

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE UN PROTOTIPO DE
SISTEMA DE ALERTA POR
DETECCIÓN DE EVENTOS SÍSMICOS**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO SUPERIOR EN REDES Y TELECOMUNICACIONES**

HUGO FERNANDO CAÑAVERAL VELÁSQUEZ

DIRECTOR: FANNY PAULINA FLORES ESTÉVEZ

DMQ, febrero 2022

CERTIFICACIONES

Yo, HUGO FERNANDO CAÑAVERAL VELÁSQUEZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



HUGO FERNANDO CAÑAVERAL VELÁSQUEZ

hugo.canaveral@epn.edu.ec

unikhugo@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por HUGO FERNANDO CAÑAVERAL VELÁSQUEZ, bajo mi supervisión.

FANNY PAULINA FLORES ESTÉVEZ

DIRECTORA

fanny.flores@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



HUGO FERNANDO CAÑAVERAL VELÁSQUEZ

DEDICATORIA

Este proyecto de titulación está dedicado a:

A mis padres German y Magdalena quienes con su amor, esfuerzo, consejos y paciencia me acompañaron hasta este escalón de superación en mi vida, en toda la trayectoria me inculcaron lo que son buenos valores, a no rendirme, ser perseverante y levantarme cuando me he tropezado. Son el pilar de mi vida.

A mis hermanos Christian y Cesar que pese a la distancia no me dejaron solo, ellos son mi inspiración de seguir creciendo como persona y profesional. Con su apoyo llegaré a conseguir todo lo que me proponga.

A mis amigos de carrera con quienes compartimos momentos de tensión, ayuda y sobre todo las desveladas de los deberes, estudio y sobre todo proyectos.

Hugo

AGRADECIMIENTO

Mi profundo agradecimiento a la Escuela Politécnica Nacional y a la Escuela de Formación de Tecnólogos, por haberme permitido ser parte de la mejor Universidad del país, darme la oportunidad de conocer y aprender de todas sus instalaciones, sobre conocer maestros que son dignos de admirar por sus conocimientos.

A la Ing. Fanny Flores quien fue mi maestra en la mayoría de materias profesionales, por colaborar en el proyecto de titulación. Sus conocimientos y profesionalidad son de admirar.

A los maestros, quienes a nivel profesional y personal me ayudaron a mi formación académica en toda mi carrera, son una motivación para seguir superándome.

Hugo

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
RESUMEN.....	VI
ABSTRACT	VII
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance.....	2
1.4 Marco teórico	3
2. METODOLOGÍA	6
3. RESULTADOS.....	8
3.1 Establecimiento de requerimientos del prototipo	9
3.2 Definición de componentes de hardware	9
3.3 Diseño del prototipo	12
3.4 Implementación del prototipo	21
3.5 Pruebas de funcionamiento.....	30
4. CONCLUSIONES	34
5. RECOMENDACIONES	35
6. REFERENCIAS BIBLIOGRÁFICAS.....	36
7. ANEXOS.....	i
ANEXO I.....	ii
ANEXO II.....	iii
ANEXO III.....	iv
ANEXO IV.....	viii

RESUMEN

El presente proyecto de titulación, “Implementación de un prototipo de sistema de alerta por detección de eventos sísmicos”, está compuesto por un Arduino Uno, un sensor SW-420, una tarjeta SIM800L y un regulador *Step-Down* XL4025 XL77; estos componentes electrónicos están integrados dentro de una caja de acrílico. La finalidad del sistema es advertir a los usuarios sobre la ocurrencia de un evento sísmico, mediante el envío de un mensaje SMS a su teléfono celular.

Este documento tiene seis secciones. En la primera parte se presenta la problemática que consiste en la poca o nula percepción que tienen las personas cuando existe un movimiento telúrico, también se explica el funcionamiento general del prototipo. Asimismo, se expone los objetivos, el alcance y los fundamentos más relevantes con respecto al *hardware* implementado. La segunda sección contiene la metodología empleada para el cumplimiento de cada objetivo.

En el tercer apartado denominado resultados, se especifica el proceso realizado que comenzó con la determinación de los requisitos; después, se evidencia la investigación y adquisición de los equipos a utilizar. Además, se incluye el diseño del sistema, los cálculos para las resistencias y las especificaciones de funcionamiento del sensor y regulador; asimismo, en la implementación se aborda la creación de la placa, el sistema de alimentación, la programación y la integración de todos los componentes en el chasis. Finalmente, se realiza las pruebas de funcionamiento y se añade videos tanto de la comprobación de operatividad del prototipo como del manual de uso.

En la cuarta y quinta sección, conclusiones y recomendaciones, se sintetizó los procesos realizados, los cuales fueron la identificación del problema, el establecimiento de requerimientos, la investigación de elementos del *hardware*, el diseño e implementación del prototipo y por último la verificación de funcionamiento. Además, en los anexos se presenta los certificados de Turnitin y de funcionamiento; asimismo, el código de programación.

PALABRAS CLAVE: Sismo, vibración, tarjeta SIM800L, Arduino Uno, sensor SW-420, placa de circuito impreso.

ABSTRACT

The present degree project, "Implementation of an alert system prototype for seismic event detection", is composed of an Arduino Uno, a SW-420 sensor, a SIM800L card and a Step-Down regulator XL4025 XL77; these electronic components are integrated within an acrylic box. The purpose of the system is to warn users about the occurrence of a seismic event, by sending an SMS message to their cell phones.

This document has six sections. The first part presents the problem that consists in the little or no perception that people have when there is a telluric movement; the general functioning of the prototype is also explained. In addition, the objectives, the scope and the most relevant fundamentals with respect to the implemented hardware are presented. The second section contains the methodology used to achieve each objective.

In the third section called results, the process carried out that began with the determination of the requirements is specified, then the investigation and acquisition of the equipment to be used is evidenced. In addition, the system design, calculations for the resistors and the operating specifications of the sensor and regulator are included; likewise, the implementation addresses the creation of the board, the power system, the programming and the integration of all the components in the chassis. Finally, the operation tests were carried out and videos of both the prototype's operability check and the user manual were added.

In the fourth and fifth sections, conclusions and recommendations, the processes carried out were synthesized, which were identification of the problem, establishment of requirements, research of hardware elements, design and implementation of the prototype and finally the verification of operation. In addition, in the annexes are presented the Turnitin and operating certificates and the programming code.

KEYWORDS: *Quake, vibration, SIM800L card, Arduino Uno, SW-420 sensor, printed circuit board.*

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El prototipo consiste en la generación de alertas tras la detección de vibraciones sísmicas producidas por la liberación de energía concentrada entre el manto y la corteza de la Tierra. La creación de este prototipo tiene como finalidad disminuir el estado de pánico durante el desarrollo de un evento telúrico y brindar la oportunidad de que los afectados puedan tomar las respectivas medidas de autoprotección. Gracias a que el prototipo cuenta con autonomía de la red eléctrica, y que su chasis es resistente al ambiente, puede ser instalado en cualquier lugar en el que sea necesario.

El funcionamiento general del prototipo se muestra en la Figura 1.1; como se puede observar, el primer elemento es un módulo sensor de vibración SW-420, el cual integra un potenciómetro que calibra el umbral de sensibilidad, un comparador de voltaje LM393 y dos Diodos Emisores de Luz (LED), uno para el encendido/apagado y otro para el estado (1L o 0L). Además, cuenta con 3 pines: Vcc para la alimentación, GND para la conexión a tierra y la salida digital (DO). Este último pin se conecta a la placa Arduino UNO, donde tras la detección de una vibración, el microcontrolador genera una alerta que posteriormente es enviada al módulo SIM800L [1].

El módulo SIM800L funciona con un Módulo de Identidad del Suscriptor (SIM), que mínimo debe manejar tecnología de Segunda Generación (2G), y una antena externa que debe ir conectada al pin de red del mismo módulo. Cuando el sensor detecta una vibración, envía una señal que pasa por el Arduino y llega al SIM800L. Después de recibir el aviso, el módulo genera un mensaje mediante el Servicio de Mensaje Corto (SMS) que, gracias a la señal emitida por la antena, llegará al celular del usuario, informándole sobre la ocurrencia del acontecimiento. Además, durante la percepción del sismo, el microcontrolador del Arduino, envía dos señales de alerta adicionales: el encendido de LEDs de colores y la activación de una bocina, ambos dispositivos ubicados en el chasis [2].

Todos los elementos antes mencionados conforman al prototipo contenido en un chasis, el cual lo protegerá de factores climáticos y de la manipulación de terceros; asimismo, brindará estabilidad a la detección de vibraciones. Cabe recalcar que el dispositivo también funciona con una batería.

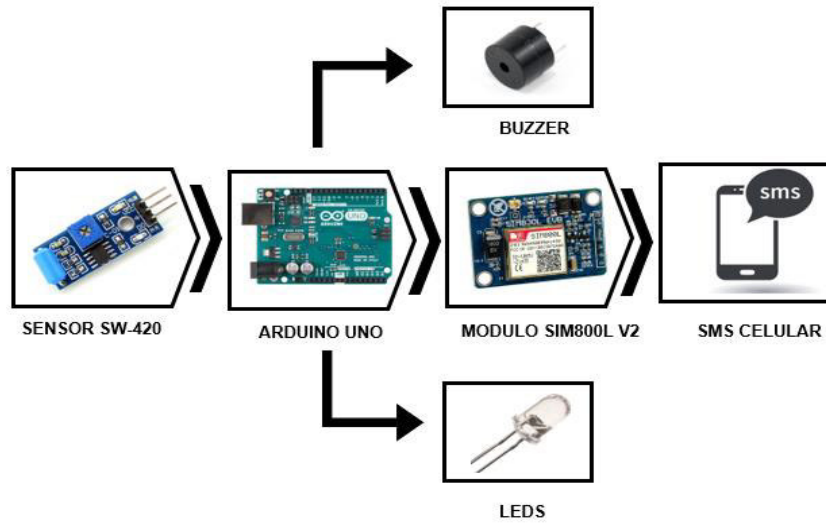


Figura 1.1 Funcionamiento general del prototipo [1] [3] [4]

1.1 Objetivo general

Implementar un prototipo de sistema de alerta por detección de eventos sísmicos.

1.2 Objetivos específicos

1. Establecer los requerimientos del prototipo.
2. Investigar los componentes de *hardware*.
3. Diseñar el prototipo.
4. Implementar el prototipo.
5. Realizar pruebas de funcionamiento.

1.3 Alcance

El análisis de requerimientos incluye una investigación sobre los dispositivos utilizados, lo que facilita el diseño del sistema. Por otra parte, el diseño abarca una vista en 3D que se desarrolló en un *software* de fácil uso y con las herramientas necesarias para el funcionamiento del prototipo, lo que incluye cálculos que justifican el armado del circuito electrónico, así como el diagrama de flujo correspondiente.

La implementación contempla la fabricación de una placa electrónica y un chasis; este último elemento garantiza una mayor estabilidad al momento de la detección de vibraciones

y provee mayor estética al prototipo. En las pruebas realizadas al sistema de alerta de la presencia de eventos sísmicos, se verificó el envío de un SMS al teléfono móvil del usuario, alertándolo para que este a su vez proceda a evacuar del lugar. Finalmente, se desarrollaron dos videos, uno en el que se demuestra que el sistema es totalmente funcional y el otro corresponde al manual de usuario del sistema.

1.4 Marco teórico

Eventos sísmicos

Los eventos telúricos o sismos son vibraciones percibidas en la superficie de la Tierra; la intensidad y duración dependen de la liberación de cierta cantidad de energía producida durante el desarrollo de una falla sísmica, erupción volcánica, movimiento de las placas tectónicas, impacto de meteoritos, minería o pruebas nucleares [5] [6].

Los sismos, temblores y terremotos son sinónimos diferenciados por la gravedad de los daños que generan; por el contrario, las réplicas son acontecimientos de menor magnitud producidas en las cercanías del epicentro tras la finalización de la proyección del evento sucedido en el hipocentro [5].

La variación impredecible del foco epicentro y falla del sismo generan una infinidad de efectos sobre la geografía del lugar; así como también, en las personas. Entre las características más comunes de este tipo de eventos se encuentran la intensidad, magnitud, duración, cercanía o lejanía del epicentro y la hora en la que se desarrolló el movimiento sísmico. Algunas de las consecuencias son movimientos leves o bruscos de la superficie terrestre, incendios, deslizamientos de tierra o maremotos con tsunamis, personas fallecidas, daños infraestructurales; y con respecto a la sociedad, lo más común es que se produzcan secuelas, tales como *shock* psicológico, pánico, traumas o Trastorno de Estrés Postraumático (TEPT) [7].

El “Informe de Situación N°65 – 16/05/2016” de la Secretaría de Gestión de Riegos, presentó información acerca del terremoto de 7.8 (°) en la escala de *Richter* producido el 16 de abril del año 2016 en el cantón de Pedernales; se reveló que 663 personas perdieron la vida, 6 274 persona resultaron heridas, varias viviendas y edificios colapsaron, y que incluso dentro de las edificaciones permanecieron personas atrapadas por varios días [8].

En Ecuador, la mayoría de los sismos no son percibidos por los habitantes, como lo expresa el medio de información “El Comercio”; por ejemplo, en la Amazonía en abril del 2020 se

registraron 24 movimientos telúricos y los habitantes no supieron de los mismos, sino hasta cuando se enteraron por los medios de comunicación o a través de familiares. A esto, el Instituto Geofísico de la Escuela Politécnica Nacional (IGEPN), lo denomina enjambre sísmico, que es la ocurrencia de varios sismos con magnitudes similares durante un tiempo corto [9].

La percepción humana con respecto a los sismos es variable; debido a que el progreso de estos eventos inesperados depende no solo de la ubicación y posición de la persona sino también de las características propias del evento en sí. Por ejemplo, estando de pie el sentir del movimiento generado por el temblor es menor en comparación al estar acostado o sentado, esto debido a que se tiene más contacto con la superficie; asimismo, si la magnitud del sismo es leve, moderada o grave [9] [10].

Como presenta el IGEPN, la mayoría de los últimos sismos se han presentado en la costa ecuatoriana; esta región peni sísmica presenta varios movimientos telúricos de baja magnitud, lo cual bordea los 3.6 (°) en la escala de *Richter*, esto se debe a que el país se encuentra en una zona de choque entre las placas Sudamericana y Nazca [5].

Arduino Uno

El desarrollo del prototipo requiere de dispositivos fundamentales como Arduino Uno que, por su microprocesador, se lo considera el corazón del sistema. Arduino pertenece a una gran comunidad, por lo cual se puede obtener una gran cantidad de información, así como también respuestas para solventar dudas. Otra de sus ventajas es el lenguaje de programación, el cual es comprensible fácilmente; además, tiene un bajo costo. En la Figura 1.2 se presenta los componentes y pines con los cuales se puede trabajar en el módulo Arduino Uno [3].

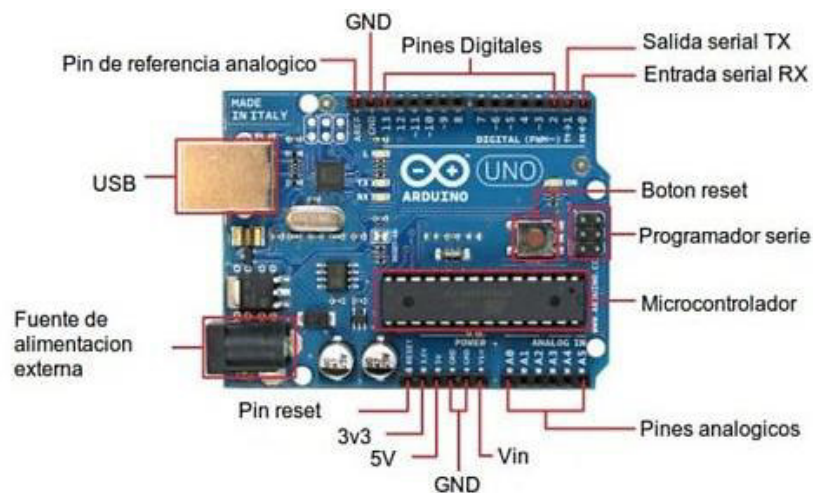


Figura 1.2 Componentes de Arduino Uno [11]

Sensor de vibración SW-420

El sensor SW-420 es mayormente utilizado como interruptor digital. Para el prototipo su uso está dedicado a recibir y procesar información; esto quiere decir, vibraciones, golpes o movimientos. El módulo del sensor tiene integrado un potenciómetro para el ajuste de la sensibilidad, un comparador de voltaje, LEDs encendido/apagado, un LED de estado y tres pines, los cuales se pueden observar en la Figura 1.3, donde se entiende que el primer pin es de alimentación, el segundo de conexión a tierra y el tercero es la salida digital [1].

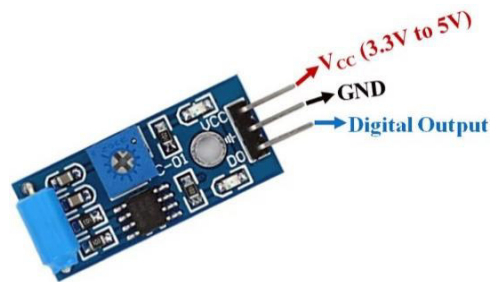


Figura 1.3 Pines del sensor de vibración SW-420 [1]

Módulo SIM800L

El proceso de la comunicación finaliza con el envío de alerta por SMS al celular del usuario mediante el SIM800L v2.0. Este dispositivo cuenta con una ranura para introducir una tarjeta SIM, pines de comunicación de Transmisor-Receptor Asíncrono Universal (UART), LEDs de encendido/apagado y estado; además de un conector para la antena. Gracias al uso de un *chip* que trabaja con mínimo 2G, es posible la conexión a cualquier red celular, lo cual incluye el envío y recepción de mensajes, a través de la banda cuádruple de frecuencia en la cual trabaja, siendo estas 850/900/1800/1900 (MHz). En la Figura 1.4 se presenta la distribución de pines [4].

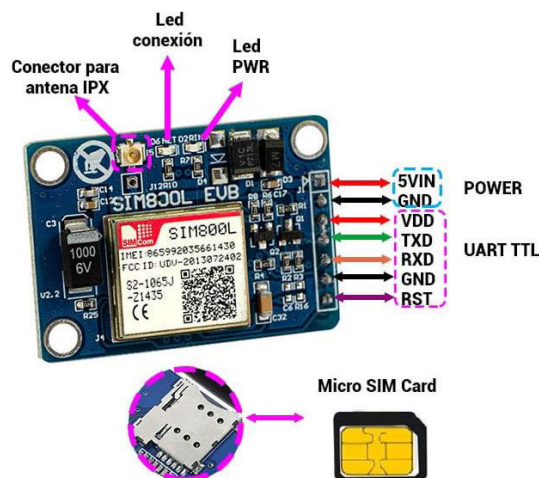


Figura 1.4 Componentes y pines del módulo SIM800L V2.0 [4]

2. METODOLOGÍA

La realización del presente proyecto se estableció en cinco etapas esenciales, en la Figura 2.1 se puede observar el proceso de desarrollo del sistema, que permitió dar cumplimiento a los objetivos específicos, lo que consecuentemente facilitó la implementación del prototipo.

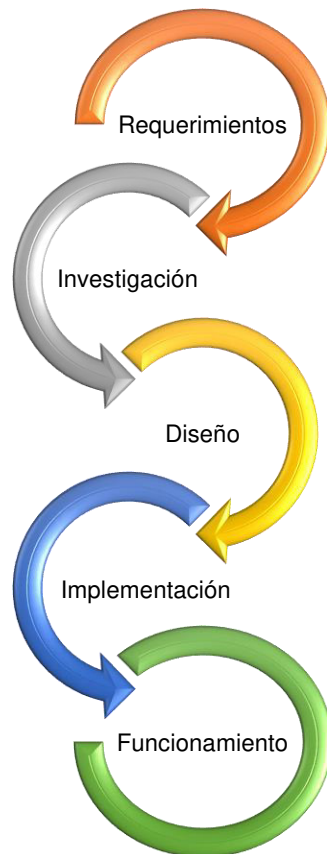


Figura 2.1 Metodología utilizada

Los requerimientos para la creación del prototipo fueron establecidos en base a un análisis previo, donde se definieron el uso y la finalidad del sistema; es decir, que se especificó que el proyecto será usado para detectar vibraciones sísmicas con el objetivo de alertar a los usuarios. También, se evaluaron las necesidades a ser cubiertas, entendiéndose que el usuario no debe tener ninguna dificultad al momento de utilizar el prototipo ni debe existir fallo en el envío de la alerta; es por ello que, se determinaron las dimensiones y funciones que realizan los elementos integrados en el proyecto. Asimismo, se dispuso que la alimentación sea independiente de la energía eléctrica y que el prototipo se comunique con el celular del usuario.

Al realizar la investigación de los posibles componentes electrónicos a ser usados y elaborar una comparativa entre sus características, se concluyó que en el prototipo se integraría un Arduino Uno por su lenguaje de programación comprensible, bajo costo y compatibilidad con el sensor SW-420, el cual se usó para detectar las distintas magnitudes de vibraciones. Además, se seleccionó el módulo SIM800L, el cual fue usado como enlace de comunicación entre el sistema y el celular del usuario, ya que al recibir la señal de alerta este módulo envía SMS. Además, con la finalidad de proteger los elementos, se utilizó el regulador de voltaje *Step-Down* XL4015; tras la elección, se efectuó la adquisición de los equipos.

En el diseño, se llevaron a cabo los siguientes procedimientos: primero el encapsulado de los componentes, luego la ubicación geométrica de los mismos y la creación del circuito en el *software Proteus*; es importante recalcar que dichas conexiones entre dispositivos no deben cruzarse, esto con el objetivo de evitar un corto-circuito. Además, durante el desarrollo de las líneas de conexión se incluyó el espacio generado por las borneras en las que se conectaron los *switches*; asimismo, se consideró el espacio del motor, el cual simula vibraciones. Una vez posicionado cada elemento, con ayuda del *software SketchUP* se realizó el diseño en 3D del prototipo junto con el chasis.

Adicionalmente, se realizaron cálculos para determinar las resistencias y los valores de funcionamiento del sensor SW-420. Por otra parte, se investigaron proyectos anteriores, con el fin de establecer los rangos de operación del sensor y definir un evento débil, moderado, fuerte y muy fuerte.

La implementación del prototipo inició con la impresión del diseño del circuito en una baquelita; para ello se usó papel couché, este material mejora la calidad del sublimado. Para completar la creación de la Placa de Circuito Impreso (PCB), se realizaron varios procesos: primero el planchado, luego el quemado del cobre con ácido férrico; posteriormente, la perforación de los puntos de conexión y por el último, se soldaron los componentes del sistema. La placa PCB fue instalada en una caja transparente de acrílico, la cual funciona como chasis del prototipo; además, cuenta con tres compartimientos: para la placa PCB, la batería y cable de conexión a la computadora. También se incluye ranuras para el *jack*, tarjeta micro SIM y agujeros de adaptación para la antena y para los *switches*.

Durante el desarrollo de la cuarta etapa se elaboró el código de programación para el funcionamiento del microprocesador integrado en el Arduino Uno, este proceso se efectuó en el Entorno de Desarrollo Integrado (IDE) de Arduino. Por otro lado, luego de la

adquisición de los equipos y de la culminación de la placa PCB, se inició el armado de la maqueta; es decir, todos los elementos colocados de forma organizada dentro del chasis.

Finalmente, se realizaron pruebas de funcionamiento; primero con la fuente de alimentación donde se verificó la operatividad del sistema con una batería de 9 (V), con conexión a la red eléctrica mediante un cargador y de igual manera con ambas fuentes. También, se concretó los ajustes finales de sensibilidad en el sensor, esto con la finalidad de verificar el cumplimiento de los rangos establecidos y asegurar un óptimo desempeño del prototipo.

3. RESULTADOS

El prototipo “Sistema de alerta por detección de eventos sísmicos” genera una serie de alarmas dentro y fuera del módulo, las cuales tienen como finalidad advertir sobre el desarrollo de un acontecimiento telúrico al usuario a través de un SMS. El funcionamiento de este modelo centra su operatividad en el envío constante de señales entre sus distintos elementos, los cuales son: un sensor SW-420, un Arduino Uno y una tarjeta SIM800L.

Las alarmas con las que cuenta el prototipo trabajan de la siguiente forma: primero el sensor tras detectar vibraciones manda una señal hacia el Arduino, el módulo al recibir el aviso, procesa la información censada mediante el uso del código de programación creado para este sistema. A continuación, se enciende uno de los cuatro LEDs integrados en el Arduino, dependiendo de la intensidad del movimiento, junto con la bocina; además, durante este proceso el Arduino se encarga de enviar otra señal hacia el SIM con la finalidad de que este remita un SMS al usuario. La notificación enviada al celular de los beneficiados incluye una alerta corta sobre el evento sísmico y el Localizador de Recursos Uniforme (URL) de la página oficial del IGEPN.

La creación de este prototipo se desarrolló en cinco etapas fundamentales: primero se establecieron los requerimientos tanto sociales como tecnológicos, luego se efectuó una investigación abordando los componentes de *hardware*. Cabe recalcar que durante el diseño del modelo se utilizaron diferentes *softwares* tanto para la programación como para la presentación en 3D. Por consiguiente, previamente a la implementación se realizó el diseño y una vez concluido este procedimiento se incorporaron todos los elementos en una caja de acrílico; finalmente, se llevó a cabo la verificación de funcionamiento del sistema.

3.1 Establecimiento de requerimientos del prototipo

El prototipo fue creado para detectar vibraciones sísmicas y alertar a los usuarios sobre la ocurrencia del acontecimiento; de igual manera, se proporciona información cualitativa del evento. La ubicación del sistema es determinante para su óptimo funcionamiento; por ello, con la finalidad de que los sensores tengan la mejor percepción posible de las vibraciones, es fundamental que la instalación se realice en un lugar estable; por ejemplo, empotrado en una pared.

En caso de que la edificación donde se vaya a instalar el prototipo tenga varias plantas, se sugiere que la instalación se realice en la planta más alta, ya que de esa forma se captará de mejor manera el movimiento sísmico. Otro punto a tomar en consideración, es la implementación del sistema en un ambiente externo, la intemperie; visto que, eso podría generar una serie de fallas en el prototipo, por ello se añadió un chasis que lo protege de factores climáticos y de la manipulación de terceros.

Al ser un sistema de alerta, del cual pudieran depender vidas humanas, el prototipo no debe depender de la energía eléctrica de la red pública ni de una red cableada. Por esa razón, se consideró un sistema autónomo de alimentación, el cual funciona mediante una batería extraíble; es decir que puede ser fácilmente reemplazada. Asimismo, el prototipo requiere de una amplia cobertura para alertar a una mayor cantidad de usuarios, por lo cual para solventar dicha necesidad se utilizó comunicación por envío de mensajes SMS.

3.2 Definición de componentes de *hardware*



La elección de la placa electrónica, se definió tras realizar una comparación entre tres módulos: Arduino Uno, Raspberry Pi y Arduino Mega; entendiéndose que funcionan con microprocesadores y son componentes electrónicos de dimensiones pequeñas que pueden ser utilizados para cumplir con un sin número de tareas. Sin embargo, la diferencia más notoria entre los dos primeros módulos, es que Arduino Uno es una tarjeta programable mediante el *software* Arduino IDE, mientras que Raspberry Pi es una microcomputadora que necesita de un sistema operativo (SO).

Debido a que, en el prototipo se utilizaron pocos elementos, las conexiones de entrada y salida (E/S) de Arduino Uno son suficientes, muy aparte de que no requiere de un SO para comunicarse con los demás elementos. Además, para la alimentación del módulo hay dos opciones: la conexión mediante Bus Universal Serial (USB) a una computadora o con

baterías, también incluye la función de suspendido al desconectar la fuente de alimentación [12].

Entre Arduino Uno y Arduino Mega, las diferencias radican en sus especificaciones técnicas más relevantes, es por ello que en la Tabla 3.1 se presenta una comparativa, que tiene como resultado la preferencia de Arduino Uno sobre la otra placa. Dicha elección se basa es los requerimientos mínimos de funcionamiento como cantidad de pines, el bajo uso de memoria y la facilidad de programación, debido a las librerías.

Tabla 3.1 Comparación entre Arduino Uno vs Arduino Mega [13]

Especificación	Arduino Uno	Arduino Mega
Dimensión de la placa (largo x ancho)	68.6 × 53.4 (mm)	101.52 × 53.3 (mm)
Memoria SRAM	2 (KB)	8 (KB)
E/S Digitales	14	54
Microcontrolador	ATmega328	ATmega2560
Salidas analógicas (PWM)	6	15
Entradas analógicas	6	16
Puerto serial	1	4
Imagen		

La selección del módulo que permite detectar el evento vibratorio resulta imprescindible. En la Tabla 3.2 se observa la comparativa de los sensores de vibración más utilizados en conjunto con una placa electrónica Arduino; el funcionamiento de los dos es similar ya que, al detectar vibración o un golpe fuerte, envían una señal de salida al microcontrolador. La principal diferencia y, es por la cual se decidió utilizar el módulo SW-420, es que tiene integrado un potenciómetro con el cual se puede regular la sensibilidad de detección; además, incluye un comparador de voltaje.

Adicionalmente, en varios proyectos y demostraciones se constató que la posición del sensor KY-002 debe ser vertical; por lo contrario, el SW-420 se puede colocar en diferentes posiciones, otorgando mayor flexibilidad al sistema [1] [14].



Tabla 3.2 Comparación entre el módulo SW-420 vs KY-002 [1] [14]

Especificación	Módulo SW-420	Módulo KY-002
Voltaje de trabajo	3.3 a 5 (V)	3.3 a 5 (V)
Corriente	15 (mA)	5 (mA)
Sensor	SW-18015P	SW-420
Sensibilidad ajustable	Sí	No
Posición del sensor	Todos los ángulos	Un solo ángulo
Imagen		

La selección entre el módulo SIM800L y el módulo SIM900A, se estableció en base a la funcionalidad del mismo, considerando que ambos representan alternativas adecuadas para el envío de SMS; sin embargo, la prioridad antes mencionada es reducir el espacio sin perder la operatividad. Teniendo dicho contexto, el SIM800L cumple tanto con las dimensiones como con la eficiencia; además, es un dispositivo recientemente lanzado al mercado, a diferencia del SIM900A que pronto quedará obsoleto [15].


Debido a la modernidad del SIM800L, se puede tener acceso sea a la versión 1.0 o a la 2.0; en la Tabla 3.3 se evidencia las diferencias entre estos modelos; no obstante, en el prototipo se implementó el SIM800L v2.0 porque funciona con una fuente de alimentación de 5 (V), la cual es compatible con el resto de elementos. Otro factor que influyó es la ubicación de los pines, ya que facilita la interconexión de los componentes sin la necesidad de espadines; adicionalmente, sin dejar la estética de lado, se consideró que la antena es más llamativa [15].

Tabla 3.3 SIM800L v1.0 vs SIM80L v2.0 [4] [16]

Especificación	Módulo SIM800L v2.0	Módulo SIM800L v1.0
Dimensiones	40 x 28 (mm)	25 x 23 (mm)
Voltaje de trabajo	5 (V)	3.4 (V) – 4.4 (V)
Corriente	1 (A)	500 (mA)
Tamaño de la SIM	Micro SIM	Micro SIM
Banda	850/900/1800 / 1900 (MHz)	850/900/1800 / 1900 (MHz)
Pines	7	12
Imagen		

La protección de los componentes electrónicos, es uno de los puntos más importantes, por lo cual tras la debida investigación se estableció que el regulador de voltaje *Step-Down* XL4015 posee la funcionalidad de generar un voltaje inferior que el de la entrada; además, permite la regulación del mismo mediante un potenciómetro. Las especificaciones del módulo se presentan en la Tabla 3.4 [17].

Tabla 3.4 Especificaciones del regulador *Step-Down* XL4015 [17]

Especificación	Regulador <i>Step-Down</i> XL4015
Dimensiones	54 x 23 (mm)
Voltaje de entrada	4 (V) – 38 (V)
Voltaje de salida	1.25 (V) – 36 (V)
Corriente de salida	5 (A) máximo
Disipador de corriente	Si >3.5 (A)
Imagen	

3.3 Diseño del prototipo

Circuito electrónico

El desarrollo del circuito fue realizado en el *software Proteus*; debido a que posibilita la simulación con componentes electrónicos; además, de permitir la creación con el *Software* de Edición y Ruteo Avanzado (ARES) de pistas de conexión. El primer paso realizado fue el encapsulado de los componentes, considerando las medidas de cada elemento integrado en el prototipo, luego los componentes fueron situados de forma espacial; es decir que al momento de generar las pistas en la placa PCB, los caminos de ningún elemento se interconecten con otro, por lo cual se elimina la falla por corto circuito.

Posteriormente, dentro del *software* se materializó la idea de las pistas en la placa; por último, se colocó la serigrafía de cada componente. El resultado de este proceso se lo puede observar en la Figura 3.1.

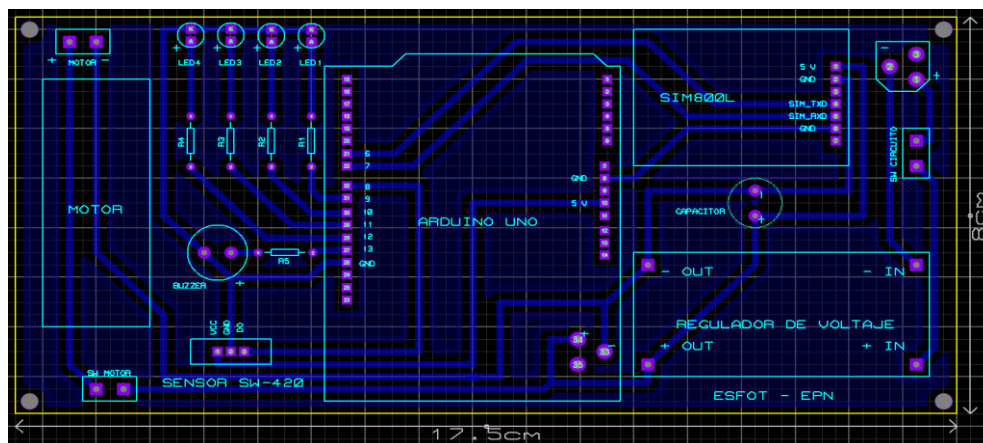


Figura 3.1 Líneas de conexión y serigrafía en Proteus

En el circuito de la Figura 3.1 se aprecia la conexión de cinco resistencias, cada una está interconectada en serie con un elemento, siendo así que cuatro de ellas están enlazadas a LEDs de colores y la quinta resistencia al *buzzer*. La consideración de estos componentes fue con la finalidad de limitar el paso de corriente; por consiguiente, prevenir la subida de tensión y así precautelar el bienestar de los componentes. El cálculo para obtener el valor de cada una de las resistencias se realizó con la Ecuación 3.1.

$$I = \frac{V}{R}$$

Ecuación 3.1 Ley de Ohm [18]

Cabe tener en cuenta que los LEDs de alta luminosidad trabajan con una corriente funcional de 20 (mA) y el voltaje varía dependiendo el color, como se muestra en la Tabla 3.5 [19].

Tabla 3.5 Voltaje de cada LED referente a cada color [19]

Color del LED	Voltaje del LED
Verde	2.4 (V)
Amarillo	1.7 (V) – 2 (V)
Naranja	2.4 (V)
Rojo	1.9 (V)

Ya teniendo el voltaje y la corriente del LED, se procede con el despeje de la resistencia en la ecuación de Ohm, como se observa en la Ecuación 3.2.

$$R = \frac{V_{Fuente} - V_{LED}}{I}$$

Ecuación 3.2 Despejar R para el cálculo de la resistencia [18]

Donde:

- V_{Fuente} : 5 (V) voltaje de Arduino Uno
 V_{LED} : 1.7 (V) - 2.4 (V) voltaje del LED
 I : 20 (mA) corriente

$$R \text{ LED verde} = \frac{5(V) - 2.4(V)}{20 \text{ (mA)}} = 130 \text{ } (\Omega)$$

$$R \text{ LED amarillo} = \frac{5(V) - 1.7(V)}{20 \text{ (mA)}} = 165 \text{ } (\Omega)$$

$$R \text{ LED naranja} = \frac{5(V) - 2.4(V)}{20 \text{ (mA)}} = 130 \text{ } (\Omega)$$

$$R \text{ LED rojo} = \frac{5(V) - 1.9(V)}{20 \text{ (mA)}} = 155 \text{ } (\Omega)$$

Considerando los resultados obtenidos al resolver la ecuación para cada LED y la tabla de resistencias comerciales, se estableció que la resistencia de 330 (Ω) con tolerancia de \pm 5% (J) y código de colores naranja/naranja/café/dorado, es la opción más acertada.

Esto no solo con la finalidad de asegurar una correcta iluminación; sino también, para garantizar un óptimo funcionamiento sin dejar de lado la estética del proyecto, siendo así que se dispuso del mismo valor de resistencia para los cuatro LEDs.

Para el caso del *buzzer*, lo más importante es limitar la corriente y no afectar el pin del Arduino, esto se concretó gracias al uso de la ley de Ohm definida en la Ecuación 3.1 y de igual forma despejada en la Ecuación 3.2.

Donde:

- V_{Fuente} : 5 (V) voltaje de Arduino Uno
 I_{Buzzer} : 1.7 (V) - 2.4 (V) voltaje del LED

$$R \text{ buzzer} = \frac{5(V)}{20 \text{ (mA)}} = 250 \text{ } (\Omega)$$

Es fundamental considerar los parámetros bajo los cuales trabaja el *buzzer*, siendo el principal el uso de una baja resistencia; de lo contrario, el volumen del sonido se reducirá de forma considerable.

A fin de no afectar el funcionamiento, se optó por la resistencia de 82 (Ω) con tolerancia de $\pm 5\%$ (J) y un código de colores gris/rojo/negro/dorado; con esta elección, se limita el paso de corriente y optimiza el volumen de sonido del *buzzer*.

Justificación de consumo energético

En la determinación del consumo energético del prototipo se analizaron los correspondientes consumos de cada uno de los componentes. Para este análisis se descartó los elementos que tienen un bajo consumo. En la Tabla 3.6 se muestra el voltaje de cada uno de los componentes principales en un rango de 3.3 a 5 (V), con lo que se puede trabajar en un valor aproximado dentro de estos valores.

Tabla 3.6 Voltaje de operación componentes principales [11] [1] [4]

	Arduino Uno	SIM800L	Módulo SW-420
Voltaje (V)	5 (V)	5 (V)	3.3 a 5 (V)

En el caso del consumo de corriente, al tratarse de un prototipo que siempre va estar enviando datos, se debe considerar el consumo energético pico de todos los elementos. La Tabla 3.7 detalla el consumo pico de los componentes principales.

Tabla 3.7 Consumo pico componentes principales [11] [1] [4]

	Arduino Uno	SIM800L	Módulo SW-420
Consumo (mA)	40 (mA)	500 (mA)	15 (mA)

Obtenidos los valores de consumo del prototipo, se calcula el consumo total de todo el sistema en la Ecuación 3.3.

$$I_{Sis} = I_{Arduino} + I_{SIM800L} + I_{SW-420}$$

Ecuación 3.3 Corriente del prototipo

Donde:

$I_{Arduino}$:	40 (mA) corriente Arduino Uno
$I_{SIM800L}$:	500 (mA) corriente SIM800L 2.0
I_{SW-420}	:	15 (mA) corriente Módulo SW-420

Por lo tanto:

$$I_{Sis} = 40 \text{ (mA)} + 500 \text{ (mA)} + 15 \text{ (mA)} = 555 \text{ (mA)}$$

Una vez obtenido el consumo total de todo el sistema, se selecciona la batería adecuada mediante la Ecuación 3.4. Cabe recalcar que las 2 (h) son el tiempo tomado en consideración bajo el cual se realizó la prueba de funcionamiento del proyecto.

$$C_{Bat} = I_{Sis} * T$$

Ecuación 3.4 Corriente del prototipo [20]

Donde:

C_{Bat}	:	Capacidad de corriente de la batería
I_{Sis}	:	555 (mA) corriente de todo el sistema
T	:	2 (h) tiempo en horas que se requiere el funcionamiento

Por lo tanto:

$$C_{Bat} = 555 \text{ (mA)} * 2 \text{ (h)} = 1.11 \text{ (Ah)}$$

El resultado obtenido mediante el uso de las ecuaciones, determinó que el prototipo requiere de una batería con la capacidad de 1.11 (Ah).

Justificación de rangos de operación del prototipo

El módulo SW-420 tiene como función abrir y cerrar el contacto eléctrico dependiendo de la vibración o fuerza que se produzca, de manera que cuando el sensor no detecta vibraciones se encuentra en estado cerrado o lanza una salida lógica baja; por el contrario, cuando detecta una vibración alta, cambia el estado a abierto o una salida lógica alta. Este funcionamiento es el resultado del trabajo en conjunto del comparador LM393 y el potenciómetro de 10 (K Ω), ambos dispositivos integrados en el sensor [20].

El LM393 es el que se encarga de comparar los voltajes; el segundo pin del LM393 está conectado al potenciómetro, y el tercer pin está enlazado al sensor de vibración. El

comparador relaciona los voltajes que registran los componentes, para al final pasar una señal a la salida digital que posee el módulo SW-420 [20].

El potenciómetro se encarga de calibrar la sensibilidad del sensor, esta se puede ajustar según se requiera; la posición del módulo es de forma horizontal y el sensor SW-420 en el lado izquierdo. Es importante resaltar que, al girar el potenciómetro en el sentido de las manecillas del reloj, se disminuye la sensibilidad y en sentido anti horario aumenta [20].

Para establecer el rango de funcionamiento del sensor SW-420, se utilizó el documento “Análisis del rendimiento del sensor de vibración SW-420 en herramientas de vibración mediante el uso de un método de lógica difusa”. En dicha investigación, se elaboró un algoritmo basado en la explicación del procedimiento realizado, debido a longevidad del proceso, en la Tabla 3.8 se resume los resultados obtenidos con respecto al nivel de vibración que puede percibir el sensor [21].

Tabla 3.8 Rango de vibraciones percibidas por el sensor [21]

Valor medio de los datos	Clasificación de resultados
0 – 299	Muy Débil
300 – 349	Débil
350 – 599	Medio
600 – 649	Fuerte
650 – 700	Muy Fuerte

La sensibilidad del sensor SW-420 fue regulada mediante el potenciómetro, siendo así que se estableció una sensibilidad mínima; de esta forma se asegura que el sensor funcione en los valores anteriormente mencionados en la Tabla 3.8. Gracias a los resultados obtenidos del documento, se establecieron los mismos valores con los cuales se trabajó en el proyecto; sin embargo, se omitió el nivel muy débil y el rango de muy fuerte, debido a que con valores muy bajos la vibración fue experimentalmente indetectable y con un nivel muy fuerte se supera los 1 000. Cabe recalcar que, esto depende de la sensibilidad que se coloca al sensor. En la Tabla 3.9 se muestra los valores establecidos para el prototipo.

Tabla 3.9 Valores de funcionalidad del prototipo

Valor medio de los datos	Clasificación de resultados
0 – 349	Débil
350 – 599	Moderado
600 – 649	Fuerte
> 650	Muy Fuerte

Diagrama de flujo

El programa comienza con el ingreso de las variables; en este caso, cuatro LEDs de diferente color (verde, amarillo, naranja y rojo), un *buzzer* (cuyo tono funciona en base a las variables frecuencia y duración), el sensor de vibración (SW-420); adicionalmente, se declara la variable del módulo SIM800L. Todas las variables tienen designado un pin del Arduino UNO.

A continuación, se realiza la designación de pines donde se determina si estos son de entrada o de salida; el único pin de entrada corresponde al sensor, el resto son de salida. Añadiendo unas líneas, se establece la velocidad de transmisión a la que trabajan los dispositivos.

Posteriormente, se añade una variable (valores) y se iguala con otra variable (vibración); con ello se consigue que el sensor SW-420 detecte los movimientos en valores numéricos. Con la ayuda del puerto serial del Arduino, se puede imprimir los valores detectados por el sensor.

Las condiciones de los valores detectados han sido definidas en cuatro rangos:

- Si se detecta movimiento entre **0 a 349**, se encenderá el **LED VERDE** por 15 (s); mediante la ejecución de comandos AT, la tarjeta SIM envía un SMS al número de teléfono establecido en el código indicando que se trata de un sismo débil.
- Si se detecta movimiento entre **350 a 599**, se encenderá el **LED AMARILLO** por 15 (s) y el sistema procede a enviar un SMS indicando que el sismo es moderado.
- Si se detecta movimiento entre **600 a 649**, se encenderá el **LED NARANJA** por 15 (s) y se procede a enviar un SMS indicando que el sismo es fuerte.
- Finalmente, si se detecta movimiento **mayor a 650**, se encenderá el **LED ROJO** por 15 (s), donde el prototipo procede a enviar un SMS indicando que el sismo es muy fuerte.

En todos los casos, se emitirá una señal audible con una duración de 10 (s) a través del *buzzer*. Adicionalmente, el SMS de alerta incluye el URL del IGEPN, permitiendo que el usuario pueda leer y verificar la información. En caso de no detectar ningún movimiento, los LEDs se mantendrán apagados al igual que el *buzzer*, el proceso de detección es reiterativo; por lo que, el sensor siempre se encontrará detectando nuevos valores de vibración para emitir nuevas alertas. En la Figura 3.2 se determina lo anteriormente mencionado a través de un diagrama de flujo.

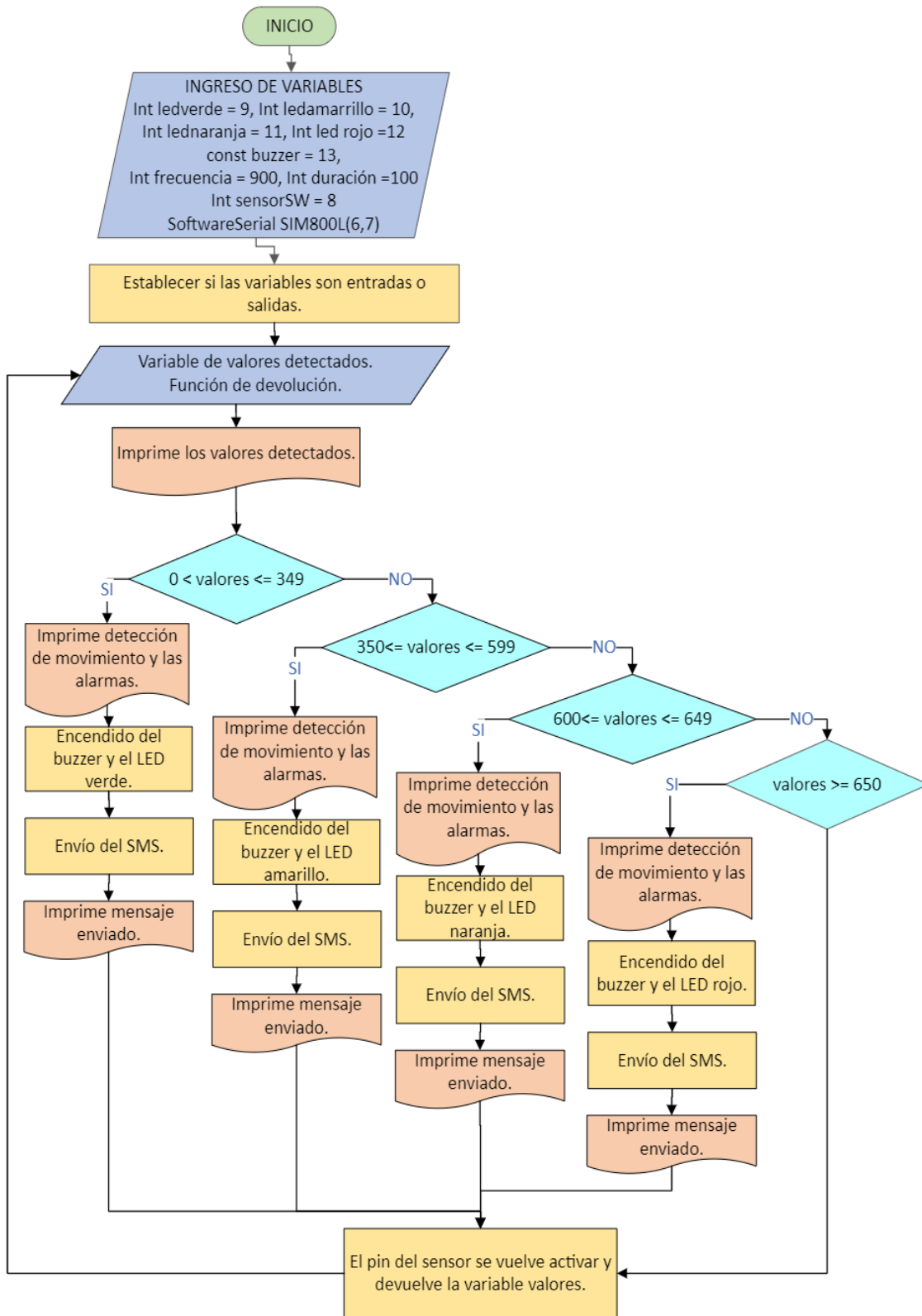


Figura 3.2 Diagrama de flujo

Vista en 3D del prototipo

SketchUp es un *software* utilizado para el diseño gráfico y el modelado de componentes en 3D. Varios profesionales plasman sus ideas utilizando la opción 3D; además, este programa cuenta con una extensa base de datos de la cual los usuarios pueden descargar modelos ya creados o importar sus propias creaciones. La interfaz del *software* es muy intuitiva con el usuario, por lo cual la interacción con la interfaz gráfica de trabajo es sencilla; asimismo, la plataforma es sumamente adaptable, permitiendo que cada usuario trabaje con su propio estilo [22].

En la Figura 3.3 se detalla el trabajo realizado en el *software SketchUp*.

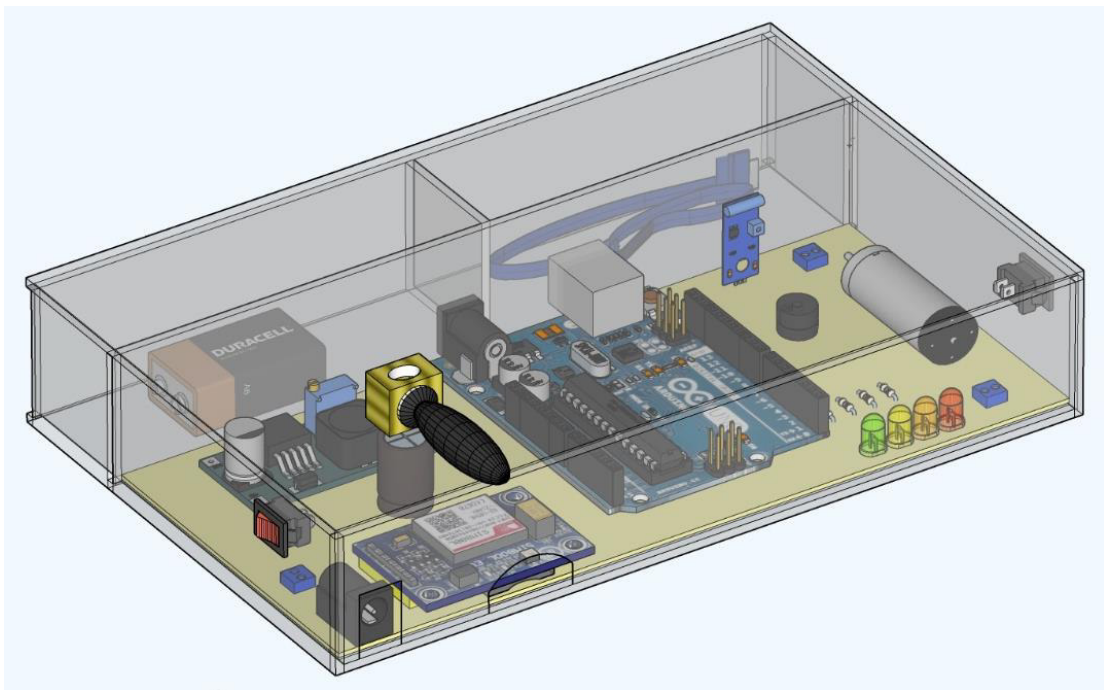


Figura 3.3 Vista 3D del prototipo

En la Figura 3.4 se evidencia la organización de los componentes; además, con la ayuda de números y la creación de una leyenda para el prototipo, se facilita el reconocimiento de cada elemento dentro del chasis de acrílico. En esta imagen se puede visualizar con mayor claridad las tres divisiones de la caja, siendo la de mayor tamaño para la placa PCB y las otras para el cable USB y la batería, respectivamente.

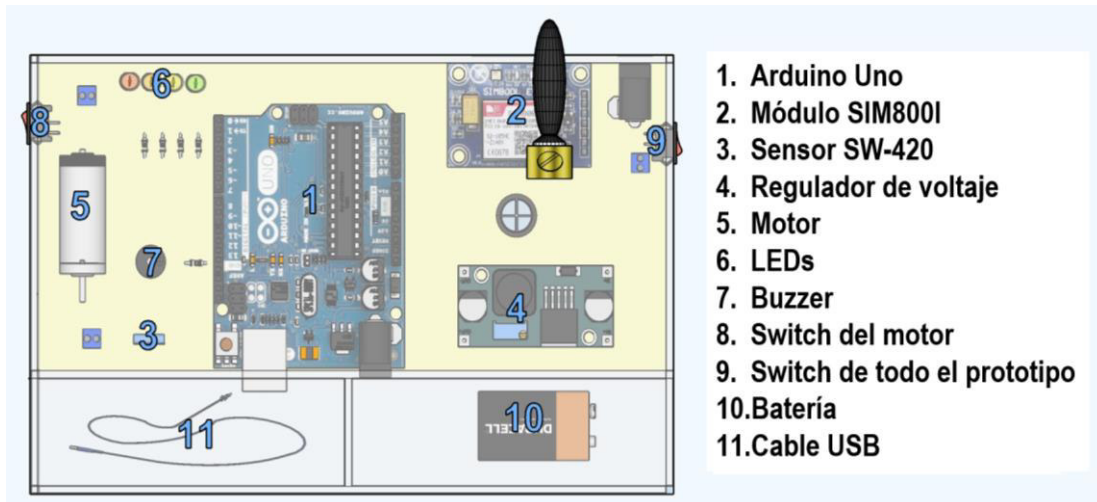


Figura 3.4 Identificación del lugar de cada componente

3.4 Implementación del prototipo

Elaboración de la placa electrónica

Previo a la elaboración de la placa electrónica, es importante contar con el circuito del prototipo en el formato PDF; en este documento debe incluirse las medidas exactas de todos los elementos en el *software Proteus*, esto con la finalidad de que los componentes coincidan en la placa PCB y no tener que imprimir varias veces.

En la Figura 3.5, se observa las líneas de conexión del prototipo y en la Figura 3.6 la serigrafía colocada en forma de espejo para que, al realizar la impresión, la sublimación en la baquelita quede de manera frontal y los dispositivos puedan coincidir sin dificultad.

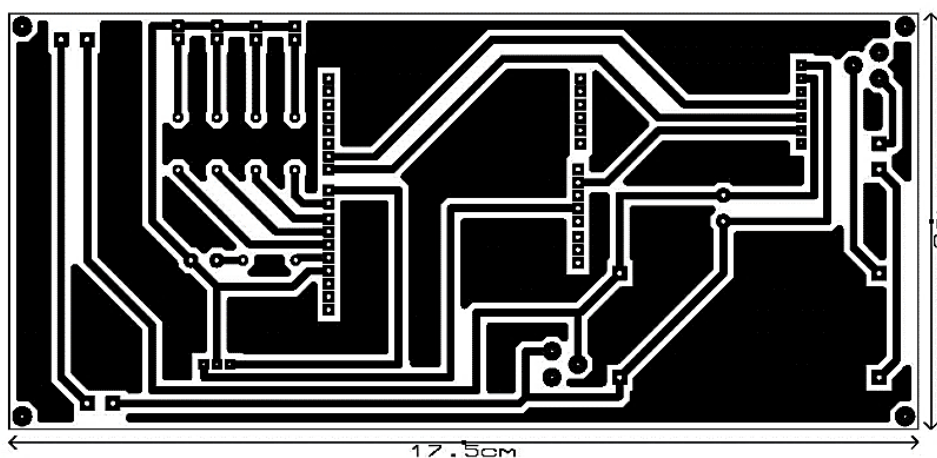


Figura 3.5 Pistas del circuito para el prototipo

En el siguiente paso, se colocó la baquelita encima del papel couché que tenía la impresión de las líneas del circuito; para sujetar bien y que al momento de la sublimación no se moviera, se puso cinta alrededor de la baquelita, consiguiendo que se pegue a otro papel. Con la ayuda de una máquina de sublimación, que se muestra en la Figura 3.8, a una temperatura de 200 (°C) por 30 (s), se puso la baquelita sujeta con el papel.



Figura 3.8 Preparación para la sublimación de la impresión a la baquelita

Cuando se consiguió la sublimación a la baquelita, con mucho cuidado se dejó enfriar tomando las precauciones de no quemarse; en el momento que se enfrió, se situó a la baquelita en una bandeja con agua y con la ayuda de los dedos se extrajo el papel couché de la baquelita. Este proceso y el anterior se realizaron también para la impresión de la serigrafía, pero en el lado de la fibra de vidrio de la baquelita; esto se presenta en la Figura 3.9 junto con el resultado obtenido.

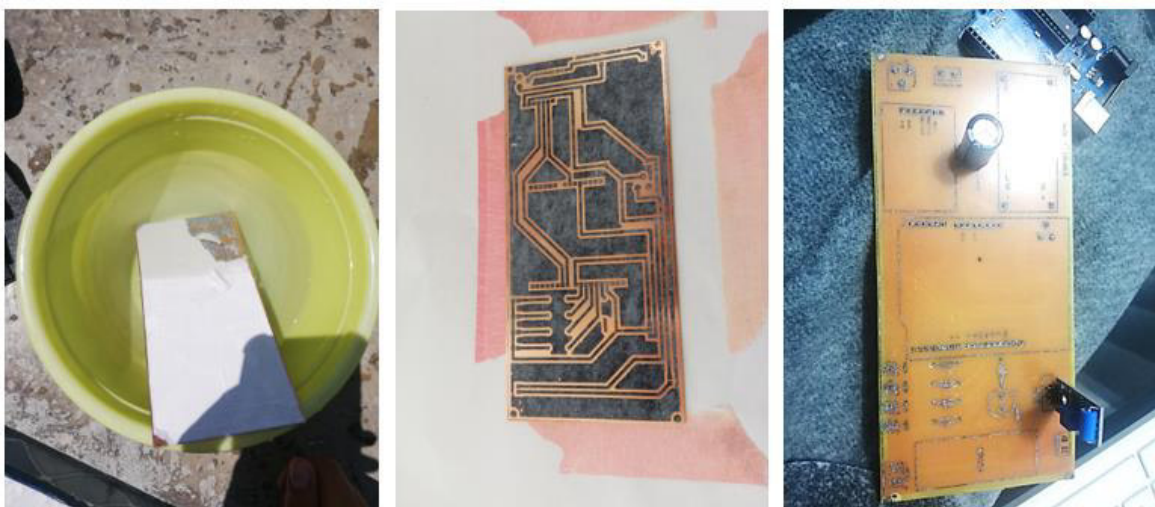


Figura 3.9 Extracción del papel couché y resultados

A continuación, utilizando guantes, se sometió a la baquelita a la acción de ácido férrico. El proceso consistió en dejarle a la baquelita en una fuente con agua cubriéndola completamente. Posteriormente, se puso el ácido férrico en la fuente, este proceso sirve para que el ácido férrico vaya disolviendo el cobre de la baquelita, dejando solo sublimación y con ello protegiendo parte del cobre. Continuando con el proceso, se dejó media hora la baquelita, observando que todo el cobre expuesto o sin sublimación se haya disuelto; se procedió luego a limpiar con agua y extraer la sublimación con lana de acero. Este proceso y resultado se muestra en la Figura 3.10.



Figura 3.10 Proceso y resultado de la utilización del ácido férrico

Una vez concluidos los anteriores procesos, a la baquelita se le llama *printed circuit board* o mejor conocida como placa PCB, en la que se pueden instalar componentes electrónicos. Pero antes de concluir con la placa, se hizo perforaciones en el lugar que van los componentes, con la ayuda de un taladro eléctrico y unas brocas pequeñas, se procedió con este paso, y como evidencia se puede observar la Figura 3.11.

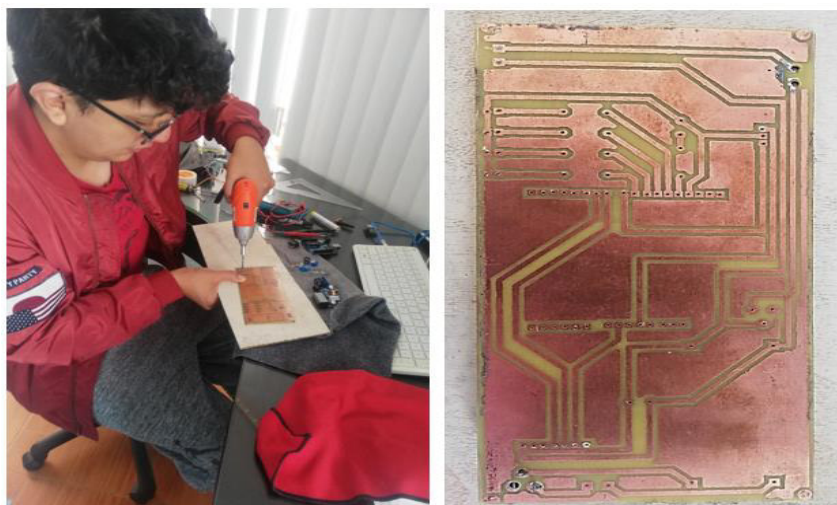


Figura 3.11 Proceso de perforación en la placa PCB

Para finalizar, se procedió con la soldada de los componentes electrónicos; para ello, con la ayuda de un caudín, pasta para soldar y estaño se fue soldando cada uno de los componentes con la precaución de que ningún punto haga corto con otro. Se puede observar en la Figura 3.12 y de igual manera el resultado colocado todos los componentes, incluyendo la conexión del motor y los *switches* que se utilizaron borneras.

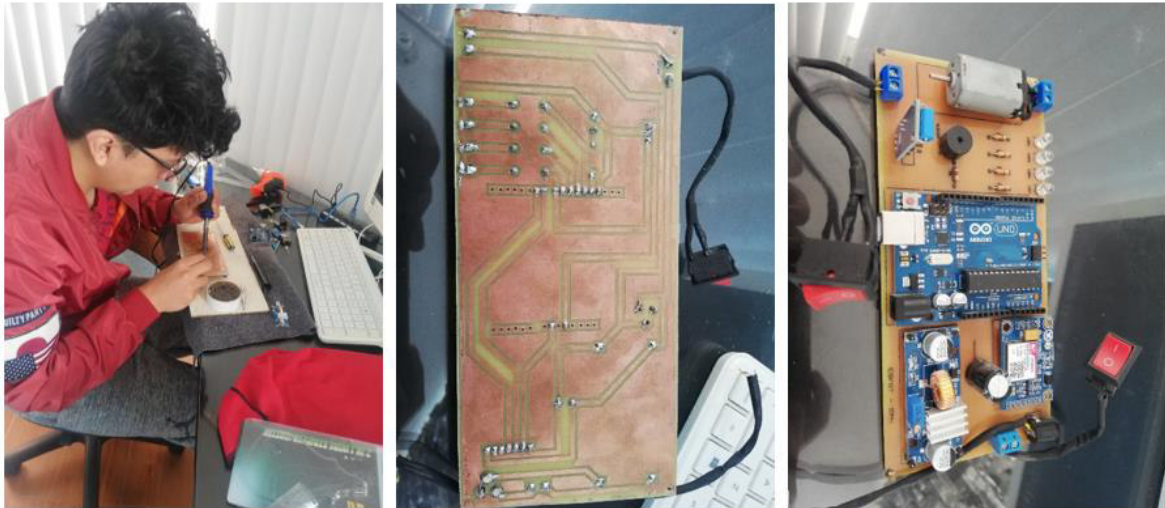


Figura 3.12 Proceso de soldar los componentes y resultados

Alimentación del prototipo

La alimentación de energía del prototipo es de corriente continua, mediante pilas o baterías; por otra parte, también se le implementó una conexión a la red eléctrica mediante un cargador conectado al *jack* que se encuentra en la placa.

Los dispositivos conectados directamente al regulador de voltaje *Step-Down* XL4015 son el Arduino Uno, el motor y el módulo SIM800L; todos ellos funcionan con 5 (V). En el caso del último, tiene una particularidad de funcionar con 1 (A) de corriente y es por ello que se conectó directamente ya que Arduino Uno no brinda como salida la corriente necesaria para el funcionamiento. En consecuencia, el voltaje que se estableció en el regulador *Step-Down* XL4015 es 5 (V) con la finalidad de precautelar el funcionamiento de los componentes; sin embargo, este equipo permite una entrada de hasta 38 (V).

La fuente de alimentación que se le colocó en el prototipo es una batería de 9 (V) de marca Energizer; estas baterías son muy comerciales y asequibles para el usuario, pero al tratarse de un prototipo que siempre está midiendo la vibración, el consumo de la energía va a ser constante y se agotará rápidamente, por lo que es aconsejable adquirir una batería LiPo (Litio y polímero) de larga duración, que por motivos económicos y el alto costo de estas baterías no se pudo adquirir.

Pero la solución más óptima es mantener conectado, mediante un cargador de celular preferiblemente de 5 (V) y 2 (A) a la red eléctrica y también mantener conectada la batería de 9 (V). Para en caso de no funcionar la red eléctrica, pasaría a funcionar solo la energía de la batería.

Programación del módulo Arduino Uno

El código de programación se detalla por secciones a continuación. En el Anexo III consta el código de programación general y en el Anexo IV se incluye el código QR y enlace de acceso al mismo.

En la Figura 3.13, se observa que se incluyen las librerías para una comunicación serial e incluye las variables de los elementos a utilizar con el pin correspondiente al cual se conectaron; para ello, también se optó por incluir la librería y conexión de los pines del módulo SIM800L.

```
#include <SoftwareSerial.h>           // Librería para una comunicación serial.
int ledverde = 9;                     // Incluir variables con el número de pin a utilizar en arduino.
int ledamarillo = 10;
int lednaranja = 11;
int ledrojo = 12;
const int buzzer = 13;
int frecuencia = 900;
int duracion = 50;
int sensorSW = 8;                    // sensor SW420
SoftwareSerial SIM800L(6, 7);        // Incluir los pines para la comunicacion con el modulo SIM800L.
```

Figura 3.13 Librerías y variables del código

En la siguiente parte del código, como se muestra en la Figura 3.14, se coloca la primera función del código, en la que se configuró el comportamiento de las variables. Por un lado, se tiene a los LEDs y la bocina que tienen un comportamiento de salida identificado como *OUTPUT* y por el otro la variable del sensor SW-420 con un comportamiento de entrada identificado como *INPUT*. Incluye también la velocidad con la que interactúan los componentes electrónicos con la computadora, en este caso se colocó 9600 baudios.

```
void setup() {                       // Inicia las variables y configuracion se se comportan como entrada o salida.
  pinMode(ledverde, OUTPUT);
  pinMode(ledamarillo, OUTPUT);
  pinMode(lednaranja, OUTPUT);
  pinMode(ledrojo, OUTPUT);
  pinMode(sensorSW, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);                // Comunicación del arduino UNO con la PC y designación del a velocidad.
  SIM800L.begin(9600);               // Comunicación del modulo SIM800L con la PC y designación del a velocidad.
```

Figura 3.14 Definición de variables y velocidad

En la Figura 3.15 se determinó una función para que vuelva a medir los datos una vez terminada cualquier acción del código; en esta parte, se ejecutó el tiempo en milisegundos en el que el sensor detecta los valores. De la misma manera, se incluyó una línea en la que los valores de la detección se muestran en el monitor serial.

```
void loop(){
  long valores = vibracion(); // Asignación de una función para la repetición de los valores detectados por el sensor.
  delay(10);                 // Función para ejecutar los valores durante un tiempo en milisegundos.
  Serial.println(valores);   // Imprime en el serial monitor los valores detectados.
```

Figura 3.15 Función de retorno de valores e impresión

Para el siguiente paso del código, que se muestra en la Figura 3.16, ya se determinó qué es lo que va a hacer cada elemento; en este caso, se colocó una línea de comando en la que para establecer el rango de valores entre 0 y menores a 349, se envía una serie de alertas. Dichas alertas incluyen el encendido del LED verde por 15 (s), la activación de la bocina por 10 (s) y el envío del SMS al número que se colocó en el código, indicando de un sismo débil y a su vez el URL del IGEPN; de igual manera, se muestra cada paso en el monitor serial.

```
if (valores >0 && valores <=349){
  Serial.println("Movimiento detectado");
  Serial.println("Activando alarma de sismo debil");
  // Función para que el buzzer suene como una alarma.
  for (int x=0; x<=15; x++){
    tone (buzzer, frecuencia, duracion);
    digitalWrite(ledverde, HIGH); // Se prende el led verde al detectar el rango de los valores detectados.
    delay (650);
  }
  // Se envia un SMS al usuario con comandos AT
  SIM800L.write("AT+CMGF=1\r\n"); //Código para dar formato ASCII y para SMS.
  delay(1000);
  SIM800L.write("AT+CMGS="+593983869323"\r\n"); //Código para nuevo SMS <<<----- AÑADIR NUMERO DE TELEFONO DESTINATARIO.
  delay(1000);
  SIM800L.write("ALERTA DE EMERGENCIA!!! sismo debil. Tome las respectivas medidas de precaucion. Mas inf. https://www.facebook.com/");
  delay(1000);
  SIM800L.write((char)26); //Enviar mensaje.
  delay(1000);
  Serial.println("SMS enviado.");
}
```

Figura 3.16 Activación de las alertas por detección de sismo débil

En el caso de que no se cumpla en la detección el rango anteriormente mencionado, se procede a notificar un rango mayor a 350 y menor a 599, activando las alertas del encendido del LED amarillo por 15 (s), al igual que el *buzzer* por 10 (s), y tomando en cuenta el envío del SMS indicando de un sismo moderado al número de teléfono que se estableció en el código. Todo esto se evidencia en la Figura 3.17.

```

// Si no cumple con la anterior función los valores detectados son de otro rango manda las alertas adecu
else if (valores >= 350 && valores <=599){
  Serial.println("Movimiento detectado");
  Serial.println("Activando alarma de sismo moderado");
  // Función para que el buzzer suene como una alarma.
  for (int x=0; x<=15; x++){
    tone (buzzer, frecuencia, duracion);
    digitalWrite(ledamarillo, HIGH); // Se prende el led amarillo al detectar el rango de los valores detectados.
    delay (650);
  }

  // Se envia un SMS al usuario con comandos AT

SIM800L.write("AT+CMGF=1\r\n"); //Código para dar formato ASCII y para SMS.
delay(1000);
SIM800L.write("AT+CMGS="+593983869323"\r\n"); //Código para nuevo SMS <<<----- AÑADIR NUMERO DE TELEFONO DESTINATARIO.
delay(1000);
SIM800L.write("ALERTA DE EMERGENCIA!!! sismo moderado. Tome las respectivas medidas de precaucion. Mas inf. https://www.facebook.com/IGEI
delay(1000);
SIM800L.write((char)26); //Enviar mensaje.
delay(1000);
Serial.println("SMS enviado.");
}

```

Figura 3.17 Activación de las alertas por detección de sismo moderado

Para los siguientes casos, las funciones corresponden a la misma acción de los anteriores casos, modificando el encendido del LED y del contenido del mensaje. Para los valores establecidos entre mayor a 600 y menor a 649, se enciende el LED naranja por 15 (s), la activación de la bocina por 10 (s) y el envío del SMS indicando de un sismo fuerte. Para los valores establecidos mayores a 650, se enciende el LED rojo por 15 (s), la activación de la bocina por 10 (s) y el envío del SMS indicando de un sismo muy fuerte. Todo esto se observa en la Figura 3.18.

```

else if (valores >= 600 && valores <=649){
  Serial.println("Movimiento detectado");
  Serial.println("Activando alarma de sismo fuerte");
  // Función para que el buzzer suene como una alarma.
  for (int x=0; x<=15; x++){
    tone (buzzer, frecuencia, duracion);
    digitalWrite(lednaranja, HIGH); // Se prende el led naranja al detectar el rango de los valores detectados.
    delay (650);
  }

  // Se envia un SMS al usuario con comandos AT

SIM800L.write("AT+CMGF=1\r\n"); //Código para dar formato ASCII y para SMS.
delay(1000);
SIM800L.write("AT+CMGS="+593983869323"\r\n"); //Codigo para nuevo SMS <<<----- AÑADIR NUMERO DE TELEFONO DESTINATARIO.
delay(1000);
SIM800L.write("ALERTA DE EMERGENCIA!!! sismo fuerte. Tome las respectivas medidas de precaucion. Mas inf. https://www.facebook.com/I
delay(1000);
SIM800L.write((char)26); //Enviar mensaje.
delay(1000);
Serial.println("SMS enviado.");
}

// Si no cumple con la anterior función los valores detectados son de otro rango manda las alertas ade
else if (valores >= 650){
  Serial.println("Movimiento detectado");
  Serial.println("Activando alarma de sismo muy fuerte");
  // Función para que el buzzer suene como una alarma.
  for (int x=0; x<=15; x++){
    tone (buzzer, frecuencia, duracion);
    digitalWrite(ledrojo, HIGH); // Se prende el led rojo al detectar el rango de los valores detectados.
    delay (650);
  }

  // Se envia un SMS al usuario con comandos AT

SIM800L.write("AT+CMGF=1\r\n"); //Codigo para dar formato ASCII y para SMS.
delay(1000);
SIM800L.write("AT+CMGS="+593983869323"\r\n"); //Código para nuevo SMS <<<----- AÑADIR NÚMERO DE TELEFONO DESTINATARIO.
delay(1000);
SIM800L.write("ALERTA DE EMERGENCIA!!! sismo muy fuerte. Tome las respectivas medidas de precaucion. Mas inf. https://www.facebook.c
delay(1000);
SIM800L.write((char)26); //Enviar mensaje.
delay(1000);
Serial.println("SMS enviado.");
}

```

Figura 3.18 Activación de las alertas por detección de sismo fuerte y muy fuerte

Cuando el programa no detecte ningún valor nombrado en los anteriores casos o simplemente no detecte movimiento o vibración, procede a apagar todos los dispositivos, como se muestra en la Figura 3.19.

```
else{                                     // Si no detecta valores entonces se procede apagar los elementos.
  digitalWrite(ledverde, LOW);
  digitalWrite(ledamarillo, LOW);
  digitalWrite(lednaranja, LOW);
  digitalWrite(ledrojo, LOW);
  digitalWrite(buzzer, LOW);
}
```

Figura 3.19 Apagado de las alertas por no detectar ningún valor

La Figura 3.20 muestra la función de devolución; una vez ejecutada dicha función, el programa finaliza y reinicia en la creación de variables mencionada en la Figura 3.15. Estas líneas de comando hacen que el pin del sensor vuelva a detectar y notificar valores, determinando si existe o no una vibración. En caso de ser positiva la detección, se procederá con la activación de los anteriores pasos.

```
long vibracion(){
  long valores=pulseIn (sensorSW, HIGH); //Espera a que el pin se ponga HIGH y devuelve a la variable valores
  return valores;
}
```

Figura 3.20 Función de devolución

Implementación del prototipo en la maqueta

En la implementación del prototipo se estableció el diseño de un chasis de acrílico de un grosor de 2 (mm). En la base del chasis se colocó un acrílico de color negro y todo lo demás incluyendo paredes y la tapa es acrílico transparente; el chasis tiene tres compartimientos, como se evidencia en la Figura 3.21. En el módulo de mayor dimensión se colocó la placa PCB con todos los dispositivos.

El chasis cuenta con dos perforaciones adecuadas en los laterales para la colocación de los dos *switches*, una perforación para el paso de un cable al compartimiento de la batería, otra perforación en la pared del módulo para la conexión del cable USB del Arduino. Además, tiene dos perforaciones en la parte delantera, una para conectar una fuente de alimentación con ayuda del *jack* y otra para la extracción de la tarjeta SIM, y para finalizar una perforación en la tapa en la que se colocó la antena del módulo SIM800L.



Figura 3.21 Implementación del prototipo en el chasis

3.5 Pruebas de funcionamiento

Una vez implementado el prototipo en el chasis, se realizaron pruebas de funcionamiento y la verificación de un desempeño óptimo.

Después de haber desarrollado el código y sin arrojar ningún error en el *software* Arduino IDE, se subió el código en la placa Arduino, como se observa en la Figura 3.22; la flecha superior indica el botón para subir el programa y la flecha inferior muestra que no existe ningún error.



Figura 3.22 Verificación del código en Arduino IDE

Para probar que cada componente cumple con su función, primero se realizaron las pruebas de alimentación tanto con la batería de 9 (V) como con el cargador de 5 (V) y 2 (A). En la Figura 3.23 se muestra el mismo voltaje de 5.162 (V), conectando solo la batería y también conectada junto con el cargador; esto se debe al regulador de voltaje anteriormente mencionado.



Figura 3.23 Comprobación de funcionamiento con batería y cargador

Una vez comprobado el funcionamiento del sistema de alimentación, se debe corroborar la funcionalidad de cada dispositivo, por dicha razón el *chip* debe estar insertado dentro del módulo SIM800L. El siguiente paso es encender el *switch* de todo el sistema ubicado en el lado de la antena. Se observa si en el módulo SIM800L el segundo LED parpadea cada 5 (s); si la palpitación es muy rápida significa que no recepta señal.

Una vez encendido el prototipo, primero se realizó la prueba de funcionamiento simulando vibraciones con el motor, este circuito se enciende del *switch* ubicado en el lado del motor, y da como resultado vibraciones de magnitud débil al activarlo tal y como se evidencia en la Figura 3.24.

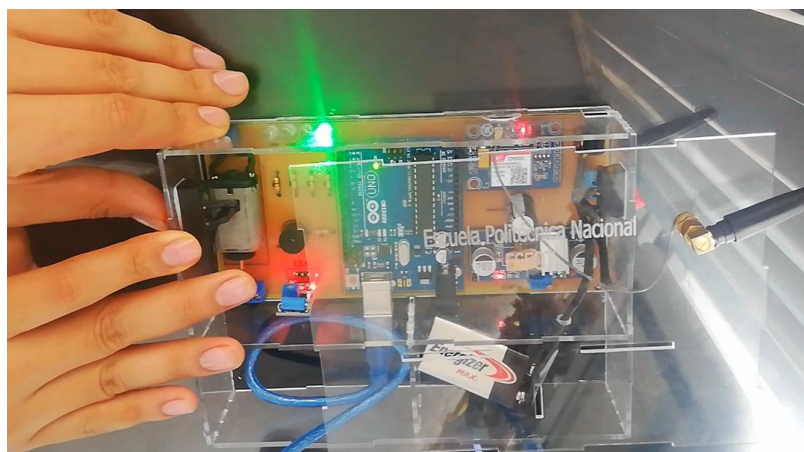


Figura 3.24 Verificación de vibraciones con el motor

Para las siguientes pruebas, los movimientos se los realizó moviendo la mesa simulando a los movimientos sísmicos con diferente intensidad de movimiento; la evidencia de los valores arrojados se observa en el *software* monitor serie de Arduino Uno. La Figura 3.25 muestra los resultados obtenidos de cada uno de los movimientos en los rangos establecidos de la Tabla 3.9 antes mencionada.

<pre>COM3 0 0 0 121 Movimiento detectado Activando alarma de sismo debil SMS enviado. 0</pre>	<pre>COM3 377 Movimiento detectado Activando alarma de sismo moderado SMS enviado. 0 0 0</pre>
<pre>COM3 621 Movimiento detectado Activando alarma de sismo fuerte SMS enviado. 0 0 0</pre>	<pre>COM3 1209 Movimiento detectado Activando alarma de sismo muy fuerte SMS enviado. 0</pre>

Figura 3.25 Detección de los movimientos en el monitor serie

Las pruebas en el prototipo se evidencian en la Figura 3.26, recordando que el encendido de cada uno de los LEDs representa el rango de movimiento y alertando la magnitud del mismo. En el caso del LED verde se indica que el movimiento detectado es un sismo débil, el LED amarillo indica que es un sismo moderado, el LED naranja indica que es un sismo fuerte y el LED rojo indica que el sismo es muy fuerte.

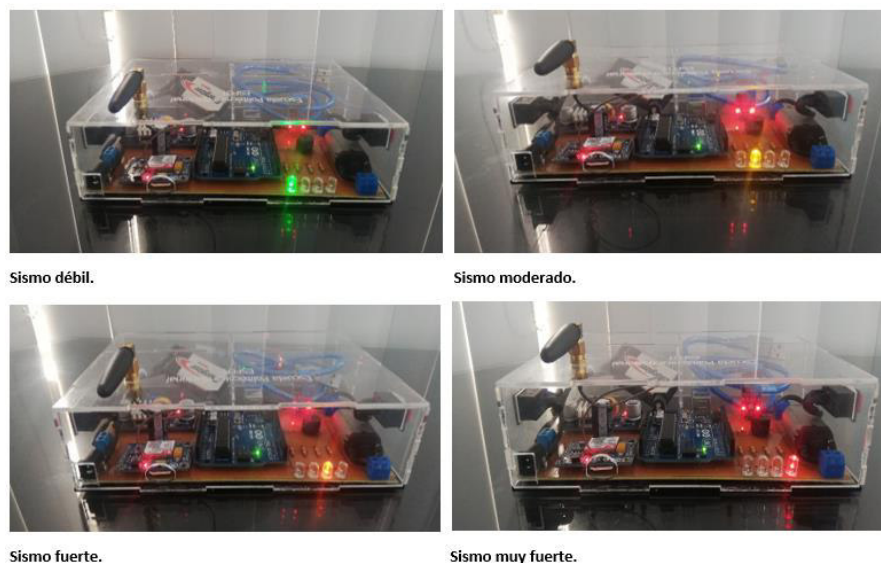


Figura 3.26 Funcionamiento del prototipo en el encendido de los LEDs

De la misma forma, en la Figura 3.27 se evidencia el correcto envío del SMS al número programado en el código, indicando el contenido del mensaje colocado en el código de cada evento suscitado.

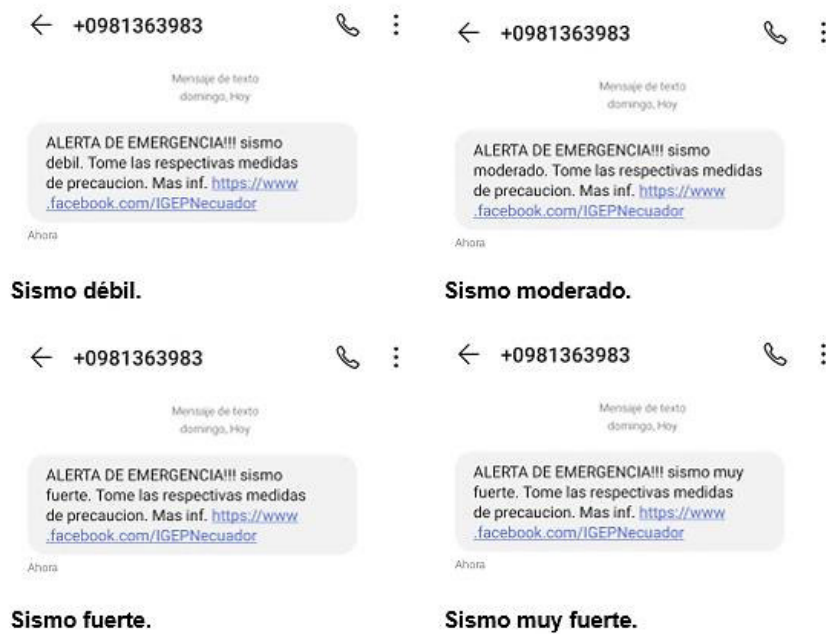


Figura 3.27 Correcto envío del SMS en cada evento suscitado

En la Figura 3.28 se puede observar el resultado del nivel de movimiento que arrojaron los valores que se mostraron como ejemplo para la demostración del funcionamiento del prototipo.

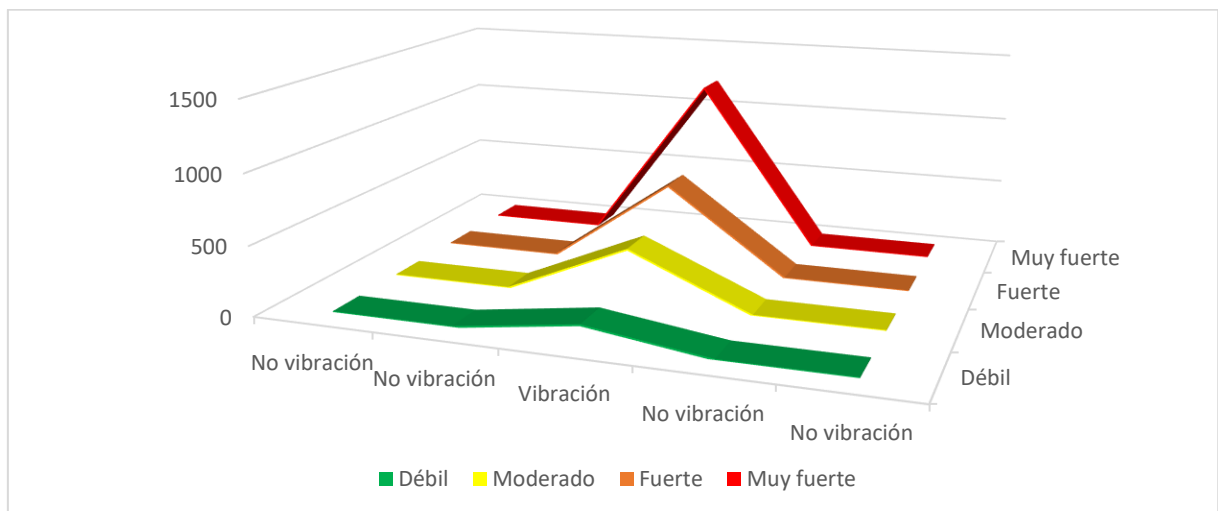


Figura 3.28 Resultado de los valores del ejemplo de demostración

Una vez cubiertos todos los puntos antes mencionados, el prototipo puede ser instalado en cualquier lugar. Además, para garantizar un eficiente mantenimiento y administración del

prototipo, se ha considerado necesario contar con un manual que detalle los procedimientos para realizar la modificación del mensaje SMS, cambiar el número telefónico y extraer la batería. En el Anexo III se incluye los códigos QR y enlaces que contienen los videos de verificación de funcionamiento del prototipo y manual de usuario.

4. CONCLUSIONES

- El sistema de alerta por detección de vibraciones sísmicas, es un proyecto innovador que integra tecnología en beneficio de la sociedad. Su implementación busca solventar la problemática generada por la poca o nula percepción humana ante los movimientos sísmicos; del mismo modo, la falta de información durante la ocurrencia de estos eventos, de esta forma es posible precautelar la integridad física de los beneficiarios.
- El establecer los requerimientos mínimos bajo los cuales debe funcionar el proyecto, es esencial porque una vez definido el uso y la finalidad del sistema es más sencillo determinar el alcance del proyecto. Además, con un análisis detallado es posible concretar ciertos puntos, tales como la necesidad de contar con una fuente de alimentación independiente, un medio de comunicación vía SMS, dimensiones y ubicación del prototipo.
- En el mercado nacional e internacional existe una infinidad de componentes electrónicos que pueden ser utilizados en el desarrollo del prototipo; sin embargo, para la adquisición es importante realizar una búsqueda exhaustiva considerando las características generales que debe abastecer el proyecto, tales como funcionalidad, compatibilidad, valor de la fuente de alimentación en la que trabajan, dimensiones y precios de los elementos.
- En la creación del diseño, el uso de programas como *Proteus* y *SketchUp*, sirvieron para la correcta colocación de todos los componentes integrados en el prototipo, incluyendo las adaptaciones en la caja de acrílico. De la misma manera, al sensor SW-420 se le puede establecer la sensibilidad de acuerdo con los valores definidos y así conseguir el buen funcionamiento del sistema.
- La programación de Arduino IDE es de alto nivel, en la que se incluye variedad de librerías útiles para los módulos utilizados y un extenso repositorio de información. Esto facilita la comprensión del código a un usuario que desconoce la programación y que es necesario cuando se requiera cambiar el número de teléfono, por ejemplo.

De la misma manera, la caja de acrílico fue necesaria; a parte de la protección que le da al prototipo, las ranuras que posee incluyendo la de extracción y colocación de la tarjeta micro SIM le proporciona al usuario una interfaz adecuada para conectarse con el prototipo.

- Mediante las pruebas, se verificó que el prototipo cumple con el funcionamiento correcto de todos los requerimientos, pero se tiene que realizar varios intentos ya que causar movimiento o vibración de la intensidad que se desea es un poco complicado. Por parte de la alimentación, conectando la batería o el cargador, no se tuvo ningún inconveniente en el funcionamiento, inclusive si están ambos conectados.

5. RECOMENDACIONES

- Es recomendable instalar el prototipo en zonas con mayor incidencia de eventos telúricos, ya que su funcionalidad se centra en la detección y alerta de los mismos. También, se debe considerar que el sector cuente con acceso mínimo a redes de segunda generación.
- Se recomienda considerar el envío del mensaje SMS a más de un número telefónico, para lo cual se debe añadir las líneas correspondientes en el código de programación. De la misma manera, se debe considerar que mediante el código es posible activar o desactivar las alertas.
- Dado que el presente proyecto se centra en la parte de la electrónica y comunicación inalámbrica, más no en el estudio de sismos, se recomienda que para futuros proyectos que estén orientados a la investigación de eventos telúricos, se definan los rangos numéricos en base a pruebas experimentales propias.
- Se recomienda considerar el reemplazo de la batería por una fuente de alimentación de larga duración; por ejemplo, una batería LiPo de larga duración o un panel solar. De la misma manera, el chasis podría ser de un material más rígido y con mayor protección en las ranuras.
- Al momento de la creación de la placa PCB, si surgen complicaciones en el proceso, como la no impresión de algunos circuitos de conexión en la baquelita, se puede corregir con un marcador permanente. Además, en el programa *Proteus* se debe colocar un margen del espacio sobrante de las líneas de conexión para que así en

el proceso de quemado de cobre no se demore mucho tiempo y tenga una mejor estética.

- Antes del funcionamiento de todo el prototipo, se recomienda revisar si el módulo SIM800L emite señal; esto se comprueba insertando primero la micro SIM, prendiendo el prototipo y verificando si el parpadeo del segundo LED es de 5 (s). Si no es así y es contante, se debe a que se colocó mal la micro SIM o a problemas de alimentación, por lo que se recomienda establecer un voltaje de 5 (V) y 2 (A).
- Finalmente, si en el lugar en el que se encuentra el prototipo no se recepta señal de la operadora, se recomienda cambiar de lugar el prototipo hasta que recepte señal de la operadora que se vaya utilizar.

6. REFERENCIAS BIBLIOGRÁFICAS

- [1] Componentes101, «Componentes101,» 2020. [En línea]. Available: <https://componentes101.com/sensors/sw-420-vibration-sensor-module>. [Último acceso: 18 11 2021].
- [2] DescubreArduino, «DescubreArduino,» 17 04 2020. [En línea]. Available: <https://descubrearduino.com/sim800l-gsm/>. [Último acceso: 18 11 2021].
- [3] Arduino.cl, «Arduino.cl,» 16 11 2021. [En línea]. Available: <https://arduino.cl/>. [Último acceso: 18 11 2021].
- [4] UNIT Electronics, «UNIT Electronics,» 15 11 2021. [En línea]. Available: <https://uelectronics.com/producto/modulo-sim800l-gsm-gprs-v2-0-con-antena/>. [Último acceso: 18 11 2021].
- [5] Instituto Geofísico de la EPN, «Instituto Geofísico de la EPN,» 2016. [En línea]. Available: <https://www.igepn.edu.ec/component/fsf/?view=faq&catid=2>. [Último acceso: 17 11 2021].
- [6] Comisión Nacional de Prevención de Riesgos y Atención de Emergencia, «CNE.go.cr,» 2021. [En línea]. Available: https://www.cne.go.cr/reduccion_riesgo/informacion_educativa/recomentaciones_consejos/sismo.aspx. [Último acceso: 17 11 2021].
- [7] Servicio Geológico Mexicano, «sgm.gob.mx,» 2011. [En línea]. Available: <https://www.sgm.gob.mx/Web/MuseoVirtual/Riesgos-geologicos/Causas-caracteristicas-e-impactos.html>. [Último acceso: 16 11 2021].
- [8] INEC, «Ecuador en cifras,» 20 04 2017. [En línea]. Available: <https://www.ecuadorencifras.gob.ec/documentos/web-inec/Bibliotecas/Libros/Memorias%2013%20abr%202017.pdf>. [Último acceso: 13 01 2022].

- [9] El Comercio, «El Comercio,» 28 05 2020. [En línea]. Available: <https://www.elcomercio.com/actualidad/ecuador/enjambre-sismico-morona-habitantes-temblores.html>. [Último acceso: 16 11 2021].
- [10] M. J, «Publimetro.co,» 23 03 2019. [En línea]. Available: <https://www.publimetro.co/co/noticias/2019/03/23/no-sienten-los-temblores-algunas-personas-esta-la-razon.html>. [Último acceso: 16 11 2021].
- [11] Ingeniería Mecafenix, «Ingeniería Mecafenix,» 25 04 2017. [En línea]. Available: <https://www.ingmecafenix.com/electronica/arduino/>. [Último acceso: 28 12 2021].
- [12] Gadget-info, «Gadget-info.com,» 2019. [En línea]. Available: <https://es.gadget-info.com/14053-arduino-vs-raspberry-pi-a-detailed-comparison>. [Último acceso: 23 11 2021].
- [13] 5Hertz, «5Hertz.com,» 2021. [En línea]. Available: https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=12. [Último acceso: 23 11 2021].
- [14] UNIT Electronics, «UElectronics.com,» 26 11 2021. [En línea]. Available: <https://uelectronics.com/producto/modulo-ky-002-sensor-vibracion/>. [Último acceso: 26 11 2021].
- [15] Components101, «Components101,» 2021. [En línea]. Available: <https://components101.com/article/sim800l-vs-sim900a-gsm-module>. [Último acceso: 22 12 2021].
- [16] UNIT Electronics, «UNITElectronics.com,» 22 12 2021. [En línea]. Available: <https://uelectronics.com/producto/sim800l-modulo-gprs-gsm-sim800-con-antena/>. [Último acceso: 22 12 2021].
- [17] UNIT Electronics, «UNIT Electronics,» 05 01 2022. [En línea]. Available: <https://uelectronics.com/producto/modulo-regulador-step-up-1-25-35v-xl4015-5-40v-5a/>. [Último acceso: 05 01 2022].
- [18] G. Rosinar, «TodaMateria,» 18 01 2022. [En línea]. Available: <https://www.todamateria.com/ley-de-ohm/>. [Último acceso: 19 01 2022].
- [19] A. «HETPTO,» 24 03 2015. [En línea]. Available: <https://hetpro-store.com/TUTORIALES/resistencia-de-led/>. [Último acceso: 05 01 2022].
- [20] AutoSolar, «¿Cómo calcular la capacidad de la baterías?,» 2021. [En línea]. Available: <https://autosolar.es/blog/baterias-placas-solares/como-calcular-la-capacidad-de-las-baterias>. [Último acceso: 05 12 2021].
- [21] Microcontrollers, «Microcontrollerslab,» 17 03 2021. [En línea]. Available: <https://microcontrollerslab.com/sw-420-vibration-sensor-module-pinout-interfacing-arduino-features/>. [Último acceso: 15 12 2021].

- [22] E. A. P. Ikhwan, S. Poltak y K. M. N. Mahyddin, «sciteperss.org,» 2020. [En línea]. Available: <https://www.scitepress.org/Papers/2020/103360/103360.pdf>. [Último acceso: 05 01 2022].
- [23] Q. Italo, «ARCUX,» 26 05 2020. [En línea]. Available: <https://arcux.net/blog/que-es-sketchup-funcionamiento/>. [Último acceso: 05 01 2022].

7. ANEXOS

ANEXO I. Certificado de Originalidad

ANEXO II. Certificado de Funcionamiento

ANEXO III. Código de programación

ANEXO IV Enlaces

Anexo 1. Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 17 de febrero de 2022

De mi consideración:

Yo, Fanny Paulina Flores Estévez, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA POR DETECCIÓN DE EVENTOS SÍSMICOS asociado al proyecto IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA POR DETECCIÓN DE EVENTOS SÍSMICOS elaborado por el/la estudiante HUGO FERNANDO CAÑAVERAL VELÁSQUEZ de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 9%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin en https://epnecuador-my.sharepoint.com/:b:/g/personal/fanny_flores_epn_edu_ec/EVZJPYwM05FNrTJz0agvsHYBe6Bete7jrgFf8phN4fm9nA?e=yPTQOf

Atentamente,

Fanny Paulina Flores Estévez
Profesora Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos

Anexo 2. Certificado de Funcionamiento



ESCUELA POLITECNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 26 de enero de 2022

**CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE
TITULACIÓN**

Yo, Fanny Paulina Flores Estévez, docente a tiempo completo de la Escuela Politécnica Nacional y como directora de este trabajo de integración curricular, certifico que he constatado el correcto funcionamiento del prototipo de un sistema de alerta por detección de eventos sísmicos, el cual fue implementado por el estudiante Hugo Fernando Cañaverl Velásquez.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios.

DIRECTOR

Ing. Fanny Flores., MSc.

Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 2. EXT: 2712
email: fanny.flores@epn.edu.ec Quito-Ecuador

Anexo 3. Código de Programación

```
#include <SoftwareSerial.h> // Librería para una comunicación serial.
int ledverde = 9; // Incluir variables con el número de pin a utilizar en Arduino.
int ledamarillo = 10;
int lednaranja = 11;
int ledrojo = 12;
const int buzzer = 13;
int frecuencia = 900;
int duracion = 50;
int sensorSW =8; // sensor SW420
SoftwareSerial SIM800L(6, 7); // Incluir los pines para la comunicación con el módulo SIM800L.
void setup(){ // Inicia las variables y configuración se comportan como entrada o salida.
  pinMode(ledverde, OUTPUT);
  pinMode(ledamarillo, OUTPUT);
  pinMode(lednaranja, OUTPUT);
  pinMode(ledrojo, OUTPUT);
  pinMode(sensorSW, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600); // Comunicación del Arduino UNO con la PC y designación del a
  velocidad.
  SIM800L.begin(9600); // Comunicación del módulo SIM800L con la PC y designación del
  a velocidad.
}
void loop(){
  long valores = vibracion(); // Asignación de una función para la repetición de los valores detectados
  por el sensor.
  delay(10); // Función para ejecutar los valores durante un tiempo en milisegundos.
  Serial.println(valores); // Imprime en el serial monitor los valores detectados.

  // Función para que los valores detectados entre un rango y manda las
  alertas adecuadas.
  if (valores >0 && valores <=349){
    Serial.println("Movimiento detectado");
    Serial.println("Activando alarma de sismo débil");
    // Función para que el buzzer suene como una alarma.
    for (int x=0; x<=15; x++){
      tone (buzzer, frecuencia, duracion);
      digitalWrite(ledverde, HIGH); // Se prende el led verde al detectar el rango de los valores
      detectados.
      delay (650);
    }
  }
}
```

```

}

// Se envía un SMS al usuario con comandos AT
SIM800L.write("AT+CMGF=1\r\n"); //Código para dar formato ASCII y para SMS.
delay(1000);
SIM800L.write("AT+CMGS=\"+593983776345\"\r\n"); //Código para nuevo SMS <<<-----
AÑADIR NUMERO DE TELEFONO DESTINATARIO.
delay(1000);
SIM800L.write("ALERTA DE EMERGENCIA!!! sismo debil. Tome las respectivas medidas de
precaucion. Mas inf. https://www.facebook.com/IGEPNecuador "); //Contenido del SMS
delay(1000);
SIM800L.write((char)26); //Enviar mensaje.
delay(1000);
Serial.println("SMS enviado.");
}

// Si no cumple con la anterior función los valores detectados son de otro rango
manda las alertas adecuadas.
else if (valores >= 350 && valores <=599){
  Serial.println("Movimiento detectado");
  Serial.println("Activando alarma de sismo moderado");
  // Función para que el buzzer suene como una alarma.
  for (int x=0; x<=15; x++){
    tone (buzzer, frecuencia, duracion);
    digitalWrite(ledamarillo, HIGH); // Se prende el led amarillo al detectar el rango de los valores
detectados.
    delay (650);
  }

// Se envía un SMS al usuario con comandos AT
SIM800L.write("AT+CMGF=1\r\n"); //Código para dar formato ASCII y para SMS.
delay(1000);
SIM800L.write("AT+CMGS=\"+593983776345\"\r\n"); //Código para nuevo SMS <<<-----
AÑADIR NUMERO DE TELEFONO DESTINATARIO.
delay(1000);
SIM800L.write("ALERTA DE EMERGENCIA!!! sismo moderado. Tome las respectivas medidas
de precaucion. Mas inf. https://www.facebook.com/IGEPNecuador "); //Contenido del SMS
delay(1000);
SIM800L.write((char)26); //Enviar mensaje.
delay(1000);
Serial.println("SMS enviado.");
}

// Si no cumple con la anterior función los valores detectados son de otro rango
manda las alertas adecuadas.
else if (valores >= 600 && valores <=649){
  Serial.println("Movimiento detectado");

```

```

Serial.println("Activando alarma de sismo fuerte");
                // Función para que el buzzer suene como una alarma.
for (int x=0; x<=15; x++){
tone (buzzer, frecuencia, duracion);
digitalWrite(lednaranja, HIGH);// Se prende el led naranja al detectar el rango de los valores
detectados.
delay (650);
}

                // Se envía un SMS al usuario con comandos AT
SIM800L.write("AT+CMGF=1\r\n");//Código para dar formato ASCII y para SMS.
delay(1000);
SIM800L.write("AT+CMGS=\"+593983776345\"\r\n");//Código para nuevo SMS <<<-----
AÑADIR NUMERO DE TELEFONO DESTINATARIO.
delay(1000);
SIM800L.write("ALERTA DE EMERGENCIA!!! sismo fuerte. Tome las respectivas medidas de
precaucion. Mas inf. https://www.facebook.com/IGEPNecuador "); // <<--- AÑADIR MENSAJE
delay(1000);
SIM800L.write((char)26); //Enviar mensaje.
delay(1000);
Serial.println("SMS enviado.");
}

                // Si no cumple con la anterior función los valores detectados son de otro rango
manda las alertas adecuadas.
else if (valores >= 650){
Serial.println("Movimiento detectado");
Serial.println("Activando alarma de sismo muy fuerte");
                // Función para que el buzzer suene como una alarma.
for (int x=0; x<=15; x++){
tone (buzzer, frecuencia, duracion);
digitalWrite(ledrojo, HIGH);// Se prende el led rojo al detectar el rango de los valores detectados.
delay (650);
}

                // Se envía un SMS al usuario con comandos AT
SIM800L.write("AT+CMGF=1\r\n");//Código para dar formato ASCII y para SMS.
delay(1000);
SIM800L.write("AT+CMGS=\"+593983776345\"\r\n");//Código para nuevo SMS <<<-----
AÑADIR NÚMERO DE TELEFONO DESTINATARIO.
delay(1000);
SIM800L.write("ALERTA DE EMERGENCIA!!! sismo muy fuerte. Tome las respectivas medidas
de precaucion. Mas inf. https://www.facebook.com/IGEPNecuador "); //Contenido del SMS
delay(1000);
SIM800L.write((char)26); //Enviar mensaje.
delay(1000);

```

```
Serial.println("SMS enviado.");
}
else{           // Si no detecta valores entonces se procede a apagar los elementos.
  digitalWrite(ledverde, LOW);
  digitalWrite(ledamarillo, LOW);
  digitalWrite(lednaranja, LOW);
  digitalWrite(ledrojo, LOW);
  digitalWrite(buzzer, LOW);
}
}
long vibracion(){
  long valores=pulseIn (sensorSW, HIGH); //Espera a que el pin se ponga HIGH y devuelve a la
  variable valores
  return valores;
}
```

Anexo 4. Enlaces

El siguiente código QR y enlace corresponden al código de programación.



https://epnecuador-my.sharepoint.com/:t/g/personal/hugo_canaveral_epn_edu_ec/EZNDsYYbQBFPITWS6kmC7BEB47KrHqEeCoKltyYe9gLVgg?e=d5ZR0X

El siguiente código QR y enlace corresponden al video de verificación de funcionamiento del prototipo.



https://epnecuador-my.sharepoint.com/:v/g/personal/hugo_canaveral_epn_edu_ec/EYfEiySY6bIHrctue-k0fb4BI57owoSDDavdSe3KmSM0YQ?e=mNHyz

El siguiente código QR y enlace corresponden al video de manual de usuario del prototipo.



https://epnecuador-my.sharepoint.com/:v/g/personal/hugo_canaveral_epn_edu_ec/EbNsFa5cYTBBvOTdkmI3MCYBdHb7-HVixqSuhQjC9by2Tg?e=blyFz1