

**PLATAFORMA DE CAPTURA, ALMACENAMIENTO, PROCESAMIENTO Y  
VISUALIZACIÓN DE DATOS DE UNA RED ACELEROMÉTRICA  
ORIENTADA AL MONITOREO DE SALUD ESTRUCTURAL**

**CAPTURE, STORAGE, PROCESSING AND VISUALIZATION DATA PLATFORM OF AN  
ACCELEROMETRIC NETWORK ORIENTED TO STRUCTURAL HEALTH MONITORING**

**MSc. Wilson Angarita\*, Dr Victor Aristizabal\*\*  
MSc Eder Acevedo\*\*\***

**\*Universidad Cooperativa de Colombia, Ingeniería y Tecnología  
TERMOTEC.**

Cra. 42 #49-95, Medellín, Antioquia, Colombia.  
(54) 4446065

wilson.angaritama@campusucc.edu.co, victor.aristizabalt@campusucc.edu.co,  
eder.acevedom@campusucc.edu.co

**Resumen:** El presente trabajo describe un prototipo de software desarrollado en el contexto del monitoreo de salud estructural, SHM por sus sigla en inglés Structure Health Monitoring, proceso que involucra la implementación de estrategias para la detección y caracterización de daños en estructuras, en este caso particular, el trabajo se centró en el desarrollo de una plataforma de captura, almacenamiento, procesamiento y visualización de datos de aceleración suministrados por una red de sensores de bajo costo tipo microelectromecánicos MEMS, como primer paso en la consecución de un sistema completo de SHM, dentro de las características principales que posee la plataforma es el uso de herramientas de software libre que permiten la reducción de costos de licenciamientos y el entorno de desarrollo web.

**Palabras clave:** *SHM, software libre, acelerómetros tipo MEMS, desarrollo de software*

**Abstract:** This paper describes a software prototype developed in the context of structural health monitoring, SHM for its acronym in English Structure Health Monitoring, a process that involves the implementation of strategies for the detection and characterization of damage to structures, in this particular case, The work focused on the development of a platform for capturing, storing, processing and visualization of acceleration data provided by a network of low-cost MEMS microelectromechanical sensors, as the first step in achieving a complete SHM system, within The main features of the platform is the use of free software tools that allow the reduction of licensing costs and the web development environment.

**Keywords:** *SHM, free software, MEMS type accelerometers, software development*

## 1. INTRODUCCIÓN

El monitoreo de salud estructural, SHM por sus siglas en Inglés (Structural Health Monitoring), es una técnica que permite controlar la seguridad, integridad y rendimiento de una estructura, permitiendo conocer en cada momento de la vida de la misma, el estado de los materiales que la constituyen y del conjunto completo de elementos que la conforman. Estos elementos y materiales pueden verse alterados por el envejecimiento debido al uso, por acciones medioambientales o por eventos accidentales tales como movimiento de tierra, terremotos o explosiones (Tuñón, D. 2016).

La implementación de esta técnica, requiere la interconexión de varios elementos, constituyéndose en un sistema que involucra etapas de adquisición de datos, la cual es responsable de establecer los diferentes sensores a utilizar en correspondencia con el tipo de variable a sensar tales como vibraciones, desplazamientos, humedad y temperatura, entre otras.

Otra de las etapas involucradas, es el acondicionamiento y transmisión de las señales entregadas por los sensores, algunos sensores involucran acondicionamiento de señal internamente por hardware, sin embargo, por lo general se hace necesario realizar otro tipo de acondicionamiento, como es, etapas de filtrado, amplificación y compensación de la respuesta del sensor debido a errores por linealización, entre otras.

Este acondicionamiento adicional a las señales entregadas por los sensores, se pueden realizar mediante el uso de hardware, como amplificadores de instrumentación, filtros activos y microcontroladores, o por medio de software directamente en la plataforma donde se desean visualizar los datos, herramientas como Matlab, Paython, LabView, entre otras, son comúnmente utilizadas. Tal es el caso de los trabajos desarrollado por Balcazar, R., & Francis, A. Chávez, R., Katherine, M., (2015) y JIMÉNEZ L., Cruz, A. & Marulanda, J. (2012), donde Matlab es la herramienta utilizada para el procesamiento de la información entregada por los sensores.

Otra de las etapas involucradas es la etapa de comunicación, encargada de llevar los datos desde el sitio de medición hasta la herramienta de visualización, tecnologías inalámbricas como wifi ,

ZigBee, Bluetooth y los protocolos de comunicación que las regulan son tenidos en cuenta en esta etapa.

Por último, la etapa de visualización, que involucra el almacenamiento de los datos y el análisis de los mismos, estableciendo comparaciones entre los datos obtenidos de los sensores y los modelos que caracterizan la estructura en estudio, para así establecer alarmas, acciones de mantenimiento y demás actividades de interés.

Esta organización por etapas de los elementos que constituyen un sistema de monitoreo estructural se representa en la **¡Error! No se encuentra el origen de la referencia..**



Fig1. Etapas en un sistema SHM

Esta organización por etapas involucra hardware y software y se sustenta en los trabajos consultados.

Algunos trabajos desarrollan las etapas completas y otros se centran en una en particular. En este trabajo, el objetivo general era el desarrollo de un prototipo de aplicación de software que permita el procesamiento, almacenamiento y visualización de los datos, que de acuerdo con la Fig1, correspondería a la etapa de aplicación.

Los adelantos tecnológicos al nivel de desarrollo de sensores, sistemas de comunicación, conectividad y procesamiento digital de señales, han permitido diseñar sistemas de monitoreo y control remoto, que han favorecido una de las potencialidades más importantes de los sistemas SHM, la cual consiste en reducir el riesgo de la estructura ante amenazas sísmicas y antrópicas, incrementando la confianza de los usuarios y ofreciendo datos reales para la

investigación y el desarrollo tecnológico (Jiménez, L. Cruz, A, & Marulanda , J. 2012)

En el trabajo de Garita, L. (2016), se presenta una revisión de enfoques de integración de información en áreas relacionadas con el monitoreo de la salud estructural, con énfasis en puentes. De acuerdo con el autor, el estudio realizado buscaba recopilar experiencias que eviten “reinventar la rueda”, así como identificar oportunidades de diferenciación, que sirvan de base para proponer enfoques novedosos de integración de la información en este campo. Este trabajo se realizó como parte del proyecto de investigación e-Bridge 2.0, del Instituto Tecnológico de Costa Rica, y que dentro de sus conclusiones y recomendaciones se destacan las apreciaciones en cuanto a la pertinencia del área de investigación y el amplio espectro que implica, además de los retos que se generan el manejar grandes volúmenes de información en tiempo real.

Este reto, da pie a pensar en la utilización de tecnologías computacionales relacionadas con administración de información tales como: Big data, computación en la nube, minería de datos, inteligencia de negocios, etc. Sin embargo, el autor recomienda que estas tecnologías potenciales sean evaluadas en función del alcance, recursos y calendario en su debido contexto, antes de ser consideradas para su implementación.

Uno de los trabajos consultados en la etapa de formulación de este proyecto, y que fue referente por abarcar la consecución completa de un sistema SHM fue (Tuñón, D. 2016), el cual planteaba como objetivo principal el desarrollo de un sistema de bajo costo para monitorizar ciertos parámetros físicos en una maqueta estructural, con el fin de realizar prácticas con los alumnos que permitan visualizar la respuesta de la estructura y realizar mediciones dinámicas de problemas de vibraciones. La Fig2 resume los componentes de hardware utilizados.

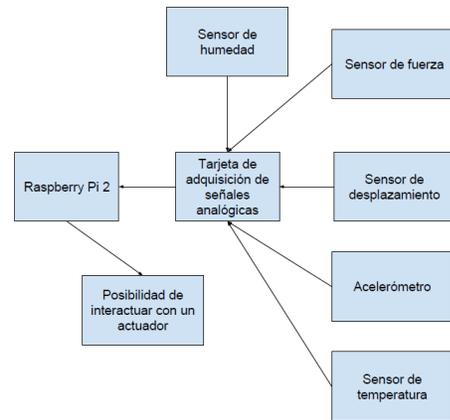


Fig2. Esquema introducción al hardware  
Fuente : (Tuñón Cabeza, 2016, pág. 41)

Esta solución involucró el uso de herramientas de software Java, JavaFX y Python para el procesamiento, y para el almacenamiento y visualización, MySQL y Glassfish respectivamente, todas ellas herramientas de uso libre, característica que es importante resaltar. La Fig 3 muestra de manera modular las herramientas incorporadas en el diseño.

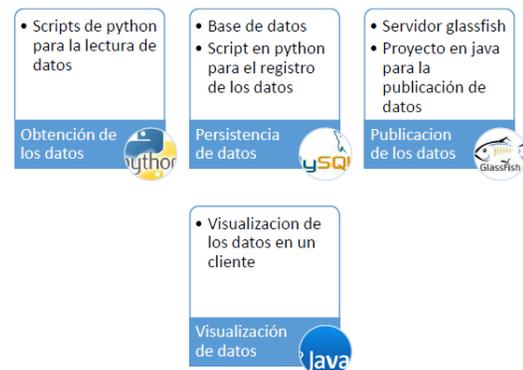


Fig 3. Esquema de capas de Software  
Fuente: (Tuñón Cabeza, 2016, pág.77)

En esta línea, el trabajo de Peralta et al. (2014), buscaba desarrollar e implementar herramientas analíticas y tecnológicas para la estimación del nivel de daño estructural de construcciones existentes en las regiones con mayor peligro sísmico de México. El sistema propuesto contempló el diseño e implementación de una plataforma integral de monitoreo estructural basada en redes de sensores acelerométricos tipo MEMs, conectados mediante dispositivos de hardware Arduino característicos por su bajo costo, capaces de proporcionar la información que a su vez se visualizaba de manera gráfica, incluyendo la

visualización en el tiempo de la aceleraciones en los eje x, y, z y análisis espectral de las mismas.

En el ámbito nacional, uno de los trabajos elaborados en esta línea de investigación, utilizados con referente fue el de la universidad del Valle, en el que los resultados del monitoreo permitieron realizar análisis y evaluación del efecto producido por la ocupación del edificio incluyendo caracterización de la influencia de las condiciones ambientales y análisis de identificación del comportamiento durante y después de dos sismos de mediana intensidad. (Jiménez L. et a 2012).

Como se ha descrito hasta el momento el SHM, implica dentro de una de sus etapas el desarrollo de aplicativos de software que permitan la captura, almacenamiento, procesamiento y visualización de datos, por tanto, los aspectos relacionados con el desarrollo de software tenían una gran importancia para la culminación del mismo.

En este orden de ideas a continuación se describe de manera general algunos aspectos propios del desarrollo de software que fueron tenidos en cuenta en la elaboración del proyecto.

Como primer paso, fue importante establecer las actividades que involucraba el desarrollo de software, las cuales definieron en gran medida la metodología utilizada en la consecución de los objetivos que planteaba el proyecto.

Para alcanzar un producto de software, se deben desarrollar una serie de actividades con resultados asociados en cada actividad, esto se conoce como proceso de Software. Algunas actividades fundamentales del proceso son: Especificación del Software, Desarrollo del software, Validación del Software y Evolución del Software. Por lo general, este proceso se representa a través un modelo, el cual no es más que una descripción simplificada del proceso, ejemplos de ellos son, modelo de flujo de trabajo, modelo de flujo de datos o de actividad y modelo de relación. Y a su vez, estos modelos en su gran mayoría se basan en los paradigmas del desarrollo de software, enfoque en cascada, desarrollo interactivo y desarrollo basados en componentes. (Sommerville, I. 2005).

En relación con este último concepto para el año 2001, el término ágil aparece en la escena del desarrollo de software, planteándose lo que se conoce como “el manifiesto ágil”, Se pretendía con esto ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados

por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas (Canos, 2012).

El desarrollo de este proyecto, no pretendió llevar con rigor una metodología en particular, pero si utilizó algunos conceptos inherentes como referente para la consecución del producto final, lo cual permitió obtener un prototipo de software que utilizó los aspectos que mejor se adaptaron a las necesidades del proyecto y al contexto en el cual se desarrolla.

Otro aspecto fundamental para en el desarrollo de software que se tuvo en cuenta en este trabajo fue, el levantamiento de los requerimientos.

Los requerimientos o requisitos de software por definición descrita en el glosario de terminología de ingeniería de software de la IEEE, son “una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo” y, “expresan las necesidades y limitaciones impuestas a un producto de software que contribuyen a la solución de algún problema en el mundo real” (Toro, A.2016).

El levantamiento de estos requisitos se constituye en una de las etapas más cruciales en un proyecto de software y comprende la definición de requisitos y la elaboración del modelo conceptual del sistema (Zapata, M. 2010).

En este mismo orden de ideas (Arias M. 2005), a través de los años se ha podido constatar que los requerimientos o requisitos son la pieza fundamental en un proyecto de desarrollo de software, ya que marcan el punto de partida para actividades como la planeación, básicamente en lo que se refiere a las estimaciones de tiempos y costos, así como la definición de recursos necesarios y la elaboración de cronogramas que son uno de los principales mecanismos de control con los que se cuenta durante la etapa de desarrollo.

La importancia de un buen establecimiento de requerimientos es que se constituyen en la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto, pues son el reflejo de las necesidades de los clientes o usuarios del sistema.

Para la documentación de los requerimientos se usó el estándar IEEE830, el cual ofrece una forma de organizar y presentar las Especificaciones de

Requisitos de Software (ERS), el formato planteado por el estándar IEEE 830 pese dentro de la misma guía para su elaboración el no obligar a que siga estrictamente la organización, si recomienda que se incluya de una u otra forma toda la información presentada en dicho estándar.

Otro aspecto del desarrollo de software que también se tomó como referente para la consecución del producto final, es la arquitectura de software ya que una arquitectura es fundamental para alcanzar tanto los requisitos funcionales, como los no funcionales además de favorecer las posibilidades de reuso, proporcionar planos para la construcción y posibilidad de evolución del sistema (Fernández, L. 2006).

El uso de conceptos y técnicas derivadas de la arquitectura de software permitió identificar los subsistemas que componen el sistema, las interfaces necesarias y las reglas de interacción entre ellos. Aunque la arquitectura de software se recomienda principalmente en el desarrollo de grandes proyectos, su utilización se sustenta en dos aspectos fundamentales, la articulación que debe tener la aplicación entre hardware y software, y la capacidad de reuso para versiones posteriores que integren más prestaciones, como por ejemplo, la incorporación de nuevos sensores e integración de aplicativos móviles que consuman recursos, entre otros.

## 2. PLATAFORMA DESARROLLADA

El aplicativo se instaló y configuró en un hosting, sobre un servidor con sistema operativo Linux, bajo un modelo de infraestructura que se observa en la Fig 4. Diagrama de Infraestructura y Redes del Aplicativo

Los usuarios administrador o invitados acceden a través de internet y las peticiones son recibidas en un datacenter por un router el cual hace la petición al firewall para poder ingresar a la red de área local, luego es enrutado por un Switch el cual lleva directamente al servidor de aplicaciones web en donde está instalado el sistema, éste se interconecta con los servidores de bases de datos relacionales de MySQL, que intervienen en los diferentes procedimientos, se realizan las acciones pertinente a cada proceso y se lleva a cabo la persistencia de los datos en los respectivos servidores virtualizados. De igual manera, los sensores instalados están apuntando a intermediador de

servicios en el esquema y transmiten sus datos a la base de datos principal.

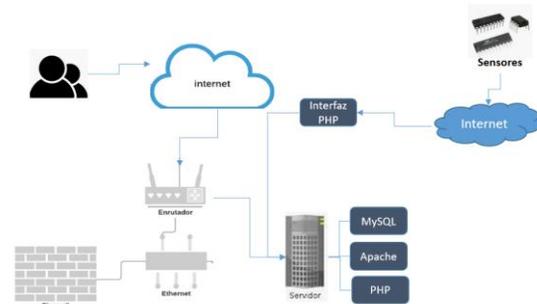


Fig 4. Diagrama de Infraestructura y Redes del Aplicativo

Las características del servidor se pueden observar en la Tabla 1 y Tabla 2 respectivamente.

Tabla 1. Información del Servidor

Artículo	Detalle
Paquete de alojamiento	M
Nombre del servidor	Servidor
cPanel Versión	82.0 (build 11)
Versión Apache	2.4.41
Versión PHP	7.2.21
Versión MySQL	10.2.26-MariaDB
Arquitectura	x86_64
Sistema operativo	linux
Dirección IP compartida	164.68.104.220
Ruta de acceso a Sendmail	/usr/sbin/sendmail
Ruta de acceso a Perl	/usr/bin/perl
Versión Perl	5.16.3
Versión Kernel	3.10.0-957.27.2.el7.x86_64

Tabla 2. Información del servicio

Servicio	Detalle	Estado
apache_php_fpm	up	ok
clamd	up	ok
cpanellogd	up	ok
cpdavd	up	ok
cphulkd	up	ok
cpsrzd	up	ok
crond	up	ok
dnsadmin	up	ok
exim (exim-4.92-2.cp1180.x86_64)	up	ok
ftpd	up	ok

httpd (2.4.41)	up	ok
imap	up	ok
ipaliases	up	ok
lmtpt	up	ok
mailman	up	ok
mysql (10.2.26-MariaDB)	up	ok
named	up	ok
nscd	up	ok
pop	up	ok
queueprocd	up	ok
rsyslogd	up	ok
spamd	up	ok
sshd	up	ok
Server Load	0.23	ok
CPU Count	6	ok
Memory Used	9.01%	ok
Swap	0.00%	ok
Disk / (/)	8%	ok
Disk /boot (/boot)	20%	ok
Disk /var/tmp (/var/tmp)	8%	ok

Fuente: Autor

La base de datos fue creada con la herramienta MySQL consta de cuatro tablas alarmas, sensores, datos y usuarios, en la Fig 5 se observa el modelo relacional de la misma.

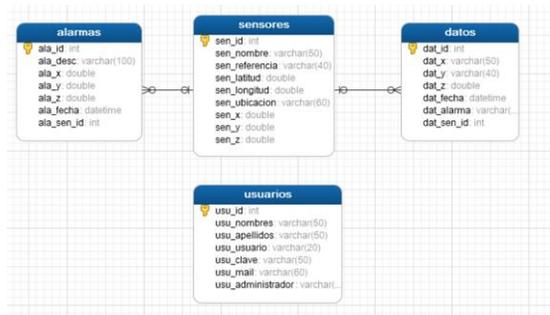


Fig 5. Modelo Relacional de la Base de Datos

Descripción de los campos de cada una de las tablas creadas en la base de datos.

**TABLA USUARIOS:** En la tabla usuarios, se ingresan los datos de las personas que van a tener acceso al sistema y a sus funcionalidades, cuenta con 7 campos: *usu\_id* (identificador único), *usu\_nombres* (nombre completo de la personas), *usu\_apellidos* (apellidos de la persona), *usu\_usuario* (usuario para ingresar), *usu\_clave*

(clave de ingreso), *usu\_mail* (correo electrónico), *usu\_administrador* (SI o No)

**TABLA SENSORES** En la tabla sensores, se aloja la información de cada uno de los sensores que transmiten la información. Cuenta con 9 campos: *sen\_id*(identificador único de la tabla), *sen\_nombre*(nombre del sensor), *sen\_referencia*(marca, o referencia), *sen\_latitud*(latitud), *sen\_longitud*(longitud), *sen\_ubicacion*(descripción del sitio en donde se encuentra el sensor), *sen\_x*(máximo valor que puede tomar X), *sen\_y*(máximo valor que puede tomar Y), *sen\_z*(máximo valor que puede tomar Z)

**TABLA DATOS:** En la tabla datos, se guardan los registros de datos que generan los sensores en tiempo real. Cuenta con 7 campos: *dat\_id*(identificador único de la tabla), *dat\_x*(valor de X), *dat\_y*(valor de y), *dat\_z*(valor de z), *dat\_fecha*(fecha y hora del registro), *dat\_alarma*(SI o NO), *dat\_sen\_id*(referencia del id del sensor)

**TABLA ALARMAS:** En la tabla alarmas se guardan los registros que superan los límites de los datos configurados para x, y, z en los sensores, al momento de detectar un registro que supere el límite, se crea un nuevo registro en esta tabla. Cuenta con 7 campos: *ala\_id*(identificador único de la tabla), *ala\_desc*(descripción de la alarma, si fue en x, y o z), *ala\_x*(valor de alarma en x), *ala\_y*(valor de alarma en y), *ala\_z*(valor de alarma en z), *ala\_fecha*(fecha-hora del suceso), *ala\_sen\_id*(referencia del id del sensor)

La estructura de paquetes del prototipo de software se observa en la Fig 6.

Inicia con la carpeta CSS, en la cual se puede encontrar las hojas de estilo utilizadas para armonizar las etiquetas HTML de los formularios.

En la carpeta IMG, se guardan todas las imágenes usadas para este proyecto.

En LIB, se encuentran algunas librerías usadas para el descargue de archivos en Excel, para los informes de alarmas y de los datos.

En la carpeta scriptBD, se encuentra el script SQL de la estructura y registros de las diferentes tablas

de la base de datos, esto en caso que se debe hacer una restauración.

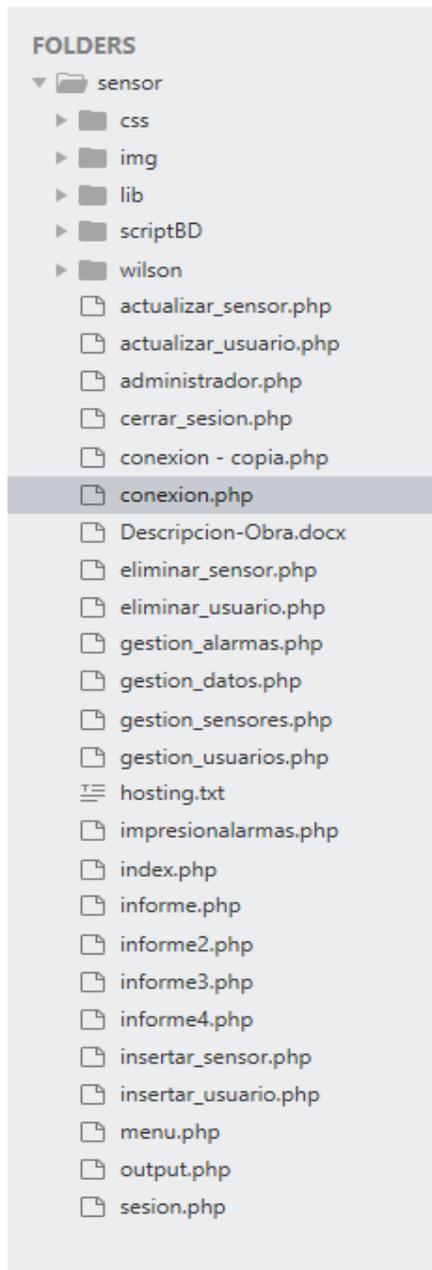


Fig 6. Estructura de paquetes (Carpetas)

En la carpeta Wilson, se encuentra el script en PHP que sirve para insertar los datos provenientes de los sensores, es la interfaz entre el sensor y la base de datos del sistema, esta interfaz se creó como prueba para el prototipo de software con el fin de probarlo con datos reales de aceleración medidos por sensores acelerométricos tipo micro electromecánicos y utilizando comunicación

inalámbrica a través de un microcontrolador de bajo costo, este desarrollo implicó el uso del sensor MPU6050 y un NODE MCU8266 que incluye microcontrolador 32-bit RISC CPU: Tensilica Xtensa LX106 a 80 MHz , con 64 KiB de RAM para instrucciones y 96 KiB de RAM para datos, soporta protocolo IEEE 802.11 b/g/n Wi-Fi que permite conexión WiFi y se programa a través del IDE arduino.

En la raíz del proyecto se encuentran todos los archivos concernientes al sistemas, tienen nombres relacionados con la información que es tratada en cada uno de ellos.

En cuanto a la tecnología y software utilizado la a continuación se describen cada una de las herramientas de desarrollo utilizadas.

**MySQL:** Se utilizó como motor de base de datos para almacenar la información, se configuró en entorno de pruebas y de producción. Mysql se usó en su versión libre, es uno de los gestores de bases de datos más usados en proyectos web en el mundo.

**XAMPP:** Para configurar el entorno de desarrollo y programación, se usó, XAMPP, está versión para el sistema operativo Windows, trae consigo herramientas como Apache, PHP, MariaDB o MySQL y Perl. Al usar este software, el entorno se configura automáticamente con el solo hecho de instalarlo. Es una herramienta para entornos de pruebas.

**Chart.js:** Es una librería desarrollada bajo el paradigma OpenSource, que sirve para realizar gráficas de diferentes tipos, está hecha con HTML 5 y Javascript, sus desarrolladores tienen las fuentes en la nube para ser usadas por los desarrolladores.

**Bootstrap:** Son un conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Mediante estas librerías se realizaron las interfaces, botones, y se personalizaron los diferentes aspectos de diseño de los formularios y etiquetas que contienen HTML 5.

**PHP:** “PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML”. [www.php.net](http://www.php.net)

Sublime Text Esta es una herramienta o editor de código, facilita al programador la escritura de los algoritmos, ya que autocompleta y señala los errores de programación.

A continuación se muestran las ventanas principales del prototipo de software desarrollado.



Fig 7. Ventana de ingreso

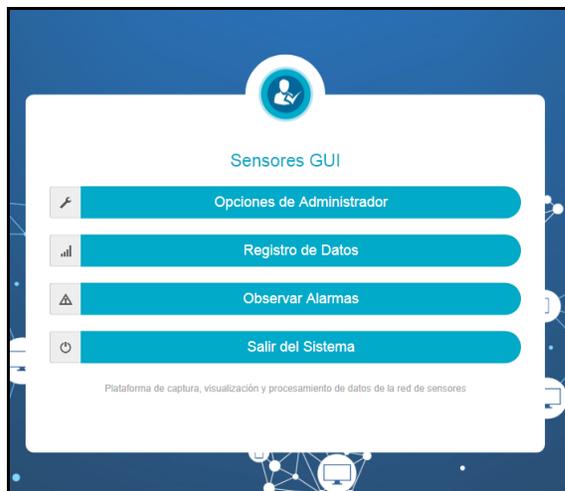


Fig 8. Menú principal

En este caso la figura 8, muestra el menú que se despliega cuando el usuario se registra como administrador, en el caso de invitado el menú no visualiza el botón opciones de administrador.



Fig 9. Menú opciones de Administrador.



Fig 10. Registro de datos de sensores

Esta ventana permite visualizar los datos registrados en la base de datos provenientes de los sensores, realizar filtrado por fecha y hora y descargar el reporte en un archivo xls para un posterior análisis con herramientas como matlab que permitan un análisis espectral de las vibraciones por ejemplo, además, incluye los botones que dan acceso a la visualización de las gráficas de aceleración, velocidad y desplazamiento para cada uno de los ejes, x,y,z.

En general se logró la consecución de una plataforma de captura, almacenamiento, procesamiento y visualización de datos proveniente de sensores acelerométricos tipo MEMS, que cumplió con los requerimientos establecidos siendo el más importante la utilización de herramientas de libre uso y en un entorno web que puede ser

consultado desde cualquier dispositivo conectado a internet.



Logs de Alarmas de los Sensores

Id	Sensor	Descripción	Fecha-Hora	Máximo en X	Alarma X	Máximo en Y	Alarma Y	Máximo en Z	Alarma Z
142	SENSOR03	ALARMA EN Y	2019-06-06 11:20:00	2	0.877	3	8	4	0.88
140	SENSOR01	ALARMA EN Z	2019-06-06 10:40:00	1	0.9247	1.99	1	0.989	2
141	SENSOR03	ALARMA EN Z	2019-06-06 10:40:00	2	0.99	3	0.5	4	5
139	SENSOR01	ALARMA EN Y	2019-06-06 10:00:00	1	0.8147	1.99	3	0.989	0.9449
138	SENSOR01	ALARMA EN X, Y, Z	2019-06-06 09:40:00	1	3	1.99	2.89	0.989	1.844

Plataforma de captura, visualización y procesamiento de datos de la red de sensores

Fig 11. Registro de alarmas de los sensores

### 3. CONCLUSIONES

El desarrollo de este tipo de plataformas contribuye notablemente al fortalecimiento de los grupos de investigación de las instituciones de educación superior, ya que abre un amplio espectro de aplicaciones IoT en un número igual de áreas del conocimiento, ya que lo que la concepción de la misma es adaptable a las diferentes aplicaciones que se aborden, sólo tendría que tenerse en cuenta el tipo de dato capturado por los sensores, temperatura, humedad, CO<sub>2</sub>, radiación, velocidad, etc, pero la concepción como tal de la plataforma sería adaptable.

Si bien es cierto que en el contexto de aplicaciones IoT se cuenta con servicios en línea disponibles en la nube algunas con versiones limitadas de acceso gratuito y versiones con mayores prestaciones pagas tales como Ubidots, thinger.io, carriots, Microsoft azure IoT hub, entre otras, el desarrollo de plataformas propias como la desarrollada en este trabajo, permiten además de tener una aplicación a la medida, el control total en la gestión de los sensores incluyendo los protocolos de comunicación y algo fundamental en esta época de la información, el control de los datos registrados, para nadie es un secreto que uno de los recursos más valiosos en la actualidad es la información.

Una de las pruebas de validación de la plataforma consistió en conectar un sensor MPU6050 por medio de un módulo MCU8266, para ello se configuro un protocolo de comunicación entre el microcontrolador y la base de datos de la plataforma realizada en MySQL, mediante la

implementación del método POST a través de solicitudes HTML, si bien, funciona, es recomendable revisar el uso de otros protocolos, tales como el MQTT (Message Queue Telemetry Transport), el cual requiere menos información para transmitir datos lo que mejoraría los tiempos de latencia, vital para ampliaciones que requieren transmitir datos a frecuencias más elevadas.

### REFERENCIAS

Arias Chaves, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, VI(10). Recuperado de <http://www.redalyc.org/resumen.oa?id=66612870011>

Balcazar, R., Francis, A., & (primero). (2015). Diseño e implementación de un sistema de monitoreo inalámbrico de parámetros de estructuras civiles. Pontificia Universidad Católica del Perú. Recuperado de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/5981>

Canós, J. H., & Letelier, M. C. P. P. (2012). *Métodologías Ágiles en el Desarrollo de Software*. Recuperado de <http://10.22.1.21:8080/jspui/handle/123456789/476>

Fernández, L. (2006) arquitectura de software revista software Guru Ao 02 N° 03 Mayo-Junio 2006 Pag 40-42. Recuperado de [http://www.ozarate.net/articulos/arquitectura\\_sw\\_s\\_g\\_2006.pdf](http://www.ozarate.net/articulos/arquitectura_sw_s_g_2006.pdf)

Garita, L (2016). Enfoques de integración de información para sistemas de monitoreo de salud estructural de puentes | *Revista Tecnología en Marcha*. Recuperado 5 de agosto de 2017, de [http://revistas.tec.ac.cr/index.php/tec\\_marcha/articloe/view/2542/2331](http://revistas.tec.ac.cr/index.php/tec_marcha/articloe/view/2542/2331)

IEEE Standard Glossary of Software Engineering Terminology," in ANSI/ IEEE Std 729-1983 , vol., no., pp.1-40, 18 Feb. 1983 doi: 10.1109/IEEESTD.1983.7435207

keywords: {glossaries;IEEE standards;software

engineering;ANSI/IEEE Std 729-1983;IEEE standard glossary;software engineering terminology;IEEE Standards;ANSI Standards;Terminology;Software engineering}, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7435207&isnumber=7435206>

Jimenez, L., Cruz, A., & Marulanda, J. (2012). Desarrollo e implementación de un sistema de monitoreo estructural en tiempo real para un edificio en escala real. Recuperado de <http://jornadasaie.org.ar/wp-content/themes/jornadas-aie-2016/jornadas-aie-antecedentes/2012/contenidos/trabajos/42.pdf>

Peralta, L. M. R., Hernández, E. I., Cardeña, S. A., Moreno, A. Y. L. H., & Hernández, J. A. H. (s. f.). Sistema de monitoreo estructural de bajo costo basado en redes de sensores inalámbricos. Recuperado de <http://saver-buildings.com/publicaciones/SistemaMonitoreo.pdf>

Sommerville I (2005). Ingeniería del Software, Madrid, España 7Ed, Pearson educación S.A.

Toro, A., & Peláez, L.E.. (2016). Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira. Entre Ciencia e Ingeniería, 10(20), 117-123. Retrieved June 27, 2019, from [http://www.scielo.org.co/scielo.php?script=sci\\_artext&pid=S1909-83672016000200016&lng=en&tlng=es](http://www.scielo.org.co/scielo.php?script=sci_artext&pid=S1909-83672016000200016&lng=en&tlng=es).

### SITIOS WEB

Chart.js. Gráficos JavaScript simples pero flexibles para diseñadores y desarrolladores. <https://www.chartjs.org> (Consultado Junio 2019)