



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“DISEÑO DE UN PROTOTIPO PARA LA DETECCIÓN DE
MOVIMIENTOS SÍSMICOS BASADO EN UN SISTEMA
EMBEBIDO EN FPGA”

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES**

MISAEL JOEL ORTEGA AVILÉS

JONATHAN ALEXIS IÑIGUEZ MORENO

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTO

A Dios principalmente por ser pilar fundamental en mi vida, por darme las fuerzas necesarias para continuar y no desmayar en todo el transcurso de la carrera. Quien ha sido mi sustento y mi escudo ante toda situación.

A mi mamá quien fue mi apoyo en todo momento, por animarme a seguir adelante y porque siempre han estado pendiente de mis estudios.

Gracias a la Escuela Superior Politécnica del Litoral por tan excelente educación, y por cada uno de los docentes de la institución quienes me han impartido buena enseñanza académica y de la vida.

A mi tutor el Ingeniero Ponguillo por su asesoramiento, que con paciencia y dedicación me orientó a realizar con éxito este proyecto.

Jonathan.

A Dios por haber abierto las puertas de tan prestigiosa universidad, por haberme dado el ánimo frente a circunstancias que parecían imposibles de resolver y por haberme dado las fuerzas para no rendirme aun cuando solo quería renunciar.

A mi madre por ser mi luz y mi guía, por haberme enseñado que nunca es tarde para estudiar y superarse. Sus consejos fueron el pilar fundamental que formaron mi carácter y me llevaron a conseguir este gran triunfo.

A mi padre y hermanos por haber creído en mí y haberme brindado su apoyo incondicional frente a cada necesidad presentada en esta larga carrera.

Gracias a la Escuela Superior Politécnica del Litoral por darme la oportunidad de formar parte de sus estudiantes y haber puesto en mi camino maestros que fueron de gran inspiración.

Misael.

DEDICATORIA

Dedicado a mi madre MERCEDES MORENO JÁTIVA quien me ha apoyado en mis estudios, tanto académica, como de la vida, por ser el motivo principal para culminar con éxito mis estudios. Y de quien he recibido palabras de aliento para avanzar y llegar a ser un profesional.

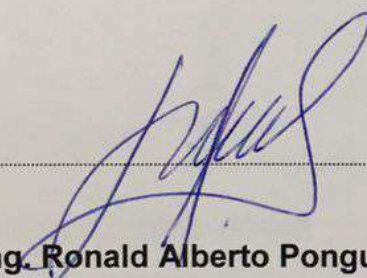
A mi familia por animarme constantemente a seguir adelante y no rendirme a pesar de las circunstancias de la vida, por ser ese apoyo fundamental necesario para darme fuerzas y avanzar.

Jonathan.

Dedicado a toda mi familia por el esfuerzo y sacrificio efectuado por ver a uno de sus miembros alcanzar esta meta. Este triunfo es para ustedes.

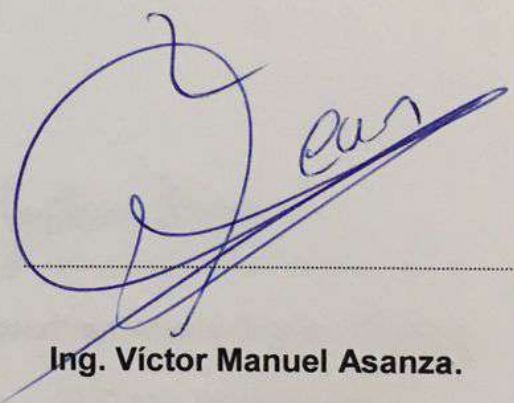
Misael.

TRIBUNAL DE EVALUACIÓN



Ing. Ronald Alberto Ponguillo.

PROFESOR EVALUADOR

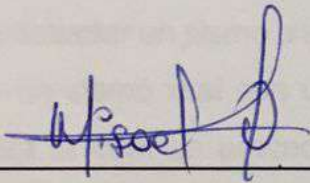


Ing. Víctor Manuel Asanza.

PROFESOR EVALUADOR

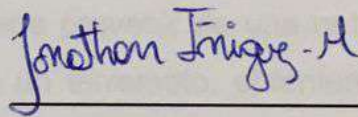
DECLARACIÓN EXPRESA

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



Misael Joel Ortega Avilés

C.C No. 0927020560



Jonathan Alexis Iñiguez Moreno

C.C No. 0917202061

RESUMEN

Los terremotos provocados naturalmente pueden afectar a cualquier país alrededor del mundo, lo cual tiene como resultado grandes daños en una población, dejando secuelas en las personas, tanto físicas como psicológicas, el impacto en los individuos es una problemática que afecta a una población, aumentando el temor, la ansiedad y una inestabilidad en su diario vivir. El formato en el que se presenta esta investigación es tipo descriptiva, analítica-retrospectiva y bibliográfica, y tiene como objetivo detectar, analizar y distinguir un sismo de un movimiento mediante el uso de FPGA y acelerómetro embebidos en una tarjeta de entrenamiento DE0-Nano. La importancia de la presente investigación es diseñar un dispositivo, microcontrolador o CPU capaz de detectar un sismo o terremoto y que logre distinguir entre un movimiento cualquiera de un sismo real con una velocidad que pueda prevenir de una catástrofe natural. Para la cual se procedió una simulación de un terremoto, obteniendo una gráfica visualizada en computadora y con el programa Matlab, para la conectividad se utilizó un dispositivo Bluetooth HC-06 y el uso de la tarjeta de entrenamiento DE0 nano usada en los laboratorios digitales en la facultad FIEC. De los cuales se obtuvo como resultado.

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
TRIBUNAL DE SUSTENTACIÓN.....	iv
RESUMEN	v
DECLARACIÓN EXPRESA.....	vi
ÍNDICE GENERAL.....	vii
CAPÍTULO 1	1
1. DELIMITACIÓN DEL PROBLEMA.....	1
1.1 Planteamiento del problema.....	1
1.2 Objetivos generales.....	3
1.3 Objetivos específicos.....	3
1.4 Justificación.....	3
1.5 Alcance del proyecto.....	4
CAPÍTULO 2.....	6
2. ESTADO DEL ARTE, MARCO TEÓRICO Y METODOLOGÍA.....	6
2.1 Antecedentes.....	6
2.2 Marco teórico.....	7
• Terremoto.....	7
• Hipocentro.....	8
• Epicentro.....	9
• Medición de terremotos.....	10
• Escalas.....	11

• Medición de escala de Richter.....	10
• Energía.....	11
• FPGA.....	12
• QUARTUS II.....	13
• Tarjeta DE0-Nano.....	14
• Partes de la DE0-Nano.....	15
• FPGA Cyclone IV.....	15
• G Sensor (AD IADXL345).....	15
• Arduino.....	16
• Módulo bluetooth.....	17
2.3 Metodología.....	18
2.3.1 Diseño general de la solución.....	18
2.3.2 Diseño del hardware sobre la tarjeta DE0-nano.....	19
2.3.3 Metodología para detección de patrones.....	20
2.3.4 Almacenamiento de datos.....	21
CAPÍTULO 3.....	23
3.1 Pruebas y resultados.....	23
3.2 Escenario 1.....	23
3.3 Escenario 2.....	27
3.4 Conclusiones y Recomendaciones.....	31
3.5 Bibliografía.....	32
3.6 Anexos.....	34

CAPÍTULO 1

1. DELIMITACIÓN DEL PROBLEMA.

1.1 Planteamiento del problema.

Existen múltiples razones por la cual la tierra o la base del suelo que pisamos pueden temblar, una de las razones es el resultado de la actividad humana, según explica Richard Davies en un reportaje para BBC el 18 de septiembre del 2015, donde afirma que:

"La primera vez que se tomó conciencia sobre este fenómeno fue en la década de los 60, en Estados Unidos, cuando se produjeron terremotos después de que se introdujera en el suelo desechos militares en forma de fluidos".

Según el investigador esta fue la primera vez en que se dieron cuenta de que al inyectar agua debajo de la tierra podría dar origen a terremotos. Otras de las actividades que pueden dar lugar a terremotos según el investigador son: la minería, la extracción de energía geotermal, rellenar un embalse con agua, inyectar residuos líquidos, inyectar fluidos, así como la actividad y métodos convencionales de extracción de gas y petróleo.

Sin embargo, los sismos o terremotos más destructivos son aquellos que se producen naturalmente, como es el movimiento de las placas tectónicas, debido a la liberación brusca y súbita de energía acumulada por deformación de la Litosfera, que se proyecta en forma de ondas sísmicas. En su mayoría los terremotos son de origen tectónico, debido a la fricción en las fallas es a menudo inestable, ocurriendo desplazamientos muy rápidos como una ruptura que se propaga dinámicamente sobre la superficie de la falla, estos movimientos generan ondas sísmicas al llegar a la superficie, provocando sacudidas sísmicas del terreno.

Los terremotos provocados naturalmente pueden afectar a cualquier país alrededor del mundo. En Ecuador desde 1949, en que un terremoto de 6,8 grados en la escala de Richter, afectara a nuestra Sierra ecuatoriana, en especial a la ciudad de Ambato, nadie hubiera podido pensar que un movimiento telúrico tan intenso se volvería a sentir. A pesar de esto, el terremoto de 7,8 grados ocurrido el 16 de abril, cuyo epicentro fue entre las provincias de Manabí y Esmeraldas, sorprendió y lleno de dolor a todos los ecuatorianos. Las provincias más afectadas han tenido dolorosas pérdidas tanto humanas como materiales según relataron los medios de comunicación.

El terremoto producido dejó secuelas en el pueblo ecuatoriano, tanto físicas como psicológicas, en promedio en la ciudad de Guayaquil un gran porcentaje manifiesta sensaciones de movimientos telúricos pero que en realidad no se trata de un sismo sino del paso de una gran masa como, por ejemplo: un camión, un bus, una grúa, etc. El impacto psicológico es mayor en las ciudades que fueron más afectadas como en la Provincia de Manabí, en el que apenas sentir u oír una vibración la población se empieza a alarmar creyendo que se podría repetir la historia de aquel 16 de abril.

En la actualidad, aún con los avances tecnológicos no se ha logrado alcanzar un dispositivo que pudiera predecir cuándo ocurrirá un terremoto, pero existen métodos para capturar sus ondas, analizarlas y calcular la llegada de ondas destructivas o movimientos de tierras violentas.

La evolución de la tecnología ha llevado a alcanzar varios métodos de análisis de sismos, pero aún no se ha logrado la completa distinción entre un sismo y un movimiento de cualquier índole que no necesariamente sea catalogada como sismo y de esta manera evitar falsas alarmas.

Por tal razón es ideal e importante encontrar o diseñar un dispositivo, microcontrolador o CPU capaz de detectar un sismo o terremoto y que logre distinguir entre un movimiento cualquiera de un sismo real con una velocidad que pueda prevenir de una catástrofe natural.

1.2 Objetivo General.

Implementar un prototipo de discriminación de movimientos y vibraciones que permita la detección temprana de actividades sísmicas en tiempo real, utilizando FPGA y acelerómetro embebidos ambos en la tarjeta de entrenamiento DE0-Nano.

1.3 Objetivos Específicos.

Diseñar un mecanismo que permita diferenciar entre un movimiento cualquiera y las vibraciones causadas por un terremoto usando como plataforma la tarjeta DE0-NANO.

Crear una red redundante entre dos sistemas embebidos FPGA que aseguren una comunicación fluida entre sí de los eventos registrados.

Diseñar un enlace de transmisión en tiempo real que permita obtener los datos de los sistemas embebidos FPGA para enviarlos a una base de datos.

1.4 Justificación.

Pensando en la problemática ya antes mencionada, en las réplicas que existieron después del terremoto del 16 de abril y el impacto psicológico de muchas personas en el que suponían la venida de otro sismo de igual magnitud al sentir un movimiento, surge la idea de crear un dispositivo que sea capaz de detectar un sismo con el uso de la tarjeta de entrenamiento DE0 nano usada en los laboratorios digitales en la facultad FIEC.

Aprovechando el conocimiento obtenido de la tarjeta DE0 nano y sus componentes integrados en la tarjeta, nos permite encontrar el acelerómetro, procesador y una característica en particular que es una frecuencia de reloj de 50MHZ permitiendo así una alta velocidad de respuesta en comparación a otros microprocesadores en una misma tarjeta alcanzando así ahorrar espacio y dinero en tratar de conseguir los diferentes componentes.

Una de las razones de mayor importancia por la que se eligió utilizar el controlador de esta tarjeta es por la velocidad a la que trabaja siendo esta mayor que muchos de los controladores, por ejemplo, la familia de pics soportan una máxima frecuencia de reloj de 20 MHz y arduinos 16MHz.

El proyecto demanda el desarrollo de un algoritmo para la detección o cambios en el acelerómetro mediante programación en la plataforma NIOS II de tal forma que se permita la adquisición y almacenamiento de datos censados por el controlador para luego proceder a realizar los cálculos pertinentes para hallar el epicentro, magnitud y tiempo de llegada a otras locaciones y luego comunicarlo a otra tarjeta la cual dará la alarma de que está por llegar un terremoto.

Los autores tienen como objetivo detectar, analizar y distinguir un sismo de un movimiento mediante el uso de FPGA y acelerómetro embebidos ambos en la tarjeta de entrenamiento DE0-Nano y de esta manera comunicar un dato significativo como es el tiempo en que llega un sismo a cierta población o ubicación.

1.5 Alcance del Proyecto.

Se realiza un estudio analítico-retrospectivo de terremotos ocurridos en diferentes lugares para obtener información relevante para el proyecto, como es la discriminación de vibraciones, el tiempo de llegada de las ondas sísmicas superficiales con el fin de armar el código que interpretará las variaciones de aceleración censados por el acelerómetro y asegurar la confiabilidad tanto en discriminación de vibraciones como en el tiempo de llegada de las ondas superficiales.

Teniendo en cuenta que el prototipo es para el censo de un sismo, para calificar la efectividad necesitamos que ocurra un terremoto y tener el equipo listo para evaluar el mismo. Por lo tanto, se procederá a tener una base de prueba la cual realizará una simulación de un terremoto, obteniendo una gráfica visualizada en computadora y con el programa Matlab, y para la conectividad con el segundo módulo se utilizará un dispositivo Bluetooth HM-11 tomando en cuenta que solo

es un prototipo y que para la prueba experimental no requerimos de largas distancias de este modo se establece comunicación entre tarjetas y así una alarma a la otra con el avistamiento de un terremoto.

Debido a que las ondas primarias, secundarias y superficiales de un sismo se pueden diferenciar se realizará las investigaciones para poder realizar una correcta interpretación de las señales captadas. Los valores a los que se tendrá acceso serán los emitidos por el acelerómetro obteniendo respuesta en X, Y y Z

Se analizan las gráficas obtenidas descartando cualquiera en la que no se aprecie la diferencia de las ondas llegadas.

Luego del análisis de la onda obtenida en el censo del sismo, se procede a establecer los puntos de inicio de la onda primaria y secundaria ya que, con la identificación de estas ondas, de su tiempo se puede obtener la distancia en la que ocurrió el sismo, luego de esto procede a enviar una notificación al segundo módulo si es que este análisis lo identifica como un terremoto.

CAPÍTULO 2

2. ESTADO DEL ARTE, MARCO TEÓRICO Y METODOLOGÍA.

2.1 Antecedentes

En la ciudad de Entre ríos, Argentina, en el año 2007 se realizó un proyecto denominado: "Detector sísmico con acelerómetro xyz mma7260q". El cual utilizaba un chip "acelerómetro xyz mma7260q" para la detección del movimiento en cualquiera de sus tres ejes, con ayuda de un microcontrolador MC68HC908QY4 para la comparación de las tensiones en el acelerómetro. El dispositivo logró desarrollar un método de detección mediante dos etapas, una de aprendizaje y otra de umbral. En la etapa de aprendizaje se leían nuevos valores detectados por el sensor acelerómetros, mientras en la etapa de umbral se comparaban con valores ingresados por el usuario. Si los valores detectados por el dispositivo superaban a los valores almacenados en la memoria, una alarma era activada para el aviso de una ocurrencia sísmica fuera de lo común [1].

Aunque es un prototipo y está en experimento, este es una aplicación basada en el acelerómetro ubicada en los teléfonos inteligentes, el cual con ayuda de una red de dispositivos puede censar y avisar el sismo ocurrido. Como resultado los smartphones y tablets que pueden detectar sismos por el acelerómetro embebido en cada dispositivo. "Cuando se detecta un temblor, los usuarios que han instalado esta aplicación reciben una notificación casi de inmediato. A medida que las ondas del sismo viajan a una velocidad determinada (de 5 a 10 km/s), **es posible notificar a la población que no ha sido alcanzada por las olas destructivas del sismo**" [2].

Proyecto fin de curso "diseño de un sistema de estabilización utilizando sensores inteligentes": Proyecto realizado con un microcontrolador como procesador de la señal emitida por el acelerómetro pudiendo interpretar la señal, o movimiento que

haya experimentado el acelerómetro. El uso del acelerómetro es indicado en aplicaciones donde se necesite tener una medida de inclinación, siempre y cuando el lugar donde se instale no esté sometido a vibraciones significativas o a movimientos que provoquen aceleraciones a lo largo de sus ejes [3].

2.2 Marco teórico

- **Terremoto.**

Un terremoto es un movimiento brusco y repentino de Tierra, causado por liberación de energía acumulada por tiempo prolongado.

La corteza terrestre está conformada por una serie de placas con un grosor aproximado de 70 km, cada una con diferente composición física-química, estas placas conocidas como placas tectónicas se acomodan constantemente en procesos que llevan años y van dando forma a lo que se conoce hoy como superficie de nuestro planeta, dándose así origen a los continentes y los relieves geográficos (véase la figura 2.1).

Por lo general estos movimientos son lentos e imperceptibles, pero en ocasiones estas placas chocan entre sí impidiendo su desplazamiento produciendo que una placa comience a desplazarse sobre o bajo otra originando lentos cambios en la topografía.

Si se dificulta el desplazamiento comienza a acumularse una energía de tensión que se liberará en algún momento y una de las placas se moverá bruscamente contra la otra rompiéndola, liberándose así una cantidad de energía originando un terremoto [4].

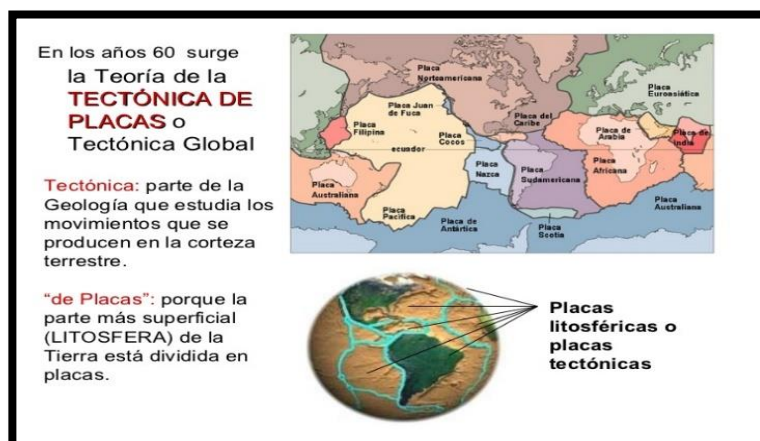


Figura 2.1. Capas tectónicas

La figura 2.1 muestra la distribución de las capas tectónicas a lo largo de la superficie terrestre, como se puede observar los límites de las placas no lo establecen los continentes y cada placa puede estar conformada por continentes y océanos.

- **Hipocentro.**

Es un punto en la profundidad de la tierra ver figura 2.2 en el que se libera energía ocasionada por un terremoto [4].

Dependiendo de la profundidad dicho hipocentro se denomina en:

- Superficial con $H < 70$ km de profundidad.
- Intermedio con $70 \text{ km} < H < 300$ km de profundidad.
- Profundo con $H > 300$ km de profundidad.

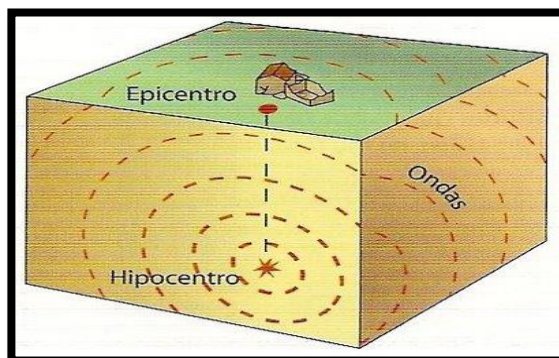


Figura 2.2. Localización de hipocentro

La figura 2.2 muestra la forma en la cual se propaga un sismo desde su origen, las ondas viajan en todas direcciones a partir de la energía que está siendo liberada desde el hipocentro.

- **Epicentro.**

Epicentro es el punto de la superficie de la tierra encima del llamado hipocentro (véase figura 2.3), justamente en este punto es donde la intensidad del terremoto es de mayor magnitud [4].

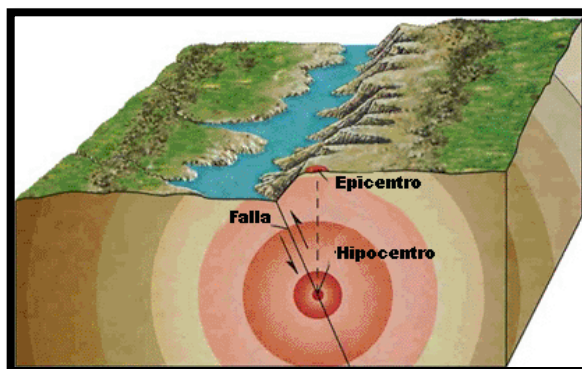


Figura 2.3. Localización de epicentro.

Se puede observar en la figura 2.3 que el epicentro se establece perpendicularmente al hipocentro sobre la corteza terrestre y la distancia que

podemos observar entre el hipocentro y el epicentro se conoce como la profundidad del evento sísmico.

Medición de los terremotos.

La medición de este tipo de sismo es a través de un instrumento de medición llamado sismógrafo (vea Figura 2.4), el cual registra la vibración de la tierra provocada por el sismo y cuyo resultado se llama sismograma. La información que nos brinda el sismograma es en su esencia magnitud y duración [4].

El sismógrafo logra registrar dos tipos de ondas:

- las ondas centrales o corporales: las cuales son ondas que viajan a través de la profundidad de la superficie.
- Las ondas superficiales: las cuales son ondas que viajan a través de la superficie terrestre y que producen mayor daño por producir mayor vibración.

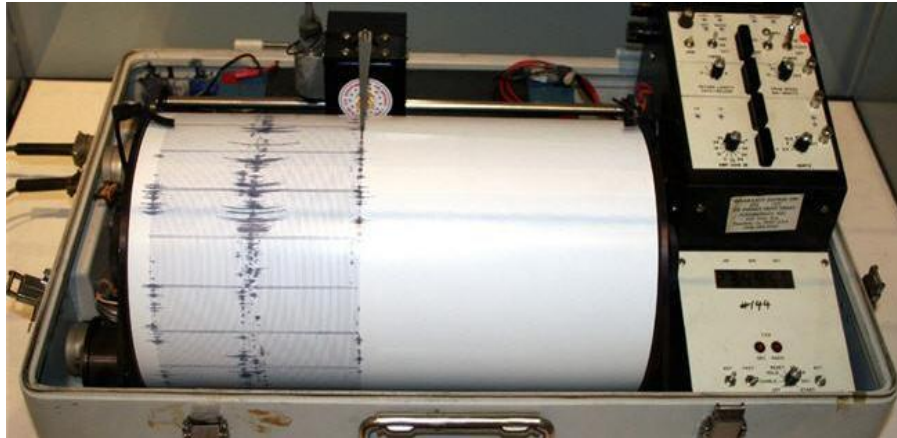


Figura 2.4. Figura de sismógrafo y sismograma.

En la figura 2.4 podemos observar el instrumento de medición de eventos sísmicos conocido como sismógrafo y el resultado de la medición mostrado en gráficas conocidas como sismograma.

- **Escalas.**

Cada terremoto tiene un valor asignado único de magnitud, el cual varía grandemente dado algunos factores como son [4].

- La distancia.
- Las condiciones del terreno.
- Los estándares de construcción.
- Pero para mayor facilidad se lo clasifica según su energía intrínseca.

- **Magnitud de escala Richter.**

Esta magnitud representa la energía liberada en un terremoto, basa su registro en la información dada por un sismógrafo.

La escala crece potencialmente de manera que cada punto de aumento puede significar un aumento de energía 10 o más veces [4].

3.5 – 5.4	a menudo se siente, causa daños menores.
5.5 – 6.0	ocasiona daños ligeros a edificios.
6.1 – 6.9	puede ocasionar daños en áreas pobladas.
7.0 – 7.9	terremoto mayor, ocasiona graves daños.
8.0 o mayor	gran terremoto. Destrucción a comunidades cercanas

- **Energía.**

La mejor manera de entender la energía disipada por un terremoto según la escala de Richter es comparándolo con la energía de detonación de TNT (véase figura 2.5).

Cada grado que aumenta en magnitud, la energía aumenta hasta 30 veces [4].

Magnitud Richter	Equivalencia en TNT	Ejemplo
-1.5	1 gr	Romper una piedra
1.0	6 onz	barreno pequeño
1.5	2 libras	
2.0	13 libras	
2.5	63 libras	
3.0	397 libras	
3.5	1000 libras	Mina
4.0	6 tn	
4.5	32 tn	Tornado
5.0	199 tn	
5.5	500 tn	Terremotos
6.0	1270 tn	
6.5	31550 tn	
7.0	199000 tn	
7.5	1 Megatón	
8.0	3.27 Megatonnes	
8.5	31.55 Megatonnes	
9.0	200 Megatonnes	
10.0	6300 Megatonnes	Falla de San Andrés
12.0	1 Gigatón	Romper la tierra en 2 o Energía solar diariamente recibida

Figura 2.5. Tabla comparativa escala de Richter – equivalencia energética liberada.

En la figura 2.5 se puede observar que a medida que aumenta la escala de Richter la diferencia de energía liberada entre cada magnitud crece de una manera exponencial, tanto así que un movimiento sísmico de magnitud 6.5 equivaldría a experimentar aproximadamente 30 movimientos sísmicos de magnitud 6.0.

- **FPGA (Fields programmable gate arrays).**

Cuya traducción al español sería arreglo de puertas programables en campo, son dispositivos semiconductores que se basan en una matriz de bloques lógicos programables, conectados por medio de interconexiones programables. Las FPGA se pueden reprogramar a los requisitos deseados de la aplicación.

Esta característica distingue a las FPGA's de los circuitos integrados específicos (ASIC's) que se fabrican para tareas específicas de diseño y su

interconexión, funcionalidad pueden ser configuradas mediante un lenguaje de descripción de hardware.

Debido a su reprogramación y reconfiguración ha llegado a ser muy bien acogido en el mundo de la electrónica digital, por su flexibilidad a la hora de diseñar y optimizar así recursos físicos y lógicos.

La FPGA tiene un rango de aplicaciones muy amplio. Siendo las principales aplicaciones:

- Procesamiento digital de señales (DSP)
- Comunicaciones
- Procesado de datos.

Algunas de las razones más importantes por la que se elige una FPGA para aplicaciones de tratamiento de señal son:

- Alta frecuencia de trabajo.
- Su capacidad de procesamiento en paralelo.
- Y su bajo precio en comparación con los ASIC [5].

- **Quartus II.**

Es una herramienta de software creado por "ALTERA" para el análisis y síntesis de diseños realizados en HDL.

Este software permite al desarrollador compilar diseños, realizar análisis temporales, examinar diagramas RTL y configurar el dispositivo de destino con el programador [6].

Existen otras ediciones de Quartus las cuales son:

- Quartus II edición web: Esta edición es una versión gratuita de Quartus II que puede descargarse o ser enviada vía email, aunque esta edición permite solo la compilación y la programación de un número limitado de dispositivos de ALTERA.

- Edición de suscripción de Quartus II: Esta versión también es descargable gratuitamente, pero debe pagarse una licencia para poder disfrutar de todas las funciones de la aplicación. Se puede utilizar la licencia gratuita de la EDICION WEB en esta edición de quartus II, pero también se restringe el número de dispositivos a utilizar con la aplicación.
- **Tarjeta DE0-nano.**

DE0-Nano es una herramienta de desarrollo para FPGA de uso académico basada en la FPGA de la marca Altera y la familia Cyclone IV, de tamaño compacta adecuada para la creación de prototipos de diseño de circuitos, como robots y proyectos portátiles (véase figura 2.6).

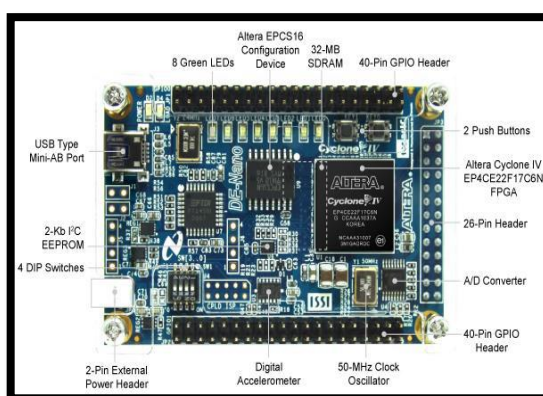


Figura 2.6. Tarjeta de desarrollo DE0-Nano.

Como se puede observar en la figura 2.6 la DE0-NANO tiene una colección de interfaces que incluyen dos cabezales GPIO externos para extender diseños más allá de la placa DE0-NANO, dispositivos de memoria integrados incluyendo SDRAM y EEPROM para un mayor almacenamiento de datos, así como periféricos generales de usuario con Leds y botoneras [7].

Las ventajas de la placa DE0-NANO son: su tamaño y peso, así como su capacidad para ser reconfigurado sin llevar un hardware extra, diferenciándose de otros tableros de desarrollo de propósito general.

Para diseños móviles donde la energía portátil es de vital importancia, la DE0-NANO ofrece a los diseñadores varias opciones de esquema de energía, incluyendo un puerto USB mini-AB, y un conector externo de alimentación de 2 pines y dos clavijas DC de 5V.

- **Partes de la DE0-Nano.**

- FPGA cyclone IV EPC4CE22
- Interfaces de entrada y salida: puertos de expansión, convertidor analógico digital, acelerómetro de 3 ejes.
- Memoria: 32 MB de SDRAM, 2 kb EEPROM.
- Conmutadores y leds: 8 leds verdes, 4 interruptores DIP, dos pulsadores sin rebote.
- Un reloj de 50 MHZ.

- **FPGA Cyclone IV.**

La familia de FPGA de Cyclone IV de Altera extiende el liderazgo de la serie FPGA de Cyclone al ofrecer los FPGAs de menor consumo de potencia y menor coste del mercado, ahora con una variante de transceptor. Ideal para aplicaciones de alto volumen y sensibles al costo, los Cyclone IV FPGAs le permiten cumplir con los requisitos de ancho de banda cada vez mayor y reducir los costos [7].

- **G Sensor (ADI ADXL345).**

El ADXL345 es un sistema de medición de aceleración de tres ejes completo con un rango de medición seleccionable de ± 2 g, ± 4 g, ± 8 g o ± 16 g (véase figura 2.7). Mide tanto la aceleración dinámica resultante del movimiento o choque como la aceleración estática, como la gravedad, lo que permite su uso como sensor de inclinación. El sensor es una superficie micro mecanizada de poli silicio construida sobre una oblea de silicio. Los resortes de poli silicio suspenden la estructura sobre la superficie de la oblea y proporcionan una

resistencia contra las fuerzas de aceleración. La deflexión en la estructura se mide utilizando capacitores diferenciales que consisten en placas y placas fijas independientes unidas a la masa móvil. La aceleración desvía el haz y desbalancea el condensador diferencial, dando como resultado una amplitud proporcional a la aceleración. La modulación de sensibilidad de fase se utiliza para determinar la magnitud y la polaridad de la aceleración [8].

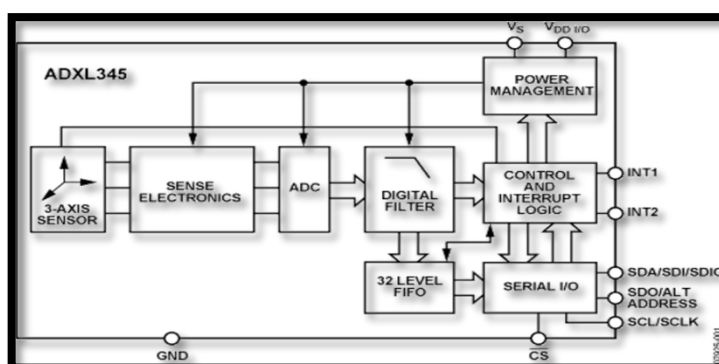


Figura 2.7. Chip acelerómetro (ADXL345) diagrama de bloques.

El diagrama de bloques mostrado en la figura 2.7 nos muestra el funcionamiento interno del chip acelerómetro ADXL345, en donde se observa que el resultado de la medición de los 3 ejes es condicionado mediante sensores electrónicos y convertido a formato digital para posteriormente presentarlo en variables enteras que puedan usarse en la programación del mismo.

- **Arduino.**

Es una plataforma para prototipos de circuitos electrónicos (open-source) basada en hardware y software flexibles (véase figura 2.8). Está pensado para diseñadores y para cualquier persona interesada en crear objetos o entornos interactivos.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores, etc.

Otra ventaja de Arduino es que las placas pueden ser ensambladas a mano o adquirirlas pre ensambladas y el software descargable gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que el diseñador libre de adaptar a sus necesidades [9].

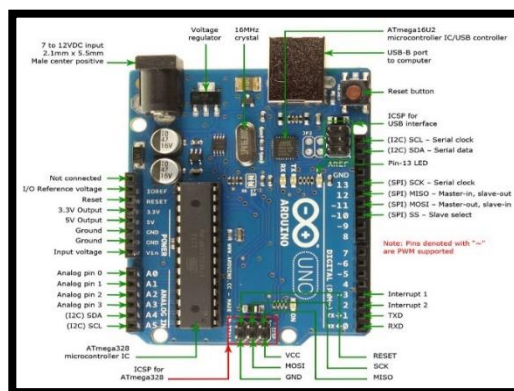


Figura 2.8. Tarjeta de desarrollo Arduino.

En la figura 2.8 se muestra la tarjeta de desarrollo Arduino UNO, consta de entradas analógicas y digitales, salidas PWM, conexión USB, alimentación de 12 VDC, botón de reset, entre otras funciones básicas de un microcontrolador.

- **Modulo Bluetooth (hc-06).**

El dispositivo Bluetooth HC-06 debido a sus puertos de transmisión y recepción de datos TXD y RXD respectivamente nos permite realizar comunicaciones inalámbricas de 10 metros de distancia máximo (véase figura 2.9).

Teniendo en cuenta el fácil manejo de estos dispositivos, su SPP (Serial Port Protocol), su reducido tamaño y al bajo consumo, el módulo de Bluetooth resulta muy atractivo para aplicaciones con microcontroladores y tarjetas de desarrollo tipo Arduino [10].

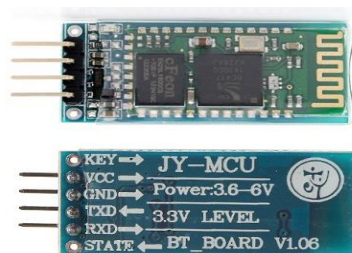


Figura 2.9. Dispositivo Bluetooth (hc-06).

La figura 2.9 muestra la parte frontal y posterior del módulo HC06, donde podemos observar que consta de 4 pines para su conexión: VCC, GND, TX y RX. El nivel de tensión soportado varía entre 3.3 VDC y 5 VDC.

2.3 Metodología.

2.3.1 Diseño general de la solución.

La tarjeta DE0-nano se encarga de realizar el procesamiento de datos y controlar el almacenamiento de información en caso de que los eventos ocurridos así lo ameriten. Todo movimiento considerado por la tarjeta DE0-nano como un posible evento sísmico será exportado vía inalámbrica para ser guardada en una tarjeta microSD a través de un Arduino UNO (véase figura 2.10).

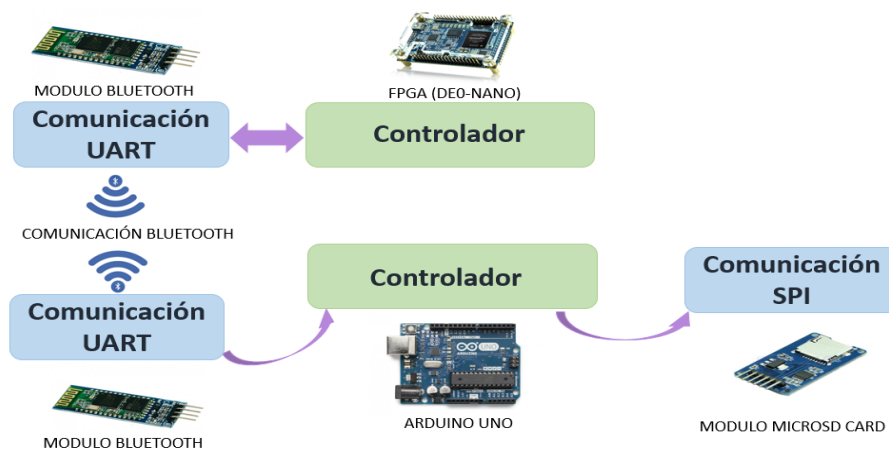


Figura 2.10. Diagrama de bloques general del sistema

La figura 2.10 muestra un diagrama de bloques del diseño general de la solución. El acelerómetro interno de la tarjeta DE0-nano se encarga de proveer la aceleración en los tres ejes (x, y, z). Esta información es procesada por la tarjeta DE0-nano para determinar si se ha producido un movimiento sísmico. De ser el caso, los datos del acelerómetro serán transmitidos vía Bluetooth a un Arduino UNO, el cual se encargará de almacenar dichos datos en una tarjeta micro SD a través de un módulo microSD.

2.3.2 Diseño del hardware sobre la tarjeta DE0-nano.

Dado el diseño general de la solución, se requiere un sistema de hardware implementado sobre la tarjeta DE0-nano que posea altas velocidades de procesamiento de datos. Este sistema estará en la capacidad de comunicarse con el acelerómetro interno de la tarjeta DE0-nano, con una computadora y con un módulo externo de comunicación serial RS-232 (véase figura 2.11).

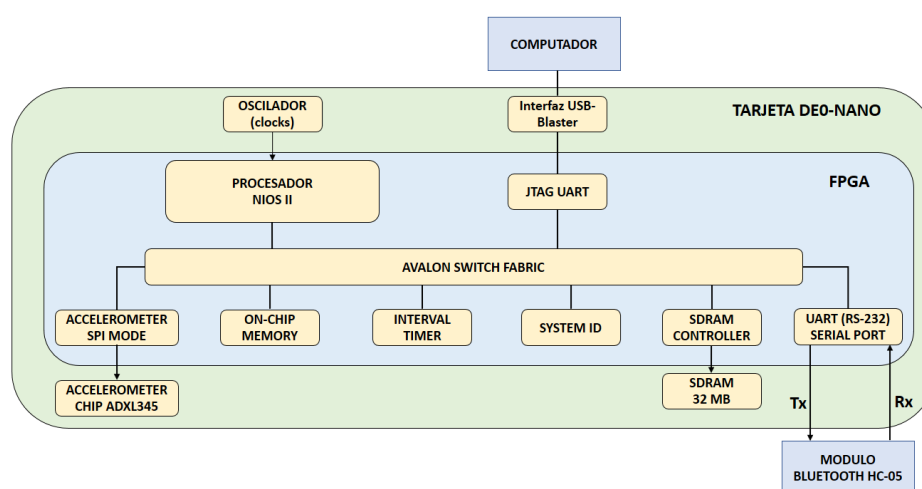


Figura 2.11. Diagrama de bloques del sistema de hardware implementado sobre la tarjeta DE0-nano

Como se puede apreciar en la figura 2.11 se han añadido los componentes necesarios para establecer la comunicación entre los dispositivos de

comunicación externos e internos. La comunicación con una computadora servirá para analizar la respuesta del sistema mediante los datos observados en la consola de Nios II.

2.3.3 Metodología para detección de patrones

Luego de efectuar un profundo análisis sobre los datos registrados por los acelerógrafos de distintas partes del mundo, se hallaron patrones comunes en las magnitudes de las ondas sísmicas. Por lo cual se establecieron rangos de amplitud y de tiempo para la detección de las ondas S, P y L (véase figura 2.12).

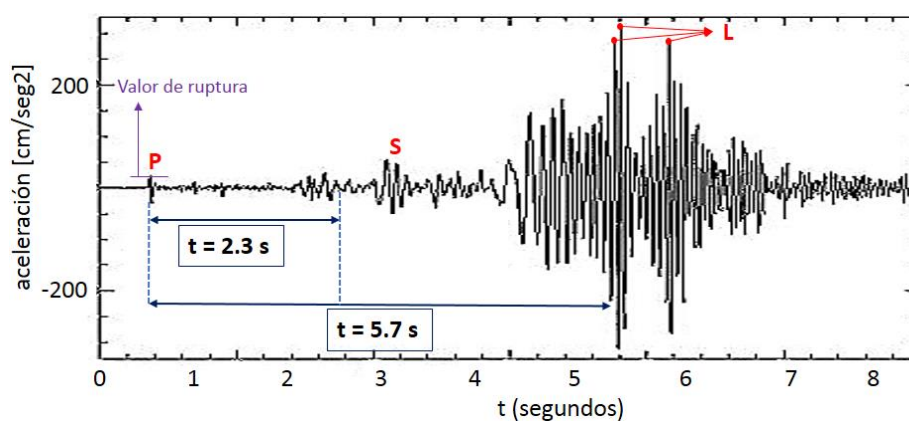


Figura 2.12. Algoritmo de reconocimiento de una onda sísmica

La figura 2.12 muestra la forma en que opera el algoritmo de reconocimiento de patrones de onda sísmica. Primero aclararemos que para este prototipo se han definido rangos de detección del epicentro entre 20 km y 50 km, ya que los estudios realizados muestran un menor porcentaje de error para estos rangos. Pues bien, si queremos un mínimo de 20 km, el tiempo entre la llegada de la onda P y la llegada de la onda S debería ser 2.3 s, y si queremos un máximo de 50 km esa diferencia de tiempo debería ser de 5.7 s, esto se debe a que se ha considerado un aproximado de 8.7 km/s como una velocidad promedio de propagación de las ondas S y P en distintas superficies.

Una vez definidos rangos de tiempo, se definieron rangos de amplitud. Inicialmente se ha definido un valor de ruptura, si el acelerómetro envía una lectura que supera este valor entonces el sistema almacena este valor como el valor pico de la onda P, luego empieza a contabilizar un tiempo de espera de 2.3 s, tiempo durante el cual podrá cambiar el valor del pico inicial de la onda P por otro que supere este valor. A partir de 2.3 s el sistema espera hasta 5.7 s la llegada de la onda S. El pico de la onda S debería ser al menos 2 veces el pico de la onda P, de ser así el sistema pasará a la última etapa en donde esperará al menos 3 picos de las ondas L. Estos valores deberán ser mayores al pico de la onda S, de tal forma que el pico de la onda S represente el 30% de cualquiera de los tres picos de las ondas L registrados.

2.3.4 Almacenamiento de datos

Se utilizó un arduino UNO el cual almacena en una tarjeta microSD, todos los datos recibidos por el puerto serial a través del módulo bluetooth HC-06. La tarjeta microSD se puede extraer en cualquier momento del módulo microSD para revisar la información almacenada.

Para lograr establecer el orden en el cual se almacenan los datos fue necesario diseñar un diagrama de flujo que permita establecer el procedimiento necesario a seguir por la tarjeta de desarrollo arduino en cada secuencia del programa con el fin de garantizar que se almacenen sólo los datos necesarios y se generen los archivos correspondientes (véase figura 2.13).

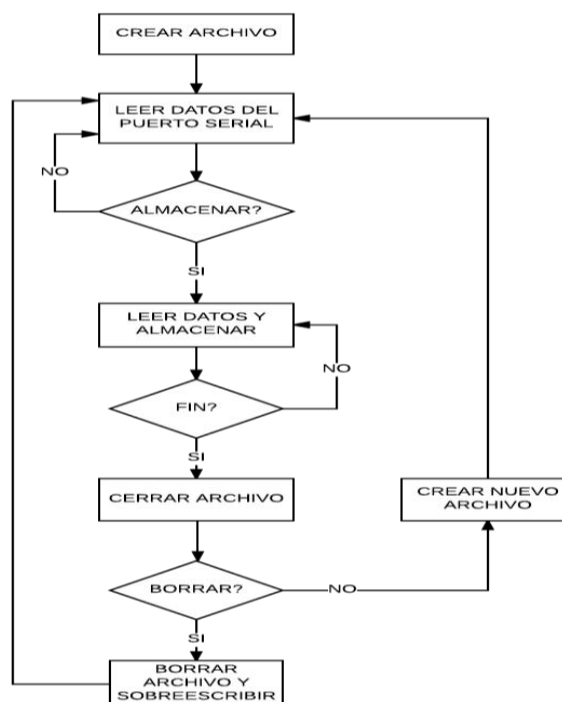


Figura 2.13 Diagrama de flujo del almacenamiento datos en la tarjeta microSD

El diagrama de flujo que se muestra en la figura 2.13 representa la forma en que los datos son almacenados a través del Arduino UNO. Primero se empieza creando un archivo y se espera la señal proveniente de la tarjeta DE0-nano para empezar a almacenar los datos leídos en el puerto serial, si luego de 500 ms la tarjeta DE0-nano no ha detectado un posible movimiento sísmico, enviará una orden hacia Arduino para indicarle que cierre el archivo, lo borre y vuelva a crear un archivo con el mismo nombre. Caso contrario, se creará un nuevo archivo con un nombre distinto para seguir almacenando datos de un nuevo evento.

CAPÍTULO 3

3. PRUEBAS Y RESULTADOS.

Se efectuaron pruebas en diferentes escenarios para poder hacer una comparación de los resultados obtenidos y así determinar si el sistema es capaz de descartar de forma autónoma ondas que no sigan el patrón de una onda sísmica característica.

3.1 Escenario 1

Para las primeras pruebas realizadas se ha utilizado un bloque de cemento de aproximadamente 30 kg sobre el cual fue colocada tarjeta DE0 como lo muestra la figura 3.1. El propósito del ajuste es poder capturar con la mayor aproximación posible las vibraciones a las cuales se encuentre sometido el bloque de cemento.



Figura 3.1. Tarjeta DE0 nano ubicada sobre un bloque de cemento.

El bloque utilizado posee un diámetro aproximado de 26 cm y fue colocado a nivel del suelo en donde se aplicaron fuerzas externas sobre el mismo.

Durante las primeras pruebas los datos fueron exportados a partir de la consola de NIOS II, luego fueron convertidos a un documento en Excel para

posteriormente exportar estos datos a la plataforma de Matlab donde se puedan apreciar las gráficas de los tres distintos ejes tal como lo muestra la figura 3.2.

	1	2	3	4	5
1	0	2	12	233	
2	10	2	9	230	
3	20	1	12	232	
4	30	1	12	230	
5	40	0	10	232	
6	50	1	11	231	
7	60	0	11	230	
8	70	2	10	233	
9	80	0	10	231	
10	90	0	10	230	
11	100	1	11	231	
12	110	1	12	231	
13	120	1	11	231	
14	130	2	12	230	
15	140	1	10	233	
16	150	0	11	230	
17	160	0	12	232	
18	170	1	9	232	
19	180	2	12	231	
20	190	1	11	233	
21	200	2	9	230	
22	210	0	11	232	
23	220	1	12	230	
24	230	0	10	233	
25	240	1	11	233	
26	250	0	11	230	

Figura 3.2. Datos exportados a Matlab

Los datos mostrados en la figura 3.2 fueron exportados de la siguiente manera: primera columna representa el tiempo en milisegundos, segunda, tercera y cuarta columna son los datos registrados por el acelerómetro en el eje X, eje Y, eje Z respectivamente. Luego de esto se realizó una gráfica en el eje X que muestra el resultado bajo situaciones normales, es decir cuando no se ha sometido a ninguna perturbación (véase figura 3.3).

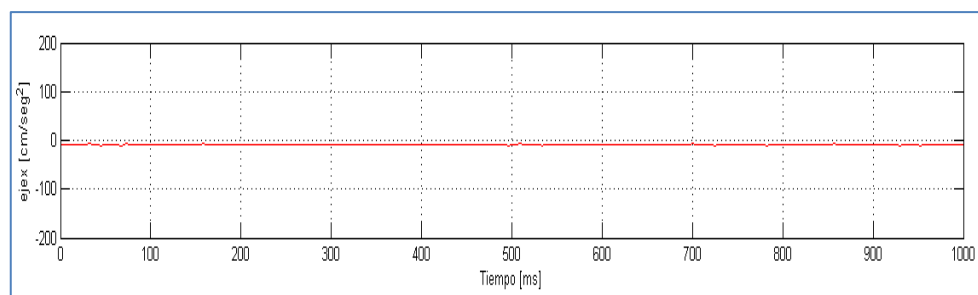


Figura 3.3. Gráfica aceleración vs tiempo en el eje X

Se puede apreciar en la figura 3.3 que los valores registrados sin perturbaciones externas se encuentran cercanos al origen, tomando valores aproximados a 0 cm/s^2 . Así también fueron obtenidos los resultados en el eje Y (véase figura 3.4).

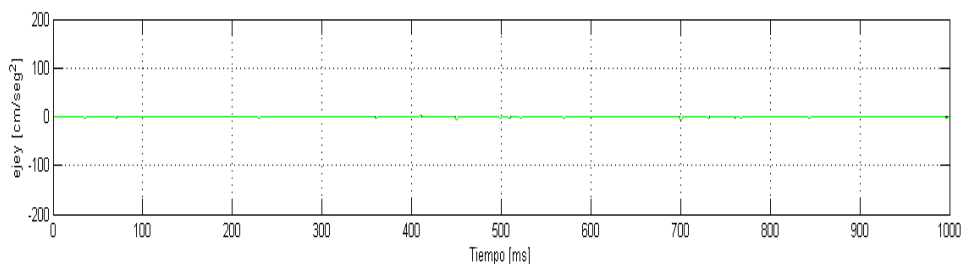


Figura 3.4. Gráfica aceleración vs tiempo en el eje Y

A diferencia del eje X, en la figura 3.4 podemos observar que esta vez los datos se han desplazado aproximadamente 10 cm/s^2 sobre el origen. De igual forma se pueden apreciar los resultados en el eje Z (véase figura 3.5).

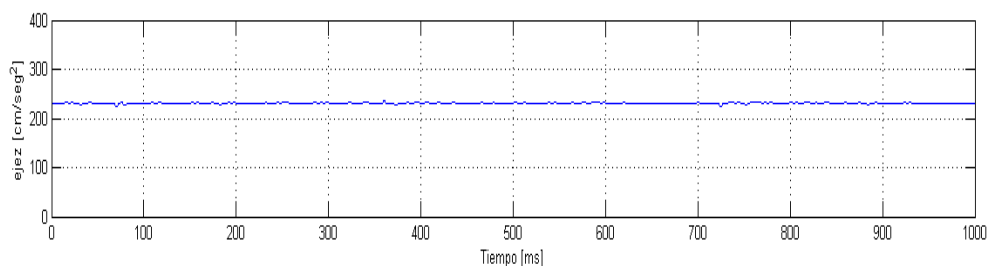


Figura 3.5. Gráfica aceleración vs tiempo en el eje Z

Los datos registrados en el eje Z que se muestran en la figura 3.5 dan a conocer que los valores para los cuales empieza a medir se encuentran desplazados alrededor de 230 cm/s^2 .

Posterior a esto una vez que ya han sido establecidos los valores de control, se efectuó las gráficas de las primeras perturbaciones efectuadas sobre el bloque de cemento, obteniendo una similitud en los tres distintitos ejes que experimentaron este movimiento (véase figura 3.6).

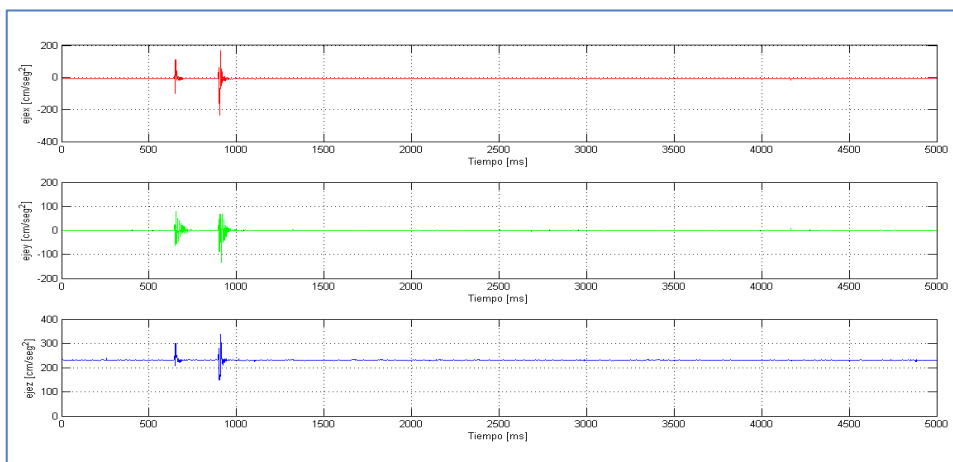


Figura 3.6 Gráfica aceleración vs tiempo en el eje X, eje Y, eje Z.

La figura 3.6 muestra los resultados que fueron exportados y graficados en Matlab, con esta gráfica se pudo determinar los valores mínimos y máximos registrados para cada eje. Estos valores son comparados con su respectivo valor de referencia (ver Tabla 1).

Eje	Val min [cm/seg2]	Val max [cm/seg2]	Valor de ref [cm/seg2]
X	-135	79	0
Y	-237	167	-9
Z	150	338	230

Tabla 1: Valores máximos y mínimos registrados

Podemos apreciar en la Tabla 1, que los valores máximos y mínimos de los tres ejes variaron con respecto a su valor de referencia al aplicar una fuerza externa sobre el bloque. El resultado de esta perturbación obtenido en la consola de Nios II se muestra en la figura 3.7.

```

aceleromete Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instanc
Posible onda P detectada
Max pico en X:-54 Max pico en Y:-1 Max pico en Z:230 tm: 0 [ms]
Esperando onda S...
Nuevo valor pico encontrado en el eje Y: -41 tm: 4 [ms]
Nuevo valor pico encontrado en el eje Y: 121 tm: 9 [ms]
Nuevo valor pico encontrado en el eje X: -68 tm: 14 [ms]
Nuevo valor pico encontrado en el eje Y: -256 tm: 931 [ms]
Nuevo valor pico encontrado en el eje X: 105 tm: 961 [ms]
Nuevo valor pico encontrado en el eje X: 187 tm: 966 [ms]
Onda S no registrada, no se trata de un evento sismico
Posible onda P detectada
Max pico en X:-79 Max pico en Y:21 Max pico en Z:234 tm: 0 [ms]
Esperando onda S...
Nuevo valor pico encontrado en el eje X: 113 tm: 4 [ms]
Nuevo valor pico encontrado en el eje Y: -58 tm: 4 [ms]
Valor pico de la onda S encontrado en el eje Y: -298 tm: 2397 [ms]
Valor pico de la onda S encontrado en el eje X: 129 tm: 2397 [ms]
Esperando ondas L para confirmación de evento sismico...
Onda L no registrada, no se trata de un evento sismico

```

Figura 3.7. Registros de movimientos en la consola de NIOS II

En la figura 3.7 la consola de Nios II muestra que se detectó varias veces una posible onda P, pero nunca se recibió la llegada de la onda S. De igual forma hubo un caso exclusivo en donde se registró la llegada de las ondas P y S, sin embargo, las ondas L no se registraron. Todos los registros jamás confirmaron la llegada de una onda sísmica debido a que no coincidían con el patrón característico.

3.1 Escenario 2

Debido a los resultados obtenidos en las pruebas efectuadas en el escenario uno, se implementó una base de prueba sobre la cual se colocó la tarjeta DE0 nano para ser sometida esta ocasión a constantes movimientos provocados por un motor DC que hará vibrar la plataforma constantemente (véase figura 3.8). Así analizaremos el tipo de onda provocada por este movimiento y si el sistema es capaz de diferenciarlo de un movimiento sísmico.

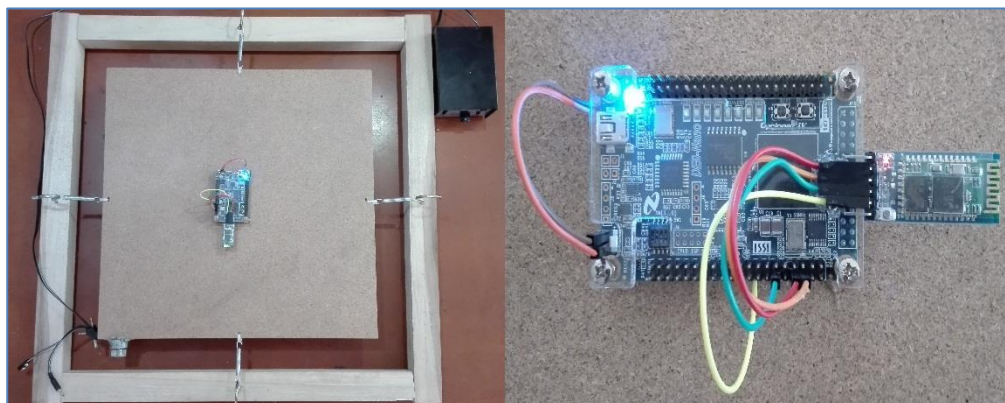


Figura 3.8. Base de prueba

La base de prueba que se muestra en la figura 3.8 está conformada por una superficie de madera sujeta en sus bordes por resortes que a su vez se encuentran enganchados a una superficie cuadrada de madera de 60 X 60 cm.

Para este escenario ya se cuenta con el sistema completo, es decir, a la tarjeta DE0 nano se le ha añadido un módulo Bluetooth el cual se encargará de enviar datos constantemente a una estación cercana la cual almacenará estos datos en caso de presentarse un evento sísmico (véase figura 3.9).

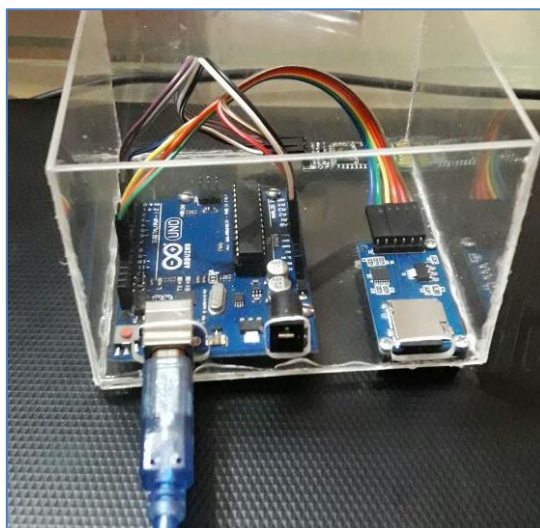


Figura 3.9. Base de datos

La base de datos que se muestra en la figura 3.9 está conformada por un Arduino UNO, un módulo microSD, un módulo Bluetooth y una tarjeta microSD de 2Gb, esta base podrá ser ubicada hasta en un rango de 10 metros lejos de la tarjeta DE0 nano. Los archivos almacenados en la tarjeta de memoria se muestran en la figura 3.10.

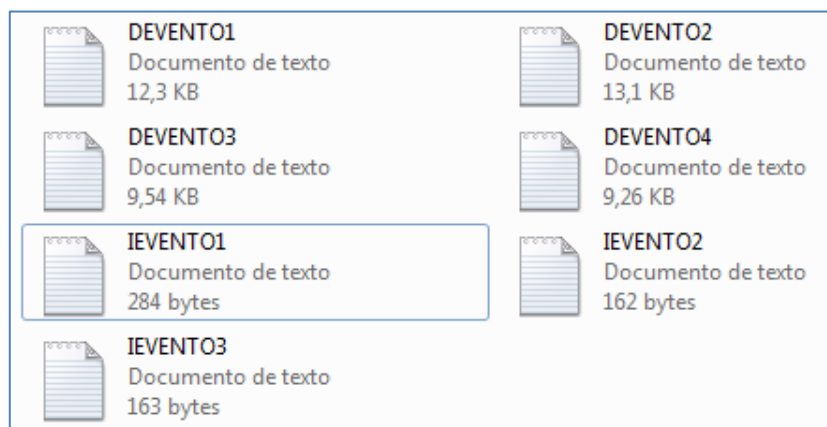


Figura 3.10. Archivos almacenados en la tarjeta micro SD

Los archivos almacenados mostrados en la figura 3.10 cuentan con el siguiente formato: “DEVENTO#” e “IEVENTO#”, donde los archivos que empiecen con la letra ‘D’ contienen los datos registrados por el acelerómetro, mientras que los archivos que empiezan con la letra ‘I’ contienen información referente al evento ocurrido como se muestra en la figura 3.11.

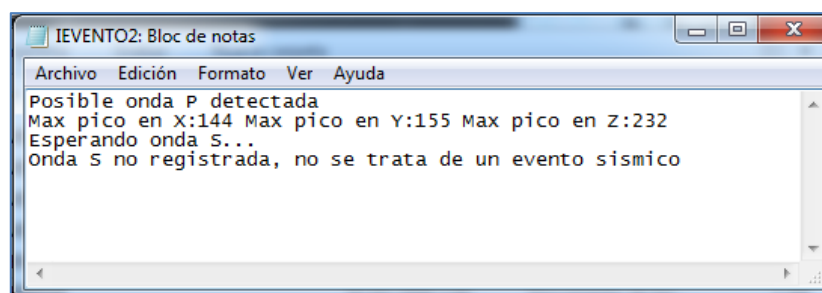


Figura 3.11. Información referente al evento registrado

El evento registrado en el archivo “IEVENTO2” que se muestra en la figura 3.11, inicialmente hace referencia a la posible detección de un evento sísmico al

registrar un valor para la onda P, luego queda a la espera de la llegada de la onda S la cual jamás llega, por lo cual se descarta la posibilidad de que se trate de un evento sísmico.

Esta información la podemos contrastar con los datos registrados en el archivo "DEVENTO1" que contiene todos los datos del acelerómetro, los cuales muestran una forma de onda como lo indica la figura 3.12.

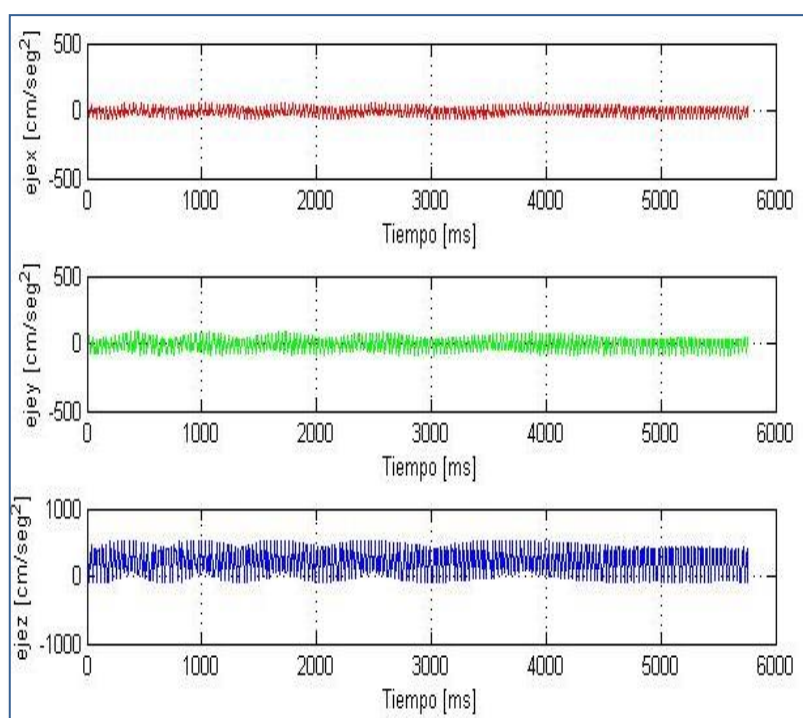


Figura 3.12. Gráficas de la aceleración registrada en los tres ejes vs tiempo.

Como podemos apreciar en la figura 3.12 la forma de onda registrada en los tres ejes no se asemeja en nada a las ondas producidas por un evento sísmico. Con esto confirmamos la información encontrada en el archivo de texto donde indicaba que no se trataba de un evento sísmico.

CONCLUSIONES Y RECOMENDACIONES

El sistema de reconocimiento de patrones de ondas sísmicas fue capaz de procesar datos en tiempo real y proveer una alerta temprana en un tiempo de 4 a 7 segundos.

El rango de detección de movimientos sísmicos es de 20 a 50 km, pudiendo mejorar esta distancia si se continúa investigando acerca de nuevos métodos de detección.

Las ondas sísmicas presentan un comportamiento poco convencional, de modo que el sistema pese a ser sometido a distintas pruebas ha logrado responder de la manera indicada.

Para encontrar una mejor aproximación de la distancia al epicentro, es necesario hallar una ecuación matemática para la curva características de las ondas S y P, y posterior a ello llevar esa ecuación a lenguaje C. Ya con esto se podría reducir el porcentaje de error que nos provee el método actual, y adicional a ello se podría ampliar el rango de detección de los movimientos sísmicos.

Las condiciones sobre las cuales debe ir la tarjeta DE0-nano no deben ser propensas a ruido de ningún tipo, y debe empernarse sobre las bases de una edificación de al menos 3 plantas. Esta edificación debe estar alejada de la ciudad donde se quiere enviar la alerta, considerando que mientras más alejada se encuentre de la ciudad más tiempo de aviso se tendrá.

Si se quiere ubicar la tarjeta DE0-nano sobre una casa o edificio para emitir la alerta, deberá asegurarse de realizar una modificación en el código para que el sistema se limite a esperar solamente las ondas S y P, evitando esperar las ondas L. Sin embargo al dejar la ubicación sobre una casa podría darse el caso de que la superficie sobre la cual se ubique la tarjeta se encuentre en constante movimiento y al no tener un análisis completo de la onda detectada es más propensa a detectar falsos positivos.

BIBLIOGRAFÍA

[1] Gustavo Daniel, (2017/09/24).detector sísmico con acelerómetro XYZ MMA7260Q[online]. Disponible en:<https://proyecto-cereza.wikispaces.com/file/view/Mediciones+del+acelerometro.pdf>

[2] Francesco Finazzi, (2015, 09,26).”SISMO DETECTOR” te avisa cuando viene un temblor [online]. Disponible en: <https://www.elsol.com.ar/nota/245772/ciencia-tecnologia/sismo-detector-te-avisa-cuando-viene-un-temblor.html>

[3] Marlon Sánchez, (2014/2015). Diseño de un sistema de estabilización usando sensores inteligentes [online]. Disponible en: <https://upcommons.upc.edu/bitstream/handle/2117/79655/PROYECTO%20FINAL%20DE%20CARRERA.pdf>

[4] ETSAC, (2017/0925). ¿Qué es un terremoto? [Online]. Disponible en: http://www.udc.es/dep/dtcon/estructuras/ETSAC/Investigacion/Terremotos/QUE_ES.htm

[5] José Manuel, (2017/09/25). ¿Qué es un FPGA? [Online]. Disponible en: <http://bibing.us.es/proyectos/abreproy/11375/fichero/MEMORIA%252FFPGAs.pdf>.

[6] Sitio oficial de altera, (2017/06/26). QUARTUS II [online]. Disponible en: https://es.wikipedia.org/wiki/Quartus_II.

[7] Altera, (2017/09/23). DE0-Nano User Manual [online]. Disponible en: <http://www.altera.com/>.

[8] ANALOG DEVICES, (2017/09/25) Digital Accelerometer ADXL345 [online].

Disponible en: <http://www.alldatasheet.com/view.jsp?Searchword=Adx1345>

[9] Arduino.cl, (2017/09/25) ¿Qué es arduino? [Online]. Disponible en:

<http://arduino.cl/que-es-arduino/>

[10] PROMETEC, (2017/09/24). Modulo Bluetooth HC-06 [online]. Disponible en:

<https://www.prometec.net/bt-hc06/#>

ANEXOS

Código principal en la tarjeta DE0-nano

```
#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <unistd.h>

int main(void){

    FILE *fp;

    char buffer [50];

    volatile char * Accelerometer = (volatile char *) 0x00009050;

    volatile int * Green LEDs = (volatile int *) 0x00009020;

    short esperar_onda_S=0,
onda_P_detectada=0,onda_S_detectadaX=0,onda_S_detectadaY=0,onda_L_de
tectadaY=0,onda_L_detectadaX=0,esperar_onda_L=0;

    int
temp,valor_ruptura=40,retornar=0,cnt_picosL_ejeX=0,cnt_picosL_ejeY=0
;

    int
Val_pico_Px=0,Val_pico_Sx=0,Val_pico_Lx=0,Val_pico_Py=0,Val_pico_Sy=
0,Val_pico_Ly=0;

    int x_axis,
y_axis,z_axis,tiempo_muestreo=0,tiempo_onda_S=0,tiempo_onda_P=0;;

    int cnt=0,dir=0,alm temp;

    char caracter;
```

```

while(1)
{
//Deteccion de la onda P
    while(onda_P_detectada==0){

        usleep(50000); //tasa de muestreo 10 ms

//Obtención del valor de aceleracion en el eje 'x'

//Select and read the lower-order byte of the x-axis value
        *(Accelerometer) = 0x32;
        x_axis = 0x000000FF & *(Accelerometer+1);

//Select and read the higher-order byte of the x-axis
value
        *(Accelerometer) = 0x33;
        temp = 0xFF & *(Accelerometer+1);

//Combine the two x-axis bytes, and sign-extend it
        x_axis += temp << 8;

        if (x_axis & 0x008000)
            x_axis |= 0xFFFF0000;
    }
}

```

```

//obtención del valor de aceleración en el eje 'y'

//Select and read the lower-order byte of the x-axis
value
    *(Accelerometer) = 0x34;
    y_axis = 0x000000FF & *(Accelerometer+1);

//Select and read the higher-order byte of the x-axis
value
    *(Accelerometer) = 0x35;
    temp = 0xFF & *(Accelerometer+1);
    //Combine the two x-axis bytes, and sign-extend it
    y_axis += temp << 8;
    if (y_axis & 0x008000)
        y_axis |= 0xFFFF0000;

//obtención del valor aceleración en el eje 'z'

//Select and read the lower-order byte of the z-axis
value
    *(Accelerometer) = 0x36;
    z_axis = 0x000000FF & *(Accelerometer+1);

//Select and read the higher-order byte of the z-axis
value
    *(Accelerometer) = 0x37;
    temp = 0xFF & *(Accelerometer+1);
    //Combine the two z-axis bytes, and sign-extend it

```

```

z_axis += temp << 8;

if (z_axis & 0x008000)
    z_axis |= 0xFFFF0000;

tiempo_muestreo++;

if (abs(x_axis)>valor_ruptura ||
abs(y_axis)>valor_ruptura){
    onda_P_detectada=1;
    esperar_onda_S=1;
    Val_pico_Px=x_axis;
    Val_pico_Py=y_axis;
    tiempo_onda_P=tiempo_muestreo;
    printf("Posible onda P detectada");
    printf("Max pico en X:%d Max pico en Y:%d
Max pico en Z:%d tm: 0 [ms]\n",Val_pico_Px,Val_pico_Py,z_axis);
    printf("Esperando onda S...\n");
}

if(tiempo_muestreo==99 && onda_P_detectada==0){
    tiempo_muestreo=0;
    dir=0;
}

```

```

        sprintf (buffer, "%d %d %d",x_axis,y_axis,z_axis);
        fp = fopen ("/dev/uart","r+");
        fprintf(fp,buffer);
        fclose(fp);

    }

//Deteccion de onda S

        while((onda_S_detectadaX==0 || onda_S_detectadaX==0)
&& esperar_onda_S==1 ){

                usleep(1000); //tasa de muestreo 1 ms
                tiempo_muestreo++;

                //Obtención del valor de aceleración en el eje
                'x'

                //Select and read the lower-order byte of the
                x-axis value

                *(Accelerometer) = 0x32;
                x_axis = 0x000000FF &
                *(Accelerometer+1);

                //Select and read the higher-order byte of
                the x-axis value

                *(Accelerometer) = 0x33;

```

```

        temp = 0xFF & *(Accelerometer+1);
//Combine the two x-axis bytes, and sign-
extend it

        x_axis += temp << 8;
        if (x_axis & 0x008000)
            x_axis |= 0xFFFF0000;

//obtención del valor de aceleración en el
eje 'y'

//Select and read the lower-order byte of
the x-axis value

        *(Accelerometer) = 0x34;
        y_axis = 0x000000FF &
*(Accelerometer+1);

//Select and read the higher-order byte of
the x-axis value

        *(Accelerometer) = 0x35;
        temp = 0xFF & *(Accelerometer+1);
//Combine the two x-axis bytes, and
sign-extend it

        y_axis += temp << 8;
        if (y_axis & 0x008000)
            y_axis |= 0xFFFF0000;

```

```

//obtención del valor aceleracion en el eje
'z'

//Select and read the lower-order byte of
the z-axis value

*(Accelerometer) = 0x36;
z_axis = 0x000000FF &
*(Accelerometer+1);

//Select and read the higher-order byte of
the z-axis value

*(Accelerometer) = 0x37;
temp = 0xFF & *(Accelerometer+1);

//Combine the two z-axis bytes, and sign-
extend it

z_axis += temp << 8;
if (z_axis & 0x008000)
    z_axis |= 0xFFFF0000;

if (tiempo_muestreo < (tiempo_onda_P + 1000)) {
    if (abs(x_axis) > abs(Val_pico_Px)) {
        Val_pico_Px = x_axis;
        printf("Nuevo valor pico
encontrado en el eje X: %d tm: %d
[ms]\n", Val_pico_Px, tiempo_muestreo);
    }
}

```



```

    }

    if(abs(y_axis)>abs(Val_pico_Py)){
        Val_pico_Py=y_axis;
        printf("Nuevo valor pico
encontrado en el eje Y: %d tm: %d
[ms]\n",Val_pico_Py,tiempo_muestreo);
    }
}

if(tiempo_muestreo>(tiempo_onda_P+2300) &&
tiempo_muestreo<(tiempo_onda_P+5750) ){
    if ((abs(x_axis)>abs(2*Val_pico_Px)) &&
abs(Val_pico_Sx)==0 ){
        onda_S_detectadaX=1;
        if
(abs(x_axis)>abs(Val_pico_Sx)){
            Val_pico_Sx=x_axis;
        }
        printf("Valor pico de la onda S
encontrado en el eje Y: %d tm: %d
[ms]\n",Val_pico_Sx,tiempo_muestreo);
    }
    if (abs(y_axis)>abs(2*Val_pico_Py) &&
abs(Val_pico_Sy)==0){
        onda_S_detectadaY=1;

```

```

        if
(abs(y_axis)>abs(Val_pico_Sy)){
            Val_pico_Sy=y_axis;
        }
        printf("Valor pico de la onda S
encontrado en el eje X: %d tm: %d
[ms]\n",Val_pico_Sy,tiempo_muestreo);
    }
}

    if (tiempo_muestreo==(tiempo_onda_P+5750)){
        esperar_onda_S=0;
    }

}

//Verificar que la onda S ha sido detectada
    if (onda_S_detectadaX==1 && onda_S_detectadaY==1 &&
esperar_onda_S==1){
        esperar_onda_L=1;
        tiempo_onda_S=tiempo_muestreo;
    }
    else{
        retornar=1;
    }
}

```

```

    if (retornar==1){
        printf("Onda S no registrada, no se trata de un
evento sísmico\n");
        tiempo_onda_P=0;
        tiempo_muestreo=0;
        dir=0;
    }
    else{
        printf("Esperando ondas L para confirmación de
evento sísmico...\n");
        while(esperar_onda_L==1  && (onda_L_detectadaX==0
|| onda_L_detectadaY==0) ){

            usleep(1000); //tasa de muestreo 1 ms
            tiempo_muestreo++;
            //Obtención del valor de aceleracion en el
eje 'x'

            //Select and read the lower-order byte of
the x-axis value

            *(Accelerometer) = 0x32;
            x_axis = 0x000000FF &
*(Accelerometer+1);

```

```

//Select and read the higher-order byte of
the x-axis value

*(Accelerometer) = 0x33;
temp = 0xFF & *(Accelerometer+1);
//Combine the two x-axis bytes, and sign-
extend it

x_axis += temp << 8;
if (x_axis & 0x008000)
    x_axis |= 0xFFFF0000;

//obtención del valor de aceleración en el
eje 'y'

//Select and read the lower-order byte of
the x-axis value

*(Accelerometer) = 0x34;
y_axis = 0x000000FF &
*(Accelerometer+1);

//Select and read the higher-order byte of
the x-axis value

*(Accelerometer) = 0x35;
temp = 0xFF & *(Accelerometer+1);
//Combine the two x-axis bytes, and
sign-extend it

```

```

        y_axis += temp << 8;
        if (y_axis & 0x008000)
            y_axis |= 0xFFFF0000;
//obtención del valor aceleracion en el eje
'z'

//Select and read the lower-order byte of
the z-axis value

        *(Accelerometer) = 0x36;
        z_axis = 0x000000FF &
*(Accelerometer+1);

//Select and read the higher-order byte of
the z-axis value

        *(Accelerometer) = 0x37;
        temp = 0xFF & *(Accelerometer+1);
//Combine the two z-axis bytes, and sign-
extend it

        z_axis += temp << 8;
        if (z_axis & 0x008000)
            z_axis |= 0xFFFF0000;

        if
(abs(x_axis)>=abs(Val_pico_Sx*100/30)){ //encontrar un pico mas alto
para el cual el pico de la onda S represente un 30%

            if
(abs(x_axis)>abs(Val_pico_Lx)){

                Val_pico_Lx=x_axis;

```

```

    }
    cnt_picosL_ejeY++;
    printf("Valor pico de la onda L
encontrado en el eje X: %d tm: %d
[ms]\n",Val_pico_Lx,tiempo_muestreo);
    }
    if
(abs(y_axis)>abs(Val_pico_Sy*100/30)){

        if
(abs(y_axis)>abs(Val_pico_Ly)){
            Val_pico_Ly=y_axis;
        }
        cnt_picosL_ejeY++;
        printf("Valor pico de la onda L
encontrado en el eje Y: %d tm: %d
[ms]\n",Val_pico_Sy,tiempo_muestreo);
    }
    if (tiempo_muestreo>tiempo_onda_S+2000){
        esperar_onda_L=0;
    }

    if(cnt_picosL_ejeX>=3){
        onda_L_detectadaX=1;
    }
}

```

```

        if(cnt_picosL_ejeY>=3){
            onda_L_detectadaY=1;
        }
    }

    if(onda_L_detectadaX==1 && onda_L_detectadaY==1 &&
esperar_onda_L==1){

        printf("evento sismico confirmado\n");
        printf("Actividad sismica registrada a
aproximadamente: %f [km]",(tiempo_onda_S/1000)*8.7);
    }

    else{

        printf("Onda L no registrada, no se trata de
un evento sismico\n");
    }

}

esperar_onda_S=0; onda_P_detectada=0;
onda_S_detectadaX=0;onda_S_detectadaY=0;onda_L_detectadaY=0;onda_L_d
etectadaX=0;esperar_onda_L=0;

temp=0;retornar=0;cnt_picosL_ejeX=0;cnt_picosL_ejeY=0;

Val_pico_Px=0;Val_pico_Sx=0;Val_pico_Lx=0;Val_pico_Py=0;Val_pi
co_Sy=0;Val_pico_Ly=0;

tiempo_muestreo=0;tiempo_onda_S=0;

```

```

    }
    return 0;
}

```

Código principal en Arduino UNO

```

#include <SD.h>

#include <SoftwareSerial.h>

SoftwareSerial Fpga_dato(6,7); // RX, TX recorder que se cruzan

int dato=0; //variable que recibe los datos por comunicacion serial
int cont=0;

int tiempo_ant=0,tiempo_sig=0;

File myFile;

unsigned short bandera=1;

int cont_event=1;

void setup()
{
    Serial.begin(9600);

    Fpga_dato.begin(9600);

    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }

    Serial.println("configuracion!");

    Serial.print("Iniciando SD ...");

    if (!SD.begin(4)) {

```



```
    Serial.println("No se pudo inicializar");
    return;
}
Serial.println("inicializacion exitosa");
}
void loop()
{
    char etiqueta[40]="";
    bool fin_evento = false;
    bool inicio = false ;
    char cadena[5]=".txt";
    while(1){

        if (Fpga_dato.available()){
            char c=Fpga_dato.read();
            if(c=='I'){
                Serial.print(c);
                inicio = true;
                while(inicio==true){
                    if (Fpga_dato.available()){
                        if(bandera==1){
                            sprintf(etiqueta,"devento%d%s", cont_event,cadena);

                            myFile = SD.open(etiqueta, FILE_WRITE);//abrimos el archivo
                            bandera=2;
                        }
                    }
                }
            }
        }
    }
}
```

```
}  
char d=Fpga_dato.read();  
if(bandera==2&& myFile){  
    if (d==';'){  
        Serial.print("\n");  
        myFile.println(" ");  
    }  
    else if(d=='D'){  
        Serial.println(d);  
        myFile.close();  
        SD.remove(etiqueta);  
        bandera=1;  
        inicio=false;  
    }  
    else if(d=='F'){  
        Serial.println(d);  
        myFile.close();  
        inicio=false;  
        fin_evento = true;  
        bandera=1;  
    }  
    else{  
        Serial.print(d);  
        myFile.print(d);  
    }  
}
```

```
    }  
  }  
}  
if(fin_evento==true){  
  while(fin_evento==true){  
  
    if (Fpga_dato.available()){  
      if(bandera==1){  
        sprintf(etiqueta,"evento%d%s", cont_event,cadena);  
  
        myFile = SD.open(etiqueta, FILE_WRITE);//abrimos el archivo  
        bandera=2;  
      }  
      char e=Fpga_dato.read();  
      if(bandera==2 && myFile){  
        if (e==';'){  
          Serial.println("\n");  
          myFile.println(" ");  
        }  
        else if(e=='F'){  
          Serial.println(e);  
          myFile.close();  
          fin_evento=false;  
          bandera=1;  
          cont_event++;  
        }  
      }  
    }  
  }  
}
```

```
    }  
    else{  
        Serial.print(e);  
        myFile.print(e);  
    }  
}  
}  
}  
}  
}  
}  
}  
}
```