



CARÁTULA DEL PROYECTO FINAL

SEDE: ITR Norte

Proyecto final de Carrera:



POSGRADO EN ROBÓTICA
E INTELIGENCIA ARTIFICIAL

Nombre del trabajo: "Sistema de código abierto para la predicción de fallas en motores de corriente alterna por vibración"

Autor: Juan Pedro de León Sum

Fecha: 02/12/2021

Tutor: Prof. Dr. Eder Mateus Gonçalves

Resumen

El presente proyecto tuvo como finalidad un sistema enfocado al mantenimiento predictivo de motores de corriente alterna analizando el aumento de vibraciones y temperaturas, que sea de bajo coste, basado y distribuido en código abierto.

En base a lo citado por [Tabarés Gutiérrez, 2019c] y [Tabarés Gutiérrez, 2018], el fenómeno de la industria 4.0 se gana su nombre y popularidad en Europa en el año 2010, junto a ese crecimiento se genera una cultura de “Movimiento Maker”. En dicho artículo se explora como surge el modelo de “Fabricación Abierta” y como el mismo afecta en la inclusión de la ciudadanía en los nuevos desarrollos. Como toma forma y se extiende por Europa el proyecto OpenMaker que tiene como objetivo crear un ecosistema que permita la transformación y la colaboración que fomente las innovaciones dentro del sector manufacturero. Basándose en el paradigma de la fabricación abierta.

En las búsquedas para este trabajo, no se encontró una iniciativa con al mismo alcance, en América Latina, existiendo proyectos aislados, que pretenden generar un entorno de software libre. Para ello nos gustaría nombrar un software desarrollado en Brasil bajo esa consigna de un HMI con amplios protocolos de comunicación llamada ScadaBR.PONTES [2021]; ScadaBR [2021]

Durante toda la evolución de dicho proyecto, se analizaron las opciones disponibles para cada paso, pero el criterio de elección contó como limitante que las herramientas utilizadas sean de código abierto. Dicho trabajo fue dividido en tres partes. Primeramente, se desarrolló el hardware y el firmware que fuera de bajo coste y de fácil obtención en el mercado local.

En segundo lugar, se desarrolló un sistema de captura de estos datos, tratamiento y almacenamiento de los mismos en una base de datos SQL. En tercer lugar, se realizó programa el algoritmo de Inteligencia Artificial basado en LSTM que analiza los datos recibidos y en base a ellos predice su comportamiento futuro pudiendo determinar cuando ocurrirá la falla del motor. También dentro de lo que determinamos como mantenimiento, tenemos que centrarnos en el punto de la calidad del mismo para ob-

tener la “Máxima disponibilidad al mínimo coste”. García Garrido [2003] , página 109. Al revisar los puntos de trabajos necesarios para realizar un mantenimiento de calidad tenemos algunos que según su característica abarcan lo que trataremos en este trabajo. García Garrido [2003] en las paginas 189 y 190, son 10 puntos de interés los que permiten un mantenimiento de calidad.

Con esto llegamos a que la red LSTM puede proveer una preposición razonable en el contexto de las pruebas realizadas, por lo que es posible un mantenimiento predictivo con dicho sistema.

Palabras clave: Mantenimiento predictivo, Redes LSTM, Open source, openmaker.

Índice de figuras

1.	Métodos analíticos en mantenimiento predictivo - Fujitsu [2019]	3
2.	Rango de vibraciones - ISO-20816 [2016]	11
3.	Esquema de una neurona	13
4.	Red neural del tipo Perceptron.	14
5.	Diferencia entre redes neuronales.Garzón [2018]	15
6.	Comportamiento de una red LSTM.Christopher [2015]	15
7.	Comportamiento de una red LSTM.Christopher [2015]	16
8.	Esquema de envío de datos entre todos los dispositivos	17
9.	Esquema Eléctrico del colector de datos	19
10.	Caja a donde se fijara el sensor al motor	21
11.	Motor monofásico	22
12.	Motor trifásico, con dispositivo colector conectado(AZUL)	22
13.	Caja con todos los dispositivos conectados	26
14.	Formato de la tabla SQL que recibo los datos	26
15.	Formato de la tabla SQL que recibo los datos	27
16.	Dispositivo que induce vibraciones de manera paulatina al motor	29
17.	Incremento de la vibración mediante dispositivo – 4500 registros	31
18.	Incremento de la vibración mediante dispositivo – 4500 registros	32
19.	Gráfica comparativa entre reales y predicciones: Fuente tabla 7	34
20.	Predicción basada en el segundo set de datos	36
21.	Predicción basada en el segundo set de datos	37

Índice

Índice de figuras	III
1. Introducción	1
2. Objetivos	5
2.1. Metas	5
3. Revisión bibliográfica y Trabajos relacionados	7
4. Sistema adquisición de datos y predicción de fallas en motores de corriente alterna.	17
5. Experimentos y Resultados	23
6. Conclusiones	37
7. Trabajos Futuros	38

1. Introducción

El mantenimiento es un conjunto de técnicas que tienen como objetivo conservar tanto las instalaciones como los equipos durante la mayor cantidad de tiempo posible. Desde el comienzo de la revolución industrial en donde las maquinas comenzaron a tomar protagonismo en las trasformaciones económicas el mantenimiento ha pasado por varias trasformaciones. Desde un principio en cuanto los mismos operarios realizaban el mantenimiento hasta la creación de centro de manteniendo especializados en esa tarea. García Garrido [2003]

Dentro de estos puntos, la mayoría depende directamente del departamento de mantenimiento y su organización. Pero si con este trabajo se consigue generar e informar el estado de los motores y su predicción de falla, se pueden entre otras cosas, trabajar sobre los puntos 8, 9 y 10 de la siguiente lista, evitando paradas no programadas, o provisoras por falta de materiales adecuados.

1. Que dispongamos de mano de obra en la cantidad suficiente y con el nivel de organización necesario.
2. Que la mano de obra esté suficientemente cualificada para acometer las tareas que sea necesario llevar a cabo.
3. Que el rendimiento de dicha mano de obra sea lo más alto posible.
4. Que dispongamos de los útiles y herramientas más adecuadas para los equipos que hay que atender.
5. Que los materiales que se empleen en mantenimiento cumplan los requisitos necesarios.
6. Que el dinero gastado en materiales y repuestos sea el más bajo posible.
7. Que se disponga de los métodos de trabajo más adecuados para acometer las tareas de mantenimiento.
8. Que las reparaciones que se efectúen sean fiables, es decir, no vuelvan a pro-

ducirse en un largo periodo de tiempo.

9. Que las paradas que se produzcan en los equipos como consecuencia de averías o intervenciones programadas no afecten al Plan de Producción y, por tanto, no afecten a nuestros clientes (externos o internos).
10. Que dispongamos de información útil y fiable sobre la evolución del mantenimiento que nos permita tomar decisiones.

Esto se resume a tener información fiable que permita verificar la evolución de las máquinas y permita actuar en consecuencia, fijando dentro del plazo identificado como aceptable la intervención del equipo que presenta un comportamiento anómalo.

El mantenimiento predictivo no es algo nuevo en la industria, es un proceso que se viene gestando desde los años 60 con el surgimiento de los primeros computadores y que comenzó a popularizarse en los años 80 con el surgimiento de los computadores personales. Tavares [1999]

El Mantenimiento predictivo es una técnica que permite anticiparnos a posibles incidencias, fallas o errores en activos y procesos de producción a través de diferentes tecnologías. Fujitsu [2019] en la pagina 11 y García Garrido [2003] en la página 17.

Otro autor que trata en profundidad esta forma de mantenimiento es Gonzáles Fernández, en su capítulo 5 se refiere a lo que el llama la **"Nueva Mantecnologia"**. *El mantenimiento predictivo debe entenderse como aquella metodología que basa las intervenciones en la maquina o instalación sobre la que se aplica, en la evolución de una determinada variable que sea realmente identificadora de su funcionamiento y fácil de medir* Gonzáles Fernández en la página 139.

Dicho autor también detalla sobre el mantenimiento predictivo basado en el análisis de vibraciones y aunque en su libro esta nombrada como parámetro referencia la ISO 10816, que era la valida en el momento de la redacción del mismo, la que utilizaremos es la 20816, que es la evolución de esta norma al año 2020.

En los tiempos actuales de Industria 4.0, donde las fabricas tienen dispositivos co-

nectados, aplicarle capas de conocimiento al mantenimiento comienza a tener un peso económico importante en la evolución de una empresa. La disponibilidad de tecnologías de Internet de las cosas, o IoT como es conocido, el descenso en el costo de poder de procesamiento generando capacidades de Inteligencia Artificial facilitan el análisis y la monitorización de los activos y entornos de los procesos. Fujitsu [2019] En la **figura 1** vemos una tabla de Métodos analíticos en mantenimiento predictivo de la página 11 de Fujitsu [2019].

Métodos analíticos en mantenimiento predictivo			
	PERIODO DE TIEMPO ANALIZADO	IMPACTO EN EL NEGOCIO	METODOLOGÍA
PASADO	ANÁLISIS DE FALLOS HISTÓRICOS	Prevenir incidencias similares o fallos para asegurar una recuperación rápida	ANÁLISIS DE INCIDENCIAS • Comprensión de las causas de fallo Data Mining, statistical analysis
PRESENTE	PREDICCIÓN DE EVENTOS PRESENTES O FUTUROS	Evitar incidencias y fallos similares; Asegurar una recuperación rápida.	PREDICCIÓN DE FALLO • Reconocimiento de síntomas o probabilidad de fallo/ incidencia Análisis de anomalías
FUTURO	PREDICCIÓN DE FUTUROS EVENTOS O ANOMALÍAS.	Prevenir incidencias y fallos futuros para prevenir los costes de mantenimiento	PLANIFICACIÓN MANTENIMIENTO • Predicción de fallo o avería con antelación Survival Analysis

Figura 1: Métodos analíticos en mantenimiento predictivo - Fujitsu [2019]

La gestión del mantenimiento predictivo trata de obtener el conocimiento en tiempo real del estado de equipos o instalaciones por medio del conocimiento de valores de determinadas variables que representan su estado en un momento dado. Estas técnicas pueden denominarse modelos de Alta Disponibilidad, porque tienen niveles de disponibilidad por encima del 90 %. Como ejemplo de estos equipos tenemos turbinas de producción eléctrica, hornos de alta temperatura y equipos rotativos que trabajan de forma continua. García Garrido [2003] Parte de estos últimos son los que serán tratados en este trabajo.

Paralelo a las pautas de ingeniería propuestas hasta el momento, también tenemos que describir un punto importante de este trabajo y que motivo la creación del mismo,

la falta de herramientas que sean de libre distribución y circulación como un todo. En el estado del arte actual y de carácter propietario, existen en el mercado varias opciones que realizan lo que se propone en este trabajo, pero en los casos citados son todas de carácter propietario y restrictivo que no le permiten a la empresa manipular a su antojo la información o generar una nueva característica para el ajuste a su realidad.

En este contexto citamos por ejemplo Weg Motor Scan de la empresa WEG WEG [2019a] y WEG [2019b], que tiene su sede principal en Brasil y por otro lado tenemos también a ABB son su ABB Smart Sensor ABB [2021] que se encargan del mantenimiento predictivo en motores.

Por eso en este trabajo definimos que todos los esquemas, software y características sean publicados bajo la consigna de Software Libre siguiendo los puntos de Open Source Initiative. Free Software Foundation [2007] en su versión 1.9.

Que de sus varios puntos destacamos los 3 primeros que son:

1. La libre distribución.
2. La inclusión del código fuente.
3. La posibilidad de permitir modificaciones y trabajos derivados

Una vez definidos los conceptos que vamos a abordar en este trabajo y la importancia que tienen los mismos en la industria actual se definen los objetivos principales que abordaremos en este trabajo

En el artículo de Timmer et al. [2007] que trata sobre la eficiencia de motores, se refieren precisamente a lo que mejora y o deteriora la eficiencia de un motor, según dicho autor tanto la temperatura como las vibraciones de un motor pueden ofrecer un bajo rendimiento eléctrico y por lo tanto aumentar los costos. Por lo que mantener un sensor que informe a al departamento de mantenimiento de esas variaciones puede ofrecer un panorama de eficiencia energética, además lo que será tratado en el de correr de este artículo.

2. Objetivos

Generar un sistema de código libre que permita que se conozca el estado de un motor en tiempo real pero que además pueda predecir el tiempo de funcionamiento del mismo, para que en los casos que permita, los encargados de mantenimiento tengan la información necesaria para estipular la detección de la maquina asociada a dicho motor en un periodo de tiempo que sea más beneficioso para la organización. Al mismo tiempo permitir que si existe interés y conocimiento, otros interesados puedan mejorar, adaptar o ampliar las funcionalidades de lo que aquí comienza como un trabajo del posgrado de Robótica e Inteligencia Artificial.

2.1. Metas

1. Crear una plataforma de lectura de datos de vibración y temperatura en motores.
 - a) Revisión de los dispositivos que envíen datos por red inalámbrica con la finalidad de cablear lo menos posible dentro de una planta ya en funcionamiento. Que dispositivos se encuentran en el mercado regional y el costo de los mismos.
 - b) Realizar consultas sobre los acelerómetros y sensores de temperatura. Siempre priorizando los dispositivos que se encuentren en la región.
 - c) Tratar de que el dispositivo resultante dentro de los límites técnicos propios, sea del menor tamaño posible.
 - d) La importación de productos no puede ser una limitante para el despliegue del sistema, se deben conseguir en Uruguay.
 - e) Desarrollar dos o más prototipos en PCB perforadas genéricas para realizar testeo por periodos de tiempo cortos con el fin analizar la viabilidad de los

mismos.

2. Crear una plataforma móvil con Raspberry Pi
 - a) Que tenga instalada una base de datos SQL , Un servidor de páginas Web para la captura de los datos enviados desde cada dispositivo inalámbrico
 - b) Que la misma pueda proveer corriente a los dispositivos si fuera necesario.
3. Crear un equipo para el procesamiento de datos con Inteligencia Artificial.
 - a) Crean una interfaz de usuario que permita el monitoreo y notificaciones del estado de los motores controlados.
 - b) Generar un módulo de Inteligencia Artificial que pueda proveer en base a los datos del pasado el comportamiento futuro de cada máquina.
4. No utilizar ningún software de licencia restrictiva que trunque el posible desarrollo futuro de la solución.

En primer lugar, definimos como la plataforma ESP-8266 en un encapsulado llamado Wemos D1 que además del micro controlador ya tiene interface de conexión USB directo al equipo. También posee librerías para que pueda ser programado desde el Ide de Arduino. Dicho micro controlador es de bajo coste, ya viene con conexión inalámbrica como centralizador de los componentes el : ESP8266 [2013], este es un controlador de acceso simple en el mercado y fiabilidad razonable. Mehta [2013]. Con esto cumplimos lo que se indicó en los objetivos del trabajo, que necesitábamos un contexto de conexión sin cables.

Como hardware base para la colocación de los sensores de vibración en nuestro caso será el MPU-6050 que incorpora un acelerómetro de 3 ejes, además de tener una tasa de resolución aceptable ya que detectan pequeños cambios en las vibraciones ya citado en: Castaño Mejía [2021] en la página 79. Para la temperatura utilizamos el LM35 que cumple con los requerimientos para nuestro trabajo según lo citado por: de S. Silva et al. [2014] , aunque en la bibliografía citada existen distintos modelos, la decisión final recayó en que dispositivos que se consiguen en Uruguay y que el costo del prototipo

sea bajo. En segundo lugar, se instalará una Raspberry Pi con Debian Linux y MariaDB como SQL, dicho equipo será el encargado de recibir los datos de todos los sensores instalados en su entorno.

En tercer lugar, se desarrollará dentro de un servidor con arquitectura x86 una interfaz de usuario que permita monitorear todos los datos obtenidos de todos los sensores y mediante el desarrollo de una Inteligencia Artificial intentará predecir el tiempo disponible ante una falla en un motor por vibración.

3. Revisión bibliográfica y Trabajos relacionados

Nos es difícil imaginar en una fábrica actual la ausencia de motores eléctricos para la realización de la tarea productiva para cada empresa. Por ende, los motores tienden a ser la fuerza bruta que provee de movimiento cintas, piezas de corte, ventilación y muchas otras formas de ejercer el trabajo necesario. En su libro *Introducción al Análisis de Vibraciones*, White [2010] además de detallar la importancia del cuidado y análisis de los parámetros vibratorios de un motor da pautas de cómo y cuándo se deben tomar las medidas que desarrollaremos en este trabajo. “Todas las máquinas deben ser probadas totalmente calentadas. La temperatura de la máquina afectará la alineación y los juegos en operación debido a la expansión termal. Una máquina fría tendrá una firma de vibración diferente de una máquina caliente y esas pueden a veces ser totalmente diferentes” White [2010] en la página 65. También especifica que se deben guardar tendencias o firmas como registro para comparar a futuro si existe una tendencia al crecimiento de dicha vibración, lo que indicaría un problema incipiente en dicho equipo White [2010] en la página 75. De estas palabras llegamos al consenso que tanto la temperatura como la vibración de un motor son importantes. Ahora bien, tenemos que estudiar cómo podremos mejorar dicho proceso de captura de datos.

Ahora hablando exclusivamente de la temperatura del mismo existen algunos fenómenos eléctricos que también producen aumento de temperatura en un motor y con esto

su deterioro potencial, según (Morales Juan 2018) sobre el análisis de temperatura por medio de termografía, concluye que la temperatura en un motor y el crecimiento de la misma lleva a fallas, aunque nuestro trabajo no utiliza termografía, resumiremos que la termografía es “ la tecnología que no requiere contacto y se fundamenta en la medición de longitudes de onda infrarrojas para comprobar temperaturas desde una distancia segura.”Montero and Francisco [2018] en la página 21, en nuestro caso medimos el incremento de la misma sobre el chasis del motor, y será por medio de un equipo para este fin. Con el advenimiento de los nuevos procesadores y el alto poder computacional a costo relativamente bajo, el gran desarrollo del estado del arte de la Inteligencia Artificial, la creación en los últimos años de las computadoras de tamaño reducido como Raspberry Pi y los micro-controladores programables de código abierto como Arduino ha llevado a estudiantes y entusiastas a generar muchos conocimientos en esta área. Ahora centrándonos en el objetivo de nuestro trabajo, que es la predicción de anomalías, en base a vibraciones, el estado del arte en este campo es bastante documentado, teniendo varios trabajos desde ambientes académicos como industriales, como enunciaremos en los párrafos a continuación.

La vibración de un motor está relacionada con su movimiento armónico. En términos generales, la causa del aumento en la vibración reside en problemas mecánicos, como: desalineaciones en los acoplamientos, rodamientos gastados o dañados, falta de sujeción. La amplitud de la vibración indica cuan grave puede ser la falla lo que refleja el estado de trabajo de la misma Moreno García et al. [2015] en la página 4. Para el caso detallado se utilizó un acelerómetro MMA7361, que es muy similar al que utilizamos en nuestro trabajo, pero se descartó, por no encontrarse en Uruguay.

En su trabajo utilizan LabVIEW, un software propietario de “National Instruments” que tiene un valor de licenciamiento de U\$S 3925 por licencia o una opción de U\$S 500 anuales. Instruments [2021]. Como conclusión en su trabajo indican que si es posible detectar la falla por vibraciones con dicho componente. Dejan claro también que para

que el sistema funcione adecuadamente es sumamente importar el conocimiento del analista en la interpretación de los datos obtenidos.

También dentro de la línea de investigación tenemos una solución propietaria de algunas empresas como WEG de Brasil, con su producto Weg MotorScan, WEG [2019b] y ABB con su ABB Smart Sensor ABB [2021], en ambos los casos, el dispositivo es de tamaño muy reducido, totalmente inalámbrico, ya que no necesita cables de datos ni de alimentación. En ambas soluciones se tienen aplicaciones para celular tanto Android como IOS que permite visualizar los datos en tiempo real. También los datos pueden ser consultados desde la nube de la empresa. Entre sus características de funcionamiento destacan el diagnóstico de vibración, medición de temperatura en superficie, tiempo de funcionamiento, intervalos de lubricación de rodamientos, y conexiones a temperaturas entre -40 a 80 grados Celsius.

Ahora bien, sus fortalezas desde el punto de vista que encaramos este trabajo pasa a ser un punto importante para lo que pensamos desarrollar. Los datos de la aplicación no quedan en propiedad de la empresa, ya que los mismos son enviados a la nube de Weg Motor Scan o de ABB Smart Scan, y dichas paginas cuentan con la inteligencia para indicar la detección de fallas, no es posible mantener un histórico de datos prolongado, ya que la propia aplicación indica que los datos permanecen 3 meses en los servidores. La forma de contrato con Weg es mensual algo similar a la de ABB que es anual, y hay que adquirir también los equipos que funcionan de nexo entre sus sensores y la red. Lo que hace que la empresa contrata dependa siempre del pago de dicha cuota para tener disponible su información.

Si nos centramos en el análisis de dichos datos de vibración para la predicción de fallas existe también una bibliografía muy extensa y trabajos muy relevantes. Utilizaremos como punto de partida en este trabajo los resultados obtenidos por Scalabrini Sampaio et al. [2019] en su artículo: "Prediction of Motor Failure Time Using An Artificial Neural Network", que analiza varias técnicas para la predicción de fallas. En dicho trabajo desglosan de manera muy detallada las técnicas utilizadas y los resultados obtenidos

en cada una de ellas. Puntualmente seleccionaron cuatro técnicas distintas de predicción: Artificial Neural Network, Regression Tree, Random Forest and Support Vector Machine Scalabrini Sampaio et al. [2019]

Se detalla un modelo de análisis de vibraciones que utilizan un acelerómetro MMA8452Q y un Arduino Uno para la lectura de los datos, para nuestro caso en particular utilizaremos el MPU-6050 por encontrarse de manera más fácil en Uruguay. Según Castaño Mejía [2021] trabaja en un artículo similar para pequeños motores de corriente continua en su mayoría, su trabajo como algunos otros detallados en la bibliografía D.S. et al. [2015] implementan estos sensores como una alternativa de bajo costo y calidad aceptable, por lo que nos parece adecuado utilizar dicho equipo para la predicción.

Visto esto deberemos marcar que es lo aceptable para nuestro trabajo, una vez detectadas las vibraciones lo que tomaremos como valores fuera de lo normal. Para esto, nos centraremos en una norma ISO 20816-1 de 2016 ISO-20816 [2016] “Mechanical vibration — Measurement and evaluation of machine vibration”, que especifica los valores máximos admitidos según ciertas categorías de motores. En dicha norma también se especifica en el capítulo 6.2, que existen una amplia gama de factores que se deben tener en cuenta sobre los criterios de vibración.

Entre ellos, ubicación del sensor, tipo de rodamientos, velocidad de rotación, masa, y otros valores, debido a la gran cantidad de factores, dicho esto tomaremos como valores aceptables para el funcionamiento del motor los valores indicados en la **figura 2**, obtenida de dicha norma ISO 20816-1:2016 en ella se pueden ver las especificaciones según el tipo de maquinaria y cuanto sería la máxima vibración para cada caso.

Las zonas quedan definidas para permitir una evaluación cualitativa de las vibraciones, esto datos siempre deben ser tomados con la maquina en estado de trabajo, y a velocidad de operación.

Otro punto a tomar en cuenta dentro de cada rango es que, a motores pequeños, menores a 15 Kw los límites de vibración deberían estar siempre más próximos del nivel

inferior, los mayores tendrían un espectro más alto de vibración y esto se deberá considerar como normal. ISO-20816 [2016]

Niveles de vibracion según ISO 20816				
Velocidad RMS mm/s	Grupo 1	Grupo 2	Grupo 3	Grupo 4
0.28	Bueno	Bueno	Bueno	Bueno
0.45				
0.71				
1.12	Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
1.80				
2.80	Aceptable	Aceptable	Aceptable	Aceptable
4.50				
7.10	Inaceptable	Inaceptable	Inaceptable	Inaceptable
11.20				
18.00				
28.00				
45.00				

Figura 2: Rango de vibraciones - ISO-20816 [2016]

Zonas de trabajo: por grupo de maquinas según ISO-20816 [2016]

- **Grupo 1:** Partes individuales de motores y maquinas, íntegramente conectadas a la maquina completa, en motores de hasta 15kw.
- **Grupo 2:** Maquinas de tamaño medio , normalmente entre 15kw y 75Kw sin apoyos especiales. Hasta 300Kw
- **Grupo 3:** Grandes maquinas motrices y otras grandes maquinas de masas rotativas.
- **Grupo 4:** Grandes maquinas motrices y otras grandes maquinas de masas rotativas, con un conjunto de salida mayor a 10MW

Condiciones de trabajo según ISO-20816 [2016]

- **Bueno :** la vibración de las máquinas recién puestas en servicio normalmente

cae dentro de esta zona.

- **Satisfactorio** : Las máquinas con vibración dentro de esta zona normalmente se consideran aceptables para Operación a largo plazo.
- **Aceptable** : Las máquinas con vibración dentro de esta zona normalmente se consideran insatisfactorias a largo plazo. Generalmente, la máquina puede operarse por un período limitado en esta condición hasta que surja una oportunidad adecuada para tomar medidas correctivas.
- **Inacceptable** : los valores de vibración dentro de esta zona normalmente se consideran lo suficientemente severos como para causar Daños a la máquina.

Como nota, en la figura 2, se indica específicamente que estos valores deben ser tomados como un punto de referencia y que la experiencia puede llevar a que existan algunas diferencias.ISO-20816 [2016]

La inteligencia artificial es un nuevo enfoque para resolver problemas dentro del campo de la computación. Como es citado en el artículo de Christopher [2015] y Matich [2001] haciendo un recorrido histórico por los inicios, ya desde 1956 Johan McCarthy acuñó dicha expresión para hacer referencia a programas de cómputos inteligentes. En 1957 Frank Rosenblatt comenzó el desarrollo de la red Perceptron el primer tipo de red neuronal que se utiliza hasta nuestros días. Con el advenimiento de los nuevos computadores y los poderes de cálculos asociados a ellos durante los años siguientes las mismas fueron evolucionando y aumentando su complejidad. Las redes neurales son un conglomerado de neuronas, un ejemplo simple de una neurona sería como se muestra en la **figura 3**.

De esto se desprende que esta neurona recibe 4 entradas de E1 a E4, de donde W1 a W4 son los pesos para cada una de esas entradas. El Bias es número que permite un ajuste de nivel, que permite transformar toda la sumatoria en un nivel más alto o más bajo, dependiendo de la necesidad.

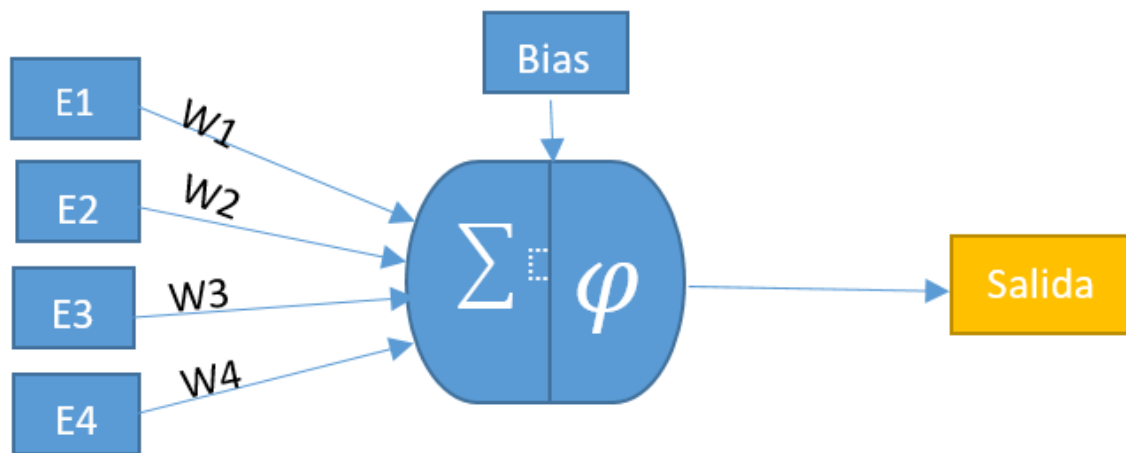


Figura 3: Esquema de una neurona

La función ϕ es denominada función de activación es generalmente una función no lineal que lleva a una saturación dependiendo de las salidas de la red. De las más cuales nombramos, por ejemplo, Función lineal, sigmoidea y tangente hiperbólica. Matich [2001]

A partir de este concepto podemos tener una idea de cuál es la representación de una red neural. Una neurona en particular no tiene ninguna función específica y ella por sí sola no puede llegar a ningún resultado. Una red neural es un conjunto de neuronas que se interconecta entre sí, de los cuales existen muchas actualmente.

Como se nombró anteriormente “El Perceptron” fue el primer intento de una red neural funcional. “Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.” Matich [2001] en la página 8. En la **figura 4** vemos como es una red del tipo Perceptron:

Este tipo de arquitectura Perceptron, tiene algunas particularidades, por ejemplo, ninguna conexión vuelve a su conexión anterior, y tampoco existen conexiones entre neuronas de la misma capa. Esto es importante para algunos trabajos, pero en el nuestro en particular se deberán aplicar otro tipo de redes neurales.

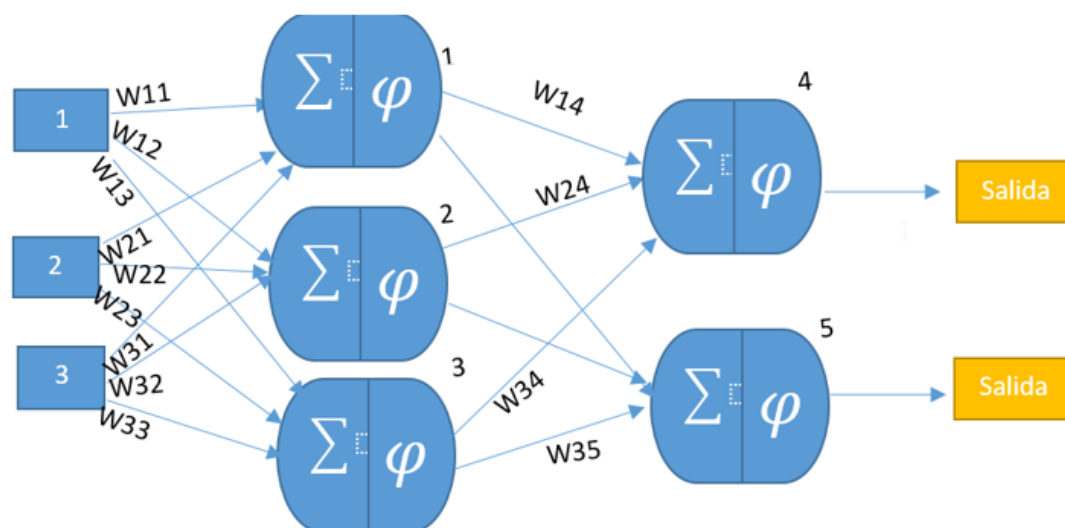


Figura 4: Red neural del tipo Perceptron.

Las predicciones en series de tiempo, es un problema conocido y al mismo tiempo su predicción es difícil para las redes clásicas, ya que no se basan en una regresión común y si tienen una dependencia de los valores ya obtenidos en la propia secuencia. Para solventar esta dificultad surgen las redes neurales recurrentes o RNN a diferencia de las anteriores las RNN, son un tipo distinto de red neural, la diferencia principal en dichas redes es que tiene bucles al contrario de lo que sucede con las Perceptron por ejemplo, esto le permite a la red que la información persista una cierta cantidad de tiempo.

Para dejar más claro el concepto que abordaremos en este trabajo y las diferencias entre las redes Perceptron, las redes neurales recurrentes (RNN) y las LSTM, anexamos el esquema de la **figura 5**, en esta figura vemos como las redes del tipo LSTM, aumentan su complejidad, con respecto a las demás.

Una red neural recurrente puede considerarse como una multiplicidad de copias de la misma red, en la cual cada una de ellas pasa la información a su sucesor. Como se indica en la **figura 6**.

Estos bucles como se muestran hacen que las redes LSTM pueden considerarse co-

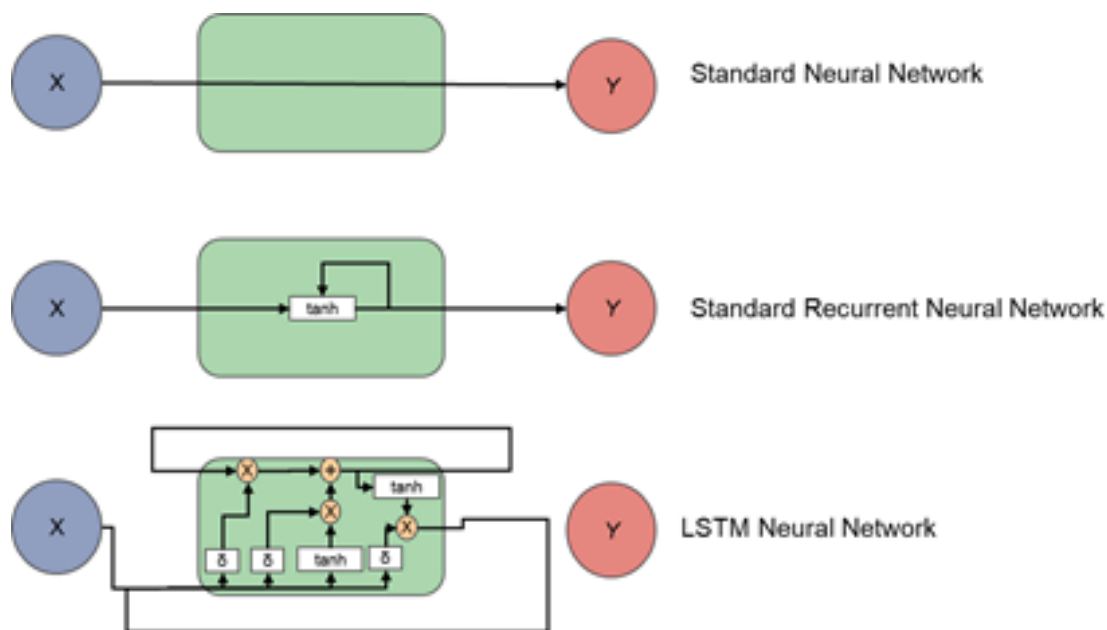


Figura 5: Diferencia entre redes neuronales. Garzón [2018]

mo múltiples copias de la misma red, .Esta naturaleza en cadena revela que las redes neuronales recurrentes están íntimamente relacionadas con secuencias y listas. Son la arquitectura natural de la red neuronal que se utiliza para esos datos.”Christopher [2015].

En la **figura 6**, un trozo de red neuronal, **A**, mira alguna entrada x_t y genera un valor h_t . Un bucle permite que la información pase de un paso de la red al siguiente.

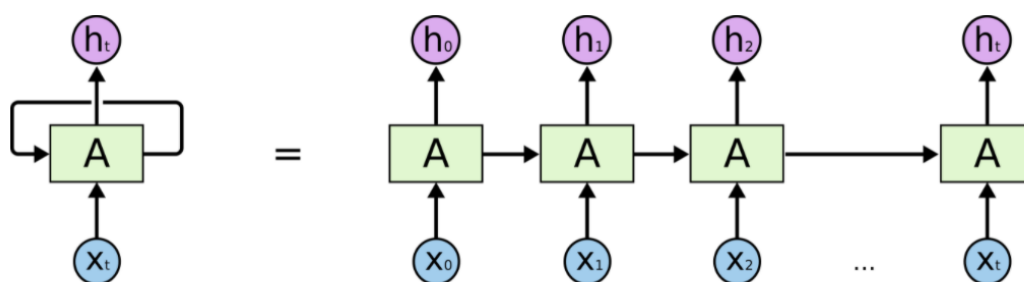


Figura 6: Comportamiento de una red LSTM. Christopher [2015]

Según lo citado por De Castilla [2019] y AbdElRahman et al. [2017] en sus respecti-

vos artículos sobre detección de anomalías por vibraciones aplicadas a motores, detallan como las redes LSTM son utilizadas para este tipo de predicción. Las LSTM o Long-Short Term Memory, son redes neurales que una de sus características principales es que la información puede persistir durante un tiempo dentro de la red. Fueron introducidas por Hochreiter & Schmidhuber en el año 1997. Las LSTM recuerdan información durante largos periodos de tiempo ya que ese es su comportamiento predeterminado. En la **figura 7** vemos como es el módulo de una red LSTM dicho módulo de repetición en un LSTM contiene capas que interactúan entre si. El esquema en principio parece complejo, pero al desmenuzar en partes su comportamiento, queda mas simple su entendimiento. Cada uno de las tres celdas que se muestran en el la **figura 7**, representa una celda de la red LSTM. La red tiene la capacidad de eliminar o agregar información a cada estado de la celda, regulada por unas estructuras llamadas puertas.

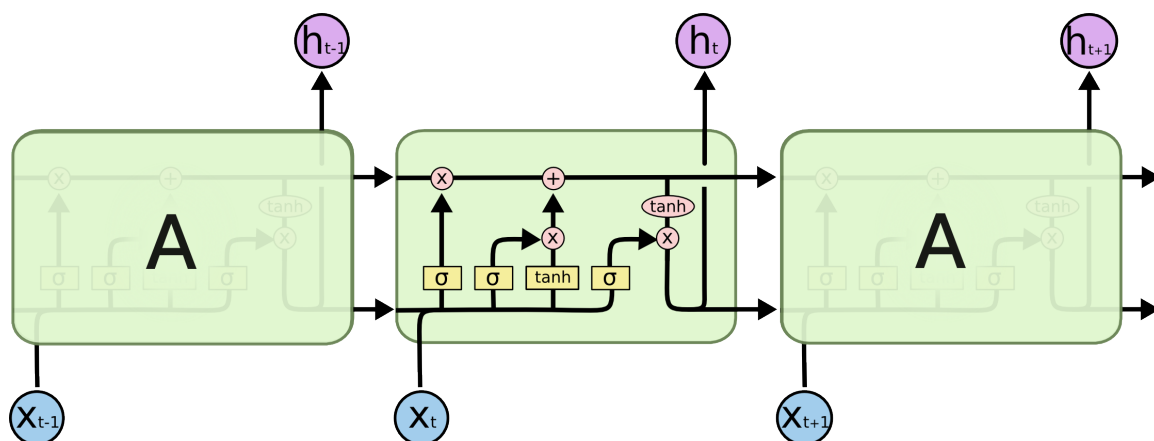


Figura 7: Comportamiento de una red LSTM. Christopher [2015]

Describiendo de manera simple nuestra LSTM:

- La LSTM decide que información vamos eliminar/olvidar o mantener
- Luego la red debe decidir que nueva información almacena en el estado de la celda.
- Se actualiza el estado anterior. correctivas.
- Se repite nuevamente todo el ciclo

El mantenimiento predictivo es también un ámbito que se recopilan informaciones a lo largo del tiempo para el monitoreo del estado de los activos, con la finalidad de encontrar patrones de comportamiento que permitan prever fallas en el futuro. Este tipo de red es especialmente atractiva para ese dominio debido a que son muy buenas de aprender de secuencias de tiempo “Este hecho se presta a que sus aplicaciones utilicen datos de series de tiempo al hacer posible mirar hacia atrás durante períodos de tiempo más largos para detectar patrones de falla”.Fidan Boylu [2019].

Los datos en las redes LSTM están relacionados entre si, haciendo que la generación de un dato, tenga influencia en la generación del próximo. Algo que no sucede en la red de clasificación, por ejemplo, que no existe relación de precedencia entre un valor predicho y el próximo.

4. Sistema adquisición de datos y predicción de fallas en motores de corriente alterna.

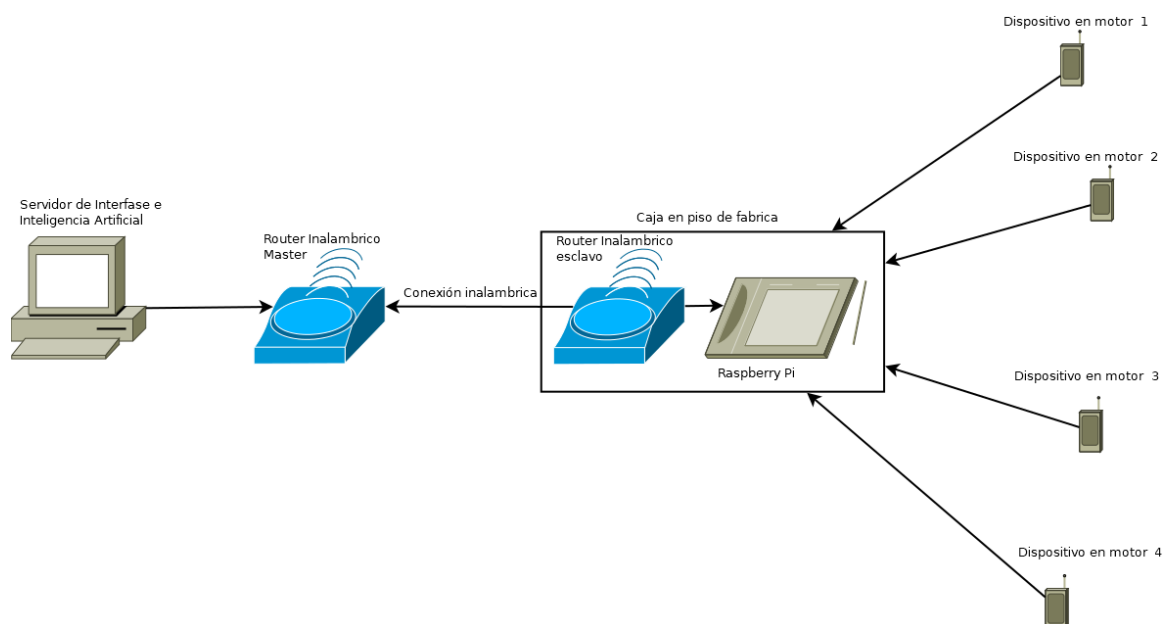


Figura 8: Esquema de envío de datos entre todos los dispositivos

En la **figura 8** se muestra la arquitectura general de trabajo para detallar como quedan funcionando todos los dispositivos tanto en las pruebas realizadas en este trabajo con un único dispositivo, como en un futuro, ya se desarrolló la plataforma para que pueda ser expandida. De esta manera es posible acoplar más sensores a un nodo central, teniendo como limitante la capacidad del receptor de datos, que de momento queda fuera del alcance de este trabajo.

Sobre la vibración y la temperatura, tomaremos dos vías distintas, ya que las mismas pueden variar según los entornos de trabajo. Para los datos de vibración, debemos definir lo que entendemos como un comportamiento fuera de rango en un motor, por lo que nos respaldaremos en los que pautan las normas internacionales ISO 20816-1:2016 sobre las vibraciones máximas admitidas para cada tipo de motor.

Para lograr todos los objetivos planteados en este trabajo se utilizará un enfoque mayormente práctico, con esto planteamos tener un modelo físico totalmente funcional. En base a lo citado en la fundamentación teórica, comenzamos con el detalle de lo construido Cumpliendo con las metas del punto uno de este trabajo, se eligieron los dispositivos:

1. Micro controlador Esp8266 – Encapsulado en una PCB llamado Wemos mini D1, con un chip ESP-12E. Este micro controlador será el encargado de recibir los datos del sensor de vibraciones y del sensor de temperatura, y mediante la red inalámbrica envía los datos al dispositivo que está en el piso de fábrica, que en el caso de este proyecto es una Raspberry Pi 3.
2. Para en analizador de vibraciones utilizamos el acelerómetro de TDK MPU-6050 encapsulado en una PCB llamado GY-521.
3. El sensor de temperatura utilizamos el LM35 de Texas Instruments en un encapsulado TO92
4. Se utilizarán 3 diodos Led de distintos colores para generar las indicaciones al

usuario del funcionamiento del dispositivo conector de datos, los mismos son “Encendido” de color verde, “Conexión con la red” de color azul, y “Error de Envío” de color rojo.

En su operación normal el sensor mantiene tanto la luz verde como la azul encendida.

Para el diseño del esquema utilizamos el Software KiCad KiCad [2020] que es licenciado bajo la licencia GNU GPL v3. En la **figura 8** está detallado el esquema eléctrico de armado de la placa.

Esquema Wemos – Mpu6050 – Lm35

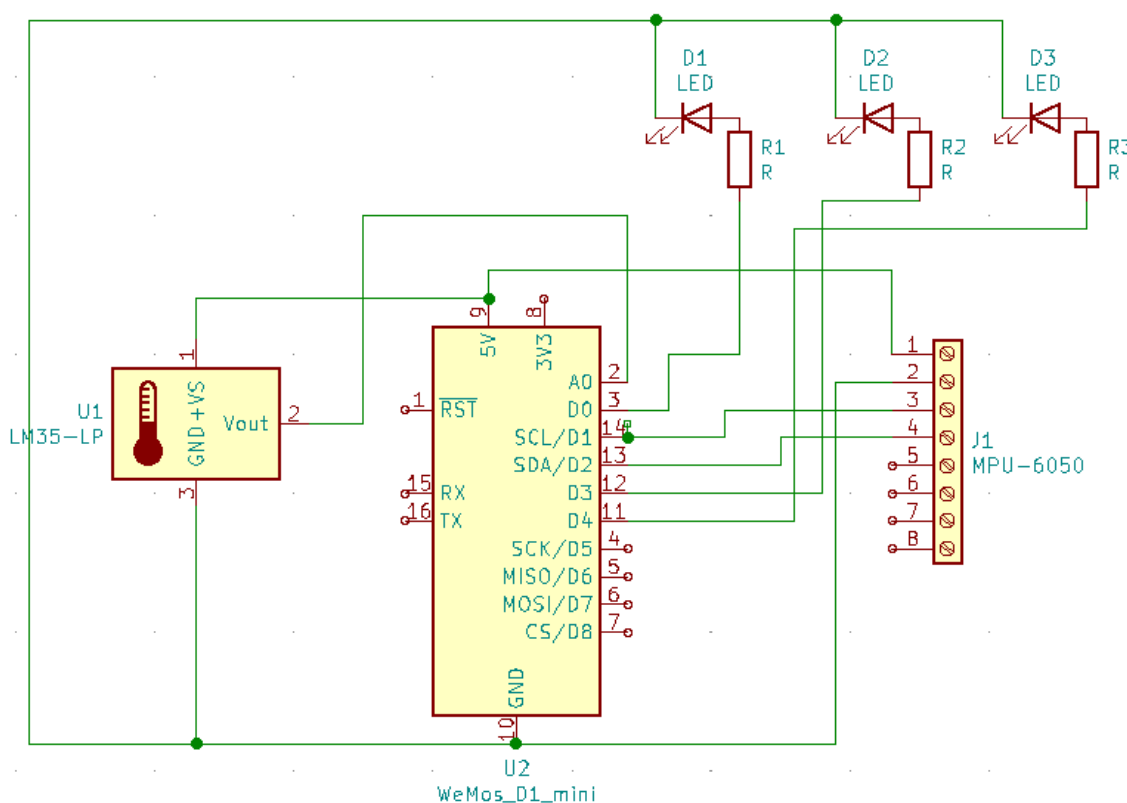


Figura 9: Esquema Eléctrico del colector de datos

Para la primera versión del dispositivo de recolección de datos, se utilizó una placa Pcb perforada genérica. Una vez soldados todos los componentes pasamos a tener una placa con la apariencia de la **tabla 1** están anexadas dos imágenes del dispositivo

colector.

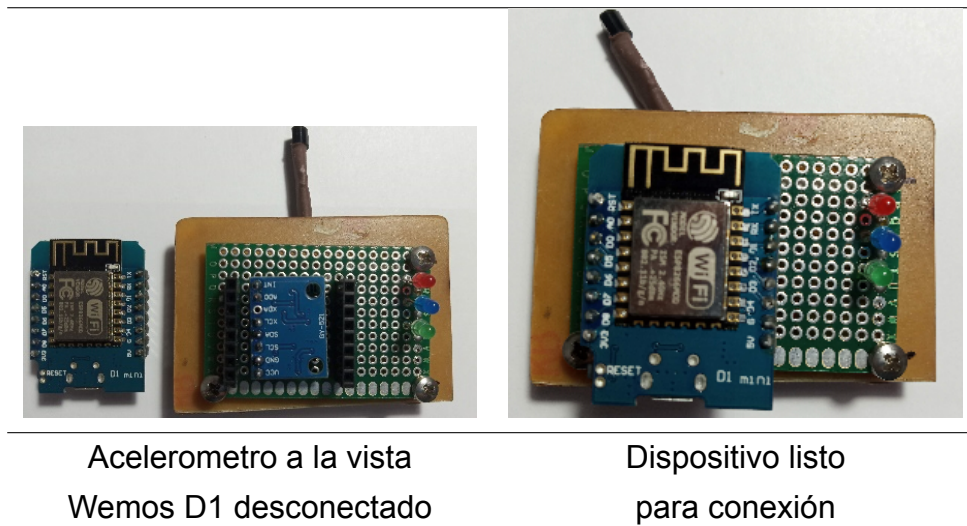


Tabla 1: Imágenes del dispositivo colector de datos

Para facilitar el trabajo y la colocación del dispositivo en el motor, se realiza el diseño de una pequeña caja que facilite la colocación en los dos motores de prueba como se muestra en la **figura 10**. Y se pasa a imprimir en una impresora 3D

Sobre el firmware que se le instala al dispositivo, el mismo fue desarrollado en el Ide de Arduino, utilizando las librerías adecuadas para el trabajo con el Wemos y el acelerómetro MPU-6050.

Otro factor que se debe tener en consideración que ya se debe solucionar es la forma de identificar cada uno de los posibles dispositivos, esta funcionalidad deja de forma independiente la programación de todos los dispositivos, para eso aprovecharemos la conexión inalámbrica del Wemos y por ende su dirección física o su Mac Address:” es un identificador de 48 bits (6 bloques de dos caracteres hexadecimales [8 bits]) que corresponde de forma única a una tarjeta o dispositivo de red. Se la conoce también como dirección física, y es única para cada dispositivo.”Wikipedia [2021].

De esta manera cada dispositivo se anunciará a sí mismo.

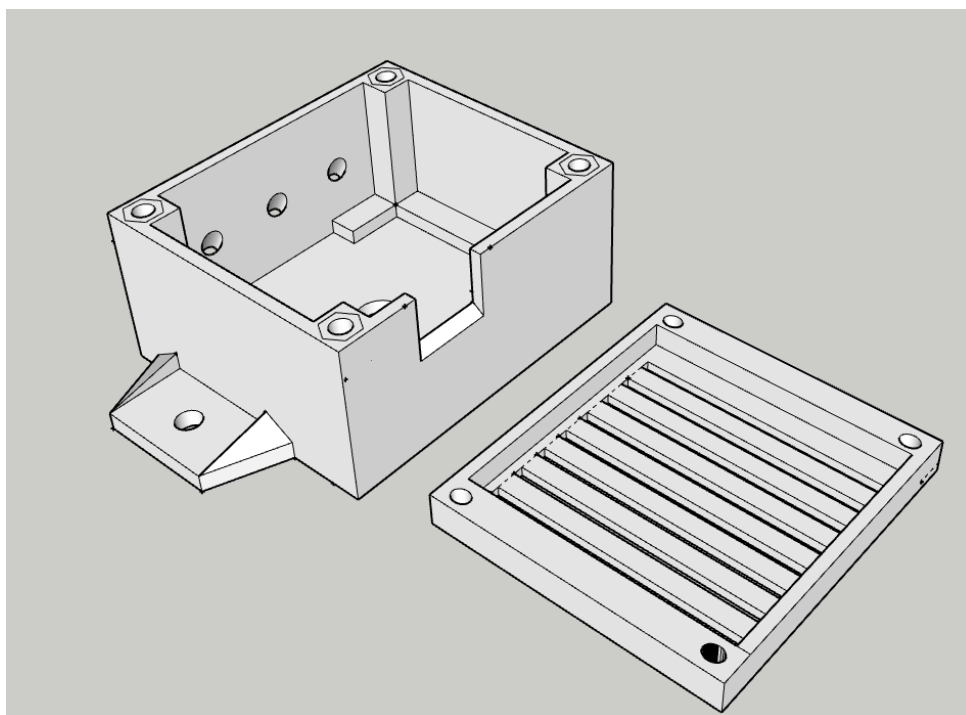


Figura 10: Caja a donde se fijara el sensor al motor

Pasando este primer paso de armado del dispositivo colector utilizamos una bancada de pruebas con dos motores de corriente alterna, uno monofásico **figura 11** y luego otro trifásico **figura 12**. Sus características de funcionamiento están en la **tabla 2**

Datos	Motor Monofásico	Motor Trifásico
Voltaje	220	380
Wats	550	1100
Hz	50	50
RPM	2840	1450

Tabla 2: Valores experimentales de la práctica X

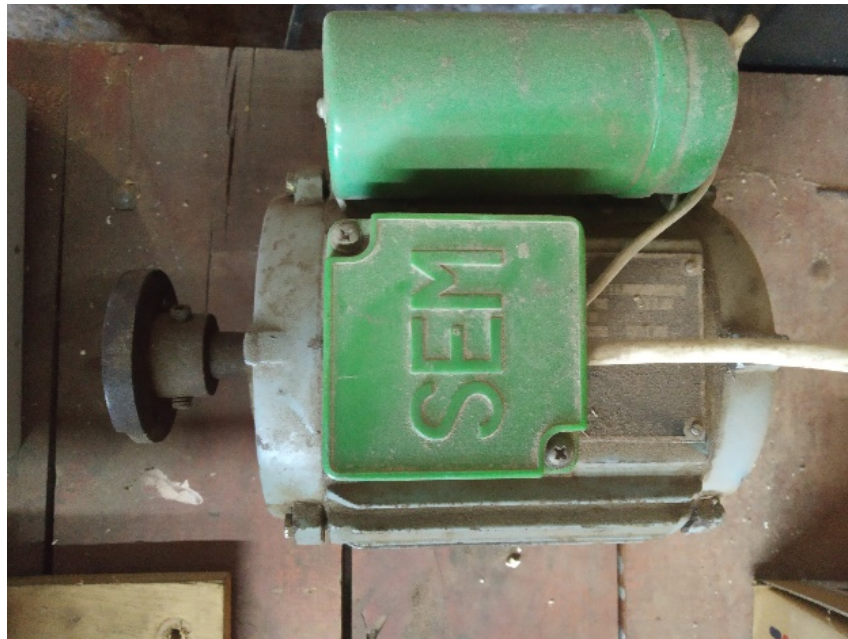


Figura 11: Motor monofásico

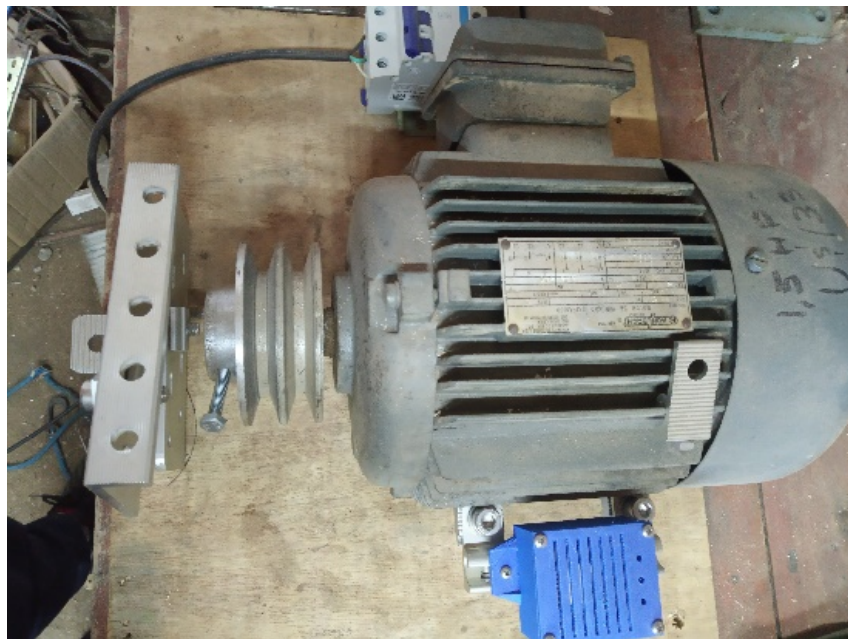


Figura 12: Motor trifásico, con dispositivo colector conectado(AZUL)

5. Experimentos y Resultados

En la primera etapa de creación de los dispositivos se tomaron 4 pequeños contrapesos de 13,39grs, 16.46grs, 32,92grs y 43,37grs respectivamente, que se adosaron en el eje de los motores, los mismo van creciendo en peso, para de esta manera generar de manera controlada un aumento de la vibración en dicho motor.

Con este contrapeso se pasa a analizar los datos que se toman por los sensores y comparar si dicho sensor percibe el cambio en el balanceo del motor. En esta etapa, se utilizó una computadora conectada al dispositivo construido para recibir los datos por puerto serial y suministrarle corriente eléctrica para su funcionamiento.

Se realizaron 5 pruebas de 9 minutos cada una para cada motor. Las pruebas se iniciaron con el motor girando si ningún peso en su eje, para luego en cada prueba subsiguiente ir cambiando los pesos en grado ascendente. Los resultados obtenidos de estas pruebas están resumidos en la **tabla 3**.

Contrapeso	Motor	Desviación X	Desviación Y	Desviación Z
Solo Eje	Monofásico	4123.872128	8003,157406	15952.25062
1	Monofásico	5593,374452	8558,753904	17758,30344
2	Monofásico	7204,056564	7660,661858	16268,43497
3	Monofásico	12227,25184	11470,08942	22107,93444
4	Monofásico	15680,80471	13542,56001	23449,16991
Solo eje	Trifásico	4138,594351	8031,888741	16025,15241
1	Trifásico	5670,003682	8590,421293	17935,88647
2	Trifásico	7232,872790	7691,687539	16342,45635
3	Trifásico	12350,38027	11597,06331	22337,19372
4	Trifásico	15730,30901	13557,66944	23546,64811

Tabla 3: Valores experimentales obtenidos del acelerómetro MPU-6050

Como es posible observar en la **tabla 3** columnas: “Desviación X” y “Desviación Y” por ejemplo es posible observar como el valor se incrementa al incrementar el des-

balanceo del eje. Dicho cálculo es la desviación estándar del promedio de los valores tomados por el sensor.

En esta primera prueba y en base a los datos obtenidos vemos como el sensor incrementa sus parámetros al introducir mayor vibración en su eje, que en las últimas pruebas era notorio el desbalanceo ocasionado, ya que se percibía sobre la mesa de pruebas.

Es oportuno aclarar que los datos mostrados en la **tabla 3** son valores enviados por el sensor, sin ningún filtro ni conversión, únicamente demuestran que el cambio en el balanceo del eje, genera una oscilación detectable por el mismo, de manera que concluimos que el sensor podría comportarse de forma adecuada para el trabajo propuesto, por lo que estaríamos cumpliendo la primera meta propuesta. También es bueno aclarar que para cada cambio de contrapeso el motor se debe apagar, para realizar el cambio en las pesas.

Se entiende que no es la forma normal de operación y que una falla no se genera de esa manera. Quedando esta prueba únicamente como validación de la capacidad del MPU-6050 para detectar las vibraciones.

Ahora seguiremos con el punto 2, que fue desarrollar una interface con Raspberry pi que pueda ser autónoma y que al mismo tiempo pueda enviar datos a un servidor central. Como características de diseño que se prefirió para este trabajo, es utilizar la menor cantidad de cables posibles y que todas las comunicaciones siempre de ser posible que se resuelvan de esa manera, a fin de no mover instalaciones de cables de red dentro de una planta en funcionamiento.

Entonces optamos por un diseño totalmente portable de la Raspberry Pi, anexándole para la comunicación con la red externa un Router inalámbrico, elementos de control de corriente serán necesarios, así como también algún modo de evitar el calentamiento de los equipos que estarán en un ambiente industrial que suele ser hostil con los equipos electrónicos.

Para que dicho esquema sea viable la interfaz inalámbrica de la Raspberry Pi será

configurada como Wireless Access Point, con un servidor de direcciones IP que permitirá que cualquier dispositivo se conecte a ella. Esto asociado el firmware colocado en cada dispositivo Wemos deja transparente para la Raspberry como identificar cada colector de datos.

Con el objetivo de que la Raspberry no utilice su procesador para tareas no relevantes al trabajo, optamos por una implementación de Raspbian sin interfaz gráfica. Para dicha selección utilizamos el criterio de la distribución recomendada por el fabricante de la placa que en este caso fue Raspberry Pi Os Lite. Raspberry [2021]

Cumpliendo con lo determinado en los objetivos instalamos un servidor web. Según el fabricante de la placa existen dos servidores web soportados por el sistema, Apache y NGINX, es este caso nos inclinamos por Apache por la experiencia con la herramienta, con la finalidad de no generar más complejidad al armado de la solución.

También deberemos instalar el lenguaje de programación PHP que permitirá que los datos enviados por los dispositivos sean almacenados en la base de datos.

Para la base de datos utilizaremos MariaDB, que es una base de datos Open Source compatible con Raspberry pi, la cual existe mucha documentación y proyectos de trabajo dentro del propio entorno de Raspberry. Raspberry [2021]

Todo el software indicado anteriormente se basa principalmente en la experiencia personal y el conocimiento de las herramientas propuestas, la simplicidad de su instalación y la amplia bibliografía existente en la web.

Una vez definido el entorno de trabajo pasamos a realizar las pruebas correspondientes. Como resultado de dicho trabajo llegamos al dispositivo que se muestra en la **figura 13**.

Descripción de la **figura 13**.

1. Raspberry PI
2. Router que conecta a la red industrial.
3. Fuente de alimentación de 12V

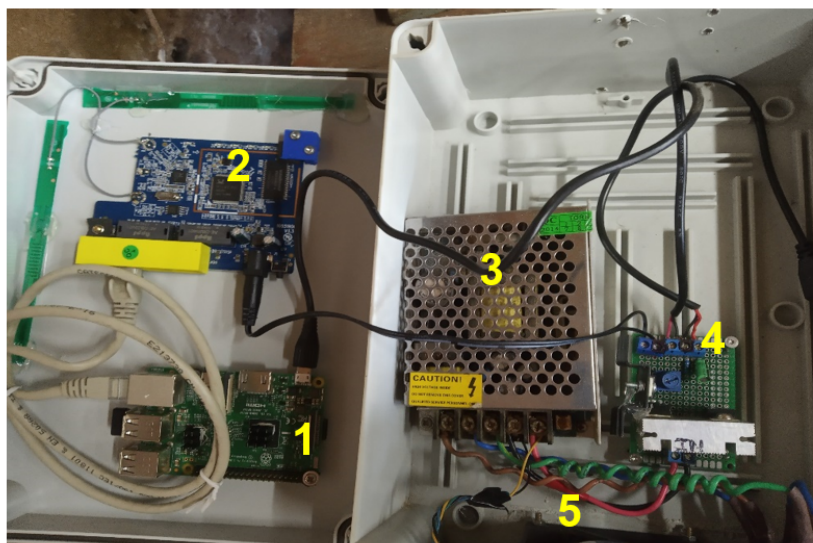


Figura 13: Caja con todos los dispositivos conectados

4. Reguladores a 9V y 5V para router y Raspberry

5. Ventilador para refrigeración

A partir de ese momento comenzamos las pruebas para que el dispositivo fuera autónomo en la recepción de datos.

Para se creó en MariaDB una base de datos con 1 tabla, que es la encargada de recibir los datos provenientes de los sensores que se conectan a esta unidad, como de detalla en la **figura 14**. La misma recibirá los datos y luego enviará los dichos al servidor que será repositorio final de los mismos, con un formato idéntico a este.

Columnas: + Agregar × Borrar ▲ Subir ▼ Bajar							
#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	Datos_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	Datos_Sensor	CHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	Datos_eje_x	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	Datos_eje_y	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
5	Datos_eje_z	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
6	Datos_tmp_s	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
7	Datos_tmp_loc	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
8	Datos_fch_hr	DATETIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Figura 14: Formato de la tabla SQL que recibo los datos

En este punto del proyecto llegamos a que el dispositivo está colocado en el motor, de manera independiente y sin ningún tipo de auxilio no previsto en el desarrollo y que el mismo envíe los datos hacia la Raspberry pi, para que la base los guarde y ellos estén disponibles para que el algoritmo de Inteligencia Artificial puede realizar los análisis de predicción.

Como es mostrado en la **figura 15**, vemos los datos recogidos desde el motor, estos datos son los que serán enviados a la red LSTM, para su análisis, y en base a su evolución prever el paso de umbral.

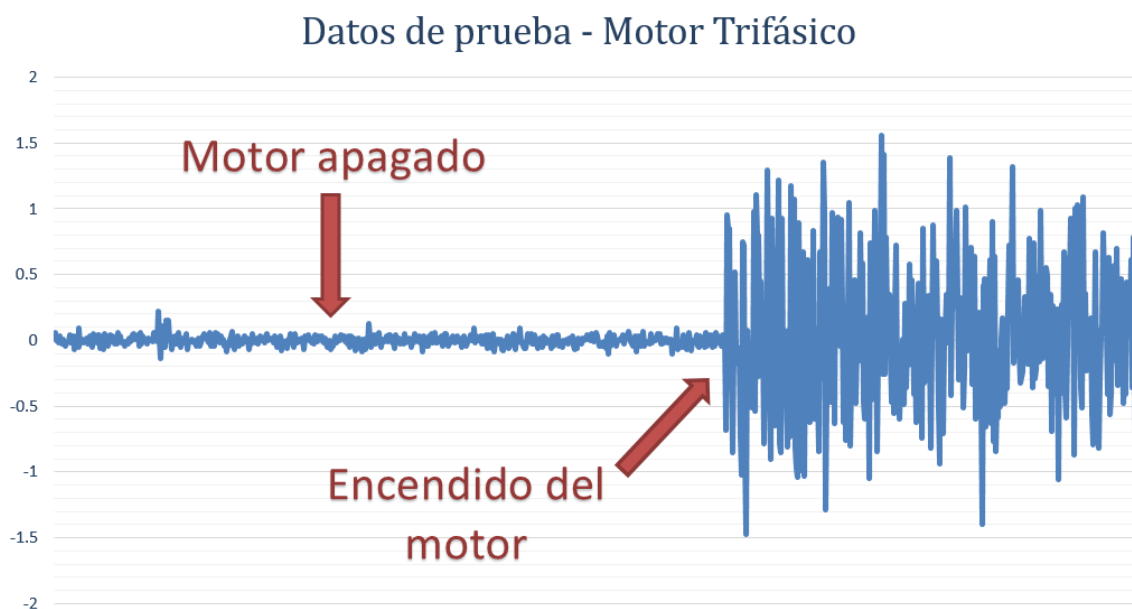


Figura 15: Formato de la tabla SQL que recibo los datos

Puesto a prueba eso, obtenemos los datos anexados en la **tabla 4** que son los datos que envía el dispositivo desde el motor en rotación. Con esta información concluimos que si es posible recibir datos de uno o más dispositivos de manera inalámbrica

Debido a la falta de un laboratorio para realizar las pruebas, fue necesario generar un dispositivo que lentamente indujera de forma sintética vibraciones en el motor.

Como se indica en la **figura 16**, este mecanismo ejerce presión sobre el rotor del motor

Dato-ID	MAC-Sensor	Datos-eje-X	Datos-Eje-Y	Datos-eje-Z	Datos-tmp-s	Datos-fhc-hr
80464	40:F5:20:28:85:C4	0.5	0.59	-5.14	23.1	6/5/2021 23:28:00
80465	40:F5:20:28:85:C4	0.46	0.58	-5.19	23.1	6/5/2021 23:28:01
80466	40:F5:20:28:85:C4	0.46	0.59	-5.17	22.9	6/5/2021 23:28:02
80467	40:F5:20:28:85:C4	0.46	0.59	-5.2	22.9	6/5/2021 23:28:03
80468	40:F5:20:28:85:C4	0.48	0.59	-5.16	22.9	6/5/2021 23:28:04
80469	40:F5:20:28:85:C4	0.49	0.61	-5.14	22.8	6/5/2021 23:28:05
80470	40:F5:20:28:85:C4	0.49	0.59	-5.16	22.9	6/5/2021 23:28:06
80471	40:F5:20:28:85:C4	0.48	0.61	-5.13	22.9	6/5/2021 23:28:07
80472	40:F5:20:28:85:C4	0.49	0.6	-5.18	23.1	6/5/2021 23:28:08
80473	40:F5:20:28:85:C4	0.46	0.58	-5.17	23.1	6/5/2021 23:28:09
80474	40:F5:20:28:85:C4	0.47	0.61	-5.2	23.1	6/5/2021 23:28:10
80475	40:F5:20:28:85:C4	0.48	0.59	-5.14	23.1	6/5/2021 23:28:11
80476	40:F5:20:28:85:C4	0.49	0.58	-5.17	23.1	6/5/2021 23:28:12
80477	40:F5:20:28:85:C4	0.49	0.6	-5.18	23.0	6/5/2021 23:28:13
80478	40:F5:20:28:85:C4	0.48	0.62	-5.17	23.1	6/5/2021 23:28:14
80479	40:F5:20:28:85:C4	0.48	0.57	-5.20	23.0	6/5/2021 23:28:15
80480	40:F5:20:28:85:C4	0.49	0.59	-5.13	23.1	6/5/2021 23:28:16
80481	40:F5:20:28:85:C4	0.51	0.57	-5.23	23.1	6/5/2021 23:28:17
80482	40:F5:20:28:85:C4	0.46	0.6	-5.18	23.1	6/5/2021 23:28:18
80483	40:F5:20:28:85:C4	0.48	0.56	-5.19	23.1	6/5/2021 23:28:19
80484	40:F5:20:28:85:C4	0.51	0.59	-5.19	23.1	6/5/2021 23:28:20
80485	40:F5:20:28:85:C4	0.47	0.58	-5.15	23.2	6/5/2021 23:28:21
80486	40:F5:20:28:85:C4	0.51	0.58	-5.17	23.1	6/5/2021 23:28:22

Tabla 4: Resultado de una consulta SQL a la tabla “Datos”

y esta va en aumento con el pasar del tiempo, haciendo que el motor comience a vibrar más a medida que el tiempo transcurre.

El Peso Variable es un recipiente con 5 litros de agua, que mediante goteo se va perdiendo el líquido de manera que la palanca comienza a ejercer fuerza sobre el rotor del motor.

Con dicho dispositivo se crearon dos conjuntos de datos distintos, con dos pesos distintos que ejercen presión sobre el eje del motor. De 4500 registros de cada uno de ellos. El primer conjunto de datos se lo separa en un grupo de 3500 y 1000 registros,

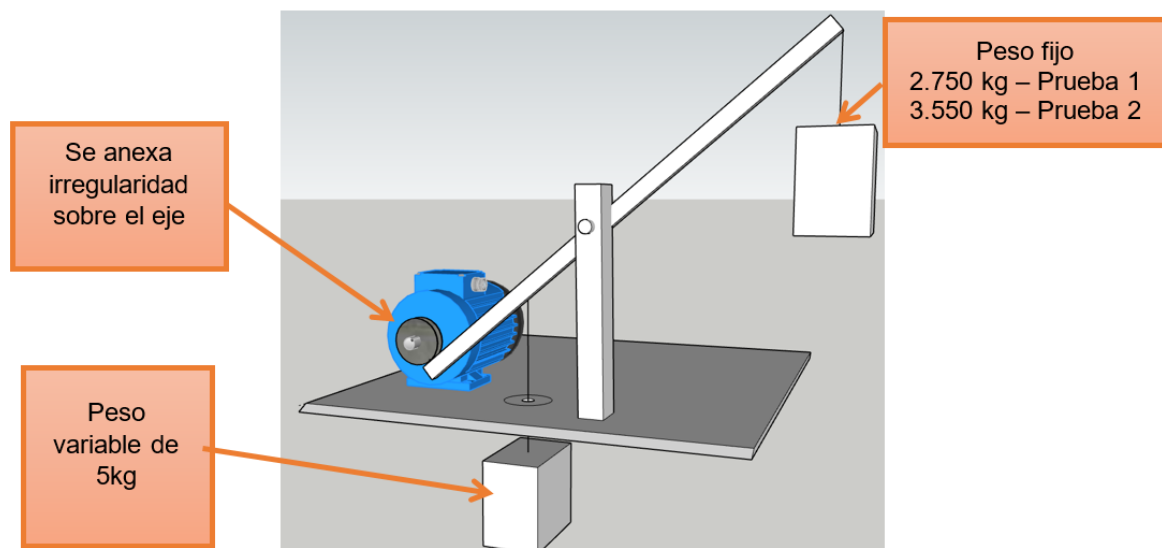


Figura 16: Dispositivo que induce vibraciones de manera paulatina al motor

los primeros para entrenamiento y los segundo para validación.

Cada uno de los registros del conjunto de datos representan 1 segundo de tiempo real. Una vez con todos los dispositivos conectados y recogiendo datos se empezó a trabajar en la arquitectura de nuestra red neuronal. Como indicamos anteriormente y utilizando como base los trabajos y artículos citados Christopher [2015] , Fidan Boylu [2019] y Garzón [2018].

En todos los casos se nombras las redes neuronales del tipo LSTM, como la más indicada para estos casos, de predicción de vibraciones. Según los citado por Fidan Boylu [2019] y Garzón [2018], se deja entendido que la configuración de dicha red es muy peculiar a cada problema en particular y que se deberá ajustar la mejor configuración para cada caso.

Con esto realizaron las pruebas correspondientes a qué tipo de algoritmo se ajustaría mejor a nuestra realidad. Para el desarrollo del modelo descrito utilizamos el lenguaje de programación Python en su versión 3.7, con las librerías correspondientes para el manejo de las redes de tipo LSTM, en nuestro caso seleccionamos Tensorflow y Keras, además de algunas otras del tipo matemático y gráficas y estadísticas, necesarias

para automatizar en análisis de los resultados obtenidos.

En principio armamos dos redes neuronales distintas y con los registros obtenidos, le pasamos los datos a todas, y cuantificamos la eficiencia y el tiempo de entrenamiento de cada una de ellas.

Dejamos en la **tabla 5**, como fue la configuración de la LSTM que dio mejor rendimiento. Durante este artículo se especificarán más ajustes que se fueron realizando en el modelo.

Model: "sequential_11"		
Layer (type)	Output Shape	Param #
lstm_12 (LSTM)	(None, 64)	16896
dense_5 (Dense)	(None, 1)	65
Total params: 16,961		
Trainable params: 16,961		
Non-trainable params: 0		

Tabla 5: Parámetros de la RED LSTM utilizada como base

Para el análisis y predicción de la red neuronal se toman los primeros 3500 para un ciclo completo de entrenamiento y validación de la red, y los siguientes 1000 para comparación exclusiva del modelo ya entrenado. El segundo conjunto de datos, será para utilizar como comparativo de predicción.

La idea de los dos experimentos surge de evitar una red que tenga sobre ajuste sobre al tener un único conjunto de datos para entrenar.

Con el primer grupo de datos se realizó todo el proceso de selección de red, análisis de datos y pruebas. En el análisis de la **figura 17** se graficaron los 4500 registros obtenidos que luego serán analizados estadísticamente en grupos de 500 para mediante la fórmula de varianza determinar cuánto la distribución entre máximos y mínimos se incrementaba con el paso del tiempo, que serán mostrados en la próxima **tabla 6**.

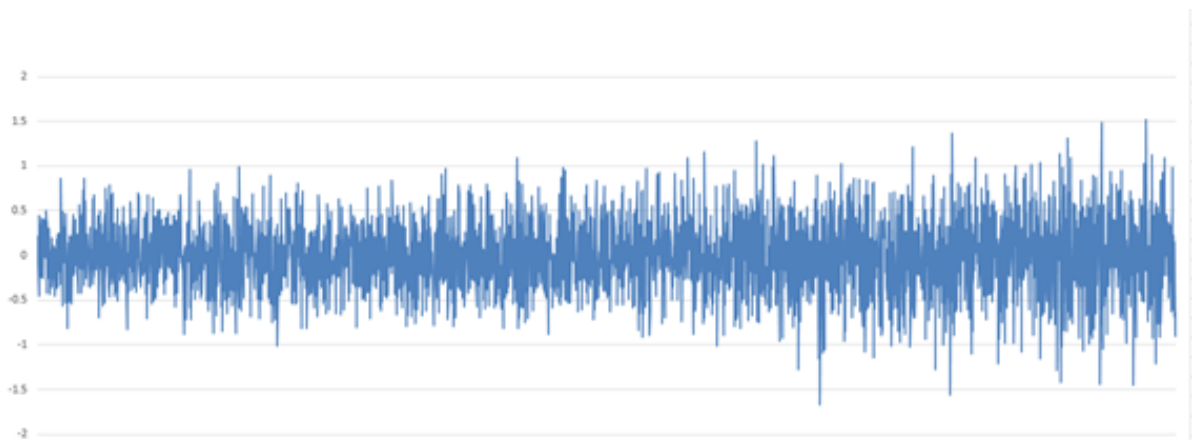


Figura 17: Incremento de la vibración mediante dispositivo – 4500 registros

Llegamos a los datos mostrados en la **tabla 6** que se detallan a continuación, estos son los datos que se recibieron a través del sensor y representan los datos obtenidos en 1 hora y 15 minutos de muestreo.

Registro	Valor Obtenido (varianza)
1 a 500	0.0798
501 a 1000	0.1089
1001 a 1500	0.1198
1501 a 2000	0.1381
2001 a 2500	0.1658
2501 a 3000	0.2150
3001 a 3500	0.2543
3501 a 4000	0.2945
4001 a 4500	0.3563

Tabla 6: Incremento de Varianza en el tiempo.

En la **figura 18** esta detallado como se va realizar el primer análisis de los datos por la red neural. Se separaron los datos, en azul los datos de entrenamiento, un total de 3000 y en naranja lo de validación que son 500 registros.

Se entrenó la red, repetidas veces, según las modificaciones citadas anteriormente.

En negro está marcada la evolución de la varianza. Los datos predichos se comparan con los datos reales y con esto se generó una puntuación de asertividad, en base al análisis estadístico, a donde se buscó cuál de las redes presentaba dispersión similar al padrón obtenido del sensor.

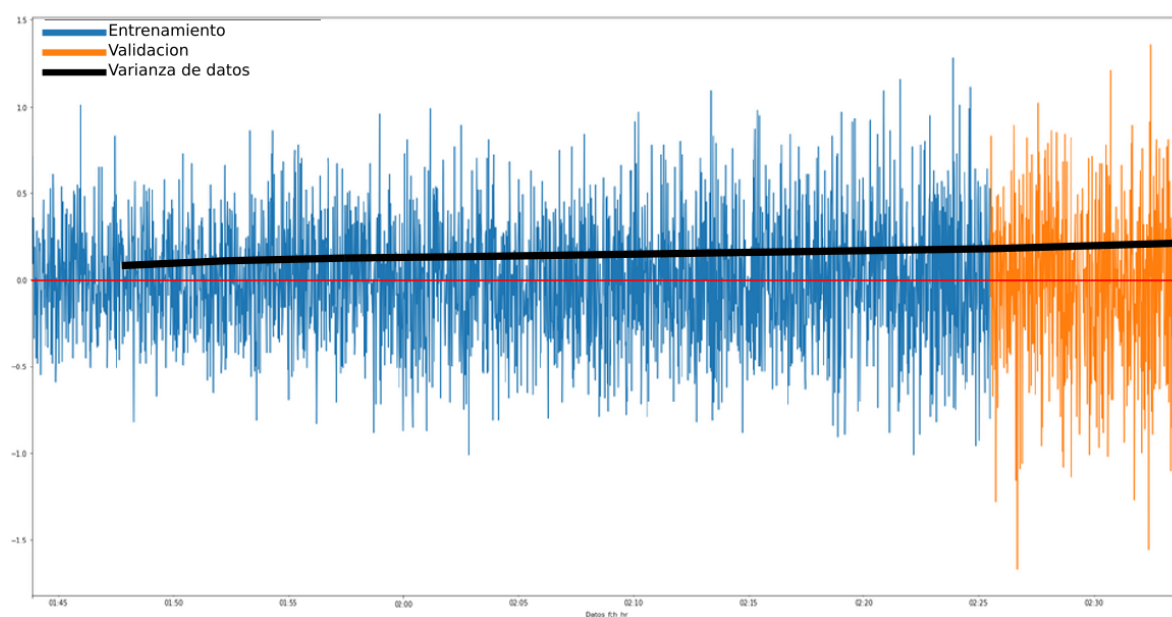


Figura 18: Incremento de la vibración mediante dispositivo – 4500 registros

A partir de ese momento hicimos cambios en la red y para cada uno de ellos se registraron las modificaciones y cuál era el impacto que estos generaban en los datos predichos.

Se probaron varias capas de redes LSTM, ajustando parámetros, parámetros de entrada, épocas. Todos estos datos fueron analizados para determinar con qué tipo de configuración el sistema se mostraba con una precisión más justa a la realidad. Se experimentó con red con una con dos capas como esta en el artículo de Fidan Boylu [2019], marcada como LSTM_1 en la **tabla 7**.

También Colocamos dentro de la función de entrenamiento un contador de tiempo con “inicio” y “fin” del entrenamiento, con la finalidad de además de tener un resultado de tiempo de proceso de entrenamiento de la red LSTM. Al final de la ejecución de la pre-

dicción el algoritmo gráfica los valores y guarda estos en formato de texto CSV para posteriormente generar los valores que de la Tabla 9 y con esto analizamos los resultados.

Se cambiaron parámetros de tamaño de red, entradas, lotes y cantidad de épocas llegamos a que dos de ellas tendrían lo necesario para adecuarse mejor a lo que buscamos. De todas estas pruebas obtuvimos los datos para generar la **tabla 7**.

PROM	VAR.	DESV.	Batch	Epoc	LSTM-1 Units	LSTM-2 Units	Test	Tiempo Entre.	OBS	% Precisión
0.581	0.187	0.433							Real	100.00
0.596	0.075	0.273	32	800	64	32	1	14:16	Red 1	68.53
0.603	0.060	0.245	32	800	128	64	2	16:36	Red 1	64.16
0.599	0.062	0.250	64	800	128	64	3	14:45	Red 1	64.66
0.627	0.131	0.362	32	800	64	32	4	12:29	Red 1	87.12
0.631	0.147	0.384	16	800	64	N/A	5	13:49	Red 2	91.93
0.573	0.153	0.392	32	800	64	N/A	6	12:21	Red 2	90.41
0.579	0.088	0.298	64	800	64	N/A	7	13:38	Red 2	71.88
0.614	0.096	0.311	16	1600	64	N/A	8	18:00	Red 2	76.34
0.597	0.144	0.380	32	1600	64	N/A	9	18:51	Red 2	89.22
0.597	0.144	0.380	64	1600	64	N/A	10	18:15	Red 2	89.22
0.615	0.162	0.403	16	800	32	N/A	11	12:24	Red 2	95.30

Tabla 7: Análisis de las simulaciones, en la primera fila, datos obtenido por el sensor.

Tres de las pruebas realizadas tienen valores superiores al 90 %. Tomamos los dos valores más altos para seguir analizando si en todos los casos se comporta de igual manera, los marcados en la tabla **tabla 7** como Test 5 y 11 respectivamente.

En la **figura 19**. comparamos los valores obtenidos de las simulaciones contra los datos reales del sensor.

Una vez concluido estos análisis, y con el fin de revalidar los datos obtenidos hasta el momento, se compara la red neuronal con los siguientes 1000 valores del conjunto de datos, con la finalidad de asegurar la persistencia en el porcentaje de acierto, y se realiza el mismo estudio, pero con la red ya entrenada únicamente solicitando que la misma prediga los próximos 1000 registros, con el objetivo de verificar si nuestra red

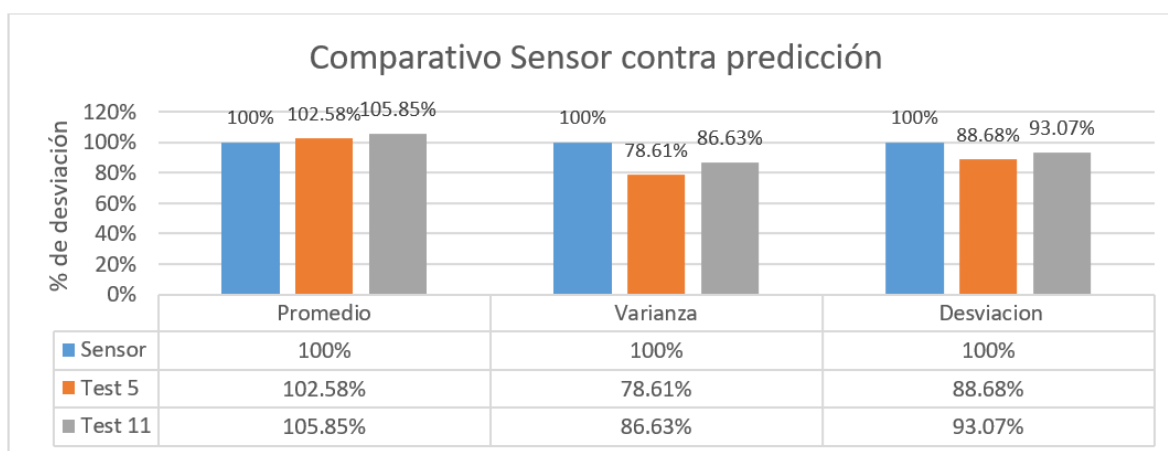


Figura 19: Gráfica comparativa entre reales y predicciones: Fuente **tabla 7**.

no sufre de sobreajuste o “overfitting” en su comportamiento.

En la **tabla 8** se hace un análisis de la varianza de todos los datos recogidos por el sensor. Como se puede apreciar en la columna marcada como Test 11 sus valores tienen una mayor similitud con los reales.

Registro	Valor Sensor	Valor Test 5 X	Error Test 5	Valor Test 11	Error Test 11
1 a 500	0.0798	X	X	X	X
501 a 1000	0.1089	X	X	X	X
1001 a 1500	0.1198	X	X	X	X
1501 a 2000	0.1381	X	X	X	X
2001 a 2500	0.1658	X	X	X	X
2501 a 3000	0.2150	X	X	x	x
3001 a 3500	0.2543	0.1845	27.44 %	0.2452	3.57 %
3501 a 4000	0.2945	0.1973	33.00 %	0.2836	3.70 %
4001 a 4500	0.3563	0.1999	43.89 %	0.3330	6.53 %

Tabla 8: Incremento de Varianza en el tiempo – Valores del sensor versus red LSTM

Teniendo la red marcada como Test 11 como la mejor, procedimos a pasarle el segundo conjunto de datos, para tratar de predecir su comportamiento y realizar la com-

paración con los datos reales, y luego tratar de determinar el punto de ruptura de dicho motor en base a esos datos.

También cabe destacar que el “Input_Shape” de la red es un lote de 30 datos, que son los que la red utiliza para predecir el próximo valor.

Quedando entonces como red neural seleccionada la descrita en la **tabla 9**

<pre> modelo = Sequential() modelo.add(LSTM(units=32, input_shape=(X_train.shape[1],1)) modelo.add(Dense(units=1)) modelo.compile(optimizer='rmsprop', loss='mse') modelo.summary() modelo.fit(X_train,Y_train,epochs=800,batch_size=16) </pre>		
Model: "sequential"		
Layer (type)	Output Shape	Param #
lstm_12 (LSTM)	(None, 32)	4352
dense_5 (Dense)	(None, 1)	33
Total params: 4,385		
Trainable params: 4,385		
Non-trainable params: 0		

Tabla 9: Red neuronal final con sus parámetros.

Llegado a este punto comenzamos a trabajar con el segundo conjunto de datos tomado anteriormente y dicho análisis queda nombrado como **figura 20**. Estos datos son los correspondientes las simulados por la red Neuronal a partir del segundo conjunto de datos para luego comparados estadísticamente. Se incluyen en esta gráfica los 4500 obtenidos del sensor marcados con la linea verde y luego se hace la predicción con la red neuronal.

Debido a la característica de los datos, tomamos grupos de 500 registros y promediamos su valor absoluto, esto debido a que la tendencia de los datos es llegar a cero, ya que varían entre positivos y negativos

Como se puede apreciar en la gráfica, los valores guardan una semejanza importante,

que mantiene lo que fue estimado en la **tabla 8**.

Teniendo esta información y como la red LSTM y entendiendo que la red predice el comportamiento de forma acertada, generamos un bucle que se repitió en el tiempo, para entender como se comporta la red en periodos mas largos de tiempo.

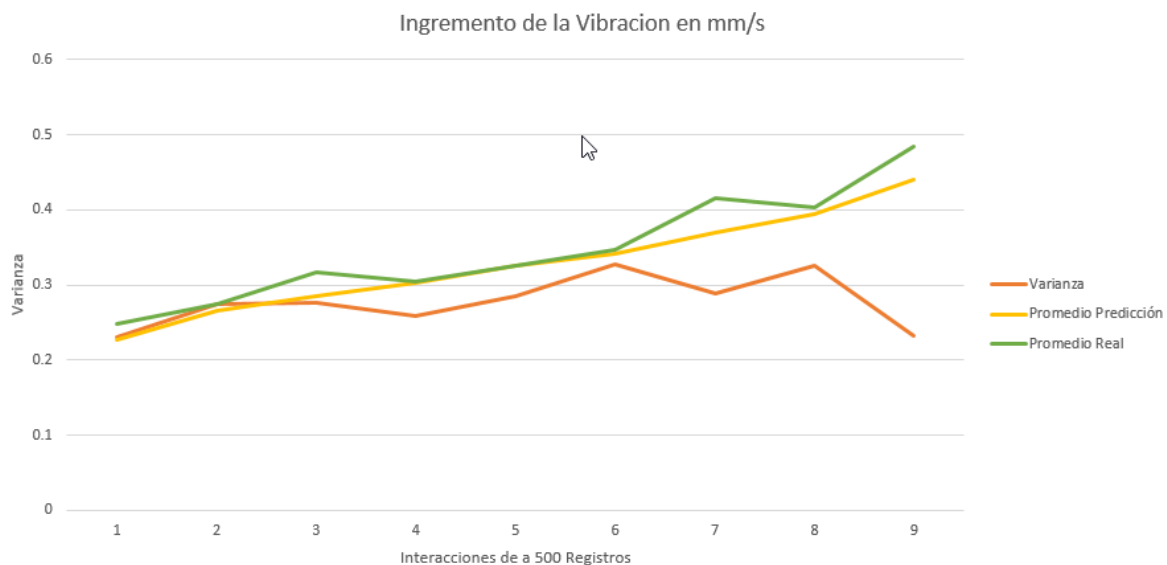


Figura 20: Predicción basada en el segundo set de datos

Este ciclo se repitió hasta el momento que el valor de varianza superara el valor de 1.50 mm/s. Todas las simulaciones se ejecutaron de manera anidada a partir de 1500 registros reales.

En la gráfica marcada como **figura 21**, se procesaron un total de 21.500 registros a futuro, recordando que cada registro equivale a 1 segundo, esto genera un aproximado de 6 horas de simulación. Los cálculos de varianza y promedios mostrados en la gráfica están agrupados de a 500 registros, para facilitar la observación de la predicción.

También cabe indicar que en la interacción 11, o sea las 4:30 horas de simulación a futuro, los valores del promedio absoluto y la varianza sumados indicado con el color marrón, en la gráfica ya superan los “1.8mm/s” lo que ya puede generar una advertencia

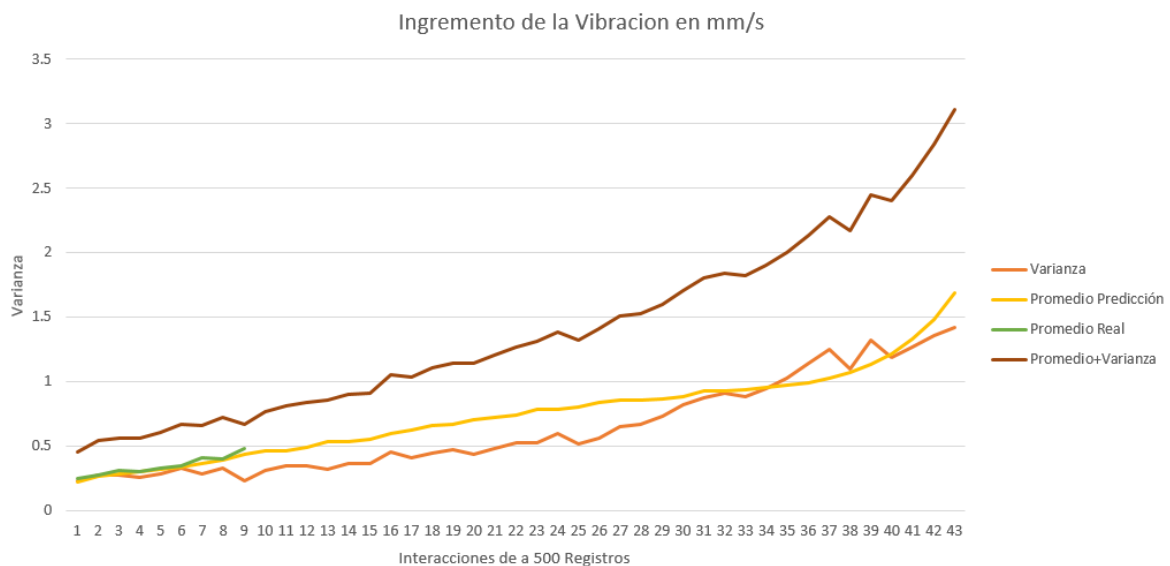


Figura 21: Predicción basada en el segundo set de datos

al centro de mantenimiento. Ya que supera lo establecido por la norma ISO 20816-1:2016, indicado en la **tabla 2**. También se puede ver que a partir de interacción 13 o a las 5:25 horas de simulación, la tendencia de la varianza, así como la del promedio cambian, y se incrementan de manera más acentuada.

6. Conclusiones

De esta manera se logra confirmar que mediante una red LSTM y en un ambiente de prueba controlado es posible obtener valores con una similitud estadística sensiblemente superior al 90 % de predicción en relación a los datos conocidos.

Sobre su predicción a futuro, como se generó el test, es esperable que la vibración continuara creciendo con el transcurrir del tiempo de manera sostenida, lo que se prueba con el incremento de la varianza de los datos.

Se notó que, en predicciones largas, la relación de crecimiento de la red neural va cambiando, lo que podría indicar que perdemos precisión con el pasar del tiempo. Debido a las limitaciones del conjunto de datos y el entorno, no es posible determinar cuál es

la causa de la variación dicho incremento.

Durante toda la realización del trabajo, se utilizó únicamente software de código abierto, esto nos lleva a concluir que es posible un sistema de mantenimiento predictivo que pueda ser construido en su totalidad con esta modalidad y pueda realizar las tareas de predicción que se planteó como objetivo.

Puede llegar a ser una semilla que aporte más conocimiento y desarrollo nuestra región en el área de hardware y software.

Todos los datos desarrollados en este proyecto, desde códigos fuente, esquemas e imágenes, se dejan disponible en GitHub con licencia GPL 3.0, para su consulta y utilización en : https://github.com/juanpdeleon/proyecto_motores.git

7. Trabajos Futuros

Completar parte del trabajo sobre el análisis de temperaturas para de igual manera que las vibraciones evaluar el sobrecalentamiento de la estructura y mediante un algoritmo de Inteligencia artificial predecir en cuanto tiempo se sobrepasaría el valor máximo indicado por la norma IEC 60085:2007 o EN:60085:2008 que son similares. Sabiendo que el modelo propuesto puede funcionar en una predicción de tiempo de algunas horas, se tratara de desarrollar una placa más compacta, posiblemente en un circuito PCB diseñado exclusivamente para tal fin, para hacer la solución física más robusta.

Probar otro tipo de envío de datos por ejemplo utilizando protocolo ModBus TCP/IP que es público y gratuito y dejar la recuperación de datos por parte de un Software como SCADA BR que también es Open Source.

Utilizar la experiencia adquirida en este trabajo, para utilizar redes del tipo LSTM en otras predicciones de tiempo relativas a otros procesos productivos.

Referencias

ABB. Abb ability™ smart sensor - motores que indican cuando realizar mantenimiento, 2021. URL <https://new.abb.com/motors-generators/es/servicios-del-motor/servicios-avanzados/smart-sensor>, (Recuperado: 01.03.2021).

E. AbdElRahman, J. Fatima El, H. James, W. Brandon, and D. Travis. Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration, 2017. URL <https://arxiv.org/pdf/1710.03753.pdf>, (Recuperado: 02.05.2021).

J. E. Castaño Mejía. Diseño e implementación de un sistema de adquisición de datos que permite predecir fallas mecánicas en los motores mediante el estudio de sus vibraciones, 2021. URL <https://repository.unad.edu.co/bitstream/handle/10596/40523/jecastanom.pdf?sequence=3&isAllowed=y>, (Recuperado: 01.10.2021).

O. Christopher. Understanding lstm networks (2015), 2015. URL <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, (Recuperado: 25.02.2021).

F. De Castilla. Detección y prognosis de anomalías aplicada a maquinas industriales - 2019, 2019. URL <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/99187/3/fernandodecastillaTFM0619memoria.pdf>, (Recuperado: 02.03.2021).

J. L. de S. Silva, M. M. Cavalcante, R. d. S. Camilo, A. L. , Galindo, and E. C. Viana. Plataforma arduino integrado ao plx-daq: Análise e aprimoramento de sensores com ênfase no lm35., 2014. URL https://www.researchgate.net/profile/Michelle-Cavalcante/publication/305771112_Plataforma_Arduino_integrado_ao_PLX-DAQ_Analise_e_aprimoramento_de_sensores_com_enfase_no_LM35/links/58343f9208aef19cb81f55b6/Plataforma-Arduino-integrado-ao-PLX-DAQ-Analise-e-aprimoramento-de-sensores-com-en.pdf, (Recuperado: 01.07.2021).

- F. D.S., I. A.Y., and Z. V.A. Trubin using of measuring system mpu6050 for the determination of the angular velocities and linear accelerations., 2015. URL http://jurnal.nips.ru/sites/default/files/A%26SE-1-2015-11_0.pdf, (Recuperado: 02.07.2021).
- ESP8266. Espressif smart connectivity platform: Esp8266, 2013. URL https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf, (Recuperado: 10.11.2020).
- U. Fidan Boylu. Deep learning for predictive maintenance - 2017, 2019. URL https://github.com/Azure/lstms_for_predictive_maintenance, (Recuperado: 02.03.2021).
- N. Free Software Foundation. General public licence, 2007. URL <https://www.gnu.org/licenses/lgpl-3.0.pt-br.html>, (Recuperado: 10.01.2021).
- Fujitsu. El mantenimiento predictivo en la industria 4.0 (2019), 2019. URL <https://observatorioindustria.org/wp-content/uploads/2019/10/Folleto-Estudio-predictivo.pdf>, (Recuperado: 10.12.2020).
- V. Garcia. Configurar el mpu6050, 2018. URL <https://www.diarioelectronico hoy.com/blog/configurar-el-mpu6050>, (Recuperado: 01.02.2021).
- S. García Garrido. *Organización y gestión integral de mantenimiento*. Ediciones Díaz dos Santos, 2006, 2003.
- J. I. Garzón. Cómo usar redes neuronales (Istm) en la predicción de averías en las máquinas - 2018, 2018. URL <https://blog.gft.com/es/2018/11/06/como-usar-redes-neuronales-lstm-en-la-prediccion-de-averias-en-las-maquinas/>, (Recuperado: 25.05.2021).
- j. Gonzáles Fernández. *Teoría y práctica del Mantenimiento Industrial Avanzado. 2da*

Edicion, publisher = Fundación Confemetal-Madrid, year=2005, keywords = Administración , mantenimiento. avanzado,.

R. Guerra. Redes neurales recurrentes, 2020. URL https://www.youtube.com/watch?v=CJYw2bYxmx4&feature=emb_logo/, (Recuperado: 25.02.2021).

O. S. Initiative. The open source definition, 2007. URL <https://opensource.org/osd>, (Recuperado: 20.10.2020).

N. Instruments. Que es labview?, 2021. URL <https://www.ni.com/es-cr/shop/labview.html>, (Recuperado: 10.02.2021).

T. Instruments. Lm35 precision centigrade temperature sensors.(08-1999 / rev 2017), 2017. URL <https://www.ti.com/lit/ds/symlink/lm35.pdf>, (Recuperado: 10.11.2020).

I. InvenSense. Mpu-6000 and mpu-6050 product specification revision 3.4., 2015. URL <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>, (Recuperado: 10.11.2020).

ISO-20816. Mechanical vibration — measurement and evaluation of machine vibration, 2016. URL <https://www.iso.org/standard/63180.html>, (Recuperado: 01.03.2021).

KiCad. A cross platform and open source electronics design automation suite, 2020. URL <https://www.kicad.org/about/kicad/>, (Recuperado: 01.11.2020).

D. Matich. Redes neuronales: Conceptos básicos y aplicaciones, 2001. URL https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/matich-redesneuronales.pdf, (Recuperado: 01.07.2021).

- M. Mehta. A breakthrough in wireless sensor networks and internet of things, 2013. URL <https://drive.google.com/file/d/1i0aCtCPRzmuLq1iEJp09hxDs7FCjb0n0/view?usp=sharing>, (Recuperado el 10-11-2021).
- M. Montero and J. Francisco. Metodología para el análisis de fallas en los motores eléctricos por medio del estudio termo gráfico, 2018. URL <http://repositorio.ucsg.edu.ec/bitstream/3317/11358/1/T-UCSG-PRE-TEC-IEM-155.pdf>, (Recuperado: 10.02.2021).
- F. E. Moreno García, J. A. Becerra Vargas, and C. A. Rendón Echeverri. Diseño de un sistema de análisis temporal y espectral para detectar fallas por vibración en motores eléctricos, 2015. URL <https://www.redalyc.org/articulo.oa?id=4139/413940775004>, (Recuperado: 10.02.2021).
- W. M. PONTES. Implantação de estação solarimétrica e de sistema supervisório com scadabr e plataforma iot em usina fotovoltaica na unilab-ce, 2021. URL http://repositorio.ufc.br/bitstream/riufc/58064/3/2021_dis_wmcspontes.pdf/, (Recuperado: 20.09.2021).
- Raspberry. Raspberry documentation and configuration system, 2021. URL <https://www.raspberrypi.org/documentation/>, (Recuperado: 15.01.2021).
- RDM. Documentación técnica – normativa de referencia, 2019. URL https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/matich-redesneuronales.pdf, (Recuperado: 17.08.2021).
- ScadaBR. Software libre hmi, gratuito e de código-fonte aberto, 2021. URL <https://www.scadabr.com.br/>, (Recuperado: 05.01.2020).
- G. Scalabrini Sampaio, A. Vallim Filho, L. Santos da Silva, and L. Augusto da Silva. Prediction of motor failure time using an artificial neural network. sensors, 2019. URL <https://doi.org/10.3390/s19194342/>, (Recuperado: 05.01.2020).

- R. Tabarés Gutiérrez. Exploring the emergent open manufacturing industry, 2018. URL <https://openmaker.eu/wp-content/uploads/2018/02/D4.3-White-paper-version-1.0.pdf>, (Recuperado: 10.01.2021).
- R. Tabarés Gutiérrez. Exploring the emergent open manufacturing industry - 2018, 2019a. URL <https://openmaker.eu/wp-content/uploads/2018/02/D4.3-White-paper-version-1.0.pdf>, (Recuperado: 5.03.2020).
- R. Tabarés Gutiérrez. La fabricación abierta:¿un camino alternativo a la industria 4.0. revista iberoamericana de ciencia, tecnología y sociedad - cts. 2019, 2019b. URL <https://www.redalyc.org/articulo.oa?id=92460273016>, (Recuperado: 5.03.2020).
- R. Tabarés Gutiérrez. La fabricación abierta:¿un camino alternativo a la industria 4.0. *Revista Iberoamericana de Ciencia, Tecnología y Sociedad - CTS*, 119:263–285, 01 2019c. URL <https://openmaker.eu/wp-content/uploads/2018/02/D4.3-White-paper-version-1.0.pdf>, (Recuperado: 10.01.2021).
- L. A. Tavares. *Administração moderna da manutenção*. Novo Polo Publicações, 1999.
- K. D. Team. About kicad, 2021. URL <https://www.kicad.org/about/kicad/>, (Recuperado: 01.11.2020).
- R. Timmer, M. Helinko, and R. Eskola. Eficiencia de motores optimización del rendimiento durante la vida útil de los motores., 2007. URL http://www.ie.com.co/pdf/ABB/02-2007/81-84%202M746_SPA72dpi.pdf, (Recuperado: 01.07.2021).
- M. Varanis, A. L. Silva, P. H. Ayres Brunetto, and R. Ferreira Gregolin. Instrumentation for mechanical vibrations analysis in the time domain and frequency domain using the arduino platform. *Scielo - Brasil*, 2016. URL <https://www.scielo.br/j/rbef/a/rT6fVfTg3VPSGbCLSBqHVBj/?lang=en#>, (Recuperado: 01.05.2021).

- G. WEG. Motores eléctricos guía de especificación (12/2016), 2019a.
URL <https://static.weg.net/medias/downloadcenter/hf9/hcf/WEG-WM0-motores-electricos-guia-de-especificacion-50039910-brochure-spanish-web.pdf>, (Recuperado: 05.01.2021).
- G. WEG. Weg motor scan, 2019b. URL <https://www.weg.net/wegmotorscan/es>, (Recuperado: 05.02.2021).
- G. White. *Intruducción al Analisis de Vibraciones*. Azima DLI, 2010, 2010.
- Wikipedia. Dirección mac, 2021. URL https://es.wikipedia.org/wiki/Direcci%C3%B3n_MAC, (Recuperado: 10.01.2021).