



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Postgrado
Maestría en Estructuras

**DISEÑO DE UN SISTEMA DE MONITOREO DE SALUD ESTRUCTURAL (SHM)
PARA EDIFICIOS DE MARCOS DE CONCRETO REFORZADO, BASADOS EN
FRECUENCIAS DE VIBRACIÓN UTILIZANDO SENSORES DE ACELERACIÓN**

Ing. Juan Adolfo Orózco Coloma

Asesorado por el MSc. Mario Enrique Rosada Orozco

Guatemala, junio de 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UN SISTEMA DE MONITOREO DE SALUD ESTRUCTURAL (SHM)
PARA EDIFICIOS DE MARCOS DE CONCRETO REFORZADO, BASADOS EN
FRECUENCIAS DE VIBRACIÓN UTILIZANDO SENSORES DE ACELERACIÓN**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA FACULTAD DE INGENIERÍA

POR:

ING. JUAN ADOLFO ORÓZCO COLOMA

ASESORADO POR EL MSc. ING. MARIO ENRIQUE ROSADA OROZCO

AL CONFERÍRSELE EL TÍTULO DE

MAESTRO EN ESTRUCTURAS

GUATEMALA, JUNIO DE 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
DIRECTOR	Mtro. Ing. Edgar Darío Álvarez Cotí
EXAMINADOR	Mtro. Ing. Armando Fuentes Roca
EXAMINADOR	Mtro. Ing. José Estuardo Galindo Escobar
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**DISEÑO DE UN SISTEMA DE MONITOREO DE SALUD ESTRUCTURAL (SHM)
PARA EDIFICIOS DE MARCOS DE CONCRETO REFORZADO, BASADOS EN
FRECUENCIAS DE VIBRACIÓN UTILIZANDO SENSORES DE ACELERACIÓN**

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Postgrado, con fecha 5 de septiembre de 2019.

Ing. Juan Adolfo Orózco Coloma

ACTO QUE DEDICO A:

Dios	Agradecido por permitirme llegar a esta meta, a él sea la honra y gloria.
Universidad de San Carlos de Guatemala	Por ser la mejor universidad de Guatemala, y por darme las herramientas necesarias en mi vida académica y profesional.
Mis padres	Con gratitud y amor por el apoyo incondicional.
Mis hermanos	Con gratitud, por todas las alegrías que hemos tenido durante este tiempo.
Mi familia	Abuelos, tíos y primos, por el apoyo en los momentos alegres y difíciles.
Mi novia	Por su apoyo incondicional durante y hasta culminar mi carrera profesional.
Mis amigos	Por su apoyo y por los buenos momentos que pasamos en la universidad y su apoyo.

AGRADECIMIENTOS A:

Dios	Por la sabiduría e inteligencia que me brindó para terminar el proceso; por cuidar y ayudar en los momentos difíciles.
Universidad de San Carlos de Guatemala	Por la formación académica recibida en cada uno de los semestres.
Mis padres	Por su sacrificio para que no me faltara nada en el transcurso de la carrera. A mi padre por su ayuda en la elaboración de este trabajo de graduación y a mi madre por ser siempre mi apoyo.
Mis hermanos	Adolfo y Liz Orózco, por su apoyo en los momentos difíciles y alegres.
Mi familia	Abuelos, tíos y primos, por el apoyo en todo momento.
Mi novia	Por su apoyo incondicional, especialmente en la realización de este trabajo y por el tiempo juntos.
Mis amigos	Por el acompañamiento en mi vida universitaria.

Ing. Mario Rosada

Por el apoyo y asesoría en mi etapa de estudiante, especialmente en la realización de este trabajo de graduación.

**Mis docentes de la
Escuela de Estudios
de Postgrado**

Por ser parte fundamental de mi crecimiento profesional y personal.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
TABLAS	IX
LISTA DE SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN	XV
PLANTEAMIENTO DEL PROBLEMA	XVII
OBJETIVOS	XXI
INTRODUCCIÓN	XXIII
1. MARCO METODOLÓGICO	1
1.1. Monitoreo de salud estructural (SHM)	1
1.2. Métodos de detección de daños	3
1.2.1. Método de variación de frecuencia de vibración	3
1.2.1.1. Problema hacia adelante	3
1.2.1.2. Problema inverso	4
1.2.2. Métodos de cambio de formas modales	4
1.2.3. Métodos de curvatura/deformación	5
1.2.4. Métodos basados en mediciones dinámicas de flexibilidad	5
1.2.5. Métodos no lineales	6
1.3. Aplicación de SHM en vigas y marcos	6
1.3.1. Vigas de concreto	7
1.3.2. Marcos de acero	7
1.4. Sensores utilizados en SHM	7
1.4.1. Sensores de deformación por nanotubos de carbono	8

1.4.2.	Transductores piezocerámicos	9
1.5.	Microcontroladores	10
1.6.	Conectividad de sensores con el microcontrolador	11
2.	PROGRAMACIÓN DE SENSOR Y MICROCONTROLADOR	13
2.1.	Comparación de sensores de aceleración.....	13
2.1.1.	Sensor ADXL345	13
2.1.2.	Características de medición.....	14
2.1.2.1.	Parámetros de entrada	14
2.1.3.	Sensibilidad	16
2.1.4.	Costo	17
2.1.5.	Compatibilidad	17
2.1.6.	Sensor MPU4050	19
2.1.7.	Características de medición.....	19
2.1.8.	Costo	20
2.1.9.	Compatibilidad	21
2.2.	Comparación de microcontroladores	21
2.2.1.	Costo	23
2.3.	Desarrollo de código de programación de sensor	23
2.4.	Calibración de sensor de aceleración.....	24
2.4.1.	Desarrollo de código para medición de aceleración .	29
2.4.2.	Cálculo de parámetros de ajuste de sensor.....	35
2.4.3.	Cálculo de ruido y sensibilidad	38
3.	EXPERIMENTACIÓN DE DISPOSITIVO DE MEDICIÓN	41
3.1.	Instalación de arreglo de sensores en edificio de marcos de concreto.....	41
3.2.	Monitoreo de largo plazo	44
3.3.	Procesamiento de datos	44

3.4.	Comparación de frecuencias de vibración teóricas y experimentales	46
3.4.1.	Selección de sismo de estudio y extracción de datos... ..	47
3.4.2.	Modelado del edificio de estudio.....	48
3.4.3.	Consideraciones del suelo.....	51
3.4.4.	Carga de aceleraciones al programa	52
3.4.5.	Cálculo de periodo de vibración teórico.....	53
3.4.6.	Cálculo de periodo de vibración experimental	55
3.4.7.	Comparación de periodos teóricos y experimentales.....	55
3.5.	Inferencia de daños estructurales y su localización.....	56
3.6.	Selección de método de análisis de daños.....	57
3.6.1.	Estado límite totalmente operacional	58
3.6.2.	Estado límite operacional.....	58
3.6.3.	Estado límite seguridad	59
3.6.4.	Estado límite precolapso.....	59
4.	DESARROLLO DE SOFTWARE DE DETECCIÓN DE DAÑOS	61
4.1.	Procesamiento de datos de arreglo de microcontroladores.....	61
4.2.	Programa de obtención de datos.....	62
4.3.	Programa de procesamiento de datos y conversión a parámetros estructurales	67
4.3.1.	Filtro Savitzky–Golay	68
4.3.2.	Periodos y frecuencias.....	70
4.3.3.	Velocidades y desplazamientos.....	72
4.4.	Creación de interfaz gráfica	72
4.4.1.	Interfaz de programa de obtención de datos	72
4.4.2.	Interfaz de procesamiento de datos y monitoreo estructural	74

4.5.	Prueba de ejecutable.....	75
5.	MANUAL DE INSTALACIÓN Y OPERACIÓN DE DISPOSITIVOS.....	79
5.1.	Información general de los dispositivos	79
5.2.	Partes del sensor.....	80
5.2.1.	Caja	80
5.2.2.	Microcontrolador Arduino Uno	81
5.2.3.	Sensor de aceleración	83
5.2.4.	Módulo microSD	84
5.2.5.	Cables y convertidores de voltaje	85
5.2.6.	Cinta doble cara.....	86
5.2.7.	Memoria microSD	87
5.3.	Ensamblaje del sensor	88
5.4.	Calibración de dispositivos	89
5.5.	Carga de código a microcontrolador.....	90
5.6.	Diseño de arreglo de sensores en un edificio.....	91
5.7.	Colocación de sensores en estructura.....	92
6.	PRESENTACIÓN Y ANÁLISIS DE RESULTADOS	95
6.1.	Análisis de resultados obtenidos	95
6.1.1.	Tiempo historia lineal.....	95
6.1.2.	Comparación de aceleraciones teóricas y experimentales por nivel.....	96
6.1.3.	Comparación de periodos de vibración e inferencia de daños	100
6.2.	Identificación de puntos problemáticos.....	100
6.2.1.	Calibración de sensores	101
6.2.2.	Consideraciones de estructura	101
6.2.3.	Fallas en fuentes de poder	101

6.2.4.	Sismos de baja magnitud.....	101
	CONCLUSIONES	103
	RECOMENDACIONES	105
	REFERENCIAS.....	107
	APÉNDICES	113
	ANEXOS	125

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Set de sensores de deformación de nanotubos de carbono en una viga en voladizo.....	9
2.	Arduino montado con XBee RF y XBee Shield	11
3.	Diagrama de bloque de sensor ADXL345	14
4.	Costo de sensor ADXL345 en Guatemala.....	17
5.	Diagrama de conexión de sensor ADXL345 con microcontrolador Arduino UNO	18
6.	Costo de sensor MPU 6050 en Guatemala	21
7.	Costo de sensores MPU6050 y ADXL335 en Guatemala	23
8.	Diagrama de sistema MEMS para medición de aceleración	24
9.	Diagrama de aceleraciones en un cuerpo inclinado.....	33
10.	Datos filtrados con una constante de 0.80	39
11.	Datos filtrados con una constante de 0.50	39
12.	Datos filtrados con una constante de 0.25	40
13.	Edificio monitoreado por arreglo de sensores	42
14.	Colocación de sensores en la estructura.....	43
15.	Instalación de sensores en estructura.....	43
16.	Ejemplo de datos extraídos de sensores	45
17.	Sismos registrados en el periodo de monitoreo	48
18.	Asignación de resortes de confinamiento de suelo	50
19.	Modelo de edificio en programa de análisis estructural.....	51
20.	Carga de aceleraciones como tiempo historia.....	52
21.	Caso de carga tiempo historia.....	53
22.	Estados límites de desempeño de estructuras.....	58

23.	Comparación de periodos con momento de inercia reducido en columnas	60
24.	Aceleraciones sin filtrar y filtradas por medio del filtro de Savitzky-Golay	69
25.	Creación de interfaz de obtención de datos	73
26.	Interfaz de monitoreo de salud estructural	74
27.	Creación de ejecutable de programa	75
28.	Programa de obtención de datos en funcionamiento	76
29.	Programa de monitoreo de salud estructural	77
30.	Sensor de aceleración en funcionamiento	80
31.	Configuraciones de impresión 3D de la caja	81
32.	Partes de microcontrolador Arduino UNO	82
33.	Sensor MPU-6050	83
34.	Módulo microSD	84
35.	Cable USB tipo A para alimentación	85
36.	Convertidor de voltaje de 5V y 2A	86
37.	Memoria microSD de 16 GB y adaptador SD	87
38.	Diagrama de conexiones de sensor	88
39.	Ensamblaje de componentes en caja	89
40.	Monitor serial de calibración del sensor	90
41.	Código cargado en software Arduino IDE para ser cargado al sensor ...	91
42.	Idealización de concentraciones de masas por nivel utilizada	92
43.	Colocación de sensores en estructura	93
44.	Aceleraciones por piso de análisis tiempo historia	96
45.	Aceleraciones nivel 1	97
46.	Aceleraciones nivel 2	97
47.	Aceleraciones nivel 3	98
48.	Aceleraciones nivel 4	98
49.	Aceleraciones nivel 5	99

TABLAS

I.	Especificaciones de parámetros de entrada sensor ADXL345.....	15
II.	Especificaciones de parámetros de ruido sensor ADXL345.....	15
III.	Especificaciones de parámetros de sensibilidad sensor ADXL345	16
IV.	Parámetros de entrada sensor MPU 6050	20
V.	Comparación de desempeño de acelerómetros ADXL335 y MPU6050 ...	22
VI.	Datos sin procesar obtenidos del sensor en reposo.....	36
VII.	Datos de aceleración normalizados	37
VIII.	Hora de inicio de registros de sensores	44
IX.	Localización de errores de continuidad en sensores.....	45
X.	Sobrecargas muertas	49
XI.	Cargas de tabiquería último nivel	49
XII.	Cálculo de módulo de Winkler.....	52
XIII.	Cálculo de periodo con fórmula empírica de AGIES	54
XIV.	Periodos y frecuencias de vibración encontradas en programa de análisis estructural.....	54
XV.	Periodos calculados de datos de los sensores.....	55
XVI.	Comparación de periodos teóricos y experimentales.....	56
XVII.	Comparación numérica de datos teóricos y experimentales	99

LISTA DE SÍMBOLOS

Símbolo	Significado
G	Aceleración debido a la gravedad
H	Altura
H_n	Altura de nivel
Θ_x	Ángulo en X
Θ_y	Ángulo en Y
Θ_z	Ángulo en Z
ω_i[^]	Frecuencia angular
Kg	Kilogramo
kg/m	Kilogramo sobre metro
kg/m²	Kilogramo sobre metro cuadrado
M_w	Magnitud de movimiento sísmico
M	Masa
{u}	Matriz de desplazamientos
{F}	Matriz de fuerzas
{G}	Matriz de rigidez
T_{piso}	Periodo de piso
T_f	Periodo fundamental
K	Rigidez
S	Segundo
φ^{^*}	Valores propios

GLOSARIO

Deflexión	Es el grado en que un elemento estructural se desplaza bajo una carga (debido a su deformación). Puede referirse a un ángulo o una distancia.
Deformación	Se refiere al cambio de tamaño o forma de un objeto. Los desplazamientos son el cambio absoluto de posición de un punto en el objeto. La deflexión es el cambio relativo en los desplazamientos externos de un objeto.
Ductilidad	Es una propiedad mecánica que se describe comúnmente como la facilidad de estiramiento de un material.
Elasticidad lineal	Es un modelo matemático de cómo los objetos sólidos se deforman y se estresan internamente debido a las condiciones de carga prescritas. Es una simplificación de la teoría no lineal de la elasticidad más general y una rama de la mecánica del continuo.
Frecuencia	Es el número de repeticiones por unidad de tiempo, de cualquier evento periódico.

Frecuencia angular

Es una medida escalar de la velocidad de rotación. Se refiere al desplazamiento angular por unidad de tiempo o la tasa de cambio de la fase de una forma de onda sinusoidal.

Microcontrolador

Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Periodo de vibración

El periodo fundamental de vibración de un edificio es un parámetro clave para el diseño sísmico de una estructura; en el caso de construcciones existentes es un valor que puede ayudar a determinar las condiciones actuales de seguridad estructura.

Sensor

Es todo aquello que tiene una propiedad sensible a una magnitud del medio; al variar esta magnitud también varía con cierta intensidad la propiedad, es decir, manifiesta su presencia y también su medida.

RESUMEN

El monitoreo de salud estructural o SHM por sus siglas en inglés, es un tipo de monitoreo que comenzó en las ramas de ingeniería mecánica y aeroespacial; sin embargo, se ha expandido a la ingeniería civil en los últimos años.

En esta investigación se presenta una alternativa de monitoreo estructural para edificios de marcos de concreto reforzado. Se estudian los tipos de sensores y microcontroladores que existen en el mercado de electrónicos de Guatemala y se comparan las características técnicas de cada uno; asimismo, en relación con el precio, se selecciona la mejor opción del mercado, se desarrolla el código de programación para hacer funcionar los sensores y almacenar datos a una memoria.

Se diseña un programa ejecutable que permite extraer los registros de aceleraciones deseados de la memoria de los sensores, y posteriormente muestra los periodos y frecuencias de vibración de cada nivel; luego se explica una metodología para inferir el estado límite en el que se encuentra el edificio y verificar si es necesario realizar alguna revisión más profunda.

PLANTEAMIENTO DEL PROBLEMA

Actualmente, en Guatemala no se cuenta en el mercado con dispositivos accesibles para el monitoreo de salud estructural (SHM) para edificios construidos con marcos de concreto.

Los edificios de concreto reforzado, pueden tener tanto errores de cálculo en el diseño como errores de campo en construcción. No existen dispositivos de monitoreo de salud estructural (SHM) en el mercado local guatemalteco, y resulta costosa la importación de los mismos; adicionalmente, no existe normativa o manual para el uso correcto e interpretación de datos de los sensores.

La tecnología de monitoreo de salud estructural tuvo su origen en la industria aeroespacial para medir vibraciones de las aeronaves, siendo relativamente nueva en la ingeniería civil en Guatemala; pero es muy común en Estados Unidos y Europa. Estos dispositivos, a pesar de la gran utilidad que tendrían en una región altamente sísmica como Guatemala, no están disponibles en el mercado guatemalteco y el costo tanto de importación como de mano de obra calificada para su uso no justifica la inversión.

Una de las razones del poco uso de estos dispositivos de monitoreo es que, en los contenidos de estudio de las universidades de Guatemala, no es prioridad el uso e importancia de los dispositivos; los planes de estudios se enfocan en el diseño y supervisión, debido principalmente a la falta de literatura en español sobre este tema de la ingeniería estructural.

Para la elaboración de dispositivos con capacidad de medición de vibraciones ambientales y posterior conversión de señales análogas y digitales a parámetros estructurales, se necesita tanto un amplio conocimiento de dinámica estructural y el comportamiento de los materiales, como del conocimiento de electrónica, microcontroladores y sensores, lo cual no es común en desarrolladores de dispositivos electrónicos e ingenieros estructurales a la vez.

Gran parte del trabajo consiste en la colocación adecuada de los sensores en la estructura para obtener datos confiables de los parámetros estructurales. Sin embargo, esto no es posible si no existe una guía o normativa que permita la correcta recolección de datos. Como consecuencia de esta problemática surge la pregunta central:

¿Se puede crear un sistema de monitoreo de salud estructural (SHM) para edificios de marcos de concreto reforzado utilizando como parámetros estructurales las frecuencias de vibración?

Preguntas secundarias:

¿Qué sensibilidad a la vibración en edificios tendrán los sensores de aceleración compatibles con los microprocesadores Arduino?

¿Cuánta será la variación entre el periodo de vibración teórico y el periodo de vibración experimental utilizando los datos recolectados con el sistema de monitoreo?

¿Cuál será la mejor configuración de acelerómetros en una estructura de marcos de concreto para recolectar mejores datos de vibración?

¿Es posible inferir fallas internas a los elementos de concreto con los cambios de frecuencia de vibración de una estructura a lo largo del tiempo?

OBJETIVOS

General

Diseñar un sistema de monitoreo de salud estructural (SHM) para edificios de marcos de concreto reforzado, basados en frecuencias de vibración utilizando sensores de aceleración

Específicos

- Establecer un dispositivo electrónico con capacidad de medir datos de vibración estructural con partes disponibles en Guatemala.
- Analizar las posibles fallas en base al análisis de datos de vibración comparando los modos y periodos de vibración teórico y experimental.
- Implementar un software de procesamiento de datos de vibración y conversión a periodos, frecuencias y modos de vibración.
- Definir una metodología para la inferencia de daños en elementos estructurales en base a los datos que se extraen de los sensores de aceleración.
- Elaborar un manual de instalación de dispositivos de medición de aceleración en edificios de concreto

INTRODUCCIÓN

El estudio del *Monitoreo de salud estructural* (SHM) se originó en la década de los 90s en la ingeniería mecánica y la aeroespacial. Se inició como un estudio de vibraciones para monitoreo en transbordadores espaciales y aeronaves comerciales, técnica que posteriormente se fue adaptando a la ingeniería civil como el monitoreo de elementos estructurales simples como vigas y elementos compuestos como armaduras y marcos.

Existen diversos métodos para el SHM. La mayoría se basa en comparar las frecuencias de vibración de la estructura no dañada con la frecuencia de vibración de una estructura dañada. La frecuencia de vibración de una estructura depende de la rigidez de los elementos que la conforman, si existe un cambio de propiedades mecánicas o geométricas de los elementos estructurales, se espera un cambio en la frecuencia de vibración de la estructura.

Este concepto se puede extender al análisis modal. una estructura responde de manera diferente dependiendo la fuerza de excitación, los periodos de vibración que activan estos modos cambian al momento de haber algún daño en la estructura debido al cambio en la rigidez, también es posible comparar la forma que tienen los modos de vibración para inferir y localizar daños en la estructura.

En la actualidad, existen diferentes tipos de sensores para el monitoreo estructural, entre ellos están los piezoeléctricos, piezocerámicos o nanotubos de carbono. Los cuales se basan en la medición de deformaciones en la estructura para posteriormente encontrar la matriz de rigidez de la estructura para su

comparación con la matriz de rigidez ideal de la estructura sin daños, también existen acelerómetros que miden únicamente la magnitud de la aceleración en las tres direcciones.

En el primer capítulo de esta investigación se estudiaron diferentes métodos para obtener parámetros estructurales que puedan ser utilizados en la inferencia de daños estructurales, así como los sensores y microcontroladores disponibles en Guatemala, que permitan recolectar datos de vibración de una estructura, evaluando sus características y compatibilidad. En el segundo, se realiza un programa computacional que permita la depuración de datos recolectados por los sensores y su posterior conversión a parámetros estructurales.

En el tercer capítulo se explica el proceso de experimentación con el dispositivo al colocarlo en un edificio de marcos de concreto reforzado y realizar el monitoreo estructural, comparando las frecuencias de vibración teóricas y experimentales e infiriendo daños en la estructura en función a esos cambios. Para interpretar estos datos estructurales, en el cuarto capítulo se programará un software computacional, que permita comparar frecuencias de vibración encontradas mediante un arreglo de sensores con las frecuencias de vibración teóricas y permita inferir y localizar los daños en la estructura.

Con base en los resultados de la experimentación, en el quinto capítulo se realiza un manual de instalación y operación de los dispositivos, detallando la manera en la que debe cargar el código al microcontrolador, la manera de colocar el arreglo de sensores en función de la estructura a monitorear y el anclaje de los dispositivos. Finalmente, en el sexto capítulo se analiza los resultados de la investigación y se identificarán puntos problemáticos que se

observaron al utilizar los dispositivos y al analizar los datos, para plantear recomendaciones de uso para los dispositivos.

1. MARCO METODOLÓGICO

Se han desarrollado múltiples metodologías para realizar monitoreo de salud estructural, cada uno puede presentar el estudio de diferentes parámetros estructurales, estos se detallan a continuación:

1.1. Monitoreo de salud estructural (SHM)

El monitoreo de salud estructural, conocido en inglés *Structural Health Monitoring* o SHM; es una técnica reciente en la ingeniería civil, que empezó a utilizarse en aplicaciones de ingeniería mecánica e ingeniería aeroespacial. Según Farrar y Worden (2006) el daño se puede definir como un efecto adverso en el desempeño de una estructura; sin embargo, este concepto no tiene sentido si no se comparan dos estados del sistema. Para esto, uno representa el estado inicial no dañado, y el otro un estado final dañado. Por lo que, para él, los estudios en el área de la ingeniería estructural son el objetivo de investigación, y en la misma se define daño como los cambios al material o las propiedades geométricas de los mismos.

En la publicación de Farrar y Worden (2006) titulada: *Introducción al monitoreo de salud estructural*, se presenta la siguiente definición sobre monitoreo de salud estructural (SHM):

El procedimiento de implementación de metodologías y estrategias de detección de daños para elementos estructurales en diferentes ramas de la ingeniería se le llama monitoreo de salud estructural. El procedimiento consiste en observar un conjunto de elementos estructurales por tiempo

prolongado, cuantificando parámetros o variables estructurales que permitan inferir daños y definir el estado actual de la estructura. Para SHM a largo plazo, el resultado de este proceso es información actualizada periódicamente con respecto a la capacidad de la estructura para continuar desempeñando su función prevista a la luz del envejecimiento inevitable y la acumulación de daños resultantes de los entornos operativos. En un evento extremo, como un terremoto o una carga explosiva imprevista, el SHM se utiliza para la detección rápida de condiciones. Este monitoreo está destinado a proporcionar, casi en tiempo real, información confiable sobre el rendimiento del sistema durante tales eventos extremos y la posterior integridad del sistema. (p. 304)

Si bien en la actualidad existen múltiples ensayos no destructivos que utilizan tecnologías como ultrasonido, magnetismo, acústica y energía radiante, ninguno ofrece resultados en tiempo real y periódicamente actualizados.

El estudio del monitoreo estructural comenzó en la década de los 90s; como primera referencia a este tema se encuentra la investigación desarrollada por Doebling *et al.* (1996), la cual presenta documentación técnica acerca de *La detección, localización, y caracterización de daño estructural de acuerdo con técnicas que examinan cambios medidos por medio de la respuesta a la vibración*. Los métodos de análisis incluyen frecuencias modales, cambios en formas de los modos y en los coeficientes de rigidez, detección de respuesta no lineal, entre otros. Asimismo, se presentan aplicaciones en la ingeniería civil, el monitoreo de vigas, armaduras, losas o placas, puentes, así como aplicaciones en estructuras aeroespaciales y estructuras compuestas.

1.2. Métodos de detección de daños

El estudio del Laboratorio Nacional de los Álamos presenta diferentes técnicas de detección de daños. Todas consideran la vibración como parámetro a medir, sin embargo, existen diferentes variables en la vibración de una estructura que sirven para comparar una estructura dañada con el diseño original.

1.2.1. Método de variación de frecuencia de vibración

Existe una gran cantidad de literatura con relación a la detección de variaciones en las frecuencias de vibración. Los cambios en propiedades estructurales de materiales provocan cambios en la frecuencia de vibración. Esta es la razón de la utilización de este parámetro para comparar el daño en una estructura (Doebbling *et al.*, 1996).

El problema de los métodos de cambios de frecuencia de vibración es la baja sensibilidad de cambios de frecuencia a los daños en la estructura. Para que se tengan valores exactos se necesitan mediciones muy precisas. Con baja sensibilidad son únicamente útiles para las estructuras con daños grandes, de acuerdo con (Doebbling *et al.*, 1996). La aplicación de métodos de cambios de frecuencias de vibración únicamente es recomendada para ambientes como control de manufactura.

1.2.1.1. Problema hacia adelante

El problema hacia adelante, o (*forward problem*) como lo describe Doebbling *et al.* (1996). es la manera más sencilla de resolver un problema de monitoreo. El proceso que se sigue en este método es modelar matemáticamente las

frecuencias modales de la estructura, para posteriormente compararlas a las medidas registradas por sensores.

1.2.1.2. Problema inverso

De acuerdo con Doebling *et al.* (1996) esta metodología consiste en el cálculo de parámetros como longitud de grietas y la ubicación, las cuales pueden provocar los cambios de frecuencias

1.2.2. Métodos de cambio de formas modales

Una forma de inferir daño en la estructura es comparando la forma de los modos de vibración de la estructura, todas las estructuras toman diferente forma en cada periodo de vibración, ya sea torsional u oscilatorio. (West, 1984) Presenta una forma de localizar daños en la estructura con el criterio de garantía modal (MAC por sus siglas en ingles), para determinar el nivel de correlación que existía entre modos de vibración de un transbordador espacial no dañado y otro modelo que fue expuesto a cargas acústicas. Los cambios en las formas de vibrar de la estructura medidas mediante el MAC permitieron localizar el daño en la estructura del transbordador.

Otra metodología para la comparación de los cambios de forma de vibrar de la estructura es presentada por (Yuen, 1985), en la cual compara los vectores propios calculados con los experimentales.

$$\{\phi^*\}_i = \frac{\{\phi^d\}_i}{\omega_i^d} - \frac{\{\phi^u\}_i}{\omega_i^u} \quad (1)$$

$$\{\phi^*\}_{i'} = \frac{\{\phi^d\}_i}{\omega_i^d} - \frac{\{\phi^u\}_{i'}}{\omega_i^u} \quad (2)$$

Los cambios en estos parámetros son simulados para la rigidez de cada elemento estructural, luego, los valores predichos son comparados con los valores medidos en cada ubicación.

1.2.3. Métodos de curvatura/deformación

Otra forma de calcular los cambios en la vibración es utilizando derivadas de las formas de los modos de vibración (Doebbling, Farrar, Prime, & Shevitz, 1996). Estos se basan en las ecuaciones generales de flexión y de esfuerzo deformación.

En la investigación de (Pandey, Biswas, & Samman, 1991) se demuestra que los cambios absolutos en la forma del modo de curvatura puede ser un buen indicador de daño en la viga ensayada. Los valores de curvatura fueron calculados a partir desplazamientos de los modos, usando una aproximación de diferenciación centrada.

1.2.4. Métodos basados en mediciones dinámicas de flexibilidad

Existe una manera de identificar daños mediante la medición dinámica de la matriz de flexibilidad y estimar los cambios en el comportamiento estático de la estructura. debido a que la matriz de flexibilidad se define como la inversa de la matriz de rigidez. De acuerdo con Doebbling (1996) el desplazamiento se define de la siguiente manera:

$$\{u\} = \{G\}\{F\} \quad (3)$$

Donde cada columna de la matriz de flexibilidad representa el patrón de desplazamiento asociado con una fuerza unitaria a cada grado de libertad.

1.2.5. Métodos no lineales

Los métodos no lineales consideran el comportamiento postelástico de los elementos estructurales. En muchos diseños actuales se contemplan comportamientos plásticos en las estructuras. En la investigación de Actis y Dimarogonas (1989), en la cual se desarrolló una viga mediante el método de elementos finitos, se determinó si la viga estaba abierta o cerrada mediante los signos del momento flexionante en la ubicación de la grieta, el estudio presenta resultados mostrando la compatibilidad generada entre la abertura y cierre de la grieta.

También existen métodos no lineales basados en el análisis modal como el presentado por Lin y Ewins (1990), en el cual se muestra un proceso sobre cómo localizar la no linealidad de un elemento utilizando análisis modal.

1.3. Aplicación de SHM en vigas y marcos

En la investigación de (Gudmonson, 1983) se presenta el caso de estudio del comportamiento dinámico de una viga con fisuras. La presencia de dichas fisuras se puede representar como una matriz de flexibilidad equivalente. La matriz de flexibilidad es determinada por medio de una ecuación de balance de energía, donde la energía elástica de la matriz de flexibilidad se balancea con la

energía de deformación de la grieta. Este método es utilizado para calcular los cambios en la frecuencia para una viga en voladizo con fisuras en las aristas.

1.3.1. Vigas de concreto

Se han realizado distintos ensayos y pruebas a vigas de concreto con agrietamiento y sin agrietamiento. La investigación de Chowdhury y Ramírez, (1992) presenta el estudio de una viga de concreto reforzado y una no reforzada comparando la respuesta modal de las que estaban en buen estado y otras que presentaban fisuras.

Algunas de las vigas estudiadas tenían defectos que simulaban grietas. La investigación examina los cambios de frecuencias y espectros de potencia causado por los cambios en resistencia y cambios de carga.

1.3.2. Marcos de acero

En la investigación de Friswell *et al.* (1994), se simulan daños a la estructura de marcos de acero. Se aplicó un algoritmo para encontrar daños en función de los cambios de vibración; los resultados fueron razonablemente exitosos.

1.4. Sensores utilizados en SHM

Existe variedad de sensores para el monitoreo de salud estructural. Todos tienen como objetivo llegar a una forma cuantificable de parámetros de vibración. Sin embargo, la variable a medir y la forma de transmisión varían en función de los sensores. A continuación, se presentan los más utilizados en investigaciones recientes.

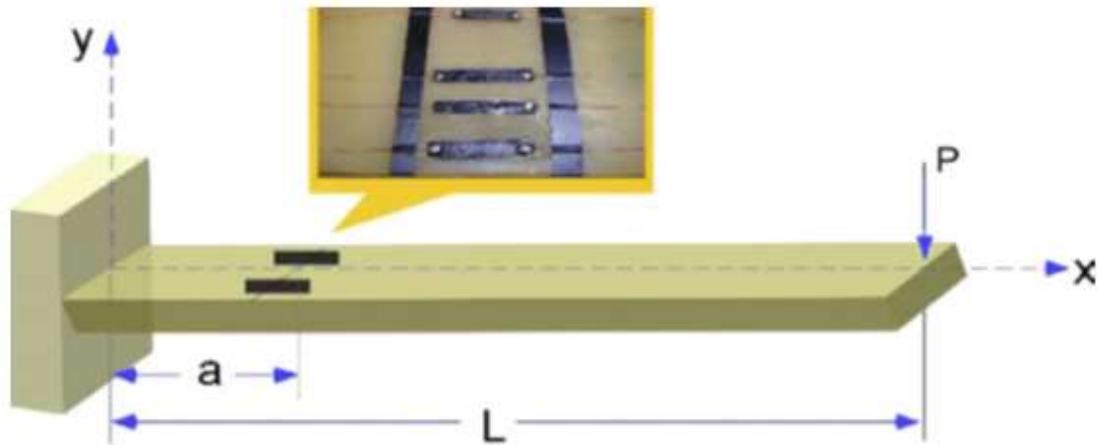
1.4.1. Sensores de deformación por nanotubos de carbono

Kang *et al.* (2006) muestran la utilización de un sensor fabricado a base de polímero de nanotubos de carbono que forman un sensor de deformación piezorresistivo para aplicaciones de monitoreo de salud estructural. El polímero mejora la unión interfacial entre los nanotubos. Los sensores de nanotubos de carbono de pared única utilizados en otras investigaciones producían mediciones de tensión distorsionadas, porque la fuerza de atracción de Van der Waals permitía el deslizamiento axial de los planos exteriores lisos de los nanotubos.

Esta investigación derivó un modelo eléctrico del sensor de deformación de nanotubos basado en espectroscopía de impedancia electroquímica y pruebas de deformación. El modelo es útil para diseñar sistemas de sensores de nanotubos. Se desarrolló una neurona artificial biomimética al extender la longitud del sensor. La neurona es un sensor de tensión continua que tiene un bajo costo, es fácil de instalar y es liviano.

La neurona tiene un ancho de banda bajo y una sensibilidad a la tensión adecuada. El sensor de neuronas es particularmente útil para detectar grandes deformaciones y grietas, y puede reducir el número de canales de adquisición de datos necesarios para el control de la salud de grandes estructuras Kang *et al.* (2006).

Figura 1. **Set de sensores de deformación de nanotubos de carbono en una viga en voladizo**



Fuente: Kang *et al.* (2006). *A carbon nanotube strain sensor for structural health monitoring*.
Consultado el 12 de noviembre de 2021. Recuperado de <https://iopscience.iop.org/article/10.1088/0964-1726/15/3/009>.

1.4.2. Transductores piezocerámicos

Los transductores piezocerámicos han surgido como nuevas herramientas para el monitoreo de la salud de estructuras a gran escala debido a sus ventajas de detección activa, bajo costo, respuesta rápida, disponibilidad en diferentes formas y simplicidad de implementación como presenta Song *et al* (2007). En dicha investigación se utilizan transductores piezocerámicos para la detección de daños en un puente de hormigón armado de 6.1 m de longitud. Los mismos están incrustados en la estructura de hormigón en ubicaciones espaciales predeterminadas antes de la fundición.

En el monitoreo realizado por Song *et al.* (2007) se involucra diez parches piezocerámicos incrustados en ubicaciones espaciales en cuatro secciones transversales diferentes.

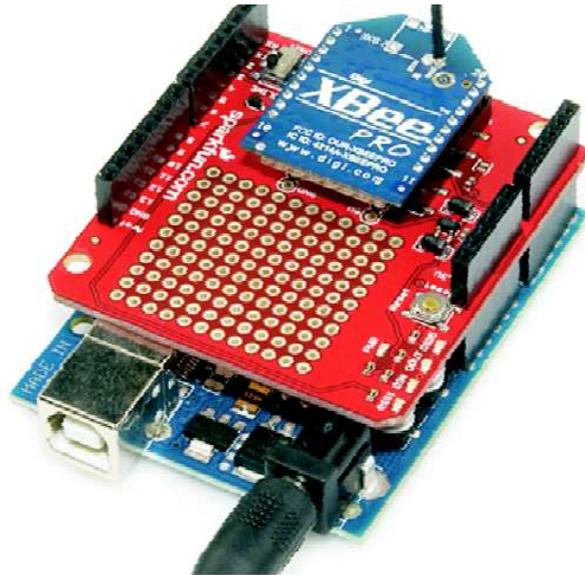
Para inducir grietas en la tapa doblada, la estructura se somete a cargas de cuatro actuadores hidráulicos con capacidades de 80 y 100 toneladas. Además de los sensores piezocerámicos, en el experimento se utilizan galgas extensométricas y transductores de desplazamiento lineal variable, los cuales convierten vibraciones en movimiento mecánico lineal; se usan también microscopios para proporcionar datos de referencia.

1.5. Microcontroladores

Existen diferentes microcontroladores de diferentes marcas, los cuales pueden trabajar con los sensores anteriormente mencionados. Sin embargo, uno de los más comunes por su accesibilidad y por utilizar software libre son los microcontroladores Arduino.

El microcontrolador Arduino Uno consiste en una placa de microcontrolador basada en ATmega328P, tiene 14 pines de entrada y salida digital (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz (CSTCE16M0V53-R0), una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio (Arduino, 2020).

Figura 2. **Arduino montado con XBee RF y XBee Shield**



Fuente: Chougle *et al.* (2010). *Low Cost Wireless Sensor Network for Structural Health Monitoring*. Consultado el 12 de diciembre de 2021. Recuperado de https://vbn.aau.dk/ws/files/45712057/Low_Cost_Wireless_Sensor_Network_for_Structural_Health_Monitoring.pdf.

1.6. **Conectividad de sensores con el microcontrolador**

Uno de los sensores basados en MEMS utilizados es el MPU6050 producido por Invensense, consiste en un chip que contiene un acelerómetro y un sensor giroscópico incorporado. El sensor del acelerómetro posee un rango de medición configurable entre ± 2 , ± 4 , ± 8 y ± 16 g con una resolución de 16 bits, mientras que el sensor giroscópico tiene un rango entre ± 250 , ± 500 , ± 1000 , ± 2000 grados/s, también configurable y con una resolución de 16 bits. El enlace del chip MPU-6050 con el Arduino se puede realizar mediante el pin serial a través del protocolo I2C (Varanis *et al.*, 2017).

El segundo sensor basado en MEMS, es el modelo ADXL-335, producido por Analog Devices; este consiste en un acelerómetro que tiene un rango fijo de medición de ± 3 g. La señal de salida del acelerómetro es analógica en forma de voltaje proporcional a la aceleración a la que fue sometido el sensor. En este caso, como la señal es digitalizada por Arduino, su resolución es de 10 bits.

2. PROGRAMACIÓN DE SENSOR Y MICROCONTROLADOR

Para el desarrollo del sensor se realizó una comparación de sensores de aceleración en el mercado y posteriormente se seleccionó el más conveniente en cuanto a sus características; posterior a eso se desarrolló el código de programación para su funcionamiento correcto.

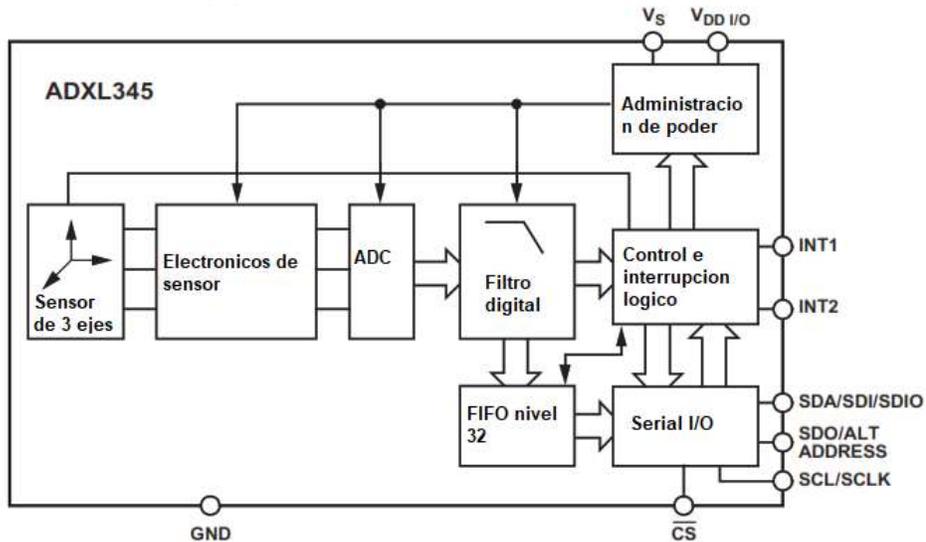
2.1. Comparación de sensores de aceleración

Los sensores de aceleración para dispositivos electrónicos funcionan mediante un mecanismo de generación de resistencia eléctrica, el cual puede variar dependiendo de las características del sensor, su sensibilidad y la filtración de ruido que ofrece. Se evaluarán dos sensores de aceleración disponibles en el mercado guatemalteco, para determinar qué alternativa funciona mejor para el monitoreo de salud estructural.

2.1.1. Sensor ADXL345

El ADXL345 es un acelerómetro digital de bajo consumo de energía, tiene una resolución de 13 bits y un rango de medición de aceleración de 16 g en ambas direcciones. El acelerómetro es capaz de medir aceleración estática debido a la gravedad y aceleración dinámica, debido a movimientos con una inclinación de los ejes de menos de un grado (Analog Devices, 2010).

Figura 3. Diagrama de bloque de sensor ADXL345



Fuente: Analog Devices (2010). *Sensores ADXL.345*. Consultado el 10 de diciembre de 2021.
Recuperado de <https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>.

2.1.2. Características de medición

Se comparó la capacidad de medición de los sensores de aceleración para determinar el más conveniente para el monitoreo de salud estructural.

2.1.2.1. Parámetros de entrada

El sensor cuenta con 3 ejes para la medición de aceleración; entre ellos el fabricante realiza pruebas en cada uno; se presenta un valor típico y en ocasiones valores mínimos y máximos.

Tabla I. **Especificaciones de parámetros de entrada sensor ADXL345**

Parámetro (entrada del sensor)	Condiciones de prueba	Min.	Típico	Máx.	Unidades
Rango de medición	Cada eje seleccionable por el usuario Porcentaje de escala completa		±2, ±4, ±8, ±16		g
No linealidad			±0.5		%
Error intereje de alineación			±0.1		Grados
Sensibilidad de eje transversal			±1		%

Fuente: elaboración propia.

Uno de los parámetros más importantes en la medición de aceleraciones con cambios abruptos en un intervalo de tiempo corto es el ruido, ya que este puede modificar los resultados de aceleraciones, el sensor ADXL345 tiene valores típicos para ruido de 0.75 LSB rms en los ejes X y Y, y valores de 1 LSB rms en el eje Z; estos valores se refieren al porcentaje del valor total del mínimo cambio de voltaje detectable.

Tabla II. **Especificaciones de parámetros de ruido sensor ADXL345**

Parámetro	Condiciones de prueba	Min	Típico	Máx.	Unidades
Ejes X-, Y-	ODR = 100 Hz para ± 2 g, resolución de 10 bits o todos los rangos g, resolución completa		0.75		LSB rms
Eje Z			1.1		LSB rms

Fuente: elaboración propia.

2.1.3. Sensibilidad

La sensibilidad de un sensor se mide en el rango de las aceleraciones: esto se puede determinar como el valor de LSB/g en cada una de las configuraciones de resolución. También existe un valor de cambio de sensibilidad de debido a la temperatura. Sin embargo, este último no afectará en gran medida las mediciones realizadas, ya que presentan un valor de 0.01 % por cada grado centígrado de cambio.

Tabla III. **Especificaciones de parámetros de sensibilidad sensor ADXL345**

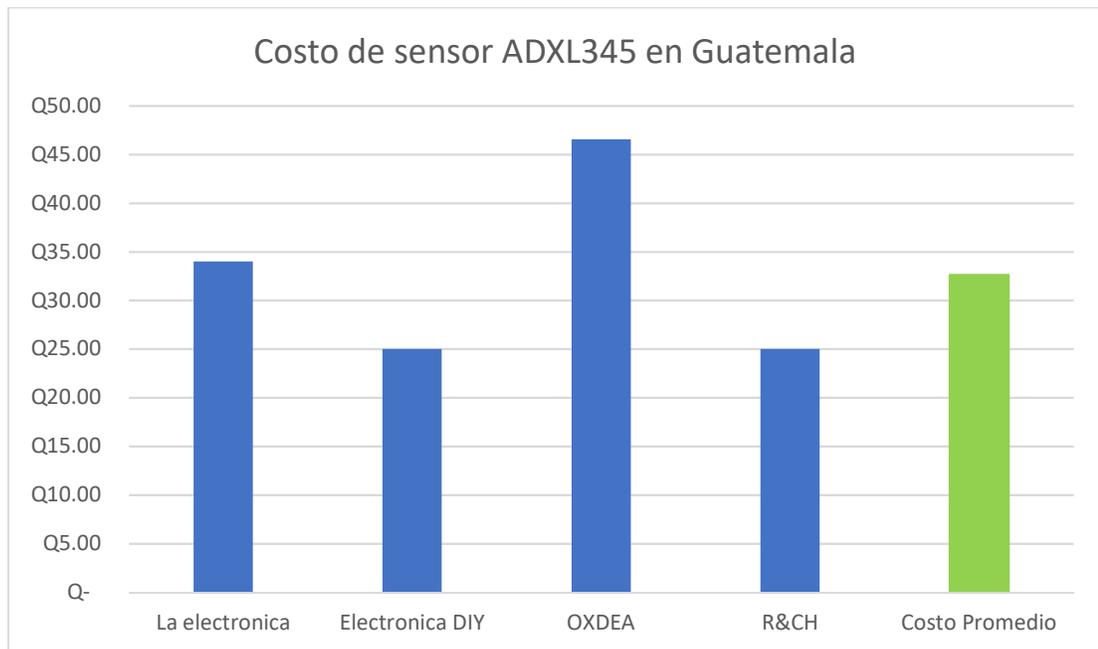
Parámetro	Condiciones de prueba	Min.	Típico	Máx.	Unidades
Sensibilidad en Xout, Yout, Zout	Todos los rangos de g, resolución full	230	256	282	LSB/g
	±2 g, resolución de 10-bit	230	256	282	LSB/g
	±4 g, resolución de 10-bit	115	128	141	LSB/g
	±8 g, resolución de 10-bit	57	64	71	LSB/g
	±16 g, resolución de 10-bit	29	32	35	LSB/g
Desviación de la sensibilidad ideal	Todos los rangos de g, resolución full	3.5	±1.0	4.3	%
	±2 g, resolución de 10-bit	3.5	3.9	4.3	mg/LSB
	±4 g, resolución de 10-bit	7.1	3.9	8.7	mg/LSB
	±8 g, resolución de 10-bit	14.1	7.8	17.5	mg/LSB
	±16 g, resolución de 10-bit	28.6	15.6	34.5	mg/LSB
Cambio de sensibilidad debido a temperatura			±0.01		%/°C

Fuente: elaboración propia.

2.1.4. Costo

Los sensores para medición de aceleraciones y giroscopios tienen un costo reducido en comparación con sensores más complejos; se realizó una investigación en distintas tiendas de componentes electrónicos para obtener el costo promedio del sensor en Guatemala, obteniéndose los siguientes resultados:

Figura 4. Costo de sensor ADXL345 en Guatemala



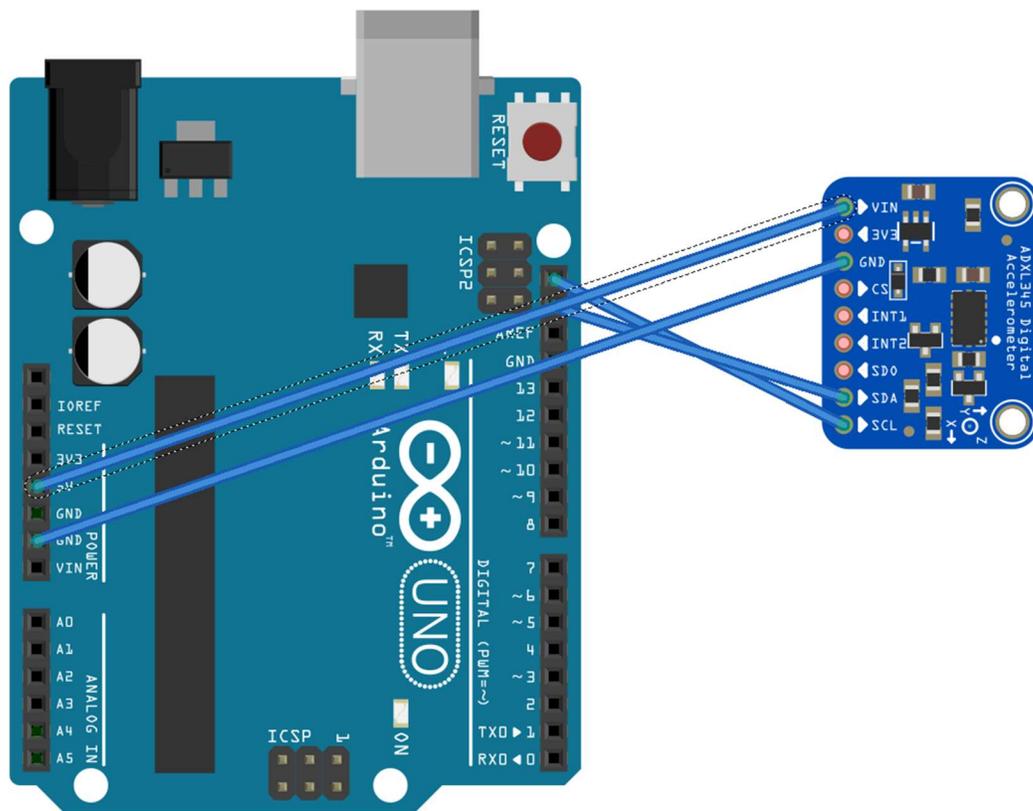
Fuente: elaboración propia.

2.1.5. Compatibilidad

Para comunicarse con algún controlador, el sensor ADXL345 tiene interfaces I2C y SPI. El SPI puede ser de tres o cuatro cables. Asimismo, los

pinos de interrupción del sensor pueden interactuar con los canales I/O del controlador. Los microcontroladores Arduino pueden ser compatibles con I2C y SPI; para las conexiones, el sensor se alimenta de una salida de 5V del Arduino (Agnihotri, 2020).

Figura 5. **Diagrama de conexión de sensor ADXL345 con microcontrolador Arduino UNO**



Fuente: Agnihotri(2020). *Arduino-mpu6050*. Consultado el 5 de diciembre de 2021. Recuperado de <https://www.engineersgarage.com/tag/arduino-mpu6050-motion-tracking-sensor/>

2.1.6. Sensor MPU4050

El sensor MPU-6050 es un dispositivo con giroscopio de 6 ejes y un acelerómetro de 3 ejes; tiene una conexión bus I2C dedicado, que acepta directamente entradas de una brújula externa de 3 ejes.

El MPU-6050 cuenta con tres convertidores de comunicación analógica a digital (ADC) de 16 *bytes* para digitalizar las salidas del giroscopio y tres ADC de 16 *bytes* para digitalizar las salidas del acelerómetro. Cuentan con un búfer de 1024 *bytes* en el *chip*, el cual ayuda a reducir el consumo de energía del sistema. Permite que el procesador del sistema para leer los datos del sensor en grupos ingrese a un modo de bajo consumo, a medida que el sensor MPU recopila más datos.

2.1.7. Características de medición

Las características de entrada del sensor son las siguientes:

2.1.7.1. Parámetros de entrada

El acelerómetro de 3 ejes del MPU-6050 utiliza masas de prueba separadas para cada eje. La aceleración a lo largo de un eje particular induce el desplazamiento en la masa de prueba correspondiente, y los sensores capacitivos detectan el desplazamiento diferencial. La arquitectura del MPU-6050 reduce la susceptibilidad de los acelerómetros a variaciones de fabricación, así como a la deriva térmica. Cuando el dispositivo se coloca sobre una superficie plana, medirá 0 g en los ejes X e Y + 1g en el eje Z. El factor de escala de los acelerómetros se calibra en fábrica.

Es nominalmente independiente de la tensión de alimentación. Cada sensor tiene un ADC sigma-delta dedicado para proporcionar salidas digitales. El rango de escala completa de la salida digital se puede ajustar a $\pm 2g$, $\pm 4g$, $\pm 8g$ o $\pm 16g$. (InverSense, 2013)

Tabla IV. **Parámetros de entrada sensor MPU 6050**

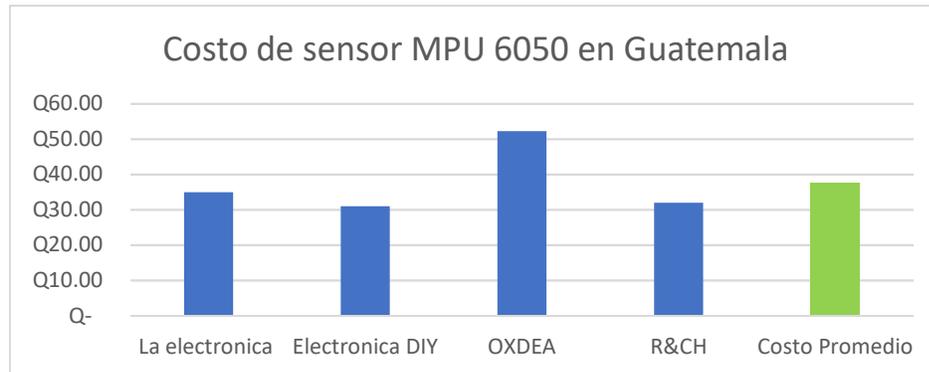
Parámetro (entrada del sensor)	Min	Típico	Max	Unidades
Rango de medición		$\pm 2, \pm 4, \pm 8, \pm 16$		g
Calibración Inicial		± 0.5		%
Sensibilidad de eje transversal		± 2		%
Desempeño de ruido		400		$\mu g/Hz$

Fuente: elaboración propia.

2.1.8. Costo

El sensor MPU 6050 en Guatemala se encuentra dentro del mismo rango de precio del sensor ADXL 345, ambos costos se encuentran alrededor de los Q35 y presentan características similares de medición. La diferencia más notable es la adición de un giroscopio en el sensor MPU 6050; este permite ser utilizado para aplicaciones en control y navegación.

Figura 6. **Costo de sensor MPU 6050 en Guatemala**



Fuente: elaboración propia, empleando Excel.

2.1.9. Compatibilidad

El MPU-6050 se comunica con un procesador del sistema mediante un SPI o una interfaz serial I2C. El MPU-6050 siempre actúa como esclavo cuando se comunica con el procesador del sistema. El LSB se establece mediante el pin 9 (AD0).

El nivel lógico para las comunicaciones con el maestro es establecido por el voltaje en VLOGIC (InverSense, 2013).

2.2. Comparación de microcontroladores

El uso de sensores de vibración no solo es utilizado para aplicaciones estructurales, sino que también es bastante común su utilización para el monitoreo de fugas en tuberías. Existe una investigación en el uso del sensor ADXL335 y el sensor MPU6050 para la fuga en tuberías por (Dziyaudidin & Ahmad, 2015), cuyo objetivo fue analizar estos sensores para detectar las

condiciones de una tubería a través de una red ZigBee, los sensores fueron examinados en 3 condiciones de prueba diferente, i) sin fugas, ii) fuga de 1 mm y iii) fuga de 3 mm. los datos se analizaron basados en dominio de frecuencia utilizando la transformada rápida de Fourier (FFT).

Tabla V. **Comparación de desempeño de acelerómetros ADXL335 y MPU6050**

Aspecto	MPU6050	ADXL335
Datos de salida ejes X, Y Z	Digital (1024 bytes FIFO buffer)	Análogo
Rango de medición	Hasta $\pm 16g$	$\pm 3g$
Convertidos de señal análoga a digital (ADC)	16-bytes (convertidor interno de sensor)	8 a 10-bytes (convertidor en microcontrolador)
Sensitividad	Hasta 16, 384 LSB/g	300 mV/g
Salida cero g X0, Y0 y Z0	$\pm 50mg$ $\pm 80mg$	1.35 – 1.65 V 1.2 + 1.8 V

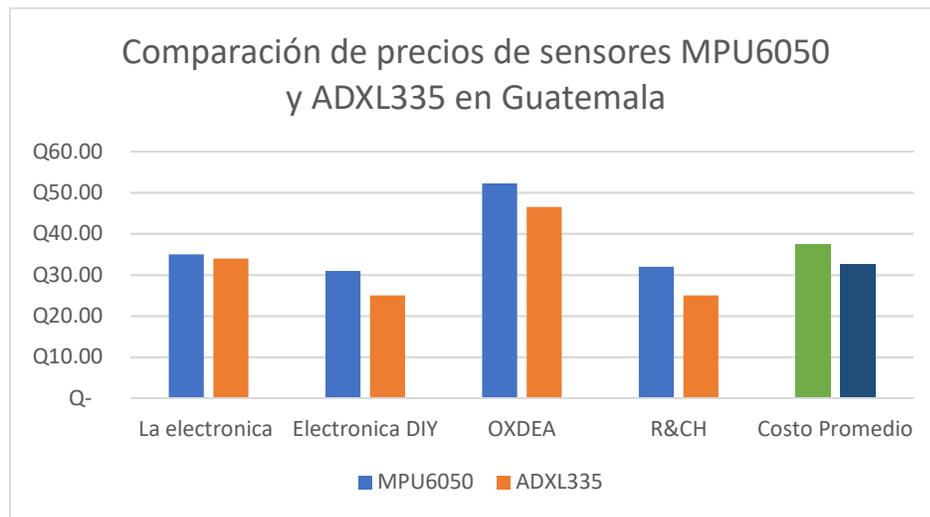
Fuente: elaboración propia.

En este estudio se concluye que ambos sensores son capaces de detectar fugas en tuberías, sin embargo, se observó que el acelerómetro MPU6050 puede detectar la vibración a una distancia mayor que el acelerómetro ADXL335. Asimismo, el rango de medición del MPU6050 es mayor que el de un ADXL335.

2.2.1. Costo

En Guatemala, el costo del sensor MPU6050 es un poco más elevado. Sin embargo, de acuerdo con las comparaciones y considerando sus características y los resultados obtenidos en otras investigaciones, resulta aceptable el costo mayor.

Figura 7. Costo de sensores MPU6050 y ADXL335 en Guatemala

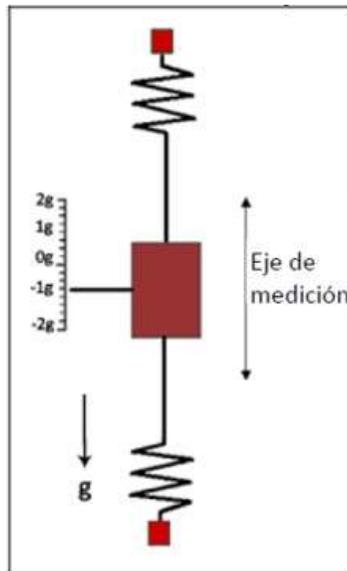


Fuente: elaboración propia.

2.3. Desarrollo de código de programación de sensor

Debido a las ventajas de medición, sensibilidad, características y costo, se utilizó un sensor MPU6050 para el desarrollo del sistema de monitoreo de salud estructural; este funciona de manera inercial utilizando un sistema de masa resorte llamados MEMS (*micro electro mechanical systems*) por sus siglas en inglés. Permite medir la aceleración mediante la Segunda Ley de Newton, por lo cual basta con dividir la fuerza dentro de la masa para encontrar la aceleración.

Figura 8. Diagrama de sistema MEMS para medición de aceleración



Fuente: Naylan Mechatronics (2016). *Acelerómetro y giroscopio*. Consultado el 10 de enero de 2022. Recuperado de https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html.

2.4. Calibración de sensor de aceleración

Se realizó una calibración preliminar de la dirección interna del sensor, debido a que es posible que este no se encuentre totalmente horizontal al ser soldado, lo que hace estas irregularidades en la orientación. Se utilizará el código de Naylam Mechatronics, (2016) para calibrarlo, el cual se explica a continuación:

Se utilizan librerías I2C para controlar el sensor. El sensor necesita las librerías I2Cdev.h y a la vez, esta necesita la librería Wire.h:

```
#include "I2Cdev.h"  
#include "MPU6050.h"
```

```
#include "Wire.h"
```

Se declaran las variables sin ningún procesamiento del acelerómetro y el giroscopio en los 3 ejes del sensor (x, y, z):

```
int ax, ay, az;  
int gx, gy, gz;
```

Se declaran las variables usadas para asignar los valores filtrados y un contador:

```
long f_ax, f_ay, f_az;  
int p_ax, p_ay, p_az;  
long f_gx, f_gy, f_gz;  
int p_gx, p_gy, p_gz;  
int counter=0;
```

Se declaran las variables para los offset:

```
//Valor de los offset  
int ax_o, ay_o, az_o;  
int gx_o, gy_o, gz_o;
```

Se inicia la sección de configuración del código y la comunicación serial para monitorear el dispositivo cuando se encuentre conectado a una computadora:

```
void setup() {  
  Serial.begin(57600);  
  Wire.begin();  
}
```

```
sensor.initialize();
```

Se establece una condición para que se pueda verificar en la pantalla si el sensor está conectado correctamente:

```
if (sensor.testConnection()) Serial.println("Sensor iniciado  
correctamente");
```

Se leen los valores de los offset declarados anteriormente:

```
ax_o=sensor.getXAccelOffset();  
ay_o=sensor.getYAccelOffset();  
az_o=sensor.getZAccelOffset();  
gx_o=sensor.getXGyroOffset();  
gy_o=sensor.getYGyroOffset();  
gz_o=sensor.getZGyroOffset();
```

Se muestran en el monitor serial los *offset* establecidos:

```
Serial.println("Offsets:");  
Serial.print(ax_o); Serial.print("\t");  
Serial.print(ay_o); Serial.print("\t");  
Serial.print(az_o); Serial.print("\t");  
Serial.print(gx_o); Serial.print("\t");  
Serial.print(gy_o); Serial.print("\t");  
Serial.print(gz_o); Serial.print("\t");  
Serial.println("\n\nEnvie cualquier caracter para empezar la  
calibracion\n\n");
```

Se debe ingresar un carácter para poder calibrarlo y se despliega en el monitor serial el mensaje de que se está calibrando:

```
while (true){if (Serial.available()) break;}
Serial.println("Calibrando, no mover IMU");
}
```

En el ciclo principal se leen las aceleraciones y velocidades angulares del sensor:

```
void loop() {

    sensor.getAcceleration(&ax, &ay, &az);
    sensor.getRotation(&gx, &gy, &gz);

    // Filtrar las lecturas
    f_ax = f_ax-(f_ax>>5)+ax;
    p_ax = f_ax>>5;

    f_ay = f_ay-(f_ay>>5)+ay;
    p_ay = f_ay>>5;

    f_az = f_az-(f_az>>5)+az;
    p_az = f_az>>5;

    f_gx = f_gx-(f_gx>>3)+gx;
    p_gx = f_gx>>3;

    f_gy = f_gy-(f_gy>>3)+gy;
```

```
p_gy = f_gy>>3;
f_gz = f_gz-(f_gz>>3)+gz;
p_gz = f_gz>>3;
```

Luego de realizar 100 lecturas se corrige el offset del sensor:

```
if (counter==100){
    //Mostrar las lecturas separadas por un [tab]
    Serial.print("promedio:"); Serial.print("\t");
    Serial.print(p_ax); Serial.print("\t");
    Serial.print(p_ay); Serial.print("\t");
    Serial.print(p_az); Serial.print("\t");
    Serial.print(p_gx); Serial.print("\t");
    Serial.print(p_gy); Serial.print("\t");
    Serial.println(p_gz);
}
```

Se ajusta el acelerómetro para poder asignar un valor exacto de 1g en el eje Z:

```
if (p_ax>0) ax_o--;
else {ax_o++;}
if (p_ay>0) ay_o--;
else {ay_o++;}
if (p_az-16384>0) az_o--;
else {az_o++;}

sensor.setXAccelOffset(ax_o);
sensor.setYAccelOffset(ay_o);
sensor.setZAccelOffset(az_o);
```

Se asigna un valor de cero grados por segundo al giroscopio en todos los ejes:

```
    if (p_gx>0) gx_o--;  
    else {gx_o++;}  
    if (p_gy>0) gy_o--;  
    else {gy_o++;}  
    if (p_gz>0) gz_o--;  
    else {gz_o++;}  
  
    sensor.setXGyroOffset(gx_o);  
    sensor.setYGyroOffset(gy_o);  
    sensor.setZGyroOffset(gz_o);  
  
    counter=0;  
  }  
  counter++;  
}
```

2.4.1. Desarrollo de código para medición de aceleración

Para el desarrollo de la codificación de los sensores se utilizó como base el código de Naylam Mechatronics (2016), modificándolo y agregándole nuevas librerías y líneas de código que permitan implementar el almacenamiento de datos utilizando un módulo microSD conectado al Arduino; dicho código se muestra a continuación:

Para utilizar el sensor MPU6050 se aplican las librerías I2Cdev.h, MPU6050.h y Wire.h. Adicional a esto se necesita la librería SD.h para controlar el módulo micro SD.

```
#include "I2Cdev.h"  
#include "MPU6050.h"  
#include "Wire.h"  
#include <SD.h>
```

Se crea la variable MyFile para poder almacenar datos, y se inicia el sensor MPU6050:

```
File myFile;  
MPU6050 sensor;
```

Se crean variables enteras de los datos del acelerómetro y giroscopio de los 3 ejes x, y, z:

```
int ax, ay, az;  
int gx, gy, gz;
```

Se crean variables tipo "Float" para almacenar intervalos de tiempo y tener aceleraciones y ángulos previos:

```
long tiempo_prev;  
float dt;  
float ang_x, ang_y;  
float ang_x_prev, ang_y_prev;
```

Se inicia la configuración y el puerto serial, la conexión I2C y el sensor. Si el dispositivo se encuentra conectado a un monitor serial se muestra un mensaje confirmando si se logró iniciar la conexión:

```
void setup() {  
  Serial.begin(57600);  
  Wire.begin();  
  sensor.initialize();  
  
  if(sensor.testConnection())Serial.println("Sensoriniciado  
correctamente");  
  else Serial.println("Error al iniciar el sensor");  
}
```

Se inicia el módulo micro SD y se muestra en el monitor serial un mensaje si se pudo iniciar o se tuvo un error:

```
Serial.print("Iniciando SD ...");  
if (!SD.begin(4)) {  
  Serial.println("No se pudo inicializar");  
  return;  
}  
Serial.println("inicializacion exitosa");  
}
```

Se inicia el ciclo principal, se abre el archivo en la memoria micro SD en formato .CSV; para poder escribir en ese archivo:

```
void loop() {  
  
  myFile = SD.open("datalog.csv", FILE_WRITE);//se abre el archivo
```

Se obtienen los valores de aceleración y rotación del sensor en los tres ejes:

```
sensor.getAcceleration(&ax, &ay, &az);  
sensor.getRotation(&gx, &gy, &gz);
```

Se realiza la conversión de valores de g a m/s^2 y en ángulos de rotación en grados

```
float ax_m_s2 = ax * (9.81/16384.0);  
float ay_m_s2 = ay * (9.81/16384.0);  
float az_m_s2 = az * (9.81/16384.0);  
float gx_deg_s = gx * (250.0/32768.0);  
float gy_deg_s = gy * (250.0/32768.0);  
float gz_deg_s = gz * (250.0/32768.0);
```

Se escriben los valores de aceleración y se guardan en el archivo de la memoria separados por comas:

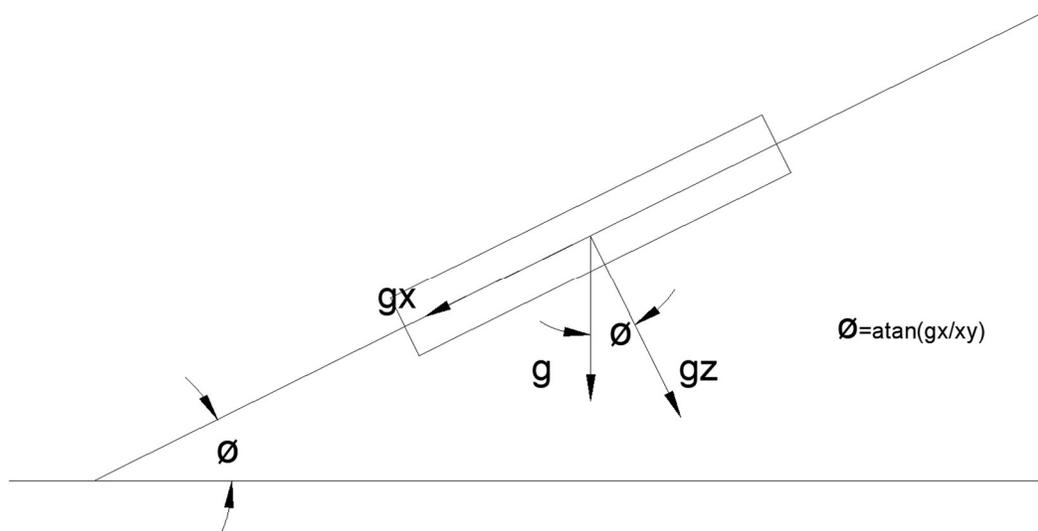
```
myFile.print(millis());  
myFile.print(",");  
Serial.print(ax_m_s2); Serial.print("\t");  
myFile.print(",");  
Serial.print(ay_m_s2); Serial.print("\t");  
myFile.print(",");  
Serial.print(az_m_s2); Serial.print("\t");  
myFile.print(",");
```

Se calcula el intervalo de tiempo entre lecturas para realizar correcciones posteriores:

```
dt = (millis()-tiempo_prev)/1000.0;  
tiempo_prev=millis();
```

Para obtener lecturas más exactas de ángulos se calculan utilizando los valores del acelerómetro y los del giroscopio. Para determinar el giro se descomponen las aceleraciones vectoriales y se encuentra el ángulo respecto de la aceleración de 1g:

Figura 9. **Diagrama de aceleraciones en un cuerpo inclinado**



Fuente: Naylan Mechatronics (2016). *Acelerómetro y giroscopio*. Consultado el 10 de enero de 2022. Recuperado de https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html.

Esto resulta en las siguientes ecuaciones:

$$\theta_x = \tan^{-1} \left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}} \right) \quad (4)$$

$$\theta_y = \tan^{-1} \left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}} \right) \quad (5)$$

$$\theta_z = \tan^{-1} \left(\frac{a_z}{\sqrt{a_y^2 + a_x^2}} \right) \quad (6)$$

Se ingresan estas ecuaciones al código:

```
float accel_ang_x=atan(ay/sqrt(pow(ax,2) + pow(az,2)))*(180.0/3.14);  
float accel_ang_y=atan(-ax/sqrt(pow(ay,2) + pow(az,2)))*(180.0/3.14);
```

Para combinar la medición del acelerómetro y el giroscopio para obtener mejores resultados, se utiliza la siguiente ecuación: el giroscopio presenta datos de velocidad angular, por lo cual es necesario multiplicarlo por el diferencial de tiempo para obtener ángulos de giro. A estos ángulos se les da una ponderación de 0.98 y se suma un 0.02 del ángulo calculado para acumular menos errores:

$$angulo = 0.98(angulo_0 + \omega_{giroscopio} * dt) + 0.02(angulo_{acelerometro}) \quad (7)$$

Se ingresan la ecuación al código para calcular los ángulos de giro:

```
ang_x = 0.98*(ang_x_prev+(gx/131)*dt) + 0.02*accel_ang_x;  
ang_y = 0.98*(ang_y_prev+(gy/131)*dt) + 0.02*accel_ang_y;  
ang_x_prev=ang_x;
```

```
ang_y_prev=ang_y;
```

Se guardan los datos de ángulos de giro en la memoria microSD separados por comas y se cierra el archivo:

```
//Mostrar los ángulos separados por un [tab]
Serial.print(ang_x); Serial.print("\t");
myFile.print(",");
Serial.println(ang_y);
myFile.print(",");
myFile.print(ang_x); myFile.print("\t");
myFile.println(ang_y);

myFile.close(); //se cierra el archivo

}
```

2.4.2. Cálculo de parámetros de ajuste de sensor

Los sensores luego de ser calibrados según los *offset* especificados, presentan variaciones de aceleración dependiendo del voltaje suministrado y por las condiciones específicas del comportamiento interno del sensor. A pesar de aplicar *offset* iniciales y realizar iteraciones hasta llevar a los valores deseados en reposo, es necesario normalizar los valores obtenidos del sensor. Esto se realiza obteniendo el promedio de valores del sensor en reposo y dividiendo los valores posteriores dentro de esta cantidad.

Tabla VI. **Datos sin procesar obtenidos del sensor en reposo**

Milisegundo	Aceleración x [m/s²]	Aceleración y [m/s²]	Aceleración z [m/s²]
420	-1.250	-0.080	10.970
620	-1.240	0.000	10.950
656	-1.190	-0.060	10.980
693	-1.320	-0.10	10.990
730	-1.280	-0.040	10.970
765	-1.210	-0.110	10.920
802	-1.180	-0.050	10.840
838	-1.230	0.000	10.870
874	-1.280	-0.040	10.990
911	-1.230	-0.040	10.970

Fuente: elaboración propia.

En la tabla número VI se presentan datos obtenidos del sensor de aceleración en reposo. Se puede observar que en el eje X se presenta un corrimiento de aproximadamente 1.29 m/s²; para calcular los valores normalizados se encuentra el promedio de 20 datos en reposo y se toma como factor en el resto de resultados obtenidos. Esto se muestra en la siguiente ecuación:

$$a_x = \frac{a_{xi}}{\sum_{i=0}^{20} a_i} \quad (8)$$

Al aplicar los datos al filtro se muestran los de aceleración en reposo de la siguiente manera:

Tabla VII. **Datos de aceleración normalizados**

Hora	Aceleración x [m/s²]	Aceleración y [m/s²]	Aceleración z [m/s²]
22:45:00	0.010	0.190	9.810
22:45:01	0.000	-1.000	9.810
22:45:01	-0.040	-0.110	9.810
22:45:01	0.060	0.490	9.810
22:45:01	0.030	-0.410	9.810
22:45:01	-0.030	0.630	9.810
22:45:01	-0.050	-0.260	9.800
22:45:01	-0.010	-1.000	9.800
22:45:01	0.030	-0.410	9.810
22:45:01	-0.010	-0.410	9.810
22:45:01	0.020	0.780	9.810
22:45:01	0.020	0.930	9.810

Fuente: elaboración propia.

Luego de aplicar la normalización a los sensores se puede observar que los valores de reposo son los esperados; los valores en X y Y son cercanos a 0, y los de Z, muy cercanos al valor de la constante de aceleración de gravedad.

2.4.3. Cálculo de ruido y sensibilidad

Debido al rápido cambio en aceleraciones registradas durante un sismo es necesario aplicar un filtro para suavizar las curvas debido a errores de medición de parte del sensor, sin modificar los datos registrados en el mismo. Para esto se aprovecha la gran cantidad de datos registrados en el sensor (en promedio 24 datos por segundo) y se aplica un filtro de datos.

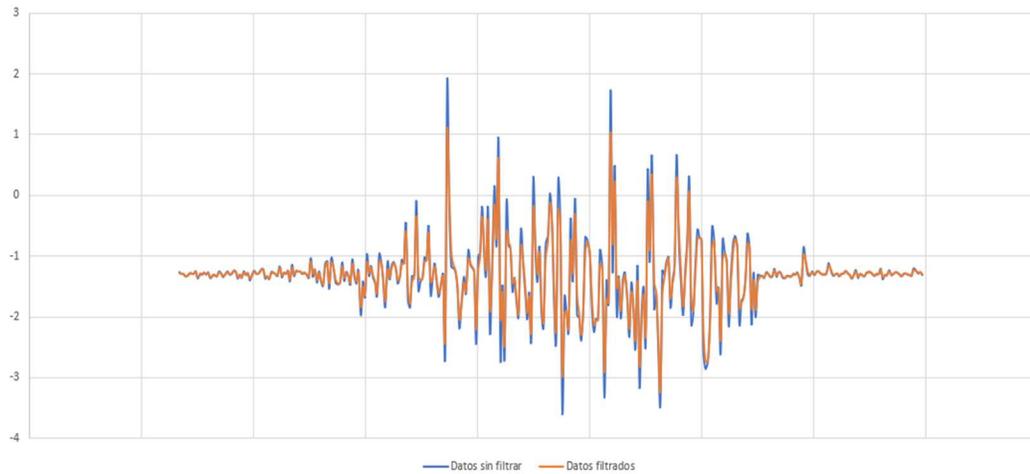
Para filtrar los datos registrados de estos sensores se utilizó un método de suavizado exponencial. Contrario a otros métodos de suavizado en los que se realizan promedios simples o ponderados, en el suavizado exponencial el peso de un valor va decreciendo mientras más alejado esté del dato que se va a suavizar.

Para realizar el promedio móvil exponencial para el sistema de monitoreo estructural se utilizó una fórmula de primer grado:

$$ax_{filtrada} = (1 - \alpha) * ax_{filtrada anterior} + \alpha * ax_{sin filtro} \quad (9)$$

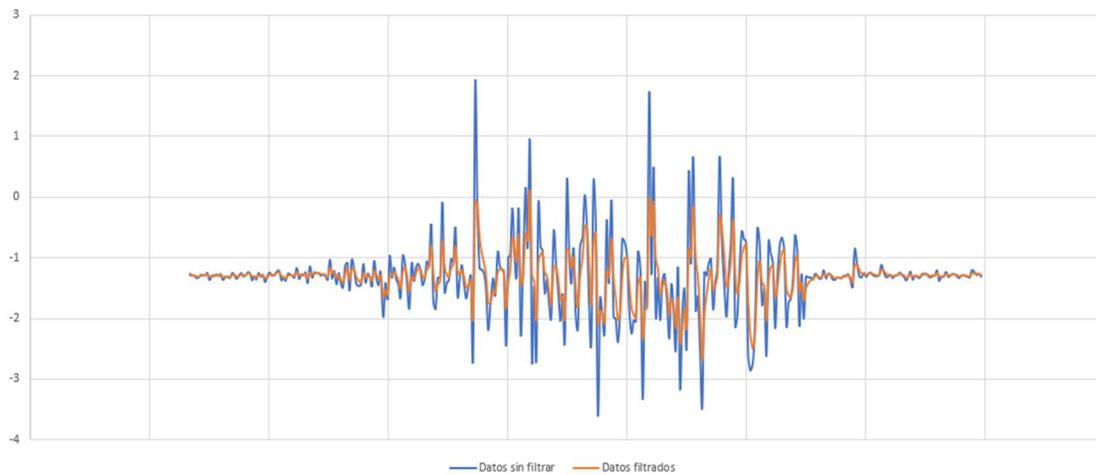
En donde α es una constante entre 0 y 1, en la que se puede configurar la cantidad de filtro aplicado a los datos. A continuación, se muestran los datos puros del sensor en color azul y los filtrados en color naranja, aplicando diferentes constantes de filtro.

Figura 10. **Datos filtrados con una constante de 0.80**



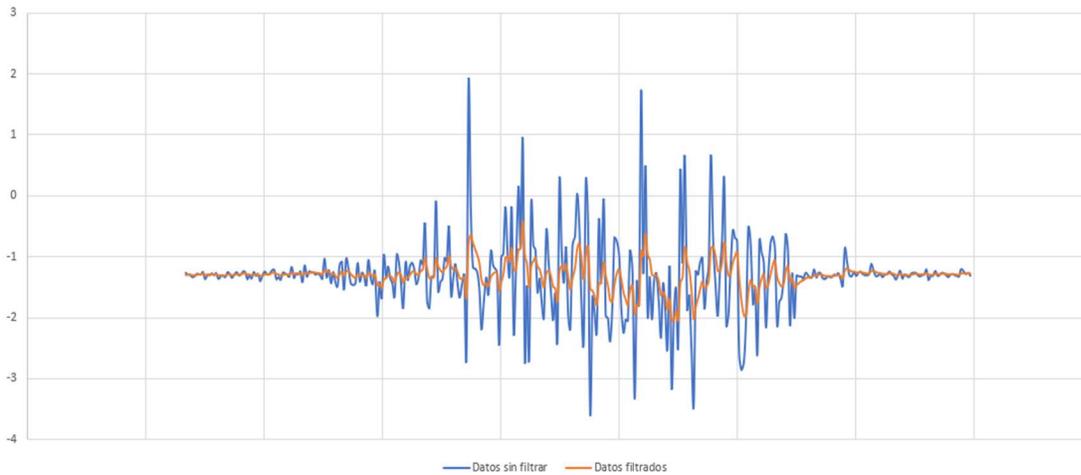
Fuente: elaboración propia, empleando Excel y un método de suavizado exponencial.

Figura 11. **Datos filtrados con una constante de 0.50**



Fuente: elaboración propia, empleando Excel y un método de suavizado exponencial.

Figura 12. **Datos filtrados con una constante de 0.25**



Fuente: elaboración propia, empleando Excel y un método de suavizado exponencial.

Debido a que el propósito de utilizar el filtro es únicamente eliminar errores y ruido en el sensor, no se debe de utilizar una constante de filtro demasiado grande, para no afectar los datos obtenidos del sensor.

3. EXPERIMENTACIÓN DE DISPOSITIVO DE MEDICIÓN

Se colocaron los sensores en un edificio de marcos de concreto reforzado y se analizaron los resultados con un modelo teórico para comparar e inferir daños estructurales.

3.1. Instalación de arreglo de sensores en edificio de marcos de concreto

Para realizar la prueba de los sensores en un edificio de concreto se solicitó permiso para instalar los mismos en un edificio ubicado en la zona 4 de la ciudad de Guatemala; el edificio está construido con un sistema de marcos de concreto reforzado, tiene un sótano y 5 niveles sobre el nivel de la calle y un sexto nivel construido posteriormente de mampostería liviana.

Se colocaron sensores de aceleración en cada nivel para obtener datos de comportamiento sísmico.

Figura 13. **Edificio monitoreado por arreglo de sensores**

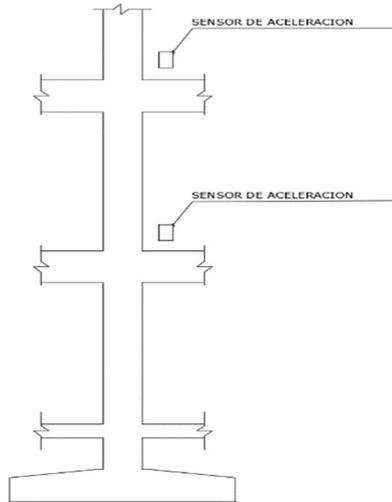


Fuente: Google Maps (2021). *Mapa ciudad de Guatemala*. Consultado el 6 de enero de 2022.

Recuperado de <https://www.google.com.gt/maps/@14.620752,-90.5118167442m/data=!3m1!1e3>.

Los sensores se colocaron lo más cercano a los diafragmas del edificio, ya que se asume en el modelo analítico que en ese punto se concentran las masas por nivel.

Figura 14. **Colocación de sensores en la estructura**



Fuente: elaboración propia.

Figura 15. **Instalación de sensores en estructura**



Fuente: [Fotografía de Juan Orozco]. (Ciudad de Guatemala. 2021). Colección particular. Guatemala.

Los sensores se colocaron el día 27 de agosto de 2021. Se debe llevar un registro de la hora de colocación para localizar los datos cuando se extraigan de las memorias. Se comenzó la colocación desde el nivel superior, tomando los siguientes registros.

Tabla VIII. **Hora de inicio de registros de sensores**

Hora	Hora de inicio de registros
Nivel 6	9:06:00
Nivel 5	9:15:00
Nivel 4	9:21:00
Nivel 3	9:28:00
Nivel 2	9:34:00
Nivel 1	9:56:00

Fuente: elaboración propia.

3.2. Monitoreo de largo plazo

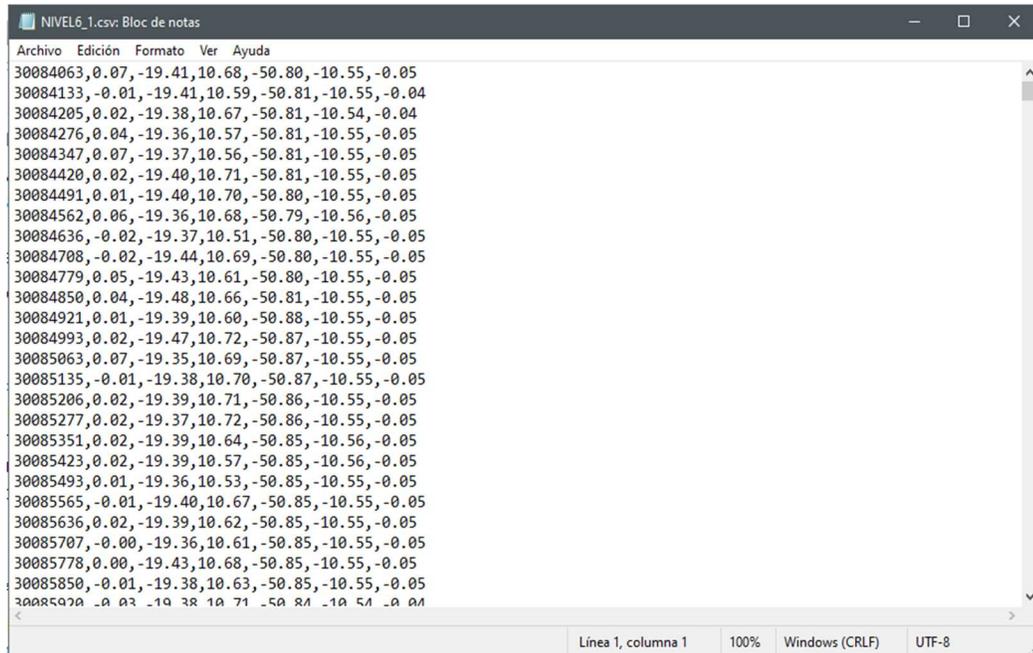
Los sensores recolectaron datos de diferentes sismos durante un mes, aproximadamente. Al desconectar los sensores y revisar los datos se observó que existen discontinuidades en los datos, lo cual se pudo haber dado por cortes de energía eléctrica o desperfectos de conexión en los sensores.

3.3. Procesamiento de datos

Posterior a extraer los datos de las memorias de los sensores se obtuvieron archivos de 545 mb, aproximadamente. Fue necesario dividir los archivos en partes de 1,000,000 lecturas para reparar errores encontrados, ya que

eventualmente faltaban comas entre datos o no se registraban datos del giroscopio.

Figura 16. Ejemplo de datos extraídos de sensores



Fuente: elaboración propia.

Luego de reparar los datos con errores encontrados en cada memoria se arreglaron las discontinuidades de los mismos. El contador de milisegundos que se ubica en la primera columna del archivo de datos se reiniciaba en ciertos puntos, por lo que se debió de reiniciar la hora y fecha de inicio con el punto anterior registrado. Luego de revisar los sensores de todos los niveles se observó que las discontinuidades no aparecían a la misma hora y fecha en los sensores de los diferentes niveles, por lo que probablemente se deba a fallos en componentes individuales de los sensores. Se cambiará la programación de los

sensores para que se reinicie el conteo de tiempo cada día, y así facilitar la localización y corrección de estos sensores.

Tabla IX. **Localización de errores de continuidad en sensores**

NIVEL 6	FECHA Y HORA FINAL DE LECTURA	LOCALIZACIÓN DE ERROR
0	'2021-08-27T17:27:24.063 +0000'	
1	'2021-08-28T19:43:58.068 +0000'	
2	'2021-08-30T11:07:00.698 +0000'	
3	'2021-09-01T01:54:29.360 +0000'	760828
	'2021-09-01T15:39:36.674 +0000'	
4	'2021-09-04T09:14:37.362 +0000'	
5	'2021-09-07T16:03:39.094 +0000'	
6	'2021-09-11T12:01:36.459 +0000'	
7	'2021-09-15T21:02:17.777 +0000'	
8	'2021-09-16T00:15:35.419 +0000'	28759
	'2021-09-20T18:55:04.938 +0000'	
9	'2021-09-26T05:19:15.456 +0000'	
10	'2021-10-02T04:32:30.633 +0000'	
11	'2021-10-08T16:46:36.659 +0000'	
12	'2021-10-09T12:31:54.363 +0000'	120802
	'2021-10-10T19:53:24.854 +0000'	

Fuente: elaboración propia.

3.4. Comparación de frecuencias de vibración teóricas y experimentales

Se compararon los periodos y frecuencias de vibración del edificio obtenidos mediante los sensores con periodos y frecuencias obtenidos en un programa de análisis estructural.

3.4.1. Selección de sismo de estudio y extracción de datos

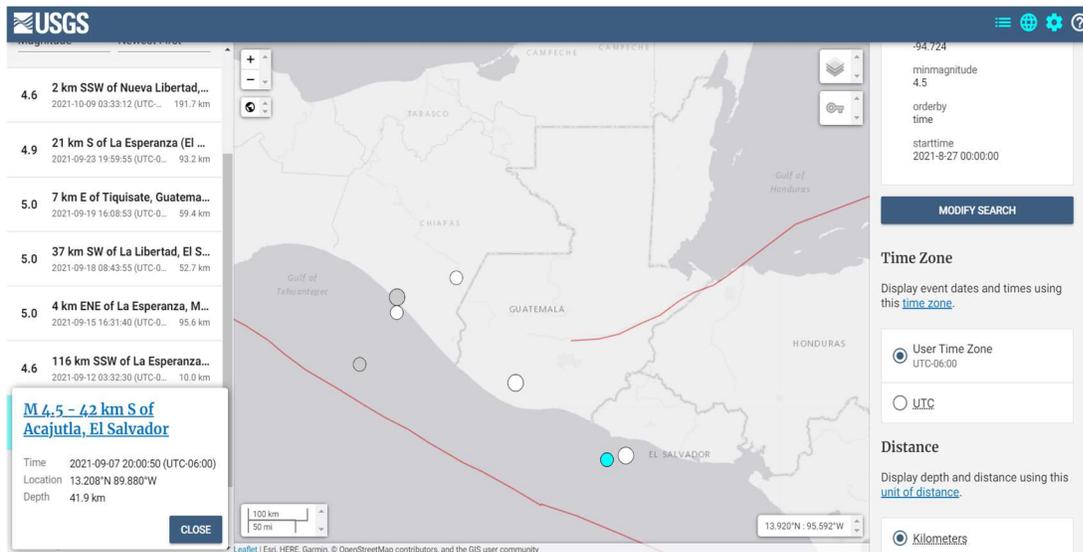
Para comparar los periodos de vibración teóricas y las experimentales obtenidas mediante los sensores de aceleración se buscaron sismos ocurridos en el periodo en el cual se registraron datos con los dispositivos, en el catálogo de sismos del Servicio Geológico de Estados Unidos.

Se identificaron sismos con magnitud mayor a 4.5 en el área de influencia de Guatemala, que corresponde a medidas que van desde 11.674 a 19.415 de latitud y -93.933 a -87.1 de longitud, en el periodo del 28 de agosto al 10 de octubre de 2021.

Luego de obtener los datos de sismos en la región, se identificaron en los registros de los acelerómetros; se utilizó el sismo que se tenía mejores registros en los sensores, ya que existían sismos mayores a 4.5 en magnitud que no se registraron correctamente.

Se seleccionó el sismo del 7 de septiembre de 2021, el cual tuvo epicentro a 42 km al sur de Acajutla, El Salvador. Dicho evento fue registrado en los 6 sensores colocados, luego de observar la duración en la que se registró la vibración en los sensores, se extrajeron los datos de un periodo de 167 segundos de los 6 sensores colocados.

Figura 17. Sismos registrados en el periodo de monitoreo



Fuente: USGS (2021) *Sismos registrados*. Consultado el 12 de diciembre de 2021. Recuperado de <https://earthquake.usgs.gov/earthquakes/search/>.

3.4.2. Modelado del edificio de estudio

Se solicitó al propietario los planos del edificio a monitorear. El edificio tiene 20 años de antigüedad y fue diseñado y construido con un sistema de marcos de concreto reforzado. Se realizó un modelo del edificio considerando los elementos presentados en el diseño, tomando las siguientes consideraciones:

- Sobre cargas muertas de 155 kg/m^2

Tabla X. **Sobrecargas muertas**

Entrepiso	Wv [kg/m²]
Azulejo de cerámica sobre cama de mortero 25 mm	110
Instalaciones	25
Acabados	20
Total	155

Fuente: elaboración propia.

- Cargas vivas de 200 kg/m² y 500 kg/m² en pasillos, de acuerdo con la tabla 3.7.1-1 de la norma NSE 2 (AGIES, 2018).
- Cargas de tabiques + techo liviano en ultimo nivel 800 kg/m

Tabla XI. **Cargas de tabiquería, último nivel**

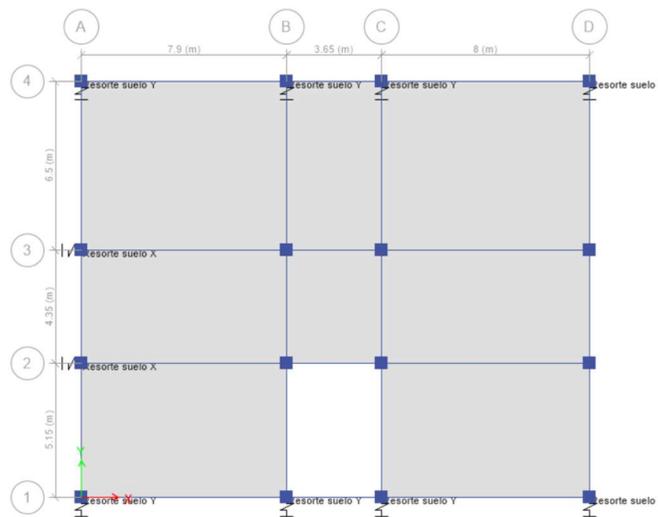
Tabiquería último nivel	Wv [kg/m]
Muros de unidades de block - tipo b	658.56
Acabados	56
Total	714.56

Fuente: elaboración propia.

- Viva de techo 50 kg/m²
- Carga de estructura de techo liviano de lámina 12 kg/m², de acuerdo con el anexo B de la norma NSE 2 (AGIES, 2018).

Se asignaron restricciones de tipo resorte de acuerdo con el valor del módulo de Winkler calculado en la tabla XII, o en los 3 lados que presenta restricciones simulando confinamiento del suelo.

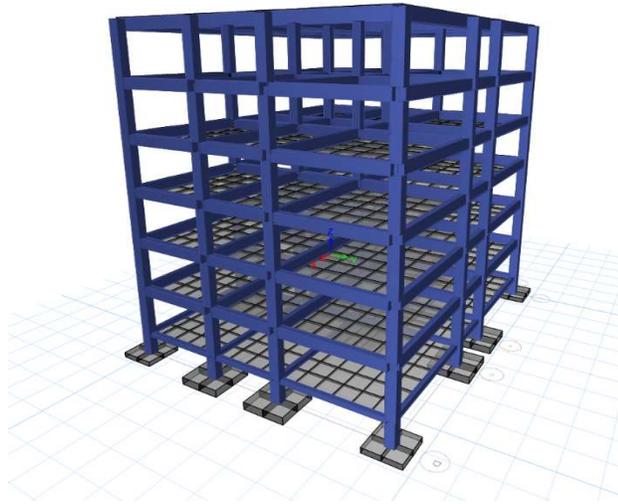
Figura 18. **Asignación de resortes de confinamiento de suelo**



Fuente: elaboración propia, empleando Visio.

- Se asignaron diafragmas rígidos en todos los niveles

Figura 19. **Modelo de edificio en programa de análisis estructural**



Fuente: elaboración propia, empleando el programa ETABS.

3.4.3. Consideraciones del suelo

Al no contar con estudio de suelos de la zona, se consultó el valor soporte de suelo que se utilizó para el diseño del edificio siendo este de 20 Ton/m². Para determinar el módulo de Winkler se utilizó la correlación de Labdad (2012); esta toma en consideración el valor soporte, mediante valores experimentales encontrados con diferentes suelos; el módulo de Winkler queda de la siguiente manera:

Tabla XII. **Cálculo de módulo de Winkler**

qs	20.00	Ton/m ²
qs	2.00	kg/cm ²
FS	3.00	
qu	60.00	Ton/m ²
Qs	6.00	kg/cm ²

Continuación de la tabla XII.

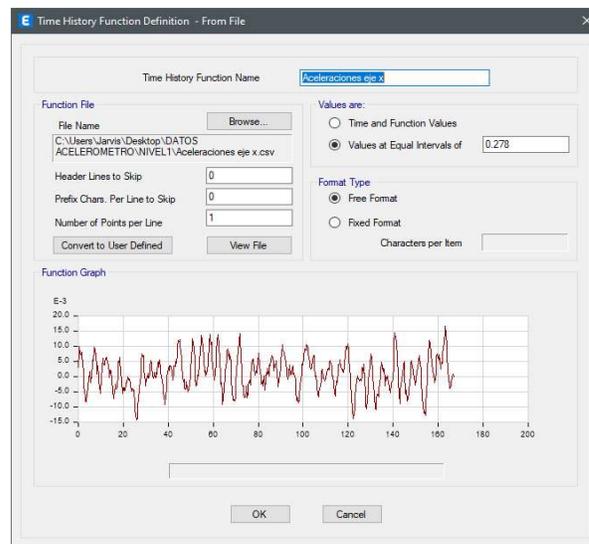
Ancho zapata	2.90	M
Largo zapata	2.90	M
Kbl	3000.00	Ton/m ³

Fuente: elaboración propia.

3.4.4. Carga de aceleraciones al programa

Se cargó al programa las aceleraciones del sensor colocado en el nivel 1. Este se ingresó como función de tiempo historia al programa. Se cargaron también las aceleraciones del eje X y del eje Y con datos en intervalos de 0.278 segundos, entre ellos.

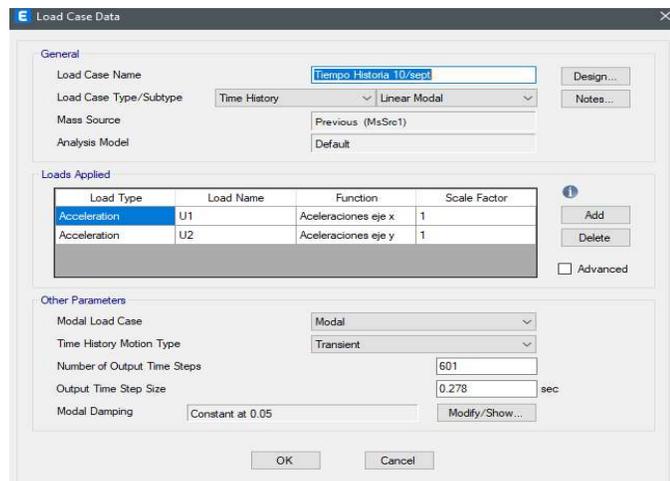
Figura 20. Carga de aceleraciones como tiempo historia



Fuente: elaboración propia, empleando el programa ETABS.

Se creó un caso de carga tiempo historia de tipo linear modal. Se asignaron en cada dirección las aceleraciones del dispositivo. El factor de escala fue de 1, considerando que no necesitan amplificarse los registros; se crearon 601 pasos de 0.278 segundos como parámetros de análisis.

Figura 21. Caso de carga tiempo historia



Fuente: elaboración propia, empleando el programa ETABS.

3.4.5. Cálculo de periodo de vibración teórico

Para poder comparar las frecuencias de vibración se utilizó la formula empírica presentada por AGIES en la norma NSE-3 (AGIES, 2018) y el cálculo de tipo modal que realiza el programa ETABS. Para calcular el periodo con la formula empírica de AGIES. Se utilizó una altura libre de 35m, considerando que cada nivel tiene 3.2m entre ellos y tomando en cuenta que el nivel del sótano no se encuentra restringido del lado del ingreso y puede presentar deformaciones. El edificio tiene marcos estructurales de concreto con fachadas rígidas, con esta fórmula se obtuvo un periodo de 0.965 segundos.

Tabla XIII. **Cálculo de periodo con fórmula empírica de AGIES**

Periodo calculado fórmula empírica		
Ta	0.965	Seg.
KT	0.047	
HN	35.000	m
X	0.850	
Altura de piso	3.200	
N	7	

Fuente: elaboración propia.

Para calcular el periodo teórico se realizó un análisis modal, utilizando ETABS, obteniéndose los siguientes modos de vibración:

Tabla XIV. **Periodos y frecuencias de vibración encontradas en programa de análisis estructural**

Caso	Modo	Periodo seg.	Frecuencia cyc/sec.	Frecuencia angular rad/sec.
Modal	1	1.081	0.925	5.811
Modal	2	1.028	0.973	6.112
Modal	3	0.892	1.120	7.040
Modal	4	0.327	3.061	19.233
Modal	5	0.312	3.203	20.126
Modal	6	0.279	3.589	22.549
Modal	7	0.176	5.696	35.787
Modal	8	0.170	5.887	36.990
Modal	9	0.154	6.501	40.845
Modal	10	0.154	6.501	40.845
Modal	11	0.154	6.501	40.845
Modal	12	0.154	6.501	40.845

Fuente: elaboración propia.

3.4.6. Cálculo de periodo de vibración experimental

Para calcular el periodo de vibración experimental se utilizó la función *findpeaks* de MATLAB. Esta encuentra los picos locales a partir de una serie de datos de aceleración, y muestra los valores de los puntos y su ubicación en el sensor. Posteriormente se determinan las separaciones temporales de dichos picos. Estas separaciones son los periodos de las ondas cíclicas de vibración. Para encontrar el periodo de cada piso se calcula el promedio de tiempo entre todos los picos.

$$T_{piso} = \frac{\sum T_{picoi}}{n_{picos}} \quad (10)$$

Luego de procesar los datos de todos los sensores se encontraron los siguientes resultados de periodos de vibración por piso:

Tabla XV. Periodos calculados de datos de los sensores

Nivel	Periodo (s)
1	1.553
2	1.478
3	2.390
4	1.270
5	1.150

Fuente: elaboración propia.

3.4.7. Comparación de periodos teóricos y experimentales

Luego de realizar el análisis tiempo-historia del sismo en estudio, se observó que el edificio muestra un movimiento traslacional en el cual predomina

el periodo de vibración 2, el cual es traslacional y por lo tanto el periodo de vibración es el mismo en todos los niveles, por lo que se calculó el porcentaje de variación entre las frecuencias teóricas y experimentales para compararlas.

Tabla XVI. **Comparación de periodos teóricos y experimentales**

Nivel	Periodo empírico fórmula AGIES	Periodo teórico	Periodo experimental (s)	Variación
1	0.965	1.020	1.553	+52 %
2	0.965	1.020	1.478	+45 %
3	0.965	1.020	2.390	+134 %
4	0.965	1.020	1.270	+25 %
5	0.965	1.020	1.150	+13 %

Fuente: elaboración propia.

Se puede observar que el periodo experimental aumenta en un rango de 12.75 – 52.25 % en todos los niveles, con excepción del nivel 3 en el cual cambia en un 134.31 %, lo cual se asume como error instrumental y se cambiará el sensor para posteriores mediciones.

3.5. Inferencia de daños estructurales y su localización

El periodo de vibración de una estructura depende de dos variables: la masa y la rigidez de la estructura; la masa no tiene variación significativa durante la vida útil de la estructura si no cambia de ocupación; sin embargo, la rigidez puede variar a medida que algunos de los elementos estructurales presentan degradación (agrietamiento, fluencia del acero, pérdida de recubrimiento, entre otros).

$$T_F = 2\pi \sqrt{\frac{\sum_{i=1}^n (W_i u_i^2)}{g \sum_{i=1}^n (F_i u_i)}} \quad (11)$$

La fórmula analítica presentada por AGIES que utiliza el método de Rayleigh, contiene términos que al ser reordenados representan rigidez. Se puede observar qué cambios en la rigidez de la estructura cambiarían el valor del periodo de vibración, por lo que los incrementos en el periodo de vibración, en función del periodo presentado en condiciones relativamente sanas de la estructura, podrían ser indicadores de pérdidas de rigidez en la estructura.

3.6. Selección de método de análisis de daños

Se realizará una comparación de periodos de vibración de la estructura relativamente sana y otra con daño, en otro momento de la ocupación; para obtener mejores resultados es recomendable tener datos del periodo al inicio de su ocupación para calibrar el modelo analítico.

Para determinar el daño en la estructura es necesario definir la curva de desempeño de una edificación. En la norma SEAOC de 1995 (Bertedero, 1995) se definen estados límites con base en la ocupación y reparación de una estructura.

Figura 22. **Estados límites de desempeño de estructuras**



Fuente: Melone (2019). *Desempeño sísmico de edificaciones*. Consultado el 5 de diciembre de 2021. Recuperado de <https://aulaseproinca.blogspot.com/2019/02/desempeno-sismico-de-edificaciones.html>.

3.6.1. Estado límite totalmente operacional

La norma define este estado límite como el punto de la gráfica de corte basal y desplazamiento en la que todos los componentes del edificio se encuentran dentro del límite elástico (Bertedero, 1995).

3.6.2. Estado límite operacional

En este punto ya existen elementos que entran al rango plástico; se presentan daños moderados en elementos no estructurales e incluso daños ligeros en elementos estructurales. Pueden requerirse reparaciones menores (Bertedero, 1995).

3.6.3. Estado límite seguridad

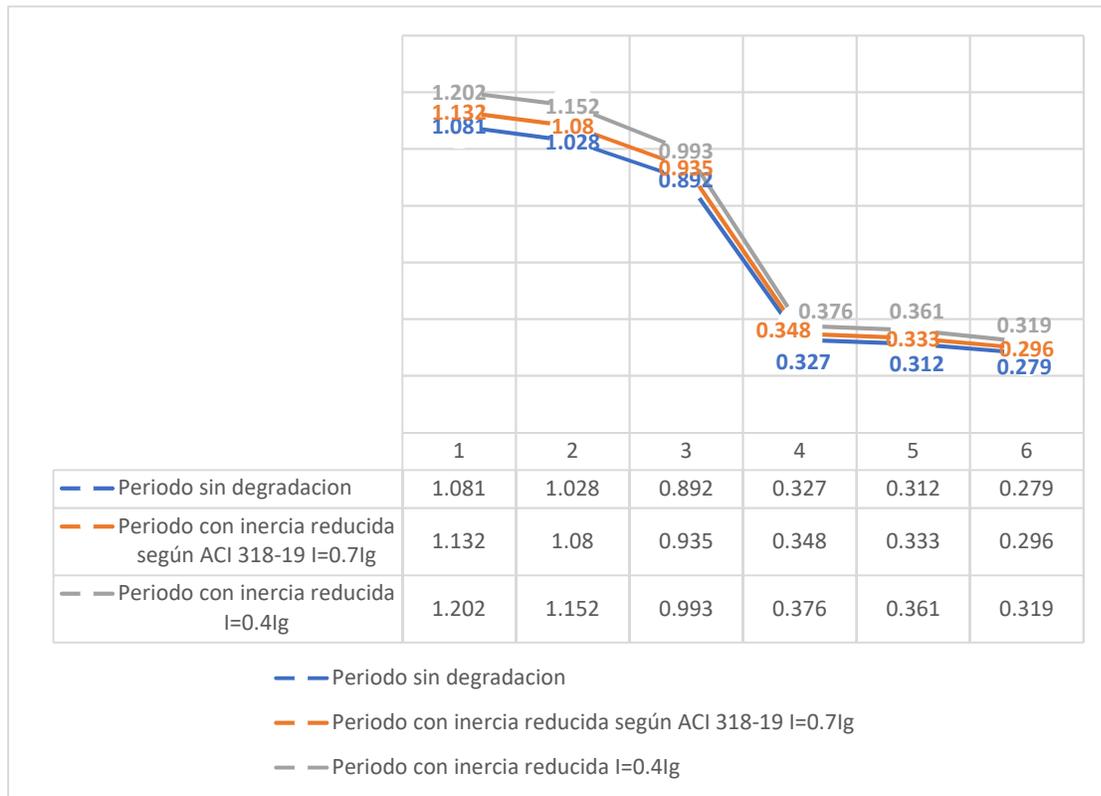
En este estado límite se degrada la rigidez y se comienza a producir daños moderados en los elementos estructurales, las instalaciones quedan fuera de servicio y con posibles reparaciones a realizar (Bertedero, 1995).

3.6.4. Estado límite precolapso

En este punto la degradación de rigidez compromete la estabilidad de la estructura, debiendo evacuar inmediatamente la estructura y las reparaciones pueden no ser factibles de realizar.

La norma ACI 318-19 para el diseño de elementos de concreto reforzado presenta una manera de simular la degradación del elemento mientras se realiza el análisis elástico; utilizando el modelo analítico del edificio se modeló la degradación de los elementos y se verificó la variación de periodos de vibración; asimismo se utilizó un valor de reducción mayor para evaluar el comportamiento mientras continua la degradación.

Figura 23. **Comparación de periodos con momento de inercia reducido en columnas**



Fuente: elaboración propia, empleando Excel.

Se observó que el periodo de vibración aumenta a medida que los elementos se fisuran; esto puede ser percibido en el monitoreo de un edificio si se comienza en una etapa en la que dicho edificio se encuentra en una condición relativamente sana, en que los elementos no se encuentran fisurados o comienzan a fisurarse. Por lo cual durante el monitoreo de salud estructural a largo plazo se pueden detectar los cambios de estados límites en seguridad y en lo operacional, observando cambios drásticos en periodos de vibración entre eventos sísmicos.

4. DESARROLLO DE SOFTWARE DE DETECCIÓN DE DAÑOS

Para procesar y sacar conclusiones de los datos obtenidos de los sensores, se creó un programa de detección de daños en la estructura.

4.1. Procesamiento de datos de arreglo de microcontroladores

Los sensores para el monitoreo de salud estructural almacenan datos en formato *.csv. Estos presentan datos de aceleración en 3 ejes en cada dirección durante un periodo largo de monitoreo. El objetivo de esta investigación es comparar frecuencias y periodos de vibración durante eventos sísmicos importantes, por lo cual se requiere de una herramienta que extraiga los datos necesarios de todos los sensores.

Para crear el software de procesamiento se utilizó el programa MATLAB, el cual permite manejar grandes cantidades de datos y posee la ventaja de tener una herramienta para crear interfaz gráfica con gran facilidad.

El software se divide en dos partes: la primera tiene como objetivo extraer los datos de aceleración de determinado evento sísmico, mientras que la segunda se encarga del procesamiento de aceleraciones de todos los niveles y encuentra periodos y frecuencias de vibración para compararse con las obtenidas de un modelo construido en software de análisis estructural.

4.2. Programa de obtención de datos

El programa debe comenzar cargando los datos de los sensores; para esto se utiliza el comando `load` de MATLAB que permite cargar datos de un archivo separado por comas `*.csv`. Los archivos extraídos de los sensores tendrán el nombre de `DATALOG1.CSV` y cambiará el número dependiendo del nivel. Se nombra el vector `accel`, por lo cual la carga de datos quedará de esta manera:

```
accel=load('DATALOG.CSV');
```

Posteriormente se asigna un vector individual a cada vector de datos, se establece como `"tmili"` al vector de tiempo en milisegundos; se asigna como `"ax"` al vector de aceleraciones en X, como `"ay"` al vector de aceleraciones en Y, como `"az"` al vector de aceleraciones en Z:

```
n=length(accel);  
tmili=accel(:,1);  
ax=accel(:,2);  
ay=accel(:,3);  
az=accel(:,4);
```

Se debe establecer el formato de tiempo que va a utilizarse, el cual tendrá valores de año, mes, día, minuto, segundo y milisegundo; se asigna a la zona horaria su formato. Posterior a ello se transforma el vector `"tmili"` en formato de milisegundos; por último, se crea el vector `"out"` el cual toma como base el tiempo asignado y se suma el vector de milisegundos:

```
T =  
Datetime(year1,mes1,dia1,hora1,minutos1,segundos1,0,'TimeZone','UTC'  
, 'F', 'uuuu-MM-dd"T"HH:mm:ss.SSS Z');
```

```

addMS = milliseconds(accel(:,1));
out = T + addMS;

```

Se crea un vector del mismo tipo para el tiempo que se deberá ingresar de acuerdo con los registros sísmicos. Para encontrar los datos entre los vectores de tiempo se crea la variable “duración”. La cual el usuario deberá ingresar en segundos. Por lo tanto, el tiempo de inicio es igual al tiempo ingresado. Entonces el tiempo final se le deben de sumar los segundos establecidos en la duración de evento ingresada:

```

T2 =
datetime(year2,mes2,dia2,hora2,minutos2,segundos2,0,'TimeZone','UTC',
'F','uuuu-MM-dd"T"HH:mm:ss.SSS Z');
Tinicio=datetime(year2,mes2,dia2,hora2,minutos2,segundos2,0,'TimeZon
e','UTC','F','uuuu-MM-dd"T"HH:mm:ss.SSS Z');
Tfinal=datetime(year2,mes2,dia2,hora2,minutos2,(segundos2+duracion),
0,'TimeZone','UTC','F','uuuu-MM-dd"T"HH:mm:ss.SSS Z');

```

Para extraer los valores del vector de tiempo, específicamente del que se necesita estudiar, se crea un ciclo de tipo “*while*” que se correrá mientras la variable tipo contador “im” sea menor al valor de tiempo final:

```

im=1;
j=1;
while out(im,1)<=Tfinal
    if out(im,1)>=Tinicio
        tiempo(j,1)=out(im,1);
        j=j+1;
    end

```

```
    im=im+1;  
end
```

Al finalizar el ciclo, se creará un vector de nombre tiempo en el que únicamente se tienen los valores de tiempo del evento que se quiere analizar. Posterior a esto se declaran nuevas variables para poder aplicar el filtro que se presentó en el capítulo 2. Finalmente, se establece un punto de partida para el ciclo que se correrá, siendo estos valores de aceleración en “X” y en “Y” de 0, y de 9.81 para “Z”:

```
gxFilt=0;  
gyFilt=0;  
gzFilt=9.81;  
alpha=filtro;  
i=1;  
nx=length(ax);  
ny=length(ay);  
nz=length(az);  
AcFiltx=zeros(n,1);  
AcFilty=zeros(n,1);  
AcFiltz=ones(n,1);  
AcFlitz=AcFiltz*9.81;  
i=1;
```

Posterior a esto se correrá un ciclo *while*, que se detendrá hasta que el contador “i” sea iguala “n”; esto corresponde a la longitud del vector de aceleración. Se declara la constante de filtro y se programa la ecuación (9) en el ciclo:

```

while i<n
    gx=ax(i);
    gxFilt = (1 - alpha)*gxFilt + alpha*gx;
    AcFiltx(i)=gxFilt;

    gy=ay(i);
    gyFilt = (1 - alpha)*gyFilt + alpha*gy;
    AcFilty(i)=gyFilt;

    gz=az(i);
    gzFilt = (1 - alpha)*gzFilt + alpha*gz;
    AcFiltz(i)=gzFilt;

    i=i+1;
end

```

Se deben crear vectores de 2 columnas que contengan el tiempo en milisegundos del evento y la magnitud de la aceleración; esto con el fin de exportar los vectores como archivos de datos *.csv para poder ser utilizados en programas de análisis estructural. En la parte del programa que calcula parámetros estructurales se asigna la primera columna con los datos del vector tiempo y en la segunda, los de las aceleraciones con datos filtrados:

```

tiempo0 = seconds(addMS((im-j+2):im));
tiempo0 = tiempo0 - tiempo0(1);
aceleracionesx = zeros(j-1,2);
aceleracionesy = zeros(j-1,2);
aceleracionesz = zeros(j-1,2);
aceleracionesx(:,1) = tiempo0;

```

```

aceleracionesy(:,1) = tiempo0;
aceleracionesz(:,1) = tiempo0;
aceleracionesx(:,2) = AcFiltx((im-j+2):im);
aceleracionesy(:,2) = AcFilty((im-j+2):im);
aceleracionesz(:,2) = AcFiltz((im-j+2):im);

```

Para que se tenga mejor visualización de parte del usuario respecto de la cantidad de filtro aplicado y del tipo de evento sísmico, se grafican las aceleraciones en los 3 ejes:

```

subplot(3,2,1);
plot(tiempo(1:j-1),AcFiltx((im-j+2):im));
title('Aceleración x (m/s2)')
ylabel('m/s2')
grid on
grid minor
hold on

```

```

subplot(3,2,3);
plot(tiempo(1:j-1),AcFilty((im-j+2):im));
title('Aceleración y (m/s2)')
ylabel('m/s2')
grid on
grid minor
hold on

```

```

subplot(3,2,5);
plot(tiempo(1:j-1),AcFiltz((im-j+2):im));
ylabel('m/s2')

```

```
grid on
grid minor
title('Aceleración z (m/s2)')
hold on
```

4.3. Programa de procesamiento de datos y conversión a parámetros estructurales

El objetivo de la investigación es comparar los periodos y frecuencias teóricas de la estructura y los medidos por los acelerógrafos. Para dicho fin los periodos teóricos serán obtenidos por medio de software de análisis estructural, simulando condiciones ideales. Los periodos experimentales serán calculados con base en los registros de los acelerómetros, lo cual se realiza mediante el siguiente código:

Antes de iniciar el cálculo, es necesario extraer los datos de todos los niveles con el software de obtención de datos. Esto generará archivos .csv de los 3 ejes de todos los niveles, los cuales deben de ser cargados al programa:

```
accelX1=load('Aceleraciones eje x.csv');
accelY1=load('Aceleraciones eje y.csv');
accelZ1=load('Aceleraciones eje z.csv');

accelX2=load('Aceleraciones eje x1.csv');
accelY2=load('Aceleraciones eje y1.csv');
accelZ2=load('Aceleraciones eje z1.csv');

accelX3=load('Aceleraciones eje x2.csv');
accelY3=load('Aceleraciones eje y2.csv');
```

```
accelZ3=load('Aceleraciones eje z2.csv');
```

```
accelX4=load('Aceleraciones eje x3.csv');  
accelY4=load('Aceleraciones eje y3.csv');  
accelZ4=load('Aceleraciones eje z3.csv');
```

```
accelX5=load('Aceleraciones eje x4.csv');  
accelY5=load('Aceleraciones eje y4.csv');  
accelZ5=load('Aceleraciones eje z4.csv');
```

```
accelX6=load('Aceleraciones eje x5.csv');  
accelY6=load('Aceleraciones eje y5.csv');  
accelZ6=load('Aceleraciones eje z5.csv');
```

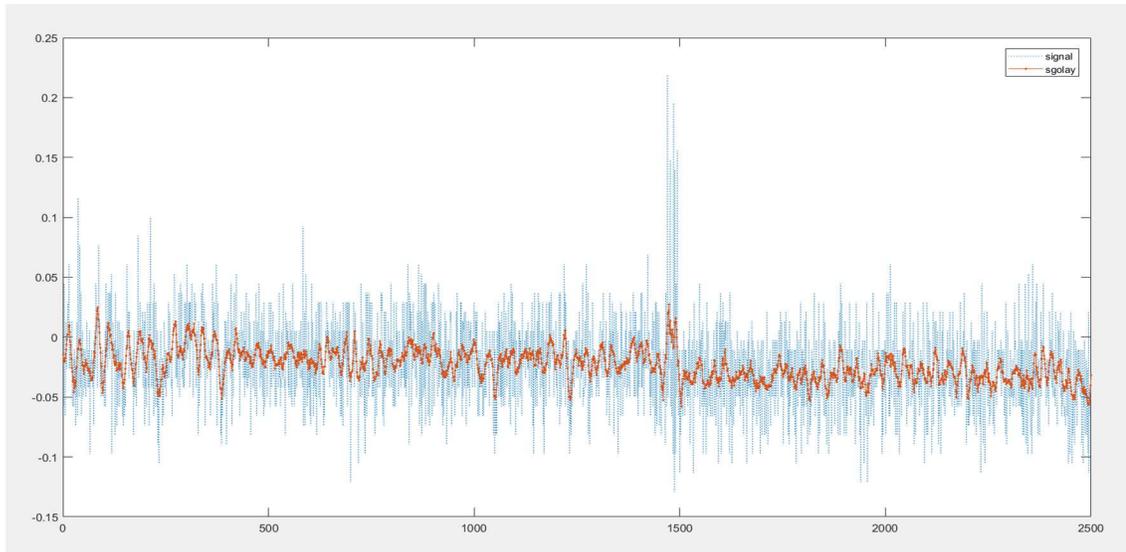
```
accelX7=load('Aceleraciones eje x6.csv');  
accelY7=load('Aceleraciones eje y6.csv');  
accelZ7=load('Aceleraciones eje z6.csv');
```

Los datos importados están filtrados con el método de promedio móvil exponencial. Sin embargo, para calcular los periodos respecto de los picos de aceleración es necesario tenerlos bien definidos, por lo cual se aplicará el filtro de Savitzky–Golay.

4.3.1. Filtro Savitzky–Golay

El filtro Savitzky–Golay se basa en una regresión polinomial local. Esta regresión permite conservar características de la distribución inicial, lo cual proporcionará una función similar pero suavizada.

Figura 24. **Aceleraciones sin filtrar y filtradas por medio del filtro de Savitzky-Golay**



Fuente: elaboración propia, empleando Excel.

El código quedaría de la siguiente manera:

```
%%%%% Datos filtro aplicado
order = 3;
framelen = 21;

%%%%%%%%%% Aplicar filtro
sgfX1 = sgolayfilt(accelX1(:,2),order,framelen);
sgfY1 = sgolayfilt(accelY1(:,2),order,framelen);
sgfZ1 = sgolayfilt(accelZ1(:,2),order,framelen);

sgfX2 = sgolayfilt(accelX2(:,2),order,framelen);
sgfY2 = sgolayfilt(accelY2(:,2),order,framelen);
```

```
sgfZ2 = sgolayfilt(accelZ2(:,2),order,framelen);
```

```
sgfX3 = sgolayfilt(accelX3(:,2),order,framelen);
```

```
sgfY3 = sgolayfilt(accelY3(:,2),order,framelen);
```

```
sgfZ3 = sgolayfilt(accelZ3(:,2),order,framelen);
```

```
sgfX4 = sgolayfilt(accelX4(:,2),order,framelen);
```

```
sgfY4 = sgolayfilt(accelY4(:,2),order,framelen);
```

```
sgfZ4 = sgolayfilt(accelZ4(:,2),order,framelen);
```

```
sgfX5 = sgolayfilt(accelX5(:,2),order,framelen);
```

```
sgfY5 = sgolayfilt(accelY5(:,2),order,framelen);
```

```
sgfZ5 = sgolayfilt(accelZ5(:,2),order,framelen);
```

```
sgfX6 = sgolayfilt(accelX6(:,2),order,framelen);
```

```
sgfY6 = sgolayfilt(accelY6(:,2),order,framelen);
```

```
sgfZ6 = sgolayfilt(accelZ6(:,2),order,framelen);
```

```
sgfX7 = sgolayfilt(accelX7(:,2),order,framelen);
```

```
sgfY7 = sgolayfilt(accelY7(:,2),order,framelen);
```

```
sgfZ7 = sgolayfilt(accelZ7(:,2),order,framelen);
```

4.3.2. Periodos y frecuencias

Para encontrar los periodos y frecuencias se utilizan los picos máximos y mínimos de las aceleraciones registradas, esto mediante la función *findpeaks* de MATLAB, la cual crea un vector con los valores picos y las posiciones de los valores en dicho vector:

```
[pkx1,locsx1] = findpeaks(sgfX1);
[pkxy1,locsy1] = findpeaks(sgfY1);
[pkz1,locsz1] = findpeaks(sgfZ1);
```

Posterior a eso se crea un ciclo de tipo *while*, el cual correrá hasta que el contador “i” sea menor a la longitud del vector “locsx1” el cual contiene los picos de aceleración. En el interior del ciclo se crea un nuevo vector llamado “Tcalcx” el cual almacenará los datos de la resta entre el tiempo de a_i y a_{i+1} , y dará como resultado los tiempos entre picos:

```
i=1;
%Encontrar periodo nivel 1
while i<nlocsx1
    Tcalcx1(i)= accelX1(locsx1(i+1),1)-accelX1(locsx1(i),1);
    i=i+1;
end
```

Para calcular los periodos por nivel, se determina el promedio del tiempo entre picos en cada eje de aceleración. Posterior a ello se encuentra el inverso de la relación para hallar la frecuencia:

```
% calcular periodos y frecuencias
%%% Nivel 1
Tx1 = mean(Tcalcx1); % periodo en segundos
fx1 = 1/Tx1; % frecuencia en Hz
Ty1 = mean(Tcalcy1); % periodo en segundos
fy1 = 1/Ty1; % frecuencia en Hz
Tz1 = mean(Tcalcz1); % periodo en segundos
fz1 = 1/Tz1; % frecuencia en Hz
```

4.3.3. Velocidades y desplazamientos

Es necesario mostrar gráficamente los desplazamientos del edificio por nivel para visualizar los modos de vibración; para ello se requiere determinar las velocidades y desplazamientos de cada nivel, las cuales se obtienen integrando numéricamente los valores de aceleración y velocidad, utilizando la función de MATLAB “cumtrapz”. La cual mediante aproximaciones trapezoidales integra un vector en el tiempo. Finalmente se multiplica por 100 para tener resultados en centímetros para mejor visualización:

```
% Integrar y encontrar velocidades y desplazamientos
%%% Nivel1
vx1 = cumtrapz(accelX1(:,2),sgfX1)*100; %velocidad en x en cm/s
ux1 = cumtrapz(accelX1(:,2),vx1)*100; %Desplazamiento en x en cm
vy1 = cumtrapz(accelY1(:,2),sgfY1)*100; %velocidad en x en cm/s
uy1 = cumtrapz(accelY1(:,2),vy1)*100; %Desplazamiento en x en cm
```

4.4. Creación de interfaz gráfica

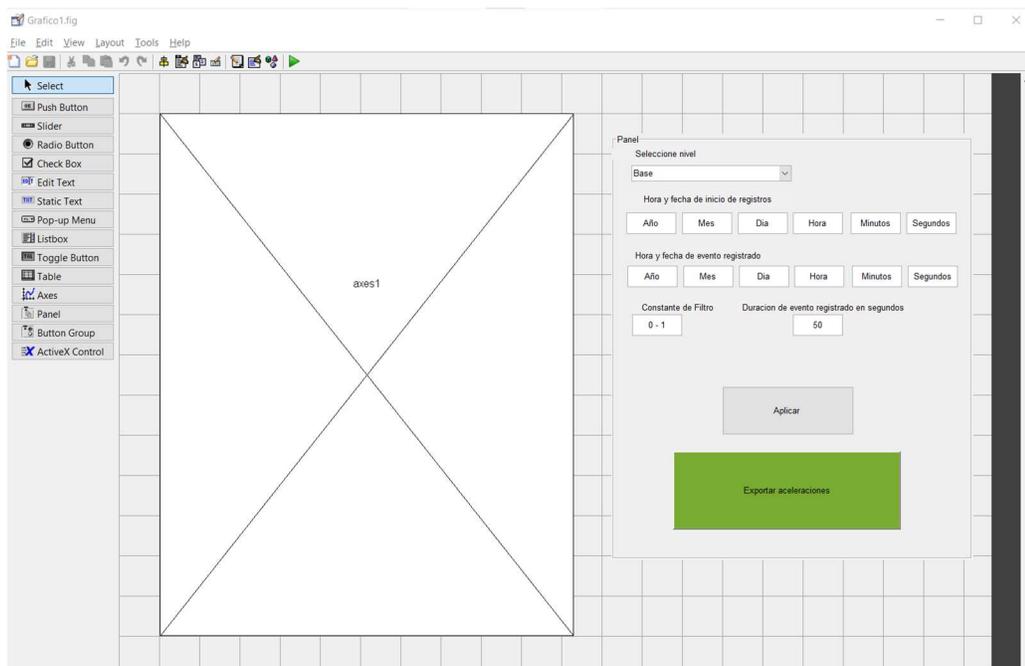
Para crear una interfaz que pueda ser utilizada por ingenieros sin ningún conocimiento de programación, se utiliza la herramienta “GUIDE” de MATLAB. Esta permite crear interfaz gráfica y posteriormente introducir el código utilizando botones y otras herramientas.

4.4.1. Interfaz de programa de obtención de datos

Para la obtención de datos se creó una interfaz con panel de gráficas a la izquierda en la cual se mostrarán las de aceleración. En el panel derecho se seleccionan los niveles por medio de un menú desplegable.

Para seleccionarlos se debe introducir la hora y fecha de colocación del sensor y del evento sísmico que se quiere estudiar. Se debe introducir la constante de filtro, dependiendo el grado de suavizado que se desee utilizar, y por último la duración del evento sísmico a estudiar; luego de esto se puede presionar el botón de aplicar. Se mostrarán los gráficos actualizados y se podrá modificar la constante de filtro, según consideración del ingeniero operador. Cuando se tengan en pantalla los datos requeridos, se presiona el botón de exportar aceleraciones, el cual creará archivos *.csv de aceleración y tiempo en los 3 ejes.

Figura 25. Creación de interfaz de obtención de datos

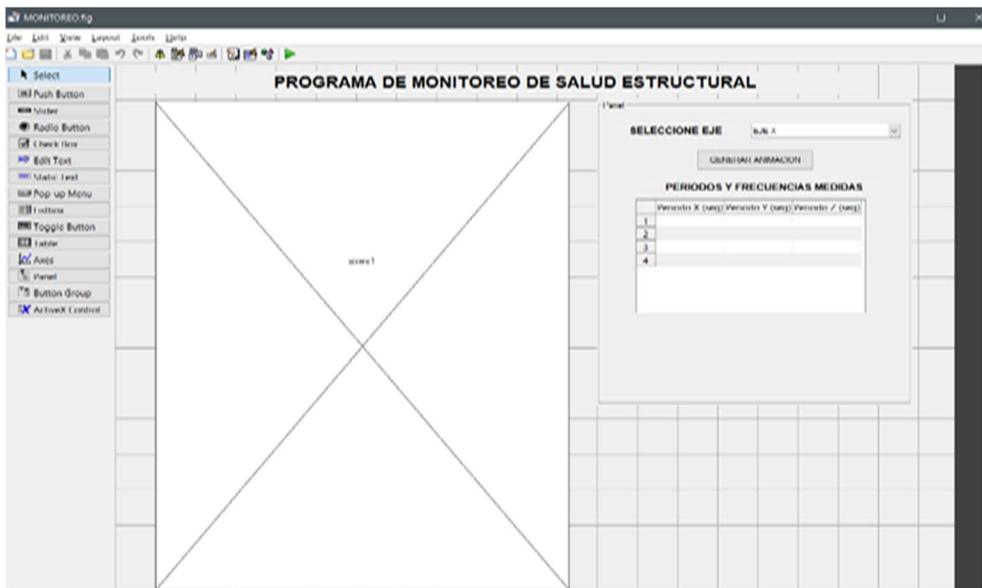


Fuente: elaboración propia, empleando la herramienta GUIDE de MATLAB.

4.4.2. Interfaz de procesamiento de datos y monitoreo estructural

Para la interfaz del programa de monitoreo de salud estructural se crea un panel izquierdo en el cual se mostrará una animación del desplazamiento de todos los niveles del edificio. En el panel derecho se podrá seleccionar el eje que se quiere observar en la animación; los parámetros de monitoreo de salud estructural se muestran en la parte derecha, en donde pueden verse los periodos por nivel registrados por los acelerómetros durante un evento sísmico.

Figura 26. Interfaz de monitoreo de salud estructural



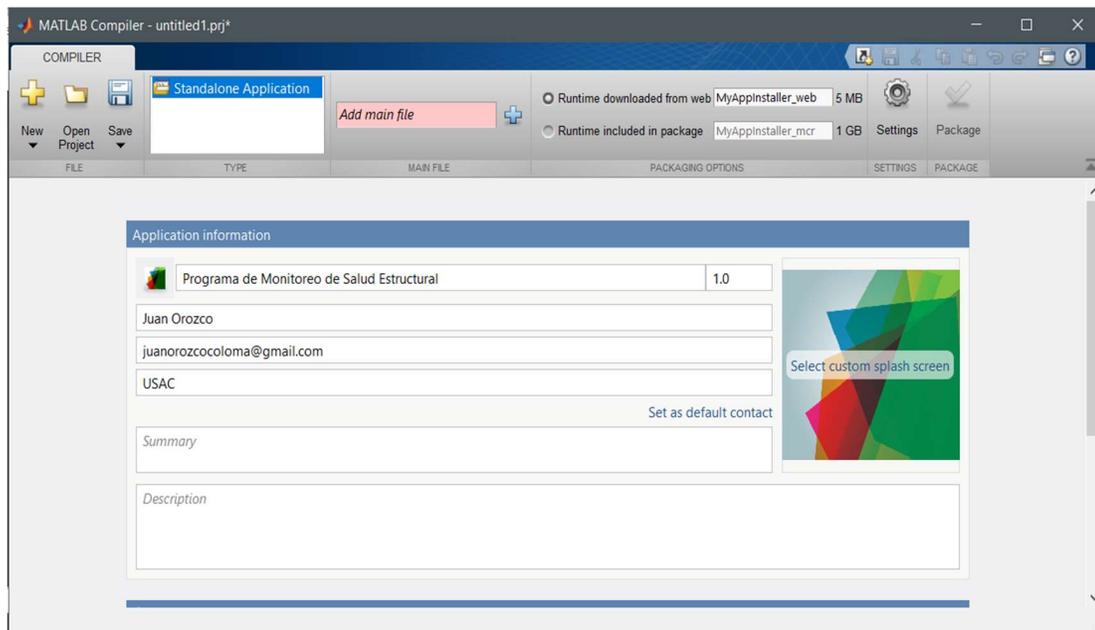
Fuente: elaboración propia, empleando la herramienta GUIDE de MATLAB.

4.5. Prueba de ejecutable

Hasta este punto el programa es ejecutable únicamente dentro del software MATLAB, por lo que es necesario crear un ejecutable de tipo *.exe para que el programa pueda utilizarse en cualquier computadora con sistema operativo de Windows.

Para exportar el programa como un ejecutable se utiliza la herramienta *Compiler* de MATLAB, la cual permite exportar un programa con la opción de descargar los paquetes necesarios de internet por el usuario final o que se incluyan en el programa; se escriben los datos del programa y se guarda el archivo.

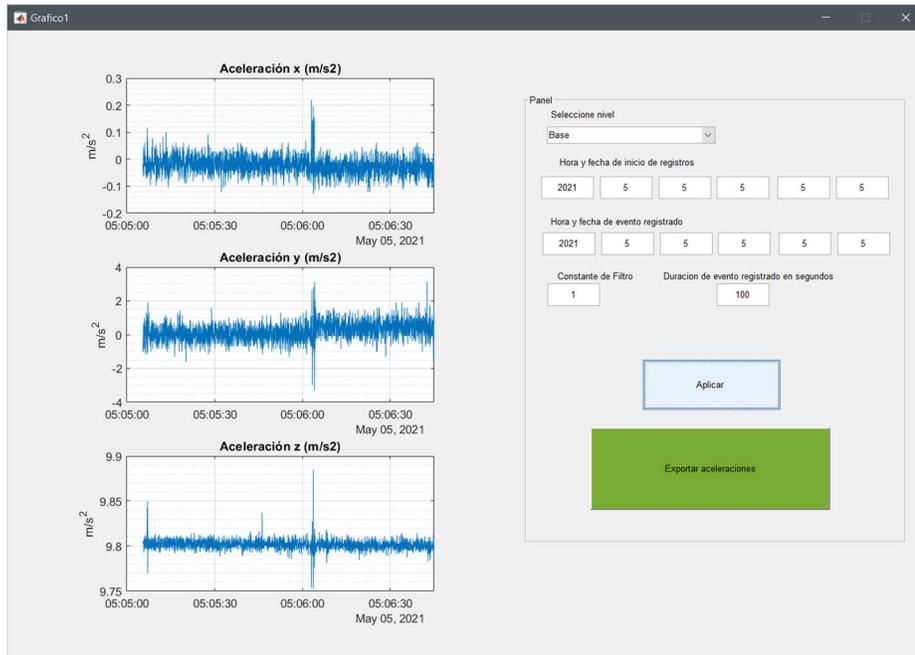
Figura 27. Creación ejecutable de programa



Fuente: elaboración propia, empleando la herramienta *Compiler* de MATLAB.

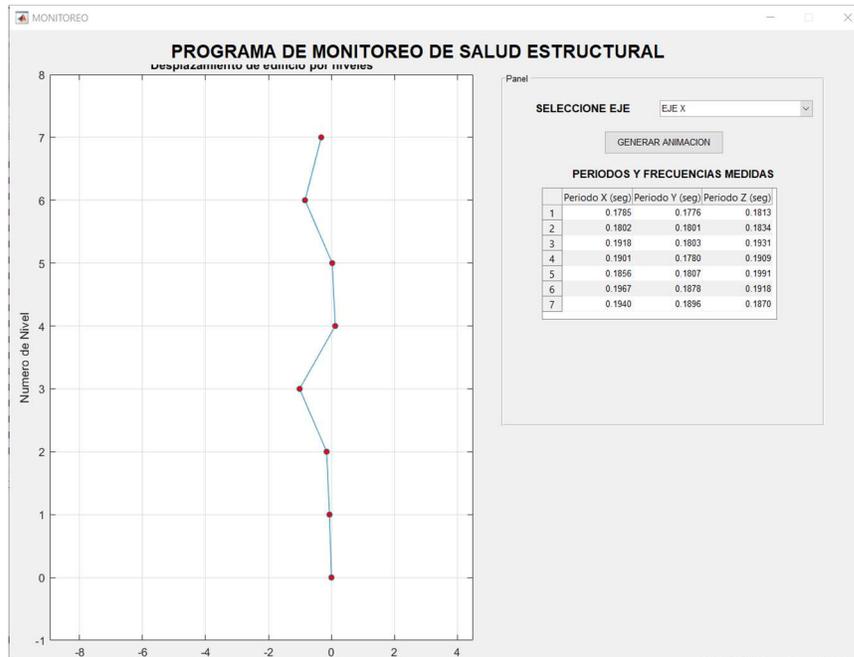
A continuación, se muestra el resultado final del software en ejecución utilizando datos de los acelerómetros colocados:

Figura 28. Programa de obtención de datos en funcionamiento



Fuente: elaboración propia, empleando la herramienta GUIDE de MATLAB.

Figura 29. Programa de monitoreo de salud estructural



Fuente: elaboración propia, empleando Excel.

5. MANUAL DE INSTALACIÓN Y OPERACIÓN DE DISPOSITIVOS

El manual de instalación y operación permitirá que el usuario pueda construir, instalar y operar los dispositivos de monitoreo.

5.1. Información general de los dispositivos

Los sensores de aceleración cuentan con la capacidad de registrar aproximadamente 10 lecturas de aceleración y rotaciones en 3 ejes por segundo, las cuales se almacenan en una memoria microSD; estos datos se pueden descargar en cualquier computadora para ser procesados por el software de salud estructural.

Los sensores requieren una conexión a corriente de 5 V y 2A; la cual usando convertidores convencionales puede ser conectada a corriente alterna de 110 V. Previo a la utilización de los sensores es necesario ensamblarlos y calibrarlos, por lo cual se recomienda verificar su funcionamiento y calibración previo a su colocación.

Figura 30. **Sensor de aceleración en funcionamiento**



Fuente: [Fotografía de Juan Orozco]. (Ciudad de Guatemala, 2021). Colección particular.
Guatemala.

5.2. Partes del sensor

Las partes del sensor son las siguientes:

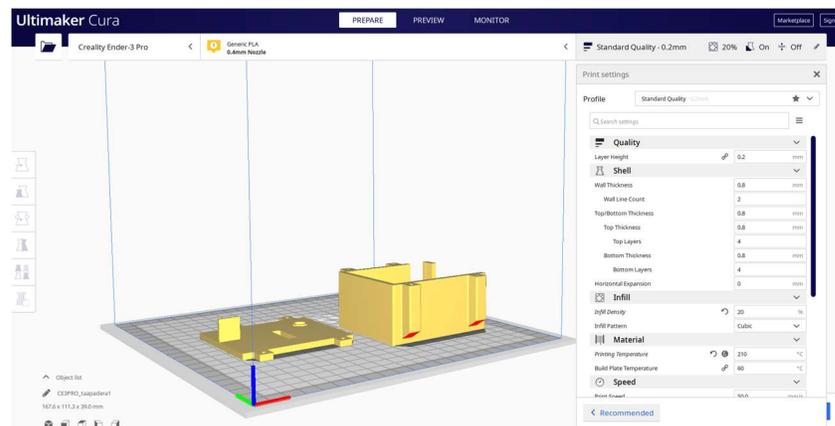
5.2.1. Caja

Los sensores cuentan con una caja compuesta de una base y una tapadera, la cual está diseñada con base en un modelo para microcontroladores Arduino Uno. Se tienen archivos de formato STL, los cuales luego de ser procesados en el software libre Cura Ultimaker, son convertidos a formato gcode. Dicho formato contiene todas las configuraciones requeridas por una impresora 3D.

Para ser impreso, se utiliza filamento de tipo PLA con las siguientes configuraciones:

- Temperatura de impresión: 200 °C (verificar especificaciones del filamento)
- Temperatura de la cama de impresión: 60 °C
- Velocidad: 50 mm/s
- Altura de la capa: 0.12 mm
- Retracción: 6 mm a 25 mm/s
- Relleno: 20 %
- Velocidad de capa inicial: 20 mm/s
- Velocidad inicial del ventilador: 0 %

Figura 31. Configuraciones de impresión 3D de la caja



Fuente: elaboración propia, empleando el programa Cura Ultimaker.

5.2.2. Microcontrolador Arduino Uno

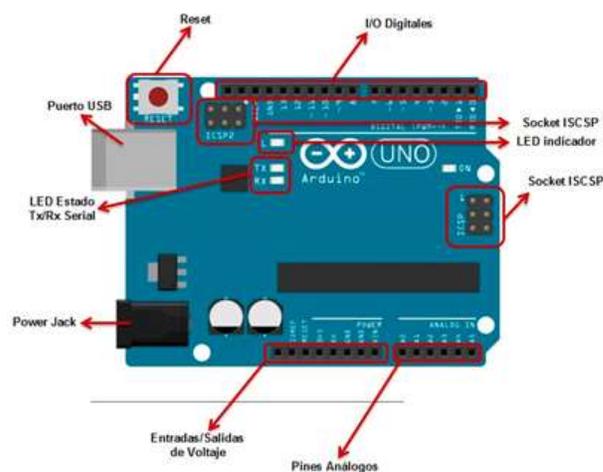
El procesador de datos del sensor es un microcontrolador Arduino Uno. Este se alimenta mediante un puerto USB tipo A, el cual funciona también como

conexión a una computadora para poder subir el código a la memoria del microcontrolador. Las características del microcontrolador son las siguientes:

- Microcontrolador: ATmega328
- Voltaje operativo: 5v
- Voltaje de entrada (Recomendado): 7 – 12 v
- Pines de entradas/salidas digitales: 14 (de las cuales 6 son salidas PWM)
- Pines de entradas análogas: 6
- Memoria flash: 32 KB (ATmega328) de los cuales 0,5 KB son usados por *Bootloader*.
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Velocidad del reloj: 16 MHZ.

Las partes del microcontrolador son las siguientes:

Figura 32. **Partes de microcontrolador Arduino UNO**



Fuente: Arduino (2020). *Tutorial Arduino Uno*. Consultado el 10 de noviembre de 2020.

Recuperado de <https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board>.

5.2.3. Sensor de aceleración

El sensor utilizado es el sensor MPU-6050; este es un dispositivo con giroscopio de 6 ejes y un acelerómetro de 3 ejes; tiene una conexión bus I2C dedicado, que acepta directamente entradas de una brújula externa de 3 ejes.

El MPU-6050 cuenta con tres convertidores de comunicación analógica a digital (ADC) de 16 *bytes* para digitalizar las salidas del giroscopio y tres ADC de 16 *bytes* para digitalizar las salidas del acelerómetro. Cuentan con un búfer de 1024 *bytes* en el *chip* el cual ayuda a reducir el consumo de energía del sistema, al permitir que el procesador lea los datos del sensor en grupos y luego ingrese a un modo de bajo consumo a medida que el sensor MPU recopila más datos.

Figura 33. **Sensor MPU-6050**



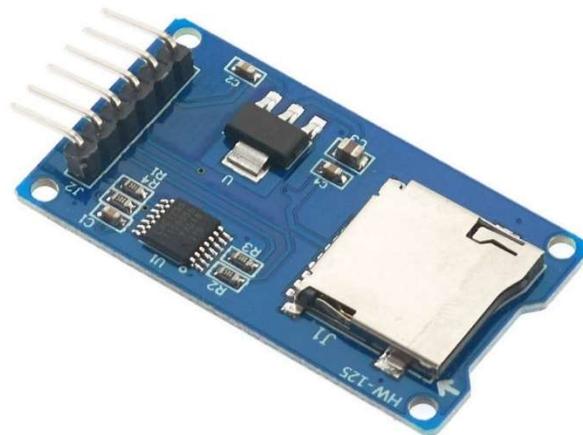
Fuente: Naylam Mechatronics (2016). *Acelerómetro y giroscopio*. Consultado el 12 de noviembre de 2021. Recuperado de https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html.

5.2.4. Módulo microSD

Para el almacenamiento de registros de aceleración se utiliza un módulo microSD, el cual recibe del microcontrolador los datos de aceleración, que a su vez los recibe del sensor de aceleración. Las características del sensor son las siguientes:

- Voltaje de alimentación: 3,3 V o 5 V
- Regulador de tensión incorporado 3,3 V
- Consumo < 200 mA
- Interfaz: SPI
- Dimensiones: 24 mm x 42 mm x 12 mm

Figura 34. Módulo microSD



Fuente: Naylam Mechatronics (2016). *Acelerómetro y giroscopio*. Consultado el 12 de noviembre de 2021. Recuperado de https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html.

5.2.5. Cables y convertidores de voltaje

El sensor utiliza 20 cables, pueden ser tipo *jumper* para prototipado, con conexión macho-hembra para conectar el Arduino al sensor y al módulo microSD. También pueden utilizarse cables soldados para mayor seguridad. Asimismo, se necesita un cable USB tipo A para conectar el Arduino a la fuente de alimentación o al convertidor de voltaje, utilizando una conexión de pared.

Para realizar la conexión a un tomacorriente de pared se utilizan convertidores de voltaje de USB tipo B que proporcionan 5 voltios y 2 amperios de corriente. Esto garantiza el suministro de energía adecuado para el sensor y todos sus componentes.

Figura 35. **Cable USB tipo A para alimentación**



Fuente: Steren (2021), *Cables para alimentación y convertidores de voltaje*. Consultado el 12 de noviembre de 2021. Recuperado de https://www.steren.com.gt/?gclid=Cj0KCQjwvLOTBhCJARIsACVIdV2ghXfyezTEIsTXHuyOeH7kTiGvJfutQ1EmLN8PZfXNk_b4zEPylkoaAkexEALw_wcB.

Figura 36. **Convertidor de voltaje de 5V y 2A**



Fuente: Steren (2021), *Cables para alimentación y convertidores de voltaje*. Consultado el 12 de noviembre de 2021. Recuperado de https://www.steren.com.gt/?gclid=Cj0KCQjwvLOTBhCJARIsACVldV2ghXfyezTEIsTXHuyOeH7kTiGvJfutQ1EmLN8PZfXNk_b4zEPylkoaAkexEALw_wcB.

5.2.6. Cinta doble cara

Con el objetivo de evitar daños en la estructura existente y lograr una fijación lo suficientemente fuerte ante cualquier vibración, se utiliza cinta doble cara de tipo industrial, la cual se coloca en la parte posterior del sensor fijándolo directamente en la pared o viga del edificio. Las especificaciones técnicas deben ser las siguientes:

- Adhesión de pelado a 90 ° 19 lb/pulg
- Tracción normal 720 kPa (105 libras/pulg²)
- Tracción normal 105 lb/pulg²
- Resistencia al cizallamiento de superposición 570 kPa (80 libras/pulg²)
- Resistencia al cizallamiento de superposición 80 lb /pulg²
- Resistencia a la temperatura a corto plazo 230 °C (450 °F)
- Temperatura mínima de aplicación 10 °C (50 °F)

5.2.7. Memoria microSD

Se recomienda utilizar memorias microSD de 16 GB de almacenamiento para cada sensor, lo cual equivale a 1 año de almacenamiento de datos del sensor. Es necesario descargar la información a una computadora y volver a colocar la memoria en el sensor para poder seguir registrando datos. También es posible utilizar memorias de menor o mayor capacidad, aumentando o disminuyendo, respectivamente, la capacidad de almacenamiento del sensor.

Figura 37. Memoria microSD de 16 GB y adaptador SD

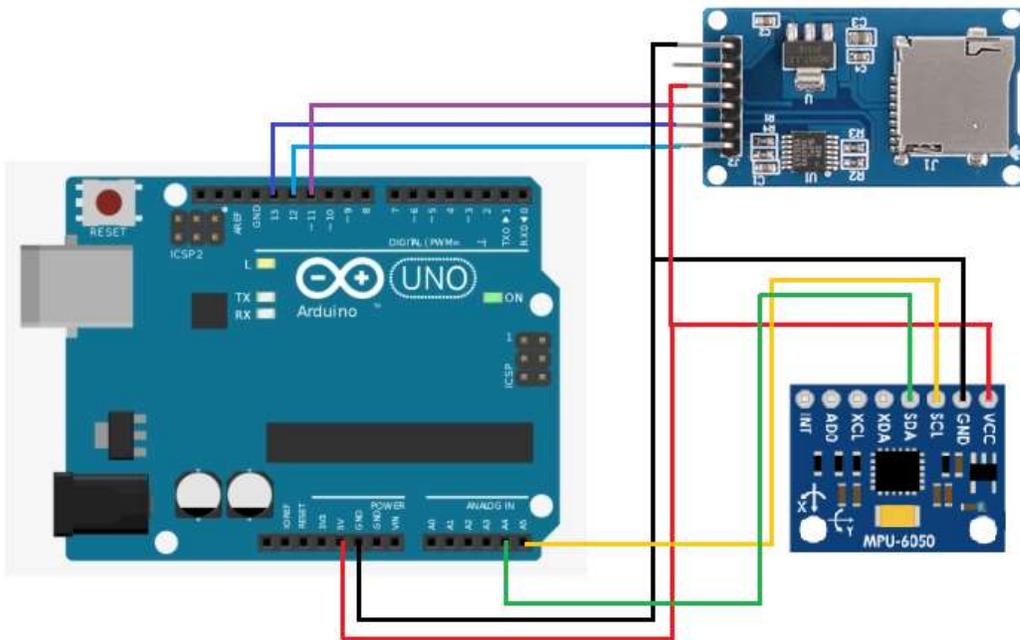


Fuente: Steren (2021), *Cables para alimentación y convertidores de voltaje*. Consultado el 12 de noviembre de 2021. Recuperado de https://www.steren.com.gt/?gclid=Cj0KCQjwvLOTBhCJARIsACVIdV2ghXfyezTEIsTXHuyOeH7kTiGvJfutQ1EmLN8PZfXNk_b4zEPylkoaAkexEALw_wcB.

5.3. Ensamblaje del sensor

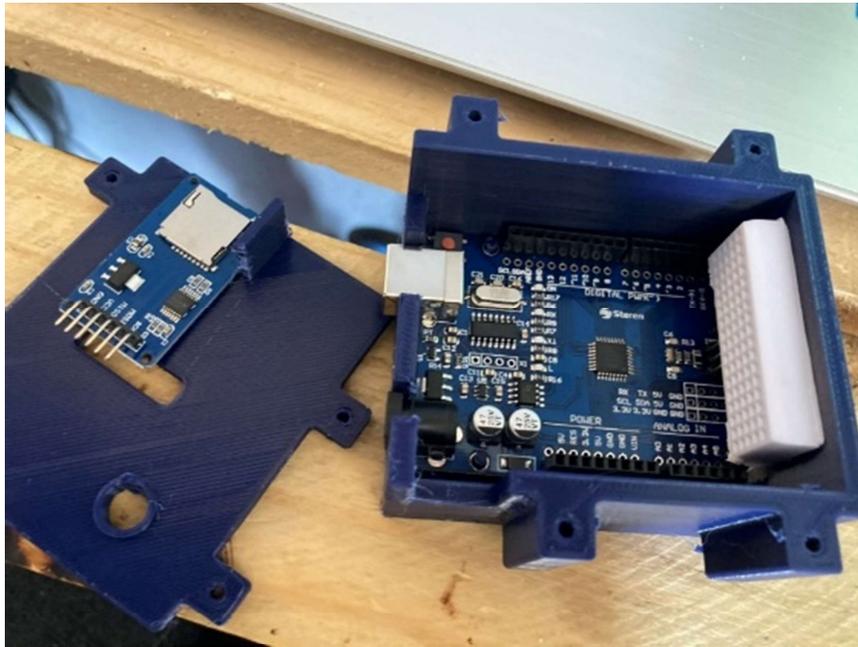
Para que se pueda cargar el código al sensor es necesario conectar correctamente los componentes. En el siguiente diagrama se muestran las conexiones necesarias. Ambos componentes (sensor y módulo microSD) deben conectarse a la salida de 5V del Arduino. Se debe verificar la orientación de los ejes del sensor de aceleración y que quede correctamente alineado a la caja para que se pueda nivelar correctamente.

Figura 38. Diagrama de conexiones de sensor



Fuente: [Fotografía de Juan Orozco]. (Ciudad de Guatemala, 2021). Colección particular.
Guatemala.

Figura 39. **Ensamblaje de componentes en caja**

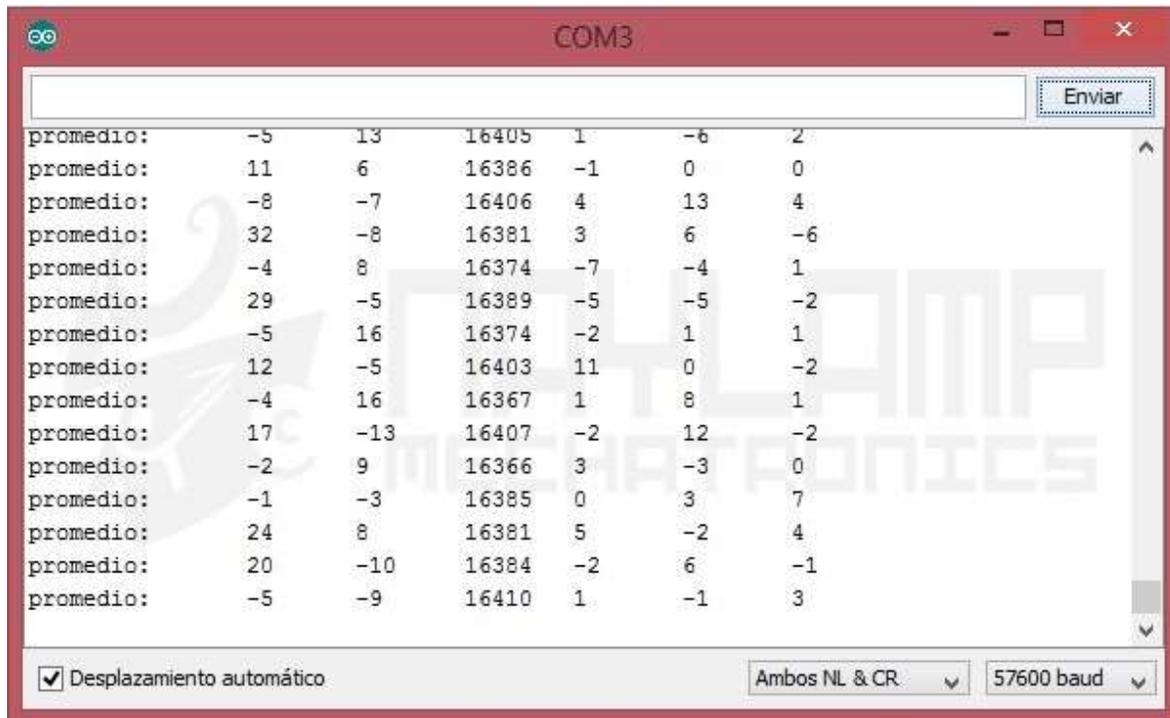


Fuente: [Fotografía de Juan Orozco]. (Ciudad de Guatemala, 2021). Colección particular.
Guatemala.

5.4. Calibración de dispositivos

Para calibrar los dispositivos es necesario tener ensamblados todos sus componentes. Luego de atornillar la caja se debe colocar en una superficie plana y conectar el sensor a una computadora. Para subir código al microcontrolador es necesario tener instalado el software de Arduino IDE, el cual es gratuito. Posterior a ello se debe cargar al sensor el código presentado para calibración, realizado por Naylam Mechatronics (2016). Luego se debe abrir el monitor serial y colocar los valores a los que se quiere calibrar el sensor; en este caso $a_x = 0$, $a_y = 0$, $a_z = 1$ g, $g_x = 0$, $g_y = 0$ y $g_z = 0$.

Figura 40. Monitor serial de calibración del sensor



Fuente: Naylam Mechatronics (2016). *Acelerómetro y giroscopio*. Consultado el 12 de noviembre de 2021. Recuperado de https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html.

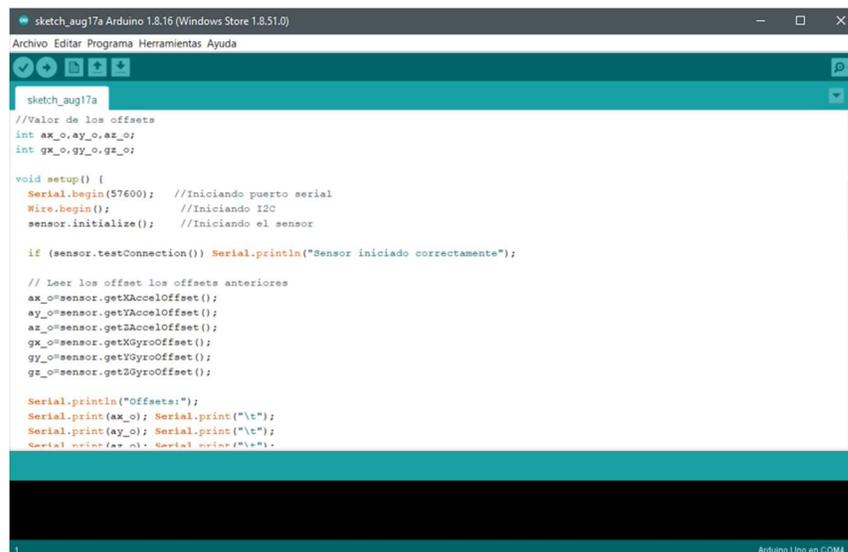
En el monitor serial se mostrará cuando los valores fluctúen cerca de los valores deseados; cuando esto suceda el sensor estará calibrado; si se quisiera verificar la calibración, se debe reiniciar el dispositivo y volver a correr el código de calibración y verificar los valores iniciales.

5.5. Carga de código a microcontrolador

El procedimiento de carga del código al sensor se realiza de manera similar al de calibración. Se debe tener el código en el software Arduino IDE.

Posteriormente se debe seleccionar la opción de compilar y subir. Es importante tener seleccionado el puerto USB en el que está conectado el Arduino.

Figura 41. **Código cargado en software Arduino IDE para ser cargado al sensor**



```
sketch_aug17a
//Valor de los offsets
int ax_o,ay_o,az_o;
int gx_o,gy_o,gz_o;

void setup() {
  Serial.begin(57600); //Iniciando puerto serial
  Wire.begin(); //Iniciando I2C
  sensor.initialize(); //Iniciando el sensor

  if (sensor.testConnection()) Serial.println("Sensor iniciado correctamente");

  // Leer los offset los offsets anteriores
  ax_o=sensor.getXAceelOffset();
  ay_o=sensor.getYAceelOffset();
  az_o=sensor.getZAccelOffset();
  gx_o=sensor.getXGyroOffset();
  gy_o=sensor.getYGyroOffset();
  gz_o=sensor.getZGyroOffset();

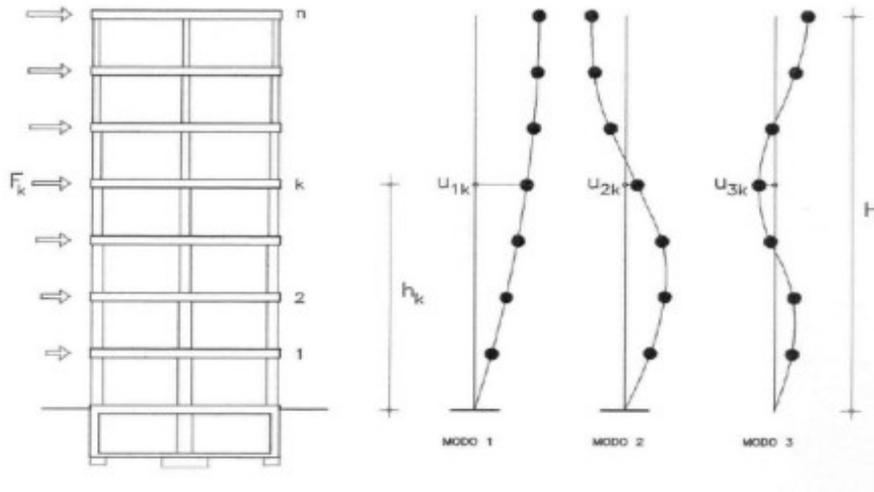
  Serial.println("Offsets:");
  Serial.print(ax_o); Serial.print("\t");
  Serial.print(ay_o); Serial.print("\t");
  Serial.print(gx_o); Serial.print("\t");
  Serial.print(gy_o); Serial.print("\t");
  Serial.print(gz_o); Serial.print("\t");
}
```

Fuente: elaboración propia, empleando software Arduino IDE.

5.6. Diseño de arreglo de sensores en un edificio

Para colocar el arreglo de sensores se utilizará la idealización general de considerar un edificio como un péndulo invertido, en el que la masa de cada nivel se concentra en las losas de cada nivel. Por lo cual es conveniente colocar un sensor en cada nivel para inferir los cambios en las aceleraciones y determinar modos de vibración.

Figura 42. **Idealización de concentraciones de masas por nivel utilizada**



Fuente; elaboración propia, empleando AutoCAD.

5.7. Colocación de sensores en estructura

Para el anclaje a elementos estructurales se debe usar cinta doble cara de tipo industrial para garantizar la suficiente fuerza de adherencia a la estructura, manteniendo al sensor fijo durante vibraciones. Previo a conectar el dispositivo es necesario tomar nota de la hora exacta de inicio, para que posteriormente se puedan sincronizar los sensores en el software de salud estructural.

Figura 43. **Colocación de sensores en estructura**



Fuente: [Fotografía de Juan Orozco]. (Ciudad de Guatemala, 2021). Colección particular.
Guatemala.

6. PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

Los resultados obtenidos de los sensores de monitoreo de salud estructural son los siguientes:

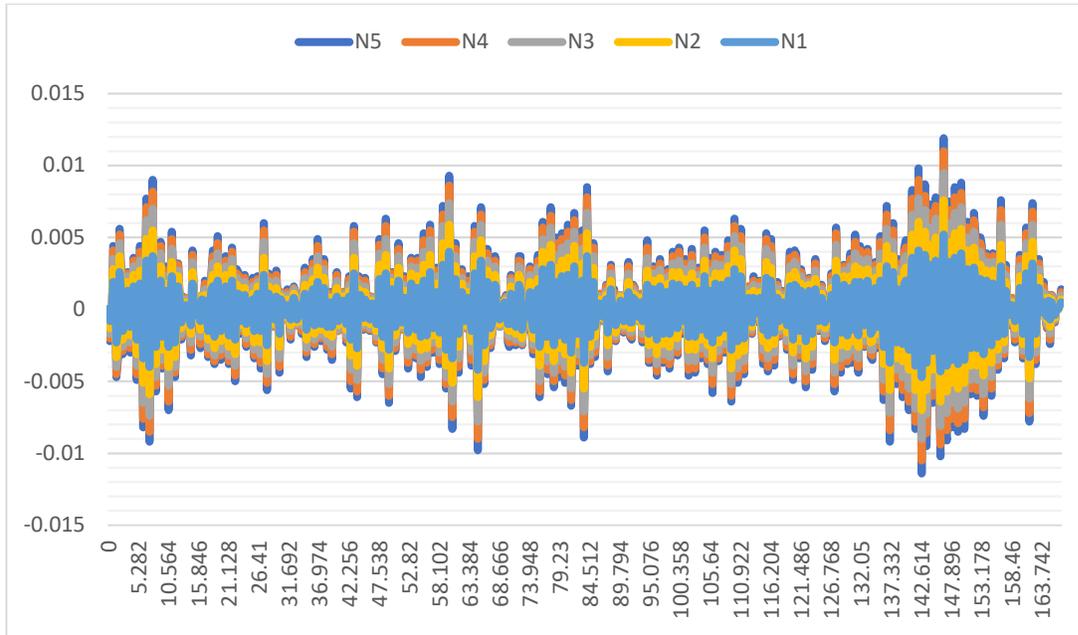
6.1. Análisis de resultados obtenidos

Se evaluaron los resultados de aceleraciones del análisis tiempo-historia lineal al modelo de ETABS del edificio y se compararon con los resultados de aceleración obtenidos de los sensores.

6.1.1. Tiempo historia lineal

Se realizó un análisis de tiempo historia lineal en el programa ETABS. Se graficaron las aceleraciones en un mismo gráfico donde se observa que las aceleraciones aumentan conforme al nivel, pero mantienen la frecuencia de vibración.

Figura 44. **Aceleraciones por piso de análisis tiempo-historia**

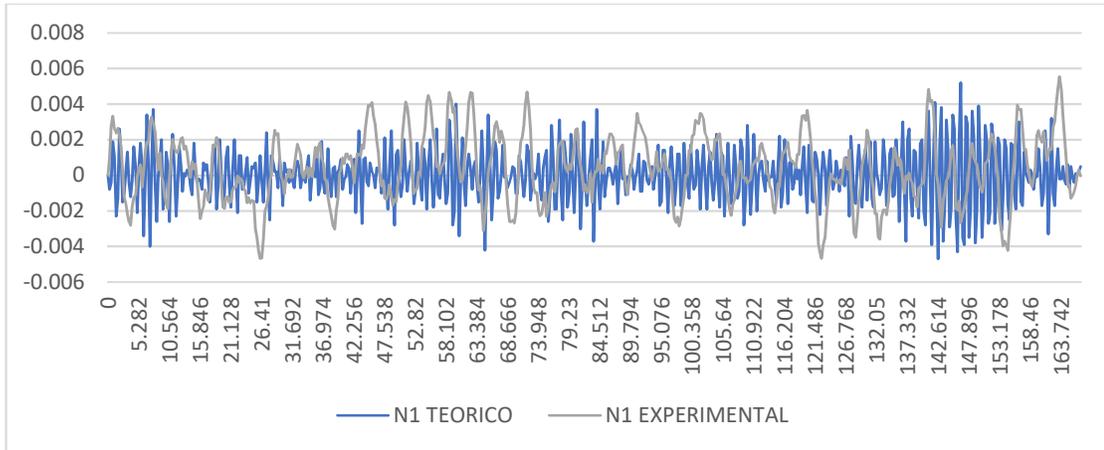


Fuente: elaboración propia, empleando Excel.

6.1.2. **Comparación de aceleraciones teóricas y experimentales por nivel**

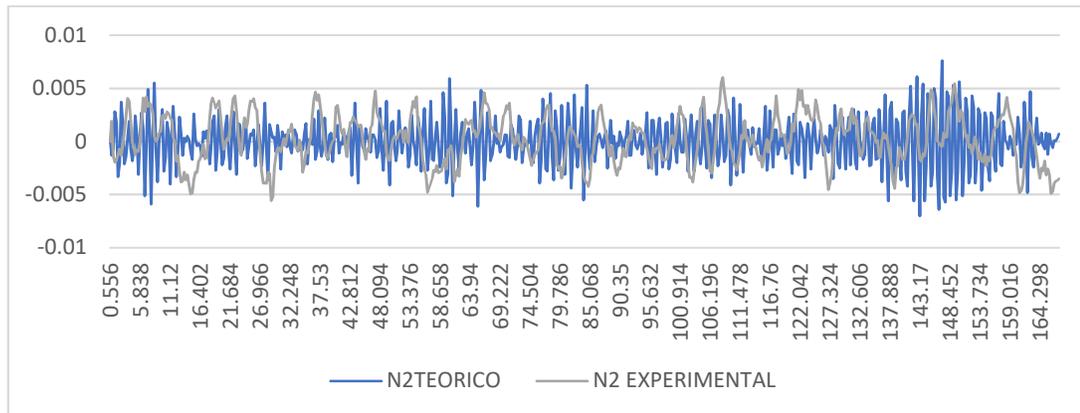
Para poder comparar gráficamente los resultados de los periodos y comprobar que coincidan en su mayoría los resultados de los sensores, a continuación, se muestran en una misma figura las aceleraciones teóricas y experimentales:

Figura 45. **Aceleraciones nivel 1**



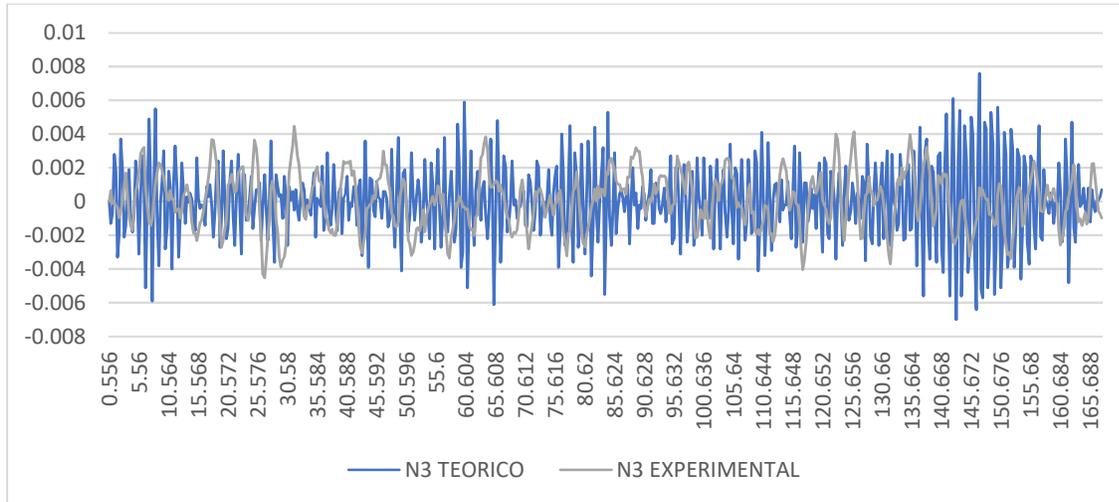
Fuente: elaboración propia, empleando Excel.

Figura 46. **Aceleraciones nivel 2**



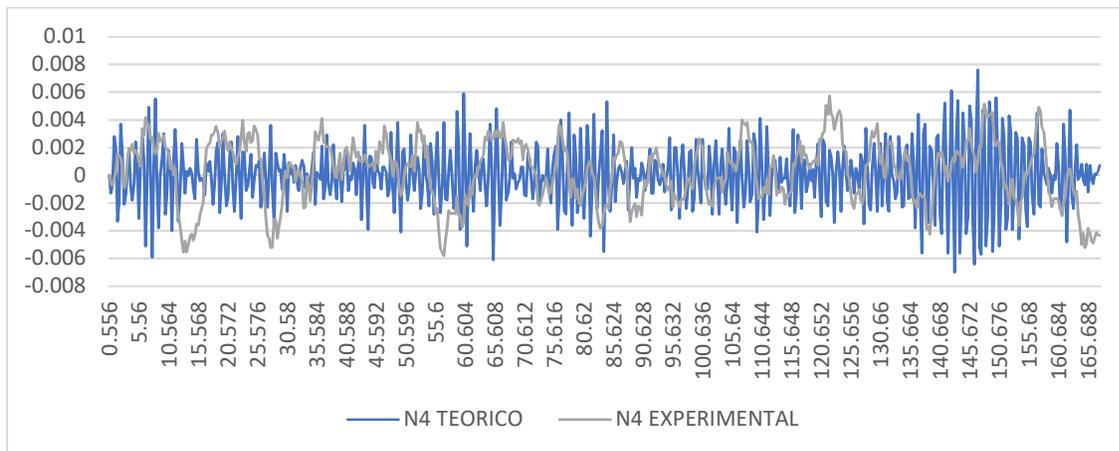
Fuente: elaboración propia, empleando Excel.

Figura 47. **Aceleraciones nivel 3**



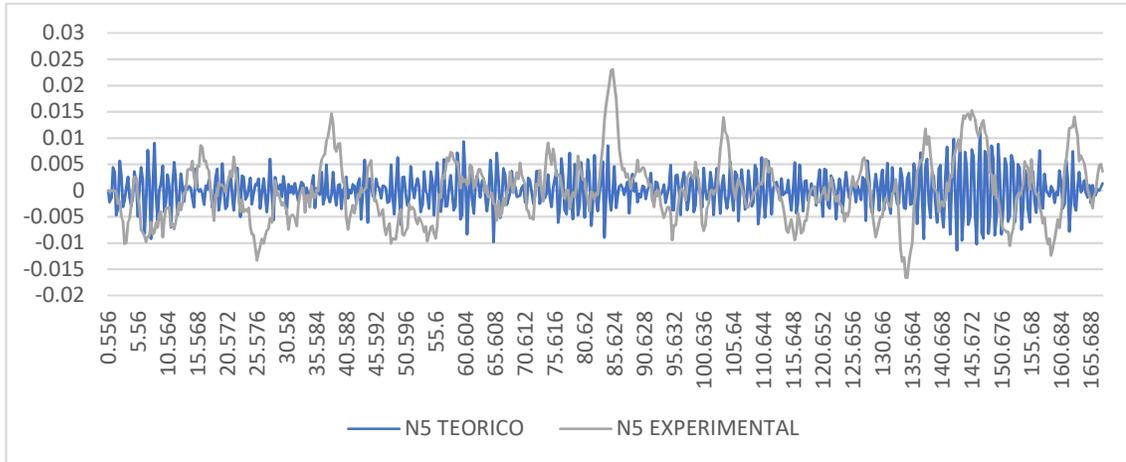
Fuente: elaboración propia, empleando Excel.

Figura 48. **Aceleraciones nivel 4**



Fuente: elaboración propia.

Figura 49 **Aceleraciones nivel 5**



Fuente: elaboración propia, empleando Excel.

Luego de calcular los periodos y frecuencias de vibración, y comparar gráficamente las aceleraciones, se observó que estas coincidían en su mayoría. Sin embargo, debido a un error instrumental en el nivel 3, no se lograron datos coherentes, por lo cual se descartarán en el análisis de resultados.

Tabla XVII. **Comparación numérica de datos teóricos y experimentales**

Nivel	Aceleración máxima (m/s ²)		Aceleración mínima (m/s ²)		Rango (m/s ²)		Aceleración promedio (m/s ²)		Porcentaje de variación
	Teórica	Experimental	Teórica	Experimental	Teórica	Experimental	Teórica	Experimental	
1	0.005	0.005	-0.004	-0.004	0.009	0.010	0.001	0.001	27
2	0.007	0.006	-0.007	-0.005	0.014	0.011	0.001	0.001	5
3	0.009	0.004	-0.009	-0.004	0.018	0.009	0.002	0.001	40
4	0.011	0.005	-0.010	-0.005	0.021	0.011	0.002	0.002	12
5	0.011	0.023	-0.011	-0.016	0.023	0.039	0.002	0.002	23

Fuente: elaboración propia.

6.1.3. Comparación de periodos de vibración e inferencia de daños

Considerando que en promedio el periodo de vibración varía en el rango de 12.75 a 52.25 %, y tomando en cuenta variaciones debido a errores instrumentales, cargas no consideradas o magnitudes diferentes a las reales y otras variables que afecten el periodo teórico del edificio como propiedades dinámicas del suelo distintas y restricciones de suelo diferentes a las modeladas, se considera aceptable la variación de los periodos como primer acercamiento al monitoreo de salud estructural.

Resulta necesario contar con monitoreo de salud estructural desde el inicio de ocupación para poder realizar una calibración del modelo analítico asumiendo un nivel de agrietamiento y comportamiento plástico casi nulo de la estructura, esto brindaría datos más exactos para la comparación de los periodos a lo largo de su vida útil. Es posible monitorear un edificio que tiene varios años de ocupación, sin embargo, se compararía con un estado en el que posiblemente ya cuente con comportamiento plástico de algunos elementos, debiendo calibrar el modelo analítico con el supuesto de tener una estructura sana o asumir algún nivel de agrietamiento y comportamiento plástico al comienzo del monitoreo.

6.2. Identificación de puntos problemáticos

Durante el desarrollo y prueba de los sensores se encontraron los siguientes puntos importantes por considerar en su utilización:

6.2.1. Calibración de sensores

Se debe verificar la calibración de los sensores entre sí, para comprobar que con vibraciones iguales los sensores midan datos iguales o muy parecidos. Esto para evitar problemas de fallos instrumentales en niveles aislados que puedan afectar el resultado final.

6.2.2. Consideraciones de estructura

Para utilizar el sistema de monitoreo de salud estructural en edificios es necesario conocer con exactitud las dimensiones de los elementos estructurales, armado y detallado, así como las características geotécnicas del suelo. Variaciones en dichos parámetros tienen una incidencia significativa en los resultados obtenidos o en el comportamiento estructural.

6.2.3. Fallas en fuentes de poder

Debido al gran tiempo que pasan los sensores conectados a energía eléctrica, se observó que existen posibilidades de que se reinicien las lecturas de aceleración. Esto hace que se pierda el control de tiempo que se lleva internamente. Entonces es conveniente generar archivos fraccionados para tener facilidad de encontrar fechas.

6.2.4. Sismos de baja magnitud

Para tener datos de aceleración con suficiente intensidad es necesario que el movimiento sísmico de estudio sea de una magnitud detectable por los sensores; esto implica una combinación de magnitud y localización del epicentro e hipocentro. Debido a la falta de capacidad predictiva para sismos que se tiene

en la actualidad, no es posible conocer cuánto tiempo se requerirá mantener el monitoreo estructural y tener datos suficientes para realizar inferencias de daños.

CONCLUSIONES

1. Es posible fabricar un dispositivo electrónico con capacidad de medir datos de aceleraciones estructurales con partes disponibles en Guatemala. Utilizando microcontroladores Arduino Uno y sensores de aceleración MPU 6050, se logran leer datos de aceleraciones suficientes durante sismos de magnitud intermedia, para poder encontrar periodos y frecuencias de vibración.
2. El monitoreo a largo plazo (mayor a 5 meses) sirvió para verificar los cambios en periodos de vibración debido a sismos. Esto permitirá inferir el grado de degradación de los elementos. Se observó que, al perder rigidez en elementos, modelados como reducción de momento de inercia, se puede inferir el grado de degradación de la estructura.
3. Para poder inferir daños en la estructura mediante el porcentaje de degradación de los elementos, fue necesario contar con datos de aceleraciones y periodos de vibración de la estructura relativamente sana al inicio de la ocupación, para que sirviera de calibración para el modelo analítico y así comparar con mediciones posteriores durante la vida útil de la estructura.
4. Es posible realizar la extracción de parámetros estructurales tales como periodos, frecuencias y modos de vibración, mediante el uso de software computacional construido en esta investigación, por medio de datos de aceleraciones registrados mediante el monitoreo estructural de la estructura

en cierto periodo de tiempo, para inferir daños en el sistema estructural del edificio.

RECOMENDACIONES

1. Incentivar el monitoreo de salud estructural a edificios para poder prevenir fallas imprevistas y realizar reparaciones o reforzamiento cuando sea necesario.
2. Calibrar correctamente los dispositivos previos a la colocación para evitar resultados erróneos al tomar mediciones e inferir daños estructurales en la estructura.
3. Monitorear constantemente el estado de los dispositivos en funcionamiento, para realizar reparaciones o cambios cuando sea necesario.
4. Colocar reguladores de voltaje en las conexiones de corriente de los sensores, para prevenir lecturas erróneas debido a subidas y bajadas de voltaje o interrupciones al suministro de energía eléctrica.
5. Realizar actualizaciones del programa de monitoreo de salud estructural para mejorar fallas encontradas y optimizar la experiencia de usuario.

REFERENCIAS

1. Alonso Alonso, R. (2016). *Cálculo sísmico de estructuras convencionales de edificación aplicando métodos lineales y no lineales implementados en diferentes normas. Estudio comparativo*. Madrid: (Tesis de maestría). Universidad Politécnica de Madrid, España. Recuperado de https://oa.upm.es/44488/1/Trabajo%20Fin%20Master_Ruben%20Alonso_Ingenieria%20Sismica.pdf.
2. Actis, R. L., y Dimarogonas, A. D. (2000). Non-linear effects due to closing cracks in vibrating beams. *ASME Design Engineering Division Publication DE-Structural Vibrations and Acoustics*, 99-104. Michigan: Los Álamos, Recuperado de <https://www.sciencedirect.com/science/article/abs/pii/S0022460X00930992>.
3. AGIES (2018). *Normas de seguridad estructural*. Guatemala AGIES. Recuperado de <https://www.agies.org/bibliotecas/->.
4. Agnihotri, N. (2020). *Engineers Garage*. USA: Recuperado de <https://www.engineersgarage.com/wp-content/uploads/2020/09/19AR-02.png>.
5. Ajay, R. (2007). *Guided-Wave Structural Health Monitoring*. Michigan: The University of Michigan. Recuperado de https://deepblue.lib.umich.edu/bitstream/handle/2027.42/77498/Raghavan_PhD_thesis_GWSHM.pdf.

6. American Concrete Institute. (2014). *Requisitos de reglamento para concreto estructural (ACI 318-14)*. Michigan, USA: ACI. Recuperado de https://www.oaxaca.gob.mx/sinfra/wp-content/uploads/sites/14/2016/02/ACI_318_2005.pdf.
7. Analog Devices. (2010). *Small, Low Power, 3-Axis + 3g Accelerometer ADXL335*. USA: One Technology Way. Recuperado de <https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>.
8. Arduino (2020). *ArduinoStore*. España: Portenta Max Carrier. Recuperado de <https://store.arduino.cc/arduino-uno-rev3>.
9. Bertedero. (1995). *Vision 200 SEAOC*. USA: Researchgate. Recuperado de https://www.researchgate.net/figure/Vision-2000-conceptual-performance-objectives-matrix-SEAOC-1995_fig1_329613989.
10. Bishop, C. M. (1994). Neural networks and their applications. *Review of Scientific Instrumentation*, pp. 1803-1832. Universidad Autónoma del Estado de Morelos. Recuperado de https://www.researchgate.net/publication/270756365_Use_of_neural_networks_and_neural_network_inverse_in_optimizing_food_processes.
11. Chougle, P. D., Kirkegaard, P. H., y Nielsen, S. R. (s.f.). *Low cost wireless sensor network for structural health monitoring*. Denmark: Department of Civil Engineering, Aalborg University. Recuperado de <https://journals.sagepub.com/doi/abs/10.1177/1475921719854528>.
12. Chowdhury, M. R., y Ramirez, M. (1992). A comparison of the modal responses for defective versus nondefective concrete test beams.

10th International Modal Analysis Conference, pp. 508-515. USA: Universidad de California. Recuperado de https://www.researchgate.net/publication/352408178_Editorial_Natural_Hazards_Engineering_Research_Infrastructure_NHERI_Mitigating_the_Impact_of_Natural_Hazards_on_Civil_Infrastructure_and_Communities.

13. Doebling, S. W., Farrar, C. R., Prime, M. B., y Shevitz, D. W. (1996). *Damage Identification and Health Monitoring of Structural and Mechanical Systems from changes in their vibration characteristics*. Washington: USDOE. Recuperado de <https://www.osti.gov/biblio/249299-damage-identification-health-monitoring-structural-mechanical-systems-from-changes-vibration-characteristics-literature-review>.
14. Dziauddin, R. A., y Ahmad, N. A. (2015). *A comparative study of wireless accelerometer sensors for water pipeline leakage*. Kuala Lumpur, Malasia: Universiti Teknologi Malaysia.
15. Farrar, C. R., y Worden, K. (2006). An introduction to structural health monitoring. *Philosophical transactions of The Royal Society*, pp. 303-315. Universidad de Sheffield. Recuperado de <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2006.1928>.
16. Friswell, M. I., Penny, J. T., y Wilson, D. A. (1994). Using vibration data and statistical measures to locate damage in structures. *Modal Analysis: The International Journal of Analytical and Experimental Modal Analysis*, pp. 239-254. Ashrafi Esfahani Hwy, Tehran, Iran. Recuperado de <https://www.sid.ir/en/journal/ViewPaper.aspx?ID=67795>.

17. González Cuevas, O. (2005). *Aspectos fundamentales de concreto reforzado*. México: Editorial Limusa.
18. Gudmonson, P. (1983). The dynamic behavior of slender structures with cross-sectional cracks. *Journal of the Mechanics and Physics of Solids*, pp. 329-345. USA: Sciencedirect. Recuperado de <https://www.sciencedirect.com/science/article/abs/pii/0022509683900030>.
19. InverSense. (2013). *MPU-6000/MPU-6050 Register map and descriptions*. Estados Unidos. Technology Drive. Recuperado de <https://stanford.edu/class/ee267/misc/MPU-9255-Register-Map.pdf>.
20. Kang, I., Schulz, M., Kim, J., Vesslin, S., y Shi, D. (2006). *A carbon nanotube strain sensor for structural health monitoring*. Cincinnati: University of Cincinnati. Recuperado de https://www.researchgate.net/publication/231050709_A_Carbon_Nanotube_Strain_Sensor_for_Structural_Health_Monitoring.
21. Labdad, F. M. (2012). Consideraciones sobre la elección de coeficientes de balasto. *Revista de obras públicas*. Argentina: Universidad Tecnológica Nacional. Recuperado de <https://www.studocu.com/es-ar/document/universidad-tecnologica-nacional/cimentaciones/coeficiente-balasto-apunte/16965894>.
22. Lin, R. M., y Ewins, D. J. (1990). *On the location of structural nonlinearity from modal testing - a feasibility study*. USA: National Center For Biotechnology Information. Recuperado de <https://www.academia>.

edu/52293797/Modal_Identification_of_Non_Linear_Structures_and
_the_Use_of_Modal_Model_in_Structural_Dynamic_Analysis.

23. McCormac, J. C. (2011). *Diseño de concreto reforzado*. New Jersey, USA: ALFAOMEGA.
24. Meli Piralla , R. (1985). *Diseño estructural*. Mexico: Editorial Limusa.
25. Melone, S. (2019). *Desempeño sísmico de edificaciones* Universitat Politècnica de Catalunya. Recuperado de https://aulaseproinca.blogspot.com/2019/02/desempeno-sismico-de-edificaciones.html?fbclid=IwAR16jSqF3j5YEt3f65--FIVBMbpOi_uah5ugCmkSe5Sgt0bUH8wTr4v8kTY.
26. Naylam Mechatronics. (2016). *Tutorial de acelerómetro y giroscopio*. Recuperado de https://naylam pmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html.
27. NXP Semiconductors. (2016). *MMA8452Q, 3-axis, 12-bit/8-bit digital accelerometer*. Estados Unidos: XP.
28. Open Circuit. (2020). *Desbloqueo del acelerómetro*. España: OC. Recuperado de <https://opencircuit.es/Producto/Desbloqueo-del-aceler%C3%B3metro-eje-triple>.
29. Pandey, A., Biswas, M., y Samman, M. (1991). Damage detection from changes in curvature mode shapes. *Journal of Sound and Vibration*, pp.321-332. Universidad de Harvard. Recuperado de <https://ui.adsabs.harvard.edu/abs/1991JSV...145..321P/abstract>.

30. Song, G., Gu, H., Mo, Hsu, T., y Dhonde, H. (2007). Concrete structural health monitoring using embedded piezoceramic trasducers. *Smart Materials and Structures*, pp.959-968. Mountain State University. Recuperado de https://www.researchgate.net/publication/228853131_Concrete_structural_health_monitoring_using_embedded_piezoceramic_transducers.
31. USGS. (2021). *earthquake.usgs.gov*. USA: Geological Survey. Recuperado de <https://earthquake.usgs.gov/earthquakes/search/>.
32. Varanis, M., Langone Silva, A., y Guilherme Mere, A. (2017). On mechanical vibration analysis of multi degree of freedom system based on arduino MEMS accelerometers. *Revista Brasileira de Ensino de Física Vol.40*, p. 1304. Recuperado de <https://www.scielo.br/j/rbef/i/2018.v40n4/>.
33. Yuen, M. M. (1985). A numerical study of the eigenparameters of a Damaged cantilever. *Journal of Sound and Vibration*, pp. 301-310. USA: Researchgate.net. Recuperado de https://www.researchgate.net/publication/292132620_A_numerical_study_of_structural_damage_detection_using_changes_in_the_rotation_of_mode_shapes.

APÉNDICES

Apéndice 1. Aceleraciones de sismos de estudio

Tiempo en segundos	N5	N4	N3	N2	N1
0	0	0	0	0	0
0.278	-0.0022	-0.002	-0.0017	-0.0013	-0.0008
0.556	-0.0015	-0.0013	-0.001	-0.0007	-0.0004
0.834	0.0044	0.0041	0.0035	0.0028	0.0019
1.112	0.0036	0.0031	0.0025	0.0018	0.0011
1.39	-0.0047	-0.0045	-0.004	-0.0033	-0.0023
1.668	-0.0034	-0.0031	-0.0026	-0.002	-0.0013
1.946	0.0056	0.0052	0.0046	0.0037	0.0026
2.224	0.0027	0.0025	0.0021	0.0017	0.0012
2.502	-0.0031	-0.0029	-0.0026	-0.0021	-0.0015
2.78	-0.0019	-0.0018	-0.0017	-0.0014	-0.001
3.058	0.0003	0.0003	0.0003	0.0003	0.0002
3.336	0.0026	0.0025	0.0023	0.0019	0.0013
3.614	-0.0013	-0.0012	-0.001	-0.0007	-0.0004
3.892	-0.003	-0.0027	-0.0023	-0.0018	-0.0012
4.17	-0.0012	-0.001	-0.0008	-0.0006	-0.0003
4.448	0.0036	0.0034	0.003	0.0024	0.0016
4.726	0.0012	0.001	0.0008	0.0006	0.0004
5.004	-0.0049	-0.0045	-0.0039	-0.0031	-0.0021
5.282	0.0009	0.0008	0.0007	0.0006	0.0004
5.56	0.0044	0.004	0.0034	0.0027	0.0018
5.838	0.0021	0.0019	0.0015	0.0011	0.0006
6.116	-0.0082	-0.0076	-0.0065	-0.0051	-0.0034
6.394	-0.0014	-0.0012	-0.0008	-0.0005	-0.0003
6.672	0.0077	0.0071	0.0062	0.0049	0.0034
6.95	0.0019	0.0017	0.0014	0.001	0.0006
7.228	-0.0092	-0.0085	-0.0074	-0.0059	-0.004
7.506	0.0012	0.0011	0.0009	0.0007	0.0004
7.784	0.009	0.0082	0.007	0.0055	0.0037
8.062	-0.0009	-0.0009	-0.0008	-0.0007	-0.0005
8.34	-0.0057	-0.0053	-0.0047	-0.0038	-0.0026
8.618	6.04E-07	-1.67E-05	-2.27E-05	-8.77E-06	1.04E-05
8.896	0.0012	0.0012	0.0012	0.0012	0.0009
9.174	0.0047	0.0044	0.0039	0.003	0.002
9.452	-0.0041	-0.0039	-0.0035	-0.0028	-0.0019
9.73	-0.0021	-0.0019	-0.0017	-0.0015	-0.0011
10.008	0.003	0.0027	0.0023	0.0018	0.0013
10.286	0.0009	0.0009	0.0009	0.0007	0.0005
10.564	-0.007	-0.0064	-0.0053	-0.004	-0.0026
10.842	0.0012	0.0013	0.0012	0.0011	0.0008
11.12	0.0054	0.0049	0.0042	0.0033	0.0023
11.398	0.0019	0.0016	0.0012	0.0007	0.0003
11.676	-0.0047	-0.0044	-0.004	-0.0033	-0.0023
11.954	-0.0015	-0.0014	-0.0012	-0.0009	-0.0006
12.232	0.0032	0.0031	0.0027	0.0023	0.0016

Continuación del apéndice 1.

12.51	0.0007	0.0007	0.0007	0.0007	0.0006
12.788	-0.0021	-0.0019	-0.0017	-0.0013	-0.0009
13.066	0.0009	0.0007	0.0005	0.0003	0.0001
13.344	0.0002	0.0001	2.24E-05	-2.50E-05	-3.63E-05
13.622	0.0008	0.0008	0.0007	0.0005	0.0003
13.9	0.0005	0.0004	0.0003	0.0002	0.0002
14.178	-0.0014	-0.0013	-0.0011	-0.0009	-0.0006
14.456	-0.0032	-0.0028	-0.0023	-0.0017	-0.0011
14.734	0.0041	0.0038	0.0033	0.0026	0.0018
15.012	0.0004	0.0003	0.0002	0.0002	0.0001
15.29	-0.0002	-0.0002	-0.0003	-0.0004	-0.0004
15.568	0.0002	0.0001	-0.0001	-0.0002	-0.0002
15.846	-0.0005	-0.0005	-0.0004	-0.0004	-0.0003
16.124	-0.0027	-0.0024	-0.0019	-0.0014	-0.0008
16.402	0.0009	0.0009	0.001	0.0009	0.0007
16.68	0.0002	0.0002	0.0002	0.0003	0.0003
16.958	0.002	0.0018	0.0014	0.001	0.0006
17.236	0.0001	-0.0001	-0.0002	-0.0003	-0.0003
17.514	-0.0033	-0.0031	-0.0027	-0.0021	-0.0015
17.792	-0.0014	-0.0012	-0.0009	-0.0006	-0.0003
18.07	0.0024	0.0023	0.0022	0.0018	0.0014
18.348	0.0041	0.0037	0.0032	0.0024	0.0016
18.626	-0.0038	-0.0036	-0.0032	-0.0027	-0.0019
18.904	-0.0008	-0.0008	-0.0008	-0.0007	-0.0006
19.182	0.0051	0.0046	0.0039	0.003	0.002
19.46	0.0003	0.0003	0.0002	0.0001	0.0001
19.738	-0.0035	-0.0032	-0.0028	-0.0022	-0.0015
20.016	-0.003	-0.0027	-0.0022	-0.0016	-0.001
20.294	0.0014	0.0014	0.0014	0.0013	0.001
20.572	0.0037	0.0034	0.003	0.0024	0.0016
20.85	-0.0009	-0.0009	-0.0009	-0.0008	-0.0006
21.128	-0.0038	-0.0036	-0.0032	-0.0026	-0.0018
21.406	2.57E-05	4.75E-05	0.0001	0.0001	0.0001
21.684	0.0043	0.004	0.0035	0.0028	0.002
21.962	-0.0002	-0.0002	-0.0002	-0.0002	-0.0001
22.24	-0.005	-0.0046	-0.0039	-0.0031	-0.0021
22.518	0.0028	0.0025	0.0022	0.0017	0.0011
22.796	0.0026	0.0024	0.002	0.0016	0.0011
23.074	-0.0017	-0.0015	-0.0013	-0.0011	-0.0008
23.352	-0.0012	-0.0011	-0.001	-0.0008	-0.0006
23.63	0.0005	0.0005	0.0004	0.0003	0.0002
23.908	0.0024	0.0022	0.0019	0.0015	0.001
24.186	-0.0026	-0.0024	-0.002	-0.0016	-0.001
24.464	-0.0006	-0.0005	-0.0004	-0.0004	-0.0003
24.742	0.0012	0.0011	0.0009	0.0007	0.0005
25.02	0.001	0.001	0.0008	0.0006	0.0004
25.298	0.0022	0.0019	0.0016	0.0011	0.0007
25.576	-0.0034	-0.0032	-0.0028	-0.0023	-0.0015
25.854	-0.0007	-0.0006	-0.0005	-0.0004	-0.0003
26.132	0.0023	0.0021	0.0019	0.0016	0.0011
26.41	-0.001	-0.0008	-0.0006	-0.0004	-0.0003
26.688	-0.0041	-0.0037	-0.0031	-0.0023	-0.0015
26.966	-0.0001	-4.04E-05	0.0001	0.0001	0.0002
27.244	0.006	0.0055	0.0047	0.0036	0.0024
27.522	-0.0008	-0.0009	-0.0009	-0.0008	-0.0006

Continuación del apéndice 1.

27.8	-0.0056	-0.0051	-0.0045	-0.0036	-0.0025
28.078	0.0025	0.0023	0.002	0.0016	0.0011
28.356	0.001	0.001	0.001	0.0009	0.0007
28.634	0.0003	0.0003	0.0003	0.0003	0.0002
28.912	0.0011	0.0009	0.0007	0.0004	0.0002
29.19	-0.0012	-0.0012	-0.0011	-0.001	-0.0007
29.468	0.0027	0.0024	0.002	0.0015	0.0009
29.746	0.0005	0.0004	0.0003	0.0002	0.0001
30.024	-0.0044	-0.004	-0.0034	-0.0026	-0.0017
30.302	0.0012	0.0012	0.0011	0.0009	0.0007
30.58	-0.0004	-0.0003	-0.0001	0.0001	0.0002
30.858	0.0008	0.0008	0.0007	0.0006	0.0003
31.136	-0.0006	-0.0007	-0.0006	-0.0005	-0.0004
31.414	0.0014	0.0012	0.001	0.0007	0.0003
31.692	-0.0006	-0.0006	-0.0006	-0.0005	-0.0003
31.97	-0.0021	-0.0019	-0.0015	-0.0011	-0.0007
32.248	0.0004	0.0004	0.0005	0.0005	0.0004
32.526	0.0016	0.0015	0.0014	0.0011	0.0008
32.804	0.0013	0.0012	0.001	0.0007	0.0004
33.082	-0.0011	-0.0011	-0.001	-0.0009	-0.0007
33.36	0.0005	0.0004	0.0002	0.0001	-1.33E-05
33.638	-0.0005	-0.0005	-0.0004	-0.0003	-0.0002
33.916	-0.0017	-0.0015	-0.0011	-0.0008	-0.0004
34.194	0.0007	0.0007	0.0007	0.0007	0.0005
34.472	0.0029	0.0026	0.0022	0.0017	0.0011
34.75	-0.0033	-0.003	-0.0027	-0.0021	-0.0014
35.028	0.0004	0.0004	0.0003	0.0002	3.50E-05
35.306	-0.0001	-0.0001	-0.0001	-9.00E-06	0.0001
35.584	-0.0008	-0.0006	-0.0004	-0.0003	-0.0001
35.862	0.0035	0.0032	0.0027	0.0021	0.0014
36.14	-0.0026	-0.0024	-0.0021	-0.0017	-0.0011
36.418	-0.0015	-0.0013	-0.0012	-0.0009	-0.0007
36.696	0.0049	0.0044	0.0037	0.0029	0.0019
36.974	-0.0009	-0.0008	-0.0008	-0.0006	-0.0004
37.252	-0.0022	-0.002	-0.0018	-0.0014	-0.001
37.53	-0.0007	-0.0007	-0.0005	-0.0004	-0.0002
37.808	0.0035	0.0032	0.0028	0.0022	0.0015
38.086	-0.0006	-0.0006	-0.0005	-0.0004	-0.0003
38.364	-0.0028	-0.0026	-0.0022	-0.0017	-0.0012
38.642	0.001	0.0009	0.0008	0.0006	0.0004
38.92	0.0012	0.0011	0.0009	0.0008	0.0005
39.198	-0.0035	-0.0031	-0.0026	-0.0019	-0.0012
39.476	0.0001	0.0002	0.0002	0.0002	0.0002
39.754	0.0007	0.0006	0.0006	0.0005	0.0004
40.032	0.0026	0.0024	0.002	0.0015	0.0009
40.31	-0.0015	-0.0015	-0.0014	-0.0011	-0.0008
40.588	0.0001	0.0001	-1.44E-05	-0.0001	-0.0002
40.866	-0.0005	-0.0005	-0.0004	-0.0003	-0.0001
41.144	0.0012	0.0012	0.0011	0.0009	0.0006
41.422	0.0009	0.0008	0.0007	0.0005	0.0004
41.7	-0.0023	-0.0021	-0.0018	-0.0014	-0.001
41.978	0.0015	0.0013	0.0011	0.0008	0.0005
42.256	0.0023	0.0021	0.0017	0.0013	0.0009
42.534	-0.0055	-0.005	-0.0042	-0.0032	-0.0021
42.812	-0.0007	-0.0005	-0.0003	-0.0001	-4.92E-05

Continuación del apéndice 1.

43.09	0.0058	0.0054	0.0046	0.0036	0.0025
43.368	4.52E-05	-2.14E-05	-0.0001	-0.0002	-0.0002
43.646	-0.0061	-0.0057	-0.0049	-0.0039	-0.0027
43.924	0.0023	0.0021	0.0018	0.0014	0.0009
44.202	0.0019	0.0018	0.0016	0.0013	0.001
44.48	-0.0011	-0.001	-0.0007	-0.0005	-0.0004
44.758	-0.0014	-0.0013	-0.0011	-0.0009	-0.0006
45.036	0.0022	0.002	0.0016	0.0012	0.0007
45.314	0.0008	0.0007	0.0005	0.0003	0.0002
45.592	0.0004	0.0004	0.0003	0.0001	1.96E-05
45.87	-0.0015	-0.0014	-0.0013	-0.001	-0.0007
46.148	0.0009	0.0009	0.0008	0.0006	0.0004
46.426	0.0008	0.0007	0.0007	0.0006	0.0004
46.704	0.0004	0.0004	0.0003	0.0002	0.0001
46.982	-0.0023	-0.0022	-0.0019	-0.0015	-0.001
47.26	-0.002	-0.0017	-0.0014	-0.001	-0.0006
47.538	0.0049	0.0045	0.0039	0.0031	0.0021
47.816	-0.0011	-0.001	-0.0009	-0.0007	-0.0005
48.094	-0.0045	-0.0041	-0.0035	-0.0027	-0.0019
48.372	0.0005	0.0005	0.0005	0.0004	0.0004
48.65	0.0063	0.0058	0.0049	0.0038	0.0025
48.928	-0.0013	-0.0013	-0.0012	-0.001	-0.0007
49.206	-0.0065	-0.006	-0.0052	-0.0041	-0.0028
49.484	0.0025	0.0024	0.0021	0.0017	0.0012
49.762	0.0026	0.0025	0.0022	0.0019	0.0014
50.04	-0.0011	-0.001	-0.0008	-0.0006	-0.0004
50.318	-0.0029	-0.0027	-0.0023	-0.0018	-0.0012
50.596	-0.0014	-0.0013	-0.001	-0.0008	-0.0005
50.874	0.0046	0.0042	0.0037	0.0029	0.002
51.152	0.0008	0.0007	0.0005	0.0004	0.0003
51.43	-0.0014	-0.0014	-0.0013	-0.0011	-0.0009
51.708	0.0009	0.0007	0.0005	0.0003	0.0001
51.986	0.0023	0.0021	0.0017	0.0013	0.0008
52.264	0.0008	0.0007	0.0005	0.0004	0.0002
52.542	-0.0041	-0.0037	-0.0032	-0.0024	-0.0016
52.82	-0.0028	-0.0024	-0.0019	-0.0013	-0.0008
53.098	0.0036	0.0034	0.003	0.0025	0.0018
53.376	-0.0001	-0.0001	-4.09E-06	0.0001	0.0001
53.654	-0.0008	-0.0008	-0.0007	-0.0007	-0.0005
53.932	-0.0034	-0.0032	-0.0028	-0.0022	-0.0014
54.21	0.0036	0.0034	0.0029	0.0023	0.0015
54.488	0.0017	0.0015	0.0013	0.001	0.0007
54.766	-0.0047	-0.0042	-0.0036	-0.0028	-0.0019
55.044	0.0012	0.0011	0.001	0.0008	0.0005
55.322	0.0053	0.0048	0.004	0.0031	0.002
55.6	0.0001	-2.48E-05	-0.0002	-0.0003	-0.0003
55.878	-0.004	-0.0037	-0.0033	-0.0027	-0.0018
56.156	-0.0013	-0.0012	-0.0009	-0.0007	-0.0004
56.434	0.0059	0.0055	0.0048	0.0038	0.0026
56.712	-0.003	-0.0028	-0.0023	-0.0017	-0.0011
56.99	-0.003	-0.0027	-0.0023	-0.0018	-0.0013
57.268	0.0005	0.0004	0.0003	0.0003	0.0003
57.546	0.0029	0.0027	0.0023	0.0018	0.0012
57.824	-0.0004	-0.0004	-0.0003	-0.0003	-0.0002
58.102	-0.0038	-0.0035	-0.003	-0.0024	-0.0016

Continuación del apéndice 1.

58.38	-0.0031	-0.0028	-0.0023	-0.0016	-0.001
58.658	0.0072	0.0066	0.0058	0.0046	0.0031
58.936	0.0043	0.0038	0.0031	0.0023	0.0015
59.214	-0.0055	-0.0052	-0.0047	-0.0039	-0.0028
59.492	-0.0048	-0.0044	-0.0038	-0.003	-0.002
59.77	0.0093	0.0086	0.0074	0.0059	0.004
60.048	0.0015	0.0014	0.0012	0.001	0.0007
60.326	-0.0083	-0.0075	-0.0064	-0.0051	-0.0034
60.604	-0.0005	-0.0004	-0.0003	-0.0002	-0.0001
60.882	0.0046	0.0042	0.0037	0.003	0.0021
61.16	-0.0014	-0.0012	-0.001	-0.0007	-0.0004
61.438	-0.0044	-0.004	-0.0034	-0.0026	-0.0017
61.716	0.0011	0.001	0.0009	0.0007	0.0005
61.994	0.0028	0.0026	0.0022	0.0018	0.0012
62.272	0.0011	0.001	0.0008	0.0005	0.0003
62.55	-0.0013	-0.0013	-0.0012	-0.0011	-0.0008
62.828	0.0018	0.0016	0.0012	0.0008	0.0005
63.106	0.0023	0.002	0.0016	0.0012	0.0008
63.384	-0.001	-0.001	-0.0008	-0.0007	-0.0005
63.662	-0.0039	-0.0035	-0.0029	-0.0022	-0.0015
63.94	-0.0013	-0.0011	-0.0007	-0.0004	-0.0001
64.218	0.0058	0.0054	0.0047	0.0037	0.0025
64.496	0.0015	0.0013	0.0009	0.0006	0.0003
64.774	-0.0098	-0.009	-0.0078	-0.0061	-0.0042
65.052	-0.0008	-0.0006	-0.0004	-0.0002	-3.59E-05
65.33	0.0071	0.0066	0.0059	0.0048	0.0034
65.608	0.0015	0.0014	0.0012	0.0009	0.0006
65.886	-0.0052	-0.0049	-0.0044	-0.0036	-0.0025
66.164	-0.0019	-0.0018	-0.0016	-0.0013	-0.0009
66.442	0.0042	0.0039	0.0034	0.0027	0.0019
66.72	0.0032	0.0029	0.0025	0.002	0.0014
66.998	-0.0027	-0.0025	-0.0022	-0.0018	-0.0013
67.276	-0.0017	-0.0017	-0.0015	-0.0013	-0.0009
67.554	-0.0003	-0.0002	-0.0002	-0.0001	1.97E-05
67.832	0.0037	0.0034	0.003	0.0024	0.0017
68.11	-0.0003	-0.0003	-0.0003	-0.0002	-0.0001
68.388	0.0007	0.0005	0.0003	0.0001	-0.0001
68.666	-0.0012	-0.0012	-0.0012	-0.001	-0.0007
68.944	-0.0011	-0.0009	-0.0007	-0.0005	-0.0004
69.222	-0.001	-0.0008	-0.0005	-0.0003	-0.0001
69.5	0.0005	0.0006	0.0006	0.0006	0.0005
69.778	0.001	0.0009	0.0008	0.0006	0.0004
70.056	-0.0022	-0.002	-0.0018	-0.0014	-0.001
70.334	-0.0026	-0.0023	-0.0019	-0.0015	-0.001
70.612	0.0024	0.0022	0.002	0.0016	0.0011
70.89	0.002	0.0018	0.0015	0.0012	0.0008
71.168	-5.34E-06	-0.0001	-0.0001	-0.0002	-0.0002
71.446	-0.0025	-0.0023	-0.0021	-0.0017	-0.0012
71.724	-0.001	-0.0009	-0.0007	-0.0005	-0.0003
72.002	0.0037	0.0034	0.003	0.0024	0.0017
72.28	0.0037	0.0033	0.0027	0.0021	0.0013
72.558	-0.0025	-0.0024	-0.0023	-0.002	-0.0014
72.836	-0.0004	-0.0005	-0.0005	-0.0006	-0.0005
73.114	-0.0001	-0.0001	-0.0001	2.63E-06	0.0001
73.392	-0.0014	-0.0011	-0.0008	-0.0004	-0.0002

Continuación del apéndice 1.

73.67	-0.0007	-0.0005	-0.0003	-0.0001	0.0001
73.948	0.003	0.0028	0.0024	0.0019	0.0012
74.226	-0.0014	-0.0013	-0.0013	-0.0011	-0.0008
74.504	-0.0031	-0.0028	-0.0025	-0.002	-0.0014
74.782	0.0005	0.0005	0.0005	0.0005	0.0004
75.06	0.0021	0.002	0.0018	0.0015	0.001
75.338	0.0038	0.0034	0.0029	0.0021	0.0014
75.616	-0.0061	-0.0057	-0.005	-0.0039	-0.0026
75.894	-0.0032	-0.0029	-0.0024	-0.0018	-0.0012
76.172	0.0061	0.0057	0.0049	0.004	0.0028
76.45	0.0027	0.0025	0.0021	0.0017	0.0011
76.728	-0.0038	-0.0036	-0.0032	-0.0026	-0.0019
77.006	-0.0045	-0.0042	-0.0036	-0.0028	-0.0019
77.284	0.0011	0.0012	0.0012	0.0011	0.0008
77.562	0.0071	0.0065	0.0057	0.0045	0.0031
77.84	-0.0017	-0.0017	-0.0016	-0.0013	-0.0009
78.118	-0.0054	-0.005	-0.0044	-0.0036	-0.0025
78.396	0.005	0.0046	0.0038	0.0029	0.0019
78.674	0.0022	0.002	0.0017	0.0014	0.0009
78.952	-0.0047	-0.0042	-0.0036	-0.0027	-0.0018
79.23	-0.0028	-0.0025	-0.002	-0.0015	-0.0009
79.508	0.0053	0.0049	0.0043	0.0034	0.0023
79.786	0.0001	4.92E-05	6.80E-06	-1.00E-06	1.21E-05
80.064	-0.0051	-0.0046	-0.004	-0.0031	-0.0021
80.342	0.0002	0.0002	0.0002	0.0002	0.0001
80.62	0.0059	0.0054	0.0046	0.0036	0.0024
80.898	0.0029	0.0026	0.0021	0.0015	0.0009
81.176	-0.0067	-0.0063	-0.0055	-0.0044	-0.003
81.454	-0.0021	-0.0019	-0.0016	-0.0012	-0.0008
81.732	0.0067	0.0062	0.0054	0.0044	0.003
82.01	-0.0016	-0.0014	-0.0011	-0.0007	-0.0004
82.288	-0.0039	-0.0036	-0.0031	-0.0024	-0.0017
82.566	-0.0001	-0.0002	-0.0002	-0.0001	-0.0001
82.844	0.0021	0.002	0.0018	0.0014	0.001
83.122	0.0055	0.005	0.0042	0.0032	0.002
83.4	-0.0089	-0.0082	-0.0071	-0.0055	-0.0037
83.678	-0.0017	-0.0014	-0.0011	-0.0008	-0.0006
83.956	0.0085	0.0078	0.0067	0.0053	0.0037
84.234	-0.0012	-0.0011	-0.001	-0.0008	-0.0005
84.512	-0.0038	-0.0036	-0.0032	-0.0026	-0.0019
84.79	-0.0004	-0.0005	-0.0005	-0.0004	-0.0002
85.068	0.0046	0.0043	0.0037	0.0029	0.0019
85.346	-0.0033	-0.003	-0.0025	-0.0019	-0.0012
85.624	-0.0008	-0.0007	-0.0005	-0.0004	-0.0003
85.902	0.0008	0.0007	0.0005	0.0005	0.0004
86.18	0.0011	0.0011	0.0009	0.0007	0.0004
86.458	0.0005	0.0004	0.0003	0.0002	0.0001
86.736	-0.0008	-0.0008	-0.0007	-0.0006	-0.0005
87.014	0.0006	0.0006	0.0004	0.0003	0.0002
87.292	0.0017	0.0016	0.0013	0.001	0.0007
87.57	-0.0043	-0.0039	-0.0033	-0.0025	-0.0016
87.848	0.0001	0.0002	0.0002	0.0002	0.0002
88.126	0.0031	0.0029	0.0025	0.002	0.0014
88.404	-0.0002	-0.0001	-0.0002	-0.0002	-0.0002
88.682	0.0014	0.0012	0.0008	0.0005	0.0002

Continuación del apéndice 1.

88.96	-0.0022	-0.0021	-0.0019	-0.0016	-0.0011
89.238	-0.0006	-0.0005	-0.0003	-0.0002	-0.0001
89.516	-0.0013	-0.0011	-0.0008	-0.0004	-0.0001
89.794	0.0009	0.001	0.001	0.0009	0.0006
90.072	0.0007	0.0006	0.0005	0.0004	0.0003
90.35	-0.0016	-0.0015	-0.0013	-0.0011	-0.0008
90.628	0.0001	0.0001	3.05E-05	-2.69E-05	-0.0001
90.906	0.0007	0.0007	0.0006	0.0005	0.0003
91.184	0.0033	0.003	0.0025	0.0019	0.0012
91.462	-0.0019	-0.0018	-0.0016	-0.0013	-0.0009
91.74	-0.0021	-0.0019	-0.0016	-0.0013	-0.0009
92.018	0.0017	0.0016	0.0014	0.0011	0.0008
92.296	0.0015	0.0014	0.0012	0.001	0.0007
92.574	-0.0005	-0.0005	-0.0004	-0.0004	-0.0003
92.852	-0.0008	-0.0008	-0.0008	-0.0007	-0.0005
93.13	-0.0002	-0.0002	-0.0002	-0.0002	-0.0001
93.408	0.0011	0.0011	0.0009	0.0008	0.0005
93.686	-0.0023	-0.0021	-0.0017	-0.0012	-0.0008
93.964	-0.0006	-0.0005	-0.0003	-0.0002	-0.0001
94.242	0.0008	0.0007	0.0006	0.0006	0.0004
94.52	0.0048	0.0043	0.0036	0.0027	0.0017
94.798	-0.0037	-0.0035	-0.0031	-0.0025	-0.0017
95.076	-0.0037	-0.0033	-0.0028	-0.0022	-0.0015
95.354	0.0028	0.0026	0.0023	0.002	0.0014
95.632	0.003	0.0028	0.0025	0.002	0.0014
95.91	0.0001	-1.64E-05	-0.0001	-0.0002	-0.0002
96.188	-0.0046	-0.0043	-0.0038	-0.0031	-0.0021
96.466	0.0015	0.0014	0.0012	0.001	0.0006
96.744	0.0035	0.0032	0.0028	0.0022	0.0016
97.022	0.0006	0.0006	0.0005	0.0003	0.0002
97.3	-0.0037	-0.0035	-0.0031	-0.0024	-0.0017
97.578	-0.0018	-0.0016	-0.0013	-0.001	-0.0006
97.856	0.0022	0.0022	0.002	0.0017	0.0012
98.134	0.0028	0.0026	0.0023	0.0018	0.0012
98.412	-0.0041	-0.0038	-0.0033	-0.0026	-0.0017
98.69	-0.0028	-0.0026	-0.0022	-0.0017	-0.0012
98.968	0.004	0.0037	0.0032	0.0026	0.0018
99.246	0.0006	0.0006	0.0005	0.0004	0.0003
99.524	-0.0009	-0.0009	-0.0008	-0.0007	-0.0005
99.802	-0.0032	-0.003	-0.0026	-0.002	-0.0013
100.08	0.0043	0.004	0.0034	0.0026	0.0017
100.358	0.0001	0.0001	2.60E-05	4.94E-05	0.0001
100.636	-0.0017	-0.0015	-0.0013	-0.001	-0.0008
100.914	-0.0016	-0.0015	-0.0013	-0.001	-0.0006
101.192	0.0035	0.0032	0.0028	0.0021	0.0014
101.47	0.0012	0.001	0.0008	0.0006	0.0004
101.748	-0.0046	-0.0042	-0.0036	-0.0028	-0.0019
102.026	0.0009	0.0009	0.0008	0.0006	0.0004
102.304	0.0042	0.0038	0.0032	0.0025	0.0017
102.582	0.0001	1.65E-05	-4.55E-05	-0.0001	-0.0001
102.86	-0.0044	-0.0041	-0.0036	-0.0028	-0.0019
103.138	0.001	0.001	0.0009	0.0007	0.0005
103.416	0.0029	0.0027	0.0024	0.0019	0.0013
103.694	-0.0011	-0.0009	-0.0008	-0.0006	-0.0004
103.972	-0.0034	-0.0031	-0.0027	-0.0021	-0.0014

Continuación del apéndice 1.

104.25	-0.0011	-0.0009	-0.0007	-0.0005	-0.0003
104.528	0.0055	0.005	0.0043	0.0034	0.0023
104.806	-0.0006	-0.0006	-0.0006	-0.0005	-0.0004
105.084	-0.0038	-0.0035	-0.0031	-0.0025	-0.0018
105.362	0.0036	0.0032	0.0027	0.002	0.0013
105.64	0.003	0.0027	0.0022	0.0017	0.0011
105.918	-0.0058	-0.0052	-0.0044	-0.0034	-0.0023
106.196	-0.0002	-0.0001	-1.29E-05	0.0001	0.0001
106.474	0.004	0.0037	0.0032	0.0025	0.0018
106.752	0.0001	0.0001	2.91E-05	-1.61E-05	-4.96E-05
107.03	-0.0036	-0.0034	-0.0029	-0.0023	-0.0016
107.308	-0.0021	-0.0019	-0.0015	-0.0012	-0.0007
107.586	0.0038	0.0035	0.0031	0.0025	0.0017
107.864	0.0009	0.0009	0.0008	0.0007	0.0005
108.142	-0.003	-0.0028	-0.0024	-0.0019	-0.0013
108.42	-0.0023	-0.0021	-0.0018	-0.0014	-0.0009
108.698	0.0048	0.0044	0.0038	0.003	0.002
108.976	0.0041	0.0037	0.003	0.0023	0.0015
109.254	-0.0064	-0.006	-0.0052	-0.0041	-0.0028
109.532	-0.0027	-0.0024	-0.002	-0.0016	-0.0011
109.81	0.0063	0.0058	0.0051	0.0041	0.0028
110.088	0.0006	0.0006	0.0005	0.0004	0.0003
110.366	-0.0051	-0.0047	-0.0041	-0.0032	-0.0022
110.644	-0.0015	-0.0014	-0.0012	-0.0009	-0.0006
110.922	0.0056	0.0051	0.0044	0.0035	0.0023
111.2	0.0009	0.0008	0.0006	0.0005	0.0003
111.478	-0.0045	-0.0042	-0.0036	-0.0029	-0.002
111.756	0.0002	0.0001	0.0001	0.0001	2.40E-05
112.034	0.0013	0.0012	0.0011	0.001	0.0007
112.312	0.0016	0.0016	0.0014	0.0011	0.0008
112.59	-0.0002	-0.0003	-0.0003	-0.0003	-0.0002
112.868	-0.0017	-0.0016	-0.0014	-0.0012	-0.0009
113.146	0.0023	0.002	0.0017	0.0013	0.0008
113.424	-0.0009	-0.0008	-0.0007	-0.0005	-0.0003
113.702	-0.0004	-0.0003	-0.0002	-0.0002	-0.0001
113.98	-0.0004	-0.0004	-0.0003	-0.0002	-0.0001
114.258	0.002	0.0019	0.0016	0.0012	0.0008
114.536	-0.0012	-0.0011	-0.001	-0.0008	-0.0005
114.814	-0.0038	-0.0034	-0.0028	-0.0022	-0.0014
115.092	0.0006	0.0006	0.0006	0.0006	0.0005
115.37	0.0053	0.0049	0.0042	0.0033	0.0022
115.648	-0.0043	-0.004	-0.0034	-0.0027	-0.0018
115.926	-0.0022	-0.002	-0.0017	-0.0014	-0.001
116.204	0.0049	0.0044	0.0038	0.0029	0.002
116.482	0.0008	0.0007	0.0006	0.0004	0.0003
116.76	-0.0039	-0.0035	-0.003	-0.0024	-0.0016
117.038	0.0018	0.0017	0.0014	0.0011	0.0007
117.316	0.0019	0.0017	0.0015	0.0011	0.0008
117.594	-0.0019	-0.0018	-0.0015	-0.0012	-0.0008
117.872	-0.0011	-0.001	-0.0009	-0.0006	-0.0004
118.15	0.0013	0.0012	0.001	0.0008	0.0005
118.428	-0.0009	-0.0008	-0.0006	-0.0004	-0.0002
118.706	0.0007	0.0006	0.0006	0.0004	0.0003
118.984	-0.0028	-0.0026	-0.0022	-0.0016	-0.0011
119.262	0.0011	0.0011	0.001	0.0008	0.0005

Continuación del apéndice 1.

119.54	0.004	0.0036	0.003	0.0023	0.0016
119.818	-0.0008	-0.0008	-0.0008	-0.0008	-0.0006
120.096	-0.0049	-0.0045	-0.0039	-0.003	-0.0021
120.374	0.0041	0.0038	0.0033	0.0026	0.0017
120.652	0.0039	0.0035	0.003	0.0023	0.0016
120.93	-0.0028	-0.0026	-0.0023	-0.0019	-0.0014
121.208	-0.0034	-0.0032	-0.0028	-0.0022	-0.0015
121.486	0.0028	0.0026	0.0023	0.0018	0.0013
121.764	0.0021	0.0019	0.0017	0.0013	0.0009
122.042	0.0004	0.0004	0.0002	0.0001	1.19E-05
122.32	-0.0054	-0.005	-0.0043	-0.0034	-0.0022
122.598	-0.0002	-0.0001	8.27E-07	0.0001	0.0001
122.876	0.0022	0.0021	0.0019	0.0017	0.0013
123.154	0.0004	0.0005	0.0005	0.0004	0.0003
123.432	-0.0042	-0.0038	-0.0033	-0.0026	-0.0017
123.71	0.0009	0.0008	0.0007	0.0005	0.0003
123.988	0.0035	0.0032	0.0027	0.0021	0.0014
124.266	0.0004	0.0003	0.0002	0.0001	-1.21E-05
124.544	-0.0015	-0.0014	-0.0013	-0.0011	-0.0008
124.822	-0.0007	-0.0006	-0.0006	-0.0005	-0.0003
125.1	0.0017	0.0016	0.0014	0.0011	0.0008
125.378	0.0003	0.0003	0.0003	0.0003	0.0002
125.656	-0.0022	-0.002	-0.0017	-0.0013	-0.0009
125.934	0.0009	0.0008	0.0007	0.0005	0.0003
126.212	0.0001	0.0001	0.0001	0.0001	0.0001
126.49	-0.0018	-0.0016	-0.0013	-0.001	-0.0006
126.768	0.0025	0.0023	0.0019	0.0015	0.001
127.046	0.0021	0.0019	0.0015	0.0011	0.0007
127.324	-0.0057	-0.0052	-0.0044	-0.0035	-0.0023
127.602	0.0057	0.0053	0.0045	0.0034	0.0022
127.88	0.0019	0.0016	0.0013	0.0009	0.0007
128.158	-0.0032	-0.0029	-0.0025	-0.0021	-0.0015
128.436	-0.0044	-0.004	-0.0034	-0.0025	-0.0016
128.714	3.42E-05	0.0002	0.0004	0.0005	0.0005
128.992	0.0031	0.003	0.0027	0.0023	0.0017
129.27	0.0003	0.0002	0.0002	0.0001	0.0001
129.548	-0.0038	-0.0036	-0.0032	-0.0026	-0.0018
129.826	0.0017	0.0015	0.0012	0.0009	0.0005
130.104	0.0039	0.0035	0.003	0.0023	0.0015
130.382	-0.0037	-0.0033	-0.0028	-0.0022	-0.0014
130.66	0.0008	0.0007	0.0006	0.0005	0.0003
130.938	0.0052	0.0047	0.0039	0.003	0.0019
131.216	-0.0026	-0.0024	-0.0022	-0.0018	-0.0013
131.494	-0.0044	-0.004	-0.0034	-0.0026	-0.0018
131.772	0.0044	0.0041	0.0035	0.0028	0.0019
132.05	-0.0001	-3.40E-05	3.11E-05	0.0001	0.0001
132.328	-0.0006	-0.0005	-0.0004	-0.0004	-0.0003
132.606	-0.0026	-0.0025	-0.0022	-0.0017	-0.0011
132.884	-0.0025	-0.0022	-0.0017	-0.0013	-0.0008
133.162	0.0042	0.004	0.0035	0.0028	0.002
133.44	0.0025	0.0022	0.0019	0.0014	0.0009
133.718	-0.0033	-0.0031	-0.0028	-0.0023	-0.0017
133.996	-0.0035	-0.0032	-0.0028	-0.0022	-0.0015
134.274	0.0024	0.0023	0.0021	0.0017	0.0012
134.552	0.0033	0.0031	0.0027	0.0022	0.0015

Continuación del apéndice 1.

134.83	-0.0027	-0.0025	-0.0021	-0.0017	-0.0012
135.108	-0.0025	-0.0023	-0.0021	-0.0016	-0.0011
135.386	0.0051	0.0047	0.0039	0.003	0.002
135.664	0.002	0.0018	0.0014	0.001	0.0006
135.942	-0.0063	-0.0057	-0.0049	-0.0038	-0.0026
136.22	2.53E-05	0.0001	0.0002	0.0002	0.0002
136.498	0.0072	0.0066	0.0056	0.0044	0.003
136.776	-0.0017	-0.0016	-0.0015	-0.0012	-0.0008
137.054	-0.0092	-0.0084	-0.0072	-0.0056	-0.0037
137.332	0.0045	0.0042	0.0037	0.003	0.002
137.61	0.006	0.0055	0.0047	0.0037	0.0026
137.888	-0.0024	-0.0023	-0.002	-0.0017	-0.0012
138.166	-0.0052	-0.0048	-0.0042	-0.0034	-0.0023
138.444	0.0035	0.0032	0.0028	0.0021	0.0014
138.722	0.0028	0.0026	0.0023	0.0018	0.0013
139	-0.002	-0.0018	-0.0015	-0.0011	-0.0008
139.278	-0.0061	-0.0056	-0.0047	-0.0036	-0.0024
139.556	0.0042	0.0039	0.0035	0.0027	0.0018
139.834	0.0048	0.0043	0.0037	0.0029	0.002
140.112	-0.0042	-0.0039	-0.0034	-0.0028	-0.002
140.39	-0.007	-0.0064	-0.0055	-0.0042	-0.0028
140.668	0.0028	0.0027	0.0026	0.0022	0.0016
140.946	0.0083	0.0077	0.0066	0.0052	0.0036
141.224	0.0006	0.0003	4.54E-05	-0.0002	-0.0003
141.502	-0.0083	-0.0078	-0.0069	-0.0056	-0.0039
141.78	0.0017	0.0015	0.0013	0.001	0.0007
142.058	0.0098	0.009	0.0077	0.0061	0.0041
142.336	-0.0012	-0.0012	-0.0011	-0.0009	-0.0007
142.614	-0.0114	-0.0105	-0.009	-0.007	-0.0047
142.892	0.0036	0.0034	0.003	0.0024	0.0017
143.17	0.0087	0.0079	0.0069	0.0054	0.0038
143.448	-0.0095	-0.0086	-0.0073	-0.0056	-0.0037
143.726	-0.0042	-0.0038	-0.0031	-0.0024	-0.0016
144.004	0.0073	0.0067	0.0057	0.0045	0.0031
144.282	0.0041	0.0037	0.0031	0.0024	0.0015
144.56	-0.0065	-0.006	-0.0053	-0.0042	-0.0029
144.838	-0.0048	-0.0044	-0.0037	-0.0029	-0.0019
145.116	0.0078	0.0072	0.0063	0.005	0.0034
145.394	0.0067	0.0061	0.0051	0.004	0.0026
145.672	-0.0044	-0.0042	-0.0039	-0.0033	-0.0024
145.95	-0.0102	-0.0094	-0.0081	-0.0064	-0.0043
146.228	0.0036	0.0035	0.0033	0.0028	0.002
146.506	0.0119	0.011	0.0095	0.0076	0.0052
146.784	-0.0082	-0.0076	-0.0066	-0.0052	-0.0035
147.062	-0.0091	-0.0084	-0.0072	-0.0057	-0.0039
147.34	0.0075	0.0069	0.006	0.0047	0.0033
147.618	0.0069	0.0063	0.0055	0.0043	0.0029
147.896	-0.0082	-0.0076	-0.0065	-0.0051	-0.0035
148.174	-0.0047	-0.0043	-0.0037	-0.0029	-0.0019
148.452	0.0085	0.0078	0.0067	0.0053	0.0036
148.73	0.006	0.0054	0.0045	0.0034	0.0022
149.008	-0.0085	-0.0079	-0.0069	-0.0055	-0.0038
149.286	-0.0046	-0.0041	-0.0035	-0.0028	-0.0019
149.564	0.0088	0.0081	0.0071	0.0056	0.0039
149.842	0.0026	0.0024	0.0021	0.0017	0.0012

Continuación del apéndice 1.

150.12	-0.0083	-0.0076	-0.0065	-0.0051	-0.0035
150.398	-0.0036	-0.0032	-0.0027	-0.0021	-0.0014
150.676	0.0061	0.0057	0.005	0.0041	0.0028
150.954	0.0046	0.0042	0.0036	0.0028	0.0019
151.232	-0.0059	-0.0055	-0.0049	-0.0039	-0.0027
151.51	-0.0049	-0.0045	-0.0039	-0.0031	-0.0021
151.788	0.0067	0.0062	0.0054	0.0043	0.0029
152.066	0.0052	0.0048	0.0041	0.0032	0.0021
152.344	-0.006	-0.0056	-0.0049	-0.0039	-0.0027
152.622	-0.0023	-0.0022	-0.002	-0.0016	-0.0012
152.9	0.005	0.0046	0.004	0.0031	0.0021
153.178	0.0044	0.004	0.0034	0.0026	0.0017
153.456	-0.0074	-0.0068	-0.0059	-0.0046	-0.0031
153.734	-0.0031	-0.0028	-0.0023	-0.0017	-0.0012
154.012	0.0039	0.0037	0.0033	0.0027	0.002
154.29	0.0032	0.003	0.0027	0.0021	0.0015
154.568	-0.0035	-0.0033	-0.0029	-0.0024	-0.0016
154.846	-0.006	-0.0055	-0.0047	-0.0037	-0.0025
155.124	0.0039	0.0037	0.0033	0.0027	0.0018
155.402	0.0036	0.0033	0.0029	0.0024	0.0017
155.68	-0.0021	-0.002	-0.0017	-0.0014	-0.001
155.958	-0.0042	-0.0039	-0.0035	-0.0028	-0.0019
156.236	0.0021	0.0019	0.0017	0.0014	0.0009
156.514	0.0076	0.0069	0.0059	0.0045	0.003
156.792	-0.003	-0.0029	-0.0026	-0.0021	-0.0015
157.07	-0.0034	-0.0031	-0.0028	-0.0023	-0.0017
157.348	0.0031	0.0028	0.0024	0.0019	0.0013
157.626	-0.0001	2.68E-05	0.0001	0.0002	0.0002
157.904	-0.0006	-0.0005	-0.0003	-0.0002	-0.0001
158.182	-0.0011	-0.001	-0.0009	-0.0007	-0.0004
158.46	0.0008	0.0008	0.0006	0.0005	0.0003
158.738	0.0002	0.0001	0.0001	0.0001	0.0001
159.016	-0.0023	-0.0021	-0.0017	-0.0013	-0.0008
159.294	-0.0003	-0.0002	-0.0001	-1.97E-05	2.97E-05
159.572	-0.0009	-0.0008	-0.0006	-0.0003	-0.0001
159.85	0.0038	0.0035	0.003	0.0023	0.0015
160.128	0.0017	0.0014	0.0011	0.0007	0.0004
160.406	-0.0034	-0.0032	-0.0029	-0.0024	-0.0017
160.684	-0.0026	-0.0024	-0.002	-0.0015	-0.001
160.962	0.0057	0.0053	0.0046	0.0037	0.0025
161.24	0.0023	0.0021	0.0017	0.0013	0.0009
161.518	-0.0078	-0.0072	-0.0062	-0.0048	-0.0033
161.796	-0.0012	-0.0011	-0.0008	-0.0006	-0.0004
162.074	0.0074	0.0068	0.0059	0.0047	0.0032
162.352	-0.0026	-0.0023	-0.002	-0.0015	-0.001
162.63	-0.0038	-0.0035	-0.003	-0.0024	-0.0017
162.908	0.0007	0.0006	0.0005	0.0004	0.0003
163.186	0.0035	0.0032	0.0028	0.0022	0.0015
163.464	-0.0003	-0.0003	-0.0003	-0.0003	-0.0002
163.742	0.0002	0.0001	-4.45E-06	-0.0001	-0.0002
164.02	0.0019	0.0016	0.0012	0.0008	0.0005
164.298	-0.0005	-0.0005	-0.0005	-0.0004	-0.0003
164.576	-0.0013	-0.0012	-0.001	-0.0007	-0.0005
164.854	0.001	0.001	0.0009	0.0008	0.0006
165.132	-0.0024	-0.0021	-0.0017	-0.0012	-0.0007

Continuación del apéndice 1.

165.41	0.001	0.001	0.0009	0.0007	0.0005
165.688	-0.0004	-0.0004	-0.0003	-0.0002	-0.0001
165.966	-0.0009	-0.0008	-0.0007	-0.0006	-0.0004
166.244	0.0002	0.0002	0.0002	0.0001	0.0001
166.522	-2.20E-05	0	2.40E-05	3.77E-05	3.53E-05
166.8	0.0006	0.0005	0.0004	0.0003	0.0002
167.078	0.0014	0.0012	0.001	0.0007	0.0005

Fuente: elaboración propia.

ANEXOS

Anexo 1. Código de programa de extracción datos

```
Function varargout = Grafico1(varargin)
% GRAFICO1 MATLAB code for Grafico1.fig
%   GRAFICO1, by itself, creates a new GRAFICO1 or raises the existing
%   singleton*.
%
%   H = GRAFICO1 returns the handle to a new GRAFICO1 or the handle to
%   the existing singleton*.
%
%   GRAFICO1('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GRAFICO1.M with the given input arguments.
%
%   GRAFICO1('Property','Value',...) creates a new GRAFICO1 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Grafico1_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Grafico1_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Grafico1

% Last Modified by GUIDE v2.5 17-Sep-2021 16:55:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Grafico1_OpeningFcn, ...
                  'gui_OutputFcn', @Grafico1_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before Grafico1 is made visible.
function Grafico1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Grafico1 (see VARARGIN)
```

Continuación del anexo 1.

```
% Choose default command line output for Grafico1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
global ax;
global ay;
global az;
global n;
global l;

global out;
global accel;
global accel1;
global accel2;
global accel3;
global accel4;
global accel5;
global accel6;
global accel7;

accel1=load('DATALOG.CSV');
accel2=load('DATALOG1.CSV');
accel3=load('DATALOG.CSV');
accel4=load('DATALOG.CSV');
accel5=load('DATALOG.CSV');
accel6=load('DATALOG.CSV');
accel7=load('DATALOG.CSV');

accel=accel1;

n=length(accel);
tmili=accel(:,1);

l=500;

ax=accel(:,2);
ay=accel(:,3);
az=accel(:,4);

T = datetime(2021,8,23,0,0,0,0,'TimeZone','UTC','F','uuuu-MM-dd" T"HH:mm:ss.SSS Z');
addMS = milliseconds(accel(:,1));
out = T + addMS;

Promediox=mean(ax);
Promedioy=mean(ay);
Promedioz=mean(az);

ax=ax/Promediox - 1;
ay=ay/Promedioy - 1;
az=az/Promedioz + 8.81;
gxFilt=0;
gyFilt=0;
gzFilt=9.81;
alpha=.5;
i=1;
```

Continuación del anexo 1.

```
nx=length(ax);
ny=length(ay);
nz=length(az);
AcFiltx=zeros(n,1);
AcFilty=zeros(n,1);
AcFiltz=ones(n,1);
AcFlitz=AcFiltz*9.81;

while i<n
    gx=ax(i);
    gxFilt = (1 - alpha)*gxFilt + alpha*gx;
    AcFiltx(i)=gxFilt;

    gy=ay(i);
    gyFilt = (1 - alpha)*gyFilt + alpha*gy;
    AcFilty(i)=gyFilt;

    gz=az(i);
    gzFilt = (1 - alpha)*gzFilt + alpha*gz;

    AcFiltz(i)=gzFilt;

    i=i+1;
end
guidata(hObject,handles)

axes(handles.axes1)
%plot(axismo1);
subplot(3,2,1);
plot(out(1-200:1+200),AcFiltx(1-200:1+200));
title('Aceleración x (m/s2)')
ylabel('m/s^2')
grid on
grid minor
hold on

subplot(3,2,3);
plot(out(1-200:1+200),AcFilty(1-200:1+200));
title('Aceleración y (m/s2)')
ylabel('m/s^2')
grid on
grid minor
hold on

subplot(3,2,5);
plot(out(1-200:1+200),AcFiltz(1-200:1+200));
ylabel('m/s^2')
grid on
grid minor
title('Aceleración z (m/s2)')
hold on
% UIWAIT makes Grafico1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

Continuación del anexo 1.

```
% --- Outputs from this function are returned to the command line.
function varargout = Grafico1_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global duracion
global ax;
global ay;
global az;
global n;
global l;
global out;
global year1;
global mes1;
global dia1;
global minutos1;
global hora1;
global segundos1;
global year2;
global mes2;
global dia2;
global minutos2;
global hora2;
global segundos2;
global filtro;
global accel;
global aceleracionesx;
global aceleracionesy;
global aceleracionesz;
global val;
global accel1;
global accel2;
global accel3;
global accel4;
```

Continuación del anexo 1.

```
global accel5;
global accel6;
global accel7;

switch val
    case 1
        accel=accel1;
    case 2
        accel=accel2;
    case 3
        accel=accel3;
    case 4
        accel=accel4;
    case 5
        accel=accel5;
    case 6
        accel=accel6;
    case 7
        accel=accel7;
    otherwise
        accel=accel8;
end

T = datetime(year1,mes1,dia1,hora1,minutos1,segundos1,0,'TimeZone','UTC','F','uuuu-MM-dd"T"HH:mm:ss.SSS Z');
addMS = milliseconds(accel(:,1));
out = T + addMS;

T2 = datetime(year2,mes2,dia2,hora2,minutos2,segundos2,0,'TimeZone','UTC','F','uuuu-MM-dd"T"HH:mm:ss.SSS Z');

Tinicio=datetime(year2,mes2,dia2,hora2,minutos2,segundos2,0,'TimeZone','UTC','F','uuuu-MM-dd"T"HH:mm:ss.SSS Z');
Tfinal=datetime(year2,mes2,dia2,hora2,minutos2,(segundos2+duracion),0,'TimeZone','UTC','F','uuuu-MM-dd"T"HH:mm:ss.SSS Z');

im=1;
j=1;
while out(im,1)<=Tfinal
    if out(im,1)>=Tinicio
        tiempo(j,1)=out(im,1);
        j=j+1;
    end
    im=im+1;
end

gxFilt=0;
gyFilt=0;
gzFilt=9.81;
alpha=filtro;
i=1;
nx=length(ax);
ny=length(ay);
nz=length(az);
AcFiltx=zeros(n,1);
AcFilty=zeros(n,1);
AcFiltz=ones(n,1);
AcFlitz=AcFiltz*9.81;
i=1;
```

Continuación del anexo 1.

```
while i<n
    gx=ax(i);
    gxFilt = (1 - alpha)*gxFilt + alpha*gx;
    AcFiltx(i)=gxFilt;

    gy=ay(i);
    gyFilt = (1 - alpha)*gyFilt + alpha*gy;
    AcFilty(i)=gyFilt;

    gz=az(i);
    gzFilt = (1 - alpha)*gzFilt + alpha*gz;
    AcFiltz(i)=gzFilt;

    i=i+1;
end
%axes(handles.axes1)
%plot(axismo1);
tiempo0 = seconds(addMS((im-j+2):im));
tiempo0 = tiempo0 - tiempo0(1);
aceleracionesx = zeros(j-1,2);
aceleracionesy = zeros(j-1,2);
aceleracionesz = zeros(j-1,2);
aceleracionesx(:,1) = tiempo0;
aceleracionesy(:,1) = tiempo0;
aceleracionesz(:,1) = tiempo0;
aceleracionesx(:,2) = AcFiltx((im-j+2):im);
aceleracionesy(:,2) = AcFilty((im-j+2):im);
aceleracionesz(:,2) = AcFiltz((im-j+2):im);
subplot(3,2,1);
hold off

subplot(3,2,3);
hold off

subplot(3,2,5);
hold off

subplot(3,2,1);
plot(tiempo(1:j-1),AcFiltx((im-j+2):im));
title('Aceleración x (m/s2)')
ylabel('m/s^2')
grid on

grid minor
hold on

subplot(3,2,3);
plot(tiempo(1:j-1),AcFilty((im-j+2):im));
title('Aceleración y (m/s2)')
ylabel('m/s^2')

grid on
grid minor
hold on
```

Continuación del anexo 1.

```
subplot(3,2,5);
plot(tiempo(1:j-1),AcFiltz((im-j+2):im));
ylabel('m/s^2')
grid on
grid minor
title('Aceleración z (m/s2)')
hold on

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global aceleracionex;
global aceleracionesy;
global aceleracionesz;

csvwrite('Aceleraciones eje x.csv',aceleracionex);
csvwrite('Aceleraciones eje y.csv',aceleracionesy);
csvwrite('Aceleraciones eje z.csv',aceleracionesz);

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject handle to axes1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global val
val= get(hObject,'Value');
%f = str2double(get(hObject,'String'))
% switch val
% case 1
% case 2
% case 3
% case 4
% case 5
% case 6
% case 7
% otherwise
% end

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array
% contents{get(hObject,'Value')} returns selected item from popupmenu1
```

Continuación del anexo 1.

```
% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%    get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)
% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
get(hObject,'Min')
get(hObject,'Max')

% Hints: get(hObject,'Value') returns position of slider
%    get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

Continuación del anexo 1.

```
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
global year1
year1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Continuación del anexo 1.

```
% Hints: get(hObject,'String') returns contents of edit3 as text
%   str2double(get(hObject,'String')) returns contents of edit3 as a double
global mes1
mes1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject   handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%   str2double(get(hObject,'String')) returns contents of edit4 as a double
global dia1
dia1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject   handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%   str2double(get(hObject,'String')) returns contents of edit5 as a double
global hora1
hora1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called
```

Continuación del anexo 1.

```
% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject   handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%   str2double(get(hObject,'String')) returns contents of edit6 as a double
global minutos1
minutos1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject   handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%   str2double(get(hObject,'String')) returns contents of edit7 as a double
global segundos1
segundos1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject   handle to edit8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit8 as text
%   str2double(get(hObject,'String')) returns contents of edit8 as a double
global year2
year2 = str2double(get(hObject,'String'));
```

Continuación del anexo 1.

```
% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%       str2double(get(hObject,'String')) returns contents of edit9 as a double
global mes2
mes2 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%       str2double(get(hObject,'String')) returns contents of edit10 as a double
global dia2
dia2 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Continuación del anexo 1.

```
function edit11_Callback(hObject, eventdata, handles)
% hObject handle to edit11 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
% str2double(get(hObject,'String')) returns contents of edit11 as a double
global hora2
hora2 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit11 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit12_Callback(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
% str2double(get(hObject,'String')) returns contents of edit12 as a double
global minutos2
minutos2 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit13_Callback(hObject, eventdata, handles)
% hObject handle to edit13 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
% str2double(get(hObject,'String')) returns contents of edit13 as a double
global segundos2
segundos2 = str2double(get(hObject,'String'));
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Continuación del anexo 1.

```
function edit14_Callback(hObject, eventdata, handles)
% hObject handle to edit14 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
% str2double(get(hObject,'String')) returns contents of edit14 as a double
global filtro
filtro = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit14 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)
% hObject handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
% str2double(get(hObject,'String')) returns contents of edit15 as a double
global duracion
duracion = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Fuente: CAPTERRA (2019). *Herramientas de extracción de datos*. Consultado el 12 de diciembre de 2021. Recuperado de <https://www.capterra.es/directory/30619/data-extraction/software>

Anexo 2. Código de programa de monitoreo de salud estructural

```
function varargout = MONITOREO(varargin)
% MONITOREO MATLAB code for MONITOREO.fig
%   MONITOREO, by itself, creates a new MONITOREO or raises the existing
%   singleton*.
%
%   H = MONITOREO returns the handle to a new MONITOREO or the handle to
%   the existing singleton*.
%
%   MONITOREO('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in MONITOREO.M with the given input arguments.
%
%   MONITOREO('Property','Value',...) creates a new MONITOREO or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before MONITOREO_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to MONITOREO_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MONITOREO

% Last Modified by GUIDE v2.5 29-Sep-2021 23:43:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @MONITOREO_OpeningFcn, ...
                  'gui_OutputFcn', @MONITOREO_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before MONITOREO is made visible.
function MONITOREO_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MONITOREO (see VARARGIN)
global accelX1;
global accelY1;
global accelZ1;
```

Continuación del anexo 2.

```
global accelX2;
global accelY2;
global accelZ2;
global accelX3;
global accelY3;
global accelZ3;
global accelX4;
global accelY4;
global accelZ4;
global accelX5;

global accelY5;
global accelZ5;
global accelX6;
global accelY6;
global accelZ6;
global accelX7;
global accelY7;
global accelZ7;

accelX1=load('Aceleraciones eje x.csv');
accelY1=load('Aceleraciones eje y.csv');
accelZ1=load('Aceleraciones eje z.csv');

accelX2=load('Aceleraciones eje x1.csv');
accelY2=load('Aceleraciones eje y1.csv');
accelZ2=load('Aceleraciones eje z1.csv');

accelX3=load('Aceleraciones eje x2.csv');
accelY3=load('Aceleraciones eje y2.csv');
accelZ3=load('Aceleraciones eje z2.csv');

accelX4=load('Aceleraciones eje x3.csv');
accelY4=load('Aceleraciones eje y3.csv');
accelZ4=load('Aceleraciones eje z3.csv');

accelX5=load('Aceleraciones eje x4.csv');
accelY5=load('Aceleraciones eje y4.csv');
accelZ5=load('Aceleraciones eje z4.csv');

accelX6=load('Aceleraciones eje x5.csv');
accelY6=load('Aceleraciones eje y5.csv');
accelZ6=load('Aceleraciones eje z5.csv');

accelX7=load('Aceleraciones eje x6.csv');
accelY7=load('Aceleraciones eje y6.csv');
accelZ7=load('Aceleraciones eje z6.csv');

% Choose default command line output for MONITOREO
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MONITOREO wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = MONITOREO_OutputFcn(hObject, eventdata, handles)
```

Continuación de anexo 2.

```
% varargin cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global val
val= get(hObject,'Value');
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array
% contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

global accelX1;
global accelY1;
global accelZ1;
global accelX2;
global accelY2;
global accelZ2;
global accelX3;
global accelY3;
global accelZ3;
global accelX4;
global accelY4;
global accelZ4;
global accelX5;
global accelY5;
global accelZ5;
global accelX6;
global accelY6;
global accelZ6;
global accelX7;
```

Continuación del anexo 2.

```
global accelY7;
global accelZ7;
global val;

%%%%% Datos filtro aplicado
order = 3;
framelen = 21;

%%%%%%%%% Aplicar filtro
sgfX1 = sgolayfilt(accelX1(:,2),order,framelen);
sgfY1 = sgolayfilt(accelY1(:,2),order,framelen);
sgfZ1 = sgolayfilt(accelZ1(:,2),order,framelen);

sgfX2 = sgolayfilt(accelX2(:,2),order,framelen);
sgfY2 = sgolayfilt(accelY2(:,2),order,framelen);
sgfZ2 = sgolayfilt(accelZ2(:,2),order,framelen);

sgfX3 = sgolayfilt(accelX3(:,2),order,framelen);
sgfY3 = sgolayfilt(accelY3(:,2),order,framelen);
sgfZ3 = sgolayfilt(accelZ3(:,2),order,framelen);

sgfX4 = sgolayfilt(accelX4(:,2),order,framelen);
sgfY4 = sgolayfilt(accelY4(:,2),order,framelen);
sgfZ4 = sgolayfilt(accelZ4(:,2),order,framelen);

sgfX5 = sgolayfilt(accelX5(:,2),order,framelen);
sgfY5 = sgolayfilt(accelY5(:,2),order,framelen);
sgfZ5 = sgolayfilt(accelZ5(:,2),order,framelen);

sgfX6 = sgolayfilt(accelX6(:,2),order,framelen);

sgfY6 = sgolayfilt(accelY6(:,2),order,framelen);
sgfZ6 = sgolayfilt(accelZ6(:,2),order,framelen);

sgfX7 = sgolayfilt(accelX7(:,2),order,framelen);
sgfY7 = sgolayfilt(accelY7(:,2),order,framelen);
sgfZ7 = sgolayfilt(accelZ7(:,2),order,framelen);

%%%% Encontrar picos

[pksx1,locsx1] = findpeaks(sgfX1);
[pksy1,locsy1] = findpeaks(sgfY1);
[pksz1,locsz1] = findpeaks(sgfZ1);

[pksx2,locsx2] = findpeaks(sgfX2);
[pksy2,locsy2] = findpeaks(sgfY2);
[pksz2,locsz2] = findpeaks(sgfZ2);

[pksx3,locsx3] = findpeaks(sgfX3);
[pksy3,locsy3] = findpeaks(sgfY3);
[pksz3,locsz3] = findpeaks(sgfZ3);

[pksx4,locsx4] = findpeaks(sgfX4);
[pksy4,locsy4] = findpeaks(sgfY4);
[pksz4,locsz4] = findpeaks(sgfZ4);
```

Continuación del anexo 2.

```
[pkx5,locsx5] = findpeaks(sgfX5);
[pky5,locsy5] = findpeaks(sgfY5);
[pkz5,locsz5] = findpeaks(sgfZ5);

[pkx6,locsx6] = findpeaks(sgfX6);
[pky6,locsy6] = findpeaks(sgfY6);
[pkz6,locsz6] = findpeaks(sgfZ6);

[pkx7,locsx7] = findpeaks(sgfX7);
[pky7,locsy7] = findpeaks(sgfY7);
[pkz7,locsz7] = findpeaks(sgfZ7);

%plot(accelZ(:,2),'-')
%hold on
%plot(sgfZ,'.-')
%legend('signal','sgolay')

%%% 1
nlocsx1 = length(locsx1);
nlocsy1 = length(locsy1);
nlocsz1 = length(locsz1);
%%% 2
nlocsx2 = length(locsx2);
nlocsy2 = length(locsy2);
nlocsz2 = length(locsz2);
%%% 3
nlocsx3 = length(locsx3);
nlocsy3 = length(locsy3);
nlocsz3 = length(locsz3);
%%% 4
nlocsx4 = length(locsx4);
nlocsy4 = length(locsy4);
nlocsz4 = length(locsz4);
%%% 5
nlocsx5 = length(locsx5);
nlocsy5 = length(locsy5);
nlocsz5 = length(locsz5);
%%% 6
nlocsx6 = length(locsx6);
nlocsy6 = length(locsy6);
nlocsz6 = length(locsz6);
%%% 7
nlocsx7 = length(locsx7);

nlocsy7 = length(locsy7);
nlocsz7 = length(locsz7);

i=1;
%Encontrar periodo nivel 1
while i<nlocsx1
    Tcalcx1(i)= accelX1(locsx1(i+1),1)-accelX1(locsx1(i),1);
    i=i+1;
end
i=1;
nlocsy1 = length(locsy1);
```

Continuación del anexo 2.

```
while i<nlocsy1
    Tcalcy1(i)= accelY1(locsy1(i+1),1)-accelY1(locsy1(i),1);
    i=i+1;
end
i=1;
nlocs = length(locsz1);
while i<nlocsz1
    Tcalcz1(i)= accelZ1(locsz1(i+1),1)-accelZ1(locsz1(i),1);
    i=i+1;
end
%Encontrar periodo nivel 2
i=1;
while i<nlocsx2
    Tcalcx2(i)= accelX2(locsx2(i+1),1)-accelX2(locsx2(i),1);
    i=i+1;
end
i=1;
nlocsy2 = length(locsy2);
while i<nlocsy2
    Tcalcy2(i)= accelY2(locsy2(i+1),1)-accelY2(locsy2(i),1);
    i=i+1;
end
i=1;
nlocs = length(locsz2);
while i<nlocsz2
    Tcalcz2(i)= accelZ2(locsz2(i+1),1)-accelZ2(locsz2(i),1);
    i=i+1;
end
%Encontrar periodo nivel 3
i=1;
while i<nlocsx3
    Tcalcx3(i)= accelX3(locsx3(i+1),1)-accelX3(locsx3(i),1);
    i=i+1;
end
i=1;
nlocsy3 = length(locsy3);
while i<nlocsy3
    Tcalcy3(i)= accelY3(locsy3(i+1),1)-accelY3(locsy3(i),1);
    i=i+1;
end
i=1;
nlocs = length(locsz3);
while i<nlocsz3
    Tcalcz3(i)= accelZ3(locsz3(i+1),1)-accelZ3(locsz3(i),1);
    i=i+1;
end
%Encontrar periodo nivel 4
i=1;
while i<nlocsx4
    Tcalcx4(i)= accelX4(locsx4(i+1),1)-accelX4(locsx4(i),1);
    i=i+1;
end
i=1;
nlocsy4 = length(locsy4);
while i<nlocsy4
    Tcalcy4(i)= accelY4(locsy4(i+1),1)-accelY4(locsy4(i),1);
```

Continuación del anexo 2.

```
i=i+1;
end
i=1;
nlocs = length(locsz4);
while i<nlocs4
    Tcalcz4(i)= accelZ4(locsz4(i+1),1)-accelZ4(locsz4(i),1);
    i=i+1;
end
%Encontrar periodo nivel 5
i=1;
while i<nlocsx5
    Tcalcx5(i)= accelX5(locsx5(i+1),1)-accelX5(locsx5(i),1);
    i=i+1;
end
i=1;
nlocsy5 = length(locsy5);
while i<nlocsy5
    Tcalcy5(i)= accelY5(locsy5(i+1),1)-accelY5(locsy5(i),1);
    i=i+1;
end
i=1;
nlocs = length(locsz5);
while i<nlocs5
    Tcalcz5(i)= accelZ5(locsz5(i+1),1)-accelZ5(locsz5(i),1);
    i=i+1;
end
%Encontrar periodo nivel 6
i=1;
while i<nlocsx6
    Tcalcx6(i)= accelX6(locsx6(i+1),1)-accelX6(locsx6(i),1);
    i=i+1;
end
i=1;
nlocsy6 = length(locsy6);
while i<nlocsy6
    Tcalcy6(i)= accelY6(locsy6(i+1),1)-accelY6(locsy6(i),1);
    i=i+1;
end
i=1;
nlocs = length(locsz6);
while i<nlocs6
    Tcalcz6(i)= accelZ6(locsz6(i+1),1)-accelZ6(locsz6(i),1);
    i=i+1;
end
%Encontrar periodo nivel 7
i=1;
while i<nlocsx7
    Tcalcx7(i)= accelX7(locsx7(i+1),1)-accelX7(locsx7(i),1);
    i=i+1;
end
i=1;
nlocsy7 = length(locsy7);
while i<nlocsy7
    Tcalcy7(i)= accelY7(locsy7(i+1),1)-accelY7(locsy7(i),1);
    i=i+1;
end
i=1;
nlocs = length(locsz7);
while i<nlocs7
    Tcalcz7(i)= accelZ7(locsz7(i+1),1)-accelZ7(locsz7(i),1);
```

Continuación del anexo 2.

```
i=i+1;
end

% calcular periodos y frecuencias
%%% Nivel 1
Tx1 = mean(Tcalcx1); % periodo en segundos
fx1 = 1/Tx1; % frecuencia en Hz
Ty1 = mean(Tcalcy1); % periodo en segundos
fy1 = 1/Ty1; % frecuencia en Hz
Tz1 = mean(Tcalcz1); % periodo en segundos
fz1 = 1/Tz1; % frecuencia en Hz
%%% Nivel 2
Tx2 = mean(Tcalcx2); % periodo en segundos
fx2 = 1/Tx2; % frecuencia en Hz
Ty2 = mean(Tcalcy2); % periodo en segundos
fy2 = 1/Ty2; % frecuencia en Hz
Tz2 = mean(Tcalcz2); % periodo en segundos
fz2 = 1/Tz2; % frecuencia en Hz
%%% Nivel 3
Tx3 = mean(Tcalcx3); % periodo en segundos
fx3 = 1/Tx3; % frecuencia en Hz
Ty3 = mean(Tcalcy3); % periodo en segundos
fy3 = 1/Ty3; % frecuencia en Hz
Tz3 = mean(Tcalcz3); % periodo en segundos
fz3 = 1/Tz3; % frecuencia en Hz
%%% Nivel 4
Tx4 = mean(Tcalcx4); % periodo en segundos
fx4 = 1/Tx4; % frecuencia en Hz
Ty4 = mean(Tcalcy4); % periodo en segundos
fy4 = 1/Ty4; % frecuencia en Hz
Tz4 = mean(Tcalcz4); % periodo en segundos
fz4 = 1/Tz4; % frecuencia en Hz
%%% Nivel 5
Tx5 = mean(Tcalcx5); % periodo en segundos
fx5 = 1/Tx5; % frecuencia en Hz
Ty5 = mean(Tcalcy5); % periodo en segundos
fy5 = 1/Ty5; % frecuencia en Hz
Tz5 = mean(Tcalcz5); % periodo en segundos
fz5 = 1/Tz5; % frecuencia en Hz
%%% Nivel 6
Tx6 = mean(Tcalcx6); % periodo en segundos
fx6 = 1/Tx6; % frecuencia en Hz
Ty6 = mean(Tcalcy6); % periodo en segundos
fy6 = 1/Ty6; % frecuencia en Hz
Tz6 = mean(Tcalcz6); % periodo en segundos
fz6 = 1/Tz6; % frecuencia en Hz
%%% Nivel 7
Tx7 = mean(Tcalcx7); % periodo en segundos
fx7 = 1/Tx7; % frecuencia en Hz
Ty7 = mean(Tcalcy7); % periodo en segundos
fy7 = 1/Ty7; % frecuencia en Hz
Tz7 = mean(Tcalcz7); % periodo en segundos
fz7 = 1/Tz7; % frecuencia en Hz

% Integrar y encontrar velocidades y desplazamientos
%%% Nivel1
```

Continuación del anexo 2.

```
vx1 = cumtrapz(accelX1(:,2),sgfX1)*100; %velocidad en x en cm/s
ux1 = cumtrapz(accelX1(:,2),vx1)*100; %Desplazamiento en x en cm
vy1 = cumtrapz(accelY1(:,2),sgfY1)*100; %velocidad en x en cm/s
uy1 = cumtrapz(accelY1(:,2),vy1)*100; %Desplazamiento en x en cm
%% Nivel2
vx2 = cumtrapz(accelX2(:,2),sgfX2)*100; %velocidad en x en cm/s
ux2 = cumtrapz(accelX2(:,2),vx2)*100; %Desplazamiento en x en cm
vy2 = cumtrapz(accelY2(:,2),sgfY2)*100; %velocidad en x en cm/s
uy2 = cumtrapz(accelY2(:,2),vy2)*100; %Desplazamiento en x en cm
%% Nivel3
vx3 = cumtrapz(accelX3(:,2),sgfX3)*100; %velocidad en x en cm/s
ux3 = cumtrapz(accelX3(:,2),vx3)*100; %Desplazamiento en x en cm
vy3 = cumtrapz(accelY3(:,2),sgfY3)*100; %velocidad en x en cm/s
uy3 = cumtrapz(accelY3(:,2),vy3)*100; %Desplazamiento en x en cm
%% Nivel4
vx4 = cumtrapz(accelX4(:,2),sgfX4)*100; %velocidad en x en cm/s
ux4 = cumtrapz(accelX4(:,2),vx4)*100; %Desplazamiento en x en cm
vy4 = cumtrapz(accelY4(:,2),sgfY4)*100; %velocidad en x en cm/s
uy4 = cumtrapz(accelY4(:,2),vy4)*100; %Desplazamiento en x en cm
%% Nivel5

vx5 = cumtrapz(accelX5(:,2),sgfX5)*100; %velocidad en x en cm/s
ux5 = cumtrapz(accelX5(:,2),vx5)*100; %Desplazamiento en x en cm
vy5 = cumtrapz(accelY5(:,2),sgfY5)*100; %velocidad en x en cm/s
uy5 = cumtrapz(accelY5(:,2),vy5)*100; %Desplazamiento en x en cm
%% Nivel6
vx6 = cumtrapz(accelX6(:,2),sgfX6)*100; %velocidad en x en cm/s
ux6 = cumtrapz(accelX6(:,2),vx6)*100; %Desplazamiento en x en cm
vy6 = cumtrapz(accelY6(:,2),sgfY6)*100; %velocidad en x en cm/s
uy6 = cumtrapz(accelY6(:,2),vy6)*100; %Desplazamiento en x en cm
%% Nivel7
vx7 = cumtrapz(accelX7(:,2),sgfX7)*100; %velocidad en x en cm/s
ux7 = cumtrapz(accelX7(:,2),vx7)*100; %Desplazamiento en x en cm
vy7 = cumtrapz(accelY7(:,2),sgfY7)*100; %velocidad en x en cm/s
uy7 = cumtrapz(accelY7(:,2),vy7)*100; %Desplazamiento en x en cm

Periodosx = [Tx1; Tx2; Tx3; Tx4; Tx5; Tx6; Tx7];
Periodosy = [Ty1; Ty2; Ty3; Ty4; Ty5; Ty6; Ty7];
Periodosz = [Tz1; Tz2; Tz3; Tz4; Tz5; Tz6; Tz7];

% data = table(Periodosx, Periodosy, Periodosz);
%
% set(handles.uitable1,'data');

Data=[Periodosx, Periodosy, Periodosz];
set(handles.uitable1,'data',Data);

n = length(accelX1(:,2));

switch val
case 1
```

Continuación de anexo 2.

```
for i=1:n
    %%% crear vector instantaneo de desplazamiento
    ut = [0, ux1(i), ux2(i), ux3(i), ux4(i), ux5(i), ux6(i), ux7(i)];
    %plot(ux1(i),0,'or','MarkerSize',5,'MarkerFaceColor','r')
    plot(ut,[0, 1, 2, 3, 4, 5, 6, 7],'-o','MarkerSize',5,'MarkerFaceColor','r')
    utotal= cat(1,ux1,ux2,ux3,ux4,ux5,ux6,ux7);
    axis([min(utotal) max(utotal) -1 8])
    grid on
    title('Desplazamiento de edificio por niveles')
    ylabel('Numero de Nivel')
    xlabel('Desplazamiento en cm')
    drawnow
    pause(0.5)
end
case 2
for i=1:n
    %%% crear vector instantaneo de desplazamiento
    ut = [0, uy1(i), uy2(i), uy3(i), uy4(i), uy5(i), uy6(i), uy7(i)];
    %plot(ux1(i),0,'or','MarkerSize',5,'MarkerFaceColor','r')
    plot(ut,[0, 1, 2, 3, 4, 5, 6, 7],'-o','MarkerSize',5,'MarkerFaceColor','r')
    utotal= cat(1,uy1,uy2,uy3,uy4,uy5,uy6,uy7);
    axis([min(utotal) max(utotal) -1 8])
    grid on
    title('Desplazamiento de edificio por niveles')
    ylabel('Numero de Nivel')
    xlabel('Desplazamiento en cm')
    drawnow
    pause(0.5)
end
end
% --- Executes when entered data in editable cell(s) in uitable1.
function uitable1_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  structure with the following fields (see MATLAB.UI.CONTROL.TABLE)
% Indices: row and column indices of the cell(s) edited
% PreviousData: previous data for the cell(s) edited
% EditData: string(s) entered by the user
% NewData: EditData or its converted form set on the Data property. Empty if Data was not changed
% Error: error string when failed to convert EditData to appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)
```

Fuente: Universidad EAFIT (2020). *Monitoreo de salud estructural de edificios*. Consultado el 12 de diciembre de 2021. Recuperado de <https://www.eafit.edu.co/innovacion/transferencia/Paginas/monitoreo-de-edificios.aspx#:~:text=El%20monitoreo%20y%20salud%20estructural,en%20la%20disminuci%C3%B3n%20de%20resistencia>.