

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería en Electrónica**



**Red de sensores para el análisis de vibración en puentes utilizando acelerómetros triaxiales.**

**Informe de proyecto de graduación para optar por el título de Ingeniero en  
Electrónica con el grado académico de Licenciatura.**

**Omar Fernando Vargas Ruiz**

**Cartago, abril 2018**

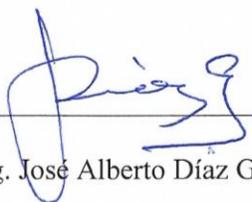


INSTITUTO TECNOLÓGICO DE COSTA RICA  
ESCUELA DE INGENIERÍA ELECTRÓNICA  
PROYECTO DE GRADUACIÓN  
ACTA DE APROBACIÓN

Defensa de Proyecto de Graduación  
Requisito para optar por el título de Ingeniero en Electrónica  
Grado Académico de Licenciatura  
Instituto Tecnológico de Costa Rica

El Tribunal Evaluador aprueba la defensa del proyecto de graduación denominado “Red de sensores para el análisis de vibración en puentes utilizando acelerómetros triaxiales”, realizado por el señor Omar Fernando Vargas Ruiz y hace constar que cumple con las normas establecidas por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal Evaluador



Ing. José Alberto Díaz García

Profesor lector



Ing. Adolfo Chaves Jiménez

Profesor lector



Ing. José Miguel Barboza Retana

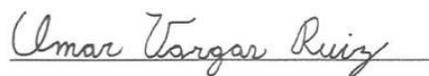
Profesor asesor

Cartago, abril 2018



## Declaración de autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado, en su totalidad, por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios. En los casos en que he utilizado material bibliográfico, he procedido a indicar las fuentes mediante citas. En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



Omar Fernando Vargas Ruiz

Cédula:3-0448-0720

## Resumen

Dentro de la valoración del estado estructural de los puentes que se realiza en nuestro país, una de las pruebas que se aplican es el análisis de vibración que estas presentan a perturbaciones mecánicas, lo que permite observar el comportamiento dinámico de la estructura y así tomar las medidas correspondientes de mantenimiento ante cualquier daño o falla que pueda estar presente en la estructura. El proyecto e-Bridge de la Escuela de Ingeniería en Construcción del Instituto Tecnológico de Costa Rica, desde el año 2011 se ha enfocado en buscar soluciones que permitan hacer los estudios de las estructuras tipo puente con un sistema integrado, el cual busca que el proceso de análisis de las diferentes variables físicas que afectan a una estructura se realice de manera confiable y con sistemas de medición implementados dentro del mismo proyecto. El presente proyecto muestra la implementación de una red de sensores por medio del uso de acelerómetros triaxiales, con el fin de conocer el comportamiento en los tres ejes coordenados de las estructuras con respecto a la medición de las vibraciones que estas presentan ante la presencia de perturbaciones mecánicas en la estructura. La comunicación entre los nodos de medición y el nodo central se realiza por medio de la topología tipo estrella de forma inalámbrica, en donde el último se encarga de almacenar los datos obtenidos de las mediciones realizadas en cada uno de los nodos secundario. El sistema integrado presenta un consumo de potencia de 1,8 W en estados de medición, siendo el estado de máximo consumo de energía. Para la verificación del funcionamiento del sistema se realizaron mediciones con una viga cilíndrica de hierro de 6 m de longitud y 4mm de espesor, donde estas se comparan con las obtenidas con un equipo comercial profesional de vibración BDI 2021 de BridgeTest. En dicha comparación se obtuvo en el peor de los casos un error en la medición de 38,11% de error.

**Palabras claves:** acelerómetro, puente, red, sensor, vibración.

## **Abstract**

In the evaluation of the state of the bridges in our country, one of the tests applied is the vibration analysis procedure, in which it is possible to observe the dynamic behavior of the structure and to take the corresponding measures of maintenance against any damage or failure that may be present in the structure. The e-Bridge project of the School of Engineering in Construction of the Technological Institute of Costa Rica, since 2011 has focused on finding solutions that allow the studies of bridge-type structures with an integrated system in which all the process of analysis of the different physical variables that affect a structure is performed reliably and with measurement systems implemented within the same project. The present project shows the implementation of a network of sensors, using triaxial accelerometers, to know the behavior in the three dimensions of the structures. The communication is made through the star type topology, where there are three auxiliary measurement units and a central processing node for the subsequent storage of the information. The data coming from the secondary nodes are stored in a file on the SD memory card of the microcontroller, for further analysis of vibration in bridge structures. For the verification of the system at the level of laboratory tests, a metallic beam of 6 m in length is used and the results obtained are compared with a commercial professional vibration equipment. The error obtained was 38,31%.

**Keywords:** accelerometer, bridge, network, sensor, vibration.

## **Agradecimientos**

Durante estos años de vida universitaria quiero agradecer a todas las personas que de una u otra forma colaboraron con mi formación profesional.

A mi profesor asesor José Miguel Barboza, siempre con su apoyo, consejos y ayuda en todo momento para que el proyecto de graduación fuese un éxito.

Al ingeniero Yeiner Arias, por su colaboración con su experiencia por la temática del proyecto.

Al profesor Miguel Hernández, con sus consejos y recomendaciones durante todo mi paso por la carrera.

A los ingenieros Giannina Ortiz y Ángel Navarro del proyecto e-Bridge, por toda la colaboración y consejos brindados para el proyecto.

A mis compañeros y compañeras de carrera, agradecerles todo lo que me ayudaron durante la carrera, en especial a Juan José Rojas, Luis Carlos Espinoza, Christian Calvo y Esteban Arias.

A mi hermosa familia, a mis señores padres y hermana, los cuales han sido el motor en mi vida para poder concluir con éxito la carrera universitaria y retribuirles algo de lo mucho que me han brindado con su apoyo y motivación.

# Índice General

Capítulo 1. Introducción.....	14
1.1 Antecedentes.....	14
1.2 Solución Implementada.....	15
1.3 Meta y objetivos .....	17
1.3.1 Meta.....	17
1.3.2 Objetivo general.....	17
1.3.3 Objetivos específicos.....	17
Capítulo 2. Marco Teórico .....	18
2.1 Análisis de Vibración .....	18
2.1.1 Sistemas de un grado de libertad (SDOF) .....	18
2.1.2 Vibración en puentes.....	20
2.2 Acelerómetros .....	20
2.2.1 Acelerómetros MEMS.....	21
2.3 Red inalámbrica de sensores .....	22
2.3.1 Topologías de las redes de sensores inalámbricos .....	22
2.3.2 Estándar de comunicación IEEE 802.15.4 .....	24
2.4 Teorema de muestreo de Nyquist.....	25
2.5 Módulos electrónicos embebidos .....	28
2.5.1 AXDL335 .....	28
2.5.2 Waspote .....	29
2.5.3 Xbee .....	31
2.5.4 Xbee USB Explorer.....	31
2.5.5 Raspberry Pi 3.....	32
2.5.6 Pantalla LCD TFT táctil 3.5”.....	33

Capítulo 3. Proceso metodológico .....	35
3.1 Obtención y análisis de la información.....	35
3.2 Alternativas de solución.....	35
3.3 Implementación de solución .....	37
Capítulo 4. Procedimiento de implementación.....	39
4.1 Diseño del módulo de medición.....	39
4.2 Conexión del módulo de medición al Waspote.....	41
4.3 Configuración de la transmisión de los Xbee .....	42
4.4 Implementación de la red de sensores .....	45
4.4.1 Transmisión y recepción desde el nodo central .....	45
4.4.1.1 Frecuencia de muestreo de la rutina del nodo central .....	46
4.4.1.2 Programación del nodo central .....	47
4.4.2 Rutina de procesamiento de datos del Waspote.....	48
4.4.2.1 Programación de los nodos secundarios.....	49
4.4.3 Interfaz gráfica para usuario .....	50
4.4.3.1 Rutina de programación de la interfaz gráfica.....	51
4.4.4 Alimentación de la red de sensores.....	52
Capítulo 5. Análisis de Resultados .....	53
5.1 Circuito acondicionador de señal de alimentación del módulo de medición.....	53
5.2 Conexión entre nodo central y secundarios .....	54
5.3 Datos almacenados en memoria tipo SD .....	57
5.4 Almacenamiento de los datos de cada nodo secundario en el nodo central .....	59
5.5 Frecuencia de muestreo del sistema. ....	63
5.6 Comprobación de la medición de aceleración del sistema utilizando una viga. ....	66
5.7 Valores máximos de tiempo activo y distancia de la red de sensores. ....	70

5.8 Consumo de potencia del sistema.....	70
Capítulo 6. Conclusiones y recomendaciones .....	72
6.1 Conclusiones .....	72
6.2 Recomendaciones .....	73
Referencias bibliográficas .....	74
Apéndices .....	77
A .1 Manual de usuario.....	77
A.1.1 Conexión de los componentes del sistema .....	77
A.1.2 Programación de los nodos .....	80
A.1.2 Ejecución del programa .....	80
A.1.3 Interpretación de los datos .....	82
A.1.4 Cierre de interfaz y del sistema.....	82
A.1.5 Posibles problemas y soluciones.....	83
A.1.6 Materiales que conforman el sistema .....	83
A.1.7 Control de versiones y contacto .....	84
A .2 Información de la empresa .....	85

# Índice de figuras

FIGURA 1. DIAGRAMA GENERAL DEL SISTEMA.....	15
FIGURA 2. SISTEMA DE UN GRADO DE LIBERTAD CON AMORTIGUADOR [6] .....	19
FIGURA 3. MODO DE VIBRACIÓN EN UNA VIGA CON 2 NODOS [7]. .....	19
FIGURA 4. ESQUEMA GENERAL DE UN ACELERÓMETRO MEMS CAPACITIVO [12]. .....	21
FIGURA 5. TOPOLOGÍA TIPO BUS [15]. .....	22
FIGURA 6. TOPOLOGÍA TIPO ÁRBOL [15]. .....	23
FIGURA 7. TOPOLOGÍA TIPO ESTRELLA [15]. .....	23
FIGURA 8. TOPOLOGÍA TIPO MALLA [15]. .....	24
FIGURA 9. RECONSTRUCCIÓN DE UNA SEÑAL EN EL TIEMPO CONTINUO [18]. .....	26
FIGURA 10. MUESTREO DE SEÑAL EN TIEMPO DISCRETO [18]. .....	27
FIGURA 11. MÓDULO ADXL335 [19] .....	28
FIGURA 12. MÓDULO WASPMOTE [20]. .....	29
FIGURA 13. CONECTORES I/O DEL WASPMOTE [20] .....	30
FIGURA 14. MÓDULO DE XBEE [21] .....	31
FIGURA 15. MÓDULO DE XBEE USB EXPLORER [23] .....	32
FIGURA 16. MÓDULO DE RASPBERRY PI 3 [25] .....	32
FIGURA 17. PANTALLA TÁCTIL LCD TFT 3,5 [26]. .....	34
FIGURA 18. ETAPAS GENERALES DEL SISTEMA.....	37
FIGURA 19. DIVISOR DE TENSIÓN PARA ALIMENTACIÓN DEL ADXL 335. ....	39
FIGURA 20. DIAGRAMA DE CONEXIÓN DEL MÓDULO ADXL335 [19] .....	40
FIGURA 21. PLACA DE CIRCUITO IMPRESO DEL MÓDULO DE MEDICIÓN. ....	40
FIGURA 22. PLACA DE CIRCUITO IMPRESO FINAL DEL MÓDULO DE MEDICIÓN. ....	41
FIGURA 23. INTERFAZ DEL SOFTWARE XCTU.....	43
FIGURA 24. TRAMA DE 64 BITS DEL MODO API [27]. .....	45
FIGURA 25. DIAGRAMA DE FLUJO DE RUTINA DEL NODO CENTRAL.....	47
FIGURA 26. DIAGRAMA DE FLUJO DE RUTINA DE LOS NODOS SECUNDARIOS .....	49
FIGURA 27. VENTANA GENERAL DE LA INTERFAZ GRÁFICA .....	50
FIGURA 28. MENSAJES EMERGENTES DE LA INTERFAZ GRÁFICA.....	51
FIGURA 29. DIAGRAMA DE FLUJO DE LA RUTINA DE LA INTERFAZ GRÁFICA.....	51
FIGURA 30. CONVERTIDOR DE 3,7 A 5 V COMERCIAL POWER BOOST 1000 [30]. .....	52
FIGURA 31. MÓDULO DE MEDICIÓN IMPLEMENTADO .....	53
FIGURA 32. TOPOLOGÍA OBTENIDA CON EL SOFTWARE XCTU.....	55
FIGURA 33. HARDWARE DEL NODO CENTRAL .....	55
FIGURA 34. NÍVEL DE RSSI EN EL MÓDULO DE MEDICIÓN .....	56
FIGURA 35. RECEPCIÓN DE DATOS EN EL NODO CENTRAL CON PYTHON .....	56

FIGURA 36. CREACIÓN DE LOS ARCHIVOS DE TEXTO POR CADA NODO SECUNDARIO EN LA RASPBERRY PI 3....	59
FIGURA 37. MONTAJE DE LA PRUEBA DE VIBRACIÓN EN LA VIGA .....	66
FIGURA 38. SIMULACIÓN DE LA RESPUESTA DE ACELERACIÓN DEL EJE Z DE LA VIGA. ....	67
FIGURA 39. ACELERÓMETRO BDI A2120 .....	67
FIGURA 40. COLOCACIÓN DE LOS SENSORES SOBRE LA VIGA .....	68
FIGURA 41. ACELERACIÓN DEL EJE Z OBTENIDA CON EL BDI A2120 .....	68
FIGURA 42. ACELERACIÓN DEL EJE Z DE LA VIGA CON EL EQUIPO DEL SENSOR ADXL335. ....	69
FIGURA 43. CONEXIÓN DE PINES EN EL WASPMOTE. ....	78
FIGURA 44. CONEXIÓN DE BATERÍA EN EL WASPMOTE. ....	78
FIGURA 45. CONEXIÓN DE LA PANTALLA TÁCTIL A LA RASPBERRY PI 3. ....	79
FIGURA 46. CONEXIÓN AL PUERTO USB DE LA RASPBERRY PI 3. ....	79
FIGURA 47. UBICACIÓN DEL SCRIPT DE PYTHON DE LA RED DE SENSORES. ....	81
FIGURA 48. COMPONENTES UTILIZADOS PARA LA RED DE SENSORES .....	84

## Índice de tablas

TABLA 1. ESPECIFICACIONES TÉCNICAS DEL MÓDULO ADXL335 [19]. ....	28
TABLA 2. ESPECIFICACIONES TÉCNICAS DEL MÓDULO WASPMOTE [20] .....	30
TABLA 3. ESPECIFICACIONES TÉCNICAS DE RASPBERRY PI 3 [25] .....	33
TABLA 4. PINES DE CONEXIÓN EN EL WASPMOTE .....	41
TABLA 5. CONFIGURACIÓN DE LOS XBEE EN EL SOFTWARE XCTU. ....	43
TABLA 6. DATOS ALMACENADOS EN TARJETA MEMORIA SD .....	58
TABLA 7. DATOS DEL ARCHIVO DE TEXTO OBTENIDOS DEL NODO 1 .....	60
TABLA 8. DATOS DEL ARCHIVO DE TEXTO OBTENIDOS DEL NODO 2 .....	61
TABLA 9. DATOS DEL ARCHIVO DE TEXTO OBTENIDOS DEL NODO 3. ....	62
TABLA 10. COMPORTAMIENTO DE CANTIDAD DE DATOS SEGÚN LA FRECUENCIA Y NÚMERO DE NODOS .....	64
TABLA 11. COMPARACIÓN DE LOS VALORES MÍNIMOS Y MÁXIMOS DE LA PRUEBA. ....	69
TABLA 12. RESUMEN DEL CONSUMO DE POTENCIA DEL SISTEMA. ....	71

# Capítulo 1. Introducción

## 1.1 Antecedentes

El proyecto e-Bridge está conformado por un grupo multidisciplinario de investigación del Instituto Tecnológico de Costa Rica que trabaja en la predicción de fallas en puentes a través del monitoreo de variables en estas estructuras. Una de estas variables importantes de monitoreo es la vibración de la estructura al estar presente una perturbación mecánica (un vehículo, un sismo, etc.). Estas mediciones permiten realizar estudios de la aceleración con que una onda de vibración mecánica viaja sobre la estructura y detectar puntos de fatiga que podrían condicionar a una falla en el puente.

Los encargados del proyecto e-Bridge, en el momento que realizan las mediciones de vibración con los equipos existentes, solamente pueden hacerlo en un solo eje del plano cartesiano a la vez (x, y o z), debido a que los acelerómetros de estos equipos están diseñados para realizarlo de dicha forma. Por lo tanto, cuando es necesario realizar mediciones de vibración en los tres cartesianos, se utilizan tres sensores que deben ser colocados en una plantilla especial que permite la ubicación de cada uno de estos según la dirección 'x', 'y' y 'z'. Otra desventaja del equipo de medición actual es que solo permite incorporar sensores para otro tipo de mediciones que sean de la misma marca del sistema, siendo estos de un costo muy elevado.

En función de lo anterior, el proyecto e-Bridge ha trabajado en colaboración con la Escuela de Ingeniería Electrónica para el diseño e implementación de sistemas alternativos para realizar monitoreo en estructuras de puentes con características deseadas. Durante el año 2007, el Ing. Adolfo Chaves fabricó un prototipo de un nodo de adquisición de datos, el cual fue el preámbulo para que en el año 2009, naciera el proyecto denominado CRTecMote, liderado por el profesor Johan Carvajal, el cual consistía en el desarrollo de un sistema de adquisición de datos ambientales para el uso de investigación científica, implementado en una arquitectura abierta y con un bajo consumo de potencia [1].

A partir de ese antecedente, nace la colaboración hacía el proyecto e-Bridge por parte de la Escuela de Ingeniería Electrónica con la implementación de un sistema de monitoreo en puentes, en donde se realizaron algunos prototipos. Entre ellos, el Prof. Francisco Navarro utilizó sensores de fibra óptica para medir la vibración y deformación en el antiguo puente del Río Virilla de la Ruta 1 [2]. Además, como parte de su tesis de maestría el Prof. Yeiner Arias, en el año 2015, desarrolló un sistema embebido en tiempo real del análisis de vibración [3].

## 1.2 Solución Implementada

Con el fin de implementar un sistema de medición para vibración en puentes se utilizó un sistema embebido que permite recolectar la información proveniente de los sensores y crear nodos de transmisión de datos a un nodo central, con el fin de desarrollar una red de comunicación de datos con topología tipo estrella. El nodo central se comunica con los nodos auxiliares para proceder al almacenamiento de los datos de medición respectivos de cada nodo, como se muestra en la Figura 1. Utilizando acelerómetros ADXL335 de tecnología MEMS (sistemas micro electromecánicos) que permiten realizar mediciones de vibración en los tres ejes coordenados de forma simultánea se definió el sistema de monitoreo requerido.

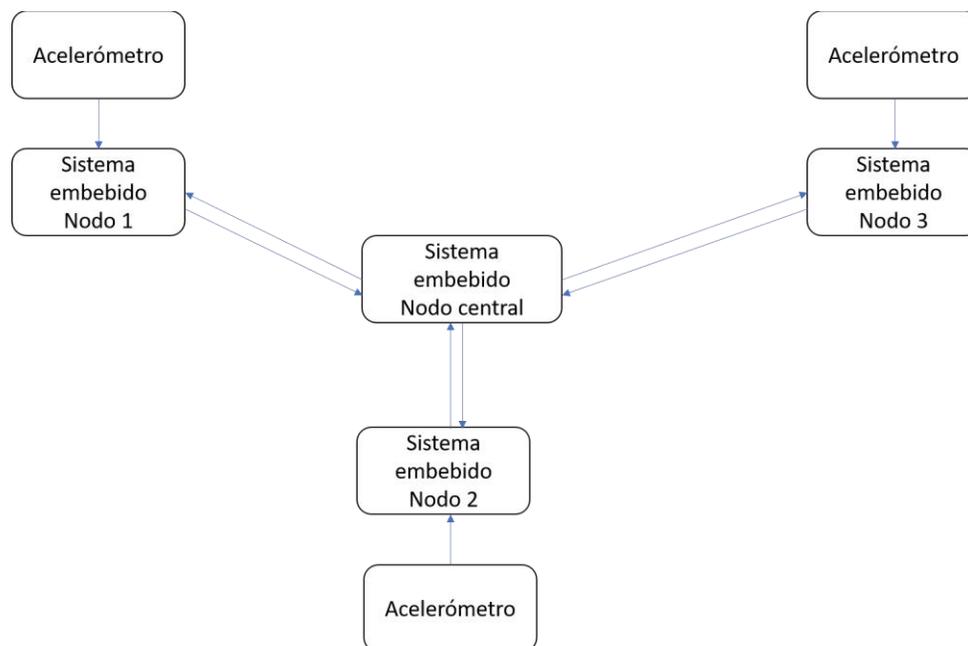


Figura 1. Diagrama general del sistema.

Para la implementación de los nodos de medición se utilizaron los sistemas embebidos Waspote del fabricante español Libelium, utilizando su versión v1.1. Los cuales permiten hacer redes de sensores para utilizarlos en múltiples aplicaciones. En la comunicación inalámbrica se utilizó el protocolo IEEE 802.15.4, el cual ya viene incorporado con los módulos Waspote, específicamente con un módulo Xbee para funcionar a una frecuencia libre de 2.4 GHz. El nodo central se desarrolló con una tarjeta de desarrollo Raspberry Pi 3, el cual ofrece múltiples ventajas, dentro de las cuales posee una conexión para una pantalla táctil que se utilizó para generar interacción con el usuario. Este módulo también cuenta con la posibilidad de almacenamiento en una memoria tipo SD, misma que se dispuso para guardar todos los datos de medición. Durante la medición los datos obtenidos de magnitud de aceleración son almacenados en el nodo central. Cada medición se compone de tres valores de aceleración definidos para cada uno de los ejes cartesianos. De esta forma, toda esta información permitirá realizar análisis posteriores del estado estructural de los puentes, objetivo deseado por el proyecto e-Bridge.

Para ejemplificar el funcionamiento del sistema, se realizaron mediciones de aceleración en una viga de 3 metros con un grado de libertad, colocando uno de los nodos secundarios en el punto central de la pieza. Estas mediciones experimentales se utilizaron para comparar la capacidad del equipo con el sistema actual de medición con el que cuenta e-Bridge y poder definir el error de medición que este presenta.

## **1.3 Meta y objetivos**

### **1.3.1 Meta**

Desarrollar un sistema capaz de obtener información de vibración en puentes para el análisis del estado estructural de los mismos.

### **1.3.2 Objetivo general**

Implementar una red inalámbrica de sensores para el análisis de vibración de puentes utilizando acelerómetros triaxiales.

### **1.3.3 Objetivos específicos**

1. Diseñar un nodo de medición capaz de medir vibración de forma triaxial.
2. Implementar tres nodos de medición con la capacidad de comunicar los datos a un nodo central.
3. Implementar la topología de comunicación entre los nodos de medición y el nodo central.
4. Almacenar los resultados de cada medición en una memoria SD desde el nodo central.

## Capítulo 2. Marco Teórico

### 2.1 Análisis de Vibración

Uno de los datos más importantes en la evaluación de las estructuras es la intensidad de vibración en diferentes modos. La clasificación de esa vibración se da en valores críticos o no críticos que pueden indicar si la estructura puede estar sujeta a sufrir un daño de manera local o global debido a la fatiga o a problemas relacionados con los materiales de la construcción [4].

Como resultado del daño, habría una reducción en rigidez y una disminución de la energía libre almacenada en el cuerpo, producto de la respuesta dinámica, la cual se rige por los parámetros del sistema, como la masa y la aceleración. El cambio en estos parámetros conduciría a modificaciones en la respuesta vibratoria debido a que se caracteriza por los parámetros modales (frecuencias naturales, formas de modo y valores de amortiguación modal). Debido a que cada modo de vibración tiene una distribución de energía diferente dependiendo de la ubicación y del tipo de daño que tenga la estructura [5].

#### 2.1.1 Sistemas de un grado de libertad (SDOF)

El comportamiento dinámico de varios tipos de estructuras como los puentes, se puede representar de una forma simplificada por un sistema de un grado de libertad (SDOF) [3], como por ejemplo el de la Figura 2. La ecuación de movimiento que describe el cuerpo está definida por:

$$m \frac{d^2x(t)}{dt^2} + c \frac{dx(t)}{dt} + kx(t) = h(t) \quad (1)$$

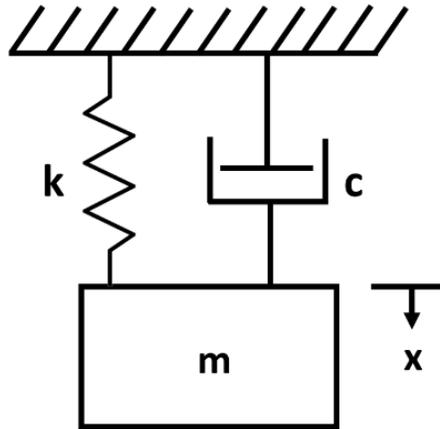


Figura 2. Sistema de un grado de libertad con amortiguador [6]

Donde  $x(t)$  es el desplazamiento,  $m$  la masa,  $c$  la constante de amortiguamiento,  $k$  la constante elástica y  $h(t)$  representa las fuerzas externas del sistema [3]. Recordando que la aceleración es la segunda derivada del desplazamiento, considerando las fuerzas externas del sistema  $h(t)$  como la señal de entrada y la aceleración como la señal de salida se tiene que función de transferencia del sistema se define por:

$$\frac{A(s)}{H(s)} = \frac{s^2}{ms^2 + cs + k} \quad (2)$$

La ecuación (2) permite estudiar el comportamiento vibratorio de un sistema considerando un grado de libertad en el análisis. Un ejemplo de un sistema de este tipo estaría dado por una viga sólida. En este caso, al ser un sistema de un grado de libertad, solamente tendrá un modo normal de vibración, como se ilustra en la Figura 3.



Figura 3. Modo de vibración en una viga con 2 nodos [7].

### 2.1.2 Vibración en puentes

Para determinar la respuesta vibratoria de las estructuras tipo puente en la actualidad se implementan redes de monitoreo, utilizando sensores para obtener los parámetros de aceleración y carga a la cual está expuesta la estructura. En particular, las redes de sensores de aceleración su colocación dependerá de la longitud del puente y de la posición de las bases del mismo, esto a que las zonas de ubicación de los nodos de medición deben ser en los puntos de mayor vibración del puente, en donde se tiene la mayor frecuencia natural del sistema, es decir en los segmentos de la estructura que se encuentran entre las bases.

### 2.2 Acelerómetros

Para medir las intensidades de vibración se utilizan acelerómetros, los cuales son dispositivos que convierten la aceleración de gravedad debido al movimiento, en una señal de tensión proporcional a la aceleración del sistema. Miden la aceleración en unidades de “g”, donde g, es la aceleración debida a la gravedad de la tierra sobre un objeto [8]. Algunas de las características principales de un acelerómetro son: sensibilidad, rango dinámico, masa, respuesta en frecuencia, ruido y rango de temperatura. Existen varios tipos dependiendo de la tecnología de fabricación y funcionamiento, de los cuales se pueden mencionar los piezoeléctricos, resistivos, capacitivos, microelectromecánicos (MEMS), entre otros [9]. Su funcionamiento se basa en la ley de Hooke y la segunda ley de Newton, donde la aceleración genera una fuerza sobre una masa (m), que al desplazarse deforma un resorte (k, constante elástica del resorte) con lo que tenemos la expresión de la ecuación 3 [10].

$$F = ma = kx \quad (3)$$

Lo que produce que la masa se desplace o retrocede debido a la aceleración “a”, si se observa un desplazamiento “x”, esto solo para un eje del plano cartesiano. Si se supone que la masa está bajo la acción de una aceleración, obtenemos la ecuación 4, que muestra la expresión general de la aceleración a la cual está expuesta el cuerpo en movimiento [10].

$$a = \frac{kx}{m} \quad (4)$$

### 2.2.1 Acelerómetros MEMS

El término MEMS proviene del término sistema microelectromecánico, compuesto por sistemas mecánicos y electrónicos que se caracterizan por un tamaño en la escala de los micrómetros. Estos sistemas están conformados por un agente mecánico que funciona como transductor, el cual reacciona ante una variación de una magnitud física. El sistema electrónico se encarga de tomar de forma eléctrica la medición del sensor y traducir esta información a un sistema que tome dichos datos para un fin concreto en cada aplicación [11]. La principal ventaja de los acelerómetros tipo MEMS es el proceso de fabricación, debido a las grandes cantidades de lotes que se pueden producir utilizando la tecnología dispuesta para la fabricación de circuitos integrados, que permite tener un menor costo económico que utilizando otro tipo de sensores [8].

Dentro de los acelerómetros MEMS existen los que son capacitivos micro mecanizados, los cuales en la parte central del acelerómetro se realiza una prueba micro mecánica suspendida, que actúa como elemento sensor, tal y como se observa en la Figura 4. Cuando se aplica una aceleración externa, la masa de prueba se desplazará con respecto al marco de referencia en movimiento. La aceleración se deduce del desplazamiento de la masa de prueba que se puede medir de varias maneras. Para el enfoque de detección capacitiva, el desplazamiento se detecta midiendo el cambio de capacitancia entre la masa de prueba acumulada y los electrodos fijos adyacentes. La baja capacitancia parásita lograda a partir de la integración monolítica es la clave para maximizar el rendimiento con esta técnica [12].

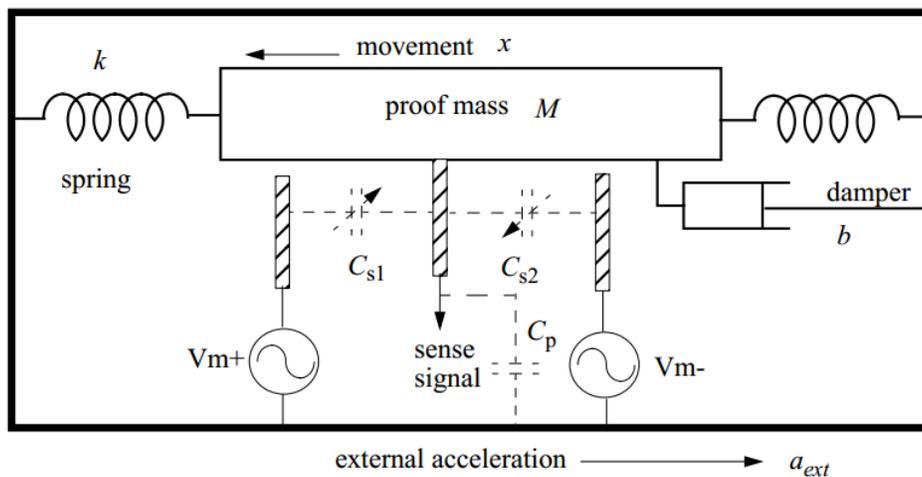


Figura 4. Esquema general de un acelerómetro MEMS capacitivo [12].

## 2.3 Red inalámbrica de sensores

Una red de sensores inalámbricos (WSN, por sus siglas en inglés) es una red compuesta por dispositivos autónomos, distribuidos espacialmente que utilizan sensores para monitorear condiciones físicas o ambientales [13]. Están conformadas por un nodo sumidero o gateway, el cual permite conectarse con los nodos secundarios según sea el tipo de topología, para obtener la información de cada uno y así procesar los datos que se requieren [14].

Las redes inalámbricas pueden tener una gama de aplicaciones muy amplia por lo que una buena especificación de los requerimientos de diseño garantizará un desempeño adecuado de la WSN para el entorno en el cual se quiere operar [1]. Gracias a la innovación de las nuevas tecnologías de fabricación de componentes electrónicos, como por ejemplo los tipo MEMS, existen redes de monitoreo en el área de salud con la valoración de pacientes, en infraestructura como en carreteras y puentes, a nivel de condiciones ambientales del agua y suelo, a nivel industrial para el control de activos y en muchas otras aplicaciones que se utilizan en la actualidad [14].

### 2.3.1 Topologías de las redes de sensores inalámbricos

Dentro de las topologías para implementar una red de sensores, podemos encontrar las siguientes:

**Tipo Bus:** Consiste en que un nodo de la red, envía un mensaje a otro nodo en específico, a pesar de que este es visto por todos los demás nodos que conforman la red, pero solo el destinatario previsto realmente acepta y procesa el mensaje [15].

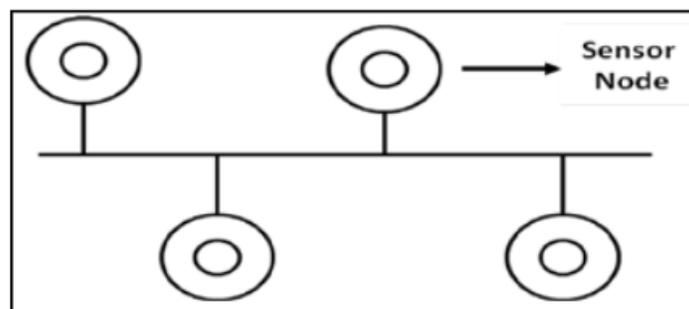


Figura 5. Topología tipo Bus [15].

**Tipo Árbol (Tree):** Formada por un concentrador central llamado nodo raíz como principal enrutador de comunicación. En la jerarquía, el hub central es un nivel por debajo del nodo principal o raíz. Este nivel inferior forma una red de tipo estrella con los nodos secundarios [15].

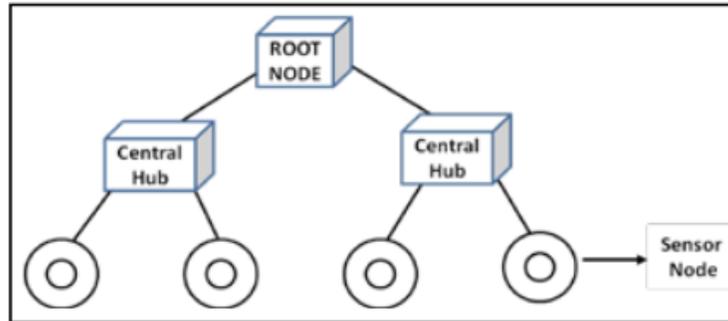


Figura 6. Topología tipo Árbol [15].

**Tipo Estrella:** En esta topología los nodos secundarios se conectan solamente al nodo sumidero y no tienen comunicación entre los otros nodos secundarios. Por lo que se puede decir que cada nodo secundario es un cliente y el nodo principal o sumidero es el servidor [15].

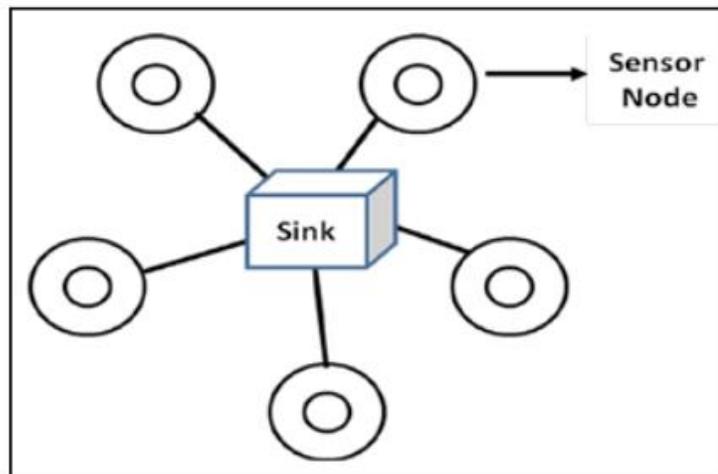


Figura 7. Topología tipo Estrella [15]

**Tipo Malla (Mesh):** esta topología implica que el mensaje puede tomar varias rutas desde el punto de origen al destino. En esa forma de conexión cada nodo se puede conectar a todos los demás nodos, en este caso se habla de una malla completa y hay redes de malla parcial en las que algunos nodos se conectan solamente de forma indirecta a otros [15].

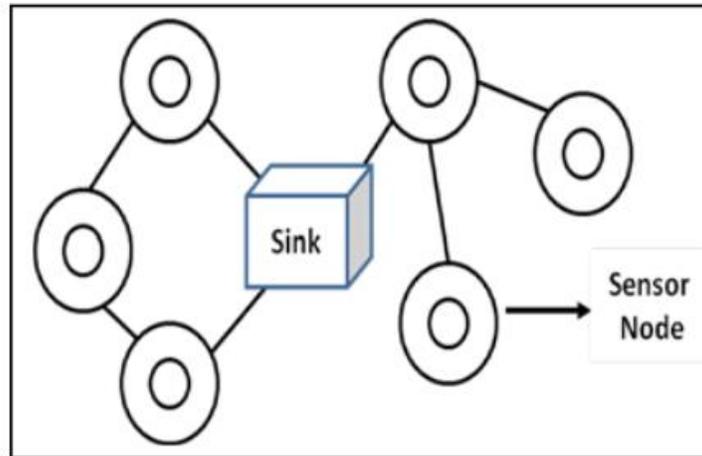


Figura 8. Topología tipo Malla [15]

### 2.3.2 Estándar de comunicación IEEE 802.15.4

Uno de los estándares de comunicación más utilizados en las redes inalámbricas de sensores es el IEE 802.15.4, debido que soporta múltiples topologías para su conexión en red, entre ellas la topología tipo estrella y la topología tipo malla. La topología por escoger es una elección de diseño y va a estar dado por la aplicación a la que se desee orientar.

Además es un estándar con baja tasa de transmisión de datos en redes inalámbricas de uso personal (LR WPAN, se su siglas en inglés) [16]. Es caracterizado por mantener un alto nivel de simplicidad, permitiendo implementaciones de bajo costo y baja potencia. Su banda de frecuencia incluye la banda industrial, científica y médica de 2.4GHz proporcionando una disponibilidad casi mundial. Además, esta banda es también utilizada por otros estándares inalámbricos IEEE 802 [17].

## 2.4 Teorema de muestreo de Nyquist

Una señal analógica se puede representar como se muestra en la ecuación 5, en una suma de sinusoides de diferentes amplitudes, frecuencias y fase, donde  $N$  es el número de componentes de frecuencia. Si se muestrea a una frecuencia de  $f_s = 1/T$ , la frecuencia más alta de una señal analógica será  $\frac{f_s}{2}$  [18].

$$x(t) = \sum_{i=1}^N A_i \cos(2\pi f_i t + \theta_i) \quad (5)$$

Las frecuencias por encima de  $\frac{f_s}{2}$  o por debajo de  $\frac{-f_s}{2}$  producen muestras idénticas en el rango de  $-\frac{f_s}{2} \leq f \leq \frac{f_s}{2}$ . Para evitar este problema de distorsión denominado como aliasing, se debe de elegir una frecuencia de muestreo suficientemente alta, por lo que selecciona:

$$f_s \geq 2f_{MAX} \quad (6)$$

Donde  $f_{MAX}$  la frecuencia más alta contenida en  $x(t)$ . Si  $f_{MAX} = B$ , por lo tanto  $f_s \geq 2B$ , entonces  $x(t)$  se puede recuperar de forma exacta utilizando la siguiente función de interpolación:

$$g(t) = \frac{\text{sen}2Bt}{2\pi Bt} \quad (7)$$

Posteriormente  $x(t)$  se puede expresar como:

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{f_s}\right) g\left(t - \frac{n}{f_s}\right) \quad (8)$$

Donde  $x\left(\frac{n}{f_s}\right) = x(nT) \equiv x(n)$  son las muestras de  $x(t)$  cada  $T$  segundos para generar una señal discreta en el tiempo, como se muestra en la Figura 9. La fórmula de reconstrucción de la señal cuando se realiza a la frecuencia mínima de muestreo  $f_s = 2B$ , esta dada por:

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2B}\right) \frac{\text{sen}2\pi B(t-n/2B)}{2\pi B(t-n/2B)} \quad (9)$$

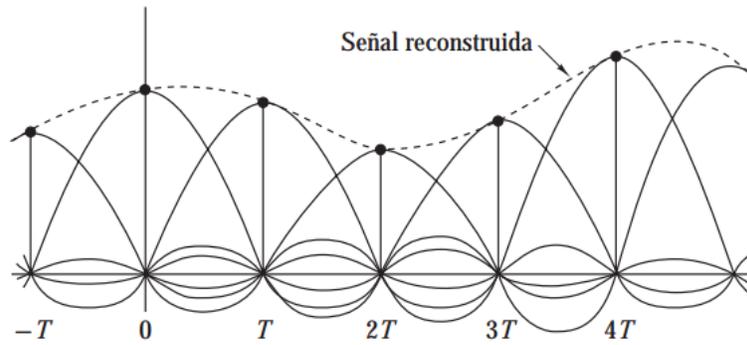


Figura 9. Reconstrucción de una señal en el tiempo continuo [18]

Para el dominio de la frecuencia, si la transformada de Fourier de una señal aperiódica  $x_a(t)$  es:

$$X_a(F) = \int_{-\infty}^{\infty} x_a(t) e^{-j2\pi Ft} dt \quad (10)$$

Y su transformada inversa de Fourier está dada por:

$$x_a(t) = \int_{-\infty}^{\infty} X_a(t) e^{j2\pi Ft} dF \quad (11)$$

El espectro de una señal discreta en el tiempo  $x(n)$ , muestreando  $x_a(t)$ , está dada por la transformada de Fourier de la ecuación 12 y su inversa por la ecuación 13.

$$X(\omega) = \sum_{-\infty}^{\infty} x(n) e^{-j\omega n} \quad (12)$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega \quad (13)$$

Como relación entre las señales, analógica  $x_a(t)$  y la discreta en el tiempo  $x(n)$ , tenemos lo siguiente:

$$t = nT = \frac{n}{T_s} \quad (14)$$

Si se sustituye la expresión 14 en la 11 y se compara con la 13, obtenemos la expresión 15, en donde se observa la relación entre la frecuencia de la señal analógica y discreta como  $f = F/F_s$ .

$$\int_{-1/2}^{1/2} X(f) e^{j2\pi f n} df = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi n F / F_s} dF \quad (15)$$

Posteriormente se realiza un cambio de variable y se obtiene:

$$\frac{1}{F_s} \int_{-F_s/2}^{F_s/2} X(F) e^{j2\pi n F / F_s} dF = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi n F / F_s} dF \quad (16)$$

De la expresión anterior, el lado derecho, su rango de integración se puede dividir en un número infinito de intervalos de anchura  $F_s$  y utilizando relaciones matemáticas y la exponencial de periodicidad compleja, se puede concluir que:

$$X(F) = F_s \sum_{k=-\infty}^{\infty} X_a(F - k F_s) \quad (17)$$

Si suponemos que en la gráfica de la Figura 10, la frecuencia de muestreo  $F_s$  es mayor de  $2B$  (frecuencia de Nyquist) entonces  $X(F) = F_s X_a(F)$ , con  $|F| \leq F_s/2$ , se puede realizar la reconstrucción de la señal analógica a partir de las muestras de  $x(n)$  [18].

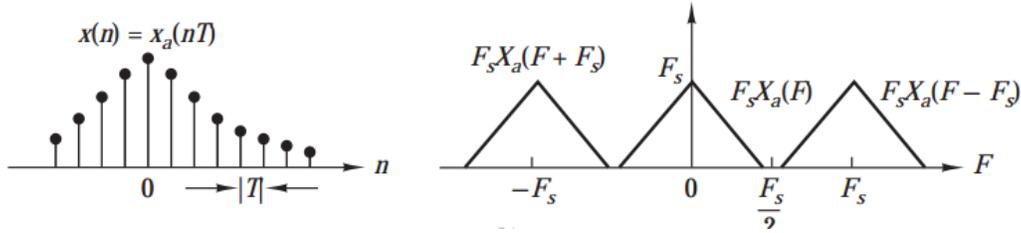


Figura 10. Muestreo de señal en tiempo discreto [18]

Donde  $X_a(F)$  está dado por:

$$X_a(F) = \begin{cases} \frac{1}{F_s} (F), & |F| \leq F_s/2 \\ 0 & |F| > F_s/2 \end{cases} \quad (18)$$

Utilizando la expresión 11, se obtiene la transformada inversa de  $X_a(F)$  como:

$$x_a(t) = \int_{-F_s/2}^{F_s/2} \left[ \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi F n / F_s} \right] e^{j2\pi F t} dF = \sum_{n=-\infty}^{\infty} x_a(nT) \frac{\text{sen}(\pi/T)(t-nT)}{(\pi/T)(t-nT)} \quad (19)$$

Por lo que al comparar la expresión 9 y 19, se puede observar que se reconstruye la señal  $x_a(t)$ , por medio del teorema del muestreo aplicado en el tiempo y la frecuencia.

## 2.5 Módulos electrónicos embebidos

### 2.5.1 AXDL335

El módulo ADXL335 es un acelerómetro triaxial de bajo consumo de potencia, con salidas de tensión acondicionadas. Su rango de operación es de  $\pm 3g$  y puede ser utilizado para aplicaciones de vibración. Funciona con la tecnología de detección de la aceleración capacitiva micro mecanizada, misma que se explicó en la sección 2.2.1 [19].

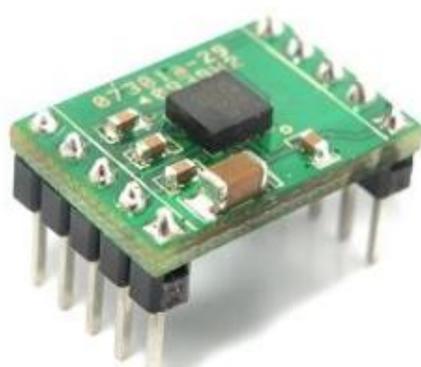


Figura 11. Módulo ADXL335 [19]

Tabla 1. Especificaciones técnicas del módulo ADXL335 [19]

ESPECIFICACIONES	CARACTERÍSTICAS
Rango de tensión de operación	1,8- 3,6 V
Ancho de banda	1600 Hz (eje X y Y), 550 Hz (eje Z)
Sensibilidad típica	300 mV/g
Consumo de corriente	350 $\mu$ A
Temperatura de operación	-40 a +85 °C

### 2.5.2 Waspote

Los módulos Waspote son desarrollados por la empresa española Libelium, la cual es una plataforma Open Source para crear redes inalámbricas de sensores con un bajo consumo de potencia. Se basa en arquitectura modular en la cual se le pueden adaptar múltiples módulos para medir diferentes variables físicas de acuerdo con las necesidades del proyecto. Para su programación utiliza su propio software, el cual es basado en el ambiente de desarrollo de la distribución Arduino, utilizando bibliotecas implementadas en el lenguaje de alto nivel C++ [20].



Figura 12. Módulo Waspote [20]

Entre los módulos principales que se le pueden acoplar están: GPS, Xbee, Bluetooth, GSM/GPRS, RFID, memoria SD y tarjetas de sensores de gases, radiación, entre otros. Además, cuenta con alimentación independiente por medio de una batería de 3.7 V, misma que se puede recargar con el mismo módulo, utilizando una conexión a través de un cable USB en un computador o un adaptador de alimentación de pared [20]. Dentro de sus características principales están las siguientes:

Tabla 2. Especificaciones técnicas del módulo Wasmote [20]

Especificaciones	Características
<b>Microcontrolador</b>	ATmega1281
<b>Frecuencia</b>	8 MHz
<b>Memoria SRAM</b>	8 KB
<b>Memoria EEPROM</b>	4 KB
<b>Memoria FLASH</b>	128 KB
<b>Tarjeta SD</b>	2 GB
<b>Peso</b>	20 g
<b>Temperatura de operación</b>	[-20°C, +65°C]
<b>Reloj</b>	RTC (32KHz)

En la siguiente figura se muestra el módulo Wasmote con sus respectivos tipos de conectores según los módulos que se desean utilizar, posee 7 pines para entradas analógicas con un ADC de 10 bits de resolución, 8 pines de entradas/salidas digitales, reloj en tiempo real (RTC), además de un bus de comunicación I2C, puertos serial y SPI. Posee 4 modos de funcionamiento, con sus valores de consumo de corriente: ON: 9mA, Sleep: 62µA, Deep Sleep: 62µA, Hibernate: 0.7µA.

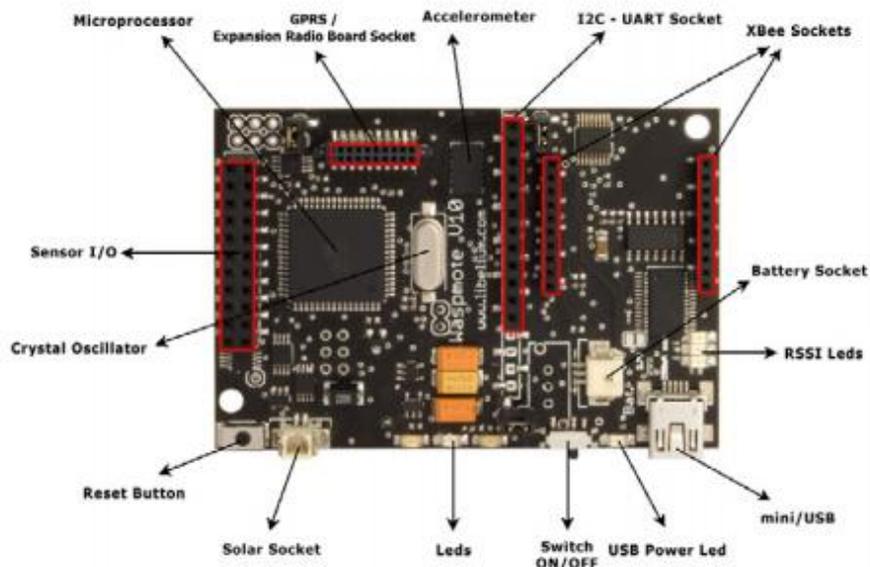


Figura 13. Conectores I/O del Wasmote [20]

### 2.5.3 Xbee

Los módulos Xbee son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan el protocolo de red llamado IEEE 802.15.4 para crear redes punto a multipunto o punto a punto. Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible [21]. Funcionan a una frecuencia de 2,4 GHz, para 16 canales en total con un ancho de banda por canal de 5 MHz. Estos módulos al establecer una topología de punto a punto con otros módulos se identifican por medio de la dirección MAC (del inglés: Media Access Control), la cual es única para cada dispositivo en la red a la hora de identificarlo [22].

Los Xbee son uno de los módulos que se acoplan a las tarjetas de desarrollo de Waspote, facilitando la implementación de comunicación entre los diferentes nodos de las redes de sensores inalámbricas que se desarrollan con estos componentes.



Figura 14. Módulo de Xbee [21]

### 2.5.4 Xbee USB Explorer

Para poder realizar la comunicación entre los diferentes módulos de una red de sensores, utilizando como medio de comunicación los Xbee, hacia un computador se requiere de un Xbee USB Explorer. El Xbee USB Explorer es una unidad base fácil de usar, la cual permite la conexión de USB a serie para la línea de los módulos Digi XBee. Lo más destacado de este dispositivo es su convertidor FT231X de USB a serie, el cual funciona para la

interpretación de la información. Por otro lado, posee un botón de reinicio y un regulador de voltaje para suministrar al XBee la alimentación necesaria para su funcionamiento. Además, se compone de cuatro LEDs que pueden usarse para depurar el XBee con las siguientes funciones: RX, TX, RSSI (indicador de intensidad de señal) e indicador de encendido [23].



Figura 15. Módulo de Xbee USB Explorer [23]

### 2.5.5 Raspberry Pi 3

La Raspberry Pi 3 es un computador del tamaño de una tarjeta de crédito es capaz de muchas de las tareas que puede ejecutar un computador personal. Puede utilizar hojas de cálculo, procesamiento de textos, reproducción de video y juegos de alta definición. Puede ejecutar varias versiones de Linux y se utiliza en muchas partes alrededor del mundo en educación de escuelas, colegios y universidades por su plataforma Open Source y de alto desempeño en la realización de múltiples tareas con la utilización de múltiples lenguajes de programación [24].



Figura 16. Módulo de Raspberry Pi 3 [25]

Tabla 3. Especificaciones técnicas de Raspberry Pi 3 [25]

<b>Especificaciones</b>	<b>Características</b>
<b>Microcontrolador</b>	Broadcom BCM2837 ARM Cortex-A53
<b>Frecuencia</b>	1.2 GHz
<b>Memoria RAM</b>	1 GB
<b>Memoria EEPROM</b>	4 KB
<b>Memoria FLASH</b>	128 KB
<b>Almacenamiento</b>	Memoria SD
<b>Alimentación</b>	5 V/2.4 A micro USB

Los cuatro puertos USB incorporados de la Raspberry Pi 3 ofrecen conectividad suficiente para un mouse, un teclado o cualquier otro dispositivo por medio de sus conectores tipo HDMI, DSI o CSI que es para acoplarle su propia cámara. Además, ofrece conexión vía Wifi con el integrado BCM43143, Bluetooth de bajo consumo, puerto Ethernet 1/100 y periféricos como 27 GPIO, UART, bus I2C y SPI [25].

### 2.5.6 Pantalla LCD TFT táctil 3.5"

Como parte de los dispositivos adaptables en la Raspberry Pi, se puede utilizar una pantalla táctil de 3,5" que se conecta a través del protocolo SPI a través de los GPIO de la tarjeta de desarrollo. Tiene una resolución de 320x480.

Tabla 4. Especificaciones técnicas de Pantalla LCD TFT 3,5" [26]

<b>Especificaciones</b>	<b>Características</b>
<b>Tipo LCD</b>	TFT
<b>Tipo de interfaz</b>	SPI
<b>Tipo de pantalla táctil</b>	Resistiva
<b>Controlador</b>	XPT2046
<b>Cantidad de colores</b>	65536



Figura 17. Pantalla táctil LCD TFT 3,5 [26]

## **Capítulo 3. Proceso metodológico**

### **3.1 Obtención y análisis de la información**

En la recolección de información para la elaboración del proyecto, se consultaron referencias bibliográficas especializadas en el monitoreo de puentes considerando los diferentes parámetros que se deben analizar para generar un estudio estructural del estado de éstos a partir de la medición de vibración presente. Además, se realizó una búsqueda de equipo electrónico que puede ser utilizado para la implementación de la red de sensores para la toma de datos requeridos.

Para comprender la teoría del análisis de vibraciones se consulta a los profesores encargados del proyecto e-Bridge, sobre las valoraciones técnicas que realizan en el área de las estructuras de acuerdo con los datos obtenidos, realizando pruebas con los equipos de vibración que ellos cuentan actualmente para comprender el objetivo que tiene que cumplir el sistema en diseño. Posteriormente, se consideró los alcances y limitaciones que presenta el equipo para medición de vibraciones con el que cuenta e-Bridge con el fin de detectar las posibles mejoras de funciones y características que se pueden dar en referencia a las nuevas tecnologías que se pueden utilizar, valorando el costo económico. También se debe considerar que el nuevo sistema tenga la capacidad de adaptar otros sensores de medición para realizar monitoreo de otras variables físicas en las estructuras, como por ejemplo deformación.

### **3.2 Alternativas de solución**

En conjunto con la coordinación del proyecto e-Bridge se propone utilizar acelerómetros triaxiales de alta y baja precisión, con la finalidad de hacer una comparación respecto al comportamiento del sistema para la medición requerida, así como en el precio de adquisición de ambas propuestas. Con respecto a los sistemas electrónicos necesarios para realizar la captura y procesamiento de los datos se propone diferentes sistemas de desarrollo embebido disponibles en el mercado, así como de los módulos de transmisión de datos para realizar la transmisión de información.

Luego de investigar se descarta utilizar un acelerómetro de alta precisión debido al costo elevado que este presenta, por lo que se sugiere trabajar con módulos de un precio accesible

y de fácil adquisición como lo son los ADXL335, lo anterior considerando el límite económico que dispone e-Bridge para este sistema. En el caso de los sistemas electrónicos embebidos, e-Bridge define que deben ser utilizados los Wasmote, ya que ellos ya han adquirido estos previamente y se desea hacer uso de ellos para la implementación de este sistema. Por tanto se decide utilizar el Wasmote para implementar los nodos de medición. Por otro lado, los Wasmote incluyen como parte de su implementación, dispositivos Xbee para la transmisión de los datos de forma inalámbrica, quedando definido por tal razón el uso de Xbee para dicha comunicación. Para el nodo central, luego de buscar diferentes alternativas, valorando especialmente la facilidad de manejo para el usuario y la adaptación de diferentes plataformas de desarrollo, se elige el microcontrolador Raspberry Pi 3, el cual permite utilizar una amplia gama de aplicaciones muy completas para una posterior ampliación del proyecto.

Como se observa en la Figura 18, el sistema consta de 5 etapas, en la primera parte se tiene cada nodo con un acelerómetro triaxial, el cual luego se conecta al módulo Wasmote, para que se inicie con el proceso de envío de los datos al nodo central por medio de los módulos Xbee de forma inalámbrica. Posteriormente la información se recibe en el sistema embebido por medio de un puerto USB y este realiza el procesamiento de los datos para su almacenamiento en la memoria de tipo SD.

La manipulación del sistema de medición por parte del usuario se realiza por medio de una pantalla táctil que se acopla a la Raspberry Pi, con el fin de que el usuario inicie el sistema de la toma de datos de los tres nodos simultáneamente. Al final de todo el proceso se tiene un archivo con los diferentes datos de medición procedentes de cada nodo auxiliar.

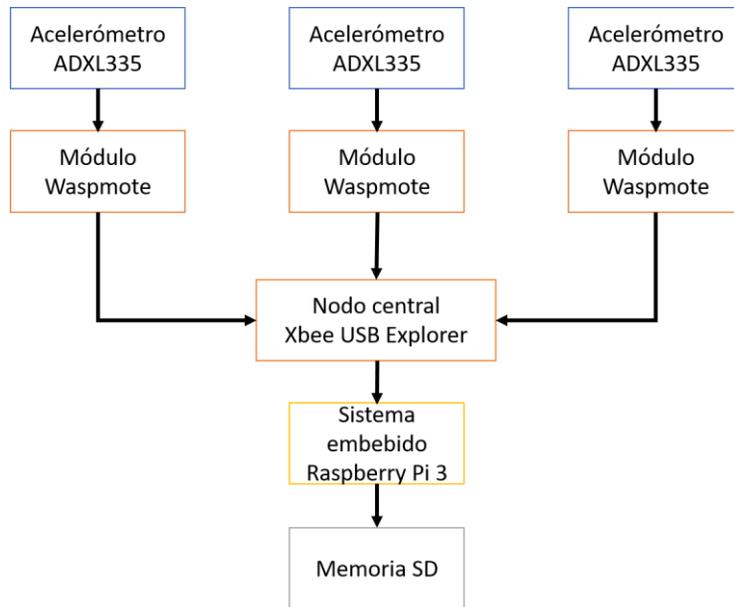


Figura 18. Etapas generales del sistema

### 3.3 Implementación de solución

En general las características del sistema a implementar son: la adquisición de los datos por medio de tres nodos secundarios, envío de esas mediciones al nodo central y por último realizar el almacenamiento de la información en el nodo principal.

Para la adquisición de los datos, primero se comprende la forma del funcionamiento del acelerómetro ADXL335 con el objetivo de conocer si se debía realizar algún procedimiento previo antes de su implementación en los nodos secundarios. Se revisa la hoja de datos del módulo y se obtiene que este opera a una tensión promedio de 3,3 V, lo cual al revisar las características técnicas de los pines del Wasp mote, solamente entrega una tensión de 5V, por lo tanto, se debía hacer un pequeño acondicionamiento en la alimentación del acelerómetro para su correcto funcionamiento.

Posteriormente luego de realizar esa adaptación, se conecta el módulo del acelerómetro al Wasp mote y se obtienen los valores de aceleración en los tres ejes. Aun así, debe realizarse una conversión a los valores obtenidos de los pines analógicos de acuerdo con la relación del procesamiento del ADC de la placa de desarrollo, para acceder a los datos reales de la aceleración. Por último, se incluye dentro del dato para el envío, como una característica

importante que brinda el Waspote la información del reloj en tiempo real de la toma de la medición.

La transmisión de los datos desde el primer nodo secundario al central se realiza por medio del módulo Xbee, utilizando la librería habilitada dentro del sistema de programación del Waspote, en donde las mediciones se empaquetan en una trama de acuerdo con el tipo de parámetros que estén configurados en la transmisión.

Para la recepción de los datos en el sistema embebido del nodo central se utiliza el lenguaje de programación Python, el cual por su versatilidad y múltiples funciones de desarrollo permite la captura y procesamiento de los datos. Las tramas enviadas son recibidas por un módulo Xbee, el cual se encuentra en un adaptador para USB en donde se realiza la comunicación a través del puerto serie del nodo central.

La solicitud del envío de los datos al nodo central se realiza utilizando Python, por medio del envío de un dato a la dirección específica que tiene cada módulo Xbee para identificarse dentro de una red. Posteriormente de ese procedimiento, los datos se reciben y son acomodados de acuerdo con lo que se requiere, utilizando como una etiqueta de identificación el nombre del nodo del cual procede los datos para poder guardarlos en un archivo de texto en la memoria SD del nodo central.

El manejo del sistema por parte del usuario se realiza utilizando una interfaz gráfica desarrollada en Python que facilita el inicio de la recolección de los datos y el almacenamiento de estos para su posterior análisis.

Para la comprobación del funcionamiento del sistema, se utiliza para ejemplificar la simulación de un puente, una viga de 3 m, la cual en su punto central se le colocará un nodo secundario y se tomarán los datos correspondientes a la aceleración en el momento que se le produzcan golpes aleatorios en partes de la estructura, obteniendo los valores en los tres ejes y posteriormente graficando la información.

## Capítulo 4. Procedimiento de implementación

### 4.1 Diseño del módulo de medición

Para el módulo de medición, luego de consultas a los encargados del proyecto, se define que debe ser de un tamaño pequeño, para facilitar su manipulación a la hora de colocarlo en las estructuras tipo puente, y además de que sea de bajo costo económico. Debido que se debe utilizar el acelerómetro ADXL 335, se tiene que según [19] su tensión nominal de operación es de 3,3 V, lo cual indica que se debe de realizar un acondicionamiento de la alimentación debido a que el Waspote posee únicamente como salida para la alimentación 5 V, por lo que se procede a realizar un divisor de tensión que permita acondicionar la tensión de alimentación del ADXL 335. Para realizar el divisor de tensión se implementa el siguiente circuito:

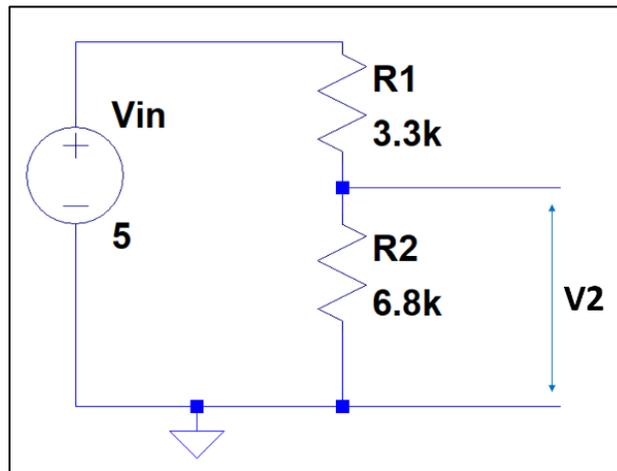


Figura 19. Divisor de tensión para alimentación del ADXL 335.

Tomando el divisor de tensión que presenta el circuito, se realiza el diseño de las resistencias R2 y R1, y se obtiene una nueva tensión según:

$$V_2 = V_{IN} \frac{R_2}{R_1 + R_2} = 5 \frac{6.8 \text{ k}\Omega}{3.3 \text{ k}\Omega + 6.8 \text{ k}\Omega} = 3,37 \text{ V}$$

Siguiendo con el proceso, en el diagrama de conexión del módulo ADXL 335 que se muestra en la Figura 20 se observa que se tiene 3 pines de salida, uno para cada eje, así como los pines de alimentación y referencia. Por lo que se implementa un circuito impreso que permita la conexión tanto de este módulo con el divisor de tensión para brindar su alimentación. En el diseño de la placa que se observa en la Figura 21, se utilizan pines tipo macho (JP1) para las conexiones provenientes del Wasp mote y pines tipo hembra (SV) para empotrar el módulo del acelerómetro; las resistencias utilizadas son de tipo superficial.

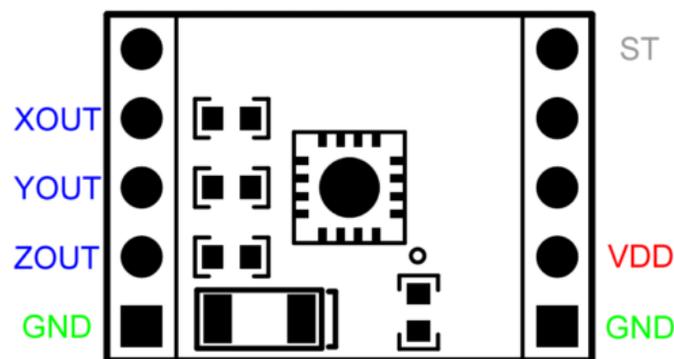


Figura 20. Diagrama de conexión del módulo ADXL335 [19]

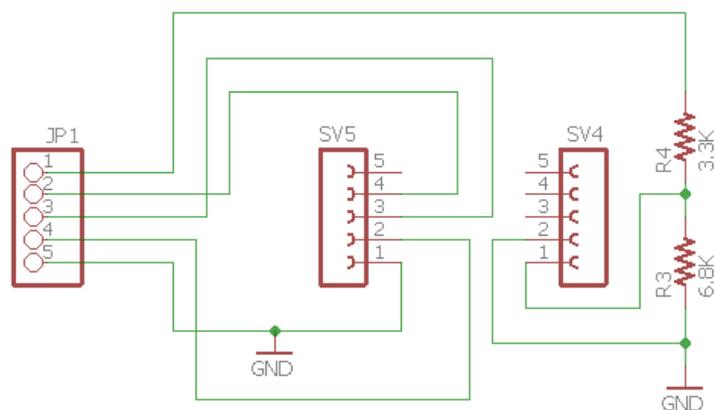


Figura 21. Placa de circuito impreso del módulo de medición.

La impresión final de la placa del circuito impreso queda de la siguiente forma:

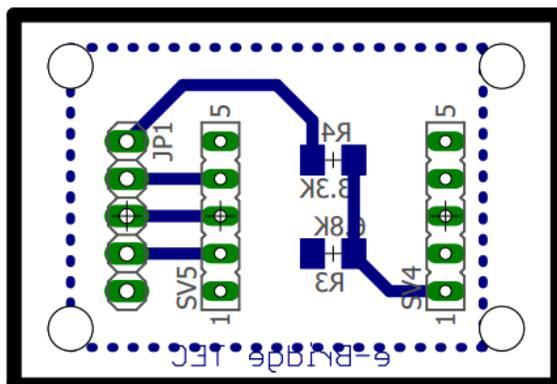


Figura 22. Placa de circuito impreso final del módulo de medición.

## 4.2 Conexión del módulo de medición al Waspote

Para proceder con la conexión del módulo de medición con la tarjeta de desarrollo Waspote se utilizan los puertos generales de entrada y salida (GPIO) de este último, en este caso se usan tres pines de entradas analógicas, la alimentación de 5 V y la referencia como se indica en la Tabla 4, utilizando cable tipo cinta de 40 cm de longitud con jumpers tipo hembra para la conexión hacia el módulo del acelerómetro.

Tabla 4. Pines de conexión en el Waspote

Pines del Waspote	Pines del ADXL335
<b>Pin Analógico 1</b>	Z out
<b>Pin Analógico 3</b>	Y out
<b>Pin Analógico 5</b>	X out
<b>Alimentación 5 V</b>	Vcc
<b>Referencia</b>	GND

Luego utilizando el software de programación IDLE de Waspote, se enciende la alimentación de los 5 V, así como se asignan los pines de los diferentes ejes. Posteriormente se realiza el manejo de los datos con la interpretación por parte del ADC, para calibrar adecuadamente el acelerómetro tomando en consideración el valor de sensibilidad en base a

la aceleración. En la hoja de datos del sensor se puede observar que el mismo para una tensión de alimentación es de 300 mV/g. Además, se debe de considerar el valor promedio que se obtenga del ADC proveniente del eje z, cuando el acelerómetro se encuentra en completo reposo, sin ninguna perturbación. En este caso el valor típico es de 512 (factor a utilizar en 0 g), por lo que se aplican las siguientes relaciones para obtener los factores de conversión que permitan conocer la aceleración real obtenida por el sensor.

$$\text{Factor de escala} = 330 \frac{\text{mV}}{\text{G}} \times \frac{1023 \text{ unidades ADC}}{3,3 \text{ V}} = 102,3 \frac{\text{unidades ADC}}{\text{G}}$$

Utilizando el factor de escala y el factor de 0 g, se obtendrá la salida real de cada eje, según la siguiente expresión:

$$\text{Valor esperado} = \frac{\text{valor de la medición} - \text{factor de 0 g}}{\text{factor de escala}}$$

### 4.3 Configuración de la transmisión de los Xbee

Para la transmisión de datos, los módulos Waspote utilizan Xbee como una de sus opciones inalámbricas. Los Xbee van empotrados en la misma placa de desarrollo, lo que permite disminuir el espacio en la implementación. La velocidad de transmisión que tienen por defecto los Waspote es de 38400 baudios, por lo que los Xbee se deben configurar a esa misma velocidad.

Para realizar la configuración de los módulos Xbee se utiliza el software proporcionado por su casa matriz Digi, denominado XCTU, el cual permite de una manera rápida cambiar los parámetros de acuerdo con lo que se necesita implementar. Así, la versión utilizada es la 6.3.8 y su interfaz se muestra en la Figura 23.

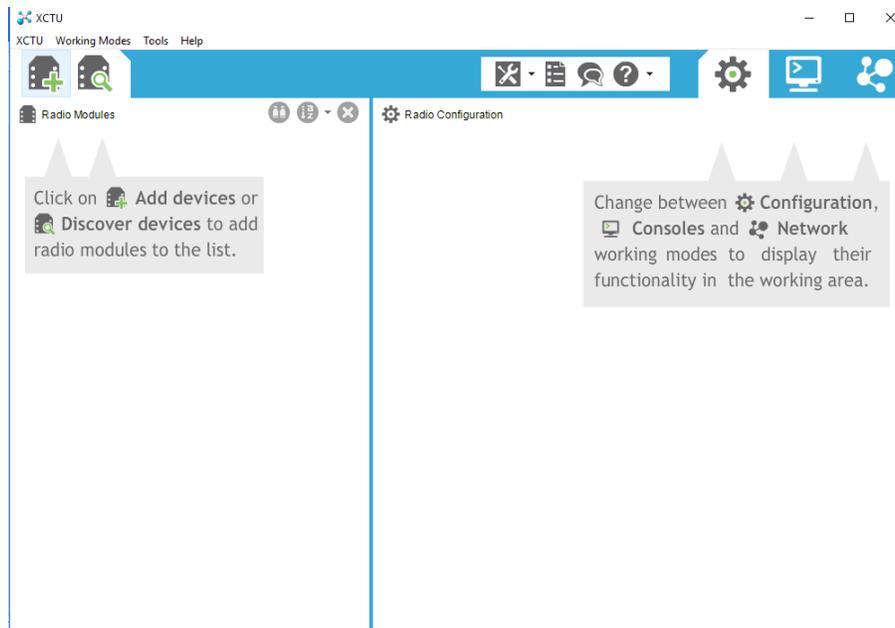


Figura 23. Interfaz del software XCTU

Para conectar cada Xbee con el software se utiliza el Xbee USB Explorer, facilitando el reconocimiento por parte del computador para determinar el puerto COM al cual está conectado. Posterior a ello en el software se le indica que realice una búsqueda de los dispositivos conectados, se elige el puerto COM asignado y de esta manera carga la información de la configuración del Xbee. Se muestran una serie de opciones, dentro de las cuales se modifican las siguientes para la implementación de la red, según se muestra en la Tabla 5.

Tabla 5. Configuración de los Xbee en el software XCTU.

	<b>Nodo central</b>	<b>Nodo 1</b>	<b>Nodo 2</b>	<b>Nodo 3</b>
<b>Canal</b>	B	B	B	B
<b>PAN ID</b>	2480	2480	2480	2480
<b>DH</b>	0	0	0	0
<b>DL</b>	FFFF	0	0	0
<b>Etiqueta</b>	Coordinador	Sensor 1	Sensor 2	Sensor 3
<b>Serial</b>	38400	38400	38400	38400
<b>Modo API</b>	API 1	API 1	API 1	API 1

El canal seleccionado fue el B, el cual opera a una frecuencia de 2,400-2405 GHz, pero se puede elegir entre 16 canales diferentes [27]. El PAN ID, corresponde al identificador de la red, compuesto por 16 bits, por lo que todos los nodos deben llevar el mismo valor para poder realizar la comunicación entre ellos. Se pueden utilizar 65536 posibilidades de identificador, para el proyecto se utiliza el 2480. Para el tipo de modo API (Application Programming Interface), se configura como API 1, donde se crea una trama de 64 bits con el objetivo de establecer la comunicación entre los nodos secundarios y el nodo coordinador de una forma ordenada y que se pueda identificar cada nodo para tener el control de la red tipo estrella que se implementa [28]. El modo tipo API de una trama de 64 bits, se compone de los siguientes elementos:

- Delimitador de inicio (Start Delimiter): Es el primer byte que indica el inicio de la trama, siempre su valor es 0x7E.
- Longitud (Length): Indica el número total de bytes de la trama, por medio de un total de dos bytes, pero no toma en cuenta el delimitador de inicio, la longitud y la suma de comprobación.
- Trama de datos (Frame data): Contiene el total de información que se recibe o transmite, estructurada según el propósito de la trama de los datos, como se observa por ejemplo en la Figura 24.
- Suma de comprobación (Checksum): Es el último byte de la trama y verifica si los datos se transmitieron correctamente, esto lo hace sumando todos los bytes de los elementos antecesores a él, excluyendo el limitador y la longitud [28].

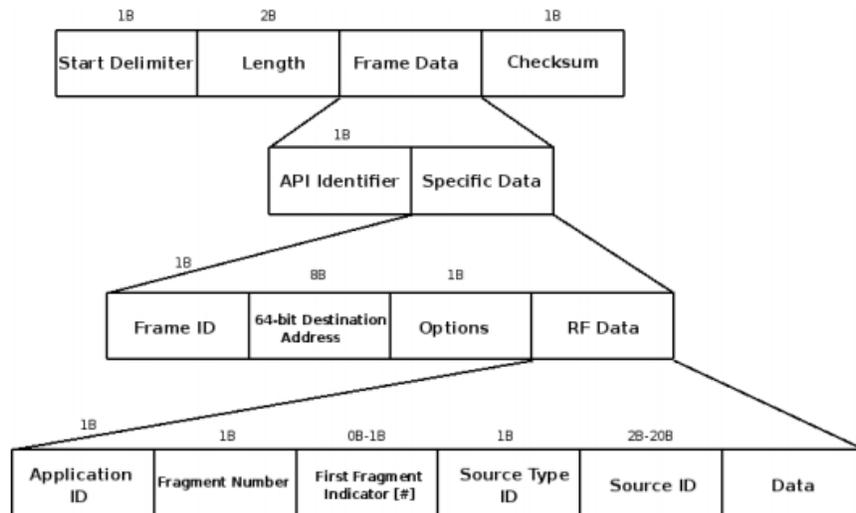


Figura 24. Trama de 64 bits del modo API [27]

Al configurar los módulos Xbee con este tipo de trama para la red permite identificar la dirección de origen del paquete recibido, lo cual facilita en la implementación conocer los datos de cada nodo secundario por separado para posteriormente procesarlos y almacenarlos.

## 4.4 Implementación de la red de sensores

### 4.4.1 Transmisión y recepción desde el nodo central

Como nodo central, se utiliza un microcontrolador Raspberry Pi 3, el cual por medio del lenguaje de programación de alto nivel Python, se crea una rutina para la solicitud de envío y recepción de los datos provenientes de los nodos secundarios. Se configura el puerto serial para una capacidad de transmisión de 38400 baudios. En uno de los puertos de la Raspberry Pi, se conecta el Xbee USB Explorer con el Xbee nombrado como coordinador, el cual será el que permita la intercomunicación entre los nodos.

Para la solicitud de envío se toma en cuenta la dirección única de cada Xbee (MAC), la cual consta de 64 bits y se encuentra impresa en la etiqueta de cada dispositivo. Con la ayuda de la librería de Xbee para el lenguaje Python, se utiliza la función `xbee.tx_long_addr()`, la cual tiene como parámetros el número de trama, la dirección de destino y por último el dato que se desea enviar, implementada de la siguiente manera, por ejemplo para el nodo 1:

```
xbee.tx_long_addr(frame=0x81, "\x00\x13\xA2\x00\x40\x69\x18\xF5", data='A')
```

Esta función se replica para los tres nodos secundarios, solamente se modifica la dirección de destino del Xbee del nodo que se quiere obtener respuesta de los datos. El número de trama se puede determinar utilizando la consola del software XCTU, observando en el proceso de envío el tercer elemento de la trama. En el momento que el dato es recibido por el Xbee a través del Wasmote enciende los leds de captura de señal, indicando que el dato es recibido correctamente.

El proceso de recepción de los datos provenientes de los nodos secundarios se realiza de una forma similar, utilizando la librería de Xbee, específicamente la función `xbee.wait_read_frame()`, la cual espera que se envíe una trama proveniente de cualquier nodo activo al cual ya le fue enviado el dato de solicitud. Posteriormente la captura de la información requerida se realiza teniendo en cuenta la estructura de la trama de Figura 24. La misma es enviada con formato tipo diccionario, en donde se convierte a lista para tomar específicamente la parte de `rf_data`, en donde se encuentra todos los datos provenientes del nodo solicitado.

Para identificar el nodo de procedencia y con la ayuda del manejo de listas en Python, cada conjunto de datos al inicio indica por medio de un indicador fijo su origen, lo que permite posteriormente realizar tres condiciones según ese identificador para guardar los datos de cada nodo secundario de forma independiente en un archivo tipo texto.

#### **4.4.1.1 Frecuencia de muestreo de la rutina del nodo central**

Para esta implementación según estudios realizados, la frecuencia máxima de la vibración a la cual están sometidas las estructuras tipo puente ronda entre los 10 a 12 Hz [29], por lo que aplicando el Teorema de Nyquist-Shannon, la rutina del nodo central se establece a una frecuencia de 30 Hz como valor mínimo.

#### 4.4.1.2 Programación del nodo central

En la Figura 25 se muestra el diagrama de flujo del software que controla las acciones en el nodo central según todas las especificaciones y características definidas en las secciones previas a esta.

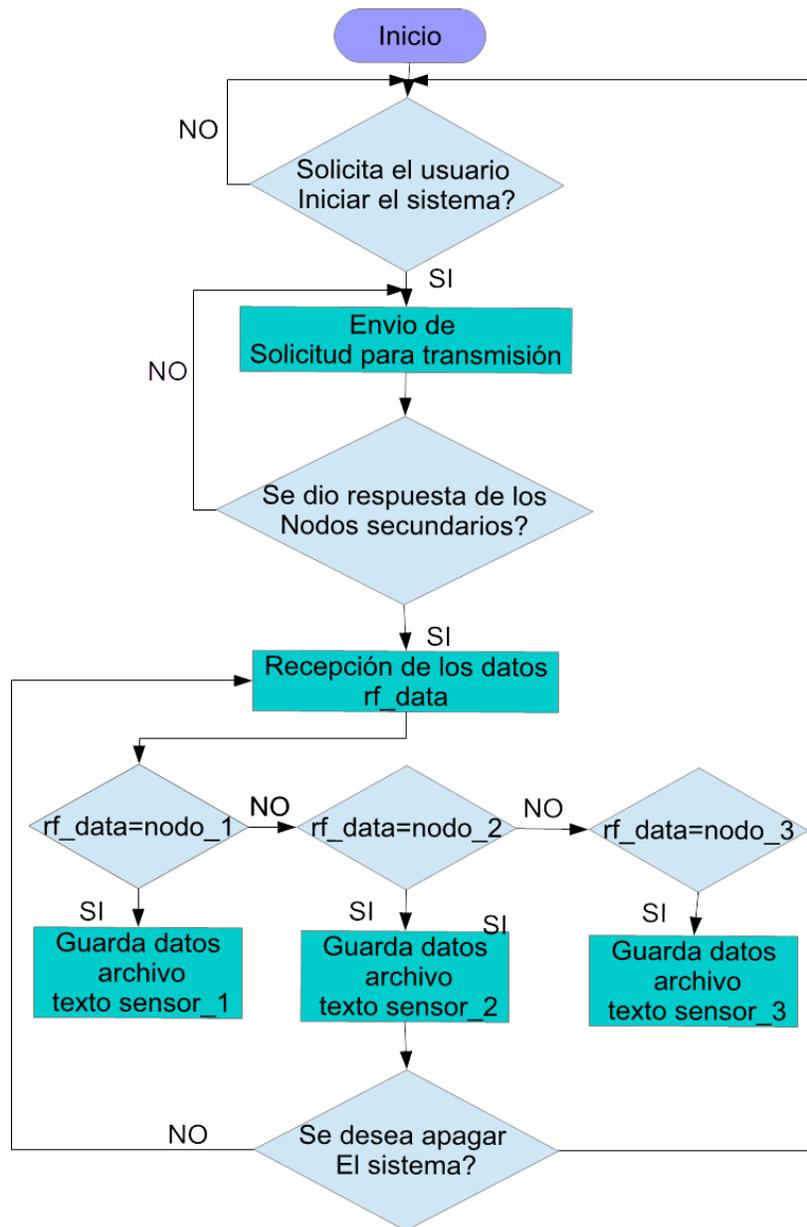


Figura 25. Diagrama de flujo de rutina del nodo central

#### 4.4.2 Rutina de procesamiento de datos del Wasmote

En los nodos secundarios, se utilizan los módulos Wasmote, los cuales se programan con su propio software, en este caso se utiliza la versión IDE 02, con el firmware API 33. Para iniciar con la estructura del procesamiento, se definen las variables a utilizar, para ello en la parte de la obtención de los datos desde el acelerómetro se utilizan variables de tipo entero, las cuales capturan la lectura de los pines analógicos del Wasmote. Además, se tiene una variable adicional que almacena el valor del reloj a tiempo real (RTC, siglas en inglés) de cada muestra. Posteriormente se realiza la transformación de los datos que provienen del ADC, con los cálculos que se explicaron en la sección 4.2. Luego de realizar ese proceso, se empaquetan los tres datos de los ejes en conjunto a la variable del tiempo y se contienen en un solo paquete tipo arreglo, listo para ser enviado.

Como respaldo del sistema, cada Wasmote cuenta con la opción de guardar los datos si se le coloca una memoria tipo SD de máximo 2 GB; por lo que al haber terminado el empaquetamiento de la información se va guardando cada muestra en un archivo tipo texto, esto previo a la secuencia de la solicitud de envío de los datos por medio del Xbee, con el fin de que si llega a tenerse un problema de comunicación entre el nodo central y el secundario, los datos medidos quedan almacenados de manera local.

Una de las características de los módulos Wasmote es que facilitan la comunicación por medio de los Xbee, los cuales, dentro de sus librerías por defecto, poseen funciones específicas para crear los enlaces con el nodo central, en particular con el Xbee coordinador. Para ello cuando el nodo central envía el dato de solicitud de medición, se utiliza la función `xbee802.treatData()`, la cual está dentro de una condición esperando y verifica si está llegando un dato para iniciar la transmisión de los datos, de no ser así, espera hasta que reciba una solicitud por parte del coordinador. Si la solicitud de transmisión llega se inicia con la rutina de envío utilizando la *función* `xbee802.send`, la cual tiene como parámetros la dirección MAC del Xbee coordinador, así como el arreglo del paquete de datos, como se observa a continuación:

```
xbee802.send("0013A20040691665", envío);
```

Hasta que el ciclo de solicitud de datos por parte del nodo central se detenga, la rutina de *espera y envío* de los Wasmote igualmente se detiene.

#### 4.4.2.1 Programación de los nodos secundarios

En la Figura 26 se muestra el diagrama de flujo del software que controla las acciones en el nodo central según todas las especificaciones y características definidas en las secciones previas a esta.

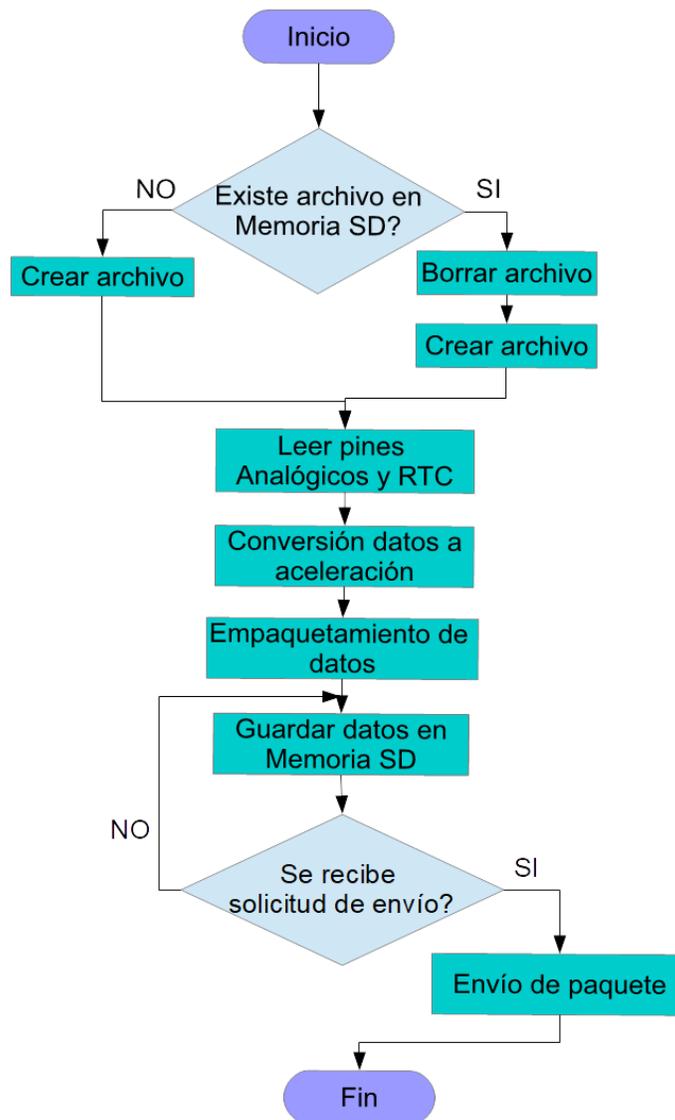


Figura 26. Diagrama de flujo de rutina de los nodos secundarios

### 4.4.3 Interfaz gráfica para usuario

Dentro de las facilidades del sistema que se le deben brindar al usuario, esta la implementación de la interfaz gráfica para la aplicación de la red de sensores, para ello se utiliza la librería de Python llamada Tkinter, la cual permite desarrollar un entorno práctico para las funciones que se requieren, como lo son el encender y apagar el sistema, informando en el momento que se oprima un botón, la acción que se está ejecutando. Para ello, se implementa una ventana con dos botones, uno para que el usuario lo oprima y así el sistema inicia con la lectura y captura de los datos, y el otro botón para detenerlo cuando se desea terminar con la rutina de transmisión de la información al nodo central.

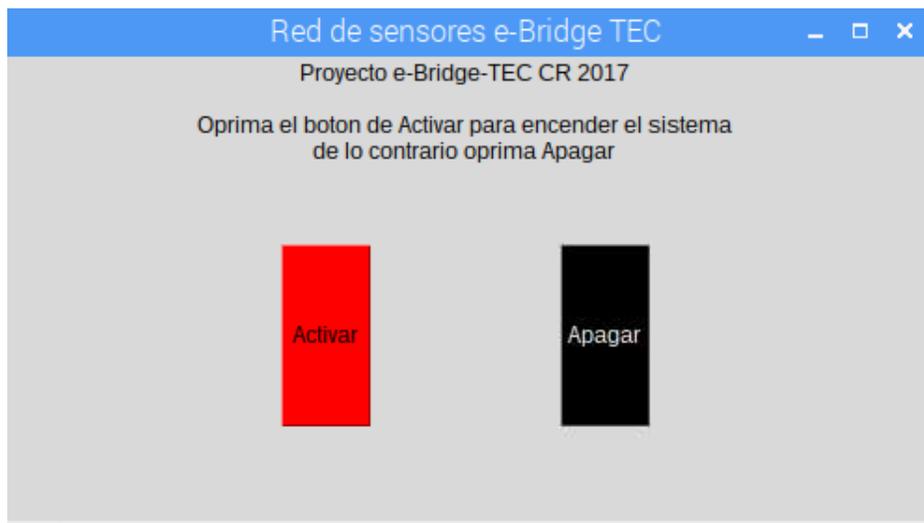


Figura 27. Ventana general de la interfaz gráfica

Si el sistema es activado y se informa con un mensaje emergente que el sistema es activado, de lo contrario indica que fue apagado, lo que permite al usuario saber cuál proceso se está ejecutando en el momento de oprimir alguno de los botones.



Figura 28. Mensajes emergentes de la interfaz gráfica

#### 4.4.3.1 Rutina de programación de la interfaz gráfica

En el siguiente diagrama de flujo se muestran los pasos de la rutina de la interfaz gráfica que permiten iniciar o detener el sistema cuando el usuario lo solicite.

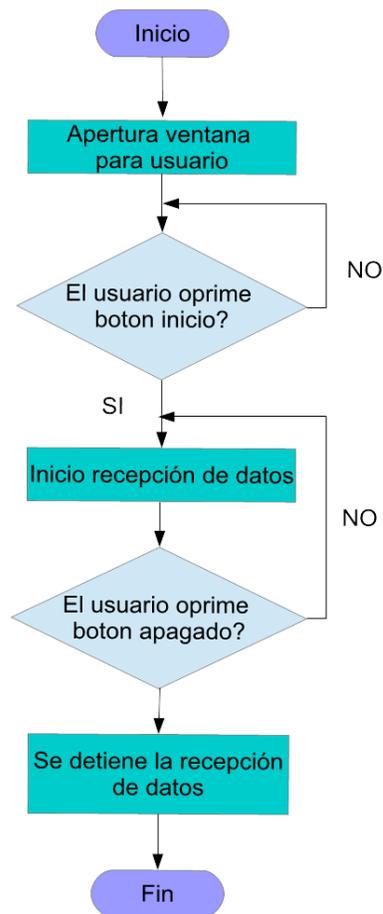


Figura 29. Diagrama de flujo de la rutina de la interfaz gráfica

#### 4.4.4 Alimentación de la red de sensores

Para el funcionamiento de la red de sensores, los Waspote se alimentan con una batería tipo Litio de 3,7 V y 1250 mAh, la cual se conecta en la misma placa de desarrollo con la ayuda del conector mini JST.

En el caso del nodo central, la Raspberry Pi opera a una tensión de 5V, por lo que para utilizarla con una batería de 3,7 V se necesitaba de un convertidor DC-DC comercial, para este caso se utiliza un PowerBoost 1000 Charger, el cual permite pasar de 3,7 a 5 V para la alimentación y además facilita cargar la batería en el mismo módulo, como el que se observa en Figura 30.

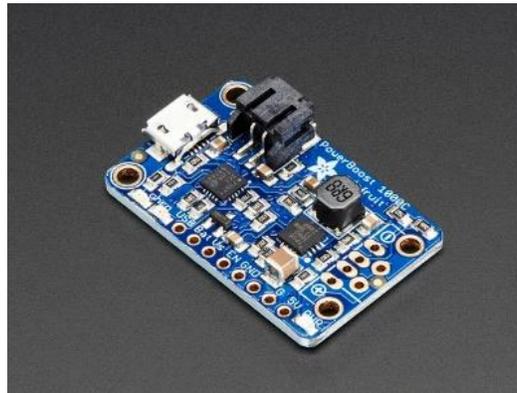


Figura 30. Convertidor de 3,7 a 5 V comercial Power Boost 1000 [30]

## Capítulo 5. Análisis de Resultados

Se presentan en esta sección los resultados obtenidos de cada etapa del sistema, explicando los procedimientos aplicados para la verificación funcional de la red de sensores. Posteriormente se analizarán los datos para determinar los alcances o limitaciones de los equipos utilizados en la solución.

### 5.1 Circuito acondicionador de señal de alimentación del módulo de medición

Luego del montaje del circuito acondicionador de señal en una tarjeta de circuito impreso para el módulo ADXL335, se verifica que cada pin de entrada como de salida, indiquen los valores de tensión requeridos para su funcionamiento y armado final del módulo de medición de la Figura 31.

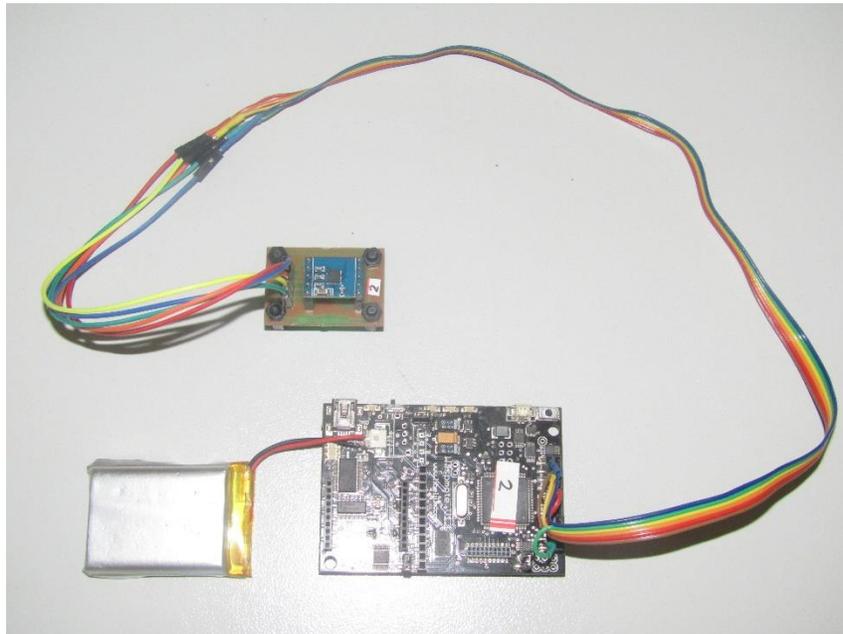


Figura 31. Módulo de medición implementado

En el nodo del divisor de tensión propuesto, se determina con la ayuda de un multímetro que su valor es menor a 3,3 V, se realiza una revisión en los tres nodos y el dato exacto es de 2,7 V en todos los módulos. Por lo que se revisa la hoja de datos del módulo ADXL 335 y su rango de operación de tensión es de 1,8 a 3,6 V, lo que muestra que el módulo trabaja adecuadamente con ese valor de tensión de alimentación. Solamente se debe tener en cuenta de modificar la constante de sensibilidad para la calibración del acelerómetro, en el momento que se requiera ajustar para obtener los valores reales de aceleración.

Si para una tensión de alimentación de 3,6 V la sensibilidad es de 360 mV/g, para 2,7 V será de 270 mV /g , debido a que tiene un comportamiento lineal con respecto a la tensión que se aplique [19]. Lo indicado anteriormente, sucede debido a que los pines de entrada/salida del Waspote tiene una corriente limitada menor a 20 mA y al momento de conectar el módulo del acelerómetro la corriente disminuye, provocando la disminución en la tensión del divisor de la alimentación del ADXL 335. Para ello, la mejor alternativa que se puede implementar es hacer un seguidor de tensión con fuente de alimentación separada del Waspote para tener un resultado de 3,3 V en la entrada del módulo.

## **5.2 Conexión entre nodo central y secundarios**

Primero se comprueba la comunicación entre el nodo central y los nodos secundarios específicamente con los Xbee, con la ayuda del software XCTU. En la Figura 32 se puede observar la topología estrella de comunicación del sistema de medición, donde se muestra el nodo coordinador y los tres nodos auxiliares con su respectiva etiqueta y dirección. Posteriormente, se implementa el nodo central, con la conexión del USB Xbee Explorer a uno de los puertos de USB y la pantalla táctil a los puertos GPIO, como se muestra en la Figura 33. En el momento que se ejecuta el código desarrollado en Python con la interfaz gráfica, se puede observar que los leds del nivel de RSSI (siglas en inglés, indicador de intensidad de señal recibida) de los Xbee encienden en su totalidad en el Waspote, como se puede observar en la Figura 34. Consultando la hoja de datos de la tarjeta indica que si los tres leds se encuentran encendidos, significa que el nivel sensibilidad y calidad de la señal que recibe a través del Xbee es mayor de 31.82 dB, por lo que la solicitud de envío de transmisión es correcta de acuerdo a lo esperado [20].

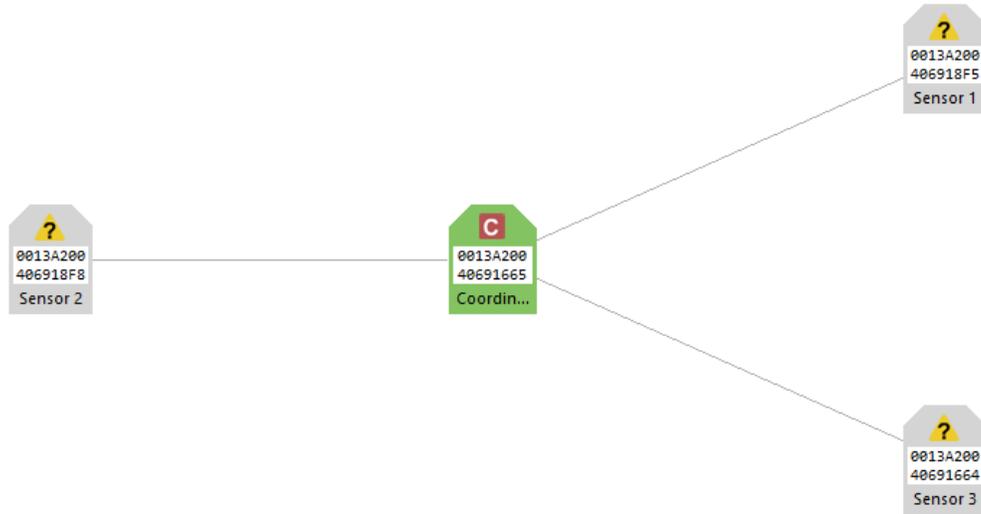


Figura 32. Topología obtenida con el software XCTU

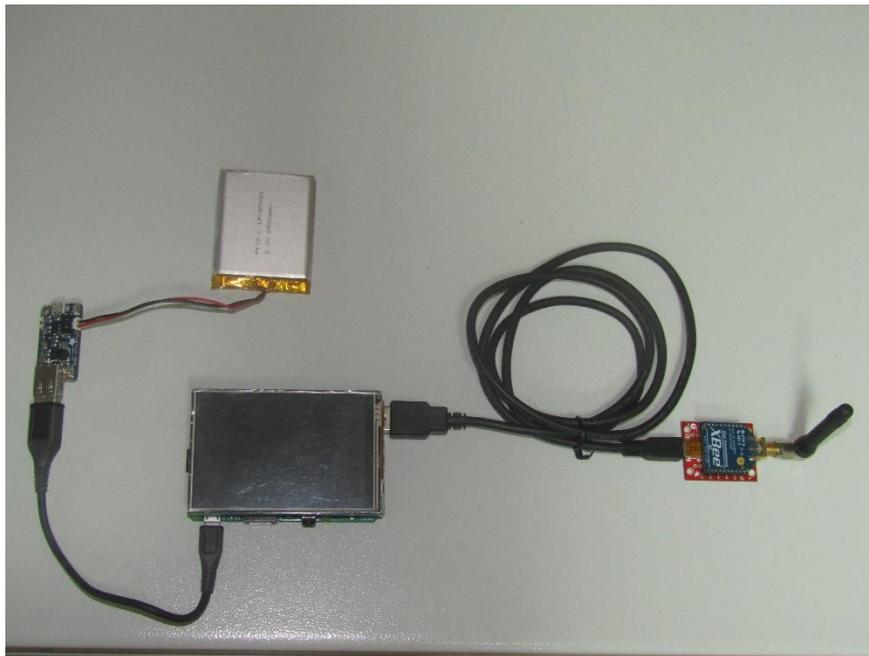


Figura 33. Hardware del nodo central



Figura 34. Nivel de RSSI en el módulo de medición

Para la recepción de los datos en el nodo central, en pruebas realizadas utilizando la consola de Python, obtenemos la información de los nodos con las características que se desean, en donde se evidencia que se logra la comunicación con los tres nodos que conforman la red.

```

Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/Desktop/red_sensores1.py =====
[ 'nodo_1', '-0.41935', '-0.68892', '-0.36559', 'Monday', '17/11/27 - 14:45.08', '\n' ]
[ 'nodo_3', '-1.35484', '-0.09677', '-0.13978', 'Monday', '17/11/27 - 14:44.37', '\n' ]
[ 'nodo_2', '-0.03226', '-0.96774', '0.26882', 'Monday', '17/11/27 - 14:44.55', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.09', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.09', '\n' ]
[ 'nodo_1', '-0.41935', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.09', '\n' ]
[ 'nodo_2', '-0.03226', '-0.96774', '0.26882', 'Monday', '17/11/27 - 14:44.56', '\n' ]
[ 'nodo_2', '-0.04301', '-0.96774', '0.26882', 'Monday', '17/11/27 - 14:44.56', '\n' ]
[ 'nodo_3', '-0.03226', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.56', '\n' ]
[ 'nodo_3', '-1.35484', '-0.09677', '-0.13978', 'Monday', '17/11/27 - 14:44.38', '\n' ]
[ 'nodo_1', '-0.41935', '-0.69892', '-0.37634', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_3', '-1.35484', '-0.09677', '-0.15054', 'Monday', '17/11/27 - 14:44.38', '\n' ]
[ 'nodo_1', '-0.41935', '-0.69892', '-0.37634', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_1', '-0.41935', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_2', '-0.04301', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.56', '\n' ]
[ 'nodo_3', '-1.34409', '-0.09677', '-0.15054', 'Monday', '17/11/27 - 14:44.38', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_2', '-0.04301', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.56', '\n' ]
[ 'nodo_3', '-1.35484', '-0.09677', '-0.15054', 'Monday', '17/11/27 - 14:44.38', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.37634', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_2', '-0.04301', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.56', '\n' ]
[ 'nodo_3', '-1.35484', '-0.09677', '-0.13978', 'Monday', '17/11/27 - 14:44.39', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_1', '-0.41935', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_2', '-0.03226', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.56', '\n' ]
[ 'nodo_3', '-1.35484', '-0.09677', '-0.15054', 'Monday', '17/11/27 - 14:44.39', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_2', '-0.03226', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.57', '\n' ]
[ 'nodo_3', '-1.36559', '-0.09677', '-0.13978', 'Monday', '17/11/27 - 14:44.39', '\n' ]
[ 'nodo_2', '-0.04301', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.57', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_3', '-1.35484', '-0.09677', '-0.13978', 'Monday', '17/11/27 - 14:44.39', '\n' ]
[ 'nodo_1', '-0.40860', '-0.69892', '-0.36559', 'Monday', '17/11/27 - 14:45.10', '\n' ]
[ 'nodo_2', '-0.03226', '-0.96774', '0.27957', 'Monday', '17/11/27 - 14:44.57', '\n' ]
  
```

Figura 35. Recepción de datos en el nodo central con Python

### **5.3 Datos almacenados en memoria tipo SD**

Utilizando una memoria tipo SD con capacidad de 2 GB, se capturan los datos provenientes del módulo de medición, obteniendo un archivo de texto, el cual se abre a través de cualquier software de procesamiento de hojas de cálculo o texto. Como se puede observar en Tabla 6, el orden la tabulación de los datos es correcta, en donde como primer dato tenemos el nodo de procedencia, posteriormente los valores de aceleración de los tres ejes, seguido por el día, fecha y la hora de la medición. La generación de este archivo es para generar un respaldo del sistema en caso de que la comunicación entre el nodo secundario y la central falle o sea interrumpida en algún momento. En el caso del tiempo de muestreo del proceso el almacenamiento de los datos se puede notar en la tabla esto no es correcto, ya que únicamente se tienen 15 mediciones por segundo, lo que representa una frecuencia de 15 Hz y no de 30 Hz. Por ello, más adelante en este capítulo se analizarán las posibles causas del problema.



## 5.4 Almacenamiento de los datos de cada nodo secundario en el nodo central

Cuando se inicia la transferencia de los datos de cada nodo secundario al principal, se inicia el proceso de almacenamiento con la creación de un archivo tipo texto por cada nodo de la red, como se puede observar en la Figura 36. La ubicación de los archivos será la misma en donde se encuentre el script principal del proyecto, denominado “red\_sensores.py”.

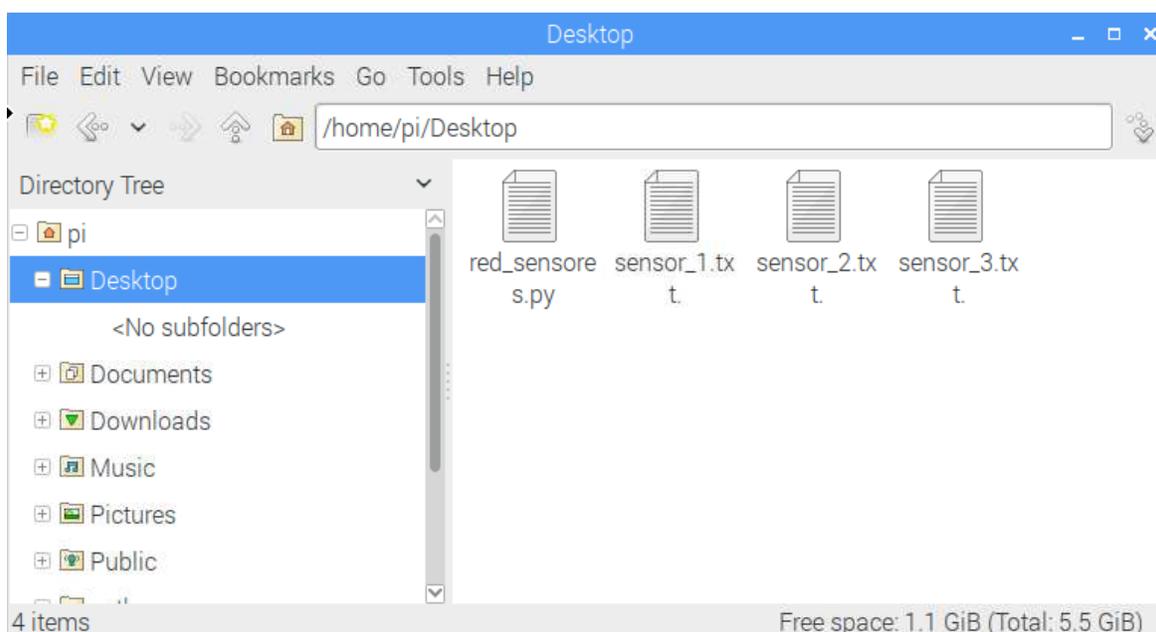


Figura 36. Creación de los archivos de texto por cada nodo secundario en la Raspberry Pi 3

Cuando se abren los archivos con la ayuda de un procesador de texto u hojas de cálculo, obtenemos las siguientes tablas que corresponden a los datos obtenidos por cada nodo. Se muestran solamente como ejemplificación una cantidad de datos limitada, ya que eso depende del tiempo en que se desea tomar las mediciones y los archivos que se generan pueden contener una gran cantidad de datos.

Tabla 7. Datos del archivo de texto obtenidos del nodo 1

<b>nodo_1</b>	<b>-0.38710</b>	<b>0.20430</b>	<b>1.08602</b>	<b>Friday</b>	<b>17/11/24</b>	<b>12:56.22</b>
nodo_1	-0.40860	0.23656	1.08602	Friday	17/11/24	12:56.23
nodo_1	-0.38710	0.20430	1.09677	Friday	17/11/24	12:56.23
nodo_1	-0.41935	0.20430	1.09677	Friday	17/11/24	12:56.23
nodo_1	-0.40860	0.19355	1.10753	Friday	17/11/24	12:56.23
nodo_1	-0.39785	0.19355	1.09677	Friday	17/11/24	12:56.23
nodo_1	-0.40860	0.21505	1.10753	Friday	17/11/24	12:56.24
nodo_1	-0.40860	0.20430	1.09677	Friday	17/11/24	12:56.24
nodo_1	-0.40860	0.20430	1.09677	Friday	17/11/24	12:56.24
nodo_1	-0.41935	0.19355	1.08602	Friday	17/11/24	12:56.24
nodo_1	-0.40860	0.20430	1.10753	Friday	17/11/24	12:56.24
nodo_1	-0.45161	0.16129	1.09677	Friday	17/11/24	12:56.24
nodo_1	-0.48387	0.20430	1.07527	Friday	17/11/24	12:56.24
nodo_1	-0.48387	0.18280	1.09677	Friday	17/11/24	12:56.24
nodo_1	-0.46237	0.20430	1.01075	Friday	17/11/24	12:56.24
nodo_1	-0.44086	0.21505	0.87097	Friday	17/11/24	12:56.25
nodo_1	-0.46237	0.27957	1.03226	Friday	17/11/24	12:56.25
nodo_1	-0.47312	0.25806	0.72043	Friday	17/11/24	12:56.25
nodo_1	-0.48387	0.25806	0.93548	Friday	17/11/24	12:56.25
nodo_1	-0.46237	0.26882	0.73118	Friday	17/11/24	12:56.25
nodo_1	-0.50538	0.26882	0.65591	Friday	17/11/24	12:56.25
nodo_1	-0.51613	0.23656	0.65591	Friday	17/11/24	12:56.25
nodo_1	-0.51613	0.20430	0.65591	Friday	17/11/24	12:56.25
nodo_1	-0.43011	0.18280	0.62366	Friday	17/11/24	12:56.25
nodo_1	-0.49462	0.19355	1.06452	Friday	17/11/24	12:56.25
nodo_1	-0.44086	0.21505	1.09677	Friday	17/11/24	12:56.25
nodo_1	-0.45161	0.18280	1.05376	Friday	17/11/24	12:56.26
nodo_1	-0.51613	0.16129	1.04301	Friday	17/11/24	12:56.26
nodo_1	-0.48387	0.13978	1.04301	Friday	17/11/24	12:56.26
nodo_1	-0.47312	0.18280	1.00000	Friday	17/11/24	12:56.26
nodo_1	-0.50538	0.21505	1.10753	Friday	17/11/24	12:56.26
nodo_1	-0.56989	0.16129	1.10753	Friday	17/11/24	12:56.26
nodo_1	-0.54839	0.19355	1.05376	Friday	17/11/24	12:56.26
nodo_1	-0.55914	0.15054	0.60215	Friday	17/11/24	12:56.26
nodo_1	-0.53763	0.17204	0.61290	Friday	17/11/24	12:56.26

Tabla 8. Datos del archivo de texto obtenidos del nodo 2

<b>nodo_2</b>	<b>0.69892</b>	<b>0.03226</b>	<b>0.82796</b>	<b>Friday</b>	<b>17/11/24</b>	<b>12:56.09</b>
nodo_2	0.69892	0.03226	0.83871	Friday	17/11/24	12:56.09
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.09
nodo_2	0.69892	0.03226	0.83871	Friday	17/11/24	12:56.09
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.09
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.03226	0.83871	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.10
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.11
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.11
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.11
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.11
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.11
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.11
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.11
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.12
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.12
nodo_2	0.68817	0.04301	0.83871	Friday	17/11/24	12:56.12
nodo_2	0.69892	0.03226	0.82796	Friday	17/11/24	12:56.12
nodo_2	0.69892	0.03226	0.82796	Friday	17/11/24	12:56.12
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.12
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.12
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.13
nodo_2	0.69892	0.04301	0.82796	Friday	17/11/24	12:56.13
nodo_2	0.69892	0.03226	0.83871	Friday	17/11/24	12:56.13
nodo_2	0.69892	0.03226	0.83871	Friday	17/11/24	12:56.13
nodo_2	0.69892	0.03226	0.83871	Friday	17/11/24	12:56.14
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.14
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.14
nodo_2	0.69892	0.03226	0.83871	Friday	17/11/24	12:56.14
nodo_2	0.69892	0.04301	0.83871	Friday	17/11/24	12:56.14

Tabla 9. Datos del archivo de texto obtenidos del nodo 3.

<b>nodo_3</b>	<b>-0.27957</b>	<b>0.72043</b>	<b>0.80645</b>	<b>Friday</b>	<b>17/11/24</b>	<b>12:56.00</b>
nodo_3	-0.19355	0.75269	0.45161	Friday	17/11/24	12:56.00
nodo_3	-0.16129	0.75269	0.38710	Friday	17/11/24	12:56.00
nodo_3	-0.15054	0.78495	0.43011	Friday	17/11/24	12:56.01
nodo_3	-0.15054	0.70968	0.41935	Friday	17/11/24	12:56.01
nodo_3	-0.15054	0.72043	0.40860	Friday	17/11/24	12:56.01
nodo_3	-0.16129	0.76344	0.43011	Friday	17/11/24	12:56.01
nodo_3	-0.21505	0.74194	0.41935	Friday	17/11/24	12:56.01
nodo_3	-0.36559	0.72043	0.46237	Friday	17/11/24	12:56.01
nodo_3	-0.46237	0.72043	0.40860	Friday	17/11/24	12:56.01
nodo_3	-0.43011	0.75269	0.36559	Friday	17/11/24	12:56.01
nodo_3	-0.37634	0.81720	0.45161	Friday	17/11/24	12:56.01
nodo_3	-0.38710	0.62366	0.31183	Friday	17/11/24	12:56.01
nodo_3	-0.31183	0.83871	0.43011	Friday	17/11/24	12:56.01
nodo_3	-0.36559	0.74194	0.39785	Friday	17/11/24	12:56.02
nodo_3	-0.31183	0.74194	0.38710	Friday	17/11/24	12:56.02
nodo_3	-0.30108	0.65591	0.38710	Friday	17/11/24	12:56.02
nodo_3	-0.33333	0.75269	0.47312	Friday	17/11/24	12:56.02
nodo_3	-0.35484	0.77419	0.46237	Friday	17/11/24	12:56.02
nodo_3	-0.25806	0.75269	0.48387	Friday	17/11/24	12:56.02
nodo_3	-0.17204	0.69892	0.43011	Friday	17/11/24	12:56.02
nodo_3	-0.22581	0.70968	0.45161	Friday	17/11/24	12:56.02
nodo_3	-0.13978	0.74194	0.43011	Friday	17/11/24	12:56.02
nodo_3	-0.10753	0.73118	0.45161	Friday	17/11/24	12:56.02
nodo_3	-0.12903	0.72043	0.47312	Friday	17/11/24	12:56.02
nodo_3	-0.07527	0.70968	0.46237	Friday	17/11/24	12:56.02
nodo_3	-0.09677	0.74194	0.48387	Friday	17/11/24	12:56.03
nodo_3	-0.16129	0.67742	0.49462	Friday	17/11/24	12:56.03
nodo_3	-0.04301	0.77419	0.52688	Friday	17/11/24	12:56.03
nodo_3	-0.09677	0.70968	0.50538	Friday	17/11/24	12:56.03
nodo_3	-0.08602	0.73118	0.50538	Friday	17/11/24	12:56.03
nodo_3	-0.07527	0.68817	0.49462	Friday	17/11/24	12:56.03
nodo_3	-0.06452	0.72043	0.51613	Friday	17/11/24	12:56.03
nodo_3	-0.05376	0.69892	0.49462	Friday	17/11/24	12:56.03
nodo_3	-0.04301	0.70968	0.51613	Friday	17/11/24	12:56.03

Según las tablas anteriores, los datos se obtienen como se esperaban, con el identificador del nodo de procedencia, los valores de aceleración de los ejes x, y, z respectivamente, seguido del día, fecha y hora de la medición. Como se puede observar en todas las tablas, en el lapso de un segundo no se obtiene una cantidad mayor de 9 datos en promedio, a pesar de que la frecuencia de muestreo estaba programada en el nodo central para obtener una cantidad de 30 datos por segundo. Las razones de este comportamiento del sistema se analizarán en la próxima sección.

## **5.5 Frecuencia de muestreo del sistema.**

En el proceso de implementación y diseño, se mencionó que la frecuencia mínima de muestreo a utilizar en el sistema era de 30 Hz, esto con el fin de obtener datos que cumplieran con el teorema de Nyquist para su posterior análisis en el dominio de la frecuencia para determinar la composición espectral de las vibraciones que sufren las estructuras tipo puente. Luego de varias pruebas que se realizaron con el sistema, se observó que la cantidad de datos por segundo a esa frecuencia indicada en el nodo central no variaba, como se indicó en la sección anterior y en la sección del almacenamiento en la memoria tipo SD, los datos obtenidos no representan los esperados.

Para determinar la razón del problema, se realizan pruebas cambiando la frecuencia del nodo central, primero con un solo nodo, luego con dos y por último con tres, los resultados obtenidos se observan en la Tabla 10.

Tabla 10. Comportamiento de cantidad de datos según la frecuencia y número de nodos

<b>Con un nodo secundario</b>	<b>Nodo central (Hz)</b>	<b>Cantidad Datos</b>
	30	6
	20	5
	10	4
	8	4
	6	3
	5	3
	4	2
	3	2
	2	2
	1	1
<b>Con dos nodos secundarios</b>	<b>Nodo central (Hz)</b>	<b>Cantidad Datos</b>
	30	8
	20	7
	10	7
	8	6
	6	4
	5	4
	4	3
	3	3
	2	2
	1	1
<b>Con tres nodos secundarios</b>	<b>Nodo central (Hz)</b>	<b>Cantidad Datos</b>
	30	9
	20	7
	10	7
	8	6
	6	4
	5	4
	4	3
	3	3
	2	2
	1	1

Al observar la tabla anterior, se puede notar que conforme baja la frecuencia del nodo central, se va acercando a la cantidad de datos solicitados por segundo y además que al tener más nodos en el sistema mejora la transmisión de los datos en las frecuencias más altas y luego en las más bajas se comporta de forma similar para los tres casos. Otra de las pruebas realizadas con el sistema fue aumentar la frecuencia de muestreo del nodo central, hasta 1 kHz, pero se obtuvieron como máximo 9 datos por cada segundo, lo que nos da indicios para la búsqueda de otras razones sobre el problema del tiempo de muestreo.

En el capítulo de implementación, se mencionó que la velocidad de transferencia utilizada fue de 38400 bps, debido a que los módulos Waspote, por defecto trabajan a esa tasa de transmisión. Al analizar las tramas de datos que se envían y se reciben a través de los nodos, su tamaño consta de 64 bits. Recordando que la frecuencia muestreo mínima es de 30 Hz o un periodo de muestreo de 33,3 ms. Si se multiplica la cantidad de bits por segundo de la transmisión por el tiempo de muestreo, nos da como resultado 1280 bits por periodo de muestreo, siendo la capacidad de transmisión del sistema.

Por lo que puede transmitir un máximo de 1280 bits, y si dividimos este valor entre el tamaño entre la trama de solicitud y la de respuesta (128 bits en total), el resultado es de 10 paquetes por segundo. Al intentar modificar las librerías de las tasas de velocidad de transmisión de los Waspote, se tiene que valor máximo que acepta es de 38400 bps. Además, cabe señalar que los Xbee si son configurables para mayores tasas de transmisión, por lo que se puede deducir que el problema del tiempo de muestreo es producto de una limitación del hardware facilitado para el proyecto, y para que poder lograr las frecuencias que se deseaban se necesita utilizar nuevas tarjetas de desarrollo en los nodos secundarios con velocidades de transmisión en rango de Mbps o Gbps, con el fin obtener mediciones que permitan realizar análisis espectral sobre las señales de vibración monitoreas. Además, es importante indicar que, como consecuencia de la limitación de hardware, las rutinas de programación para el proceso de información implementadas son afectadas y no se realizan de la forma esperada, especialmente en los nodos secundarios en el almacenamiento de los datos en la tarjeta de memoria SD. Por lo que para mejorar esto se deben desarrollar subprocesos independientes, uno para el almacenamiento local en cada nodo y otro para la transmisión, con el fin de evitar dependencias entre los procesos.

## 5.6 Comprobación de la medición de aceleración del sistema utilizando una viga.

Como parte de las pruebas realizadas al sistema, se utilizó una viga de hierro en forma circular de 6 m de largo por 2 pulgadas de diámetro y 4 mm de espesor, con un grado de libertad, como se observa en la *Figura 37*. Con la ayuda del software SAP 2000, se realiza una simulación de la estructura, se le indican las características físicas de la viga, aplicando una perturbación a 1,5 m de distancia desde uno de los soportes, obteniendo como resultado la aceleración del eje z en la gráfica de la *Figura 38*.



Figura 37. Montaje de la prueba de vibración en la viga

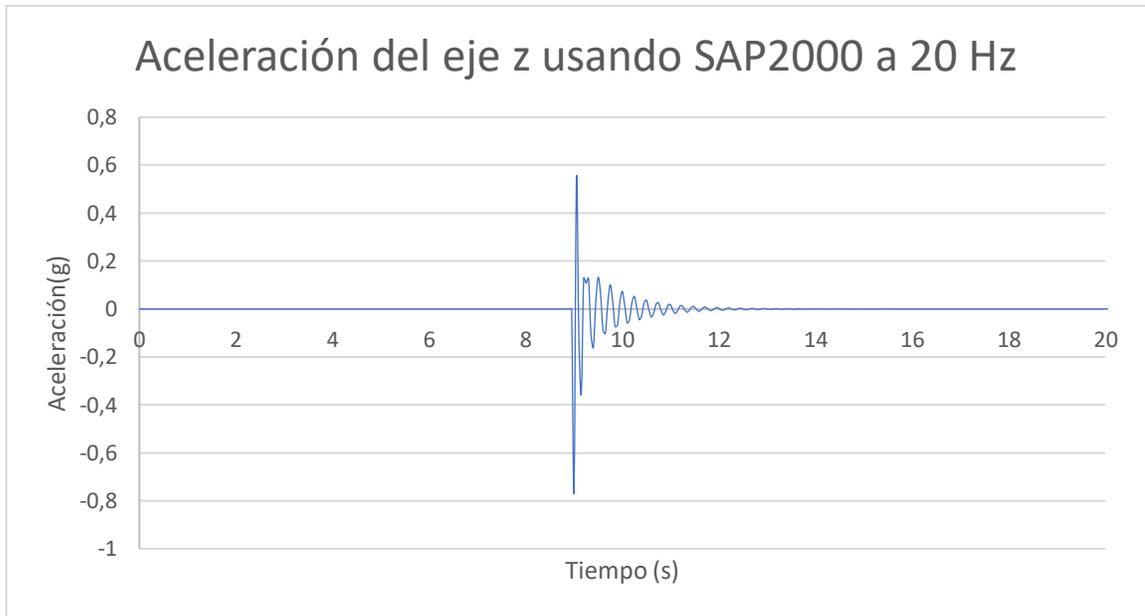


Figura 38. Simulación de la respuesta de aceleración del eje z de la viga.

Para la comprobación del funcionamiento del sistema, se utilizó como referencia el equipo de medición de la marca Bridge Test, con un acelerómetro uniaxial modelo A2120, con una sensibilidad de rango 5 g, como el que se observa en la Figura 39 [31].



Figura 39. Acelerómetro BDI A2120

Se colocó el sensor BDI A2120 junto al sensor ADXL335 del nodo secundario, como se muestra en la Figura 40, a una distancia de 3 m, es decir en el centro de la estructura y se aplicó un golpe a una distancia de 1.5 m en la viga, en donde ambos equipos tomaron la aceleración en función del tiempo del eje z, ya que es el eje que se ve mayormente afectado

por la perturbación generada por el golpe generado sobre la viga, obteniendo la gráfica de la Figura 41 con la ayuda del software STS Live de BDI y la del equipo del proyecto con la ayuda de Microsoft Excel en la Figura 42.

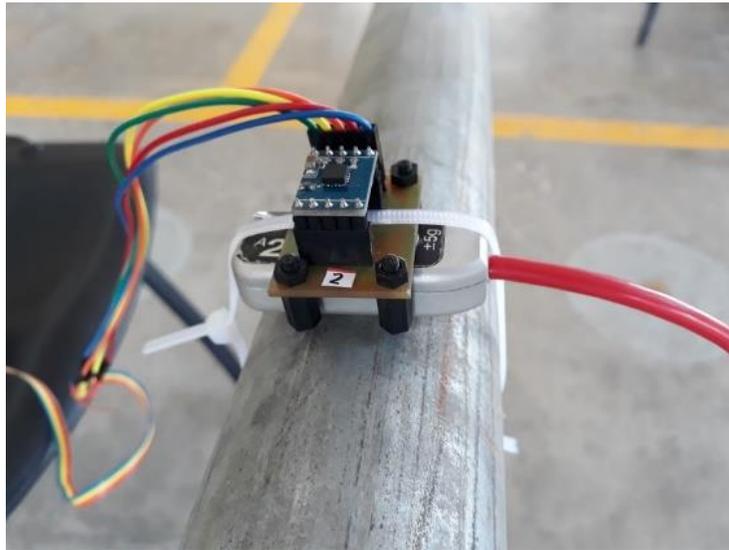


Figura 40. Colocación de los sensores sobre la viga

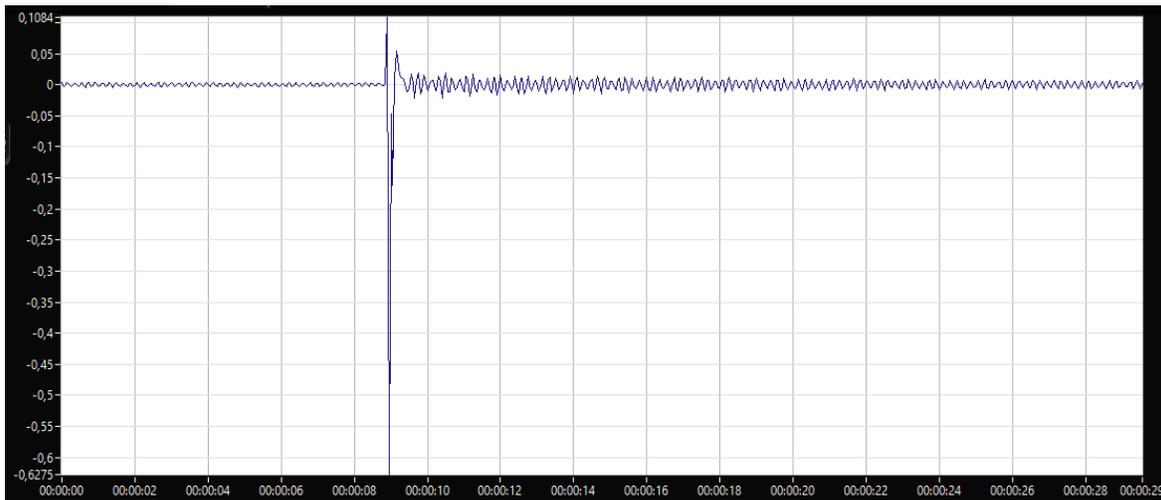


Figura 41. Aceleración del eje z obtenida con el BDI A2120



Figura 42. Aceleración del eje z de la viga con el equipo del sensor ADXL335.

Se puede observar tanto en la figura de la simulación como en las dos figuras anteriores, que los resultados tienen un comportamiento similar, pero existen diferencias en la amplitud de la señal, ya que con el sensor BDI 2120, este cuenta con una sensibilidad de 400 mV/g para un rango de 5 g [31], y para el caso del ADXL335 su sensibilidad es de 270 mV/g de un rango de 3g, como se explicó en la sección 5.1.

Con respecto a las mediciones realizadas, se puede analizar con los puntos máximos y mínimos de las gráficas, el porcentaje de error entre los equipos de medición, calculado con la diferencia de los valores obtenidos del sensor ADXL 335 y del equipo BDI, dividido entre la referencia que sería este último sensor, lo cual se resume en la siguiente tabla.

Tabla 11. Comparación de los valores mínimos y máximos de la prueba.

	<b>Puntos Máximos</b>	<b>Puntos Mínimos</b>
<b>Sensor BDI 2120</b>	0,1084	-0,6275
<b>Sensor ADXL 335</b>	0,09677	-0,3871
<b>Diferencia</b>	0,01163	0,2404
<b>Porcentaje de error (%)</b>	10,72	38,31

En la Tabla 11 se muestran los porcentajes de error obtenidos con el sensor ADXL335 tomando como referencias los valores obtenidos con el sensor BDI 2120. En el caso del valor máximo es de 10,72 % y en el caso del mínimo es de 38,31 %. Esta variación especialmente en los puntos mínimos se debe a que existe un error de calibración o ajuste del sensor ADXL 335, que se observa en la gráfica con una magnitud de -0,01076, desde el inicio de la prueba, lo que afecta los resultados obtenidos.

### **5.7 Valores máximos de tiempo activo y distancia de la red de sensores.**

Durante las pruebas realizadas, se comprueba que el sistema activado es capaz de obtener datos durante un rango de tiempo mayor a los 90 min, almacenando las mediciones en los archivos de texto constantemente sin mayor problema. En cuanto a la distancia, se hacen dos tipos de pruebas, con línea vista entre nodo central y secundarios, y la de obstáculos como vehículos, personas y estructuras, presentes entre los nodos. Para el primer caso se obtiene una distancia máxima de 50 m y para el segundo caso una distancia entre los 10 a 15 m.

### **5.8 Consumo de potencia del sistema**

Uno de los temas más importantes de las redes de sensores es sobre el consumo de potencia en dos ámbitos: en el momento de la transferencia de los datos y el otro cuando el sistema se encuentra en reposo. En el caso del sensor ADXL335, si observamos su hoja de datos, su consumo de potencia es de 1,05 mW cuando se encuentra en operación [19]. Para el módulo Waspote en reposo su consumo es de 205  $\mu$ W, y en operación de 30 mW. Los módulos de Xbee en el momento de transmitir y recibir datos su gasto de energía es de 1 mW, y en el momento que se encuentra en modo de reposo es de 330  $\mu$ W [20]. En el caso del nodo central, la tarjeta de desarrollo Raspberry Pi 3, su consumo en reposo es cercano a 1,2 W y a pleno rendimiento es de 1,8 W [25]. Con los datos anteriores, se puede afirmar que el componente que más consume energía es el microcontrolador del nodo central, por las características que ofrece y por el alto desempeño que tiene en el momento de su funcionamiento. Por lo que es importante seleccionar una batería con bastante capacidad que soporte estos niveles de potencia. Pero en general como se muestra en la Tabla 12, el sistema

muestra un gasto de energía bajo que es lo que se busca en las implementaciones de redes de sensores autónomas.

Tabla 12. Resumen del consumo de potencia del sistema.

<b>Componente</b>	<b>En reposo</b>	<b>Activo</b>
<b>ADXL 335</b>	105 $\mu$ W	1,5 mW
<b>Waspote</b>	205 $\mu$ W	30 mW
<b>Xbee</b>	330 $\mu$ W	1 mW
<b>Raspberry Pi 3</b>	1,2 W	1,8 W

## Capítulo 6. Conclusiones y recomendaciones

### 6.1 Conclusiones

- El sistema de medición de aceleración de vibraciones con el sensor ADXL335 presenta un 38,31% de error en comparación con el sensor BDI 2120 comercial.
- El consumo de potencia del sistema completo durante el estado de medición, transmisión y almacenamiento de datos es de 1,8 W.
- La distancia de comunicación entre los sensores y el sistema central de almacenamiento de datos es de máximo 10 metros con obstáculos y de 50 metros sin ellos.
- Los datos de medición se almacenan correctamente en archivos de texto separados, uno por cada nodo de la red en el momento que el usuario realice una medición.
- La capacidad de la velocidad de transmisión de datos del Waspote utilizado en el proyecto no permite obtener una frecuencia de muestreo suficiente para realizar un análisis espectral de los datos de vibración en las estructuras tipo puente.
- La red de sensores se implementó con los tres nodos secundarios y el nodo central, permitiendo la comunicación entre ellos utilizando una topología tipo estrella en la misma.

## 6.2 Recomendaciones

- Implementar en el circuito del acondicionamiento de señal un seguidor de tensión con alimentación externa, para que el módulo ADXL 335 opere con su tensión nominal de 3,3 V. Esto permitiría darles mayor longitud a los sensores desde el punto de colocación hacia la conexión en el nodo.
- Utilizar otros acelerómetros que ofrezcan mejor precisión y sensibilidad ante los cambios de posición, por el tipo de aplicación, se sugiere utilizar los que tienen capacidad a un rango menor de 2g.
- A pesar que las placas de desarrollo Wasp mote están diseñadas para utilizarlas en redes de monitoreo de sensores, estas no permiten realizar aplicaciones a altas velocidades de transmisión, por lo que se sugiere cambiar las tarjetas de desarrollo para la implementación de los nodos secundarios, de forma que sean capaces de procesar la información a tasas de transferencias entre los rangos de Mbps o Gbps, utilizando como medio de transmisión dispositivos de tecnología Wifi, como por ejemplo con el protocolo IEEE 802.11.
- Para la programación en los nodos secundarios, optar por dos subprocesos independientes, uno para el almacenamiento local y otro para la transmisión en tiempo real, esto para que la lentitud de la rutina del procesamiento no afecte el almacenamiento de datos local.
- En la alimentación de la red, se puede implementar el uso de pequeñas celdas solares para aprovechar la energía y reducir la dependencia de las baterías.
- Elaborar un sistema de almacenamiento de datos a nivel de un servidor en internet, que permita guardar las mediciones en tiempo real con el objetivo de ofrecer un respaldo de la información para su posterior análisis.
- Para la interacción con el usuario, mejorar la interfaz gráfica, la cual permita visualizar los datos en tiempo real por medio de tabulaciones y por medio de gráficas respecto al tiempo de duración de las mediciones. Además de usar una pantalla de mayor tamaño para facilitar la visualización de los datos.

## Referencias bibliográficas

- [1] J. Carvajal, ““ Desarrollo tecnológico de un sistema de adquisición de datos ambientales para su uso en proyectos de investigación científica : Arquitectura abierta CRTecMote ””, Instituto Tecnológico de Costa Rica, 2010.
- [2] F. Navarro y D. Ureña, “Mediciones y análisis de vibraciones en el puente del Virilla”, *Investig. TEC*, núm. 16, 2013.
- [3] Y. Arias, “Sistema de Tiempo Real para el Análisis de Vibración en Puentes”, Instituto Tecnológico de Costa Rica, 2015.
- [4] H. Wenzel, *Health Monitoring of Bridges*, 1a ed. Viena, Austria: Wiley, 2009.
- [5] O. S. Salawu y C. Williams, “Bridge Assessment Using Forced-Vibration Testing”, *J. Struct. Eng.*, vol. 121, núm. 2, pp. 161–173, 1995.
- [6] V. Meruane, “Vibraciones Mecánicas”, p. 10.
- [7] H. Varas, J. Loayza, y A. Ibañez, “Vibraciones de estructuras”, Buenos Aires, 2001.
- [8] G. F. Galindo, “Construcción y Validación de un Sensor de Vibraciones usando un Sistema Micro-Electrico-Mecanico (MEMS)”, Universidad Austral de Chile, 2010.
- [9] M. Arenas, “Sistema Para la Adquisición y monitorización de aceleración mediante microprocesador”, *Univ. Sevilla*, pp. 39–54, 2008.
- [10] P. Ochoa, M. Villegas, J. L. Pons, M. A. Bengochea, y J. F. Fernández, “Piezocomposites metal-cerámica como elementos activos en acelerómetros”, *Bol. la Soc. Esp. Ceram. y Vidr.*, vol. 41, núm. 1, pp. 126–130, 2002.
- [11] V. Kaajakari, “MEMS Tutorial: Pull-In Voltage in Electrostatic Microactuators”, *Pract. MEMS*, núm. 5, pp. 1–2, 2009.
- [12] Z. Gang, “Design and Simulation of A CMOS-MEMS Accelerometer”, B.S, Tsinghua University, 1994.
- [13] NI, “¿Qué es una Red de Sensores Inalámbricos?”, *National Instruments*, 2009. [En línea]. Disponible en: <http://www.ni.com/white-paper/7142/es/>. [Consultado: 19-oct-

2017].

- [14] D. Culler, D. Estrin, y M. Srivastava, “Overview of sensor networks”, *Computer*, vol. 37, núm. 8, pp. 41–49, 2004.
- [15] D. Sharma, S. Verma, y K. Sharma, “Network Topologies in Wireless Sensor Networks : A Review”, *Int. J. Electron. Commun. Technol.*, vol. 4, pp. 93–97, 2013.
- [16] I. Howitt y G. Jore A., “IEEE 802.15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)”, *Wirel. Commun. Netw.*, vol. 2011, núm. September, p. 314, 2011.
- [17] I. Howitt y J. A. Gutierrez, “IEEE 802.15.4 low rate - wireless personal area network coexistence issues”, *2003 IEEE Wirel. Commun. Networking, 2003. WCNC 2003.*, vol. 3, núm. C, pp. 1481–1486, 2003.
- [18] J. Proakis y D. Manolakis, *Tratamiento Digital de Señales*, 4<sup>o</sup> edición. Madrid: Pearson Prentice Hall, 2007.
- [19] I. Studio, “ADXL335 Module”, 2011. [En línea]. Disponible en: [http://store.iteadstudio.com/images/produce/Sensor/ADXL335Module/ADXL335Module\\_DS.pdf](http://store.iteadstudio.com/images/produce/Sensor/ADXL335Module/ADXL335Module_DS.pdf). [Consultado: 13-sep-2017].
- [20] Libelium, “Waspote Datasheet.”, 2009. [En línea]. Disponible en: [http://www.libelium.com/v11-files/documentation/waspote/waspote-datasheet\\_eng.pdf](http://www.libelium.com/v11-files/documentation/waspote/waspote-datasheet_eng.pdf).
- [21] Ingeniería MCI, “Que es Xbee?” [En línea]. Disponible en: [www.xbee.cl/que-es-xbee/](http://www.xbee.cl/que-es-xbee/). [Consultado: 20-sep-2017].
- [22] B. Mitchell, “What’s MAC Address and How Does It Word?” [En línea]. Disponible en: <https://www.lifewire.com/introduction-to-mac-addresses-817937>. [Consultado: 10-oct-2017].
- [23] Sparkfun, “SparkFun XBee Explorer USB - WRL-11812 – SparkFun Electronics”, *Sparkfun.com*. [En línea]. Disponible en: <https://www.sparkfun.com/products/11812>. [Consultado: 02-oct-2017].
- [24] CRCibernética, “Raspberry Pi”, *CRCibernética*. [En línea]. Disponible en:

<http://www.crcibernetica.com/raspberry-pi-3/>. [Consultado: 13-sep-2017].

- [25] Raspberry Pi Foundation, “RASPBERRY PI 3 MODEL B”, *RaspberryPi*. [En línea]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Consultado: 08-sep-2017].
- [26] CRCibernética, “3.5 inch Pi TFT LCD Touchscreen(320\*480)”, *CRCibernética*. [En línea]. Disponible en: <http://www.crcibernetica.com/3-5-inch-pi-tft-lcd-touchscreen-320-480/>. [Consultado: 10-oct-2017].
- [27] Libelium, “Waspote 802.15.4 Networking Guide”. [En línea]. Disponible en: [http://www.libelium.com/v11-files/documentation/waspote/waspote-802.15.4-networking\\_guide.pdf](http://www.libelium.com/v11-files/documentation/waspote/waspote-802.15.4-networking_guide.pdf).
- [28] Digi, “Xbee API MODE”, *Digi International*, 2015. [En línea]. Disponible en: <http://docs.digi.com/display/RFKitsCommon/XBee+API+mode>. [Consultado: 07-nov-2017].
- [29] S. Kim, T. Oberheim, S. Pakzad, D. Culler, J. Demmel, y G. Fennes, “Structural Health Monitoring of the Golden Gate Bridge”, 2003.
- [30] Adafruit, “Powerboost 1000C”. [En línea]. Disponible en: <https://learn.adafruit.com/adafruit-powerboost-1000c-load-share-usb-charge-boost/overview>. [Consultado: 20-oct-2017].
- [31] BDI, “Accelerometer Operations Manual”, *V 1.2*, 2013. [En línea]. Disponible en: <http://bditest.com/wp-content/uploads/Accel-Operations-Manual-v1.2.pdf>. [Consultado: 17-nov-2017].

# Apéndices

## A .1 Manual de usuario

Este manual de usuario explica los pasos para utilizar el sistema de la red de sensores de vibración, les sugerimos tener a mano la hoja de datos del Waspote que la encuentran en la referencia [20].

### A.1.1 Conexión de los componentes del sistema

Para iniciar con el uso de la red de sensores los primeros pasos es la conexión de los diferentes componentes del sistema:

1. Conectar el módulo del acelerómetro a la placa Waspote, según los pines indicados en la Tabla 4. Los colores de los cables corresponden a las siguientes señales:
  - Verde: Referencia (GND)
  - Azul: Alimentación 5 V
  - Rojo: salida eje x
  - Anaranjado: salida eje y
  - Amarillo: salida eje z
  - Café: sin conexión

La conexión debe quedar como se observa en la Figura 43.

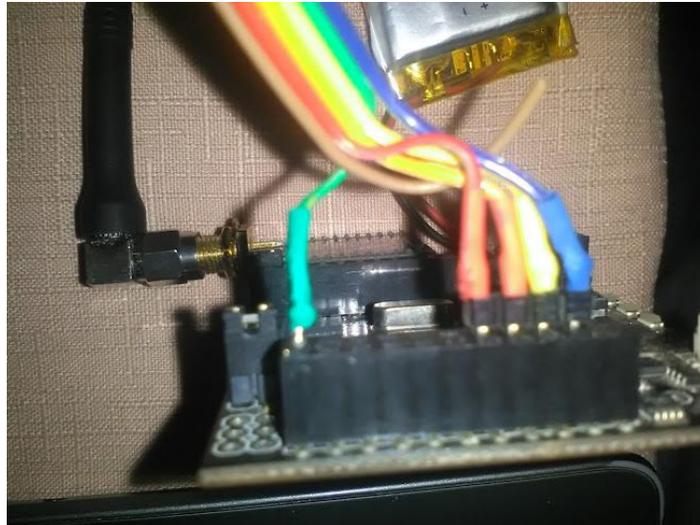


Figura 43. Conexión de pines en el Wasp mote.

2. Conectar la batería en el puerto mini JST que se encuentra en uno de los costados del Wasp mote, como se observa en la Figura 44.



Figura 44. Conexión de batería en el Wasp mote.

3. Colocar la pantalla táctil en los pines GPIO de la Raspberry Pi, como se muestra en la siguiente figura:



Figura 45. Conexión de la pantalla táctil a la Raspberry Pi 3.

4. Conectar en el puerto 0 USB (configurado el script de Python en ese puerto) de la Raspberry Pi el módulo del Xbee USB Explorer, utilizando un cable USB tipo A a mini tipo B.



Figura 46. Conexión al puerto USB de la Raspberry Pi 3.

5. Conecte la batería al convertidor DC-DC en el puerto JST y con la ayuda de un cable USB tipo A a tipo B micro, se conecta al puerto de alimentación de la Raspberry Pi 3, para que el sistema quede conectado como la Figura 33.

### **A.1.2 Programación de los nodos**

Para programar los nodos de los Wasmote, se debe utilizar el software Wasmote IDE V02, el cual se encuentra disponible en la página [www.libelium.com](http://www.libelium.com) . Para compilar y actualizar las placas con el script implementado, primero se debe conectar la tarjeta de desarrollo por medio de un cable USB tipo A a tipo B mini, además que no se debe tener empotrado el módulo Xbee, de lo contrario a la hora de la compilación nos mostrará un mensaje con un error.

Se debe encender el módulo en el switch que se encuentra a un costado de la placa, posteriormente se carga el script desarrollado, el cual estará disponible junto a este documento y tiene una extensión .pde. Dependiendo del nodo al cual se le subirá el código, en el script en la parte donde se encuentra la función “sprintf” se debe cambiar el número de nodo correspondiente. Posteriormente se presiona el botón de cargar. El tiempo de compilación y subida del código es aproximadamente de 2 minutos para que finalice. Luego de haber concluido la carga, se apaga el Wasmote y se vuelve a colocar el Xbee en su lugar. Se repite el proceso para los tres nodos de la red.

### **A.1.2 Ejecución del programa**

En el caso del nodo central, en el momento que se le conecta la batería, este enciende y como pantalla principal se encuentra el escritorio. Ahí mismo se encuentra un archivo denominado “red\_sensores.py”, el cual está compilado con el lenguaje de programación Python, como se observa en la Figura 47.

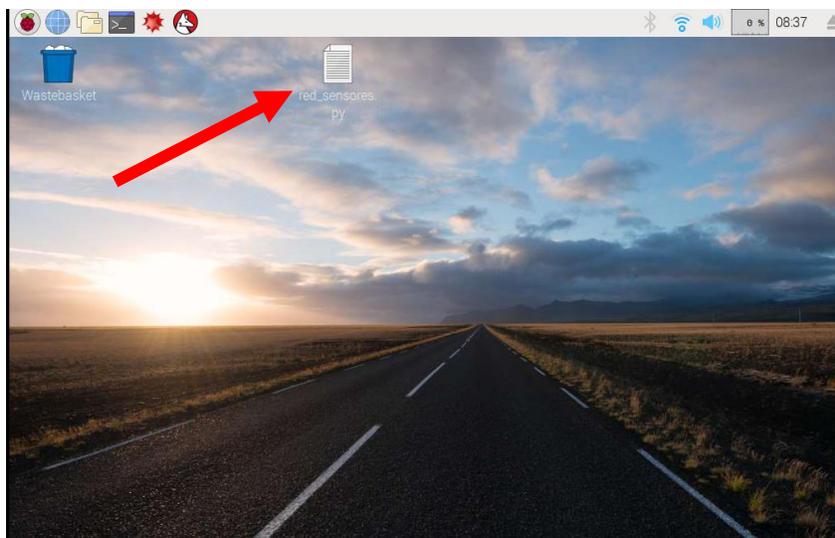


Figura 47. Ubicación del script de Python de la red de sensores.

Al darle doble click al archivo se abrirá la interfaz de Python y en el menú de “Run”, se escoge la opción “Run module”. En este momento se abre la ventana de interfaz gráfica como se mostró en la Figura 27, por lo que antes de seguir se deben encender los Wasp mote, para estar listos con el inicio de la transmisión de los datos. En la ventana de interfaz se muestran los dos botones, el de “Activar” y “Apagar”, al oprimir el botón de Activar, se muestra un mensaje emergente que indica “Sistema Activado” y en ese momento inicia la transferencia de datos y la creación de los archivos de texto por cada nodo de la red, los cuales se ubican en la misma dirección en donde se encuentre el script de ejecución, como se muestra en la Figura 36. Luego si oprime el botón de Apagar, se muestra una ventana emergente que indica “Sistema apagado” y ahí finaliza la transmisión de los datos. Si se desea volver activar accionar la red, se vuelve a oprimir el botón Activar.

Además de los archivos generados en el nodo central, si el Wasp mote tiene conectada una memoria micro SD, se crea un archivo tipo texto con los datos obtenidos igualmente cuando el sistema esta activado. Brindando un respaldo local en caso si se pierde la comunicación entre el nodo central y el auxiliar.

### **A.1.3 Interpretación de los datos**

En el momento que el sistema se encuentra activado, una opción para verificar si se están obteniendo los datos de los nodos es minimizar la ventana de interfaz gráfica y se observa la consola de Python y ahí se mostrarán en tiempo real las mediciones que se están obteniendo, como se muestra en la Figura 35.

Utilizando cualquier editor de texto o de hojas de cálculo, se abren los archivos generados y los valores que se muestran corresponden según el orden a lo siguiente:

- Identificador: nodo\_1, nodo\_2 o nodo\_3.
- Valor de aceleración en x.
- Valor de aceleración en y.
- Valor de aceleración en z.
- Día de medición.
- Fecha de medición.
- Hora de medición.

### **A.1.4 Cierre de interfaz y del sistema**

Para finalizar con la rutina del script, posteriormente de haber presionado el botón Apagar, se cierra la ventana de la interfaz. Luego se apagan los módulos Waspote por medio del switch y la Raspberry se puede realizar de dos maneras utilizando un teclado y en la terminal poner el comando `sudo shutdown -h now` o sino desconectando el cable de alimentación.

### **A.1.5 Posibles problemas y soluciones**

- Si el nodo central no enciende, revisar el nivel de carga de la batería, en el convertidor si se enciende un led rojo indica que la batería debe ser recargada.
- Si los Wasmote no encienden, revisar la conexión de la batería y la carga de las mismas.
- La interfaz gráfica inicia, pero no se visualiza que se están recibiendo datos, primero verificar si los Wasmote están encendidos, sino desconectar el módulo de Xbee USB Explorer y volver a conectarlo, e intentar de nuevo.
- Se están obteniendo los datos, pero en cierto momento se detiene la recepción, en ese caso reiniciar el sistema cerrando la interfaz gráfica y Python, y se vuelve a iniciar nuevamente.
- La pantalla táctil no enciende o se queda pegada; utilizar la Raspberry Pi, mediante una conexión de acceso remoto vía SSH. Acceder a la siguiente [link](#) y seguir los pasos.

### **A.1.6 Materiales que conforman el sistema**

- 1 Raspberry Pi 3
- 3 módulos de acelerómetros ADXL335
- 3 módulos Wasmote v 1.1
- 4 Xbee 802.15.4 S1
- 1 Xbee USB Explorer
- 1 pantalla táctil capacitiva TFT para Raspberry de 3,5 “
- 3 baterías tipo LIPO de 3,7 V/1250 mAh
- 1 batería tipo LIPO de 3,7 V/2500 mAh
- 1 Convertidor DC-DC Power Boost 1000 C de 3,7 V a 5 V
- Cable tipo cinta de 6 líneas



Figura 48. Componentes utilizados para la red de sensores

### **A.1.7 Control de versiones y contacto**

Versión de la implementación:

V 1.1

Contacto para más información:

Omar Vargas Ruiz

Teléfono: (506) 89168288

Correo: [omarvargasruiz@gmail.com](mailto:omarvargasruiz@gmail.com)

## **A .2 Información de la empresa**

**Nombre:** Centro de Investigación en Vivienda y Construcción

**Zona:** Cartago, Cartago, Oriental, Los Ángeles

**Dirección:** Tecnológico de Costa Rica, sector este del campus

Teléfono: 2550-2309 Fax: no Apartado: 30101

**Actividad Principal:** Investigación y extensión de construcción.

### Información del encargado en la empresa:

**Nombre:** Giannina Ortiz Quesada

**Puesto que ocupa:** Coordinadora proyecto e-Bridge

**Departamento:** CIVCO, Escuela Ingeniería en Construcción

**Profesión:** Ingeniera en Construcción **Grado académico:** Maestría

Teléfono:25502423 Ext.: 2423

**Email:** gortiz@tec.ac.cr

### Información del asesor en la empresa:

**Nombre:** Ángel Navarro Mora

**Puesto que ocupa:** Encargado de pruebas en estructuras

**Departamento:** Evaluación de puentes y estructuras

**Profesión:** Ingeniero en Construcción **Grado académico:** Licenciatura

Teléfono:25502246 Ext.:2246

**Email:** ahnavarro@tec.ac.cr