo y un servidor por medio de conexión a internet.
- Analizar y procesar los datos obtenidos del sismógrafo para disponer de una primera estimativa de la profundidad y magnitud del evento.
- Desarrollar una página web para presentar la información, tal como un mapa de la ubicación geográfica del sismógrafo, valores estimados de la profundidad y magnitud.

Capítulo II

Fundamento teórico

Sismología

La sismología es la disciplina dedicada al estudio de los terremotos y la propagación de ondas sísmicas en las capas interiores de la tierra. Es una rama de la Geofísica la cual ha proporcionado gran información sobre la composición y el estado dentro de la tierra. Las investigaciones sismológicas son de manera local o regional, dentro de sus estudios esta el determinar fallas subterráneas del interior de la Tierra. En los últimos tiempos esta rama se ha dedicado a la predicción de terremotos y, más exitosamente, a evaluar los riesgos sísmicos en diferentes ubicaciones geográficas. Los sismólogos además se han dedicado al estudio de las vibraciones del suelo provocadas por la actividad humana tal como la minería, el objetivo de estas investigaciones es la discriminación entre explosiones y terremotos naturales (Enciclopedia Británica, s.f.).

Sismos

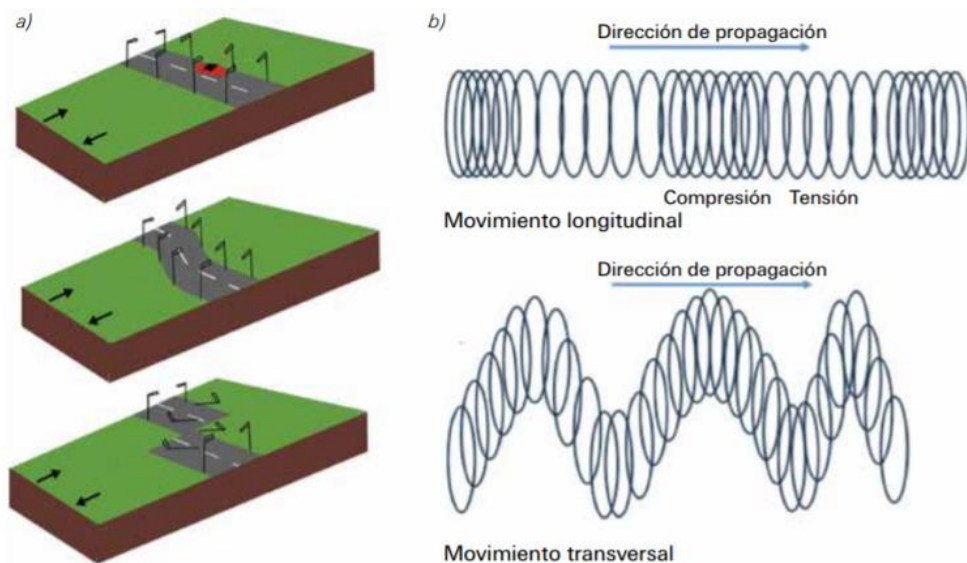
El estudio de la tectónica de placas ayuda a comprender el desplazamiento relativo que existe entre ellas, además las deformaciones y fuerzas de fricción que se originan en la frontera de la corteza. Dichos desplazamientos y deformaciones son los que provocan los movimientos súbitos que se transmiten en forma de onda hacia la superficie y comúnmente se lo conoce como sismo. Por lo tanto los movimiento telúricos o sismos se originan por el movimiento, fricción y deformación de las placas tectónicas.

El desplazamiento relativo es equivalente a un resorte el cual va acumulando energía potencial más bien conocida como energía sísmica, dicha energía se acumulará hasta llegar al límite elástico que posea el medio, entonces una vez superado el límite de dicho medio se

romperá y provocará que se desplace súbitamente originando un sismo (Espindola Castro, 2018).

Figura 2

Descripción del origen de un sismo.



Nota. Tomado de *¿Qué son los SISMOS, dónde ocurren y cómo se miden?* (p.11), por V.Espinola Castro, X. Pérez Campos, 2018, Ciencia.

Ondas sísmicas

La tierra es considerada un objeto con características elásticas, siendo así un medio adecuado para la propagación de ondas sísmicas. Las perturbaciones provocadas por un movimiento telúrico en cualquier lugar de la Tierra origina ondas energéticas o comúnmente llamadas ondas sísmicas. Las ondas sísmicas se clasifican dentro de las ondas con características elásticas ya que tienen la capacidad de propagarse por un medio, en este caso dentro de la Tierra. Las ondas sísmicas se clasifican en dos grupos, en ondas de cuerpo y ondas superficiales (Zúñiga Dávila, 2011).

Las ondas de cuerpo son aquellas que se expanden desde su origen hasta llegar al exterior de la Tierra y se clasifican en:

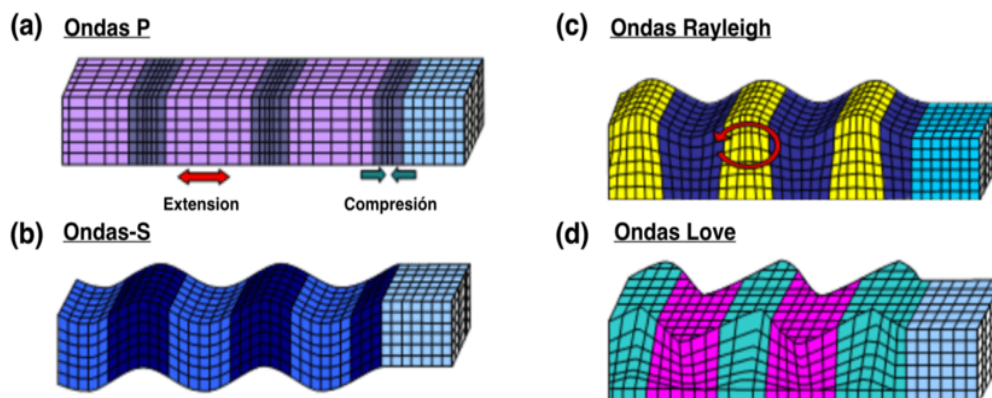
- **Ondas Primarias o P** : son las primeras ondas en salir hacia la superficie de la Tierra. Su velocidad de propagación es de casi unos 7.5 km/s, esta velocidad cambia dependiendo del medio por el cual se propague. Las ondas primarias producen oscilaciones en el medio en el mismo sentido con el cual estas se están propagando.
- **Ondas Secundarias o S** : también conocida como ondas transversales, se las nombra secundarias ya que estas ondas llegan después de las ondas primarias, tienen una velocidad de propagación aproximada de 4.2 km/s, esta velocidad de igual forma puede variar dependiendo del material en el que se propaguen. Estas ondas tienen una dirección perpendicular de propagación.

Las ondas superficiales son aquellas que se extienden a través de la superficie de la Tierra, estas ondas llegan a obtener su máxima amplitud en una superficie libre y se clasifican en:

- **Ondas Rayleigt**: estas ondas describen una trayectoria elíptica retrógrada, estas ondas son similares a las ondas que se producen en una superficie de agua.
- **Ondas Love**: son ondas que poseen un movimiento horizontal sobre la Tierra.

Figura 3

Clasificación de ondas Sísmicas.



Nota. El gráfico muestra los tipos de ondas sísmicas presentes en una señal sísmica. Tomado de *Correlación cruzada de ruido sísmico para la obtención de perfiles profundos de velocidad de onda de corte en la cuenca de Santiago* (p.14), por M. Sáez, 2016

Características de un sismo

En un sismo las características principales que se obtienen son las descritas a continuación (Chile Sismología, s.f.):

- **Hipocentro y Epicentro:** Al hipocentro se lo puede describir como el punto interior donde se origina una separación de las placas interiores de la Tierra originando un sismo. El epicentro es un punto en la superficie exterior de la tierra ubicado de manera vertical sobre el Hipocentro.
- **Escala:** es la relación que existe entre la magnitud medida en el plano y la homóloga del terreno, esta relación es variable de un plano a otro.
- **Latitud y Longitud:** es un sistema de referencia para la localización de un punto en la Tierra. La latitud nos proporciona la ubicación geográfica de un lugar al norte a al sur. Mientras que la longitud es la localización geográfica al este o al oeste. Tanto la

latitud como la longitud se miden en ángulos que van desde 0° dentro del meridiano de origen hasta 180° de la línea internacional de cambio de fecha.

- **Magnitud:** es una medida que representa la cantidad de energía liberada en forma de ondas. Permite calcular el tamaño relativo de un temblor a través del logaritmo en base 10 de la amplitud máxima ya sea de la onda P (primaria) o S (secundaria).

Actividad sísmica en Ecuador

El Ecuador está conformado por una región continental que se extiende desde 81° W hasta 75° W en longitud y desde 1.25° N hasta 5° S en latitud aproximadamente, además de las islas Galápagos. El territorio ecuatoriano está conformado por 4 regiones Costa, Sierra, Oriente y Región Insular, de las cuatro, la región Sierra es la que presenta mayor cantidad de fallas en su superficie debido a la geodinámica de la región. Nuestro país por su ubicación geográfica se encuentra en movimiento telúrico constante. Debido a que una parte del territorio pertenece a la microplaca “Bloque andino”, dicha placa a su vez pertenece a la placa sudamericana, la cual está en una interacción con las placas Nazca, Coco y Caribe. En esta zona el movimiento está en 50 y 60 mm/año. Además una parte de la región sur del país mantiene interacción entre la placa Sudamericana y la placa Nazca. Por lo tanto debido a estas fallas el país es una zona propensa a sismos (Quinde Martínez, 2016).

Estudios muestran que en la provincia Pichincha la ciudad de Quito localizada en la región interandina se ha producido cada 50 años actividad sísmica con epicentros localizados a diferentes distancias de la ciudad (Valverde, 2002). Las provincias más propensas a presentar actividad sísmica son las provincias de Esmeraldas y Manabí como lo menciona la tesis doctoral perteneciente a la Universidad Politécnica de Madrid realizada en Quito, denominada “Desarrollos Metodológicos y Aplicaciones hacia el cálculo de la Peligrosidad Sísmica en el

Ecuador Continental y Estudio de Riesgos Sísmicos en la Ciudad de Quito” (Parra Cárdenas, 2016).

Dichas afirmaciones se pueden evidenciar con los eventos ocurridos en el país en los últimos años tal como el terremoto del 16 de abril del 2016 que causó daños en las provincias de Esmeraldas y Manabí dejando un registro de 663 personas muertas y 28775 personas sin hogar (INEC, 2017).

En el Ecuador existen instituciones dedicadas al estudio de la actividad sísmica, entre ellas las más importantes son el Instituto Geofísico de la Escuela Politécnica Nacional - IGEPN y la Red Nacional de Sismógrafos del Ecuador (RENSIG) (Quinde Martínez, 2016), estas instituciones son las encargadas del monitoreo continuo de actividad sísmica en el país.

Principios de un sistema de alerta sísmica temprana.

Un terremoto (del latín *terraemōtus*, a partir de *terra* «tierra», y *motus* «movimiento»), también conocido como sismo y temblor, aunque el primer término está relacionado para referirse a sismo de gran magnitud, coloquialmente la palabra sismo es más utilizada.

Sismo es un fuerte movimiento telúrico que está relacionado con la liberación de grandes cantidades de energía en las capas internas del planeta y su propagación es en forma de ondas en todas direcciones. Tienen como características ser de corta duración e intensidad variable en función de los materiales de las capas terrestres. Es un fenómeno impredecible y de capacidad destructiva de gran potencial, pudiendo generar daño a la población y pérdidas materiales.

En la actualidad la predicción de sismos es difícil, como es de naturaleza estocástica, además de ser un fenómeno impredecible ya que libera acumulación de energía en cualquier lugar provocando así el origen de un nuevo sismo. Los eventos sísmicos tienen una cualidad de

repetirse en el mismo lugar geográfico, y mientras más tiempo pase desde el último evento, mayor serán las probabilidades de que se produzca.

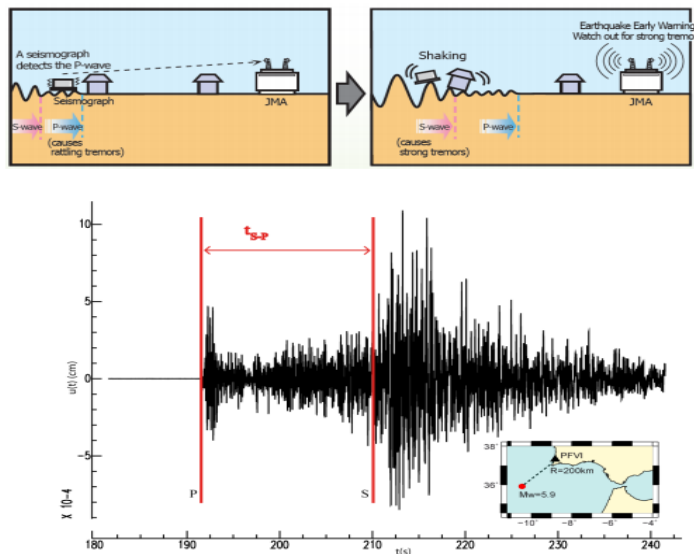
Los sistemas de alerta temprana de terremotos están fuertemente desarrollados en los Estados Unidos y Japón, porque disponen de regiones con alta actividad sísmica y de ámbito delicado para su población, pero aun teniendo alta tecnología no pueden predecir con exactitud los acontecimientos, pero en base al desarrollo del análisis de datos recolectados han podido determinar técnicas para determinar cuando un evento se está ejecutando (Cooper, 1868).

En el estado Nipón en el año 1982 se desarrolló el primer EEWS (Earthquake Early Warning System), que se basaba en las conexiones de los puntos de monitoreo por las líneas de tren (Saita, 2003). Un sistema de características similares se incorpora en México usando la misma metodología, aunque no es hasta el año 1992 que se desarrolla un sistema que se basaba en la detección de la onda primaria también denominada onda P, para proteger las líneas ferroviarias de Japón.

En los sistemas de monitoreo donde su información viaja por redes o señales electromagnéticas, los datos de la onda P viajarán a mayor velocidad que los de la onda S, por lo tanto si al detectar la onda P podemos generar una alerta con anticipación a la llegada de las ondas superficiales, obtendremos una alerta que brinda la oportunidad de proveer cortes anticipados de electricidad desde las subestaciones, para los trenes y acudir a un lugar seguro.

Figura 4

Principios básicos del sistema de alerta sísmica.



Nota. La ilustración muestra las diferentes etapas de un sistema de alerta temprana sísmica cuando se presenta un evento sísmico. Tomado de: *The role of governments in Hazard mitigation* (p.93), por T. Islam, 2016.

La diferencia de tiempo que existe entre la onda P y la onda S se la denomina tiempo de Arribo, es un dato muy importante para el análisis de distancia al origen del evento y también usado para los cálculos de trilateración.

Instrumentación para medición sísmica

Sismógrafo

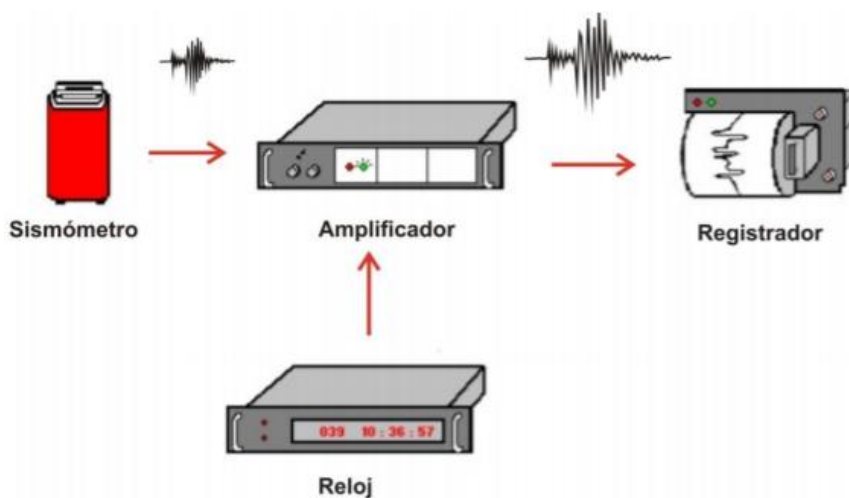
Es un instrumento encargado de registrar los movimientos del suelo provocado por las ondas sísmicas que se propagan en la Tierra en función del tiempo. Dicho instrumento fue creado a finales del siglo pasado pero con el tiempo ha ido mejorando, en la actualidad los sismógrafos se conforman por componentes electrónicos, los cuales han ayudado a mejorar su precisión en la detección de sismos (Espíndola, 1994).

El sismógrafo está conformado de cuatro elementos principales como se detalla a continuación (INPRES, s.f.):

1. Sismómetro: detecta el movimiento del suelo, tanto en una dirección vertical como horizontal, y entrega una señal eléctrica del evento detectado.
2. Amplificador: amplifica la señal eléctrica entregada por el sismómetro para posteriormente ser registrada y analizada.
3. Registrador: grafica la señal amplificada en función del tiempo.
4. Reloj: es un tiempo añadido al sistema para el registro de los datos.

Figura 5

Esquema de elementos de un sismógrafo.



Nota. El gráfico anterior representa los elementos que conforman un sismógrafo utilizado en la mayoría de centros dedicados a la detección de eventos sísmicos. Tomado de: SÍSMICA, INSTITUTO NACIONAL DE PREVENCIÓN SÍSMICA. por INPRES.

Acelerógrafo

Los acelerógrafos son dispositivos enfocados en la medición de la variación de aceleraciones de un lugar determinado. Estos instrumentos están conformados por tres sensores ortogonales, su registro lo realizan en las componentes norte-sur y este- oeste. Gracias a estos instrumentos se ha podido realizar en muchos países una red acelerográfica permanente la cual ayuda en el registro de eventos sísmicos de manera permanente, los datos obtenidos de esta red son procesados y analizados a través de cálculos, para así determinar los valores máximos de la aceleración y su escala de intensidad (IGEPN, s.f.). En el Ecuador existe la Red Nacional de Acelerógrafos RENAC y está distribuida por todo el país.

Figura 6

Distribución de estaciones en la Red de Acelerógrafos en el Ecuador.



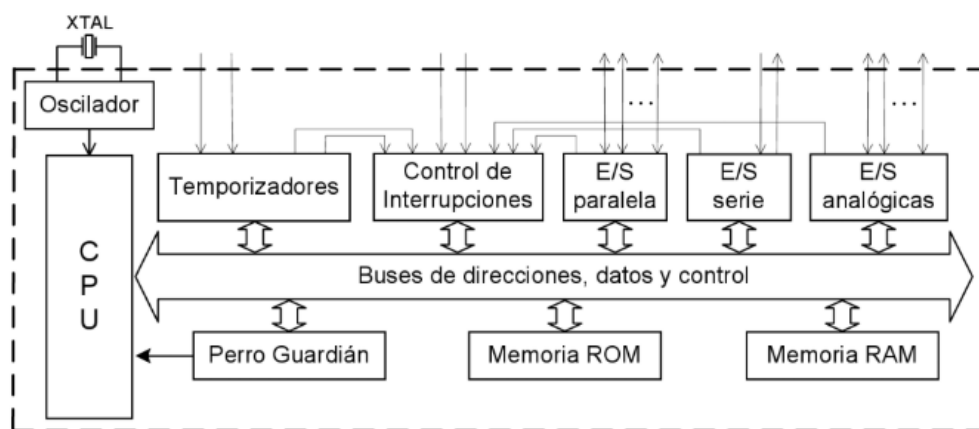
Nota. En la imagen se presenta la distribución en el país de las estaciones de sismógrafos, pertenecientes a la RENAC del IGEPN. Tomado de: Instituto Geofísico de la Politécnica Nacional, por IGEPN.

Microcontroladores

Los microcontroladores son dispositivos que han ayudado a la mayoría de los avances tecnológicos que tenemos hoy en día, estos dispositivos fueron desarrollados para cubrir diversas aplicaciones. Son utilizados en diferentes ramas de la electrónica tales como la automoción, equipos de comunicación y telefonía, equipos médicos, equipos industriales, artefactos del hogar, etc. Dichos dispositivos cumplen el rol de ejecutar un programa almacenado de manera permanente en sus memorias, esta función la pueden realizar gracias a que están compuestos básicamente por un procesador, memorias (RAM y ROM), puertos de salida y entrada, un reloj y un reset. Son chips pequeños de fácil programación y una flexibilidad total (Valdés Pérez, 2007).

Figura 7

Componentes de un Microcontrolador.



Nota. El gráfico representa la estructura interna de un Microcontrolador. *Tomado de Microcontroladores: Fundamentos y Aplicaciones con PIC*, por F. Valdés Pérez, R. Pallás Areny, 2007.

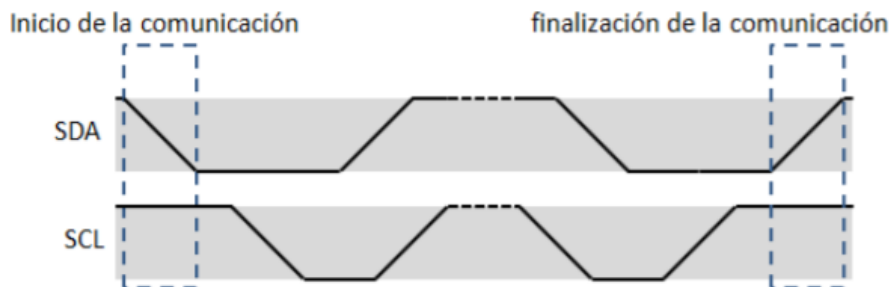
Protocolo Inter Integrated Circuit

I2C es el acrónimo de Inter Integrated Circuit, esta es una interfaz que se deriva del módulo puerto serie síncrono maestro (MSSP), que es utilizado para la comunicación con otros dispositivos programables los cuales poseen los modos de hardware, maestro, modo esclavo y multi-maestro, lo cual les permite realizar transmisiones de datos por medio de tramas. La comunicación I2C se realiza a través de un bus bidireccional que se encuentra conectado por medio de dos líneas hacia la entrada de voltaje por la resistencia “pull-up” (Herrera Zuleta, 2010):

- **Serial clock (SCL):** es la línea de reloj utilizada para la sincronización de datos. Posee dos estado uno de entrada y uno de salida cada una funciona dependiendo si es utilizada ya sea por el esclavo o maestro respectivamente (Herrera Zuleta, 2010).
- **Serial Data (SDA):** es una línea por la cual se transmiten los datos en serie semidireccional, esta línea puede ser manejada tanto por un maestro como por un esclavo (Herrera Zuleta, 2010).

Figura 8

Comunicación I2C.



Nota. El gráfico indica los periodos de tiempo de SDA y SCL de la comunicación I2C. Tomado de *Comunicación I2C Prototipo Brazo Robótico BM001*, por O. Herrera Zuleta, 2010.

La comunicación I2C presenta una velocidad de transmisión de datos variable, esta velocidad varía dependiendo de las características requeridas por el sistema, además de la capacidad de la línea y de la resistencia “pull-up”. Las velocidades estándar de la comunicación I2C son las siguientes:

- Modo estándar: 100 kbit/s
- Modo rápido: 400 kbits/s
- Modo alto 1 Mbits/s
- Alta velocidad: 3.4 Mbits/s

La resistencia “pull-up” está determinada por la alimentación del bus, velocidad de transmisión y cantidad de dispositivos límites que se pueden conectar (Herrera Zuleta, 2010).

Protocolo Serial Peripheral Interface

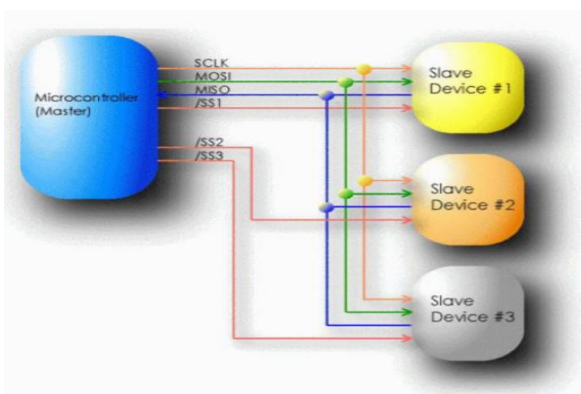
El protocolo Serial Peripheral Interface o SPI es un bus que transmite paquetes de 8 bits por medio de 3 líneas de transmisión. Se destaca porque todos los dispositivos conectados a este bus pueden ser transmisor y receptor al mismo tiempo, ya que posee comunicación full duplex. De las tres líneas que posee dos son las encargadas de transmitir los datos y la tercera es la línea del reloj o sincronización. Los dispositivos que se conecten a este bus pueden ser maestros o esclavos. Se denomina maestro al dispositivo que da inicio a la transferencia de datos mediante el bus y también es el encargado de generar dos señales, la de reloj y la de control. El esclavo es el dispositivo que está controlado por la línea del bus Select Slave, es decir el esclavo se activa si y solo si esta línea es seleccionada. El bus SPI transmite datos por medio de un registro de desplazamiento, todas las líneas del bus transmiten información en una sola dirección, estas líneas son (Ingeniería en Microcontroladores, s.f.):

- **SCLK:** la línea del reloj, es la encargada de la sincronización en la transferencia de datos, la genera el maestro.

- **MOSI**: es la línea encargada de transportar los datos desde el maestro hasta el esclavo.
- **CS**: cada uno de los dispositivos esclavos es seleccionado por la línea Chip Select.
- **MISO**: es la encargada de transportar los datos desde el esclavo hasta el maestro.

Figura 9

Comunicación SPI.



Nota. El gráfico muestra las líneas de comunicación entre un microcontrolador como maestro y dispositivos como esclavos para una comunicación SPI. *Tomado de Ingeniería en Microcontroladores*, por Ingeniería en Microcontroladores.

Protocolo Message Queue Telemetry Transport

MQTT es un protocolo con comunicación M2M (Machine to Machine) el cual se desempeña en entornos de bajo ancho de banda. Este protocolo es utilizado para mensajería ligera de Cliente/Servidor el cual se basa en publicaciones y suscripciones a los denominados "Topic", se ha vuelto muy habitual en el área de IOT o Internet de las cosas debido a su fácil implementación en hardware de dispositivos altamente limitados.

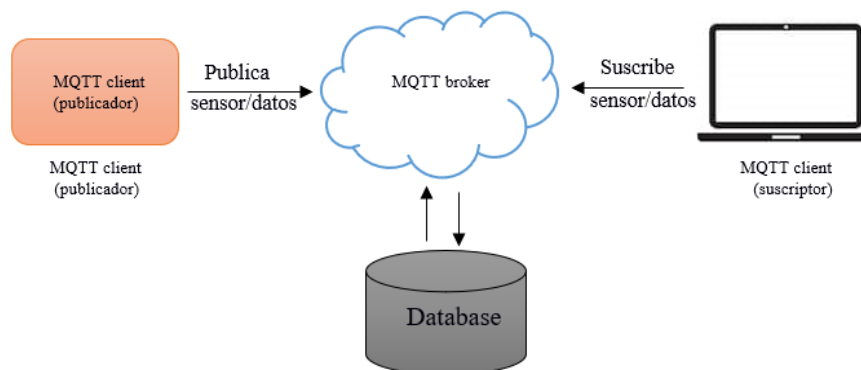
MQTT es conocido por ser un protocolo ágil ya que todos sus mensajes están identificados por medio de cabeceras. Todos los mensajes poseen un encabezado estático de 2 bytes, un encabezado variable (no necesariamente lo tienen todos los mensajes), el contenido

del mensaje que está limitada a 256 megabytes (MB) y por último un nivel para la calidad de servicio (QoS). (ARDUINO - GENUINO, s.f.) .

El protocolo MQTT se basa en el protocolo TCP/IP el cual soporta comunicación bidireccional y sin pérdidas de datos. Este protocolo posee varios elementos entre los cuales están (ARDUINO - GENUINO, s.f.):

Figura 10

Elementos de una comunicación por MQTT.



- **Broker:** es un servidor encargado de distribuir la información recibida hacia los clientes, es decir discrimina a que clientes pertenece cada mensaje.
- **Publicador:** es un cliente que envía información hacia el broker con un topic específico, para posteriormente dicha información sea distribuida hacia otro cliente que se haya suscrito a ese topic.
- **Suscriptor:** es un cliente que se suscribe al topic que le interesa, cuando un cliente se ha suscrito a un topic todos los mensajes que se han enviado hacia el broker con ese tema serán recibidos por dicho cliente.
- **Topic:** es el nombre que se asigna al mensaje, ya que cada cliente puede publicar o suscribirse a un topic.

- **Payload:** es toda la información o contenido de un mensaje.

El funcionamiento básico de este protocolo consiste en cuatro etapas:

Figura 11

Diagrama de las Etapas de comunicación con el protocolo MQTT.



Primero un cliente inicia la conexión con el broker por medio del puerto estándar 1883 o por un puerto personalizado 8883; el puerto es seleccionado por los coordinares del broker, una vez establecida la conexión se procede la autenticación por la petición de usuario y contraseña estos son enviados del cliente hacia el servidor. En la fase de comunicación el cliente realiza acciones de publicación y también de suscripción, cuando el cliente publica se inicia el envío de un bloque con datos a un topic específico para que posteriormente el broker lo distribuya hacia los suscriptores que lo requieran. Y por último en la etapa de terminación el cliente envía un mensaje de desconexión al broker (ARDUINO - GENUINO, s.f.).

Módulos Electrónicos

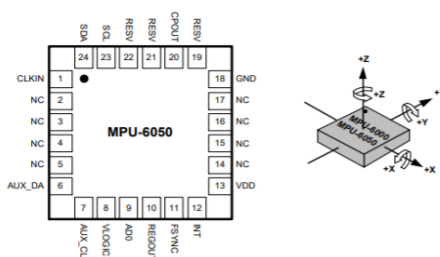
En este punto cabe destacar, que los dispositivos utilizados para el desarrollo el prototipo de sismógrafo o estación sismográfica parte del proyecto de titulación fueron los establecidos por el Señor Han Sehsub asesor de KOICA del Ministerio de Educación, entre los cuales se encuentran el acelerómetro MPU6050, Módulo Micro SD, módulo GY-GPS6MV2 (GPS) y ESP32, que fueron proporcionados al inicio del proyecto por el mismo asesor.

Módulo acelerómetro 6050

MPU6050 es un sensor perteneciente a la familia de acelerómetros MEMS (Micro-Electromechanical Systems) del tipo capacitivo de silicio, su principal características es que utiliza comunicación I2C, es muy popular en la manufacturación de dispositivos inteligentes gracias a su funcionalidad y tamaño, que lo ha convertido en un dispositivos imprescindible para teléfonos inteligentes, tablets, etc. Debido al enorme valor que agrega a la experiencia del usuario final en aplicaciones como: comandos de gestos para aplicaciones y control del teléfono, juegos mejorados, realidad aumentada, captura y visualización de fotografías panorámicas, y navegación para peatones y vehículos. Este dispositivo consta de 6 ejes de los cuales 3 pertenecen al acelerómetro Z, N (norte), E (este) y 3 al giroscopio X, Y, Z. En la Figura 12 se presenta el MPU6050 con su respectiva distribución de pines y orientación del mismo (IncenSense, s.f.).

Figura 12

Distribución de pines y orientación del MPU6050.



Nota. El gráfico indica la distribución de pines del módulo MPU6050, además de presentar su orientación tanto para el acelerómetro como para el giroscopio. Tomado de *MPU-6000 and MPU-6050 Product Specification Revision 3.4*, por IncenSense.

En el proyecto se utiliza el módulo GY-521 el cual tiene integrado un acelerómetro y giroscopio del tipo MEMS MPU6050, en la Figura 13 se presenta dicho módulo. En la Tabla 1 se describen las características principales del MPU6050.

Figura 13

Módulo GY-521.



Nota. El gráfico presenta la apariencia del módulo comercial MPU6050. Toma de *Naylamp* por *Naylamp Mechatronics*.

Tabla 1

Características del Módulo MPU6050.

Descripción	Características
Alimentación	3-5V
Sensibilidad	±2g ±4g ±8g ±16g
Resolución	16 bits
Comunicación	Digital I2C 400 KHz I2C modo rápido

Nota. La tabla muestra las características principales del módulo MPU6050 que se tomaron en cuenta en el presente proyecto. Tomado de *MPU-6000 and MPU-6050 Product Specification Revision 3.4*, por *Naylamp Mechatronics*.

Módulo Micro Secure Digital

El módulo Micro SD es utilizado como interfaz de lectura y escritura de una tarjeta Micro SD Card. Por medio de la programación el usuario puede realizar dichas acciones, permite la ejecución de archivos de audio , control MCU, ARM. La lectura se la puede realizar a través del

bus SPI ya que contiene las líneas SCKL, MOSI, MISO y CS . Este tipo de módulo tiene la ventaja de conectar una memoria externa casi ilimitada para los proyectos, por lo tanto es muy beneficioso para el registro de información de la estación (Electrónica, 2020).

Figura 14

Módulo SD Card.



Nota. La imagen muestra la presentación comercial del módulo SD Card. Tomado de *Electrónico Caldas*, por Electrónico Caldas.

Módulo de Posicionamiento Global

La serie de módulos NEO-6 es una familia de receptores GPS que presenta el alto rendimiento en posicionamiento. Estos receptores flexibles y rentables ofrecen numerosas opciones de conectividad, su arquitectura compacta y opciones de energía y memoria hacen que los módulos NEO-6 sean ideales para dispositivos móviles que funcionan con baterías con costos muy estrictos y limitaciones de espacio. El motor de posicionamiento u-blox 6 de 50 canales cuenta con un Time-To-First-Fix (TTFF) de menos de 1 segundo. El motor de adquisición, con 2 millones de correlacionadores, es capaz de realizar búsquedas masivas de espacio paralelo de tiempo y frecuencia, permitiéndole encontrar satélites al instante. La tecnología innovadora suprime las fuentes de interferencia, dando a los receptores GPS NEO-6 un excelente rendimiento de navegación. (Ublox, s.f.).

Figura 15

Neo-6 u-blox 6 GPS.

13	GND	GND	12
14	MOSI/CFG_COM0	RF_IN	11
15	MISO/NC	GND	10
16	CFG_GPS0/SCK/ Reserved	VCC_RF	9
17	Reserved/NC	Reserved	8
18	SDA2	VDDUSB	7
19	SCL2	USB_DP	6
20	TxD1	USB_DM	5
21	RxD1	EXTINT0	4
22	V_BCKP	TIMEPULSE	3
23	VCC	SS_N/NC	2
24	GND	Reserved	1

NEO-6
Top View

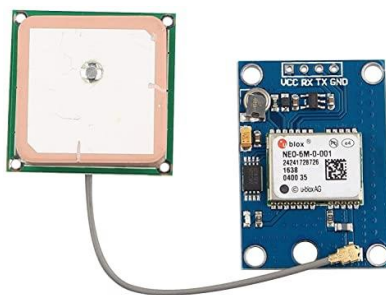
Nota. La imagen presenta el chip Neo-6 u-blox 6 GPS con su respectiva distribución de pines.

Tomado de *Ublox*, por Ublox.

La Figura 16 presenta el módulo GY-GPS6MV2 utilizado en el presente proyecto, dicho módulo está basado en el chip receptor Neo-6, además incluye una antena cerámica. En la Tabla 2 se detalla las características que posee este módulo.

Figura 16

Módulo GY-GPS6MV2.



Nota. La imagen representa la apariencia del módulo comercial del GY-GPS6MV2. Tomado de *Ublox*, por Ublox.

Tabla 2

Características del Módulo GY-GPSMV2.

Descripción	Características
Alimentación	3-5V
Receptor	Ublox NEO 6M
Interfaz	UART, comunicación asíncrona
Modelo	GY-GPSMV2

Nota. La tabla anterior muestra las características especiales del módulo GY-GPSMV2. Tomado de *Ublox*, por Ublox.

Esp wroom 32

El ESP WROOM 32 es un módulo que integra Wi-Fi y Bluetooth, dado que posee dichas tecnologías es usado para el desarrollo de productos de IoT. Al tener integrado Bluetooth, Bluetooth LE y Wi-Fi amplía la gama de aplicaciones para las cuales puede ser utilizado. El Wi-Fi permite conectarse a una red LAN (Local Area Network) por medio de un router con conexión a internet, en cambio el Bluetooth permite conectarse directamente a otros dispositivos. El chip integrado está diseñado para ser escalable y adaptado. Este integrado posee una gran suma de periféricos entre ellos : sensor táctiles capacitivos, sensor de efecto Hall, amplificador de bajo ruido, protocolo SPI, UART, I2S e I2C (Naylamp Mechatronics, s.f.).

Figura 17

Distribución de pines del ESP WROOM 32.



Nota. La imagen muestra el chip ESP WROOM 32. Tomado de *Naylamp*, por Naylamp Mechatronics.

En la Figura 18 se presenta el módulo ESP32 utilizado en el presente proyecto, dicho módulo está basado en el chip ESP WROOM 32. En la Tabla 3 se detalla todas las características más relevantes de este módulo.

Figura 18

Placa de desarrollo ESP32 NodeMcu-32.



Nota. La imagen nos presenta la placa ESP32 NodeMcu utilizada en el presente proyecto.

Tomado de *Naylamp*, por Naylamp Mechatronics.

Tabla 3

Características de la placa de desarrollo ESP32 NodeMcu32.

Descripción	Características
Alimentación	USB 5V DC
Voltaje Int/Out	3.3 V DC
Frecuencia	Hasta 240MHz
Wi-Fi	802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
Bluetooth	v4.2 BR/EDR and Bluetooth Low Energy (BLE)
Memoria ROM	448 KByte
Memoria RAM	520 KByte
Antena	Integrado en PCB
Desempeño	Hasta 600 DMIPS

Nota. La tabla presenta las principales características de la placa ESP32 NodeMcu 32. Tomado de *Naylamp*, por Naylamp Mechatronics.

Interfaces de Programación


Durante el desarrollo del proyecto se ha hecho uso de diferentes herramientas de programación las cuales han permitido realizar los códigos tanto para el software como para el hardware dichas herramientas se detallan a continuación.

Entorno de desarrollo integrado Arduino

Arduino IDE es un entorno de desarrollo de aplicaciones libre que fue creado por Windows, MacOS, Linux. Es utilizado comúnmente para escribir y cargar programas a placas de desarrollo compatibles con Arduino. Este entorno admite lenguajes como C y C++, esta interfaz proporciona una biblioteca de software de Wiring el cual entrega varios procedimientos de entrada y salida (Arduino, s.f.).

Figura 19

Interfaz de Arduino IDE.



```
Blink 5
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
32 Arduino/Genuino Uno on COM1
```

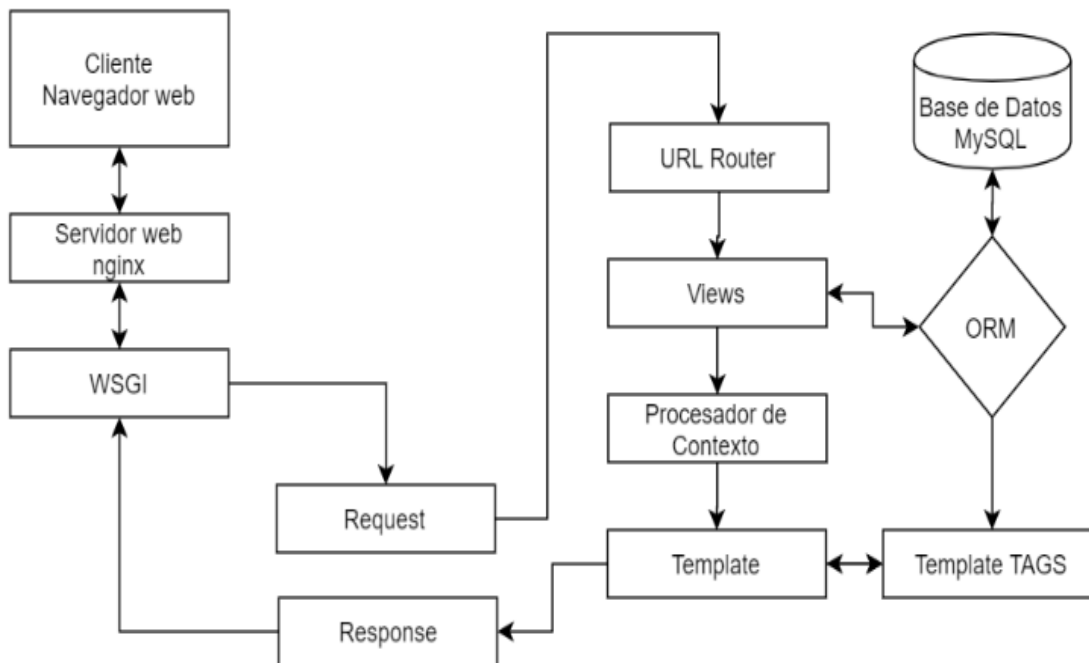
Nota. La anterior imagen muestra la interfaz de Arduino IDE en la cual se desarrollara el código del Hardware del proyecto. Tomado de *Arduino*, por Arduino.

Django

Django es un framework web que usa el lenguaje de Python, que se basa en el esquema MVC (Modelo, Vista y Controlador). Django maneja tareas comunes de desarrollo web, es extremadamente rápido y escalable, seguro e increíblemente eficaz, convirtiéndola en una herramienta robusta para el desarrollo de aplicaciones web permitiendo códigos limpios, simplificados y mucha facilidad para mantenerlos.

Figura 20

Esquema Django.



En el esquema de diseño MVC, los modelos en Django representan los modelos de datos subyacentes en el backend, mientras que las plantillas representan la vista. Curiosamente y no debe confundirse, la función de vista de Django funciona como un controlador. Django adopta la filosofía del acoplamiento suelto, donde diferentes partes de un sistema dependen libremente entre sí, por lo tanto, una sola parte del sistema tiene un rol que puede reemplazarse fácilmente con otros componentes funcionales similares que hacen que los componentes sean ortogonales.

El patrón de diseño MVC también permite modelos sueltos, vistas y componentes de controladores. Por ejemplo, las plantillas se usan para generar HTML de forma dinámica en el marco de Django y se pueden usar otros motores de plantillas en lugar del propio sistema de plantillas de Django (Solórzano Ávila, 2018).

Virtualenv

Virtualenv es una herramienta de software libre, que crea entornos Python con características independientes, por lo tanto, aborda el problema de las dependencias, las versiones y los permisos (Honores, 2015).

VisualCode

VisualCode es una interfaz de editor de texto avanzado altamente configurable que permite una edición eficiente del texto, una herramienta que se puede utilizar mediante la comunicación SSH con el shell del servidor para codificar el proyecto. A menudo conocido como "editor del programador", VisualCode es muy útil para el programador, ya que es ligero y rápido, ya que proporciona una forma fácil de editar texto con frecuencia (Wilstrom, 2014).

Pycharm

Es una herramienta muy popular entre los IDE para python, incluye grandes características y prestaciones para depurar código python, además soporta comunicación SSH con servidores para realizar la programación correspondiente, permitiendo al desarrollador la respuesta inmediata de la puesta en marcha para detectar posibles errores. Aunque en principio el proyecto se realizó en VisualCode con el tiempo se vio la necesidad de cambiar de IDE para obtener mejores prestaciones para la depuración de errores (Chávez, 2016).

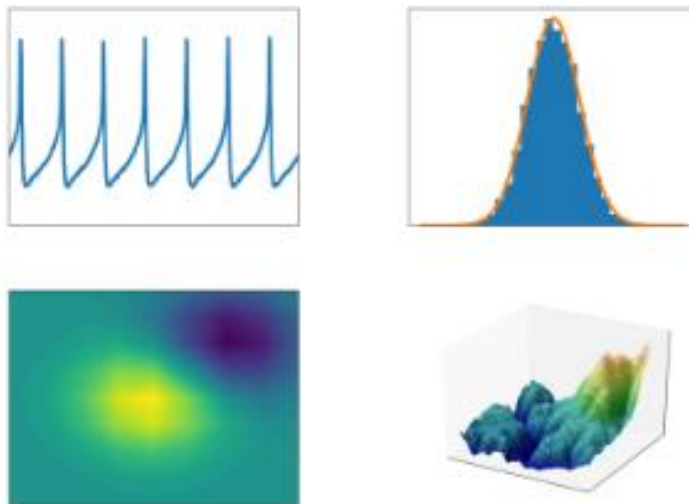
Matplotlib

Es una biblioteca desarrollada desde el 2003 por un neurobiologo llamado John Hunter, inspirado en el programa MATLAB de la empresa Mathworks, actualmente es un esfuerzo comunitario de muchas personas, para el desarrollo y mantenimiento de la librería.

Algo relevante que tenemos que apreciar de MATLAB es que las funciones de graficación ya están integradas en el sistema mientras que en python, debemos agregar la librería posterior a la instalación de un ambiente de trabajo configurado.

Figura 21

Librería Matplotlib.



Nota. Las gráficas muestran resultados obtenidos de graficar datos con la herramienta Matplotlib. Tomado de *Matplotlib*, por Matplotlib.

Mysql

El mercado oferta varias opciones en el ámbito de servidores de base de datos, entre ellas se encuentra la empresa Oracle una de las más demandadas en este campo ya que posee el servidor MySQL el cual ofrece un servicio de base de datos SQL (Structured Query Language), además de ser muy rápido, dar soporte de multi hilos de ejecución de procesos, ingresos de solicitudes de varios usuarios y ser fuerte. Según su página web el servidor MySQL fue diseñado para entornos de producción con tiempos críticos, con alto tráfico de usuarios, además de ser integrado en un paquete de fácil distribución.

Entre las opciones de los usuarios que deseen ocupar esta plataforma, se dispone de doble licencia, donde la primera son opciones de Open Source, que cuenta con términos y condiciones GNU (General Public License), y como una segunda alternativa disponemos de una opción de pago que es una licencia comercial estándar de MySQL. Como muchos productos que

tiene demanda, este en especial está en constantes actualizaciones, y para los desarrolladores dispone también de un manual de referencias.

Las principales características de MySQL:

- Alta confiabilidad en los datos.
- Está escrito en código C y C++
- Testeado con muchos compiladores.
- Funciona en varias opciones de sistemas operativos.
- Funcionalidades completas para multi-hilo por medio de la utilización de hilos del kernel. (Wilstrom, 2014)

Base de datos

Las bases de datos o más conocida por su abreviatura en las siglas BD son indispensables en el almacenamiento de datos digitales, a tal punto de llegar a ser el giro de negocio para muchas empresas como google o facebook entre otras. Las BD tienen una estructura de almacenamiento muy conveniente ya que organiza los datos de tal manera que las búsquedas se realizan de forma rápida y eficiente.

La base de datos es un conjunto de estructuras de datos que se almacenan en una memoria estática, que además de funciones de escritura, tiene opciones de lectura y búsquedas basadas en parámetros para generar filtros.

Las bases de datos están basados en tablas, y en su interior se disponen de columnas por cada dato nuevo que se agregue y tantas filas como parámetros se requiera. Esta información sostenida en una fila, contendrá toda la información asociada con el registro.

En la siguiente lista se presentan las características más relevantes de las BD:

- Los registros disponen de autonomía lógica y física.
- Mínima redundancia de datos.

- Los usuarios pueden acceder a la información en lazos simultáneos.
- Los datos permanecen íntegros a los originales.
- Los procesos de seguridad tienen características confiables
- Tienen opciones de respaldo de datos.
- Con el tiempo se ha consolidado un lenguaje para trabajar con las BD

Desde los inicios de la informática una tecnología denominada DBMS que en sus siglas en inglés es: Database Management Systems o Sistema de Gestión de BD. Estos sistemas tienen la capacidad de ordenar datos y recuperarlos de una forma confiable y rápida.

Al desarrollar una base de datos se tiene que escoger diferentes formas de organizarla, en la actualidad existen diferentes estructuras, modelos y paradigmas, cada opción nos brinda mejores prestaciones en algunos aspectos, y dificultades en otros. Sus diferencias están encaminadas en la jerarquía, su forma de comunicar los datos y la relación entre las distintas estructuras de datos. A todo este conjunto de características se las conoce como modelos de BD o DB (Data Base), y brinda opciones de implementación en los algoritmos entre diferentes alternativas de gestión (Wilstrom, 2014).

Lenguajes de programación

C++

C++ es un lenguaje enfocado en el desarrollo de una variedad de aplicaciones y sistemas operativos, además es la base de otros lenguajes de programación como los son Java, C++ o C#.

Este lenguaje posee varias características entre ellas las más importantes es que es un lenguaje estructurado, no depende de un hardware, permite tener el control absoluto de todo lo que sucede en el ordenador, su programación es de forma rápida y potente, posee una gran variedad de datos operadores y variable (Lucas, 2020).

JavaScript

JavaScript es un lenguaje de programación que permite el desarrollo en páginas web, tal como crear contenido nuevo y dinámico en páginas web, por ejemplo mostrar actualizaciones de contenido, interactuar con mapas, animaciones gráficas 2D/3D, etc. Cabe destacar que JavaScript por lo general trabaja conjuntamente con HTML, CSS (MDN web docs, s.f.).

Python

Es un lenguaje de programación que tiene características de alto nivel, que permite al programador la facilidad para leer e implementar con rapidez. Es de código abierto, lo que significa que es libre y gratuito, incluso para aplicaciones comerciales. Python puede ejecutarse en sistemas Mac, Windows y Unix y también ha sido portado a máquinas virtuales Java, .NET entre otras que existen en el mercado.

Este lenguaje está basado en un intérprete y no en un compilador, por lo tanto su código es ejecutado por un segundo programa que se instala en el computador Host, como Ruby o Perl, en muchas ocasiones se usa para crear aplicaciones web. También es compatible con una serie de programas de imágenes 2D y 3D, lo que permite a los usuarios crear complementos y extensiones personalizados con Python.

Python es un lenguaje que tiene una curva de aprendizaje menos pronunciada, permite al programador incursionar con mayor facilidad que otros lenguajes que existen, su sintaxis es de estructura reducida y sencilla. Este lenguaje es altamente aplicado en el campo de redes neuronales como aplicaciones de machine learning o deep learning ya que existe una gran comunidad que fomenta y colabora el uso con muchas librerías de código abierto y libre distribución. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta,

generando archivos .pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones de acuerdo a (González, 2020).

En el capítulo expuesto se ha sustentado teóricamente los conceptos a utilizar en el proyecto de grado en el cual se ha descrito de manera breve y concisa los elementos a usar tanto en software como en hardware. En el capítulo III se explicará el desarrollo del hardware del prototipo planteado, para que posea las características deseadas, entre ellas la principal detectar movimientos sísmicos significativos mediante el análisis de los datos obtenidos acelerómetro, para lo cual se hará uso del algoritmo STA/LTA, que se alojará en el microcontrolador de la tarjeta ESP32. Además se presentará el diseño del circuito electrónico y la estación sismográfica terminada.

Capítulo III

Desarrollo del hardware del prototipo de sismógrafo para el sistema de monitoreo de sismos

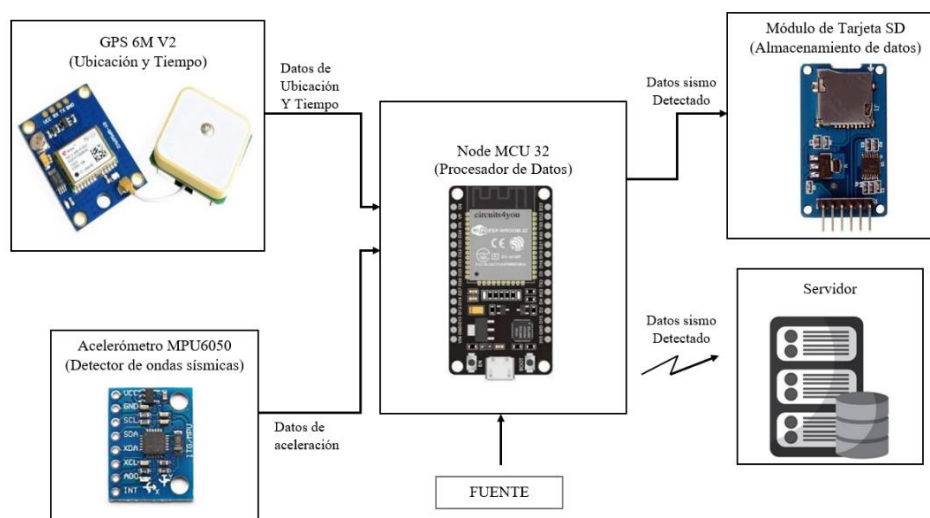
Hardware del sistema

Estructura del Hardware

El hardware del proyecto está conformado por varios módulos los cuales van conectados entre sí, la conexión de los módulos se representa en un diagrama de bloques en la Figura 22 con sus respectivas descripciones.

Figura 22

Diagrama de bloques prototipo de sismógrafo.



El prototipo de sismógrafo o estación está desarrollado para que posea las siguientes características:

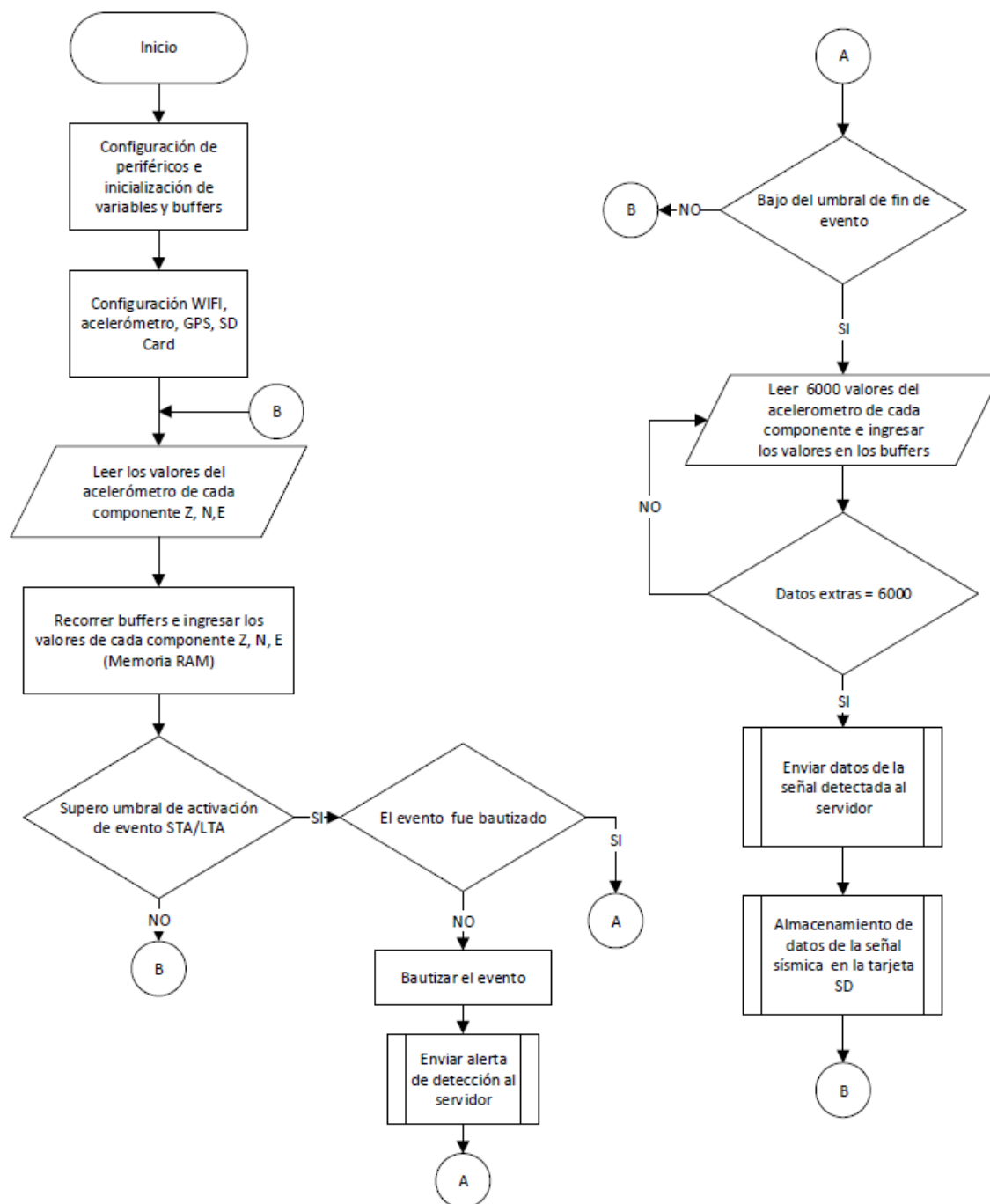
- El código se desarrolló en la interface Arduino IDE versión 1.8.5
- Se hizo uso de una computadora portátil (Laptop) con las siguientes características:
procesador Intel Core i7, sistema operativo de 64 bits, memoria RAM de 8 GB,
Windows 10

- El programa se ejecutará en la placa Node MCU 32 la cual tiene un chip ESP WROOM 32 y trabaja bajo el estándar IEEE 802.11 b/g/n Wi-Fi.
- Proporciona la ubicación geográfica, hora y fecha en tiempo real cuando se detecte un movimiento sísmico.
- Se encargará de recolectar y analizar los datos proporcionados por el sensor MPU6050 de los componentes Z, N, E para la detección de movimientos telúricos.
- Envía una alerta al servidor cuando se ha superado el umbral de activación indicando la detección de un evento sísmico, además bautiza dicho evento con un nombre único.
- Envía hacia el servidor los datos de los tres componentes Z, N, E cuando se ha detectado un evento sísmico, conjuntamente con datos adicionales como: hora de detección, nombre de la estación, componente, valores.
- Almacena datos de los eventos sísmicos detectados en la tarjeta SD.

La Figura 23 muestra el diagrama de flujo del Hardware del prototipo de sismógrafo, en el cual se presenta la lógica que sigue el programa desarrollado. A dicho programa se lo ha nombrado *Sismografo_ESPE2020.ino*. El diagrama de flujo sigue el proceso cuando el prototipo de sismógrafo ha detectado un movimiento fuerte, por lo que notificará de la detección de un evento al servidor para que este declare si es un evento real o simplemente es un movimiento causado por algún agente externo. Una vez enviado el mensaje de alerta el hardware continúa con el proceso de almacenamiento de datos en la memoria RAM de la ESP32, si no recibe ningún mensaje del servidor alertando que no es un sismo real este procederá a enviar los datos de los tres componentes de la señal detectada, para que sean almacenados en la base de datos del servidor. Una vez finalizado el envío de los datos procede a almacenarlos en una tarjeta SD como respaldo. Dicho proceso se detalla de manera más explícita durante este capítulo.

Figura 23

Diagrama de flujo del Hardware.



Librerías de los módulos

Para realizar las respectivas configuraciones de los módulos se ha visto necesario el uso de las librerías desarrolladas por Arduino pertenecientes a cada módulo.

Wire.Esta librería permite la comunicación con dispositivos que utilicen el protocolo I2C.

Dentro de este sistema se utiliza dos líneas de transmisión: la de datos SDA y la de reloj SCL.

Dicha librería fue utilizada para trabajar con el acelerómetro MPU6050 (Arduino, s.f.).

Tabla 4

Funciones de la librería Wire para el acelerómetro MPU6050.

Función	Descripción
Wire.beginTransmission(address)	Inicia una transmisión dirigida hacia los dispositivos esclavos I2C con la dirección especificada.
Wire.write(register)	Se encarga de escribir los datos de un dispositivo esclavo en respuesta a la solicitud del maestro o coloca en cola bytes para su transmisión desde un dispositivo maestro a un esclavo.
Wire.endTransmission(bolean)	Encargada de finalizar una transmisión con un dispositivo esclavo el cual inicio con <i>beginTransmission()</i> , y por último transmite los bytes que estan en cola por <i>write()</i> . Si se envía un argumento booleano este tiene hacer las siguientes acciones: <ul style="list-style-type: none"> ● Verdadero, enviar un mensaje de detención luego de la transmisión, liberando el bus I2C . ● Falso, enviar un mensaje de reinicio posterior a la transmisión. El bus no se liberará.

Wire.requestFrom(address,register,boolean)	<p>Función perteneciente al maestro para solicitar bytes a un dispositivo esclavo. Los bytes son obtenidos con la función <code>read()</code>.</p> <p>Si se envía un argumento booleano este tiene las siguientes acciones:</p> <ul style="list-style-type: none"> • Si es verdadero, este envía un mensaje de detención luego de la solicitud, liberando el bus I2C. • Si es falso, se envía un mensaje de reinicio al finalizar la solicitud. El bus no se liberará.
Wire.read()	<p>Con esta función se lee un byte transmitido desde un dispositivo esclavo a un maestro luego de una llamada a <code>requestFrom()</code>.</p>

Nota. Esta tabla describe las características de las funciones utilizadas de la librería WIRE.

Tomado de Arduino, por Arduino.

Time. Esta librería permite obtener los parámetros de fecha y tiempo ya que al iniciar esta librería lo primero que realiza es declarar una variable especial llamada `time_t` de 32 bits. Esta variable almacena en un formato especial los segundos transcurridos desde el 1 de enero de 1970 hasta la actualidad, con lo cual se realizan los cálculos automáticos de fecha y hora, por medio de las funciones descritas en la Tabla 5.

Esta librería conjuntamente con la librería TinyGPS++ se utilizó para trabajar con el módulo GY-GPS6MV2 el cual nos proporcionará ubicación y tiempo para algunos propósitos del proyecto (Arduino, s.f.).

Tabla 5

Funciones de la librería Time para el GPS-GPS6MV2.

Función	Descripción
setTime(hour,minute,second,day,month,year)	Esta función nos permite actualizar en que día y hora estamos y que a partir de allí mantenga adecuadamente esa fecha hasta su próximo reseteo.
adjustTime()	Esta función nos permite ajustar la hora dependiendo la zona horario donde nos encontremos.

Nota. Esta tabla describe las características de las funciones utilizadas de la librería TIME.

Tomado de Arduino, por Arduino.

Tinygps++. Esta librería fue diseñada para proporcionar la mayoría de las funcionalidades de GPS NMEA entre ellas posición, fecha, hora, altitud, velocidad y rumbo (Arduino, s.f.).

Tabla 6

Funciones de la librería TinyGPS++ para el GPS-GPS6MV2.

Función	Descripción
encode()	Esta función proporciona los datos NMEA seriales.
now()	Esta función devuelve el número de segundos desde el 1 de enero de 1970.

Nota. Esta tabla describe las características de las funciones utilizadas de la librería TinyGPS++.

Tomado de Arduino, por Arduino.

File System, Secure Data, Serial Peripheral Interface. La librería **FS (File System)** nos permite utilizar funciones del sistema de ficheros, mientras que la librería **SD** permite leer y escribir en tarjetas **SD**. La tarjeta **SD** y el microcontrolador se comunican mediante **SPI**, por lo que también se hace uso de la librería **SPI** la cual permite la comunicación con dispositivos externos **SPI**. Estas tres librerías se usan para trabajar con el módulo **Micro SD (Arduino, s.f.)**.

Tabla 7

Funciones de las librerías FS, SD y SPI para el módulo Micro SD.

Función	Descripción
read()	Esta función permite leer los datos seriales entrantes.
SD.open(filepath, mode)	Es la encargada de abrir un archivo en la tarjeta SD. Cuando el archivo se abre para escritura, y si no existe se creará.

Nota. Esta tabla describe las características de las funciones utilizadas de las librerías **FS, SD, SPI**.

Tomado de Arduino, por Arduino.

Todas las librerías descritas en esta sección fueron integradas para el fin del proyecto y desarrollo del prototipo de sismógrafo.

Algoritmo detección de eventos sísmicos sta/lta

El algoritmo **STA/LTA** (*Short time average/long time average*) es utilizado generalmente en aplicaciones que desean registrar tantos eventos sísmicos como sea posible. Este algoritmo de activación estándar es muy utilizado en grabadores sísmicos, softwares de procesamiento en tiempo real de redes sísmicas, las cuales están destinadas a la detección de eventos sísmicos (Tmkoczy, 1999).

El algoritmo **STA/LTA** procesa las señales sísmicas a través de dos ventanas móviles, una ventana de corta duración denominada **STA** y otra ventana de larga duración denominada **LTA**. La

ventana de tiempo corto (STA) es sensible a eventos sísmicos, comúnmente toma intervalos temporales (t_{STA}) de 1 a 4 segundos, mientras que la ventana de tiempo largo (LTA) proporciona información sobre la amplitud temporal del ruido sísmico en el sitio, normalmente toma intervalos temporales (t_{LTA}) de 30 a 60 segundos (Tmkoczy, 1999).

El valor de la relación STA/LTA se determina por medio de dos ventanas por lo que se debe definir el tamaño de la ventana corta (STA) con el total de número de muestras N y el tamaño de la ventana larga (LTA) con el total de número de muestras M , para el cálculo de N y M se inicia a partir de la tasa de muestreo F_s de la siguiente forma (Paredes Bernal, 2015).

$$N = F_s * t_{STA} \quad (1)$$

$$M = F_s * t_{LTA} \quad (2)$$

Se considera a x_i como cada valor del registro en función del tiempo (Paredes Bernal, 2015), en (Wong J., 2009) definen el cálculo de la relación STA/LATA como:

$$STA = \frac{1}{N} \sum_{j=i-N}^i x_j^2 \quad (3)$$

$$LTA = \frac{1}{M} \sum_{j=i-M}^i x_j^2 \quad (4)$$

Si:
$$j \leq 0, x_j = \frac{x_1 + x_2}{2} \quad (5)$$

Por lo tanto la relación STA/LTA viene dada por:

$$radio_i = \frac{STA}{LTA} \quad (6)$$

Niveles de Umbral

El algoritmo STA/LTA se caracteriza por realizar una comparación continua de umbrales con la relación STA/LTA, es decir si la relación supera el umbral de activación(*trigger*) se indica la detección de un evento y de inmediato se empieza a guardar los datos de la señal detectada hasta que la relación STA/LTA cae por debajo del umbral de desactivación (*detrigger*), indicando así el fin del evento. Cabe destacar que el valor de *detrigger* debe ser menor al valor *trigger* (Tmkoczy, 1999).

Parámetros previos y posteriores al evento

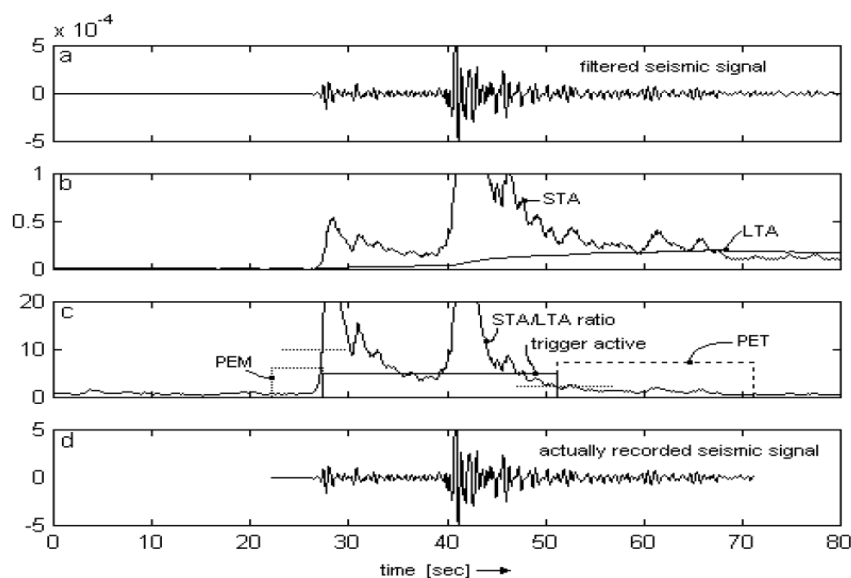
Adicionalmente de los datos del evento sísmico detectado las redes de sensores sísmicos añaden una cantidad de datos pertenecientes a segundos antes de sobrepasar el umbral de activación (*trigger*), al cual se lo ha nombrado como PEM (pre-evento o en sus siglas en inglés pre-event time); además se añade otra cierta cantidad de datos después de haber alcanzado el valor del umbral de desactivación (*detrigger*) denominado PET (post-evento o sus siglas en inglés post-event time), entonces los datos que se registraron del evento detectado son: datos del pre-evento, datos del evento y datos del post-evento, esto se realiza con el fin de no perder información de la señal sísmica detectada (Tmkoczy, 1999).

Procesamiento del algoritmo sta/lta

A continuación se hace referencia a un ejemplo planteado por (Tmkoczy, 1999) en el cual se puede ver de forma detallada la forma como se procesa una señal sísmica mediante el algoritmo de disparo STA/LTA.

Figura 24

Ejemplo del funcionamiento del Algoritmo de disparo STA/LTA.



Nota. Las gráficas muestran el resultado de aplicar el algoritmo STA/LTA en una señal sísmica.

Tomado de *Understanding and parameter setting of STA/LTA trigger algorithm*, por A. Tmkoczy, 1999.

En la Figura 24 a se puede ver una señal sísmica entrante, la Figura 24 b muestra los valores absolutos del cálculo tanto de STA y LTA respectivamente, mientras que en la Figura 24 c podemos observar los valores absoluto pertenecientes a la relación STA/LTA de los datos de la señal a través del tiempo, además del tiempo en el cual estuvo el disparador (línea continua), los umbrales de activación y desactivación (líneas punteadas), que en este caso han elegido 10 para el umbral de activación (trigger) y 2 para el umbral de desactivación (detrigger). Por lo tanto se puede observar que cuando la relación superó el umbral de 10 el disparo se activó y cuando bajo de 2 se desactivo. Por último podemos observar los tiempos de PEM y PET representados por los cuadros de líneas entrecortadas. En la Figura 24 d se puede observar la señal sísmica registrada (Tmkoczy, 1999) .

Desarrollo del algoritmo para la detección de ondas sísmicas

Para la detección de eventos sísmicos se ha desarrollado un código el cual se ejecuta en la tarjeta ESP32, para lo cual se ha creado la función denominada *read_data()* como se muestra en el Segmento de código 1.

Segmento de código 1

Función *read_data()*.

```

7 void read_data(){
8
9   Wire.beginTransmission(MPU_addr);
10  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
11  Wire.endTransmission(false);
12  Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
13
14  //*****Ingreso de datos*****
15  for(int j=0;j<=10000-2;j++){
16    buff_N[j]=buff_N[j+1];
17    buff_E[j]=buff_E[j+1];
18    buff_Z[j]=buff_Z[j+1];
19  }
20  buff_N[10000-1]=Wire.read()<<8|Wire.read();// 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
21  buff_E[10000-1]=Wire.read()<<8|Wire.read();// 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
22  buff_Z[10000-1]=Wire.read()<<8|Wire.read();// 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
23

```

Primeramente se ha creado tres buffers de característica circular (desplazamiento de datos) como se puede ver en las líneas 5 a la 18 del Segmento de código 1, estos buffers permiten el alojamiento de los valores de aceleración correspondientes a los tres ejes tanto Z, N, E obtenidos del acelerómetro MPU6050 el cual está configurado con sensibilidad de $\pm 2g$; el acelerómetro nos entrega valores sin procesar, por lo tanto es necesario transformarlos a unidades de m/s^2 como se puede ver en las líneas de código 27 y 37 del Segmento de código 2. Cada buffer puede alojar hasta 10000 datos, a continuación se ha procedido a implementar los fundamentos matemáticos del algoritmo STA/LTA mediante código como se puede observar en el Segmento de código 2. Dicho algoritmo solo se aplica a los datos obtenidos de la componente

Z, debido al limitante de memoria RAM que se dispone de la tarjeta ESP32. Los valores que tomarán las variables de dicho algoritmos se detallan en la Tabla 8, estos valores fueron elegidos de acuerdo a las recomendaciones presentadas en el trabajo llamado “*Understanding and parameter setting of STA/LTA trigger algorithm*” (Tmkoczy, 1999). Este documento indica que los valores adecuados de STA son de 1 a 3 segundos y para LTA esta de 30 a 60 segundos. El valor de umbral de activación (*trigger*) inicial adecuado es de 4 para sitios con presencia de eventos sísmicos débiles.

Tabla 8

Valores del Algoritmo STA/LTA.

Variable	Valor
<i>F_s</i>	100 Hz
<i>t_{STA}</i>	2 s
<i>t_{LTA}</i>	60 s
<i>N</i>	200
<i>M</i>	6000
<i>Umbral_{activacion}(th_on)</i>	4
<i>Umbral_{desactivacion}(th_off)</i>	1
<i>PEM</i>	2 s
<i>PET</i>	60 s

Segmento de código 2

Programación del algoritmo STA/LTA en el eje Z.

```

24 //*****Algoritmo STA/LTA*****
25 lta_sum=0.0;
26 for(int j=3999;j<=10000-1;j++){
27     float tmp=buff_Z[j]*conv-11.52985;
28     if(tmp<0){
29         tmp=tmp*(-1);
30     }
31     lta_sum=lta_sum+tmp;
32 }
33 lta=lta_sum/6000;
34
35 sta_sum=0.0;
36 for(int j=9799;j<=10000-1;j++){
37     float tmp1=buff_Z[j]*conv-11.52985;
38     if(tmp1<0){
39         tmp1=tmp1*(-1);
40     }
41     sta_sum=sta_sum+tmp1;
42 }
43 sta=sta_sum/200;
44 sta_to_lta=sta/lta;
45

```

Librerías para la comunicación Mqtt y mensajería en formato Json

Las librerías que se describirán a continuación son las que nos permiten realizar la comunicación y el envío de datos hacia el servidor a través de protocolo MQTT.

Arduinojson

Esta librería incorpora funciones para serializar y deserializar objetos (clave/valor) de manera sencilla en formato JSON (JavaScript Object Notation), este es un formato de texto plano para almacenar datos estructurados. Mediante esta librería se generan los templating por medio de los cuales se indicará la memoria que se le asigna al fichero Json (ArduinoJson, s.f.).

Tabla 9

Comando de la librería ArduinoJson.

Comando	Descripción
StaticJsonDocument<length> doc	Este comando permite generar un espacio de memoria dedicado al fichero Json

Nota. Esta tabla describe las características de la función utilizada de la librería ArduinoJson.

Tomado de ArduinoJson, por ArduinoJson.

Wi-Fi

Esta librería permite la conexión a Internet, acepta conexiones entrantes o un cliente que realiza conexiones salientes (Arduino, s.f.).

Pubsubclient

Esta biblioteca permite realizar mensajes simples de publicación / suscripción mediante un servidor el cual permite una comunicación MQTT (Arduino, s.f.).

Tabla 10

Función para protocolo MQTT.

Función	Descripción
client.publish(topic,char(message))	Esta función permite publicar un mensaje en un topic específico en un broker

Nota. Esta tabla describe las características de la función utilizada de la librería WIFI y PubSubClient . Tomado de Arduino, por Arduino.

Desarrollo del código para la comunicación con el servidor

Para lograr las características deseadas del prototipo de sismógrafo se ha desarrollado varias funciones, las cuales se comunican con el servidor mediante el protocolo MQTT por lo que cada una de ellas se suscriben a diferentes “topic”.

- *Alerta_Evento()* esta función es la encargada de alertar la detección de un evento sísmico hacia todas las estaciones de la red de sismógrafos a través de un mensaje por MQTT hacia el servidor o broker al topic “IDxx/Urg”, para que posteriormente el servidor envíe dicho mensaje a las demás estaciones. Este mensaje contiene tres parámetros importantes: “Utm” o nombre con el que se ha bautizado el evento, un indicador “Cmd” de alerta y el nombre de la estación. Son necesarios estos tres aspectos debido a que se ha implementado un sistema de nombramiento de evento tanto en el servidor como en las estaciones; es decir que la primera estación de la red de estaciones sismográficas que haya detectado un evento sísmico, será la encargada de bautizar dicho evento y a través de un mensaje notificará la detección y bautizo de dicho evento.

El formato que se utiliza para el envío de este mensaje está bajo Json, este formato está compuesto de tres claves con sus respectivos valores como se indica a continuación:

$$\{Utm: hhmmssms, Cmd: 1, Est: nombre_{estación}\}$$

Donde la clave *Utm* contiene la hora (hora-minutos-segundos-milisegundos) a la que el sistema detectó el evento este tiempo se ha obtenido de los datos proporcionados por el módulo GPS, además este será el nombre o ID con el que se identificará al evento, *Cmd* tiene el valor de 1, ya que en el servidor y las estaciones se ha establecido este valor como un indicativo para un evento detectado, *Est*

contiene el nombre de la estación que está alertando la detección de un evento. El código correspondiente a esta función se presenta en el Segmento de código 3, en donde en el condicional *if* de la línea 19 se especifica al “topic” al cual se dirige en el broker y se inserta el mensaje en formato Json.

Segmento de código 3

Función Alerta_Evento().

```

5 void Alerta_Evento() {
6
7   StaticJsonBuffer<100> JSONbuffer;
8   JsonObject& alert = JSONbuffer.createObject();
9
10  alert["Utm"] = Evt;
11  alert["Cmd"] = "1";
12  alert["Est"] = estacion;
13
14  char JSONmessageBuffer[100];
15  alert.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer));
16  Serial.println(F("Sending message to MQTT topic.."));
17  Serial.println(JSONmessageBuffer);
18
19  if (client.publish("IDxx/Urg", JSONmessageBuffer) == true) {
20    Serial.println(F("Success sending message"));
21  } else {
22    Serial.println(F("Error sending message"));
23  }
24  Serial.println(F("-----"));
25 }

```

- *Callback()* se ha optado por usar esta función para recibir notificaciones desde el servidor ya sea de estaciones externas que han detectado un evento sísmico o del servidor cancelando una alerta o falsos positivos. Mediante esta función la estación recibe un mensaje, una vez capturado el mensaje se procede a leer el contenido o payload de este, el payload contiene la trama de información enviada por la función *Alerta_Evento()*, si en el payload se verifica que el mensaje proviene de cualquier estación dentro de la red se procede a guardar el nombre con el que se ha bautizado el evento (*Utm*) para cuando esta estación detecte el evento ya no nombre dicho evento sino que use el que se le ha notificado. En el Segmento de

código 5 se muestra la función `callback()` y en las líneas de la 14 a la 26 podemos ver como se lee el contenido del payload y se guarda el Utm recibido.

Segmento de código 4

Función `callback()`.

```

4 void callback(char* topic, byte* payload, unsigned int length) {
5
6   Serial.print("Message arrived [");Serial.print(topic);Serial.print("] ");
7
8   for (int i = 0; i < length; i++) {
9     Serial.print((char)payload[i]);
10  }
11  Serial.println();
12
13  //Alerta de una estación externa
14  if ((char)payload[36] >= '1' && (char)payload[36] <= '4') {
15    alertas=alertas+1;
16    //Alerta
17    digitalWrite(BUILTIN_LED, HIGH);
18    delay(1000);
19    digitalWrite(BUILTIN_LED, LOW);
20
21    for (byte i = 0; i < 9; i++) {
22      Utm[i]=payload[i+8];
23    }
24    for (byte i = 0; i < 1; i++) {
25      Cmd[i]=payload[i+26];
26    }
27    //Almacena el nombre del evento detectado por otra estación
28    if(flag_call_server==0){
29      now_1 = millis();
30      Evt=String(Utm);
31      cmd=String(Cmd);
32      flag_call_server=1;
33    }

```

Además a través de esta función también se verifica la alerta de “falsos positivos” es decir, cuando el servidor anteriormente notifica que alguna estación había detectado un evento y al pasar del tiempo no se ha cumplido que un mínimo de del 50% de las estaciones notifique dicho evento, el servidor procede a enviar un mensaje de “falso positivo” hacia todas las estaciones, con las mismas claves/valores que se envían cuando se da alerta de un evento, para que estas borren el nombre que se le había asignado a ese evento, mediante el siguiente formato:

{Utm: 000000000, Cmd: 2, Est: 0}

El servidor notifica a todas las estaciones por medio de un mensaje como estación cero y el valor de *Cmd* igual a dos, para que las estaciones sepan que se ha alertado un falso positivo, como se indica en el Segmento de código 5.

Segmento de código 5

Función Callback().

```

38 //Alerta falso positivo
39 if ((char)payload[36] == '0') {
40     for (int i = 0; i < 9; i++) {
41         Utm[i]=payload[i+8];
42     }
43     for (byte i = 0; i < 1; i++) {
44         Cmd[i]=payload[i+26];
45     }
46
47     Evt=String(Utm);
48     cmd=String(Cmd);
49     flag_call_server=0;
50     alertas=0;
51     for(int i = 1; i <= 3; i++){

```

- *JSON_Z()*, *JSON_N()*, *JSON_E()* estas funciones se encargan de enviar los datos de cada componente Z, N, E pertenecientes a la señal sísmica detectada hacia el servidor estos mensajes son enviados al "topic" "ID01/datos" para que sean procesados y almacenados en la base de datos, en el Segmento de código 6 se muestra una de las tres funciones ya que las tres siguen el mismo concepto. Estas funciones de igual manera envían mensajes en formato JSON con las siguientes claves y valores:

{Evt: hhmssms, Est: nombre_{estación}, Pqt: #paquete, Cmp: z ó n ó e, Time: hhmssms, Value: {d₁, d₂, d₃, ..., d₅₀}}

Donde *Evt* contiene la hora a la que se detectó el evento; es el mismo nombre o ID que se envió a través del mensaje de alerta en la clave *Utm*, *Est* el nombre de la estación, *Pqt* el número de paquete que se está enviando, en este punto cabe

recaltar que el total de datos de cada componente se han dividido en paquetes de 50 datos para su envío; *Cmp* indica a qué componente pertenecen los datos que se están enviando, *Time* contiene una hora aproximada de cuando ingresaron los primeros datos del sismo, *Value* contiene los datos correspondientes a la señal sísmica, dichos valores son enviados en unidades de cm/s^2 ya que los procesos que se realizan posteriormente en el servidor tiene la necesidad de trabajar en dichas unidades. En las líneas de la 21 a la 24 del Segmento de código 6 se puede observar cómo se realiza la transformación de m/s^2 a cm/s^2 ya que los valores que nos entrega el acelerómetro se encuentran en m/s^2 .

Segmento de código 6

Función JSON_Z envío de datos de la componente Z.

```

11     StaticJsonBuffer<730> JSONbuffer;
12     JsonObject& JSONencoder = JSONbuffer.createObject();
13     pqst_send=pqst_send+1;
14     JSONencoder["Evt"] = Evt;
15     JSONencoder["Est"] = estacion;
16     JSONencoder["Pqt"] = String(pqst_send);
17     JSONencoder["Cmp"] = "z";
18     JSONencoder["Time"] = String(tiempo_1_J)+String(Sec_1)+"."+String(milsec);
19     JsonArray& values = JSONencoder.createNestedArray("Value");
20     for(byte j=0;j<50;j++){
21         //Conversion de m/s2 a cm/s2
22         float valor1=(buff_2[incremento+start_data+j]*conv-11.52985)*100;
23         String valor2=String(valor1,4);
24         float val_cm=valor2.toFloat();
25         values.add(val_cm);
26         if(incremento+start_data+j>=9999){
27             j=51;
28         }
29     }

```

- *Registrar_Evento()* esta función la encargada de que el evento detectado sea registrado dentro de la base de datos alojada en el servidor web. De la misma manera se envía un mensaje por MQTT hacia el servidor, pero en este caso un

mensaje tipo *String* como lo podemos observar en la línea 10 del Segmento de código 7 y este mensaje se envía hacia el "topic" *"ID01/info"*.

Segmento de código 7

Función Registrar_Evento().

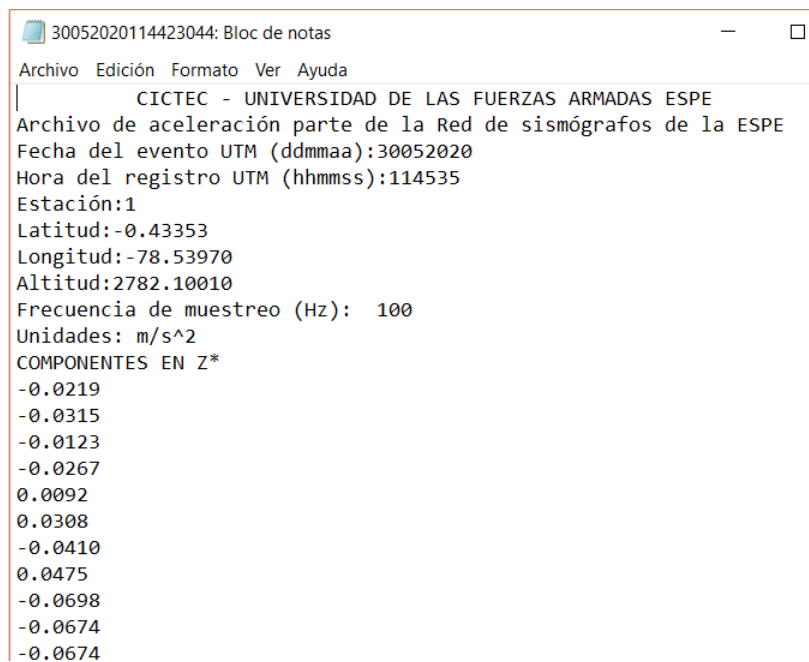
```

5 void Registrar_Evento() {
6
7   String messageBuffer;
8   char   JSONmessageBuffer[250];
9
10  messageBuffer="INSERT INTO sismo_estacion (estacion,latitud,longitud,evento,"+
11      "paquete,tiempo,magnitud,toa,radio,creado) VALUES ('" + estacion +
12      "'" + "," + "'" + String(latitud_re) + "'" + "," + "'" + String(longitud_re) + "'" + "," +
13      "'" + Evt + "'" + "," + "'" + String(pqts_register) + "'" + ",NOW(),'0.0','0.0','0.0',NOW())";
14
15  messageBuffer.toCharArray(JSONmessageBuffer,250);
16  Serial.println(F("Sending message to MQTT topic.."));
17  Serial.println(JSONmessageBuffer);
18
19  if (client.publish("ID01/info", JSONmessageBuffer) == true) {
20      Serial.println(F("Success sending message"));
21  } else {
22      Serial.println(F("Error sending message"));
23  }
24
25  Serial.println(F("-----"));
26 }
--

```

Almacenamientos de datos

Por último se creó una función que ayude a almacenar los datos del evento sísmico almacenados en los buffers en una tarjeta SD, a esta función se la ha denominado *logSDCard()*, esta función es la encargada de crear un archivo *.txt*, el nombre del archivo se forma por fecha y hora (ddmmaahhmmssms) el cual se estructura de la siguiente manera: una cabecera, la cual cuenta con información básica del evento detectado y el cuerpo en el cual se listan los valores de las componentes de la señal sísmica. En la Figura 25 se muestra un ejemplo de un archivo creado por dicha función.

Figura 25*Ejemplo archivo.txt*


```

30052020114423044: Bloc de notas
Archivo Edición Formato Ver Ayuda
CICTEC - UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
Archivo de aceleración parte de la Red de sismógrafos de la ESPE
Fecha del evento UTM (ddmmaa):30052020
Hora del registro UTM (hhmmss):114535
Estación:1
Latitud:-0.43353
Longitud:-78.53970
Altitud:2782.10010
Frecuencia de muestreo (Hz): 100
Unidades: m/s^2
COMPONENTES EN Z*
-0.0219
-0.0315
-0.0123
-0.0267
0.0092
0.0308
-0.0410
0.0475
-0.0698
-0.0674
-0.0674

```

Diseño de la placa electrónica

La placa electrónica o PCB (Printed Circuit Board) se diseñó de acuerdo a los requerimientos propuestos para el prototipo de sismógrafo o estación sismográfica, para ello primeramente se verificó los módulos que van a ser incorporados a la placa y las interconexiones entre ellos, a continuación se realizó un esquema del circuito para verificar las conexiones entre los módulos. Una vez ubicado los pines de conexión entre módulos se procedió a armar el prototipo de sismógrafo en un protoboard para verificar su funcionamiento y poner a prueba tanto el hardware como el código desarrollado, una vez finalizadas las pruebas se procedió a diseñar el circuito impreso. Además se diseñó una etapa de potencia que reduzca el voltaje de 110 V a 5V que es el voltaje con el que trabajan los módulos, esta etapa es indispensable ya que la fuente de energía de la estación sismográfica será la red eléctrica, además se contempló una etapa de fuente independiente la cual será sustentada por una

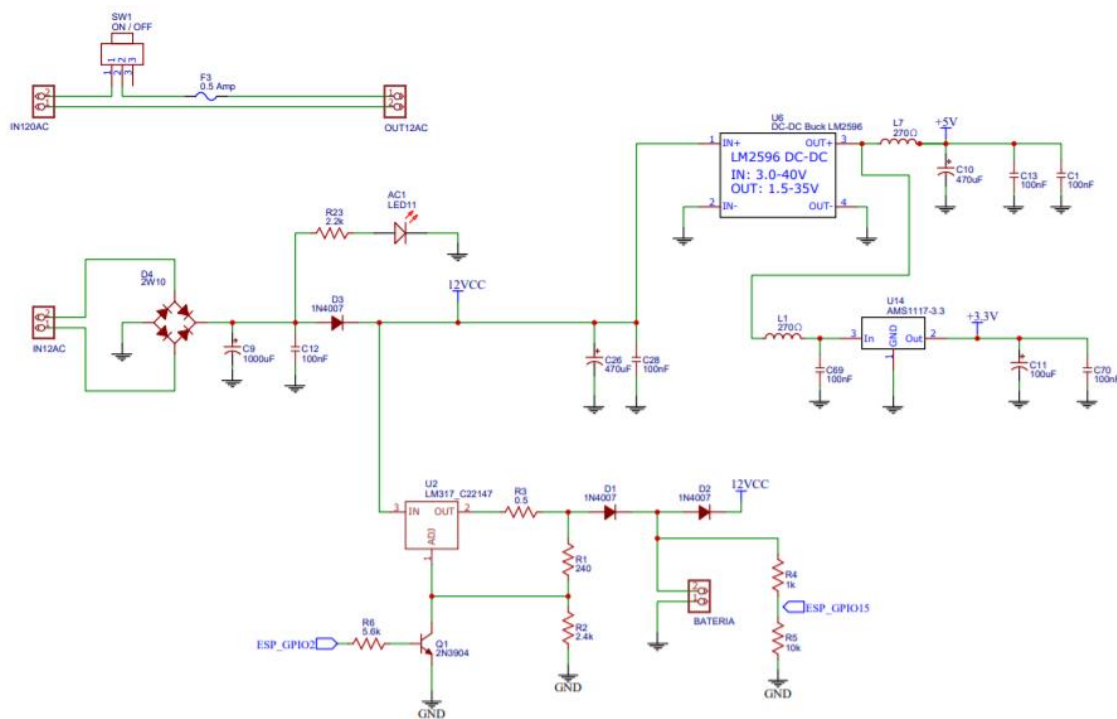
batería en caso de que se presente un fallo eléctrico o caída de energía de la red eléctrica. Esta etapa se la muestra en la Figura 26 a. Mientras que la Figura 26 b muestra la interconexión de los módulos.

Para el diseño de la PCB se utilizó el programa Proteus 8 Professional, el cual contiene la herramienta ARES, esta herramienta permite realizar el enrutado, edición y ubicación de los componentes. Previamente se debe realizar un diagrama en ISIS como se muestra en la Figura 26, lo que nos permite indicarle a ARES como los componentes están conectados en la PCB.

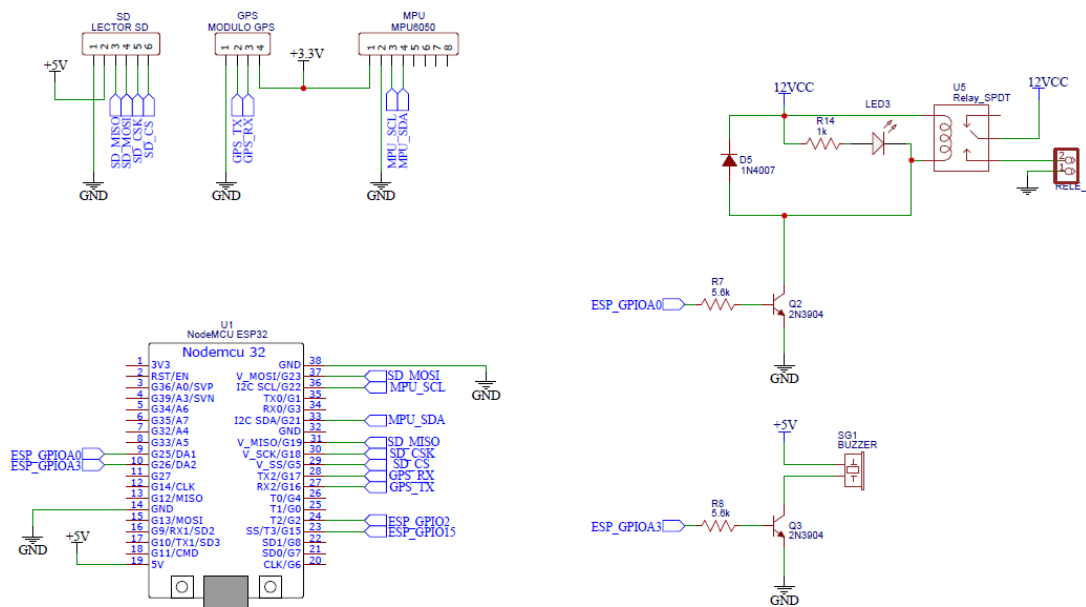
Figura 26

Esquemático del circuito del Prototipo

a)



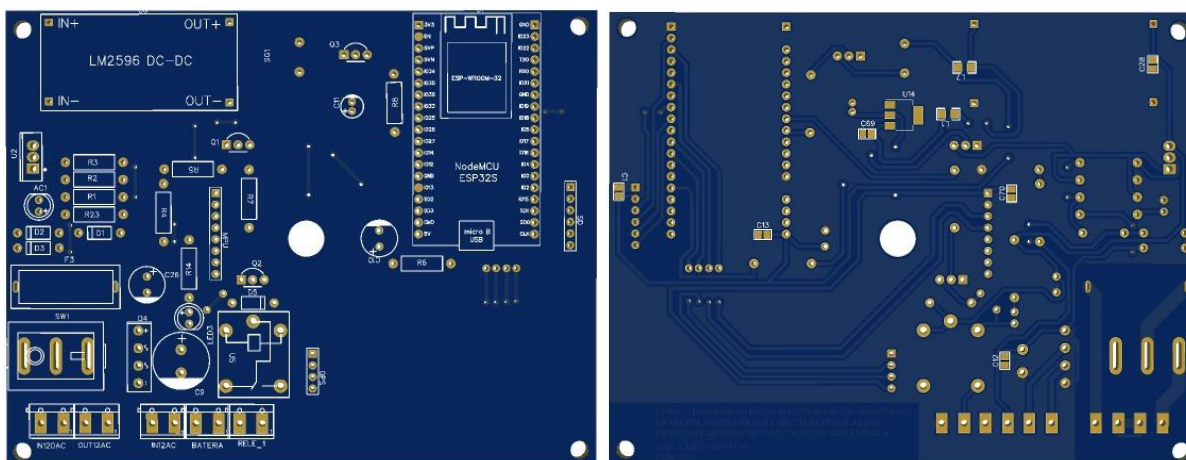
b)



Una vez terminado el diagrama en ISIS se procede al diseño de la PCB que finalmente se ve como se indica en la Figura 27.

Figura 27

PCB del prototipo de Sismógrafo.



En este capítulo se ha presentado el desarrollo del código fuente que hace posible la detección de movimientos telúricos a través del uso del algoritmo STA/LTA, además se ha realizado la comunicación inalámbrica por medio de mensajes MQTT para el intercambio de mensajes con el servidor web, también se implementó la estación sismográfica que será instalada. En el siguiente capítulo se explicará de forma detallada todo el proceso que involucró la creación del servidor web y los servicios alojados en él, los cuales permiten la comunicación con la estación sismográfica y la visualización de los datos de los eventos sísmicos registrados mediante una página web.

Capítulo IV

Desarrollo del software del sistema de monitoreo de sismos

Implementación del servidor

Los servicios de hosting están muy popularizados en la actualidad para aplicaciones relacionados con el Cloud Computing, estos permiten obtener un servidor en la nube que provee a los usuarios obtener información procesada por un servicio tipo backend, aunque existen muchos otros proveedores que ofrecen servicios donde los datos pueden ser procesados en Frontend en la computadora del usuario. Para logra los objetivos de desarrollo del proyecto se contrató el servicio de una empresa externa la cual proveyó un servidor con las siguientes características:

- Servidor linux x64
- 25 GB de almacenamiento
- 1 GB de memoria RAM
- Sistema operativo Ubuntu 16.04
- 1 núcleo de procesador

El presente proyecto se utilizó una computadora portátil personal (laptop) a la cual se le configuró como cliente para realizar una conexión SSH hacia el servidor, estas configuraciones permiten la programación, configuración y despliegue del software del proyecto generando un ambiente óptimo para la depuración, dicha computadora posee las siguientes características:

- Procesador Core i7 , Windows 10
- 1 TB de almacenamiento
- 8 GB de memoria RAM
- Sistema Operativo de 64 bits

Se implementó el servidor web “Nginx” para la administración de funciones web, este servidor proporciona funciones que permiten el envío de páginas HTML, estilos CSS y código Javascript. Cuando un cliente se conecte a dicho servidor, para que este pueda observar una página web con todas las funciones agregadas en ella con el uso de un dispositivo de navegación web. En el servidor cuando es la primera vez que se realiza la configuración y levantamiento, si todo fue hecho correctamente podremos observar una página web básica Figura 28, que nos indica que está corriendo el servicio.

Figura 28

Servidor en ejecución Nginx.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Nota. La figura presenta una página HTML básica que un cliente puede observar cuando se conecta al servidor en etapa de desarrollo.

Servicios alojados en el servidor

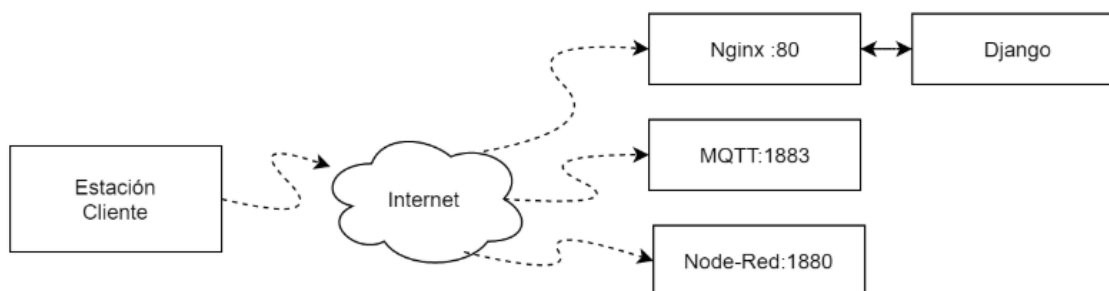
Todo servidor web dispone en su interior un conjunto de servicios que se ejecutan en todo momento para proveer a los clientes un sinfín de prestaciones. Para este proyecto se alojó varios servicios en el servidor web, a continuación se presenta una lista de dichos servicios:

- Servicio web Nginx
- Broker MQTT
- Node-Red
- Base de datos MySQL

El servicio Nginx provee páginas HTML a los usuarios que se conectan al dominio previamente configurado, básicamente presta las mismas funcionalidades del servidor Apache el cual es utilizado con más frecuencia para otros contexto de backend, dentro de este servicio se ejecuta Django para la maquetación de páginas HTML, el servicio Nginx hace el uso del puerto 80. El servicio de Broker MQTT permite el envío y recepción de mensajes ligeros los cuales permiten la comunicación entre las estaciones y el servidor, dichos mensajes contendrán datos de los sismos detectados, este servicio se ejecuta por medio del puerto 1883. Node-Red facilita el trabajo con el broker MQTT y gestiona el almacenamiento en la base de datos MySQL, el puerto 1880 es el que permite gestionar todas las funcionalidades Node-Red.

Figura 29

Conjuntos de servicios del servidor.



En la Figura 29 podemos observar un diagrama que representa a la estación que se conecta al internet para poder recibir los servicios que están alojados en el servidor; también podemos observar los nombres de los servicios y las direcciones de puertos para cada uno.

Implementación broker Mqtt

Como se había descrito en la sección anterior denominada Hardware se estableció que el prototipo de sismógrafo se comunique con el servidor a través del protocolo MQTT, por lo que se procedió a la implementación de dicho protocolo.

Para la implementación del protocolo MQTT se ha hecho uso de un broker denominado Mosquitto, este broker fue seleccionado ya que presenta las siguientes características: es de libre distribución, código abierto y principalmente puede ser incorporado en cualquier computador o un servidor web en la nube.

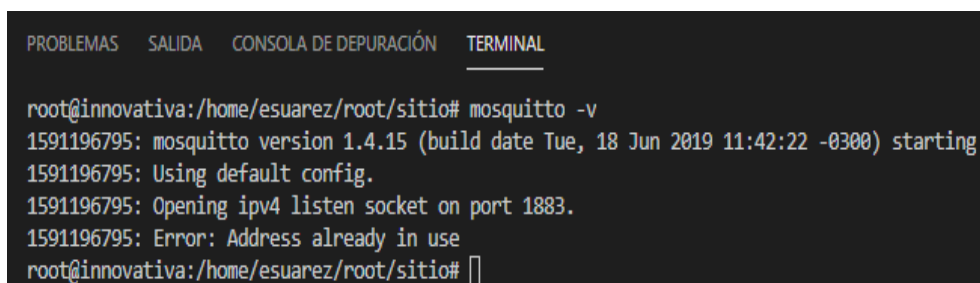
El broker es un código que se ejecuta como un servicio dentro de un servidor, cuando se inicia el sistema operativo del servidor donde se encuentra alojado el broker, este código se ejecuta de manera automática; su principal función es estar pendiente de la comunicación cliente/servidor. Cada cliente que se encuentre conectado al servidor está conectado a un hilo independiente que consume recursos del sistema, si el cliente corta la comunicación el hilo se rompe, y se libera los recursos usados.

En función de las características del servidor, se establece un número finito de clientes que pueden estar conectados al mismo tiempo, este valor es de un máximo de 65535.

Una vez alojado el broker en el servidor se establece el puerto donde este va a trabajar, en nuestro caso se activa el puerto 1883 que será usado exclusivamente para la comunicación, por medio de este puerto los clientes establecerán la comunicación para el intercambio de información.

Figura 30

Ejecución del broker Mosquito.

A terminal window with a dark background and light text. At the top, there are four tabs: 'PROBLEMAS', 'SALIDA', 'CONSOLA DE DEPURACIÓN', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal shows the following text:

```
root@innovativa:/home/esuarez/root/sitio# mosquitto -v
1591196795: mosquitto version 1.4.15 (build date Tue, 18 Jun 2019 11:42:22 -0300) starting
1591196795: Using default config.
1591196795: Opening ipv4 listen socket on port 1883.
1591196795: Error: Address already in use
root@innovativa:/home/esuarez/root/sitio#
```


En la Figura 30, se muestra la ejecución del broker Mosquito alojado en el servidor, como se puede observar se encuentra activo el socket para recibir información desde los clientes a través del puerto 1883.

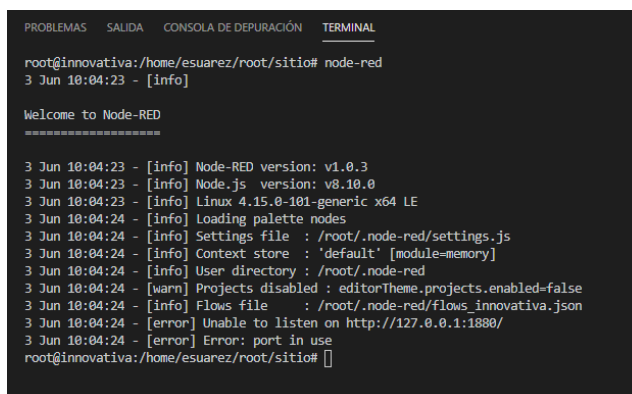
Implementación de node red

Dentro del servidor también se ha alojado el framework Node-red, el cual se ejecuta como un servicio. En este framework primeramente se han realizado configuraciones básicas tales como establecer un puerto de comunicación exclusivo en este caso el 1880. Esta herramienta nos permite programar mediante flujos las características asincrónicas que se requirieron en el presente proyecto. Las mismas que son la de poder escuchar a las estaciones remotas en tiempo de latencia mínimos, lo cual Django no permite desplegar las funcionalidades de sincronización que se ejecuten en procesos dentro del servidor denominado backend.

En la Figura 31, se muestra el estado de funcionamiento de Node-Red en el servidor y se lo hace mediante consola en el sistema operativo linux, se ha ingresado la palabra reservada node-red para desplegar las características principales de este, tales como la versión, la localización de los archivos, el puerto activo o de escucha.

Figura 31

Características implementadas en Node-Red.



```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
root@innovativa:/home/esuarez/root/sitio# node-red
3 Jun 10:04:23 - [info]

Welcome to Node-RED
=====

3 Jun 10:04:23 - [info] Node-RED version: v1.0.3
3 Jun 10:04:23 - [info] Node.js version: v8.10.0
3 Jun 10:04:23 - [info] Linux 4.15.0-101-generic x64 LE
3 Jun 10:04:24 - [info] Loading palette nodes
3 Jun 10:04:24 - [info] Settings file : /root/.node-red/settings.js
3 Jun 10:04:24 - [info] Context store : 'default' [module=memory]
3 Jun 10:04:24 - [info] User directory : /root/.node-red
3 Jun 10:04:24 - [warn] Projects disabled ; editorTheme.projects.enabled=false
3 Jun 10:04:24 - [info] Flows file : /root/.node-red/flows_innovativa.json
3 Jun 10:04:24 - [error] Unable to listen on http://127.0.0.1:1880/
3 Jun 10:04:24 - [error] Error: port in use
root@innovativa:/home/esuarez/root/sitio#
```

Adquisición de paquetes de datos por el broker y node red

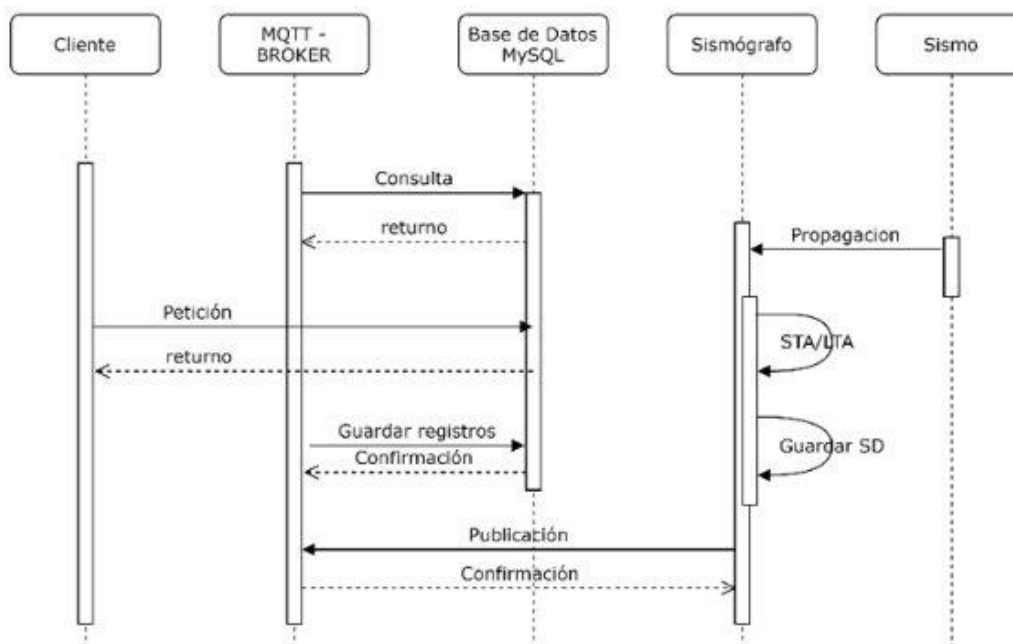
La red de monitoreo sísmico que está conformada por los prototipos sismográficos a los cuales llamaremos estaciones, las mismas realizan un monitoreo continuo de las señales sísmicas. Cuando una de las estaciones ha detectado un evento sísmico se inicia la comunicación entre la estación y el host que se encuentra en la nube. Es decir la estación empieza a enviar datos hacia el servidor, para su posterior procesamiento, entre ellos calcular la localización del epicentro, la magnitud del sismo, tiempo de arribo o TOA entre otras en un carácter aproximado.

Mediante el Diagrama UML 1 se puede visualizar a cada una de las etapas del sistema de monitoreo, es decir desde que se inicia el evento sísmico hasta que llega la información al servidor. Todo empieza cuando se genera la liberación de energía en las capas internas de la tierra, esta energía se propaga en forma de ondas hacia la superficie, cuando las ondas llegan a las estaciones, estas empiezan a procesar la información obtenida del sensor por medio del algoritmo STA/LTA, si en el proceso del evento telúrico el sistema sobrepasa el valor de umbral que se encuentra configurado en el software de la estación, el primer paso que se ejecuta es el empaquetamiento y posterior envío de los datos en estructura JSON hacia el servidor MQTT, la información que es enviada corresponden a los datos de las componentes del acelerómetro que son los ejes X,Y y Z; el envío de los datos la estación almacena los mismos datos enviados en archivo de texto plano en una memoria SD como respaldo.

Un cliente u operador es un usuario externo, este puede ser un servidor o un cliente web, este puede ingresar al sistema mediante identificación por usuario y contraseña para visualizar y procesar los datos, es decir aplicar algoritmos de cálculo de la ubicación del epicentro, cálculo de la magnitud del evento sísmico entre otros.

Diagrama UML 1

Proceso automático desde la Estación de monitoreo telúrico hasta el almacenamiento en el servidor MySQL, con uso de broker MQTT.



Desarrollo del algoritmo para detectar falsos sismos

Los eventos sísmicos son de carácter estocástico y pueden producirse en cualquier momento de manera abrupta, debido a esto las estaciones deben monitorear y analizar los datos obtenidos del acelerómetro las 24 horas del día, usando el algoritmo denominado STA/LTA, uno de los algoritmos más estables para propósitos sísmicos. Por lo que hay que tomar en cuenta que durante el transcurso del día se pueden producir vibraciones producidas por agentes externos. Debido a esta situación se vio necesario desarrollar un algoritmo en el servidor que permita discriminar entre movimientos telúricos provenientes del desplazamiento de las placas tectónicas y movimientos o vibraciones provocados por el hombre tales como: explosiones en canteras, el paso de maquinaria pesada, etc., para así evitar alertas de “falsos sismos” o “falsos positivos”, es decir prevenir que la estación alerte eventos que no son de

origen sísmico. Para garantizar la no ocurrencia de eventos provocados por el hombre es importante destacar que las estaciones se deben instalar en lugares estratégicos, alejados de la presencia humana o lugares de frecuente tránsito vehicular.

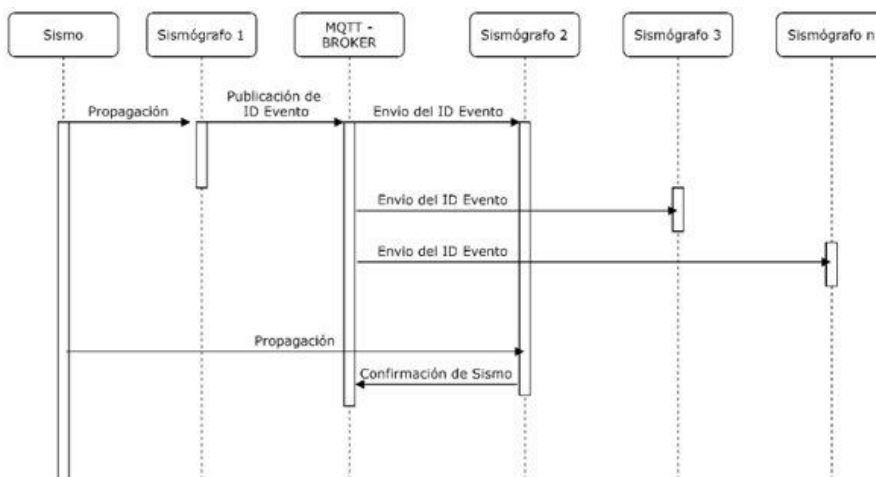
El algoritmo STA/LTA detecta el crecimiento de la aceleración procedente del sensor, estos datos si tienen una tendencia positiva en función del tiempo, llegarán a un umbral establecido, que determina que se está generando un evento de carácter sísmico pero sin tener la seguridad de que lo sea.

El prototipo de sismógrafo solo tiene la capacidad de detectar vibraciones sea cual sea su procedencia ya que se basa en el algoritmo STA/LTA el cual activa una alerta de detección de evento si se supera el umbral preestablecido. Una vez superado el umbral la estación envía el mensaje de alerta hacia el broker MQTT al topic *"IDXX/Urg"* perteneciente a Node-Red, que permite la notificación a todas la estaciones cuando una de ellas ha detectado un posible evento sísmico.

En el Diagrama UML 2 se describe el sistema de detección de falsos sismos, partiendo desde que se inicia un sismo el cual se propaga por los distintos tipos de capas hasta llegar a una estación de monitoreo sísmico que forma parte de la red de sismógrafos, la primera estación en detectar genera la alerta notificando al broker de la detección de un evento, si dentro de un lapso de tiempo establecido una segunda estación notifica una alerta al broker acerca de un evento sísmico se genera la confirmación de un evento real.

Diagrama UML 2

Interacción de un sismo con las estaciones de monitoreo sísmico para descartar falsos eventos telúricos.



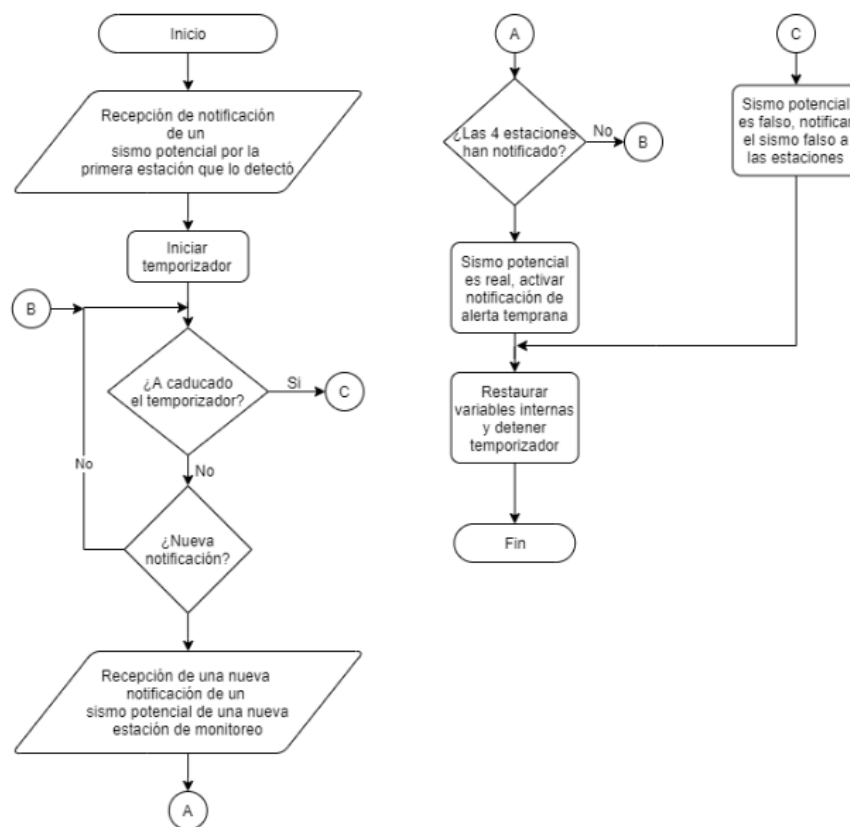
En el entorno de Node-Red están colocados los algoritmos que están pendientes de la llegada de alertas provenientes de las estaciones y cuando se disparan inician un temporizador de 20 segundos. Si otras estaciones notifican al broker que han detectado un evento sísmico dentro del tiempo establecido por el temporizador, se confirma que el evento detectado por las estaciones es un sismo real. Caso contrario si ninguna otra estación ha notificado la detección de un evento, y el temporizador se ha caducado, el servidor procede a enviar un mensaje a todas las estaciones notificando un falso positivo. El mensaje que envía el servidor para alertar de un falso sismo es el siguiente:

{Utm: 000000000, Cmd: 2, Est: 0}

Teniendo múltiples notificaciones de las distintas estaciones, podemos tener un sistema de alerta temprana ante eventos telúricos. En la Figura 32 se presenta el algoritmo desarrollado para determinar falsos sismos detectados por las estaciones de monitoreo sísmico.

Figura 32

Diagrama de flujo del algoritmo de detección de falsos sismos.



Automatización de datos

El Node-Red trabaja para poder recibir los mensajes que provienen de los clientes que son las estaciones de monitoreo telúrico y estos son procesados en las estructuras JSON para posteriormente ser guardados en la memoria del sistema. El uso de la herramienta Node-Red en el servidor permite obtener características asincrónicas dentro del sistema, lo que significa poder obtener los paquetes de información en cualquier momento.

Node-Red en el transcurso del proyecto ha sido de gran ayuda ya que una de sus características es la emulación del envío de paquetes, simulando los mensaje que envía una

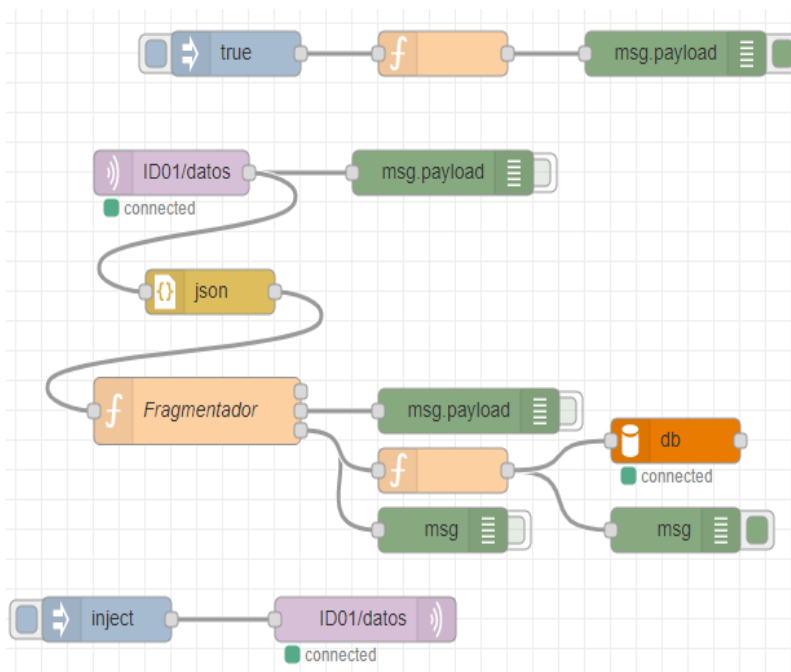
estación de monitoreo sísmico. Esto es gracias a los bloques “inject” los cuales poseen configuraciones internas que permite incorporar información similar a la que un bloque va a enviar, separada en dos contextos, el topic y payload.

El proceso de inyectar datos lo hace por medio de un topic el cual indica que son dirigidos al bloque “inject”, esta información es recibida posteriormente por bloque “ID0x/datos” que permite enviar mensajes MQTT para luego ser cargados en el bloque del payload o contenido del mensaje para así ser enviados al broker y finalmente sean almacenados en la base de datos.

Gracias a la simulación por medio del bloque inject se pudo desarrollar la codificación por flujos en node-red para la funcionalidad de adquisición de paquetes y posterior alojamiento en la memoria del host como se muestra en la Figura 33.

Figura 33

Adquisición y alojamiento de datos en la memoria del host con Node-Red.



De todo este proceso de adquisición de paquetes de información, el bloque más importante es el fragmentador localizado en la parte céntrica de la figura anterior. Este bloque contiene un función como su nombre lo indica encargada de fragmentar y reorganizar los datos internos de cada uno de los paquetes en estructura JSON recibidos, para reconstruirlos y convertirlos en sentencias, entonces enviarlos a la base de datos.

En el Segmento de código 8 se puede observar una parte de la función fragmentador, que se encarga de extraer los datos en nuevas variables del paquete recibido, correspondiente a las líneas 1 a la 6. Posteriormente se fragmentan en datos individuales en la función *for* a partir de la línea de código 12, extrayendo dato por dato la información del sismo recibida, para posteriormente entregarlos a una función que se encargará de enviar la información al sistema de almacenamiento MySQL por medio de sentencias.

Segmento de código 8

Función Fragmentador.

```

1  var evento={"payload":msg.payload.Evt.toString()};
2  var estacion={"payload":msg.payload.Est.toString()};
3  var paquete={"payload":msg.payload.Pqt.toString()};
4  var componente={"payload":msg.payload.Cmp.toString()};
5  var Time={"payload":msg.payload.Time.toString()};
6  var valores={"payload":msg.payload.Value.toString()};
7
8  Time.payload = Number(msg.payload.Time);
9
10 var m_out = [];
11 var words = msg.payload.Value.toString().split(",");
12 for (var w in words) {
13     //var T={payload:Time.payload.toFixed(4)};
14     //var V={payload:words[w]};
15     //outputMsgs.push(T.payload+" "+V.payload);
16     var newmsg={evento:evento.payload,estacion:estacion.payload,
17     m_out.push(newmsg);
18     Time.payload=Time.payload+0.010;
19     //timepo_truncado=Math.trunc(Time.payload)
20 }
21
22 return [Time,valores,m_out];

```

A continuación se realiza la formulación de sentencias que serán enviados al sistema para guardar los datos en el MySQL, esta acción se realiza en la ejecución del Segmento de

código 9, en este apartado se unifican toda la información acerca del sismo, tal como el ID del evento, nombre de la estación, el número de paquete que se está enviando, tiempo recogido de ese dato, a qué componente pertenecen dichos datos, el valor correspondiente a la aceleración y al final la referencia de tiempo expresada en hora/fecha.

Segmento de código 9

Almacenamiento de datos de las estaciones en la DB.

```

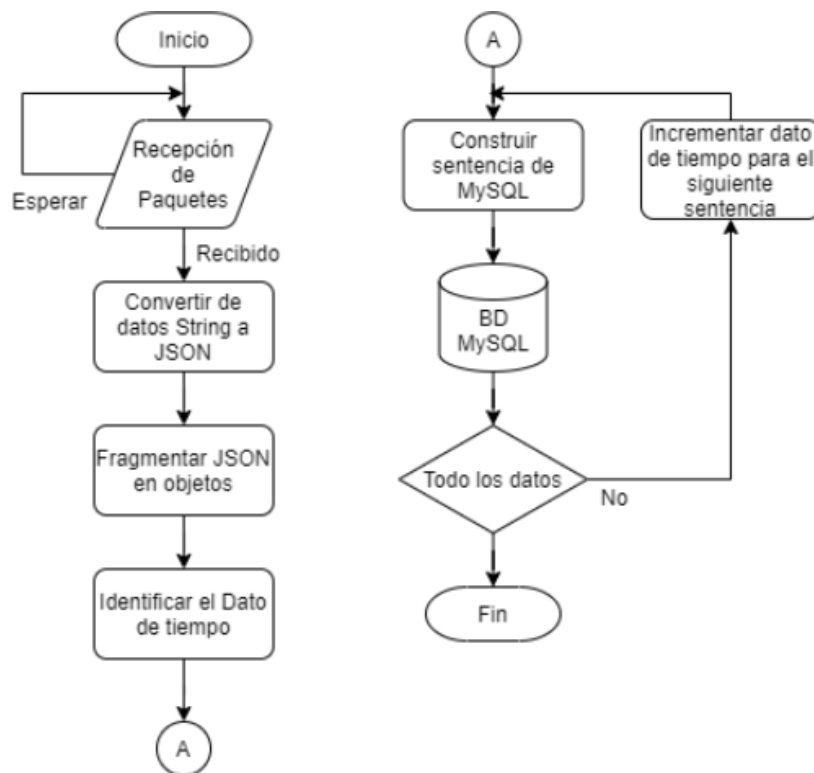
2 msg.topic="INSERT INTO sismo_data (evento,estacion,paquete,tiempo,componente,valor,creado)"
3 + "VALUES ('"+msg.evento+"','"+msg.estacion+"','"+msg.paquete+"','"+
4 + " '"+msg.payload+"','"+msg.componente+"','"+msg.valor+"',NOW())";
5 return msg;

```

El algoritmo para lograr que el servidor pueda codificar los mensajes que están en estructura JSON para a su vez enviarlos al guardar en la base de datos.

Figura 34

Diagrama de flujo del manejo JSON.

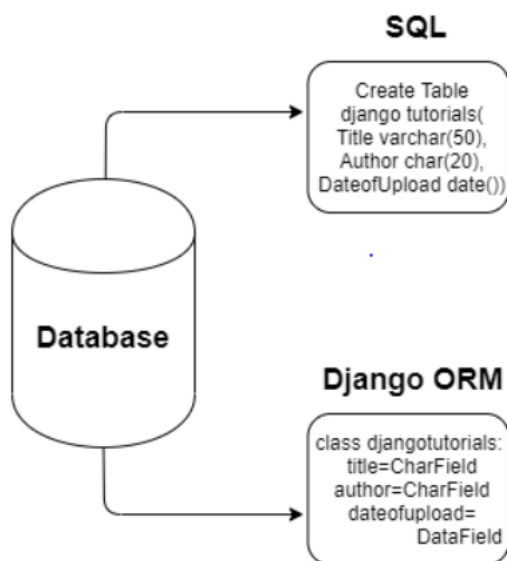


Implementación de modelos de django para la base de datos

Un modelo en Django es una clase importada de la biblioteca “*django.db*” el mismo que permite relacionar el sistema de almacenamiento con el servidor. Mediante esta clase se puede obtener los datos para ser usados en la página web, porque relaciona directamente esta estructura con la información que se encuentra almacenada. Para trabajar con modelos en Django se hace uso de la herramienta Django ORM la cual permite que por medio del lenguaje Python generar relaciones y estructuras de datos en vez de lenguaje de base de datos. (Data Flair, 2020)

Figura 35

Diferencia entre el SQL y Django ORM.

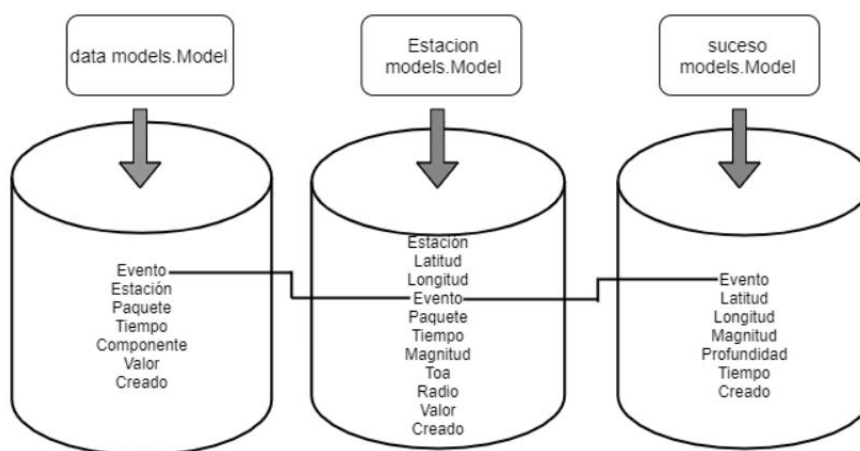


Se crearon modelos que tienen el propósito para realizar la interfaz para la interacción de la información mediante sentencias de Django las cuales permiten leer, guardar, actualizar o hasta la creación de nuevos datos. Para hacer un buen uso de los modelos, y mantener buenas prácticas en la programación del proyecto es importante tener creada la DB y dado de alta los accesos para lograr una migración exitosa.

Por lo tanto se desarrolló la base de datos MySQL con el propósito de organizar una DB (*Data Base*) relacional, esto quiere decir que los datos están ligados con identificadores a otras tablas para obtener más información. En el proyecto se dispone de tres tablas: la primera almacena los datos de los sismos detectados, la siguiente tabla datos de la estación remota y la última tabla almacena la aceleración del instrumento. En la siguiente figura se puede observar los parámetros de las tablas que se usan para guardar los eventos y los datos de las estaciones. La línea que las interconecta (línea negra) representa la comunicación entre las tablas a través del ID del evento. Con el código ID permite optimizar las operaciones para lograr mejores resultados en tiempos de búsquedas.

Figura 36

Models usados para organizar la base de datos.



Como se puede observar en la Figura anterior disponemos de tres cilindros que representan a las tablas de la DB con sus respectivos parámetros donde se almacenarán los datos según como corresponda.

Base de datos y su relación con django

Django trabaja de una manera especial con el MySQL haciendo que se use de otra manera la configuración, este mecanismo se denomina models que trabaja a través de clases y funciones. Como se puede observar en el Segmento de código 10 se realiza el llamado a las clases para leer y presentar sólo los datos de las estaciones que se encuentran guardados en la nube y al mismo tiempo se obtiene los datos de los últimos eventos que se hayan registrados.

Segmento de código 10

Envío de datos cuando al cliente.

```
def home(request):
    sucesos=suceso.objects.order_by('-creado')[:3]
    est = Estacion.objects.order_by('-creado')[:4]

    if sucesos[0].magnitud == 0.0:
        white_data = ['Sismo sin procesar']
    else:
        white_data = ['Sismo prcesado']

    args = {'sucesos':sucesos, 'est': est, 'write': white_data}

    return render(request, 'sismo/home.html', args)
```

Dentro del segmento de código 10 se indica la verificación de depuración para que se imprima información si existe datos por procesar dentro del servidor, tan solo para efectos de señalización, adicionalmente se conjuga todos los datos en un diccionario de python para ser enviado a la maquetación que se realizará en “sismo/home.html”, para que se pueda visualizar en la página web.

Implementación de las views en django

Las Views de Django permiten organizar al desarrollador funciones y URLs, ya que se puede conectar funciones a URLs específicos para cada función, por lo tanto cuando un cliente ingresa vía web al dominio del servicio web del proyecto, las URLs direccionan a las funciones correspondientes a este en las Views. En el Segmento de código 11, se presenta una Views

perteneciente a un evento para listar datos, de esta manera al ingresar en esta opción, se lanza las funciones de filtración para obtener los datos de la estaciones remotas y de los eventos registrados en la DB.

Además las Views de Django permiten trabajar de dos maneras, la primera que es la más sencilla, trabaja por medio de clases, donde todo está maquetado pero con ciertas limitaciones, por ejemplo no se permite agregar o modificar el diccionario de python con mayor cantidad de atributos, y la segunda es cuando se trabaja con funciones, mediante esta se puede realizar una mayor gestión de datos, por ellos se hizo uso de este último método para el desarrollo del proyecto.

Segmento de código 11

Función “estacionListar”.

```
def estacionListar(request, evento_id):
    est = Estacion.objects.filter(evento=evento_id)
    sucesos = suceso.objects.filter(evento=evento_id)

    args = {'sucesos': sucesos, 'est': est}

    return render(request, 'sismo/estacion_list.html', args)
```

Organización de las urls en django

Las URLs son los nombres para subdividir las distintas páginas web que el servicio web va a generar, de esta manera cuando un cliente se conecte al dominio solo deberá colocar el nombre del dominio del servidor o en su defecto la dirección IP pública y se direccionará a la página de inicio, que será generada por la función del View con el nombre de “home”. Como se puede ver reflejado en el segmento de código 12 en la tercera línea del código.

Las otras URLs que se pueden observar en el segmento de código permite que el usuario pueda ir a las demás páginas web que se va a gestionar desde la página web principal

denominado “home”. El usuario debe ingresar por medio de un navegador de internet para poder observar la página web HTML con varios botones los cuales redirigirán a los diferentes vínculos con las otras URLs que se expresan en el código.

Segmento de código 12

URLs de Django.

```
app_name = 'sismo'

urlpatterns = [
    path('', views.home, name='inicio'),
    path('sucesolistar/', views.sucesolistar, name='suceso_listar'),
    path('estacionlistar/<evento_id>/', views.estacionlistar, name='estacion_listar'),
    path('suceso_only/<evento_id>/', SucesoOnly.as_view(), name='suceso_only'),

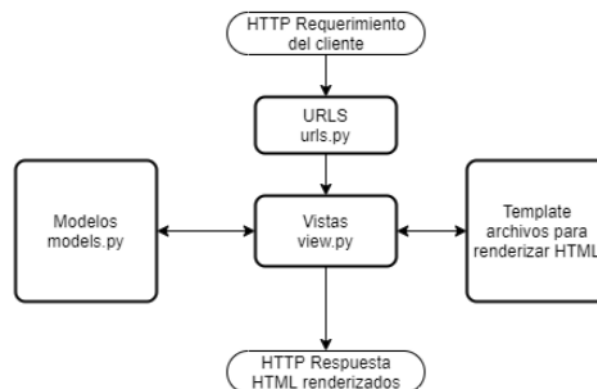
    path('listar/<evento_id>/<estacion>/', views.detalles, name='data_listar'),
    path('ver/<evento_id>/<estacion>/', views.ver, name='data_ver'),
```

Implementación de la template base del proyecto en django

Como se ha explicado anteriormente las URLs gestionan las distintas páginas web, que a su vez son maquetadas por las funciones Views de Django. Las funciones Views llaman a los archivos HTML que son la maqueta de la página web, y con la combinación de los diccionarios que contienen los datos leídos desde la base de datos MySQL.

Figura 37

URLs, Views y Template HTML.



En la Figura 37 se puede observar un diagrama de bloques que representa las etapas que se dan cuando un cliente se conecta al servidor y solicita una página en especial o tan solo la página de inicio. Las URLs están relacionadas con las Views que interaccionan con los modelos que sirven para leer los datos del almacenamiento MySQL por medio del uso de la estructura de modelos. Todos los datos que se extraen del almacenamiento se guardan en diccionarios que se envían a los Templates HTML para la maquetación o renderización. Completado estas etapas se envía al usuario un página HTML con todos los datos listos y asignados en tablas o mapas.

Incorporación de la librería matplotlib en django

La gran ventaja que presenta Django es el uso de todas las bibliotecas científicas, numéricas y analíticas de Python. “Entre ella tenemos la biblioteca Matplotlib, esta es una biblioteca de trazado 2D de Python que produce cifras de calidad de publicación en una variedad de formatos impresos y entornos interactivos en todas las plataformas.” (Cowhite Software, 2020)

Django es un entorno de desarrollo que usa lenguaje Python, por esta razón es compatible con todas las librerías que normalmente son usadas, la diferencia que existe cuando se trabaja con el entorno Django es el proceso de incorporación, ya que no solo se debe realizar la instalación, también se debe incorporar la biblioteca en los Apps del servidor en un ámbito global.

En el segmento de código 13, se muestra la incorporación de la librería en el servidor mediante la palabra reservada “*django_matplotlib*” para usarlo de la forma estándar con el propósito de renderizar gráficos de los análisis que se realicen, en este caso de los eventos sísmicos.

Segmento de código 13

Librería matplotlib.

```

46 INSTALLED_APPS = [
47     'django.contrib.admin',
48     'django.contrib.auth',
49     'django.contrib.contenttypes',
50     'django.contrib.sessions',
51     'django.contrib.messages',
52     'django.contrib.staticfiles',
53     'sismo',
54     'django_matplotlib',
55 ]

```

Preprocesamiento de datos en django

El frameworks de Django posee una variedad de herramientas para trabajos destinados al análisis de datos, gracias a que este lenguaje tiene una sintaxis muy asimilable y su codificación es análoga al Matlab o R.

Para procesar la información de un evento telúrico en el frameworks de Django debemos leer los datos almacenados, como se puede observar en el siguiente Segmento de código.

Segmento de código 14

Codificación y almacenamiento.

```

raw_values_e = data.objects.filter(evento=evento_id, estacion=estacion, componente='e').order_by('tiempo')
squares = [dato.valor for dato in raw_values_e]
raw_data_e = []
for d in squares:
    raw_data_e.append(float(d))

raw_values_n = data.objects.filter(evento=evento_id, estacion=estacion, componente='n')
squares = [dato.valor for dato in raw_values_n]
raw_data_n = []
for d in squares:
    raw_data_n.append(float(d))

raw_values_z = data.objects.filter(evento=evento_id, estacion=estacion, componente='z')
squares = [dato.valor for dato in raw_values_z]
raw_data_z = []
for d in squares:
    raw_data_z.append(float(d))

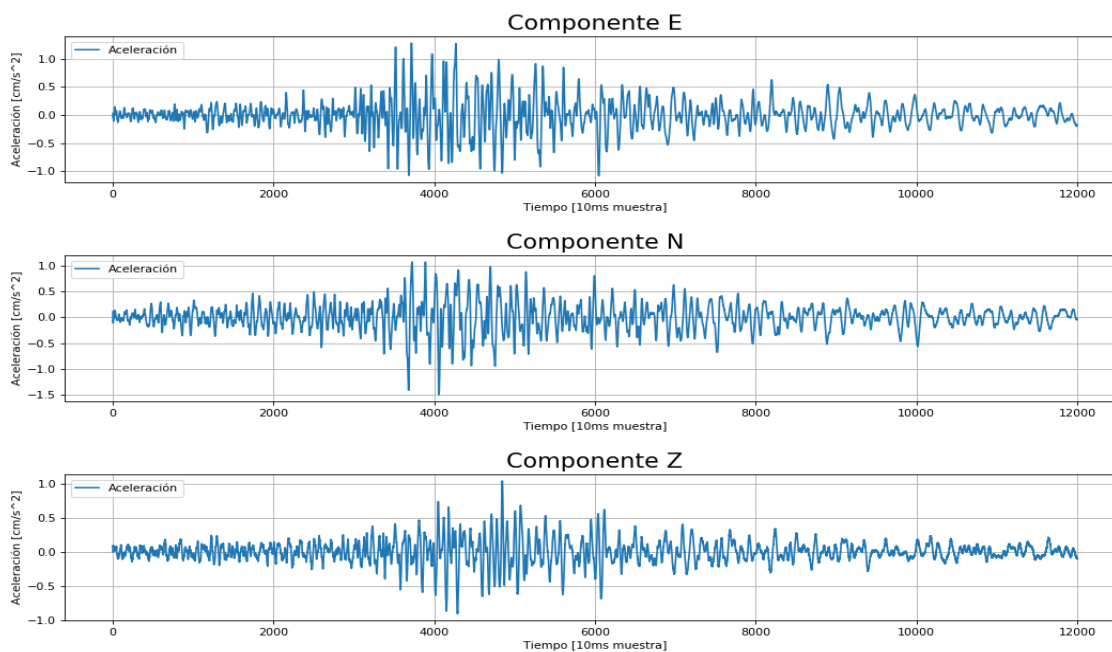
```


Las líneas de código que se muestran anteriormente permiten generar las sentencias de MySQL de manera automática, para la búsqueda de los datos correspondientes a los valores de cada componente de un sismo mediante el código del evento deseado. Una vez extraído los datos estos se grafican en una interfaz 2D mediante la biblioteca Matplotlib. Estas gráficas representan *aceleración vs tiempo* por lo que se encuentra en escala de $c\ m/s^2$ vs s , ya que son los mismo datos enviados desde la estación y como se había mencionado en secciones anteriores los datos que envía la estación hacia el servidor se encuentran en unidades de $c\ m/s^2$ y el tiempo de muestreo es de 100 muestras por segundo.

En la Figura 38 se puede observar los datos de las componentes N, E, Z de un sismo que se tenía almacenado en el MySQL.

Figura 38

Aceleración vs Tiempo.

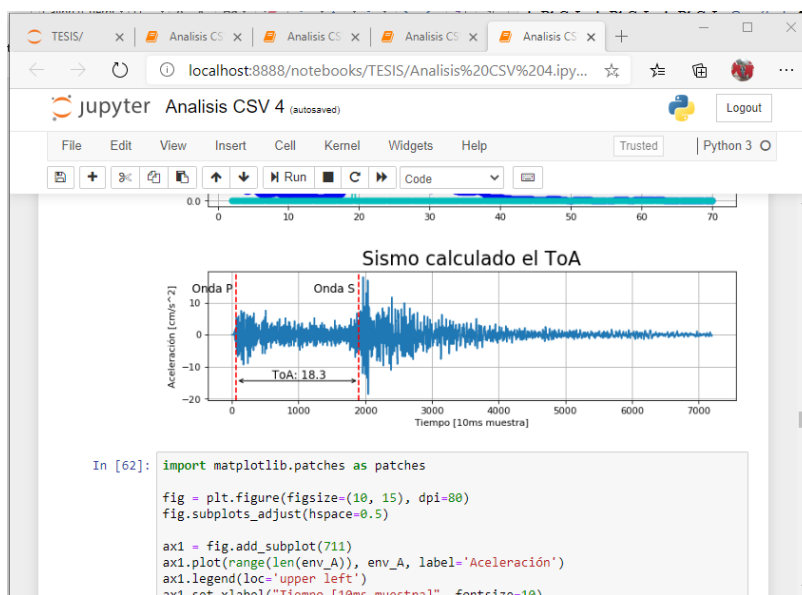


Python y jupyter para la depuración de algoritmos

Existen diferentes IDE para desarrollar algoritmos en Python como por ejemplo VisualCode, PyCharm entre otros, pero uno de los más eficientes para realizar depuración de manera rápida es Jupyter. Esta herramienta permite fraccionar el código de desarrollo y evaluarlo por bloques para determinar las alternativas más favorables para lograr el objetivo deseado. En la siguiente figura se puede observar el entorno de Jupyter donde se han procesado datos de sismos y evaluado los resultados antes de ser llevados al servidor web.

Figura 39

Entorno gráfico de Jupyter.



Algoritmos para procesar datos de eventos telúricos de base datos conocidas.

La información de los eventos telúricos que se evaluaron en los algoritmos fueron obtenidos de la página web oficial del Instituto Geofísico de la Escuela Politécnica Nacional (IGEPN, s.f.), dichos datos son proporcionados a través de esta página mediante archivos planos con extensión .txt los cuales son de libre acceso. En la Figura 40 se presenta una captura del archivo de texto plano, que contiene la información de una estación monitoreo telúrico que

forma parte de la estructura interconectada de acelerógrafos del IGEPN, el formato de este archivo dispone de un encabezado en el cual se describe los datos de la estación como: ID del evento, fecha del evento, hora del registro, código de la estación, componente almacenada, frecuencia de muestreo y las unidades en las que están almacenados los datos.

Figura 40

Formato del archivo plano .txt de las estación ALIB.

```

E_201605180700_ALIB_5R53E2_100.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Archivo de aceleración parte de la Red Nacional de Acelerógrafos (RENAC) Ecuador
Instituto Geofísico - Escuela Politécnica Nacional (IG-EPN)

Evento: 201605180757
Fecha del evento UTM (aammdd):                2016    5    18
Hora del registro UTM (hhmmss):  7    58 0.00
Estación:                                     ALIB
Componente:                                    E
Frecuencia de muestreo (Hz):                1.000000e+02
Unidades: cm/s^2

-0.0273 -0.0292 -0.0273 -0.0343 -0.0317 -0.0506
-0.0581 -0.0751 -0.0914 -0.0971 -0.1053 -0.1072
-0.1002 -0.0921 -0.0745 -0.0594 -0.0374 -0.0079
0.0123 0.0468 0.0745 0.1015 0.1210 0.1398
0.1442 0.1467 0.1379 0.1285 0.1090 0.0858
0.0644 0.0393 0.0261 0.0160 0.0078 0.0066
0 0110 0 0111 0 0251 0 0286 0 0187 0 0556

```

Se ha desarrollado un algoritmo que que lea el formato del archivo .txt para la realización de las diferentes fases requeridas por el proyecto, ya que se ha tomado estos datos para la comprobación del correcto funcionamiento de los algoritmos. Dicho algoritmo se basa en la localización del primer dato de la aceleración, para seguidamente ir extrayendo dato por dato y colocarlo en un nuevo arreglo de memoria, para finalmente almacenarlo en un formato más compacto y organizado, que en este caso archivos con extensión .csv, ya que este genera las mejores prestaciones para organizar los datos, como se puede observar en la Figura 41. Este tipo de extensión permite trabajar de mejor manera con el lenguaje de programación que estamos usando denominado python, que está muy generalizado para aplicaciones de Machine Learning y Big Data. De esta manera cada vez que se requiere obtener los datos, ya no será necesario abrir los archivos individuales, sino que ya estaría compactado en un solo archivo,

organizado en columnas para una rápida lectura y así realizar evaluaciones de los diferentes algoritmos.

Figura 41

Formato del archivo .csv.

	A	B	C	D	E	F	G	H	I
1	#	ALIB_E	ALIB_N	ALIB	AMIL_E	AMIL_N	AMIL	ASDO_E	ASDO_N
2	1	0.0273	0.1112	0	0.0345	0.0361	0	0.7944	3.9746
3	2	-0.0292	-0.1017	0.0927	-0.0433	-0.0645	0.0584	-2.2184	3.8681
4	3	-0.0273	-0.1061	0.094	-0.0613	-0.1064	0.0876	-2.4556	3.7219
5	4	-0.0343	-0.0948	0.0946	-0.0859	-0.1524	0.096	-2.3837	3.7742
6	5	-0.0317	-0.0954	0.0915	-0.1017	-0.1858	0.0873	-2.0601	3.6374
7	6	-0.0506	-0.0973	0.0946	-0.1077	-0.2157	0.1072	-1.4835	3.0822
8	7	-0.0581	-0.0784	0.0933	-0.1036	-0.2176	0.11	-0.7479	2.2943
9	8	-0.0751	-0.0702	0.0733	-0.1128	-0.2261	0.1023	0.0068	1.6326
10	9	-0.0914	-0.0621	0.082	-0.1156	-0.2305	0.069	0.5848	1.1876
11	10	-0.0971	-0.0488	0.0921	-0.1159	-0.2214	0.0143	1.0902	0.7616
12	11	-0.1053	-0.0394	0.0726	-0.0964	-0.188	-0.0122	1.4335	0.4225
13	12	-0.1072	-0.035	0.0833	-0.0724	-0.1398	-0.0044	1.6007	0.4395
14	13	-0.1002	-0.0274	0.0833	-0.0484	-0.1061	0.051	1.6287	0.5624
15	14	-0.0921	-0.0237	0.0701	-0.0279	-0.0758	0.1107	1.4024	0.5196
16	15	-0.0745	-0.0148	0.0802	-0.0093	-0.0453	0.1676	0.8976	0.3778
17	16	-0.0594	-0.018	0.0726	0.0276	-0.0084	0.1837	0.2141	0.0954

Estructura de datos json para datos enviados por la estación de monitoreo

JSON es un formato de estructura de datos, este formato fue escogido por su práctica y rápida implementación, como se puede observar en la Figura 42.a se presenta el formato de comunicación de las estaciones al servidor web para el almacenamiento de datos, cada estructura JSON permite enviar hasta 50 datos del acelerómetro que se han procesado por la estación remota de monitoreo sísmico. La estructura permite enviar un encabezado con la identificación del evento, código de la estación, número del paquete, tipo de componente, tiempo del evento y un vector con los datos del acelerómetro con una longitud limitada de 50 datos, por cuestiones de memoria RAM de la estación de monitoreo sísmico.

Figura 42

Estructura JSON del acelerómetro.

a)	<pre> 1 { 2 "Evt": "030310003", 3 "Est": "1", 4 "Pqt": "1", 5 "Cmp": "n", 6 "Time": "151035.845", 7 "Value": [] 2620 }</pre>	b)	<pre> 8 0.1112, 9 -0.1017, 10 -0.1061, 11 -0.0948, 12 -0.0954, 13 -0.0973, 14 -0.0784, 15 -0.0702, 16 -0.0621,</pre>
----	---	----	--

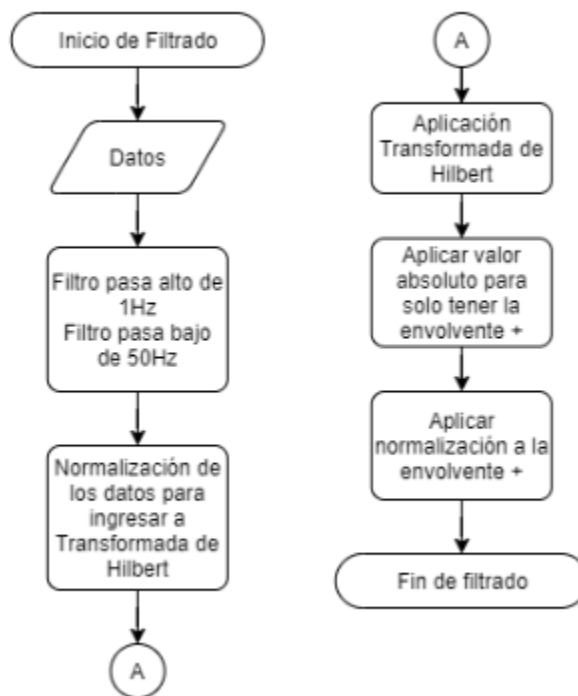
Filtrado de datos del sismo

Hay que tomar en cuenta que lo más importante para poder analizar valores proveniente de instrumentación es las fases de filtrado, que de manera fundamental disminuyen el ruido que se solapan en el sistema.

El algoritmo para el proceso de filtrado de datos telúricos se representa mediante el diagrama de la Figura 43, el proceso inicia con la extracción de datos desde el sistema de almacenamiento y esa información se la coloca en la memoria RAM en un array de una dimensión para someterlo al proceso de filtración. Los datos deben pasar a un formato denominado punto flotante por sus características de numeración con decimales. Los datos que se pueden procesar deben tener una longitud de 7000 a 12000 valores que el servidor procesará.

Figura 43

Algoritmo de tratamiento de los datos.



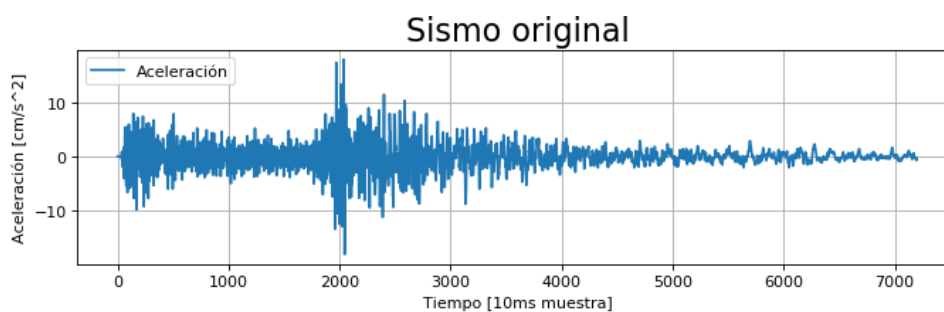
El algoritmo de filtrado primeramente aplica un filtro pasa bajo y un filtro pasa alto a los datos. En síntesis es un proceso de filtro pasa banda para obtener los datos sin la incorporación de ruido según el artículo científico titulado “*Procesamiento digital de señales sísmicas en entorno MATLAB*” de (Jiménez, 2005) quien afirma que existe ruido sísmico en el rango de 0.1 hasta 1 Hz que se puede eliminar con un filtro pasa alto, además según el país donde esté instalado la estación de monitoreo sísmicos puede existir filtración de armónicos que provengan de la red de suministro eléctrico, que puede ser de 50 Hz o 60 Hz según el país. En este caso amerita usar un filtro pasa bajo, con una frecuencia de corte por debajo de los 60 Hz ya que esta frecuencia es la usada para la generación eléctrica y además se debe tomar en cuenta que los datos de un evento telúricos que estén en el rango superior a los 40 Hz tienen escasez de información relevante para el análisis, por lo que se establece el valor de 50 Hz para el filtro ya

que es un valor menor al de la red eléctrica y superior al rango donde disminuye la información sísmica.

Para el desarrollo del algoritmo y comprobar su correcto funcionamiento se hizo uso de la información sísmica obtenida de la página web del (IGEPN, s.f.) cuyos datos son de uso público para fomentar la investigación de este tipo de fenómenos, los datos están expresado en unidad de centímetros sobre segundo al cuadrado con una velocidad de muestreo de 100 Hz como se puede observar en la Figura 44.

Figura 44

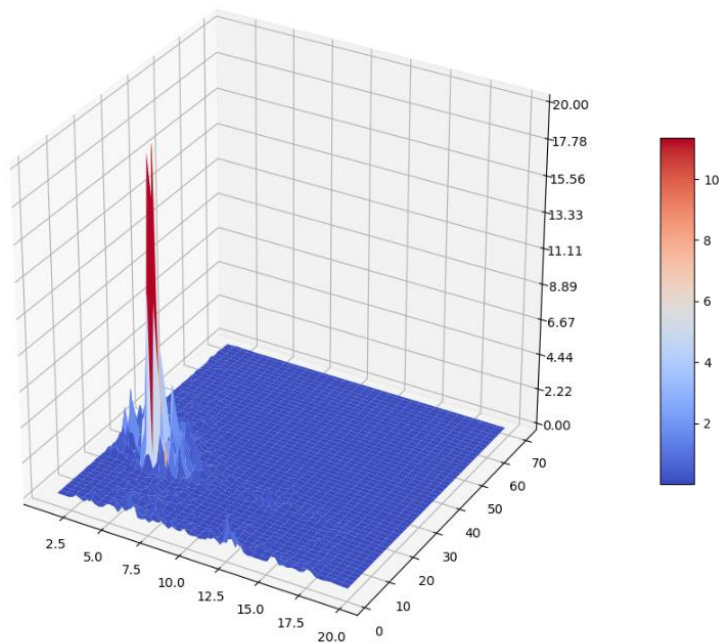
Señal original del sismo del 7 de septiembre del 2018.



Con el uso del espectrograma se puede representar en un gráfico tridimensional de la energía de una señal telúrica, que en este proyecto es la información del instrumento que mide la aceleración cuando detecta movimientos en sus distintos ejes. En la Figura 45 se muestra la representación tridimensional de un evento sísmico detectado por la estación AGYE perteneciente a la IGEPN, que esta instalada en la ciudad de Guayaquil a una distancia epicentral de 98.63 Km en este caso particular; el gráfico corresponde a los datos de la componente Z, ya que es la componente que se analiza para propósitos del proyecto.

Figura 45

Espectrograma del sismo del 7 de septiembre del 2018.

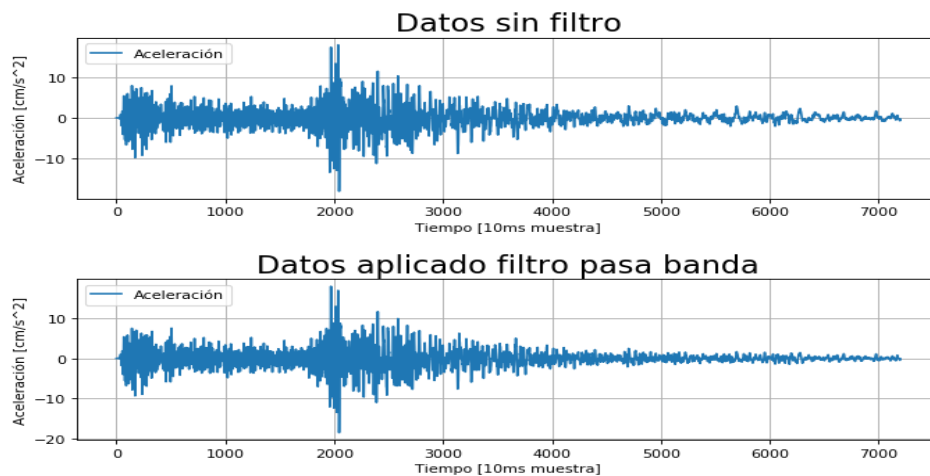


Al analizar la gráfica anterior del espectrograma de la señal se puede determinar que carece de energía en las frecuencias superiores a los 20 Hz, con lo cual se puede afirmar la teoría planteada por (Jiménez, 2005), acerca de las frecuencias que se debe usar en el filtro pasa banda.

Después de realizar el primer proceso de filtrado se obtienen los siguientes resultados, como se puede observar al aplicar el filtro pasa banda a la señal de la Figura 46 a se obtiene una disminución del ruido y la señal mostrada en la Figura 46 b, en la cual se puede evidenciar claramente la disminución del ruido especialmente en la zona del coda del sismo, que es el área donde empieza a disminuir la energía del movimiento telúrico, pero logrando mantener la misma información en la zona de la onda primaria y secundaria.

Figura 46

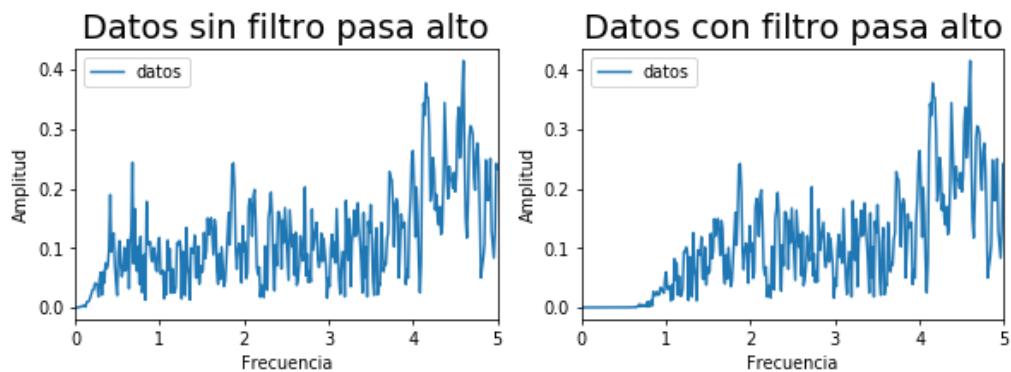
Señal del sismo y filtración.



En la Figura 47 se presentan los espectrogramas en dos dimensiones de la señal antes y después de aplicar el filtro pasa alto; con lo cual se demuestra la capacidad de eliminar las frecuencias menores a 1 Hz denominado ruido sísmico.

Figura 47

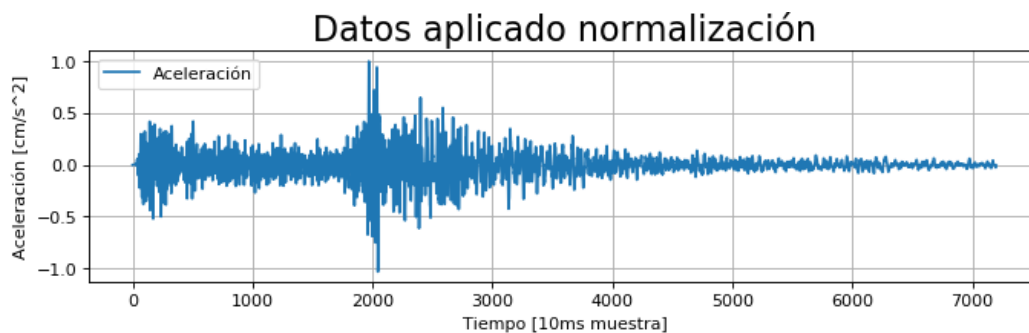
Espectrograma y filtros.



Una vez filtrada la señal se aplica una normalización a la señal resultante para mantener los datos a una escala teórica común como se ilustra en la Figura 48.

Figura 48

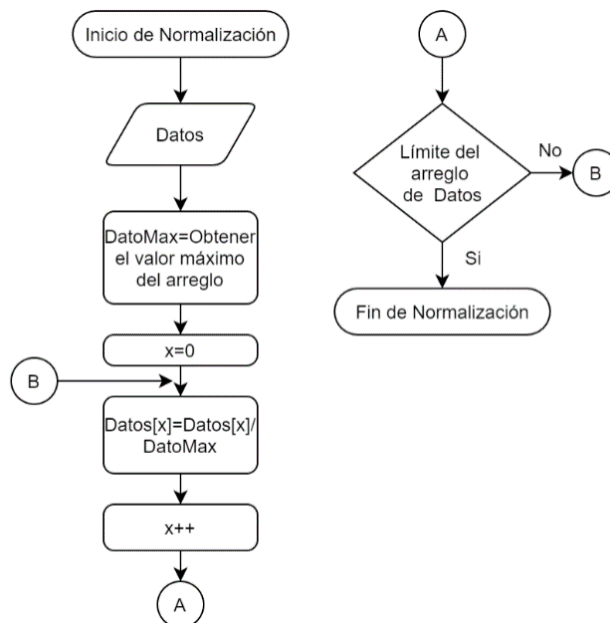
Normalización de la señal.



Con la ayuda del diagrama de la Figura 49 se presenta el proceso para lograr la normalización de una señal, la normalización consiste en extraer el valor más alto, y usarlo para reajustar la escala a un nuevo rango contemplado en un límite inferior de -1 y superior de 1, dando como resultado un nuevo gráfico que mantiene la misma proporcionalidad pero ahora en una escala que facilita el análisis.

Figura 49

Algoritmo de normalización.



Seguidamente de la normalización se aplica la transformada de Hilbert que es una importante herramienta matemática para obtener una función envolvente de la señal sísmica que se está analizando.

$$H(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(u)}{u-t} du = -\frac{1}{\pi t} * f(t) \quad (7)$$

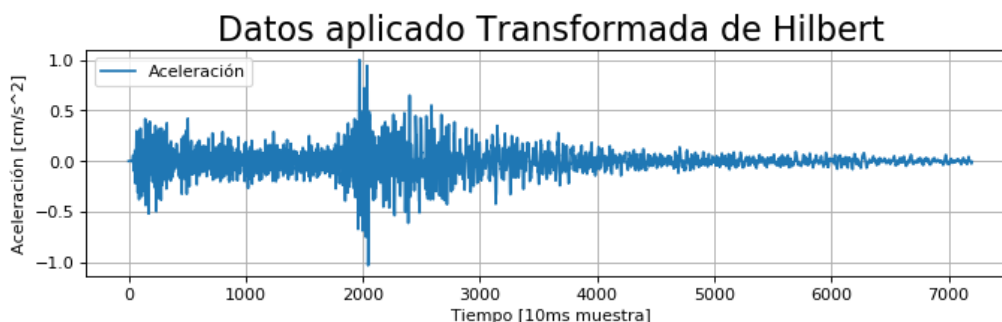
La ecuación 7 representa la Función de Hilbert, esta ecuación se la usa aplicando al espectro de la función que corresponde a la Transformada de Fourier como podemos ver en la ecuación 8. La transformada de Hilbert permite utilizar las amplitud del espectro en la misma proporción a la señal, pero con un desfase de $\pm\pi/2$ según el signo.

$$\mathfrak{F}\left\{-\frac{1}{\pi t}\right\} = i \operatorname{sgn}(s) \quad (8)$$

En la Figura 50 se presenta el resultado de ejecutar la transformada de Hilbert, tomando en cuenta que los datos están en un ámbito positivo y negativo. Debemos aplicar nuevamente la normalización de estos datos antes de seguir con el siguiente proceso porque existen picos que se desbordan de los límites.

Figura 50

Transformada de Hilbert resultante.

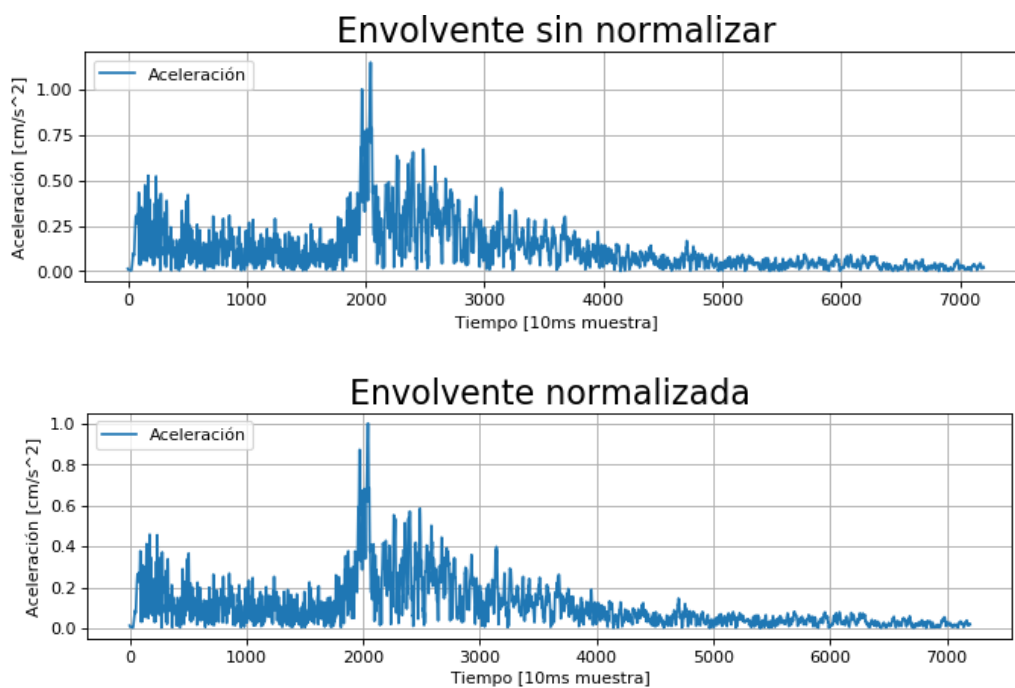


Una vez finalizado el proceso para obtener la envolvente de una señal se aplica el algoritmo de valor absoluto a todos el vector que contiene la señal para obtener datos que se

puedan analizar, obteniendo como resultado una señal envolvente que se presenta en la Figura 51 a. Pero hay que tomar en cuenta que los datos deben estar dentro del rango de la normalización, en este caso es necesario ejecutar una segunda normalización para obtener la envolvente final, la cual ya se encuentra dentro de la escala deseada como se puede observar en la Figura 51 b. Y se pueden realizar algoritmos posteriores a esta señal como por ejemplo la detección de la onda Primária (P).

Figura 51

Envolvente de la señal sísmica.



Todo el proceso anteriormente descrito fue desarrollado en lenguaje python en el cual al ejecutar un archivo de código también llamado script, el mismo que puede contener pocas líneas de código permitiendo el análisis de gran cantidad de información brindando gran rapidez al contexto de análisis telúricos.

Algoritmo para detectar la onda primaria

Los algoritmos que se usan en la actualidad son a menudo métodos de disparo que están enfocados a la discriminación de ondas de cuerpo de campo cercano, están asociadas en dos métodos que son muy utilizados. El primer método es usado para analizar y distinguir la energía cinética, que se obtiene de la estación de monitoreo sísmico, este proceso discrimina los datos y obtiene los valores para generar el disparo (Peña, 2014).

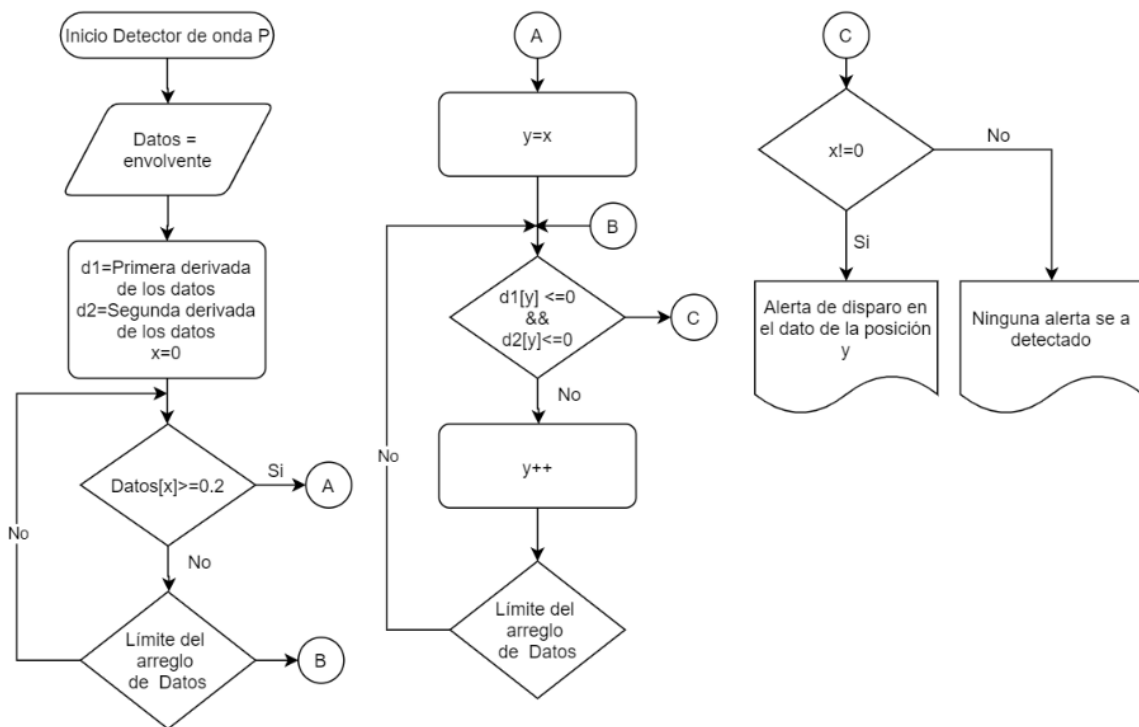
El análisis morfológico de las ondas telúricas es el segundo método, y esta metodología analiza los datos obtenidos enfocándose en el comportamiento de sus pendientes, velocidad de cambio entre otros parámetros. A los mencionados métodos se los denomina discriminadores de base que permiten posteriormente escalar a procesos de mayor complejidad como pueden ser incluir algoritmos con redes neuronales, el análisis de patrones entre otros. También con estos discriminadores de base permite lograr que un programa pueda desarrollar un aprendizaje basado en el contexto de que surjan patrones repetitivos, para desembocar finalmente en el disparo (Peña, 2014).

Los métodos de energía han sido muy popularizados ya que consisten en algoritmos que requieren menos recursos de procesamiento y generan mayor seguridad al detectar y producir el disparo, se basan en procesar el cuadrado de la velocidad en la información que se está ingresando de la estación. Este algoritmo se lo denomina STA/LTA (Bormann, 2002).

El algoritmo empieza obteniendo un arreglo unidimensional con los datos de la envolvente previamente procesada en la sección anterior. Según (Jiménez, 2005) debemos analizar la envolvente normalizada para poder determinar la fase P. En la figura 52 se puede visualizar el comportamiento del conjunto de instrucciones para la identificación de la onda P.

Figura 52

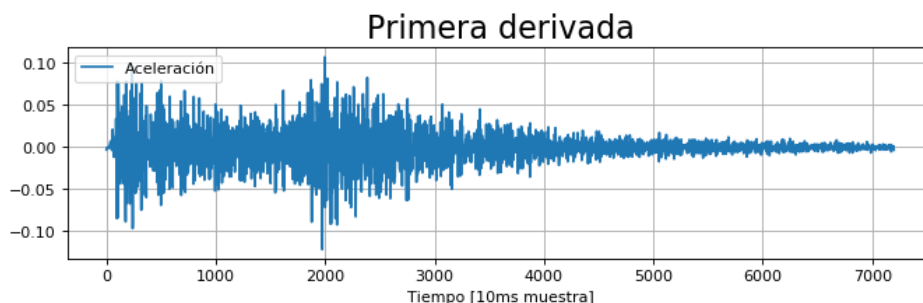
Algoritmo para la detección de la onda P.



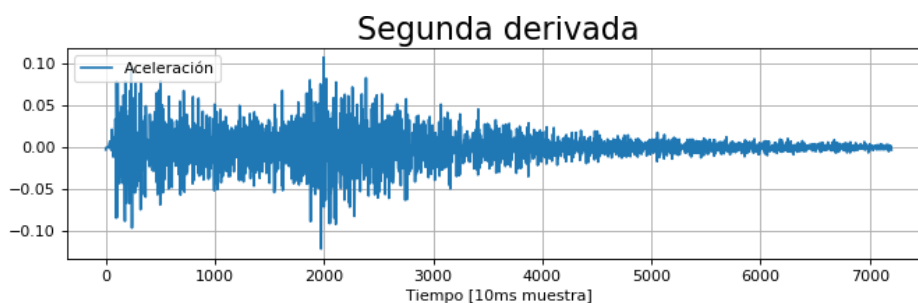
A los datos de la envolverte del sismo, se aplica la primera y segunda derivada mediante la librería Math del entorno de python, posteriormente se guarda en nuevos arreglos de memoria para usarlos, denominados $d1$ y $d2$ respectivamente. En las siguientes figuras se presenta la primera derivada bajo un algoritmo de discretización a los datos de la envolverte de la señal telúrica; la Figura 53 corresponde a la primera derivada mientras que la Figura 54 representa la segunda derivada a la función envolverte.

Figura 53

Resultado de la primera derivada de la envolvente.

**Figura 54**

Segunda derivada de la envolvente.

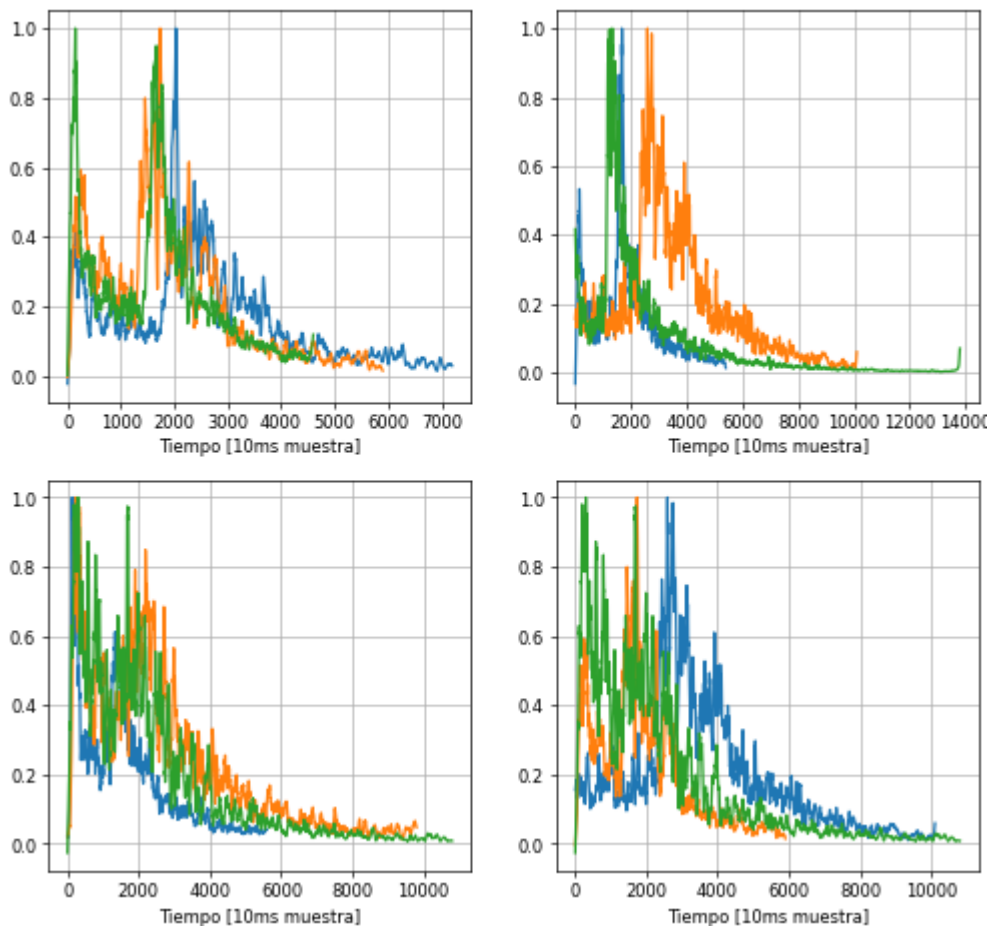


Los valores usados para determinar los disparos en la detección de un potencial inicio de la fase P de una onda sísmica debe ser calculado de manera experimental, y ajustarse a las características de la zona si se va a usar el método de análisis de energía cinética (Toledo, 2014).

Se usaron los datos del IGEPN, donde se analizó 50 archivos de ondas sísmicas provenientes de 30 estaciones de monitoreo sísmicos que conforma una conjunto de equipos interconectados denominada Red Nacional de Acelerógrafos (RENAC) y los acelerógrafos del proyecto de Oleoducto de Crudo Pesados (OCP). Para efectos de representación se organizaron 12 señales telúricas en la Figura 55 donde cada subgráfico representa tres señales con sus respectivo color para que no exista solapamiento de los gráficos. Las señales graficadas son las envolventes procesadas por la etapa de filtración.

Figura 55

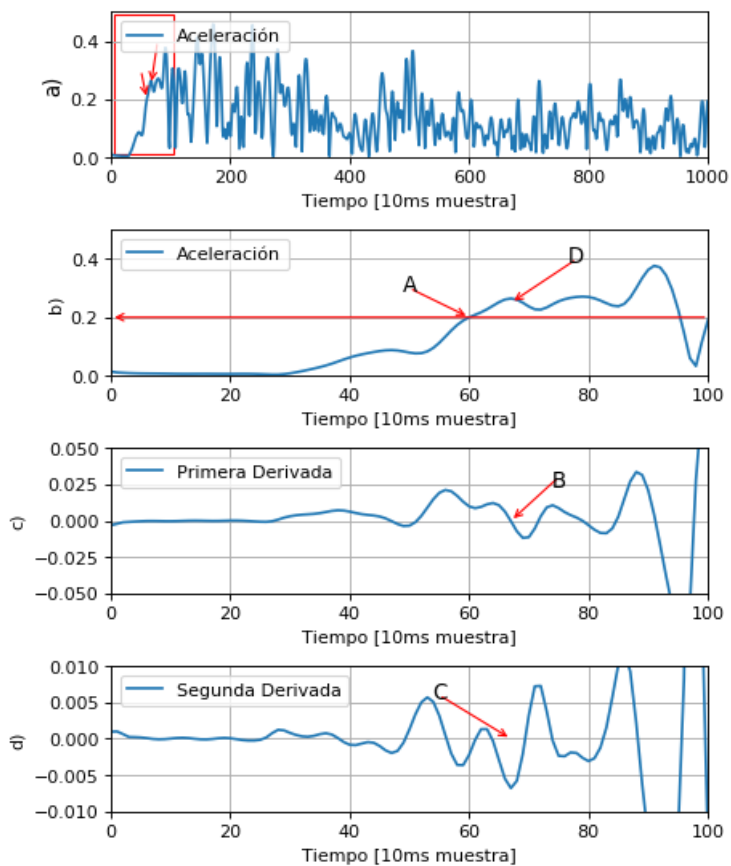
Comparativo de diferentes envolventes.



En base al análisis se determinó que un valor de umbral apropiado para el disparo de la fase P sería de 0.2 unidades, porque la señal concuerda con el punto de inflexión cuando la energía cinética empieza a elevarse y además se relaciona con la presencia de la onda P. Por lo tanto cuando una señal telúrica llega al servidor, y pasa por el proceso de filtración y posteriormente se obtiene la envolvente de la misma, este arreglo de datos es sometido a una evaluación para determinar si supera o no el umbral establecido para generar el disparo.

Figura 56

Análisis de la envolvente.



El funcionamiento del algoritmo para la detección de la onda P o también llamada onda primaria de un evento telúrico pasa por varias fases; en la Figura 56.a se visualiza una señal denominada envolvente que como su nombre lo indica es una función que abarca todos los puntos máximos locales, en la figura 56 b se puede observar un área ampliada de la gráfica donde se presentan los puntos más importantes donde es notorio el crecimiento de la energía para el análisis, como se puede observar están dentro de los primeros 100 datos de la envolvente de la señal sísmica, mediante una línea roja se representa el umbral de disparo, y adicionalmente la zona donde cruza el umbral, se ha señalado con la letra “A”.

En la figura 52 se presenta el algoritmo que permite la detección y está representado en un diagrama de flujo, donde se puede comprender cómo se realiza el análisis de la información de la envolvente para poder identificar al punto de información que supere el valor de umbral que fue establecido. De forma inmediata se ejecutan los cálculos para obtener la primera y segunda derivada, en primer lugar se empieza a identificar y ubicar a los datos que cumplan la característica de ser menor o igual a cero, en las siguientes figuras 56 c y 56 d se presentan las zonas señaladas con la letra "B" y "C" que señalan a los datos con signo negativo. El dato que posea un valor mayor se presenta en la figura 56 a correspondiente a la zona señalada con la letra "D", dando respuesta a la localización de la onda P que es el objetivo.

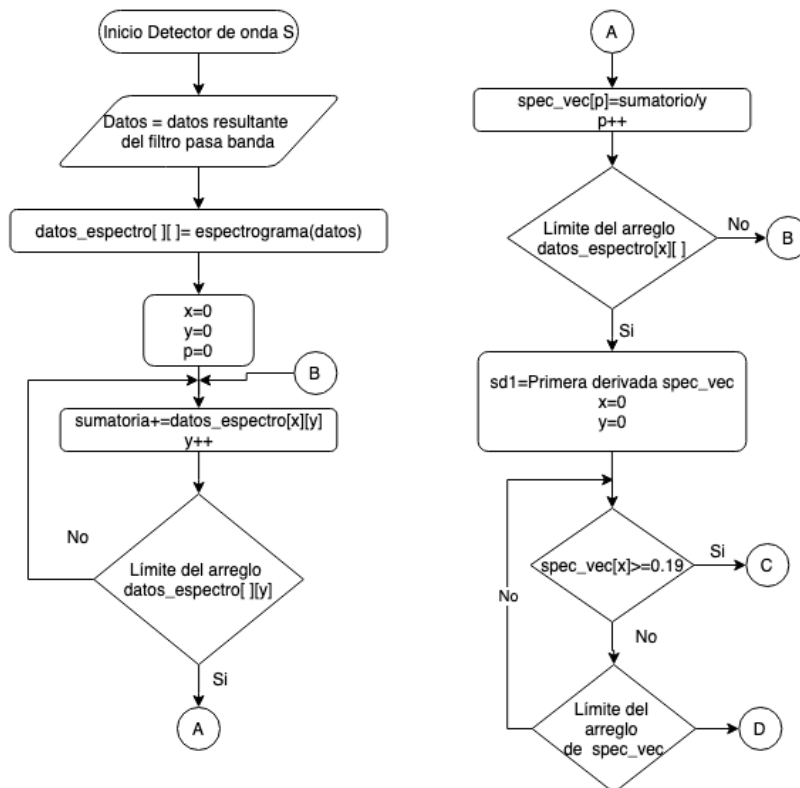
Algoritmo para detectar y localizar la onda secundaria

La onda secundaria de un movimiento telúrico forma parte de las ondas de volumen que se propagan de forma transversal a la onda de propagación. Las ondas de cuerpo S tienden a viajar con mayor lentitud que la onda de cuerpo P, además que en masas de composición líquida tiende a disiparse.

El conjunto de instrucciones para la detección de la onda S está organizado en dos etapas importantes, la primera consiste en la detección del disparo y la segunda en el reconocimiento del inicio de la onda secundaria, de esta manera se presenta en la Figura 57 un diagrama de flujo describiendo el comportamiento del algoritmo para detectar el disparo, este se basa en determinar cuando los datos del arreglo resultante de la densidad espectral promedio han superado un umbral preestablecido. Tomando en cuenta que los datos vienen en un arreglo unidireccional de memoria y los datos fueron sometidos a la etapa de filtración que se explicó anteriormente.

Figura 57

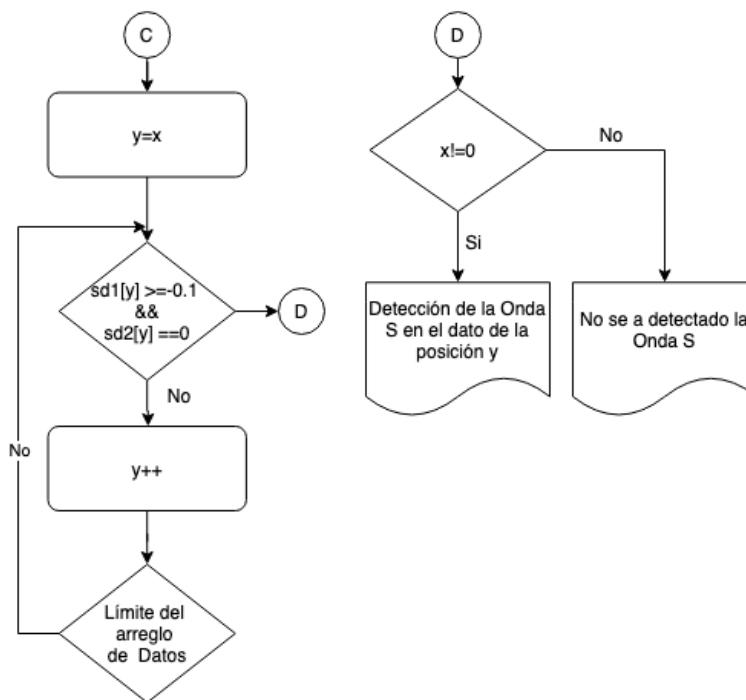
Algoritmo para la detección del disparo.



La lógica para el reconocimiento de la onda secundaria posterior a la detección del disparo se representa en el diagrama de flujo como podemos observar en la Figura anterior con numeración 58, y el diagrama está relacionado a la segunda etapa del algoritmo que se basa en analizar los datos desde el tiempo del disparo, y realizando el análisis con los vectores la primera y segunda derivada del arreglo resultante de la densidad espectral promedio para determinar el punto de inflexión de la señal que se explicará posteriormente.

Figura 58

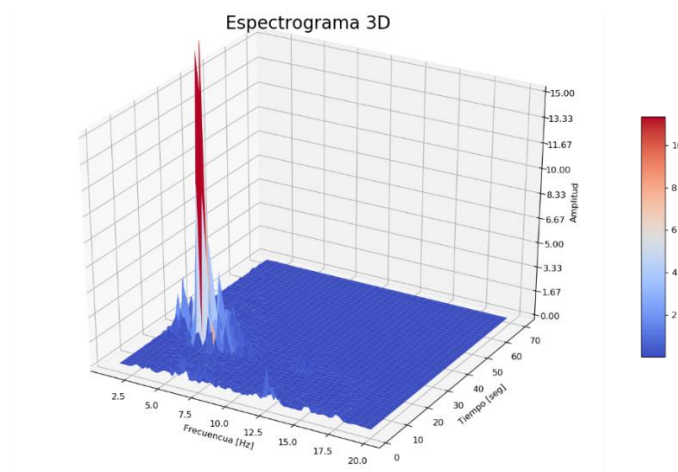
Algoritmo para la detección de la onda S.



Según el documento científico de (Chi, 2015) titulado “Nuevas estrategias de identificación automática de tiempos de llegada de ondas P y S”, presenta nuevos algoritmos para aplicaciones en sismogramas, entre las estrategias menciona el método de los espectrogramas (ME) que se basa en la determinación del espectro de Fourier que proviene de procesador información de eventos telúricos. Inicialmente para hacer uso de este metodo debe ejecutar un análisis de la información del evento sísmico representados en espectrogramas , como se puede observar en la Figura 59, se presenta la representacion en tres dimensiones de una señal sísmica donde existen pequeños impulsos de energia en los primeros cinco segundos de color azul denotando la onda P (primaria), mientras que a los veinte segundos existe un cumulo de energia fuerte represetando con el color rojo que toma el nombre de onda S (secundaria).

Figura 59

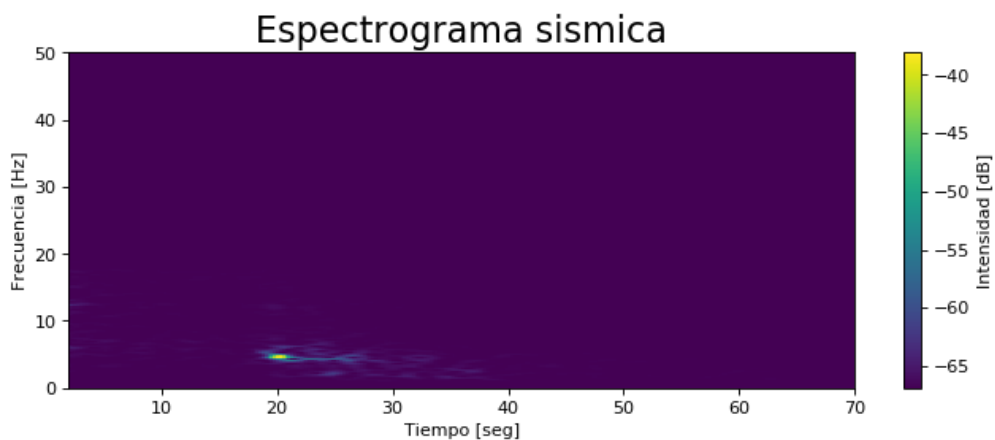
Espectrograma de una señal sísmica.



Los espectrogramas tienen diferentes formas de representación, una de ellas es la representación en los 3 ejes por lo tanto los 3 planos ortogonales como se están en la Figura 59 aunque según el requerimiento se puede realizar una representación de 2 dimensiones como esta en la Figura 60. Donde la intensidad de los colores representan la intensidad de la señal que está ocurriendo en esa etapa de tiempo, conjuntamente con las frecuencias que están involucradas.

Figura 60

Espectrograma de una señal sísmica filtrada.

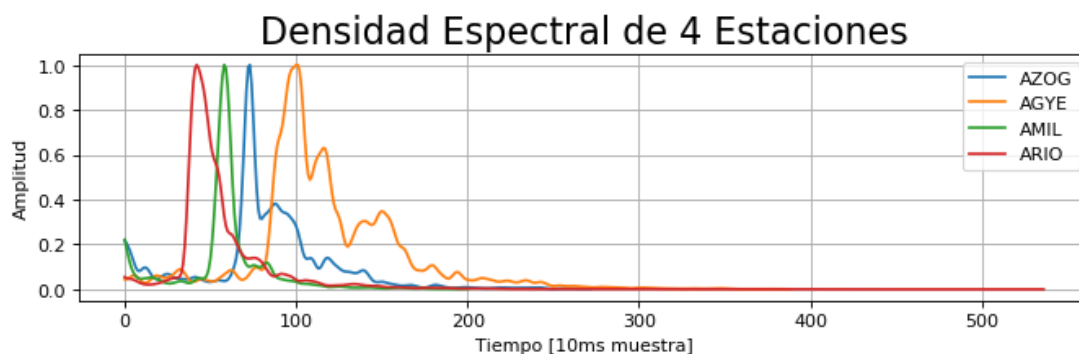


Para hacer uso del método de los espectrogramas, primero se obtiene el espectrograma de los información telúrica en ventanas de 0.5 Hz, dando como resultado vectores de datos con las intensidades relacionadas con el tiempo. Considerando que tenemos datos desde frecuencias menores con un valor de 1 Hz hasta los 50 Hz , este último valor que corresponde al resultado del filtro pasa bajo incorporado.

Con los datos de los vectores que salen del espectrograma se operó la suma de cada unos de las componentes respetando su posición dentro del arreglo, dividiéndolo para la cantidad de vectores existentes, a este resultado (Chi, 2015) lo denomina densidad espectral promedio que permite localizar en qué rango de tiempo se desarrolló mayor cantidad de energía en las diferentes frecuencias que involucraron el sismo que se este analizando. Como se puede observar en la Figura 61 se realizó el análisis de un mismo sismo, pero los datos pertenecen a cuatro estaciones sísmicas del IGEPN diferentes, donde se aplicó la densidad espectral promedio y podemos claramente diferenciar el comportamiento donde inicia el incremento de energía de la onda S.

Figura 61

Densidad espectral promedio de los datos del IGEPN.

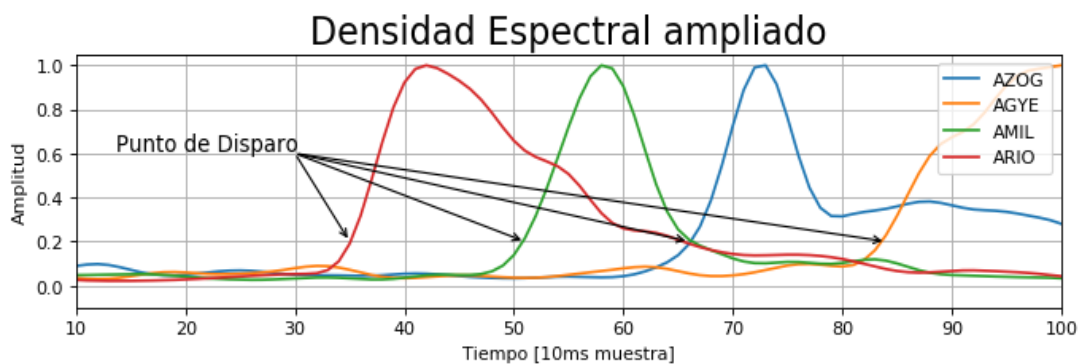


Según (Jiménez, 2005) para determinar el valor de umbral de disparo debemos analizar los datos de instrumentación de la misma ubicación geográfica, y tomando en cuenta que la

velocidad que las ondas telúricas, tiene diferentes respuestas que son el resultado de la composición del suelo provocando cambio en el área decreciente entre las dos elevaciones de energía, y el valle entre las dos concentraciones de energía. Se establecieron 0.19 unidades ya que proporcionó la detección más realista para determinar el crecimiento de la onda S perteneciente a un evento telúrico y para efectos ilustrativos la figura 62.

Figura 62

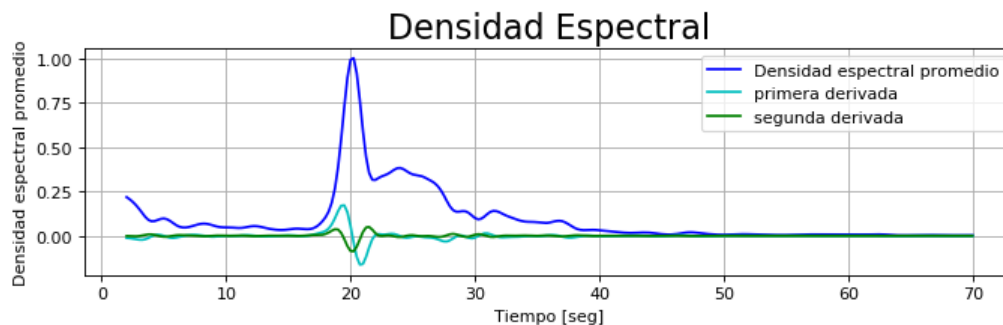
Umbral de disparo de la onda S.



Para obtener la densidad espectral promedio debemos calcular la primera y segunda derivada para comprender de mejor manera la Figura 63 están las diferentes funciones para su compresión, combinando el análisis de los vectores en las diferentes funciones podemos obtener los máximos y mínimos locales como también los puntos de inflexión, este último se intenta localizar para determinar la onda secundaria del evento sísmico.

Figura 63

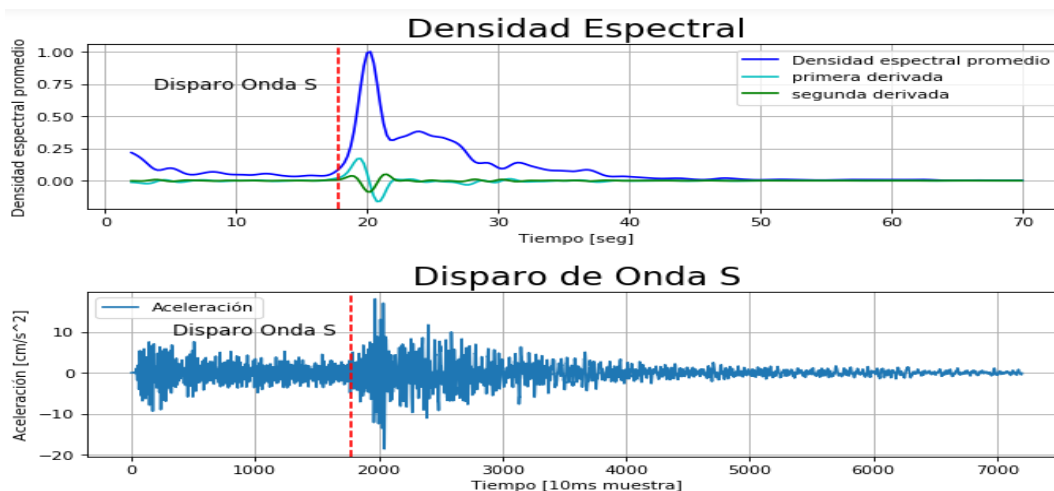
Densidad espectral promedio.



En la Figura 64 a se puede observar cuando la función de densidad espectral promedio supera el umbral establecido, genera el disparo de la onda secundaria. El disparo permite validar el inicio de una potencial onda secundaria como se puede contrastar con la Figura 64 b en la cual se muestra el evento sísmico original, entonces comparando entre ambas gráficas se comprueba que donde se generó el disparo es el punto donde la energía empieza a crecer, por lo tanto se entiende que es el inicio de la onda secundaria.

Figura 64

Densidad espectral promedio y localización de ondas S.

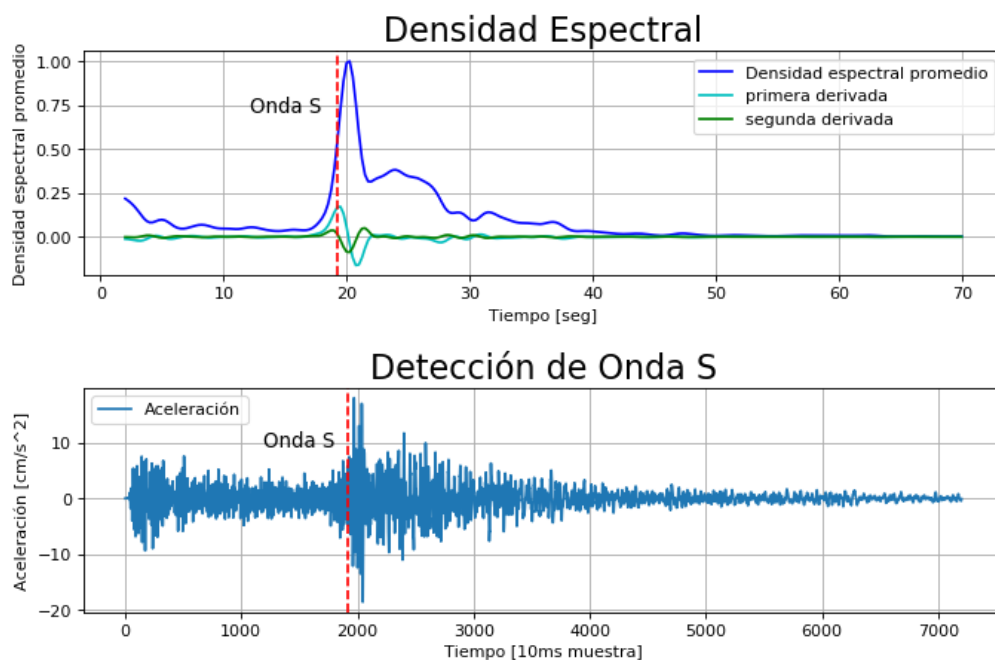


Como ya se ha obtenido la primera y segunda derivada de la densidad espectral promedio, se inicia con la validación dato por dato en las funciones de las dos derivadas para determinar el punto de inflexión correspondiente, cuando se calcula que el punto en cuestión su segunda derivada es igual a 0 y su primera derivada tiene un signo positivo, justo en ese instante del tiempo se a logrado llegar a determinar, la existencia de la onda S de un evento telúrico y con estos datos se puede determinar posteriormente el tiempos de arribo o más conocido como TOA.

En la Figura 65 a se presenta la detección de la onda secundaria del evento sísmico de la Figura 64 pero en este caso usando el método del espectrograma y el criterio de punto inflexión, y la Figura 65 b con la que se marca con una línea entrecortada de color rojo señala la ubicación en el tiempo la identificación aproximada de la onda de cuerpo S.

Figura 65

Localización de onda S.

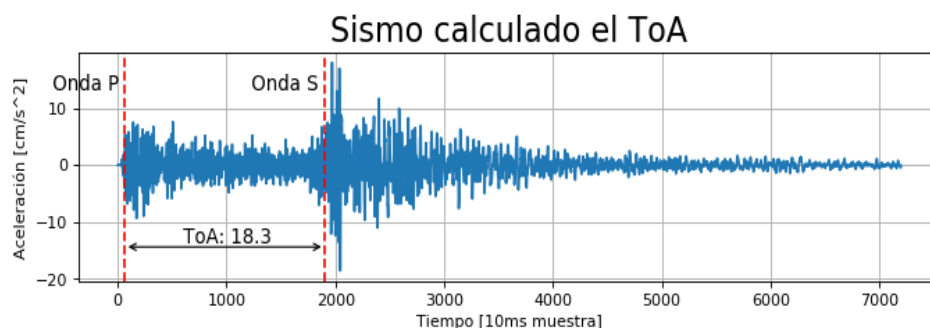


Algoritmo para calcular el tiempo de arribo

Para determinar el tiempo de arribo de un evento sísmico se debe previamente detectar y localizar la onda primaria y onda secundaria, basada en los algoritmos previamente expuestos, en el presente proyecto se hará uso de esta teoría aunque en la ciencia de la sismología existen variedad de criterios que se pueden usar para dicha detección. Usando la resta de los tiempos en que se detectaron las ondas; y se debe usar las unidades en segundos para tener errores en los cálculos y disminuir inconvenientes.

Figura 66

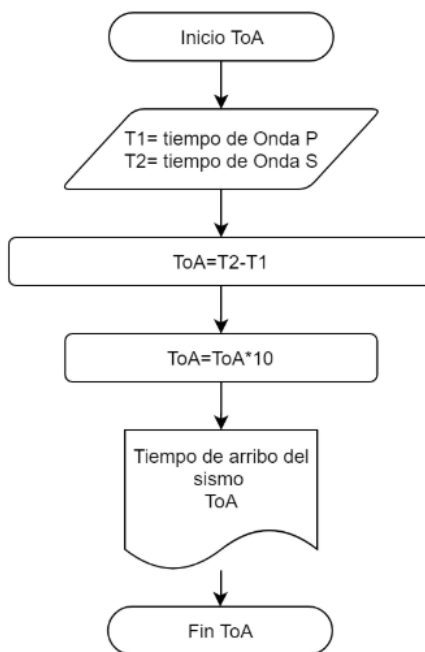
Identificación de las onda de cuerpo P y S.



En la Figura 66 se puede observar que se están usando la misma información con la que se usó para detectar las ondas superficiales, y arrojó como resultado del proceso 18.3 segundos. El algoritmo representado en un diagrama de flujo se puede observar en la Figura 67 para determinar el tiempo de arribo, simplemente es la resta de los tiempos en que el sistema los detectó a las onda de cuerpo y ese tiempo está entregado en milisegundos, y hay que realizar la respectiva transformación para presentar los resultados en segundos.

Figura 67

Diagrama de flujo para calcular ToA.



Algoritmo para cálculos de trilateración

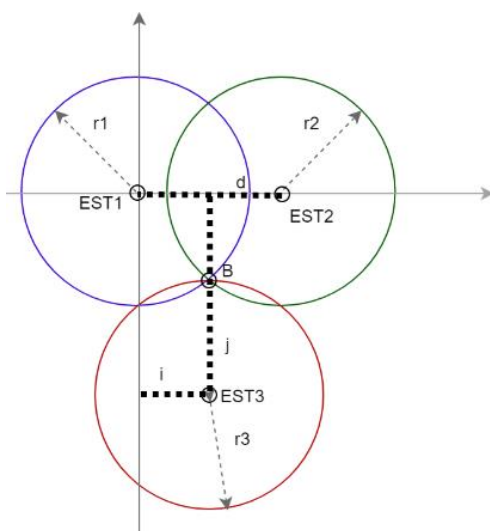
Para el desarrollo del algoritmo del cálculo de la trilateración es necesario disponer del Tiempo de Llegada (ToA) de al menos tres estaciones de monitoreo sísmico, con este valor mínimo se puede realizar el cálculo con márgenes de error. Según el documento científico titulado “*Algoritmos de radiolocalización basados en ToA, TDoA y AoA*” de (Garcia, 2015), determina que para sistemas no coplanares, es decir cuando las estaciones no están a la misma altura. Se requiere de cuatro estaciones para poder obtener como resultados la menor cantidad de error y para sistemas coplanares de al menos tres estaciones.

Para los cálculos de la trilateración de eventos sísmico se debe conocer las ubicación geográfica GPS de las estaciones de monitoreo sísmico hasta con 4 decimales de exactitud, para la primera etapa lo que se realiza es el análisis con 3 estaciones y luego se pivotea para agregar la cuarta estación. En la Figura 68 se ha nombrado a las estaciones como EST1, EST2 y EST3

respectivamente, de ahí se escoge a la primera estación como punto referencial para el plano que usaremos para los cálculos iniciales.

Figura 68

Cálculos de trilateración.



Las unidades usadas para los cálculos son grados, las mismas unidades que ya tienen las estaciones cuando proporcionan su ubicación GPS. Partiendo de esta premisa se convirtieron las distancias de los radios que se calcularon previamente con el tiempo de arribo en grados para que todos los cálculos estén en armonía.

$$r_1^2 = x^2 + y^2 + z^2 \quad (9)$$

$$r_2^2 = (x - d)^2 + y^2 + z^2 \quad (10)$$

$$r_3^2 = (x - i)^2 + (y - i)^2 + z^2 \quad (11)$$

Para la trilateración se debió partir con tres esferas como se señala en las ecuaciones anteriores, después se resta la segunda a la primera, para resolver x

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d} \quad (12)$$

Se sustituye la ecuación (12) en la (9) correspondiente a la primera esfera, dando como resultado la fórmula del círculo. En la siguiente ecuación se refleja la solución para la intersección que se relaciona a las dos primeras esferas.

$$y^2 + z^2 = r_1^2 - \frac{(r_1^2 - r_2^2 + d^2)^2}{2d} \quad (13)$$

Se procede a igualar la ecuación (13) con la ecuación (11) correspondiente a la de la tercera esfera dando como resultado:

$$y = \frac{r_1^2 - r_3^2 - x^2 + (x-i)^2 + j^2}{2j} = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x \quad (14)$$

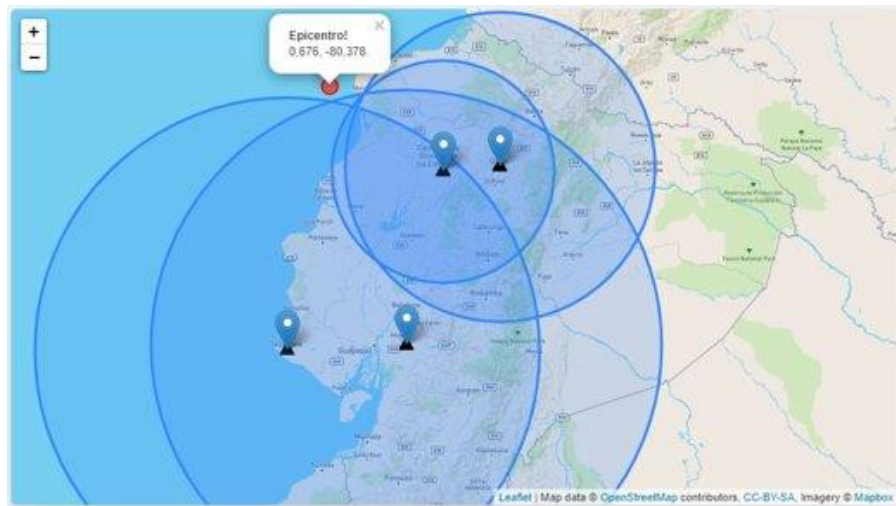
Finalmente tenemos la solución para la ubicación x, y y z . Para la solución de z puede dar como resultado cero, uno o dos resultados porque proviene de una raíz cuadrada.

$$z = \sqrt{r_1^2 - x^2 - y^2} \quad (15)$$

Se repite este procedimiento pivoteando con la cuarta estación para disponer de un segundo punto referencial para generar una media entre los dos puntos geométricos para poder determinar el punto el epicentro con el menor error posible. Como se puede observar en la Figura 69 es un mapa donde se ha localizado el epicentro del evento telúrico en base a datos del IGEPN

Figura 69

Ubicación geográfica del epicentro.



Desarrollo del algoritmo de cálculo de magnitud de un sismo

En base al trabajo titulado *“Determinación de la magnitud local ML a partir de acelerogramas de movimientos fuertes del suelo”* de (Espinosa, 1989), donde propone una expresión empírica para determinar la magnitud local de un evento sísmico en base al análisis de las aceleraciones horizontales del movimiento del suelo, expresado de la siguiente forma.

$$M_L = \log_{10} A_{cc} - \log A_1(\Delta) \quad (16)$$

Según la ecuación se expresa que A_{cc} es la amplitud máxima que el acelerógrafo registro del evento telúrico, almacenados en unidades de cm/seg^2 . Para la variable A_1 es la distancia basada en un factor de amplitud deducida de forma empírica. Por lo tanto según la distancia que nos proporcionó como resultado el tiempo de arribo debemos revisar en la tabla para poder disponer el factor para el cálculo.

Figura 70

Valores de las amplitudes.

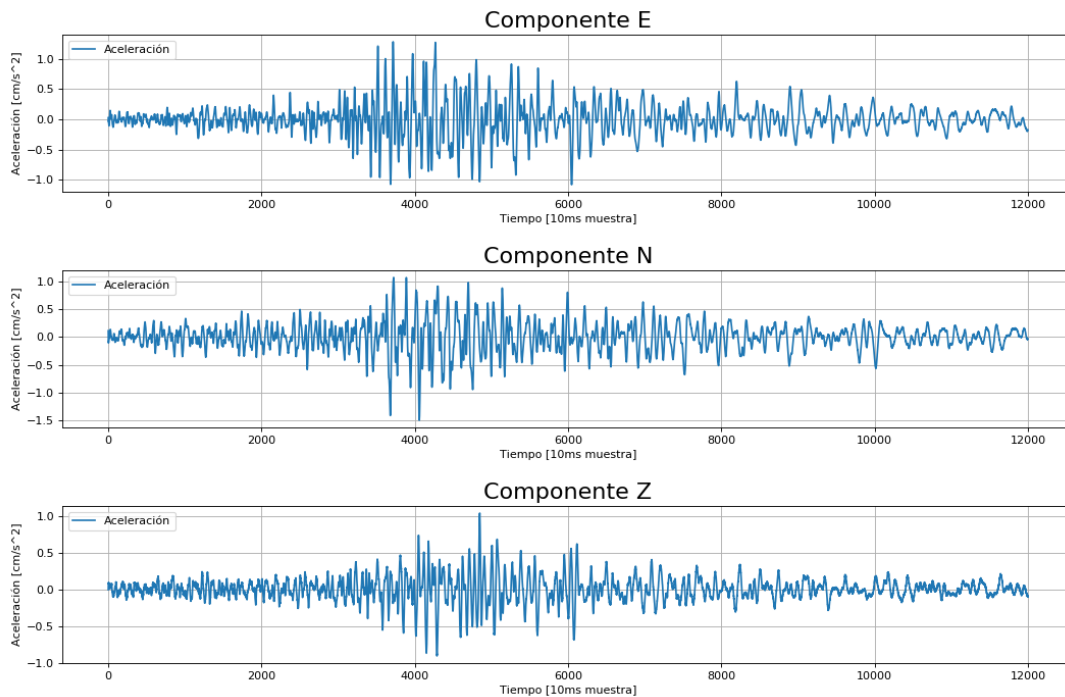
Δ	$A(\Delta)$	Δ	$A(\Delta)$	Δ	$A(\Delta)$	Δ	$A(\Delta)$	Δ	$A(\Delta)$
1	3.31	61	4.75	121	5.36	181	5.76	241	6.06
2	3.32	62	4.76	122	5.37	182	5.76	241	6.06
3	3.33	63	4.78	123	5.37	183	5.77	243	6.07
4	3.37	64	4.79	124	5.38	184	5.77	244	6.07
5	3.40	65	4.80	125	5.39	185	5.78	245	6.08
6	3.45	66	4.82	126	5.40	186	5.78	246	6.08
7	3.47	67	4.83	127	5.40	187	5.79	247	6.09
8	3.52	68	4.84	128	5.41	188	5.80	248	6.09
9	3.55	69	4.85	129	5.42	189	5.80	249	6.09
10	3.57	70	4.87	130	5.43	190	5.81	250	6.10
11	3.62	71	4.88	131	5.43	191	5.81	251	6.10
12	3.66	72	4.89	132	5.44	192	5.82	252	6.11
13	3.70	73	4.90	133	5.45	193	5.82	253	6.11
14	3.74	74	4.91	134	5.46	194	5.83	254	6.12
15	3.78	75	4.92	135	5.46	195	5.83	254	6.12
16	3.82	76	4.94	136	5.47	196	5.84	256	6.12
17	3.85	77	4.95	137	5.48	197	5.84	257	6.13
18	3.89	78	4.96	138	5.48	198	5.85	258	6.13
19	3.92	79	4.97	139	5.49	199	5.86	2596.06	6.14

Nota. La figura anterior muestra los datos correspondientes a los valores numéricos de las amplitudes $(\Delta) = -\log_{10} A_1$ en cm/seg^2 . Tomado de *Determinación de la magnitud local ML a partir de acelerogramas de movimientos fuertes del suelo*, por A. Espinoza, 1989.

Como se puede observar en la Figura 71 se presenta la descomposición de los tres ejes del registro de un acelerómetro, si el acelerómetro está orientado al norte magnético, las componentes toman el nombre de N, E y Z respectivamente para los ejes x,y y z. En base a la propuesta de (Espinoza, 1989) se realiza la extracción del valor máximo de los componentes N y E, para generar una media aritmética entre ambos valores para posteriormente usar la ecuación (16), dando como resultado de las operaciones matemáticas el resultado de la magnitud local del evento.

Figura 71

Aceleraciones registradas.



Implementación de visualización de mapas

En la página web se realizó la incorporación de mapas debido a que brindan una mejor organización para observar las ubicaciones y de los datos que se han calculado previamente en el backend del servidor con información procedente de las estaciones, para ello se usó la librería Leaflet, que está desarrollada en Javascript y es de código abierto. Se escogió esta librería porque brinda mejores prestaciones que las librerías de Google Map, entre ellas se puede señalar que el uso de Google Map requiere asociar una un método de pago para obtener el token de identificación de desarrollador, y aunque es gratuita para usos de poco tráfico puede generar cobros si el tráfico incrementa. Mientras que Leaflet es una solución a la altura de Google Map brindando las mismas prestaciones pero sin riesgo de pagos posteriores.

Leaflet tiene muchas opciones para la adición de capas que mejoran las prestaciones de visualización de mapas en entornos web, esto se logra con el uso asociado de OpenStreetMap para agregar esta función, y escoger la mejor capa para el propósito del desarrollador, en este caso se estableció el uso de dos capas, la primera capa del mapa permite tener un tono más opaco para que resalte los símbolos de las estaciones y los últimos eventos como se presenta la imagen de la figura 72, al mismo tiempo es la primera página que el operador observará al conectarse al sistema. También se incorporó una capa topográfica para mejorar la observación de las cadenas montañosas.

Figura 72

Vista de la página principal.

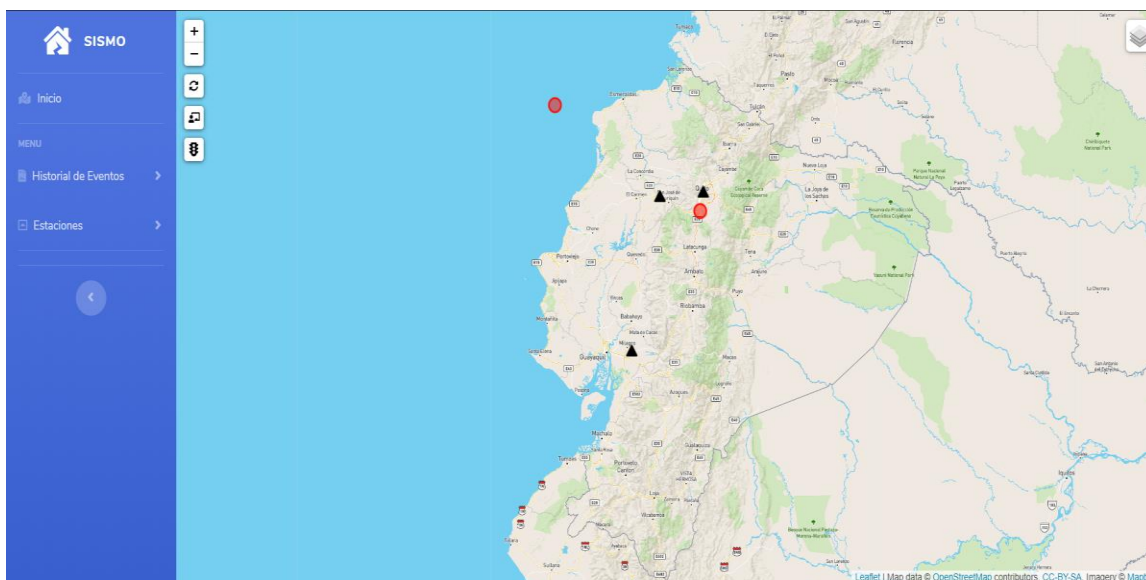


Figura 73

Capa de relieves.



Para el diseño se inicia con la incorporación del mapa en el cuerpo de página HTML, con el Id denominado “*mapid*”, con el atributo de un ancho que cubra toda la pantalla del cliente, con una altura de 500 pixeles, conjuntamente tiene la configuración HTML para que se centre el mapa, todo ello contenido en una clase. Las líneas de código que expresan lo anteriormente mencionado se contempla en el segmento de código 15.

Segmento de código 15

Parámetros del mapa.

```

10  % block page_content %
11
12  <div class="container">
13      <div class="row justify-content-center">
14          <div class="col-xl-10 col-lg-12 col-md-9">
15              <div class="card o-hidden border-0 shadow-lg my-1">
16                  <div id="mapid" style="width: 100%; height: 500px;"></div>
17              </div>
18          </div>
19      </div>
20  </div>

```

Como se puede observar en el Segmento de código 16, donde las líneas de código tienen el propósito de incorporar las capas de visualización del mapa, con la especificación del

nivel de zoom en el HTML con un valor de 18 unidades. El mapa se puede cargar en una localización predeterminada, en este caso se estableció la posición geográfica con las coordenadas -1.5 y 79.0 para latitud y longitud respectivamente, esta ubicación y combinado con el nivel de zoom, nos genera que se visualice el mapa del Ecuador desplegado en toda la pantalla.

Segmento de código 16

Sentencias Javascript.

```

96 <script>
97
98   var mymap = L.map 'mapid' .setView [-1.5,-79], 7 ;
99
100  L.tileLayer 'https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token=pk.eyJ1I
101    maxZoom: 18,
102    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>
103      <a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
104      'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
105      id: 'mapbox/streets-v11'
106    } .addTo mymap ;

```

Con la ayuda de la librería ShapeMarker se realizó la implementación de Figuras geométricas que se despliegan en el mapa, para cada figura se establecen parámetros de ubicación geográfica, tamaño y color. En el segmento de código número 16 es donde el algoritmo crea un objeto denominado “*circulo3*” que se presenta el epicentro calculado, tiene los atributos de un margen de color negro y color de cuerpo rojo con un parametro de opacidad del 50%. La representación de este círculo se puede observar en la Figura 69.

Segmento de código 17

Incorporación de figuras geométricas.

```
121     var circulo3 = L.shapeMarker [lat,lon], {
122         fillColor: "#ff0900",
123         fillOpacity: 0.5,
124         weight: 2,
125         color: "#ff0900",
126         shape: "circle",
127         radius: 8
128     } .addTo mymap .bindPopup "<b>Epicentro!</b><br />
```

En el presente capítulo se mostró la formación del servidor web con sus respectivos servicios tales como el servidor Nginx el cual aloja a todos los demás servicios, el broker MQTT que permite la comunicación inalámbrica con la estación sismográfica, Node- Red para la comunicación en tiempo real con la estación y la conexión con el MySQL que permite el almacenamiento de la información registrada. Además se puede observar el desarrollo del servidor donde el operador por medio de un computador con acceso a internet puede acceder a la página web del sistema para visualizar los registro de eventos telúricos. En este punto todo el desarrollo del sistemas del prototipo de sismógrafo se ha completado por lo que en el siguiente capítulo se realizará las pruebas del mismo mediante la instalación, configuración y calibración de las estaciones en dos sedes de la Universidad de las Fuerzas Armadas "ESPE".

Capítulo V

Pruebas y resultados

Estación de monitoreo sísmico

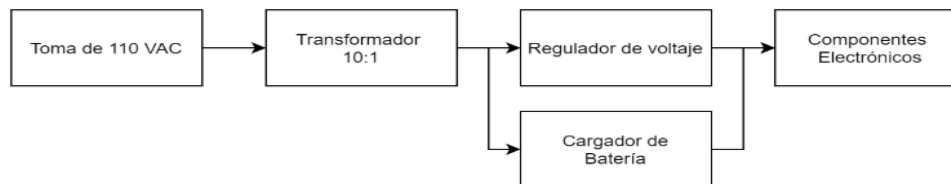
Estabilidad y consumo de energía de la estación de monitoreo sísmico

La tarjeta electrónica posee dos fuentes de alimentación, pero la fuente principal es la que se encuentra conectada a la red eléctrica de corriente alterna por medio de un transformador 10:1, mientras que la segunda es una fuente de respaldo provista por una batería de 12 voltios de corriente directa con una capacidad de 4 amperios. Al realizar las pruebas de estabilidad para el cambio de fuente principal a la de respaldo dio como resultado el 100% de su funcionalidad. Este cambio se da cuando se presente una interrupción de la alimentación AC ya que se genera un cambio de los niveles de voltaje, por lo que la batería se conecta de forma automática por medio de diodos rectificadores, y con la capacidad de los condensadores electrolíticos generan que la ESP32 no llegue a generar ningún cambio en su comportamiento de conectividad hacia la red ni cambios en la lectura de sus sensores.

En la Figura 74 se presenta un diagrama de bloques de la fuente alimentación de la estación de monitoreo sísmicos, como se puede observar la estación se conecta a la red eléctrica para alimentar a sus elementos electrónicos y realizar la carga de la batería para los casos de fallas eléctricas.

Figura 74

Diagrama de bloques de la alimentación de la estación.



El equipo tiene un consumo de 0.05 amperios como se puede observar en la Figura 75, que a su vez se alimenta con 117 voltios de corriente alterna dando como resultado una carga de 5.85 watt/hora, si consideramos que el kilowatt/hora es a USD 0.08 para Ecuador daría una facturación de energía eléctrica de USD0.33 mensuales para tener una estación de monitoreo sísmico 24 horas encendido todos los días.

Figura 75

Medición del consumo en amperios a la entrada del transformador.



Tiempo de funcionalidad con respaldo de batería

La batería cumple la función de respaldo de energía de manera temporal a la estación de monitoreo sísmico con un voltaje de 12 voltios y 4 amperios en los casos de una falla eléctrica. En teoría con una capacidad de 4000 mA de la batería y un consumo de 108.9 mA de la tarjeta electrónica daría un tiempo de respaldo de 36.73 horas. Como se puede ver en la Figura

76 se realizó la respectiva medición de corriente que es consumida por la tarjeta electrónica con todos sus elementos y módulos tales como GPS, lector de memoria SD y MPU6050 en total funcionamiento.

Figura 76

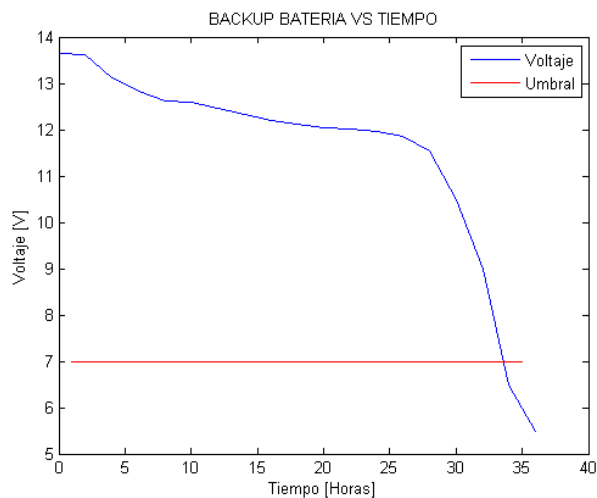
Medición del consumo en amperios.



En las pruebas de funcionamiento de la estación sin conexión a la red eléctrica, realizadas cada dos horas se obtuvo el comportamiento del voltaje de la batería dando como resultado lo que se puede observar en la Figura 77. Se determinó que la tarjeta electrónica deja de funcionar cuando el voltaje de la batería desciende por debajo de los 6.75 voltios, a este valor se denominó umbral de apagado y se lo ilustra con una línea roja en la Figura 77.

Figura 77

Comportamiento de la batería.



Según las pruebas realizadas se tiene como resultado un respaldo de energía de aproximadamente 34 horas en contraste a las 36.73 horas que se calculaban de manera teórica dando como resultado un error de 7.43%..

Sensibilidad del acelerómetro mpu6050

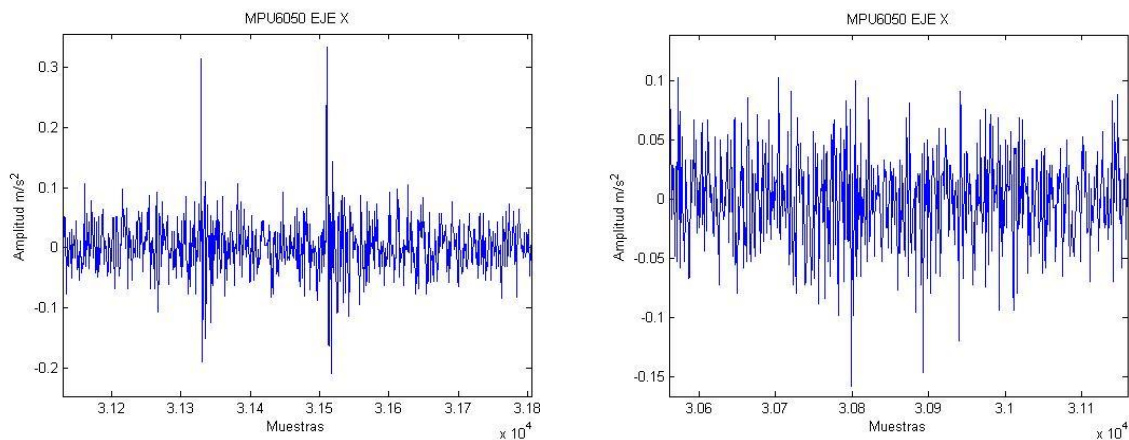
Pruebas de ruido. Las pruebas de ruido se realizaron con el fin de contemplar los niveles de ruido que es capaz de detectar el sensor MPU6050, por lo que se evaluó dicho sensor en ambientes con una presencia mínima de perturbaciones. Los horarios en los cuales se vio conveniente realizar las pruebas fueron a partir de la medianoche debido a que los niveles de ruido ambiental en estos horarios son mínimos.

Para el procesamiento de los datos adquiridos de los tres ejes X, Y, Z del acelerómetro se hizo uso del software Matlab en su versión 2014a, del cual se usó la función “detrend” para calibrar el “offset” o componente DC presente por la presencia de la gravedad que afecta al eje Z por consiguiente también afecta a los ejes X, Y. Gracias a esto se puede obtener valores reales del acelerómetro. Se tomaron datos durante dos horas seguidas, de 00:00 a 2:00 am por lo que se separó el análisis de datos en dos etapas, una hora por cada etapa, con una tasa de muestreo de 100 Hz y el MPU6050 con sensibilidad de 2g.

Los datos pertenecientes al eje X mostraron que el valor mínimo absoluto alcanzado fue de 0.1556 m/s^2 en el horario de 01:00 a 2:00 am siendo este valor el más bajo, mientras que el valor máximo absoluto obtenido fue de 0.7565 m/s^2 entre las dos horas.

Figura 78

Datos de ruido pertenecientes.

**Tabla 11**

Pruebas de ruido del sensor MPU6050 eje X.

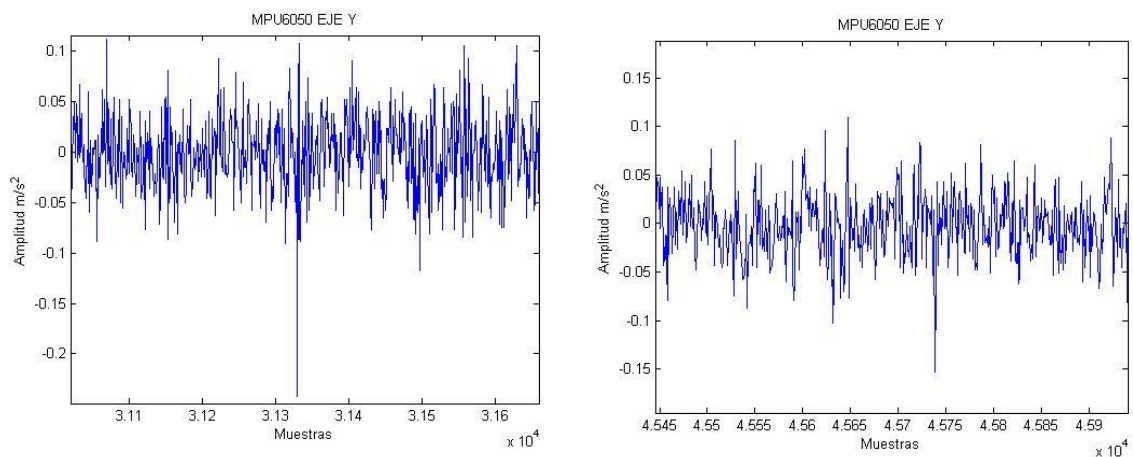
EJE X				
Hora AM	Mínimo (m/s^2)	Máximo (m/s^2)	Pico Pico (m/s^2)	Promedio (m/s^2)
00:00 – 01:00	-0.1614	0.3342	0.4956	0.4222
01:00 – 02:00	-0.1582	0.07638	0.23458	0.4216

Los datos obtenidos del eje Y reflejaron que el valor mínimo absoluto fue de

0.0192 m/s^2 en el horario de 01:00 a 2:00 am siendo este valor el más bajo, mientras que el valor máximo absoluto obtenido fue de 0.2609 m/s^2 en el mismo horario del valor mínimo.

Figura 79

Datos de ruido pertenecientes al eje Y.

**Tabla 12**

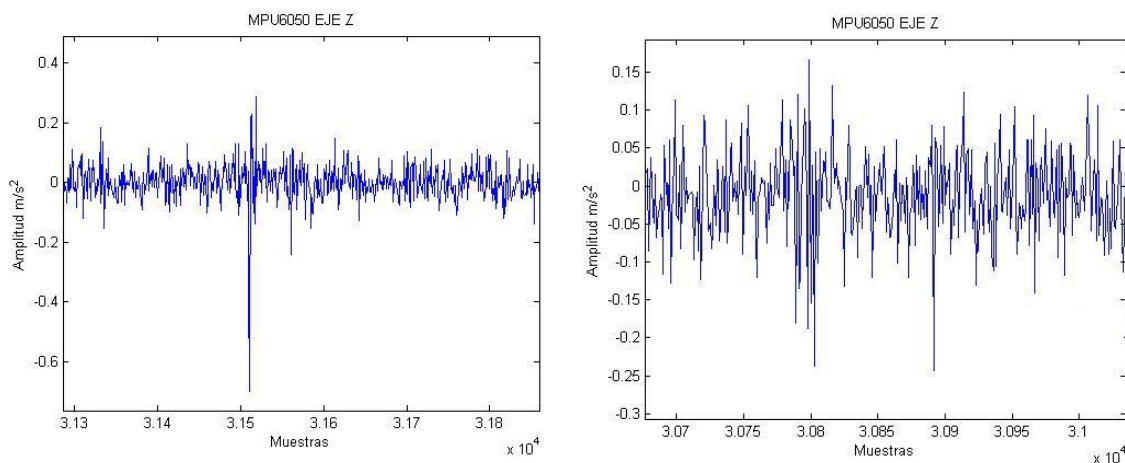
Pruebas de ruido del sensor MPU6050 eje Y.

EJE Y				
Hora AM	Mínimo (m/s^2)	Máximo (m/s^2)	Pico Pico (m/s^2)	Promedio (m/s^2)
00:00 – 01:00	-0.2421	0.1074	0.3495	0.1081
01:00 – 02:00	-0.1563	0.0462	0.2025	0.1080

Y por último los resultados obtenidos de los datos del eje Z reflejan que el valor mínimo absoluto entre las dos horas es de $0.000346 m/s^2$ en el horario de 00:00 – 01:00 am y el valor máximo de $0.874156 m/s^2$.

Figura 80

Datos de ruido pertenecientes al eje Z.

**Tabla 13**

Pruebas de ruido del sensor MPU6050 eje Z.

EJE Z				
Hora	Mínimo	Máximo	Pico Pico	Promedio
AM	(m/s^2)	(m/s^2)	(m/s^2)	(m/s^2)
00:00 – 01:00	-0.7026	0.2286	0.3495	0.1716
01:00 – 02:00	-0.2428	0.08036	0.2025	0.1741

Pruebas de la estación sismográfica

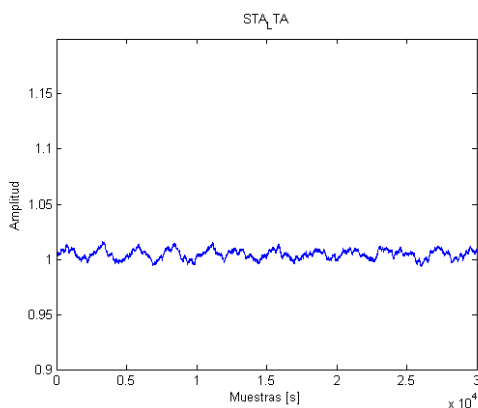
Algoritmo sta/lta

Para comprobar el correcto funcionamiento del algoritmo STA/LTA implementado en el prototipo de sismógrafo se realizaron varias pruebas, entre ellas primeramente se analizó los valores del promedio STA/LTA sin presencia de movimientos significativos durante un periodo de tiempo de 20 s con lo cual se obtuvo los resultados mostrados en la Figura 81, en la gráfica se

puede observar que los valores oscilan entre 0.98 y 1.03. Estos datos son importantes ya que a partir de estos valores se establece el umbral de activación (`trigger_On`) y desactivación (`trigger_OFF`) de dicho algoritmo.

Figura 81

Comportamiento de los valores STA/LTA.

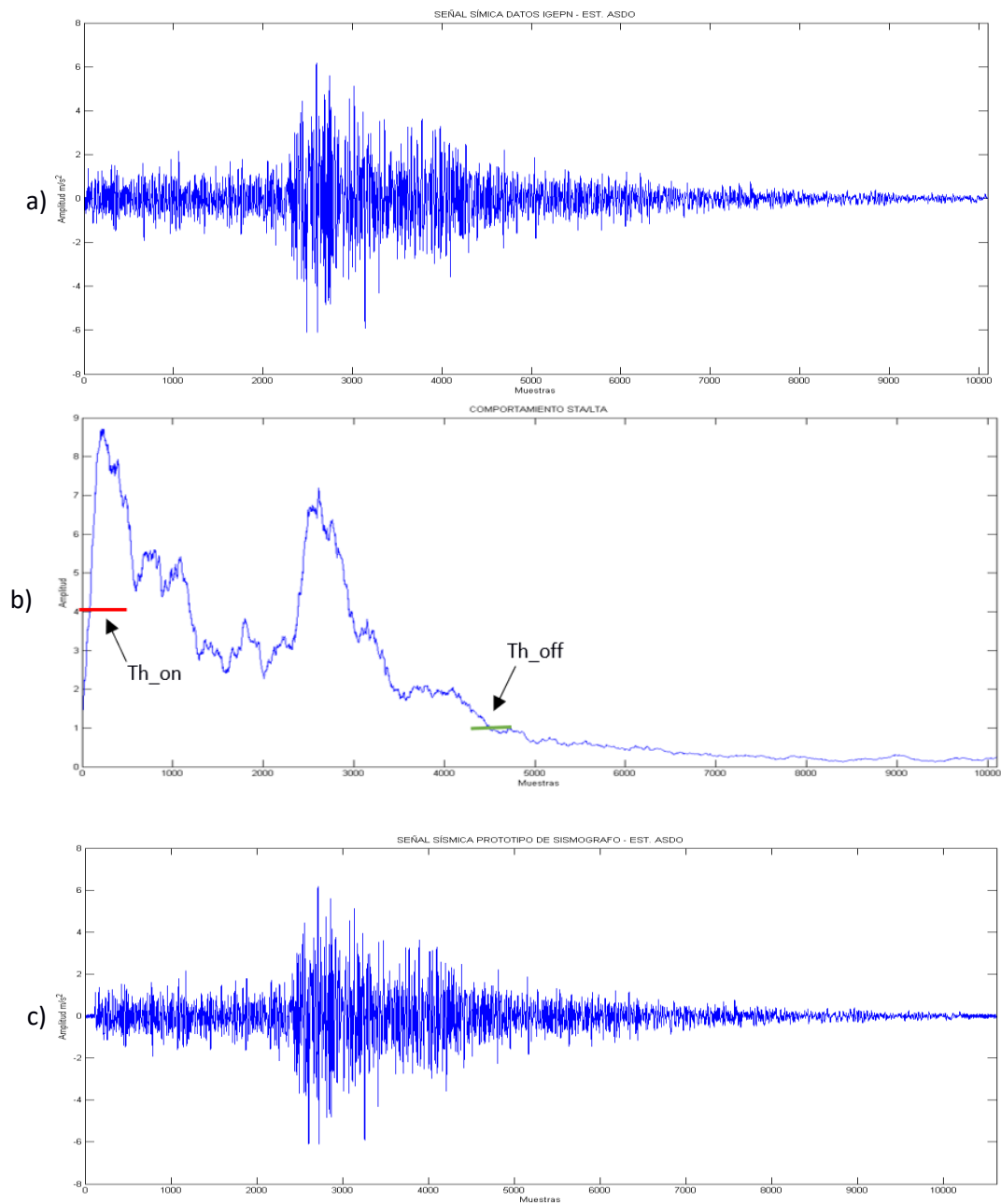


Así como se realizaron pruebas del correcto funcionamiento del servidor con datos obtenidos del IGEPN, para probar el prototipo de sismógrafo se realizó un procedimiento similar, en este caso el propósito es poner a prueba la detección de señales sísmicas por medio del algoritmo STA/LTA, para lo cual se programó un código de simulación, el cual permite leer datos de un archivo de texto plano, este archivo posee el mismo formato que el archivo que se guarda como respaldo después de haber registrado un evento. En dicho archivo se almacenó los datos proporcionados por el IGEPN de una de las estaciones de la red de sismógrafos del RENAC, que en este caso son los datos de la estación ASDO, que es una de las cuatro estaciones que se seleccionó para la pruebas del servidor. Como se explicó en secciones anteriores los parámetros del algoritmo STA/LTA no son valores estáticos sino que dependen mucho del lugar en donde se realice la instalación de la estación sismográfica, por lo que para dichas pruebas se establecieron los mismos valores presentados en la Tabla 8 del Capítulo 3. Para el análisis de los resultados

obtenidos del algoritmo se hizo uso de la herramienta Matlab para la graficación del comportamiento de los valores STA/LTA y las gráficas de la señal ingresada y la señal obtenida.

Figura 82

Análisis del algoritmo STA/LTA.



En la Figura 82 a se presenta la señal detectada por el IGEPN la cual va a pasar por el algoritmo STA/LTA, mientras que en la Figura 82 b se observa el comportamiento del algoritmo STA/LTA mientras se van leyendo los datos; cuando el promedio STA/LTA sobrepasa el umbral de activación (Th_{on}) se inicia con la alerta detección de un evento y se inicia el almacenamiento de los datos incluido el PEM hasta que dicho promedio este por debajo del umbral de desactivación (Th_{off}), para proceder a guardar el PET y así finalmente obtener la señal detectada en su totalidad. En la Figura 82 c se presenta la señal almacenada por el prototipo de sismógrafo, en esta gráfica se puede observar la señal completa ingresada más datos pertenecientes al ruido ambiental, con lo que se comprueba que el Algoritmo se está ejecutando correctamente.

Servidor web

Pruebas Servidor y página Web

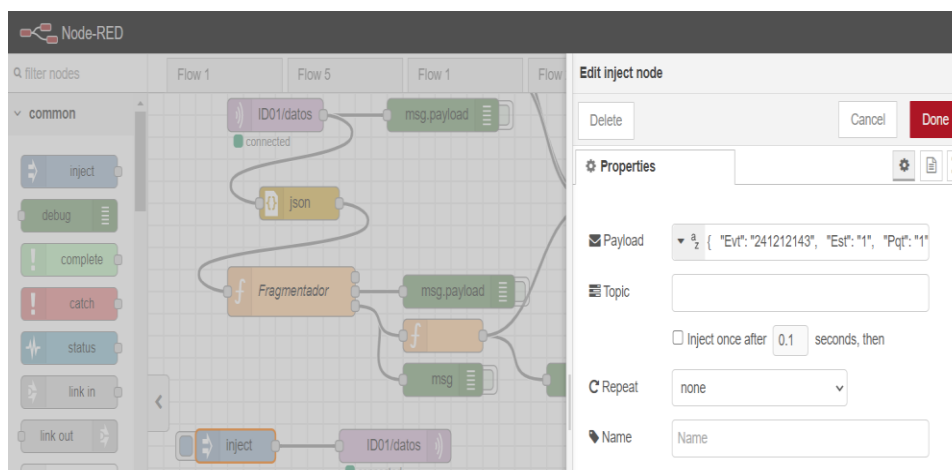
Para la verificación del correcto funcionamiento de los servicios alojados en el servidor y la visualización de los datos a través de la página web, se realizó una simulación de inyección de datos correspondientes a un sismo real, para su posterior procesamiento mediante cálculos y así obtener datos útiles tales como la magnitud, profundidad, TOA y radio. Los datos utilizados en esta simulación fueron adquiridos por medio del sitio web del IGEPN, los cuales corresponden a los eventos del miércoles 18 de mayo del 2016 registrados por las estaciones de la Red Nacional de Acelerógrafos (RENAC). El RENAC cuenta con 23 estaciones distribuidas por todo el país, de las cuales hemos seleccionado cuatro de estas para este propósito, entre ellas las estaciones ALIB (Santa Elena), AMIL (Milagro), ASDO (Santo Domingo de los Tsáchilas) y EPNL(Quito).

Para iniciar con la simulación primeramente, por medio de Node Red se inyectaron los datos en formato Json en el bloque *“inject”* hacia el topic *“ID01/datos”* para simular los

mensajes que son enviados por la estación sismográfica, este procedimiento se realizó con los datos de las tres componentes Z, N, E de cada estación. A este evento se lo ha nombrado con el ID "241212143" y a las estaciones como 1,2,3 y 4 para ALIB, AMIL, ASDO y EPNL respectivamente.

Figura 83

Inyección de datos hacia la base de datos.



Una vez finalizada la inyección de los datos se verifica su almacenamiento en la base de datos MySQL alojada en el servidor web como se muestra en la siguiente figura:

Figura 84

Datos almacenados en la base de datos MySQL.

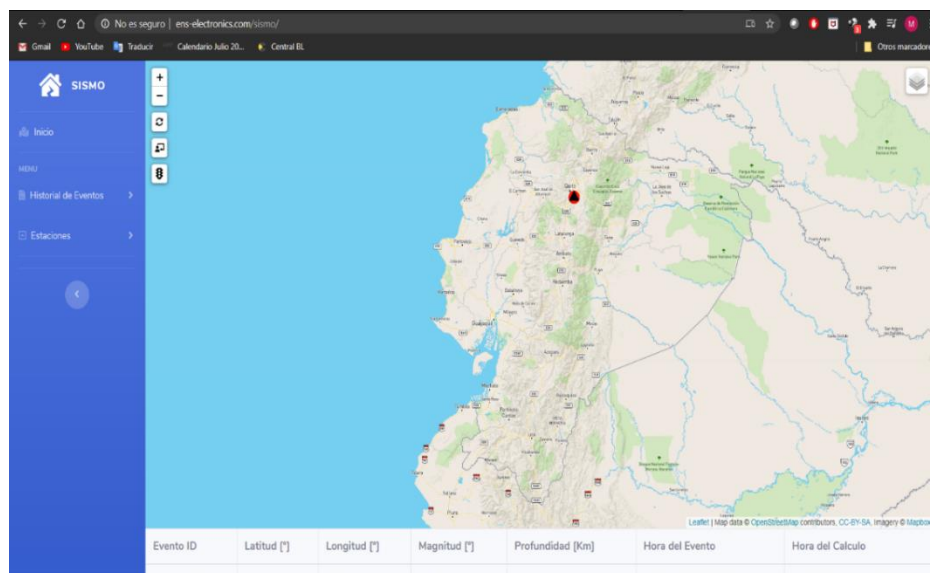
```
50 rows in set (0.01 sec)
mysql> mysql> SELECT * FROM sismo_data LIMIT 50;
```

id	evento	estacion	paquete	tiempo	valor	creado	componente
198276	241212143	1	3	151035.8450	0.0273	2020-04-18 12:34:31.000000	e
198277	241212143	1	3	151035.8550	-0.0292	2020-04-18 12:34:31.000000	e
198278	241212143	1	3	151035.8650	-0.0273	2020-04-18 12:34:31.000000	e
198279	241212143	1	3	151035.8750	-0.0343	2020-04-18 12:34:31.000000	e
198280	241212143	1	3	151035.8850	-0.0317	2020-04-18 12:34:31.000000	e
198281	241212143	1	3	151035.8950	-0.0506	2020-04-18 12:34:31.000000	e
198282	241212143	1	3	151035.9050	-0.0581	2020-04-18 12:34:31.000000	e
198283	241212143	1	3	151035.9150	-0.0751	2020-04-18 12:34:31.000000	e
198284	241212143	1	3	151035.9250	-0.0914	2020-04-18 12:34:31.000000	e
198285	241212143	1	3	151035.9350	-0.0971	2020-04-18 12:34:31.000000	e
198286	241212143	1	3	151035.9450	-0.1053	2020-04-18 12:34:31.000000	e
198287	241212143	1	3	151035.9550	-0.1072	2020-04-18 12:34:31.000000	e
198288	241212143	1	3	151035.9650	-0.1002	2020-04-18 12:34:31.000000	e
198289	241212143	1	3	151035.9750	-0.0921	2020-04-18 12:34:31.000000	e
198290	241212143	1	3	151035.9850	-0.0745	2020-04-18 12:34:31.000000	e
198291	241212143	1	3	151035.9950	-0.0594	2020-04-18 12:34:31.000000	e
198292	241212143	1	3	151036.0050	-0.0374	2020-04-18 12:34:31.000000	e
198293	241212143	1	3	151036.0150	-0.0079	2020-04-18 12:34:31.000000	e

Seguidamente se verificó en la página web el registro del evento sísmico ingresado, para lo cual primero se ingresó a la página web mediante el URL "<http://ens-electronics.com/sismo/>", el cual despliega la página de inicio como se puede ver en la Figura 85, como se puede observa la interfaz gráfica muestra el mapa del Ecuador y dos tablas donde se alojan los datos de los últimos tres sismos registrados por el sistema. Además sobre el mapa se muestran las dos figuras geométricas (círculo - triángulo) programadas como el indicativo de la o las tres últimas estaciones que alertaron un evento.

Figura 85

Página de inicio de la página web.



Una vez en el sitio web se ingresó a la opción Historial de eventos que presenta la página, donde se listan mediante una tabla todos los eventos registrados como se puede observar en la Figura 86, y se verifica el registro del sismo.

Figura 86

Página del historial de eventos sísmicos.

ID Evento	Latitud [°]	Longitud [°]	Magnitud [°]	Fecha/Hora
241212143	0,885	-80,638	6,7	21/03/2020 09:33:48
030310.003	-0,433076	-78,539994	0,0	21/03/2020 09:33:48

Como se puede ver en la imagen anterior el evento fue registrado de manera exitosa, y se muestra en la tabla de historial de eventos sísmicos, al ingresar a dicho evento por medio del ícono celeste de base de datos, se presenta otra página como se observa en la Figura 87 en la cual se detalla mediante una tabla datos más específicos de dicho eventos tales como: las estaciones que registraron el mismo evento, la ubicación geográfica de cada estación (latitud, longitud), el ID del evento, el número total de paquetes que envió cada estación, el tiempo, la magnitud local, el tiempo de arribo o TOA y el Radio. Como se muestra en la Figura 88.

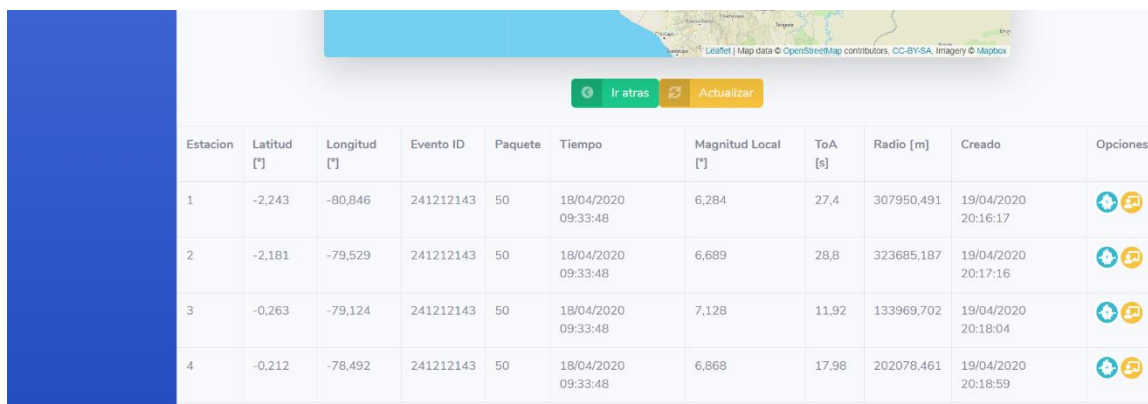
Figura 87









Página de información detallada.

Estacion	Latitud [°]	Longitud [°]	Evento ID	Paquete	Tiempo	Magnitud Local	ToA [s]	Radio [m]	Creado	Opciones
1	-2,243	-80,846	241212143	50	18/04/2020 09:33:48	6,284	27,4	307950,491	19/04/2020 20:16:17	
2	-2,181	-79,529	241212143	50	18/04/2020 09:33:48	6,689	28,8	323685,187	19/04/2020 20:17:16	

Figura 88

Datos generada para el evento.

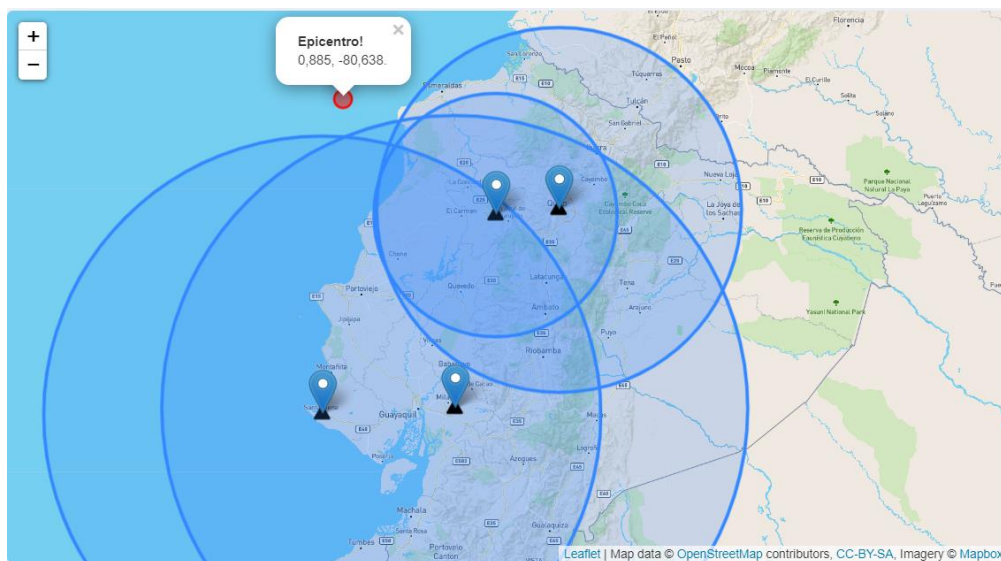


Estacion	Latitud [°]	Longitud [°]	Evento ID	Paquete	Tiempo	Magnitud Local [°]	ToA [s]	Radio [m]	Creado	Opciones
1	-2.243	-80,846	241212143	50	18/04/2020 09:33:48	6,284	27,4	307950,491	19/04/2020 20:16:17	 
2	-2,181	-79,529	241212143	50	18/04/2020 09:33:48	6,689	28,8	323685,187	19/04/2020 20:17:16	 
3	-0,263	-79,124	241212143	50	18/04/2020 09:33:48	7,128	11,92	133969,702	19/04/2020 20:18:04	 
4	-0,212	-78,492	241212143	50	18/04/2020 09:33:48	6,868	17,98	202078,461	19/04/2020 20:18:59	 

En la Figura 87 también se muestra un mapa sobre el cual están ubicadas las estaciones que detectaron el evento sísmico, alrededor de cada estación se grafica una circunferencia representando el radio que alcanzó la señal sísmica detectada por cada una. Y mediante un círculo de color rojo se representa la ubicación del epicentro calculado. Todo lo mencionado con anterioridad se presenta a mayor escala en la Figura 89.

Figura 89

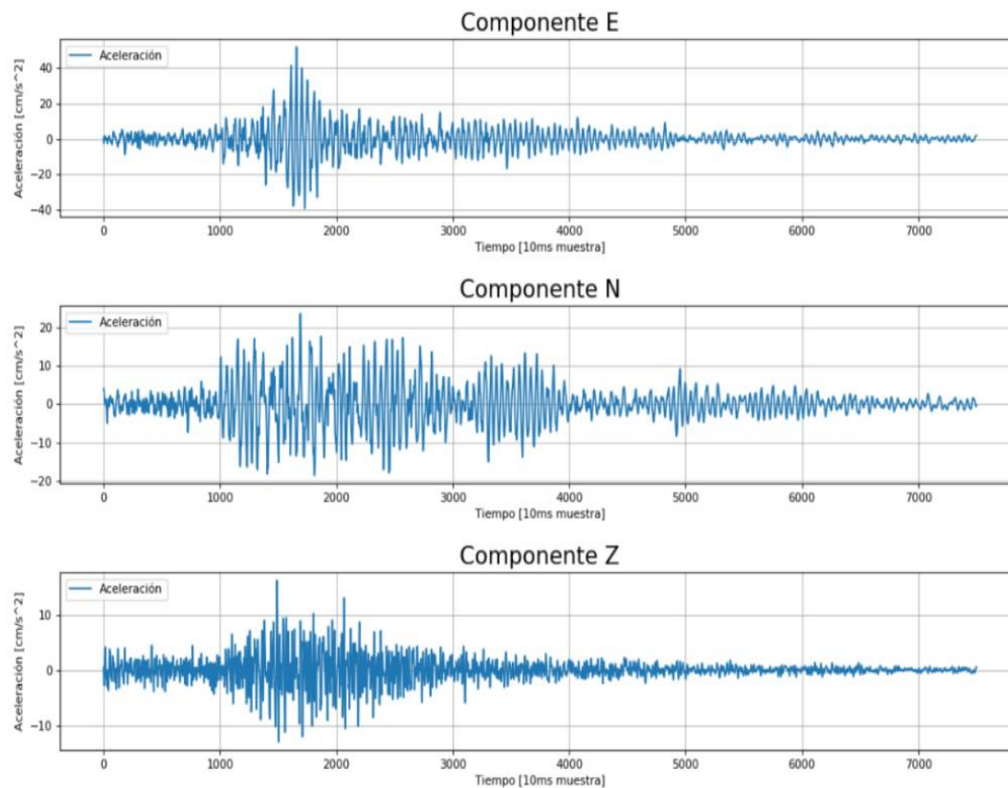
Ubicación geográfica de estaciones.



En la tabla presentada en la Figura 88 en el apartado opciones se presentan dos íconos el primero (ícono celeste) permite obtener las gráficas de la señal sísmica detectada por la estación correspondiente, mientras que la segunda opción permite realizar los cálculos de la magnitud, TOA y el radio. Primero se ingresa a la opción de las gráficas, para lo cual se eligió una de las cuatro estaciones en este caso la estación 3 correspondiente a los datos proporcionados por la estación ASDO perteneciente al RENAC, dentro de esta opción se despliega lo mostrado en la Figura 90.

Figura 90

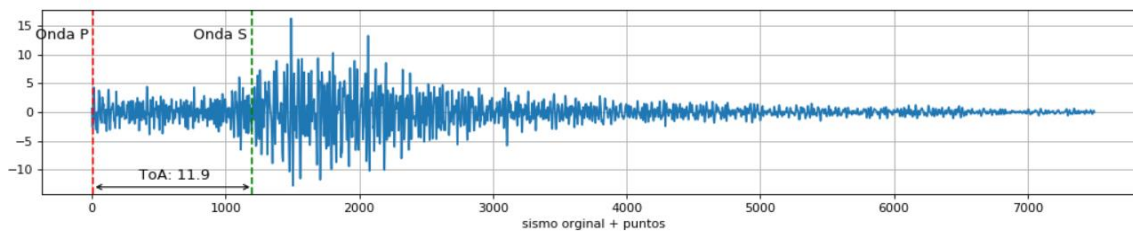
Componentes Z, N y E.



Al ingresar a la segunda opción (ícono amarillo) presentada en el apartado opciones de la tabla de la Figura 88 se realizaran todos los calculos correspondientes a la magnitud local, TOA, Radio, Epicentro y profundidad. Además permite observar las gráficas correspondientes al análisis realizado para el cálculo del TOA y el reconocimiento de la onda primaria y secundaria; este análisis solo se realiza con los datos de la componente “Z”, como se presenta en la Figura 91.

Figura 91

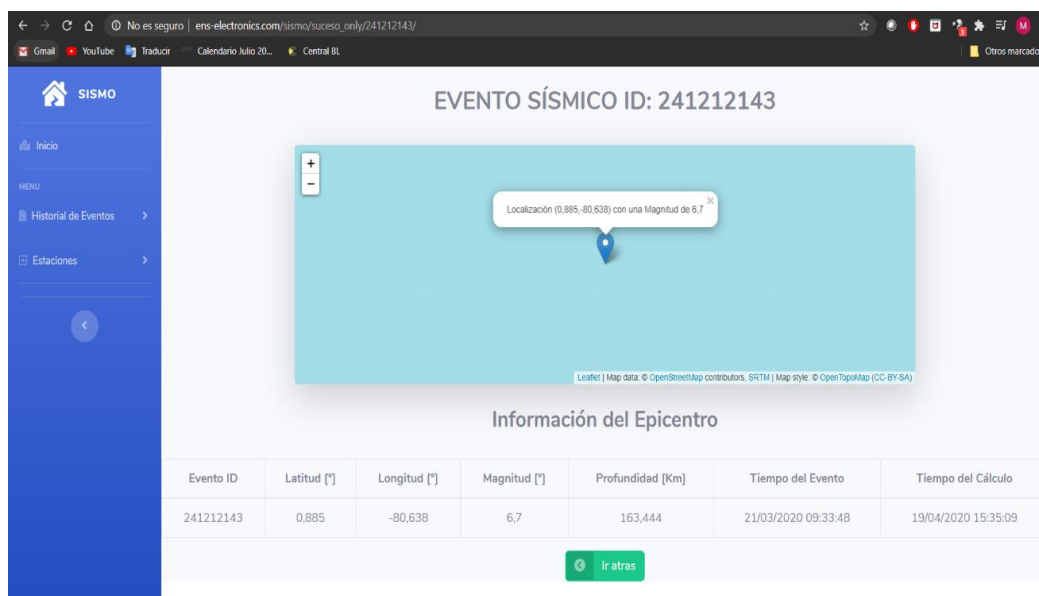
Cálculo del TOA y detección de la onda P y S.



Y por último al regresar a la página de Historial de Eventos y ingresar en la segunda opción es decir en el ícono de ubicación del apartado opciones de la tabla, se desplegará una nueva página la cual presentará información más general acerca del sismo registrado en este caso del evento de simulación “241212143”, tal como el ID o nombre del evento, sus coordenadas geográficas, magnitud, profundidad entre otras; además sobre el mapa se presenta la ubicación del epicentro del evento sísmico detectado por las estaciones, como se puede observar en la Figura 92.

Figura 92

Epicentro de un evento registrado.



Lo detallado anteriormente indica que la página web está trabajando en su totalidad, ya que presenta mediante interfaz gráfica todas las funciones desarrolladas en el servidor web; tales como la extracción de los datos almacenados en la base de datos para su presentación mediante tablas, la presentación de los diferentes mapas, gráficas de señales y los respectivos resultados aproximados de los cálculos de la magnitud, profundidad, TOA, radio, onda primaria y onda secundaria. Con lo cual ya se puede realizar un análisis comparativo de los resultados obtenidos con los datos del IGEPN, los cuales se presentan en la Tabla 14.

Los datos presentados en la Tabla 14 son los valores obtenidos de procesar los datos de cada señal detectada por cada estación simulada, tales como la magnitud local la cual ayuda a calcular la magnitud del epicentro, el TOA (Time of arrived) el cual ayuda a calcular el radio o la distancia que existe entre la estación y el epicentro.

Tabla 14

Comparación de datos obtenidos.

Estación	Latitud (°)	Longitud (°)	Magnitud Local	TOA (s)	Radio (m)
1 (ALIB)	-2.243	-80.846	6.284	27.4	307950.491
2 (AMIL)	-2.181	-79.529	6.689	28.8	323685.187
3 (ASDO)	-0.263	-79.529	7.128	11.92	133969,702
4 (EPNL)	-0.212	-78.492	6.868	17.98	202078,461

En la Tabla 15 se presenta un análisis entre los valores calculados por el sistema de monitoreo de sismos y los valores obtenidos por el IGEPN acerca del sismo del 18 de mayo del 2016. Como se puede apreciar el valor de la magnitud calculada por el sistema desarrollado es de 6.7 mientras que el valor obtenido por el IGEPN es de 6.6 dando un error experimental de 1.51 % el cual está dentro del error aceptado, además el cálculo del Radio se tiene un error

promedio de entre el 5% al 6 % ya que este se basa en el cálculo del TOA el cual al mismo tiempo se basa en la detección de la onda primaria y secundaria.

Tabla 15

Comparación de datos obtenidos por el sistema.

	Sistema de Monitoreo	IGEPN	ERROR (%)
Magnitud	6.7	6.6	1.515
Radio (m) Est. ALIB	307950.491	329570	6.559
Radio (m) Est. AMIL	323685.187	313400	3.282
Radio (m) Est. ASDO	133969,702	136630	1.947
Radio (m) Est. EPNL	202078,461	190300	6,189

Instalación de la estación

Se realizó la instalación de dos estaciones sismográficas, la primera ubicada en la Universidad de las Fuerzas Armadas “ESPE” Matriz localizada en la ciudad de Sangolquí en las coordenadas latitud de -0.3191 y longitud de -78.446, esta estación a la cual se ha nombrado Estación 1 se encuentra en el Centro de Investigación de Aplicaciones Militares (CICTE) y la segunda estación a la cual se le ha denominado Estación 2 en las instalaciones de la Universidad de las Fuerzas Armadas - sede Santo Domingo ESPE-ES con las coordenadas latitud de -0.2261 y longitud de -78.3434. Las dos estaciones se encuentran en lugares que se facilitaron por medio de la gestión del tutor académico. Adicionalmente se orientó el eje X del sensor MPU6050 hacia el norte magnético conforme a las características de los acelerómetros profesionales.

Figura 93

Estación 1 instalada.

**Figura 94**

Estación 2 instalada.



Pruebas de la señal Wi-Fi en el lugar de la instalación

Como se ha detallado en capítulos anteriores la estación sismográfica cuenta con comunicación inalámbrica Wi-Fi para su comunicación con el servidor (envío y recepción de

mensajes), por lo que es muy importante que la señal de la red WI-FI sea estable. Las dos instalaciones donde se ubican las estaciones sismográficas tanto en ESPE Matriz como en ESPE sede Santo Domingo prestan servicio de internet, en los dos casos la red con mejor calidad de servicio fue la red inalámbrica con el nombre “INVITADOS”, a estas dos redes se les realizó un monitoreo a través de la aplicación N – PERF instalada en un celular con sistema operativo Android, dicha aplicación permite determinar las características de una conexión a Internet, además que para realizar dicho monitoreo hace el uso del protocolo TCP utilizando los puertos 8080 el mismo usado en la comunicación MQTT del proyecto. Los parámetros de medición de la App son la velocidad de subida y bajada, latencia, Jitter entre otras.

Figura 95

Resultados presentados por la app N-PERF.

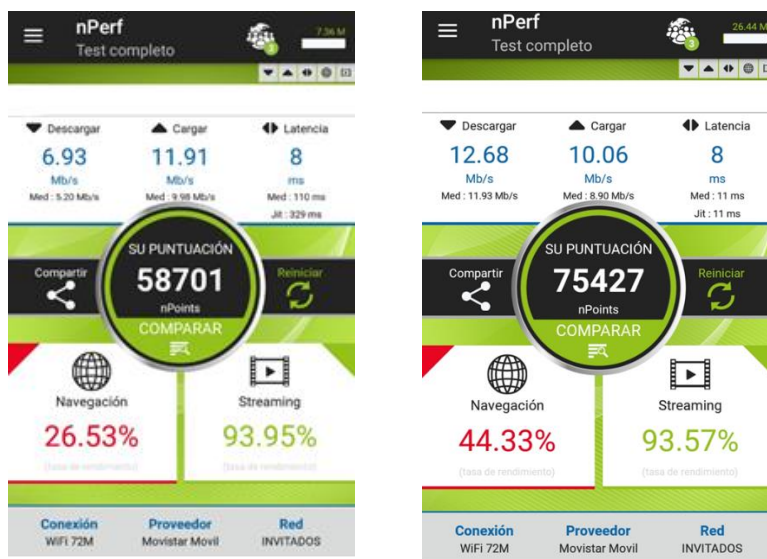


Tabla 16

Datos del eje Z obtenidos.

RED WIFI "INVITADOS"							
SEDE ESPE	Enlace de Subida Pico [Mb/s]	Enlace de Subida Medio [Mb/s]	Enlace de Bajada Pico [Mb/s]	Enlace de Bajada Media [Mb/s]	Latencia Pico [ms]	Latencia Media [ms]	Jitter [ms]
MATRIZ	11.91	09.98	6.93	5.20	8	110	329
Santo Domingo	12.68	11.93	10.06	8.90	8	11	11

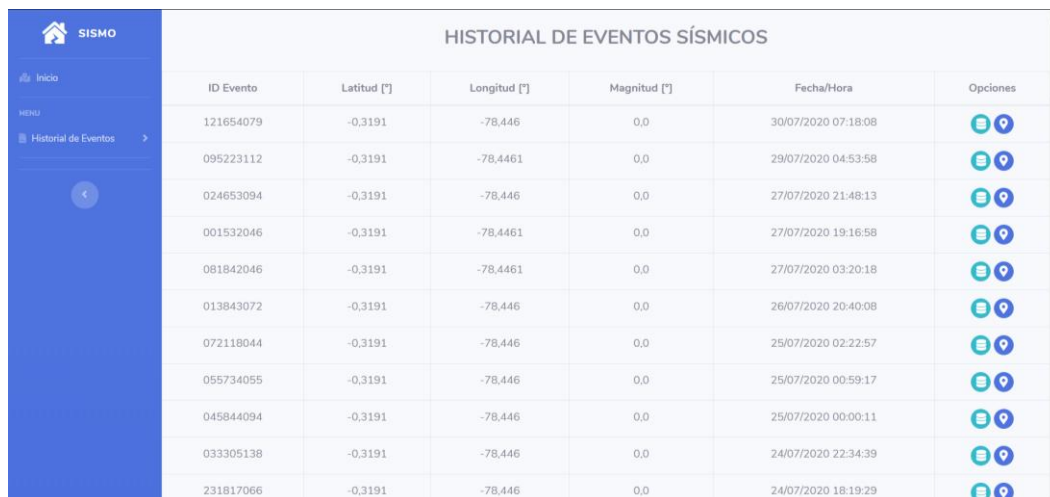
Los resultados obtenidos del monitoreo realizado a las redes nos indican que son redes óptimas para el envío y recepción de mensaje ya que tanto para el enlace de subida como el de bajada pasa los 6 Mb/s lo cual es suficiente para el intercambio de mensajes entre la estación y el servidor.


















Pruebas del sistema total de monitoreo sísmico

Una vez instalado las estaciones se realizó el monitoreo continuo durante cuatro semanas, tanto para la estación ubicada en ESPE matriz y ESPE sede Santo Domingo, dentro de este lapso de tiempo se corroboró el correcto funcionamiento del sistemas completo ya que no existieron fallas ni en hardware ni en software. De las dos estaciones la única estación que detectó movimientos significativos o que sobrepasaron el umbral de activación del algoritmo STA/LTA fue la estación ubicada en ESPE matriz pero ninguno de estos movimientos fueron movimientos telúricos, como se puede observar en la Figura 96 esta estación registró varios eventos detectados los cuales se ven reflejados en el Historial de eventos sísmicos.

Figura 96

Historial de los registros.

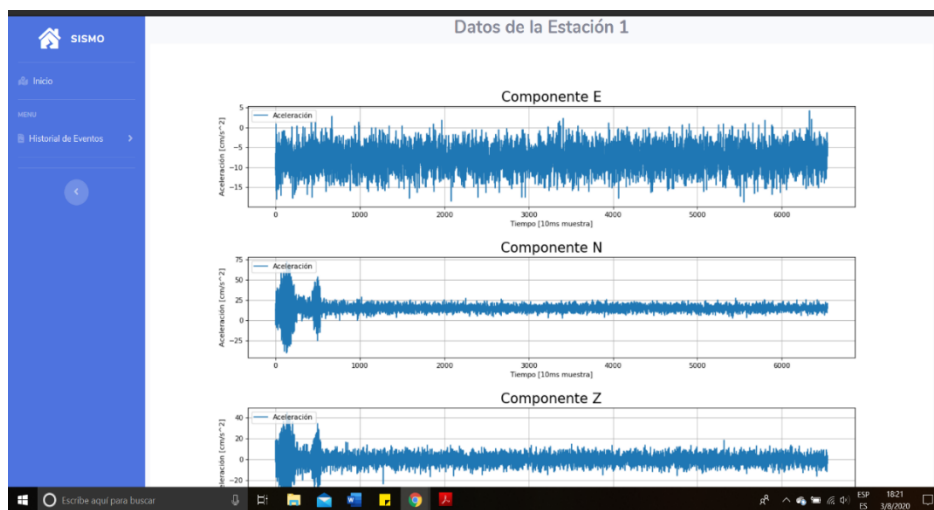


ID Evento	Latitud [°]	Longitud [°]	Magnitud [°]	Fecha/Hora	Opciones
121654079	-0,3191	-78,446	0,0	30/07/2020 07:18:08	 
095223112	-0,3191	-78,4461	0,0	29/07/2020 04:53:58	 
024653094	-0,3191	-78,446	0,0	27/07/2020 21:48:13	 
001532046	-0,3191	-78,4461	0,0	27/07/2020 19:16:58	 
081842046	-0,3191	-78,4461	0,0	27/07/2020 03:20:18	 
013843072	-0,3191	-78,446	0,0	26/07/2020 20:40:08	 
072118044	-0,3191	-78,446	0,0	25/07/2020 02:22:57	 
055734055	-0,3191	-78,446	0,0	25/07/2020 00:59:17	 
045844094	-0,3191	-78,446	0,0	25/07/2020 00:00:11	 
033305138	-0,3191	-78,446	0,0	24/07/2020 22:34:39	 
231817066	-0,3191	-78,446	0,0	24/07/2020 18:19:29	 

Los eventos registrados por la estación se dieron debido al paso de transporte pesado con material pétreo cerca de donde se encuentra instalada la estación, lo cual provocó la activación del umbral, esto se puede comprobar con las gráficas obtenidas de dichos eventos como se presenta en la Figura 97.

Figura 97

Historial de datos almacenados.



Las gráficas presentadas en la Figura 97 muestra que son movimientos provocados por el paso de transporte o algún agente externo ya que no tienden a seguir el comportamiento de una señal sísmica. Pero gracias a los eventos registrados se pudo comprobar que el sistema de monitoreo sísmico trabaja en su totalidad y cumple con los requerimientos establecidos del proyecto desarrollado.

Los resultados finales de las estaciones de monitoreo Telúrico se presentan en este capítulo. Es decir el funcionamiento del prototipo en conjunto (hardware y software). Primeramente se realizaron pruebas separadas tanto del hardware como del software. En el hardware (estación) se verificó el correcto funcionamiento del circuito electrónico y el código que se ejecuta en este, mientras que en el software (servidor web) se realizó varias pruebas mediante la inyección de datos de sismos reales obtenidos de la página oficial de la IGEPN. Para finalmente poner en funcionamiento al sistema completo.

Conclusiones

- Se desarrollaron dos prototipos electrónicos dotados con los dispositivos de instrumentación necesaria para la detección de eventos sísmicos, la cual se conforma de un acelerómetro para determinar movimientos telúricos significativos, un módulo GPS para ubicación y tiempo, un módulo WIFI para la conexión a internet y un módulo SDCard para el respaldo de datos. El prototipo usa la energía que proviene de la red eléctrica como fuente principal para su funcionamiento pero también cuenta con un respaldo de batería para evitar la suspensión del funcionamiento ante cualquier falla eléctrica. Gracias al diseño desarrollado de dicho prototipo se ha logrado que este funcione 24/7 con un 98% de conectividad estable con el servidor, permitiendo brindar un alto grado de confiabilidad en el equipo. Además, el gabinete IP65 donde se

encuentra alojado el equipo electrónico garantiza una protección ante las distintas condiciones ambientales.

- El uso del protocolo MQTT permitió el intercambio de mensajes entre las estaciones y la nube, dicho protocolo está basado en la pila TCP/IP el cual garantiza el envío y recepción de datos, además de caracterizarse por ser un protocolo de intercambio de mensajes con su propio algoritmo de detección de error basado en la longitud del topic y payload, además de disponer de niveles de calidad de conectividad permitiendo que se descarte la pérdida de paquetes de información entre la estación remota y el host.
- Mediante los algoritmos STA/LTA se logró determinar de manera confiable la discriminación y detección de la onda P que forma parte de las ondas de cuerpo, que en la gran mayoría de eventos analizados en un ambiente controlado, mientras que el algoritmo de densidad espectral promedio permitió discriminar y detectar la onda S, que inmediatamente podremos lograr el resultado del tiempo de Arribo también llamado ToA con un mínimo de error. Un paso importante es el cálculo de la magnitud de evento telúrico y para lograr este cálculo lo hacemos usando la ecuación basada en las aceleraciones horizontales que fue propuesta por Richter, dando un error menor al 1% en los cálculos y el método de trilateración se usó para los cálculos de epicentro y profundidad.
- La página web desarrollada cumple las características requeridas para el proyecto tales como la presentación de mapas con la ubicación de las estaciones sismográficas, las mismas que cuando detectan un evento sísmico se presentará de forma visual en el panel, complementado con datos e información relevantes de los eventos registrados y gráficas de las señales de dichos eventos. La plataforma tiene una organización intuitiva y amigable para el operador brindando rapidez en la verificación de los datos recibidos.

- Dentro del sistema de monitoreo sísmico se diseñó un algoritmo para la discriminación de falsos sismos, el mismo que comprueba que al menos dos estaciones de monitoreo sísmico de la red alerten la detección de un evento sísmico, permitiendo al algoritmo determinar la existencia de un sismo real, brindando un tiempo de anticipación en el rango de segundos, antes de que llegue la onda secundaria que es la causante de situaciones destructivas, esta etapa se comportaría como una alerta temprana que se podría conectar con las agencias de control pertinentes.

Recomendaciones

- Las estaciones de monitoreo sísmico tienen un gran potencial con toda la infraestructura de telemetría, almacenamiento y procesamiento de datos, se podría considerar el mejorar el instrumento de medición por un geófono analógico y convertidor de ADC de 24 bits para obtener una mayor sensibilidad de datos sísmicos.
- Aunque la premisa era desarrollar un prototipo de bajo costo, la ESP32 tiene limitaciones de memoria RAM para procesar los datos de los tres ejes del acelerómetro, se recomienda mejorar el hardware por una tarjeta con mayores prestaciones de memoria como por ejemplo una Raspberry Pi que permitiría aumentar la velocidad de procesamiento y envío de paquetes de datos a la nube.
- Para mejorar la plataforma web, se podría implementar el acoplamiento a otras redes sísmicas para poder corroborar la información calculada por el sistema de monitoreo sísmico, brindando al operador la disponibilidad de validar sismos incluso con los datos del IGEPN, ya que la página que se desarrolló tiene gran potencialidad de poder conectarse con otras redes o servicios.
- El módulo GPS que se usó para el proyecto es un modelo que solo recibe datos de la constelación de satélites GPS, pero en la actualidad existen varias constelaciones de

satélites que en conjuntos se las conoce como GNSS que brinda a los dispositivos receptores disminuir el error de localización, si se podría considerar colocar un módulo de geolocalización con mayores prestaciones, generaría una mejor precisión en la localización de la estación.

Referencias bibliográficas

- Cowwhite Software. (25 de 06 de 2020). *Cowwhite Software*.
<https://django.cowwhite.com/blog/creating-charts-and-output-them-as-images-to-the-browser-in-django-using-python-matplotlib-library/>
- ARDUINO - GENUINO. (s.f.). *Descubre Arduino*.
<https://descubrearduino.com/mqtt-que-es-como-se-puede-usar-y-como-funciona/>
- Arduino. (s.f.). *Arduino*. www.arduino.cc
- ArduinoJson. (s.f.). *ArduinoJson*. <https://arduinojson.org/>
- Bastidas, N., Berríos, L y Gonzáles, M. (2008). Sismógrafo. *Creando Revista Científica Juvenil*, 93-96. <http://erevistas.saber.ula.ve/index.php/creando/article/view/1652>
- Bormann, P. (2002). *IASPEI: New Manual of Seismological Observatory Practice (NMSOP)*.
 ForschungsZentrum.
- Chávez, R. (2016). *Análisis del framework django para la implementar aplicaciones web con base de datos mariaDB y metodología de desarrollo SCRUM*. [Tesis de Grado, Universidad Técnica del Norte de Ecuador]. <http://repositorio.utn.edu.ec/handle/123456789/5420>
- Chi, R. (Octubre de 2015). *Nuevas estrategias de identificación automática de tiempos de llegada de onda Py S*. XIV Congreso Geológico Chileno, Santiago de Chile, Chile.
- Chile Sismología. (s.f.). *Glosario Términos Habituales*.
<http://www.sismologia.cl/links/glosario.html>
- Condo, Y. (2008). *Detección, localización y análisis de sismos tsunamigénicos: Tsunamis Tremors*[Tesis de Grado, Universidad Nacional de San Agustín de Arequipa de Perú].
<http://repositorio.igp.gob.pe/handle/IGP/952>
- Cooper, J. D. (1868). *Earthquake indicator*.
- CSN. (s.f.). *Centro Sismológico Nacional Universidad de Chile*. www.sismologia.cl/

- Data Flair. (2020). *Data Flair*. <https://data-flair.training/blogs/django-models/>
- Demera, A. G. (2016). *Diseño e implementación de un sistema de alerta temprana de sismos mediante redes SWAP con nodos panStamp NRG para la ESPOCH*. [Tesis de Grado, Escuela Politécnica de Chimborazo de Ecuador].
<http://dspace.esPOCH.edu.ec/handle/123456789/6421>
- Electrónica, A. (2020). *AG Electrónica*. <http://www.agspecinfo.com/pdfs/O/OKY3001.PDF>
- Electrónico Caldas. (s.f.). *Electrónico Caldas*.
<https://www.electronicoscaldas.com/es/conectores-para-tarjetas-de-memoria-y-sim-card/930-modulo-para-memoria-micro-sd-y-micro-sdhc.html>
- Enciclopedia Británica. (s.f.). *Enciclopedia Britannica*.
<https://www.britannica.com/science/seismology>
- Espindola Castro, V. P. (2018). ¿Qué son los SISMOS, dónde ocurren y cómo se miden? *Ciencia*, 8-15.
https://www.revistaciencia.amc.edu.mx/images/revista/69_3/PDF/QueSonSismos.pdf
- Espíndola, J. M. (1994). *Terremotos y Ondas Sísmicas*.
- Espinosa, A. (1989). Determinación de la magnitud local ML a partir de acelerogramas de movimientos fuertes del suelo. *Física de la Tierra*, 106-129.
<https://revistas.ucm.es/index.php/FITE/article/download/FITE8989110105A/1259>
- García, A. (2015). Algoritmos de radiolocalización basados en ToA, TDoA y AoA. *Ingeniería y Región*,
- González, R. (27 de 05 de 2020). *Mundo Week*. <http://mundogeek.net/tutorial-python/>

- Herrera Zuleta, O. (2010). *Comunicación I2C Prototipo Brazo Robótico BM001*, Corporación Universitaria Autónoma del Cauca, <https://docplayer.es/18986556-Comunicacion-i2c-prototipo-brazo-robotico-bm001.html>
- Honores, J. (2015). *Análisis, diseño e implementación de una aplicación web utilizando el framework Django mediante la metodología Das para la administración online de citas médicas para el Hospital Básico la Cigüeña*. [Tesis de Pre Grado, Universidad Técnica de Machala de Ecuador]. <http://repositorio.utmachala.edu.ec/handle/48000/6060>
- GEPN. (s.f.). *Instituto Geofísico*. www.igepn.edu.ec/red-nacional-de-acelerografos
- IGEPN. (s.f.). *Instituto Geografico de la Escule Politecnica Nacional*. www.igepn.edu.ec
- Incense. (s.f.). *MPU-6000 and MPU-6050 Product Specification Revision 3.4*.
<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- INEC. (13 de Abril de 2017). *Memorias 13 abr 2017*.
www.ecuadorencifras.gob.ec/documentos/web-inec/Bibliotecas/Libros/Memorias%2013%20abr%202017.pdf
- Ingeniería en Microcontroladores. (s.f.). *Microcontroladores* www.i-micro.com/pdf/articulos/spi.pdf
- INPRES. (s.f.). *Sísmica, Instituto Nacional de Prevención Sísmica*. www.inpres.gov.ar.
- Islam, T. (2016). *The role of fovermments in Hazard mitigation*.
- Jaramillo Echeverría, R. (1994). *Diseño y construcción de un sistema digital de adquisición de datos y calibración de sensores sísmicos para el Instituto Geofísico de la EPN* [Tesis de Grado, Escuela Politécnica Nacional de Ecuador].
<http://bibdigital.epn.edu.ec/handle/15000/10751>
- Jiménez, C. (2005). *PROCESAMIENTO DIGITAL DE SEÑALES*.

- Lucas, J. (15 de 05 de 2020). *OpenWebinars*. <https://openwebinars.net/blog/que-es-c/>
- Martínez, P. Q. (2016). *Estudio de peligro sísmico de Ecuador y propuesta de espectros de diseño para la Ciudad de Cuenca*. Ingeniería sísmica, http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-092X2016000100001
- Matplotlib. (s.f.). *Matplotlib*. <https://matplotlib.org/>
- MDN web docs. (s.f.). *MDN web docs*. https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript
- MECHATRONICS, N. (2020). *NAYLAMP MECHATRONICS*. <https://naylampmechatronics.com/espressif-esp/382-modulo-esp32-esp-wroom-32.html>
- Moncayo Theurer, M. (2017). Análisis sobre la recurrencia de terremotos severos en Ecuador. *Revista Prisma Tecnológico*, 12-17.
- Naylamp Mechatronics. (s.f.). *MPU6050*. https://naylampmechatronics.com/763-thickbox_default/modulo-mpu6050-acelerometro-giroscopio-i2c.jpg
- Otero, D. Z. (s.f.). *ONDAS SÍSMICAS, su importancia para la geofísica y la humanidad*. Santander.
- Paredes Bernal, R. D. (2015). *Influencia de los elementos no estructurales en la respuesta dinámica de un edificio. Aproximación teórica-experimental* [Tesis de Grado, Escuela Politécnica Nacional de Ecuador]. <http://bibdigital.epn.edu.ec/handle/15000/12031>
- Parra Cárdenas, H. (2016). *Desarrollos Metodológicos y Aplicaciones hacia el cálculo de la Peligrosidad Sísmica en el Ecuador Continental y Estudio de Riesgos Sísmicos en la Ciudad de "Quito"* [Tesis Doctoral, Universidad Politécnica de Madrid de España]. <http://oa.upm.es/39353/>

- Peña, P. (2014). *Algoritmo de detección de ondas p invariante de escala: caso de réplicas del sismo del 11 de marzo de 2010* [Tesis de Grado Universidad de Chile de Chile].
<http://repositorio.uchile.cl/handle/2250/131361>
- Quinde Martínez, P. R. (2016). Estudio de peligro sísmico de Ecuador y propuesta de espectros de diseño para la Ciudad de Cuenca. *Ingeniería sísmica*, 94,1-26.
http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-092X2016000100001
- Ramírez-Herrera, M. T. (2011). Evidencias ambientales de cambios de nivel de la costa del Pacífico de México: terremotos y tsunamis. *Revista de Geografía Norte Grande*, 49, 99-124. https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-34022011000200007
- Rios, J. (2018). *Diseño de un acelerómetro para monitoreo sísmico* [Tesis de Grado, Tecnológico Nacional de México de México].
<http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/view/575>
- Saez, M. (2016). *Correlación cruzada de ruido sísmico para la obtención de perfiles profundos de velocidad de onda de corte en la cuenca de Santiago* [Tesis de Grado Universidad de Chile de Chile]. <http://repositorio.uchile.cl/handle/2250/138954>
- Saita, J. e. (2003). *UrEDAS: The Early Warning System for Mitigation of Disasters Caused by Earthquakes and Tsunamis*.
- Salazar, A. (2018). *Implementación de un prototipo para el monitoreo sísmico utilizando acelerómetros de bajo costo* [Tesis de Pre Grado, Escuela Politécnica Nacional de Ecuador]. <http://bibdigital.epn.edu.ec/handle/15000/19626>
- Solórzano Ávila, J. A. (2018). *Desarrollo de una aplicación web multiplataforma usando el framework Django, para publicitar eventos sociales, aplicado en el municipio del cantón*

- Morona* [Tesis de Pre Grado, Escuela Superior Politécnica de Chimborazo de Ecuador].
<http://dspace.esPOCH.edu.ec/handle/123456789/9103>
- Sotelo, A. (2016). *Sismógrafo electrónico de medición y detección de movimiento telúricos con salida impresa, para la carrera de ingeniería geológica de la Unan-Managua* [Tesis de Pre Grado, Universidad Nacional Autónoma de Nicaragua de Nicaragua].
<https://repositorio.unan.edu.ni/3207/>
- Sotomayor Verdugo, E. (2012). *Estudio de un sistema satelital de comunicación para alerta y reacción ante eventos sísmicos* [Tesis de Grado, Universidad de Chile de Chile]. <http://repositorio.uchile.cl/handle/2250/112126>
- Theurer, M. (2017). Terremotos mayores a 6.5 en escala Richter ocurridos en Ecuador. *Ingeniería*, 21, 5-64. <https://www.redalyc.org/pdf/467/46753192005.pdf>
- Tmkoczy, A. (1999). Understanding and parameter setting of STA/LTA trigger algorithm. En A. Tmkoczy, *New Manual of Seismological Observatory Practice*.
- Ublox. (s.f.). *NEO-6-GPS*. [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf)
- Valdés Pérez, F. P. (2007). *Microcontroladores: Fundamentos y Aplicaciones con PIC*. Marcobo SA.
https://www.researchgate.net/publication/31840667_Microcontroladores_fundamentos_y_aplicaciones_con_PIC_FE_Valdes_Perez_R_Pallas_Areny
- Valencia, L. (2014). *Modelo Óptimo de una red de sensores inalámbrica para monitorización volcánica en tiempo real* [Tesis de Pre Grado,, Universidad de las Fuerzas Armadas ESPE de Ecuador]. <http://repositorio.espe.edu.ec/handle/21000/9505>
- Valverde, J. F. (2002). *Microzonificación sísmica de los suelos del Distrito Metropolitano de la ciudad de Quito*. <https://biblio.flacsoandes.edu.ec/libros/digital/51553.pdf>

- Wilstrom, E. (2014). *Diseño y desarrollo en django de una aplicación web para la edición, creación y gestión de cuestionarios tipo test on-line* [Tesis de Grado, Universidad Carlos III de Madrid de España]. <http://hdl.handle.net/10016/26673>
- Wong J., H. L. (2009). Automatic time-picking of first arrivals on noisy microseismic data. (CREWES, *University of Houston*.
<https://pdfs.semanticscholar.org/7736/aad97bffc33efe1a5a77e16d7807715c12.pdf>
- Wu, Y.-M. H. (2007). Determination of earthquake τ_c y P_d [Archivo PDF].
<http://www.eqh.dpri.kyoto-u.ac.jp/src/eew/pdf/K04yihminwu.pdf>
- Zambrano Vizuite, A. P. (2015). Sistema Distribuido de Detección de Sismos Usando una Red de Sensores Inalámbrica para Alerta Temprana. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 260-269. <https://doi.org/10.1016/j.riai.2015.06.002>
- Zúñiga Dávila, R. (2011). *SISMOLOGÍA* [Archivo PDF].
http://www.foroconsultivo.org.mx/eventos_realizados/proteccion2_230407/ponencias/14_zuniga.pdf