

**INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERÍA ELECTROMECAÁNICA**

“Diseño de un sistema experto para el monitoreo y alerta de modos de falla para el sistema de bombeo de agua potable en el Tecnológico de Costa Rica”

REALIZADO POR:

Antony Andrés Alfaro Alvarado

COORDINADOR DE PRÁCTICA:

Ing. Ignacio del Valle Granados

II SEMESTRE 2020



Carrera evaluada y acreditada por: Canadian Engineering Accreditation Board

Bureau Canadien d'Accréditation des Programmes d'Ingénierie

CARTA DE ENTENDIMIENTO

Fecha: 19 de enero del 2021

Señores
Instituto Tecnológico de Costa Rica
Sistema de Bibliotecas del Tecnológico

Yo Antony Andrés Alfaro Alvarado

carné No. 2016112120, si autorizo no autorizo, al Sistema de Bibliotecas del Tecnológico (SIBITEC), disponer del Trabajo Final de graduación, del cual soy autor, para optar por el grado de Licenciatura, en la carrera de Ingeniería en Mantenimiento Industrial, presentado en la fecha 19 de enero del 2021, con el título Diseño de un sistema experto para el monitoreo y alerta de modos de falla para el sistema de bombeo de agua potable en el Tecnológico de Costa Rica.

para ser ubicado en el Repositorio Institucional y Catálogo SIBITEC, con el objetivo de ser visualizado a través de la red Internet.

Firma de estudiante:



Correo electrónico:

alfaroanthon@estudiantec.cr

Cédula No.:

1-1663-0445



Diseño de un sistema experto para el monitoreo y alerta de modos de falla para el sistema de bombeo de agua potable en el Tecnológico de Costa Rica por Antony Andrés Alfaro Alvarado se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

Agradecimientos y dedicatoria

Primero agradecer a Dios por permitirme llegar hasta acá y realizar este trabajo. A mi familia, por darme el apoyo necesario para concluir esta etapa, en especial, a mis padres que fueron un pilar y siempre estuvieron a mi lado, a pesar de las dificultades en todo este largo trayecto, no solo de mi estancia en la universidad, sino que en toda mi vida y que, gracias a su amor, trabajo y sacrificio, he logrado llegar hasta este punto.

A todos mis amigos y compañeros del TEC que, desde el inicio, fueron un gran soporte y apoyo en las distintas etapas de la universidad y se mantuvieron con el paso del tiempo, gracias por todos esos momentos que sin lugar a duda nunca olvidaré.

A todos los profesores con los que tuve la dicha de compartir y con algunos de ellos tuve la oportunidad de ser su asistente, cada uno de ellos me enseñó muchísimas cosas tanto para mi vida académica como personal y a los cuales les estoy profundamente agradecido.

Al equipo CIDEM de la escuela de Ingeniería Electromecánica, por permitirme participar en el proyecto e.Wave, el cual me permitió surgir a nivel personal y aplicar muchos de los conocimientos adquiridos en los cursos, y donde encontré, más que compañeros, una familia en la que confiaré y los tendré presentes siempre.

A los profesores Carlos Piedra y Manuel Centeno, así como al Departamento de Mantenimiento por brindarme la oportunidad de realizar mi proyecto de graduación en el TEC y por confiar en mí para salir adelante en esta nueva etapa.

Y, por último, a mis amigos Alonso, Adrián, Esteban Alvarado, Esteban Jaén, por ser de las primeras personas en recibirme en el TEC y por tantas experiencias a lo largo de mi estancia en la universidad, donde nunca faltaron las risas y dejarme una amistad inolvidable.

Contenido

1.	Resumen.....	10
2.	Abstract.....	11
3.	Introducción.....	12
3.1.	Reseña de la empresa.....	13
4.	Descripción del proyecto	16
4.1.	Información del estudiante	16
4.2.	Datos de la empresa.....	16
4.3.	Planteamiento del problema	17
4.4.	Objetivos.....	20
4.4.1.	Objetivo general	20
4.4.2.	Objetivos específicos.....	20
4.5.	Justificación.....	21
4.6.	Viabilidad	25
4.7.	Alcance	26
4.8.	Limitaciones	28
4.9.	Metodología.....	29
5.	Marco teórico.....	30
5.1.	Mantenimiento en la Industria 4.0.....	31
5.2.	Inteligencia artificial.....	34

5.2.1.	Características de la inteligencia artificial	36
5.2.2.	Campos de aplicación de la IA.....	38
5.3.	Machine Learning.....	39
5.4.	Redes Neuronales	45
5.5.	Red neuronal profunda (Deep Neural Network)	48
5.5.1.	Función sigmoide	51
5.5.2.	Función ReLU	51
5.5.3.	Función tangente hiperbólica	52
5.5.4.	Función de error (Cost and Loss Function).....	53
5.5.5.	Función de descenso de gradiente (Gradient Descent)	55
5.6.	Sistemas expertos	59
5.7.	Python.....	63
5.8.	Librerías y módulos en Python.....	64
5.8.1.	Numpy.....	66
5.8.2.	Tkinter	67
5.8.3.	Pandas.....	70
5.9.	Arduino.....	71
5.10.	Módulo Wifi ESP8266.....	73
5.11.	Bombas centrífugas.....	75
5.12.	Motor eléctrico.....	78

5.12.1. Vibración.....	81
6. Desarrollo del proyecto (En relación con el objetivo general)	85
6.1. Auditoría de equipos (Primer objetivo específico).....	87
6.2. Modos de falla (Segundo objetivo específico)	90
6.3. Parámetros por observar (Segundo objetivo específico)	93
6.4. Formas de muestreo y ubicaciones (Tercer objetivo específico)	95
6.5. Establecimiento de alarmas iniciales	99
6.6. Modelo para el procesamiento y análisis de datos (Tercer objetivo específico)	102
6.6.1. Arduino.....	102
6.6.2. Redes Neuronales.....	110
6.6.3. Almacenado de datos	119
6.6.4. Interfaz gráfica	121
6.7. Determinación de las recomendaciones y acciones de mantenimiento	124
6.8. Comparación económica (Cuarto objetivo específico)	126
6.9. Beneficios de la aplicación de un sistema experto	129
7. Conclusiones	131
8. Recomendaciones	132
9. Bibliografía	133
10. Apéndices.....	141

Índice de tablas

Tabla 6.1. Síntomas de problemas en bombas centrífugas.	91
Tabla 6.2. Causas de problemas en bombas centrífugas.....	92

Índice de figuras

Figura 4.1. Desviación en el problema del monitoreo de condición en sistema de bombeo	18
Figura 4.2 Diferencias entre el mantenimiento preventivo y correctivo.....	21
Figura 5.1. Bases y aplicaciones de la inteligencia artificial.	39
Figura 5.2. Regresión Logística y Lineal.....	42
Figura 5.3 Hiperplano en la máquina de vector soporte.. ..	43
Figura 5.4 K-vecinos más cercanos.	44
Figura 5.5. Perceptrón de una red neuronal	46
Figura 5.6. Múltiples neuronas conectadas en forma de red.....	47
Figura 5.7. Surgimiento del Aprendizaje Profundo.. ..	49
Figura 5.8. Red Neuronal Profunda y Red Neuronal simple	50
Figura 5.9 Función Sigmoide.....	51
Figura 5.10. Función ReLU.	52
Figura 5.11 Función tangente hiperbólica.....	53
Figura 5.12 Representación de la función de costo.	55
Figura 5.13 Razón de aprendizaje.....	59
Figura 5.14 Base de condiciones de un sistema experto.....	61
Figura 5.15 Importado de módulos en Python.....	65
Figura 5.16 Ejemplo de uso de la librería NumPy.	67
Figura 5.17. Arduino UNO.	71

Figura 5.18. Módulo Wifi ESP8266-01.....	73
Figura 5.19 Conexión del protocolo UART.....	75
Figura 5.20. Bomba centrífuga con acople directo.	76
Figura 5.21. Principio de funcionamiento de una bomba centrífuga.	77
Figura 5.22. Partes de una bomba centrífuga.	78
Figura 5.23. Motor de inducción de jaula de ardilla	79
Figura 5.24. Placa de características de un motor eléctrico de inducción.	80
Figura 5.25. Clases de aislamiento.....	81
Figura 5.26 Parámetros de vibración.	83
Figura 5.27. Vibración en maquinaria y su estado, basado en ISO 10816-3.....	84
Figura 6.1 Procedimiento para el monitoreo de condición	86
Figura 6.2. Bombas centrífugas instaladas.....	87
Figura 6.3 Bombas centrífugas y motores instalados.	88
Figura 6.4. Parámetros por monitorear para evitar fallas en los equipos.....	94
Figura 6.5. Sensor de vibración piezoeléctrico.....	95
Figura 6.6. Puntos recomendados de medición de vibraciones.	97
Figura 6.7. Sensor de temperatura TMP36.	98
Figura 6.8. Valores de alerta por vibraciones.	101
Figura 6.9. Definición de variables Arduino UNO	103
Figura 6.10. Configuración de sensores.....	103
Figura 6.11. Ciclo de ejecución Arduino.	105
Figura 6.12. Conexión del módulo WIFI ESP8266-01.....	107
Figura 6.13. Configuración del módulo ESP8266-01.....	107

Figura 6.14. Programación del servidor web.	109
Figura 6.15. Adquisición de datos del servidor web.	110
Figura 6.16 Diferencias entre el Machine Learning y Aprendizaje Profundo.	112
Figura 6.17. Representación de datos de vibración para el entrenamiento.	114
Figura 6.18. Dimensiones de la red neuronal para vibraciones.	115
Figura 6.19. Llamado a la función principal de la red neuronal.	116
Figura 6.20. Iniciar las constantes aleatoriamente.	117
Figura 6.21. Resultados de la red neuronal para vibración.	118
Figura 6.22 Resultados de la red neuronal para temperatura.	119
Figura 6.23 Almacenado de parámetros mediante pickle.	119
Figura 6.24. Guardado de datos.	121
Figura 6.25. Interfaz gráfica del sistema experto.	124
Figura 6.26 Recomendaciones para alertas de falla.	125
Figura 6.27. Costo total de dispositivos para el Sistema de monitoreo.	128
Figura 10.1 Comandos AT.	141
Figura 10.2 Datos de placa del motor Baldor.	142
Figura 10.3 Datos de placa de motor US Electric.	143
Figura 10.4 Zonas de alerta en mm/s para vibración.	144
Figura 10.5 Bombas instaladas (se muestran 2 de 3).	145
Figura 10.6 Relé inteligente para el control de las bombas.	146
Figura 10.7 Conexión del Relé inteligente.	147

1. Resumen

En el presente documento, se explora el desarrollo de un sistema experto para el monitoreo y alerta de fallas en el sistema de bombeo de agua potable, en el campus central del Tecnológico de Costa Rica, ubicado en Cartago. Con base en los resultados de una exploración literaria previa, basada en normas y experiencias de otros desarrolladores, se establece que los parámetros más representativos para determinar la condición de las bombas centrífugas son: vibración, temperatura, y corriente eléctrica.

La forma en la que las variables serán capturadas será mediante sensores de bajo costo para la primera etapa de implementación del sistema (a modo de prototipo, donde se obviarán por el momento los parámetros eléctricos), luego, se podrán cambiar por instrumentos más precisos y de grado industrial. Las señales se enviarán a un Arduino UNO R3, donde serán convertidas a sus unidades correspondientes (g en vibración y °C en temperatura) y, posteriormente, transmitidas a un servidor web, mediante el módulo ESP8266-01.

En la etapa de análisis, se hará uso de una red neuronal profunda o DNN, por sus siglas en inglés, lo cual permite, apoyado en un historial de mantenimiento, determinar las regiones y umbrales donde se produzcan las alarmas, de modo que el sistema pueda aprender con el paso del tiempo. Cuando se detecta una anomalía, se brindan recomendaciones para eliminar el problema basadas en Karassik (2001). Finalmente, se construirá una interfaz gráfica mediante el módulo Tkinter, para que el usuario pueda leer e interpretar los datos de forma más fácil e intuitiva.

Palabras clave: Sistemas expertos, red neuronal profunda, Python, Tkinter, bombas centrífugas, mantenimiento, alertas, vibración, temperatura.

2. Abstract

In this document, the development of an Expert System for monitoring and warning of failures in the water supply pumping system is explored, at the campus of Tecnológico de Costa Rica located in Cartago. From the results of a previous literary exploration, based on standards and experiences of other researchers, it is established that the most representative parameters to determine the condition of centrifugal pumps are vibration, temperature, and current consumption.

The way in which the variables will be captured will be through low-cost sensors for the first stage of system implementation (as a prototype, where electrical parameters will be ignored), then they can be changed for more precise instruments and industrial grade later on. The signals will be sent to an Arduino UNO R3, where they will be converted to their corresponding units (g in vibration and °C in temperature) and later transmitted to a web server through the ESP8266-01 module.

In the analysis, it will be performed by a deep neural network or DNN, which allows supported by a maintenance history, to determine the regions and thresholds where alarms occur, so that the system can learn over time. When an anomaly is detected, recommendations are provided to eliminate the problem based on Karassik (2001). Finally, a graphical interface will be built using the Tkinter module, so that the user can read and interpret the data in an easier and more intuitive way.

Keywords: Expert Systems, deep neural network, Python, Tkinter, centrifugal pumps, maintenance, alerts, vibration, temperature.

3. Introducción

La vida misma como se conoce tiene su fundamento en uno de los elementos de forma tan pura y básica como lo es el agua. El cuerpo humano está constituido por entre un 65 % y un 75 % de este líquido. Estos niveles pueden bajar de forma significativa cuando se realizan actividades como deportes, trabajo o, simplemente, por un día caluroso; por este motivo, el agua perdida principalmente en forma de sudor debe ser repuesta a la mayor brevedad posible, ante lo cual las personas acuden a bebederos por el consumo de agua potable.

Según menciona la Organización Mundial de la Salud [OMS] (2006), “la gran mayoría de los problemas de salud relacionados de forma evidente con el agua se deben a la contaminación por microorganismos (bacterias, virus, protozoos u otros organismos)”. Por consiguiente, los sistemas de suministro de agua potable deben estar en constante monitoreo en forma experta tanto en sus equipos como en la calidad del fluido, con el fin de evitar que se produzcan fallas que conlleven a problemas en la distribución, brindando una serie de alertas para preparar una intervención adecuada. Esto es aún más crítico cuando se habla de una universidad, como en este caso, donde se encuentran diariamente miles de personas en sus instalaciones, por ello, debe garantizarse que puedan acceder a una fuente de agua potable en el momento que lo deseen y con la seguridad de que su vida no correrá peligro si se consume este líquido vital.

Un faltante de agua potable se considera como impensable en la actualidad, ya que, por derecho, se debe garantizar el acceso, esto se torna más crítico cuando se debe satisfacer una alta población, como lo tiene el Tecnológico de Costa Rica, por ello, el sistema de bombeo y de suministro de este líquido debe estar en perfectas condiciones prácticamente en todo momento, reduciendo las fallas imprevistas y paros repentinos en el sistema, así como también manteniendo niveles de inocuidad del agua en los estándares establecidos por el OMS y el Acueductos y Alcantarillados (AyA).

3.1. Reseña de la empresa

En los años sesenta y setenta, comenzó un auge en el desarrollo de la tecnología que impulsaba cada vez más al país a capacitar a las personas hacia lo que se concebía como un futuro tecnológico donde las empresas comenzaban a tener maquinaria cada vez más especializada, métodos más precisos y tecnológicamente avanzados, Costa Rica necesitaba personal con una capacidad más técnica que académica. En ese momento, según menciona Cordero (2013), solo se tenía un colegio de Artes y Oficios (COVAO) en la provincia de Cartago y era necesarias personas con una formación técnica de grado superior, es por esto que el abogado Jorge Luis Villanueva Badilla y el médico Fernando Guzmán Mata deciden crear una institución donde se brinde esa educación tan necesaria, Guzmán se había formado en México y había obtenido información de cómo fue creado el Instituto Tecnológico de Monterrey.

En esa época, las únicas opciones de educación superior en el país eran la Universidad de Costa Rica y la Escuela Normal de Heredia, las cuales no figuraban dentro de la capacitación técnica deseada, es por ello que, para tomar una referencia, debe orientarse hacia el extranjero, donde se encontraban opciones como el Instituto Tecnológico de Massachusetts, ubicado en Estados Unidos, y el Instituto Tecnológico de Monterrey. Al ver esto, Guzmán y Villanueva presentan un proyecto de ley a la Asamblea Legislativa para crear en Cartago un Instituto Tecnológico en el país.

La Asamblea Legislativa, en relación con este proyecto aún tenía la duda de si era lo correcto instalar esta nueva institución en la provincia de Cartago o era mejor colocarla en la provincia de Alajuela, ya que según menciona (Cordero, 2013) “El ambiente económico de Cartago no era el mejor. Los embates de tres desastres naturales y la guerra de 1948 habían provocado repetidas migraciones”.

Sin embargo, existió un esfuerzo de toda la provincia impulsado por grupos de profesionales, estudiantes e incluso la iglesia para lograr que se estableciera en Cartago para promover el desarrollo de dicha provincia. Para el jueves 10 de junio de 1971, el congreso aprueba que el Instituto Tecnológico de Costa Rica tuviera su sede principal en Cartago bajo el consecutivo legislativo 4777, produciendo una gran celebración en toda la provincia como lo reseña esta institución en su página oficial (Tecnológico de Costa Rica [TEC],2020):

Desde las dos de la tarde se efectuaron desfiles de colegios del cantón central de Cartago, con representaciones de estudiantes de Limón y de otros lugares de la vieja metrópoli. Los desfiles culminaron frente al Gimnasio del Colegio San Luis Gonzaga, donde se llevó a cabo la firma de la ley. A este acto asistió el presidente de la República, José Figueres Ferrer, y miembros de su gabinete; el presidente de la Asamblea Legislativa, Lic. Daniel Oduber; algunos diputados, magistrados de la Corte Suprema de Justicia; munícipes de Cartago e invitados especiales.

Así, luego de un período de construcción, el curso lectivo comenzó con menos de un centenar de estudiantes y con solo tres carreras profesionales como oferta académica: Mantenimiento Industrial, Producción Industrial y Construcción, logrando la primera graduación tres años después de que se llevara a cabo la primera lección en esta institución. No obstante, con el paso de los años, esta casa de enseñanza ha incrementado considerablemente la cantidad de carreras que oferta, llegando a las 29 carreras académicas, 12 bachilleratos, 8 maestrías, 2 doctorados y 16 licenciaturas y más de más de ocho mil estudiantes, también el campus se ha renovado hasta llegar a una cifra de 900 hectáreas de terreno y 145 000 metros cuadrados de construcción, la cantidad de docentes y funcionarios se ha incrementado hasta los 1 300 y graduando a más de 18 000 estudiantes.

El TEC ha incrementado sus sedes logrando estar en 3 provincias del país, como lo son Limón, Alajuela y San José, en el caso de Alajuela, se tiene dos sedes una ubicada en el centro de la provincia y la otra ubicada en el cantón de San Carlos, siendo esta la segunda más grande, luego del campus central. Los esfuerzos del TEC por ser una institución de enseñanza superior de calidad han sido reconocidos por diversos entes entregando variadas certificaciones, tal es el caso del Canadian Engineering Accreditation Board o mejor conocido por sus siglas CEAB, este órgano canadiense ha certificado a varias carreras en el TEC, indicando que se encuentran a un nivel similar a las que se imparten en universidades de otras latitudes y, a nivel país, se cuenta con la certificación de SINAES (Sistema Nacional de Acreditación de la Educación Superior) en casi todas las carreras que imparte.

Por otra parte, en los últimos años el TEC ha comenzado investigaciones en diversos ámbitos mediante proyectos de Investigación, donde tanto profesores como estudiantes, se unen para obtener conocimientos sobre temas específicos, así como la implementación de estos para crear un beneficio para el país o el mundo, como, por ejemplo, el Laboratorio de Plasmas, Movilidad Eléctrica, Laboratorio de Investigación de Energía Eólica y el Centro de Investigación de Energía Marina, solo por mencionar algunos de los existentes en la institución.

Por consiguiente, estos esfuerzos han contribuido para obtener un reconocimiento de entes internacionales que califican a las universidades por labores investigativas, calidad de educación, formación de profesores, publicaciones, entre otros rubros, obteniendo recientemente un puesto entre las mejores 1000 universidades del mundo. Lo que confirma que esta institución pública va por buen camino en la excelencia de la educación que brinda a los costarricenses y su contribución con la sociedad.

4. Descripción del proyecto

4.1. Información del estudiante

Nombre completo: Antony Andrés Alfaro Alvarado.

Número de cédula: 1-1663-0445.

Número de carné: 2016112120.

Edad: 24 años.

Números de teléfono: 8724-7279.

Correos electrónicos: alfaroanthon@estudiantec.cr; alfaroanthony20@gmail.com

Dirección exacta de domicilio: 900m noroeste de la plaza de deportes de Chahuites, Carrizal, Alajuela.

4.2. Datos de la empresa

Nombre: Instituto Tecnológico de Costa Rica.

Actividad Principal: Educación Superior.

Dirección: Calle 15, Avenida 14, 1 km Sur de la Basílica de los Ángeles, Provincia de Cartago, Cartago, 30101.

Contacto: Ing. Manuel Centeno López.

Teléfono: 2550-9191.

Correo electrónico: mcenteno@itcr.ac.cr

4.3.Planteamiento del problema

La forma de monitorear el sistema de agua potable en el campus del TEC, en la actualidad, es de forma presencial y totalmente manual, lo que conlleva a que, si no se tiene una planificación adecuada, existirán fallos imprevistos que pondrán en riesgo el suministro de agua potable, además, parte de las experiencias que han marcado al personal de mantenimiento es la detección de fallas por casualidad, cuando un colaborador se desplazaba por la zona del sistema de bombeo.

Asimismo, debido a que la ubicación de las oficinas y talleres de mantenimiento en el campus está alejadas del sistema de bombeo resulta difícil estar pendiente del estado actual por medio de ruidos o situaciones que se puedan presentar y, dado que este no es el único sistema que necesita ser monitoreado, el personal necesita de herramientas que faciliten su labor concentrándose en los problemas más significativos sin dejar de lado el estado de este sistema de bombeo de tanta trascendencia para las personas que utilizan el TEC.

Dado lo anterior, el departamento de mantenimiento tomó la decisión de contratar a la empresa RQL Ingeniería, con el fin de tercerizar esta labor de preservar en buen estado el funcionamiento del sistema de bombeo de agua potable, tanto al tanque elevado como a la bomba sumergida.

El problema radica en que, según comentarios de parte del personal de Administración de Mantenimiento, las visitas de la empresa al sistema de bombeo son mediante intervalos de dos meses, esto resulta en un gran inconveniente, ya que no se tiene conocimiento del estado de las bombas, hasta que se realice una de las visitas programadas, el resto del tiempo las máquinas permanecen libres de supervisión, donde la única forma de conocer si existe un problema en alguna de ellas, será mediante el faltante de agua potable en el campus.

Con el sistema actual, solo se conoce el estado 6 días en el año, los restantes 359 días resulta en una incertidumbre si las bombas operan bajo condiciones adversas, las cuales podrán provocar que la vida útil del bien se reduzca, por ejemplo, si la temperatura del motor llega a valores superiores a los cuales fue diseñado (Clase de aislamiento), tendrá repercusiones en el aislamiento y provocará que se degrade con el tiempo, reduciendo la salud de la máquina y acortando su duración en funcionamiento, lo cual puede ser evitado si se conociera en tiempo real la temperatura con suficiente antelación para establecer una acción preventiva.

A partir de las afirmaciones anteriores, se plantean dos escenarios, uno de ellos es el sistema actual que se tiene para el monitoreo de condición en el sistema de bombeo bajo una tercerización y visitas programadas, y el otro es la forma idónea en que debería de realizarse el mantenimiento bajo los lineamientos de la Industria 4.0, donde las intervenciones se aplicarían basadas en la condición de la máquina, aprovechando la vida útil de los componentes al máximo. La diferencia entre los escenarios es la desviación que debe atenderse para acercarse lo más posible a la idoneidad, una representación gráfica de los escenarios se observa en la figura 4.1.

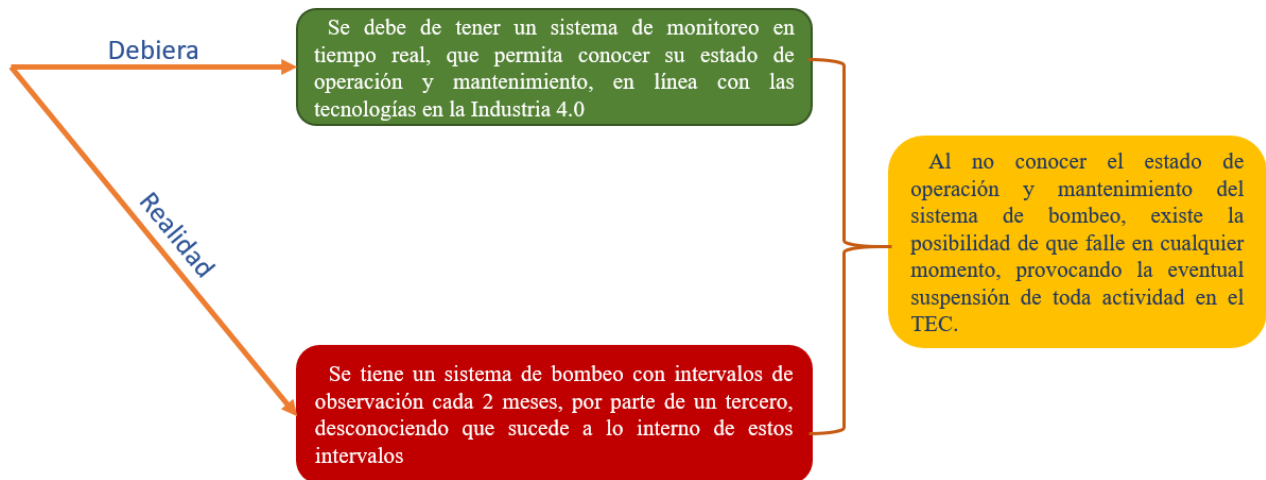


Figura 4.1. Desviación en el problema del monitoreo de condición en sistema de bombeo. Fuente: Elaboración propia.

Con el fin de subsanar este problema, es necesario contar con un sistema que permita el monitoreo de las variables y parámetros más representativos y, a su vez, los muestre en pantalla, para que el personal de mantenimiento esté enterado en todo momento, de las condiciones, bajo las cuales están trabajando las bombas, además de ello, el sistema tiene que tener la capacidad de interpretar los datos y verificar si se encuentra en una zona donde puede afectar la disponibilidad de la máquina, así como recomendar acciones para corregir estos inconvenientes.

Un sistema experto como tal sería una solución para este problema ya que permite tomar los datos provenientes de los sensores y basado en condiciones de “Si, entonces”, pueda determinar basado en umbrales de alerta, para avisar al usuario de posibles fallas, así como sugerir acciones correctivas, el problema es que no se tienen suficientes datos como para formular manualmente los valores límite para accionar la alarma, así que se deberán buscar métodos alternativos para definirlos.

4.4.Objetivos

4.4.1. Objetivo general

- Generar un sistema experto para el monitoreo y detección de modos de falla para el sistema de bombeo y suministro de agua potable de la Sede Central del Tecnológico de Costa Rica, mediante un software computacional basado en el lenguaje Python y apoyado con un sistema de visualización de estado y alertas.

4.4.2. Objetivos específicos

- Determinar la problemática existente en el monitoreo, adquisición de datos y labores de mantenimiento presente en el sistema de bombeo, mediante un análisis en el rendimiento del sistema y la cantidad de fallas en el último año.
- Establecer los criterios que serán considerados para una selección de una monitorización adecuada y sujeta a las prioridades del equipo de suministro de agua potable a través de normativas y requerimientos de funcionamiento mediante investigaciones de expertos.
- Establecer el método de recolección y tratamiento de datos avanzado es el que se ajusta de mejor forma al proceso hacia una incorporación a la Industria 4.0 y automatización en las decisiones de mantenimiento, con base en los estándares internacionales y en un mejoramiento en la fiabilidad de los equipos.
- Comparar los costos asociados a la implementación de un sistema experto basado en sistemas computacionales avanzados como Redes Neuronales o Inteligencia Artificial, en relación con los métodos que se emplean en la actualidad para una solución de problemas tanto en mantenimiento preventivo, predictivo o correctivo.

4.5. Justificación

En el tema de la preservación en buenas condiciones de los equipos, es de importante trascendencia evitar que estos presenten problemas graves por tiempos prolongados, ya que desencadenaría en daños catastróficos o que involucren un gasto más significativo para lograr ponerlos en marcha en su funcionamiento normal, en ocasiones, incluso, resulta más rentable la sustitución en lugar de la reparación del activo.

En el mantenimiento preventivo y predictivo, su eje principal es solucionar el problema del ya costoso mantenimiento correctivo, puesto que, mediante estas técnicas, se puede actuar con tiempo suficiente (vida temprana y vida útil) antes de que se produzcan daños irreversibles (vida de desgaste) y, además, logra alertar al personal para evitar paros repentinos en el proceso donde esté involucrada la máquina, como se puede observar en la figura 4.2, con el fin de evitar que la máquina entre en la vida de desgaste.

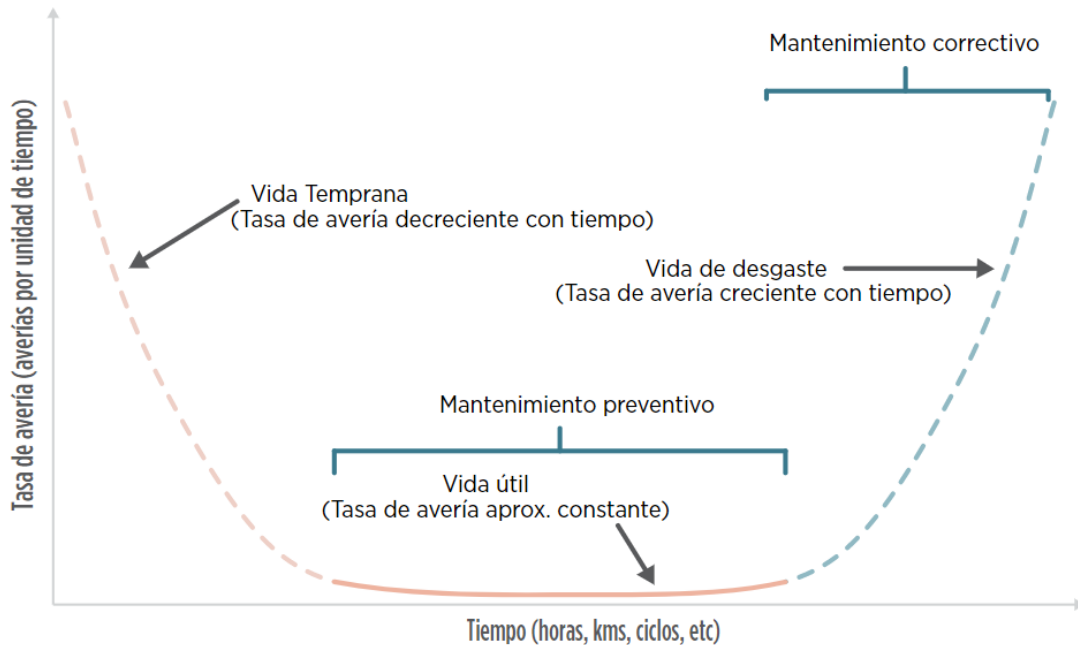


Figura 4.2 Diferencias entre el mantenimiento preventivo y correctivo. Fuente: Pastor, 2019.

Ahora bien, para cualquier institución es muy importante la reducción de gastos, sin sacrificar la utilidad o competitividad de esta, esto incluye a la sección de Mantenimiento, pues una decisión errónea, innecesaria o inadecuada puede repercutir fuertemente en los gastos y el presupuesto tanto del área de mantenimiento como en la empresa o establecimiento como tal, esto lo corrobora Ghazali (2018) en su estudio, pues dice que:

El mantenimiento de la planta también tiene un impacto peculiar en el negocio de una empresa. El evento es caro; podría drenar los recursos de la empresa, generar peligro para la confiabilidad, peligro potencial para la seguridad y tiene el riesgo de que el costo y la duración se excedan debido a la incertidumbre técnica.

Por otra parte, se necesita una herramienta que sea parte de la nueva revolución industrial como lo es la llamada “Industria 4.0”, ya que, recientemente, los avances en sistemas cibernéticos sensorizados, como el “*Machine Learning*” o Redes Neuronales, promueven la adopción de sistemas de detección temprana de fallas en cualquier equipo que se desee, con el objetivo de reducir los costos de mantenimiento y aumentar su eficiencia operativa. Esta toma de datos en conjunto con una interfaz, que permita interpretar las condiciones actuales de los sistemas o elementos y compararlos con una base de datos basado en la experiencia del personal de mantenimiento o con sistemas existentes en otros ámbitos, en conjunto con un sistema de decisiones multicriterio producirá la opción más viable y eficiente posible.

En concreto, la herramienta servirá para integrar todos los datos posibles o medibles tanto numéricamente como cualitativamente, estos últimos usualmente obtenidos de análisis visuales, y lograr tomar la mejor decisión posible basado en estos datos y revisando múltiples criterios, como, por ejemplo: ambiental, salud del equipo, vida útil restante, relación costo beneficio, entre otros.

Como se mencionó en la sección del planteamiento del problema, si se sigue con el método actual del mantenimiento y supervisión de los equipos, se tendrá una incertidumbre respecto al estado de operación de las bombas en la mayor parte del tiempo, solo 6 días en el año se sabe con certeza los parámetros bajo los cuales opera la máquina. Por ello, es necesario la implementación lo más pronto posible de un sistema que permita un monitoreo continuo de datos representativos, así como un sistema de alertas de posibles fallos (tanto inmediatos como a un futuro cercano), para planificar las intervenciones, realizar acciones correctivas o prever el cambio o refacciones de piezas si es necesario.

Cuando se refiere al mantenimiento de los equipos y más aún con la presencia de nuevas técnicas y tendencias, se debe ser proactivo, al emplear las tecnologías de la Industria 4.0 para tener un monitoreo continuo, de modo que se puedan aplicar métodos como Redes Neuronales o *Machine Learning* para que los sistemas puedan ellos mismos determinar comportamientos irregulares o parámetros fuera de rango normal de operación, convirtiéndose en expertos de su condición solo que de forma artificial.

Así, al realizar este tipo de proyectos, se tiene la ventaja de ir migrando poco a poco hacia sistemas más autónomos en la toma de decisiones y donde la intervención humana no sea un factor, minimizando las observaciones y juicios subjetivos, cambiando así la mentalidad del personal de mantenimiento hacia sistemas preventivos tradicionales y apegándose hacia la Industria 4.0, la cual es hacia donde la mayoría de las empresas buscar estar.

En este caso, debido a que no se tiene un compendio de información respecto a mediciones o parámetros de funcionamiento, se torna complicado en etapas tempranas de implementación del presente proyecto realizar predicciones precisas de cuando fallará, sin embargo, con un almacenado de datos, luego de algún tiempo de funcionamiento (algunos meses) y con datos suficientes, se puede prescindir del contrato de visitas tercerizadas bajo intervalos definidos.

Además, se podrá hacer un mantenimiento basado en condición producto de los datos almacenados, logrando extender el aprovechamiento de los recursos al tomar el máximo de vida posible sin comprometer la integridad de la bomba, pero en esta primera etapa se espera que el modelo logre determinar fallas con suficiente antelación para ser reparadas, sin afectar la vida del bien.

Cabe destacar que otra parte de las ventajas que posee este proyecto es la de servir como base para una transformación en los cursos que se brindan en la escuela, por ejemplo, los profesores Carlos Piedra Santamaría y Sebastián Mata Ortega pueden tomar la base de datos para ejemplificar e investigar sobre ellos formas de interpretación de los valores, agruparlos, buscar tendencias y formular nuevos sistemas expertos cada vez más complejos, de forma que cursos como Métodos Cuantitativos para la toma de decisiones se ejemplifiquen mediante datos reales y así adquirir experiencia en ámbitos que las empresas cada vez buscan más.

4.6. Viabilidad

El sistema de suministro de agua no cuenta con sensores inalámbricos o con equipo que permita tomar los datos y enviarlos hacia un entorno global, por lo que para la realización del presente proyecto es necesario una serie de requisitos desde el punto de vista humano, técnico y económico, puesto que se debe comenzar con un análisis de la situación actual y, *a posteriori*, formular un método de solución basado en sistemas relacionados al *Big Data*, *Machine Learning*, Métodos Difusos e inclusive podría necesitar de Redes Neuronales para una toma de decisiones autónoma.

Dado lo anterior, tomando como base la situación del país referente a la cuarta revolución industrial, es necesario una persona capacitada en el tema que pueda brindar algún tipo de referencia y asesoría del método de solución a implementar, así como su desarrollo, afortunadamente, se cuenta con el apoyo de distintos profesores del área administrativa de la carrera de Ingeniería en Mantenimiento Industrial, donde, incluso, algunos de ellos se encuentran en labores investigativas producto del doctorado enfocado en este tema en el extranjero, lo que permite contar con el requisito humano.

Por otra parte, con respecto a los costos económicos asociados y material, se tiene la ventaja de que al ser implementado en una universidad pública del país, algunos de los sensores necesarios para la toma de datos se pueden solicitar mediante colaboraciones con diversas empresas o conseguirse a un precio menor producto de dichos convenios, la parte de programación será mediante plataformas de tipo “*Open Source*” (gratuitas) y la computadora será la personal del estudiante a ejecutar este proyecto, no se necesitará la contratación o el pago de personal, dado que ninguno de los trabajadores del área de mantenimiento del campus serán los responsables de la instalación.

4.7. Alcance

El proceso de implementar sensores en diversas máquinas y elementos puede llegar a ser un tanto complicado y requerir de mucho tiempo para su puesta en marcha y calibración respectiva, además, no todos los componentes de un sistema deben poseer monitores de variables, estos se aplicarán a los elementos más críticos, tanto por su función como por su costo de fallo y reparación.

La toma de datos y el proceso de análisis donde se formularán las decisiones se aplicará al sistema de bombeo del campus central del Tecnológico de Costa Rica, contemplando las tres bombas centrífugas que se emplean para enviar el agua proveniente del municipio o la extraída del río subterráneo, mediante la bomba de pozo hacia el tanque elevado, para, luego, ser trasladada a su proceso de limpieza y desinfección.

No se abarcará todo el sistema de distribución de esta sede ni tampoco los bebederos, ya que muchos de ellos son dispositivos fabricados por terceros con un funcionamiento y tratamiento de agua que excede los límites del presente proyecto, ni tampoco la bomba sumergida de 40 HP, debido a que al estar a 150 metros de profundidad se dificulta, en gran medida, su extracción y la colocación de medidores.

En este proyecto, además de abarcar la implementación de sensores, se tomará en cuenta la formulación de un modelo que permita captar las señales y evaluar en tiempo real los datos provenientes del sistema al compararlos con unos parámetros previamente programados para enviar una respuesta hacia los actuadores, corrigiendo la posible falla de forma automática o alertando al personal de mantenimiento, indicando cuál ha sido la causa, así como el procedimiento que debe realizarse para subsanar el defecto correspondiente.

El proyecto no involucra la evaluación de la gestión de mantenimiento del departamento en todos los ámbitos y equipos que se encuentran en el campus central, sino solo hacia el sistema de bombeo, por ello, no se involucró a este departamento, ni se realizó ninguna evaluación sobre su forma de realizar mantenimientos preventivos o predictivos, ya que ellos no se involucran en el proceso del mantenimiento salvo cuando se realiza el tipo correctivo.

La aplicación de métodos basados en la Industria 4.0 se debe realizar de forma paulatina, ya que requieren de cierta inversión que puede ser alta si se pretende abarcar todos los equipos dentro de una empresa o institución, además, para que el sistema sea viable toma tiempo ajustar los sensores y dejar que el sistema pueda aprender por sí mismo, por lo que tomaría años integrar todos los sistemas existentes, por estos motivos se plantea para efectos de este proyecto solo acaparar el sistema de bombeo de agua potable.

4.8.Limitaciones

Debido a que los sistemas inteligentes y expertos necesitan de una programación avanzada, será de suma importancia contar con librerías que permita realizar la escritura de código concerniente a realizar tareas complejas y un tratamiento de datos avanzado, con el fin de proveer un entorno de inteligencia artificial mediante redes neuronales, además, el monitoreo de datos se pretende hacer mediante un prototipo de bajo costo que permita obtener mediciones de las variables más representativa, por ello, si las librerías no están a disposición o se necesitara formular librerías propias, se incrementaría significativamente la complejidad de este proyecto.

Con base en la situación actual que enfrenta el mundo respecto al SARS CoV-2, comúnmente llamado coronavirus, y ante las restricciones de operación del Tecnológico de Costa Rica, existe la limitante en la cantidad de visitas que se puede realizar para la toma de mediciones y demás datos ya que no se puede estar todos los días en la institución y el acceso debe ser autorizado y el resto se trabajará a distancia, esto conlleva a que el dinero que se puede destinar a la compra de los sensores y demás requisito sea baja, por lo que se plantea emplear componentes de bajo costo, pero que, posteriormente, puedan ser sustituidos por los de preferencia.

La comunicación efectiva entre los profesores expertos en el tema y el estudiante será de forma virtual, ya que estos deben realizar teletrabajo por disposiciones institucionales y uno de ellos se encuentra realizando estudios de doctorado en Francia, ante lo cual, si alguna situación se presentase, no existirá un acercamiento constante de forma personal y deberá acordarse una hora en específico a la cual se puede solicitar el asesoramiento y la guía a diferencia de la situación si los profesores, el estudiante y el personal de mantenimiento estuvieran en el campus.

4.9. Metodología

Con el fin de abarcar los objetivos planteados en este trabajo, se siguieron una serie de actividades bajo los siguientes criterios.

- Recopilación de información sobre el estado actual: se realizaron visitas a la estación de bombeo, con el fin de determinar la problemática existente, así como los equipos instalados y la función que realizan.
- Determinar los parámetros de mayor importancia para un monitoreo: basado en una investigación apoyado en normas y estadísticas de investigadores y fabricantes se determinará cuáles variables son las más representativas, con el objetivo de diagnosticar correctamente al equipo sin incurrir en un gasto de recursos excesivos (tanto computacionalmente como en materia económica).
- Investigar los distintos tipos de analizar los datos: determinar cuáles métodos de inteligencia artificial se ajustan de mejor forma al comportamiento o tendencia de los datos ya sea apoyado en el *Machine Learning* o en Redes Neuronales según corresponda.
- Programación del método elegido: basado en la plataforma y lenguaje de programación de Python se escribirán las líneas de código que responde a la tendencia y permita clasificar el estado de la máquina como normal o en alerta.
- Revisión de las acciones de mantenimiento: en referencia a la alarma y el tipo de parámetro que provocó la anomalía, se deberán brindar recomendaciones al personal de mantenimiento para corregir la avería a la mayor brevedad posible.
- Comprobación de funcionamiento y revisión de precisión: se debe comprobar que el aprendizaje es el correcto sometiendo al programa a datos de prueba y ratificar su precisión.

5. Marco teórico

En un entorno cada vez más virtual, es de esperar que la capacidad de las máquinas y los equipos en términos de procesamiento de datos y toma de decisiones sea cada vez mayor. Día tras día, se observa en la televisión y en internet un nuevo producto capaz de entender y hacer diversas actividades, como, por ejemplo, Alexa, Google Home, entre otros, los cuales, mediante comandos de voz, pueden activar los altavoces en el hogar para escuchar esa canción que tanto se desea.

Estos dispositivos, en su funcionamiento se basan en la Inteligencia Artificial (IA) dotando a la máquina la capacidad de tomar decisiones y a pensar cómo lo haría un ser humano al escuchar y formular una respuesta, dentro de este ámbito, existen diversas formas de hacer que un sistema como este entienda el comportamiento, uno de ellos es el uso de las Redes Neuronales (NN), las cuales algunos expertos tienden a explicarlas similar al funcionamiento del cerebro humano, donde cada neurona hace pasar su información a otra para entender comportamientos cada vez más complejos, estos conceptos son de suma importancia para el desarrollo de este proyecto y se explicarán con mayor detalle en secciones posteriores de este documento.

Ahora bien, tanto la IA como las NN son tecnologías y desarrollos que se han expandido hacia diversos ámbitos como lo son la robótica y la automatización, en el área de mantenimiento se ha visto un auge en el uso de estos como indicadores en sistema de monitoreo como medio para la alerta de posibles fallas y sus causas basados en un historial de fallas de la propia máquina, una forma común de cómo se incorporan es mediante los sistemas expertos donde el conocimiento humano se emplea como guía para resolver problemas que requerirían de expertos humanos.

5.1.Mantenimiento en la Industria 4.0

El mantenimiento suele pasar desapercibido y, en ocasiones, se puede ver como un gasto y, por ende, un departamento que debe ser reducido lo máximo posible, esta afirmación no puede estar más alejada de la realidad, ya que una avería en alguno de los procesos o equipos puede representar en grandes pérdidas económicas o un gran desembolso de dinero para ser subsanadas.

Los equipos deben permanecer en excelente estado, con la mayor disponibilidad posible para evitar paros imprevistos que afectan la industria y su producción de bienes o servicios, el término de mantenimiento industrial hace referencia a todas las acciones que son llevadas a cabo con el fin de alargar la vida útil y el buen funcionamiento de la maquinaria perteneciente al proceso de producción, existen varios tipos de mantenimiento, sin embargo, los básicos son:

- **Mantenimiento correctivo:** consiste en reparar las averías en la máquina una vez que aparecen.
- **Mantenimiento preventivo:** Intervención constante de la maquinaria para evitar que se presenten problemas (programas rutinarios).
- **Mantenimiento predictivo:** se basa en mediciones de parámetros para lograr predecir cuándo fallará la máquina y el momento óptimo para realizar una reparación.

En la cuarta revolución industrial (Industria 4.0), se hace un constante hincapié hacia el último tipo de mantenimiento, ya que con este se logra aprovechar al máximo la vida útil de los componentes, y así nunca se presentará una avería sin previo aviso.

Debido a que, en muchas ocasiones, los equipos instalados son muchos y, además, las mediciones de los parámetros deben realizarse en el sitio, el esfuerzo humano y el tiempo dedicado a esta labor resulta en un aprovechamiento de recursos innecesarios, así el departamento se vuelve ineficiente.

Con el auge en el internet de las cosas (IoT), se busca tener mediciones las 24 horas del día, y los 7 días de la semana, en cada equipo que se considere necesario para monitorear su condición, cada uno de los datos se envían a través de internet hacia un servidor o computadora donde puedan ser interpretados y analizados por alguna persona del departamento de mantenimiento, así no requiere desplazarse hacia el equipo para tomar los datos.

El sistema informático ayudará a predecir cuándo una máquina posee algún desgaste o comportamiento irregular, provocando que su vida útil se vea comprometida. Las tendencias y valores de alerta serán modificados con el paso del tiempo, haciendo que el sistema sea cada vez más refinado y pueda dar predicciones más precisas y exhaustivas, lo cual, si lo realizara una persona, tendría que estar muchísimo tiempo observando la máquina sin parar y tomando nota de los parámetros, algo que para una computadora no supone ningún inconveniente.

Como menciona EVTECH (2020), el análisis de la *Big Data* es el principal fundamento del mantenimiento 4.0, debido a que en él se encuentra el historial de fallas, entorno, uso, comparaciones hacia equipos similares y cualquier cosa que tenga que ver con el activo.

Para que el sistema de mantenimiento pueda aprender con el paso del tiempo requiere de programas que puedan modificar sus parámetros y constantes para adecuarse cada vez más a la tendencia de la máquina, esta parametrización estándar asociada a diversos sensores, logra que, si se altera una señal, el personal de mantenimiento reciba una alerta temprana para intervenir, evitando paradas inesperadas, como lo menciona EVTECH (2020): “La implementación de la inteligencia artificial hará posible detectar patrones que escapan al ser humano más perspicaz, que nos ayudará a mantener a tope la confiabilidad de los equipos”.

Tanto es el avance de la Industria 4.0, que los fabricantes de los dispositivos o captadores de parámetros (sensores) ha mirado hacia nuevos rumbos aprovechando el internet y la intercomunicación de datos (*Big Data*), como lo indican Basco et al. (2018):

El fabricante ha cambiado su modelo de negocio; no sólo vende maquinaria para la industria, sino también servicio de detección de fallas, detección de niveles críticos de insumos, actualización del software, información y mantenimiento del hardware, brindándole al cliente la tranquilidad de que sus equipos están controlados y operan siempre bajo condiciones de seguridad e idoneidad.

A modo de ejemplo bajo la afirmación anterior, se toma a la empresa metalúrgica alemana Trumpf, cuyo negocio principal fue, durante mucho tiempo, la fabricación de herramientas y maquinarias para la industria. A partir del surgimiento de la cuarta revolución industrial, la empresa cambió su modelo de negocios y encontró la oportunidad para construir un producto nuevo basado puramente en software y datos: se trata de una plataforma llamada “Axoom IoT”, que conecta las máquinas construidas por Trumpf y otros fabricantes, y utiliza los datos que recoge de éstas para ayudar a los clientes a organizar su producción. Basco et al. (2018) también menciona a modo de cierre que:

[...] la combinación de tecnologías como los sistemas ciber físicos, la IoT, la computación en la nube y Big data, permiten obtener información en tiempo real, procesarla y analizarla, dejando atrás el modelo de prevención (aplicado al mantenimiento de equipos, a la demanda de insumos, al comportamiento del mercado) y permitiendo nuevas formas para predecir sucesos.

Estos modelos predictivos se basan en la información proveniente de los sensores instalados en la empresa (sistemas ERP, CRM, SCM y equipos con conexión M2M), o de fuentes externas, como productos inteligentes que están en posesión de los clientes y se encuentran conectados a Internet.

5.2.Inteligencia artificial

Según Turban (2007), la inteligencia artificial es un área de la ciencia computacional capaz de emular el funcionamiento del pensamiento humano, sobre este término convergen una serie de ideas, pero la mayoría de los expertos coinciden en que su desarrollo se basa en dos conceptos principales:

- Entender el comportamiento de los humanos y su forma de pensar, con la finalidad de proveer una definición bastante concreta de qué es la inteligencia.
- Enfrentar el desafío de representar y duplicar el procesamiento de datos como número e imágenes similar a como lo haría el cerebro humano mediante máquinas como computadoras y robots.

Con base en lo anterior, se podría mencionar que una definición para la IA serían las cualidades de comportamiento y acciones que posee una máquina, donde, si en lugar de ella las realizara un ser humano, se le podría catalogar de inteligente, aunque otros autores mencionan que puede ser un campo de estudio sobre como las computadoras realizan acciones donde por el momento las personas son mejores.

Uno de los ejemplos más famosos de aplicación de la inteligencia artificial es la máquina desarrollada por IBM “Deep Blue”, la cual es un programa que tiene la capacidad de jugar ajedrez, pero la diferencia es que sus competidores son jugadores profesionales, este sistema

logra aprender de cada juego y formula tácticas para vencerlos, llegando, incluso, a vencer al campeón mundial en su momento Garry Kasparov a quien usualmente solo le ganaban personas que tenían un intelecto bastante alto.

Existen diversas pruebas que se le realizan a las computadoras y múltiples sistemas para determinar el grado de similitud que tienen respecto a la forma de pensar del ser humano, como, por ejemplo, la famosa y controversial Prueba Turing, diseñada por el propio Alan Turing en 1950, cuando se colocaba una computadora y una persona a entablar una comunicación mediante texto en recintos separados, con la finalidad de medir cuanto tiempo tardaba la persona en darse cuenta de que estaba hablando con una máquina (Oppy y Dowe, 2020).

Ahora bien, es común realizarse la pregunta de ¿cuáles habilidades o comportamientos se consideran como un signo de inteligencia? La respuesta a esta interrogante la formuló Turban en su libro publicado en el 2007, donde identificó nueve conceptos que se detallan a continuación.

- Aprender o entender de las experiencias que se han vivido a lo largo de la vida.
- Tomar el sentido de mensajes o comunicados ambiguos o dudosos donde se puede incluir enunciados contradictorios.
- Rápida capacidad de respuesta frente a una nueva situación que satisfaga el problema donde puede implicar distintas respuestas o no, a esto se le conoce como flexibilidad al entorno.
- Hacer uso de la razón en la solución de problemas y una conducta efectiva.
- Lidiar con situaciones confusas o intrincadas.
- Entender e inferir el contexto en un sentido racional ordinario.

- Aplicar el conocimiento atribuido para manipular el entorno hacia un beneficio que puede ser personal o colectivo.
- Pensar y razonar sobre múltiples situaciones.
- Reconocer la importancia relativa de distintos elementos en una situación.

5.2.1. Características de la inteligencia artificial

La inteligencia artificial tiene ciertas características que ayudan a describir su comportamiento y funcionamiento, entre ellas, se encuentra el procesamiento de los datos, ya que el pensamiento de las personas comúnmente se basa en el reconocimiento de símbolos como lo son: la escritura, los números, las imágenes, patrones, el entorno como tal, entre otros más, sin embargo, la capacidad de las máquinas reside en emplear operaciones aritméticas basadas en números (procesado numérico), por lo tanto, la forma en que una computadora interactúa es mediante cálculos matemáticos para formular una respuesta.

A raíz de lo anterior, surge la necesidad de transformar numéricamente los objetos o texto según el contexto, con el fin de que una computadora pueda realizar tareas que requieran de este tipo de símbolos. Muestra de ello es la capacidad que tienen diversos dispositivos para el reconocimiento de escena donde su principio es emplear numéricamente los colores en cada píxel. Los píxeles son los elementos corresponden a la mínima unidad de cada imagen, la composición radica en la combinación de tres colores azul, rojo y verde o comúnmente llamado RGB, por sus siglas en inglés, cada color posee un grado de pigmentación que va desde los 255 (máxima pigmentación) hasta el 0 (no hay presencia de ese color), la unión de estos píxeles forma una imagen, entre más píxeles existan más grande es la imagen o también más definida será.

Otra de las características, corresponde al uso de heurística y algoritmos, la primera se define, según Beekman (1999), como “una heurística es una regla empírica donde a diferencia de los algoritmos rápidos y precisos, conduce hacia juicios que, según se indica la experiencia es probable que sean reales”.

El autor también cita algunos ejemplos que se utilizan en el día a día, como, por ejemplo, “para aflojar una tapa pegada, suméjla en agua caliente”, en este caso, otras personas han tenido esa experiencia y, a pesar de que otras no lo han realizado antes, se creen en estos principios y se ponen en práctica para lograr solucionar el problema, en el caso de ejemplo, es desatorar una tapa. Otra forma de ver la heurística corresponde a conocimiento de tipo intuitivo, o a las tan mencionadas reglas de dedo empleadas por expertos en la solución práctica y rápida de algún contratiempo, estas no siempre resultan ser la solución definitiva, pero logran atenuar en gran medida.

La principal ventaja de aplicar la heurística es que no se tiene que estar pensando o iterando en el caso de las computadoras cada vez que se presenta un problema similar, ya que aplicaría el mismo método en su resolución o, en ocasiones, busca dentro de su conocimiento el problema más similar y la solución aplicada, de este modo se forma lo que se conoce como “heurística de vecino más cercano”, simplificando y acelerando la dificultad y el tiempo en proveer un desenlace satisfactorio al dilema.

Finalmente, y tal vez la característica más importante, es la capacidad de aprender que tienen los seres vivos, emular esto en una computadora o máquina resulta bastante complicado, puesto que se debe dotar de sistemas que permitan un reajuste en su conocimiento basado en sus experiencias con el entorno, pero, al mismo tiempo, deber ser suficientemente flexible para aplicar ese conocimiento en otros ámbitos.

A raíz de lo anterior, surge la famosa técnica de *Machine Learning*, donde, mediante distintas tácticas programables, se puede hacer que una máquina reescriba su propio código para ajustarse a las necesidades basado en su propia experiencia, dentro de esta categoría, se encuentran diversos métodos para aprender los cuales se detallan más adelante.

5.2.2. Campos de aplicación de la IA

El uso de la inteligencia artificial es bastante amplio, algunos de los ejemplos se pueden observar en la figura 5.1, donde en la parte superior del árbol se mencionan sus campos de aplicación o especialización, el reconocimiento de la voz es uno de ellos, además de ser uno con el que más contacto se tiene, ya que esta tecnología se observa hoy en día en asistentes digitales como Siri de Apple, Cortana de Microsoft, Alexa en Amazon o el Google Assistant en el entorno Android.

Los otros campos que están en la copa del árbol son, algoritmos genéticos, sistemas expertos, *Machine Learning*, Redes Neuronales o tutorías inteligentes, donde el sistema se encarga de enseñar a otras máquinas o a las personas, robótica, como lo es ASIMO, el robot humanoide creado por la empresa Honda que emula a un ser humano en su andar y pensar, juegos de video logrando simular como si se estuviese jugando con otra persona, visión computacional, programación automática y lógica difusa encargada de transformar lenguaje ambiguo en matemático en la toma de decisiones.

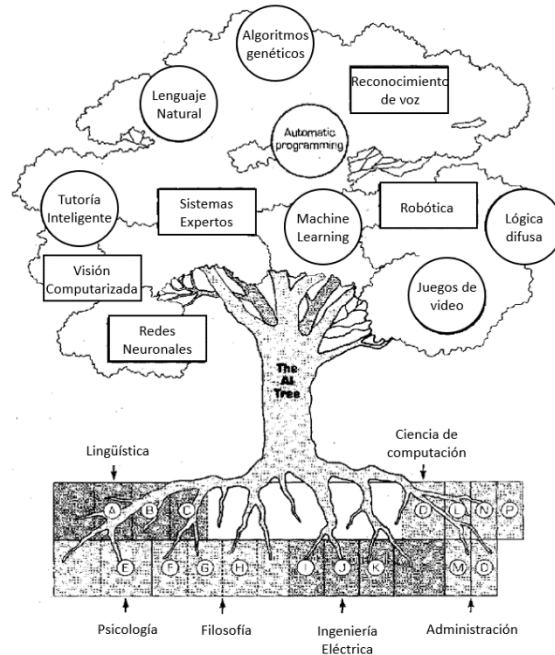


Figura 5.1. Bases y aplicaciones de la inteligencia artificial. (Traducido por Antony Alfaro). Fuente: Turban, 2007.

5.3. Machine Learning

El campo del *Machine Learning* es una rama de la inteligencia artificial que se enfoca en la construcción de aplicaciones, en las que se logre aprender de una base de datos, con el fin de mejorar la precisión con el paso del tiempo en referencia a cuando fue programado, sin la necesidad de depender de una reescritura de código, es decir, el sistema logra reescribirse obtener resultados cada vez más precisos.

Como lo menciona IBM Cloud Education (2020), en la ciencia de datos, donde se ve involucrado el campo de *Machine Learning* los algoritmos son entrenados para encontrar patrones y características dentro de la gran base de datos que se le proporciona al inicio para lograr tomar decisiones y predicciones cuando se le suministran nuevos datos, cuanto mejor sea la programación del algoritmo de solución, mayor precisión tendrá el sistema y mejorará cada vez que los datos sean evaluados.

Un ejemplo clásico de la aplicación de este método es el filtrado de correo electrónico, con el fin de encontrar mensajes no deseados y colocarlos en una carpeta que no se muestre al cliente o usuario, para así no entorpecer su labor o llenar la bandeja de entrada con promociones u otros correos, desplazando los importantes. Cuando el correo es enviado a la denominada “*spam*”, es posible tomar un mensaje de esta carpeta indicando que este “no es un correo no deseado”, cada vez que se pulsa a este botón, el sistema analiza el cuerpo de mensaje, de esta forma, cuando ingrese una nueva correspondencia, se verificará la similitud que tiene con el correo que recién se ha tomado de la carpeta y si posee la suficiente coincidencia no se enviará a la correspondencia no deseada.

En el desarrollo del *Machine Learning*, IBM (2020) recomienda los siguiente cuatro pasos, con el fin de crear un algoritmo satisfactorio, pues esta guía ha sido empleada por distintos científicos de datos en conjunto con los profesionales de negocios a quienes el modelo estará siendo entregado. Sin embargo, las opciones de desarrollo podrían modificarse en cuanto el profesional a cargo comience a adquirir experiencia y formule sus propios pasos y metodología, pero, al comenzar estos pasos, funcionan como una buena guía.

- 1. Selección y preparación del set de datos:** en el proceso de la selección de datos a emplear como insumos para el aprendizaje del sistema se debe tener cierto cuidado, ya que, si se eligen criterios o variables que no son representativas, se podrían presentar incongruencias en el resultado de la predicción del método. Se recomienda colocar, al menos, 2 tipos de set de datos, uno para el entrenamiento y otro para la prueba de la precisión del modelo luego de pasar por el aprendizaje. El primer set de datos puede no provenir directamente del sistema que se desea solucionar, sino, de fuentes que son similares al equipo o proceso por resolver.

También, los datos que se proveen pueden ser con etiquetas o sin ellas y lo más aleatorio posible con el fin de abarcar la mayor cantidad de posibilidades.

2. Selección del algoritmo o modelo para el Machine Learning: la elección del modelo depende de la cantidad y tipo de datos que se posean y el tipo de problema que se desea resolver, Microsoft menciona que, además, se debe definir ¿qué se quiere hacer con los datos?, y tomar en cuenta la precisión, tiempo de entrenamiento que se desee, así como si el comportamiento es lineal o no. Generalmente, los modelos se agrupan en tipos como clasificación, regresión y agrupación o clústeres, en el primer grupo se encuentran: regresión logística de una o múltiples clases, bosques de decisión de dos o más clases, máquina de vectores de soporte, árboles de decisión, por citar los más representativos.

En el segundo grupo se encuentra: regresión lineal, regresión multinomial, regresión de bosque de decisión y, finalmente, en el último grupo, está la agrupación por los K-vecinos más cercanos, este método se basa en la distancia euclídea para encontrar los datos que se encuentren más cerca del punto central de datos y formar un grupo. En la figura 5.2, se observa un ejemplo de la regresión logística y la lineal, en el primer caso se muestra que la línea de color verde separa los dos conjuntos de datos (color azul en la parte superior y color rojo en la parte inferior).

En el caso de la regresión lineal, se emplea cuando se busca predecir el valor de una variable dependiente basado en la independiente, ambos métodos poseen campos de aplicación distintos, pero muchos confunden su aplicación por ser regresión.

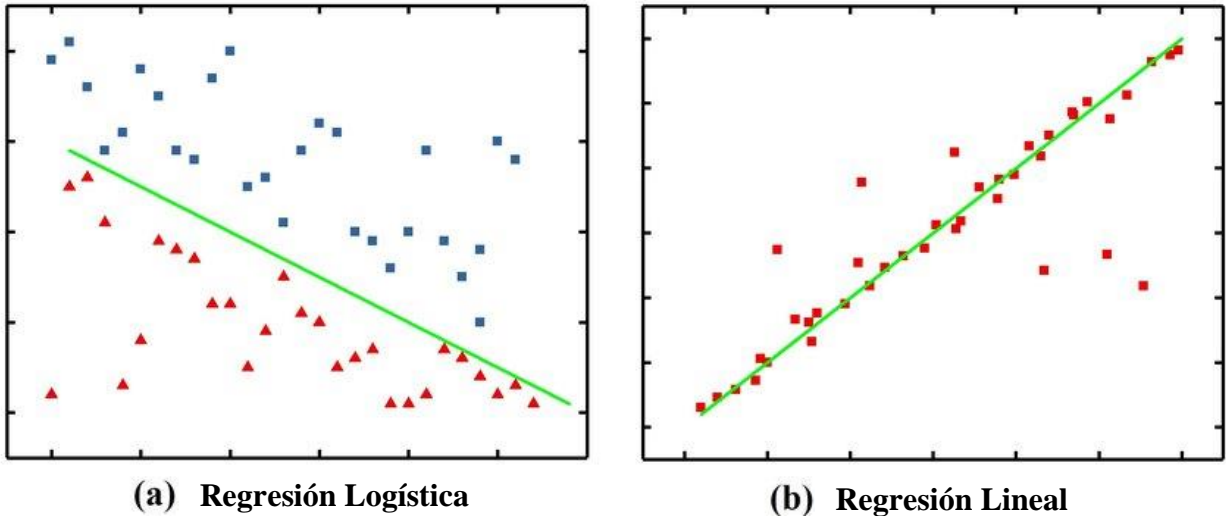


Figura 5.2. Regresión Logística y Lineal. Fuente: Fang, 2019.

Una variación de la regresión logística corresponde a la máquina de vectores de soporte (SVM) el cual emplea un hiperplano para la separación de los datos, de acuerdo con Amat Rodrigo (2017), “en un espacio p -dimensional, un hiperplano se define como un subespacio plano y afín de dimensiones $p-1$. El término afín significa que el subespacio no tiene por qué pasar por el origen”.

Por tanto, si se basa en la cantidad de variables que se tiene el problema en un espacio de dos dimensiones, el hiperplano tendrá 1 dimensión, es decir, una recta. Si se comienzan a incrementar las dimensiones esta recta se transformará en distintas formas, como, por ejemplo, un espacio tridimensional, un hiperplano tendrá dos dimensiones, ahora la recta se convierte en un plano convencional y así sucesivamente.

La principal ventaja de este método reside en que, si el espacio entre los dos conjuntos a separar es relativamente grande, la cantidad de líneas rectas o planos que los puede dividir son infinitas, pero no estarán equidistantes entre los clústeres, SVM busca el clasificador óptimo de separación, que será el hiperplano que se encuentre más alejado de los datos de entrenamiento, tal y como se

observa en la figura 5.3, donde la distancia perpendicular de cada observación al plano debe ser la mayor posible.

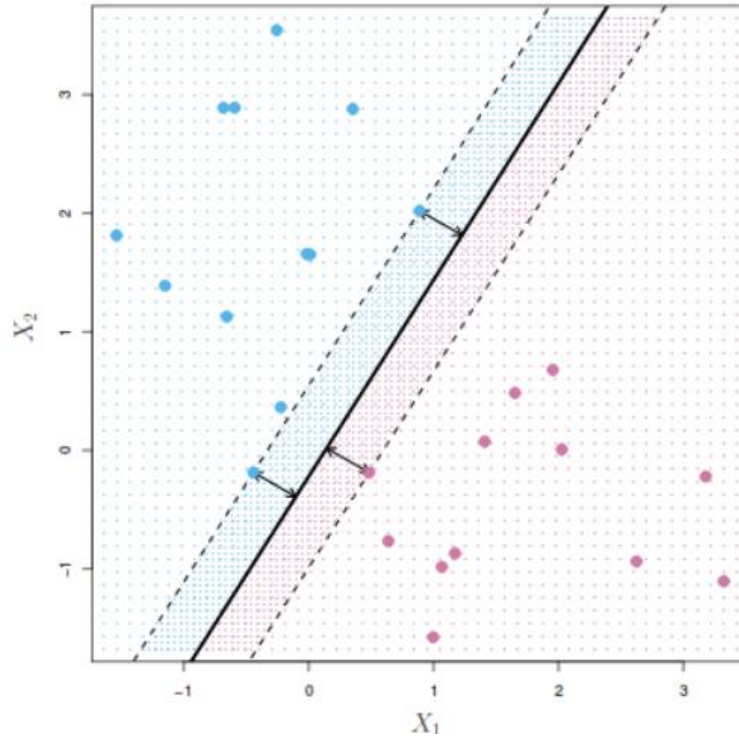


Figura 5.3 Hiperplano en la máquina de vector soporte. Fuente: Amat Rodrigo, 2017.

En el último grupo, se mencionó el método de K-vecinos más cercanos (KNN), este modelo permite un agrupamiento de datos basado en la cercanía entre ellos como se observa en la figura 5.4, donde los datos se asociaron y representaron por medio de colores azul, naranja y verde, una vez formados los clústeres el dato nuevo que ingresa al sistema (en la Figura 5.4, se muestra como X_e) coloca en posición y por medio de las coordenadas se puede identificar la cercanía hacia cada uno de los otros datos que ya fueron agrupados, luego, se busca cual grupo posee con mayor frecuencia la menor distancia euclídea y se incorpora la nueva observación.

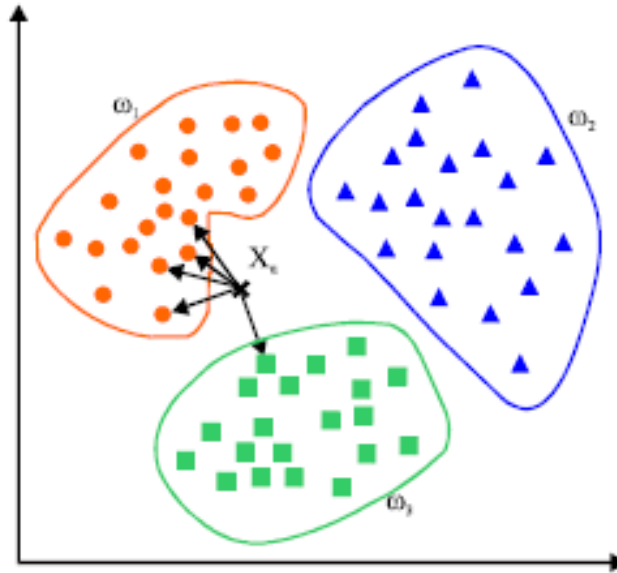


Figura 5.4 K-vecinos más cercanos. Fuente: Mahmoud Afifi, 2020.

En el método anterior, el proceso de aprendizaje es un tanto diferente, pues se puede decir que el sistema comienza a aprender en el momento que clasifica los datos, ya que debe evaluar cada incorporación basado en reglas como la mínima distancia euclídea, la cual es la mínima distancia perpendicular de un punto a otro, si se tuviese por ejemplo dos ejes o características (w y b) la ecuación sería:

$$d = \sqrt{\Delta w^2 + \Delta b^2} \quad (1)$$

Donde Δw corresponde a la distancia entre puntos en el eje w y Δb es la distancia en el eje b

3. Entrenar el algoritmo para crear el modelo: una vez elegido el algoritmo más adecuado para el problema, se procede a alimentar al método con los datos de entrenamiento antes mencionados y, de forma iterativa, se ajustarán las variables necesarias comparando la salida esperada con la generada, la cantidad de iteraciones puede ser limitada para evitar el sobreentrenamiento (*overfitting*) que sucede cuando el sistema se ajusta perfectamente a la serie de datos pero no a otros casos destinados a funcionar. El resultado pasa de ser un algoritmo

a un modelo como tal, ya que ahora las variables han sido modificadas para un problema en específico.

- 4. Uso y mejoras continuas al modelo:** una vez que se crea el modelo ya está listo para su uso, sin embargo, el sistema no puede quedarse con solo un entrenamiento, puesto que los procesos deben actualizarse y aprender la mayor cantidad de comportamientos que se logran mediante un mantenimiento periódico. Un ejemplo de esto es un robot barredor, el sistema de comprender que un objeto se ha cambiado de lugar, lo mismo sucede con los autos con piloto automático, el comportamiento de las personas es distinto cada día, de modo que debe comprender cuando frenar y acelerar sin provocar incidentes y esto se logra aprendiendo en cada conducción.

Por otra parte, la forma de aprender de los algoritmos puede ser de dos tipos, supervisado o no supervisado. En el primero de ellos, el modelo aprende por sí mismo, pero mediante datos que son debidamente identificados o etiquetados, por ejemplo, si se desea que un algoritmo aprenda a determinar cuándo una imagen es un gato, debe suministrársele una serie de fotografías o dibujos con su respectiva etiqueta indicando si es o no un gato, generalmente este tipo de aprendizaje requiere menos cantidad de datos. El segundo método utiliza datos sin etiquetar, pero requiere una cantidad muy grande de ellos, además, no requiere de la intervención del ser humano para aprender.

5.4.Redes Neuronales

La idea de proveer a un sistema la capacidad de análisis de un ser humano siempre ha estado latente entre los científicos e ingenieros, con el fin de crear seres artificiales, con un nivel de inteligencia similar o incluso superior al de las personas, pero la forma en que un cerebro procesa los datos y la información mediante las neuronas es realmente compleja, Nielsen (2019) en su

libro, menciona que un simple ejemplo como la visión requiere de más de 140 millones de neuronas para procesar una imagen y descifrar su contenido.

Basados en la idea del funcionamiento cerebral, se desarrollan los perceptrones entre 1950 y 1960 por el científico Frank Rosenblatt, un perceptrón toma dos o más entradas binarias para producir una única salida, también del tipo binario, para hacerlo utiliza pesos para cada entrada, estos son números reales que representan la importancia de la entrada respecto a la salida, en la figura 5.5 se representan con la letra w.

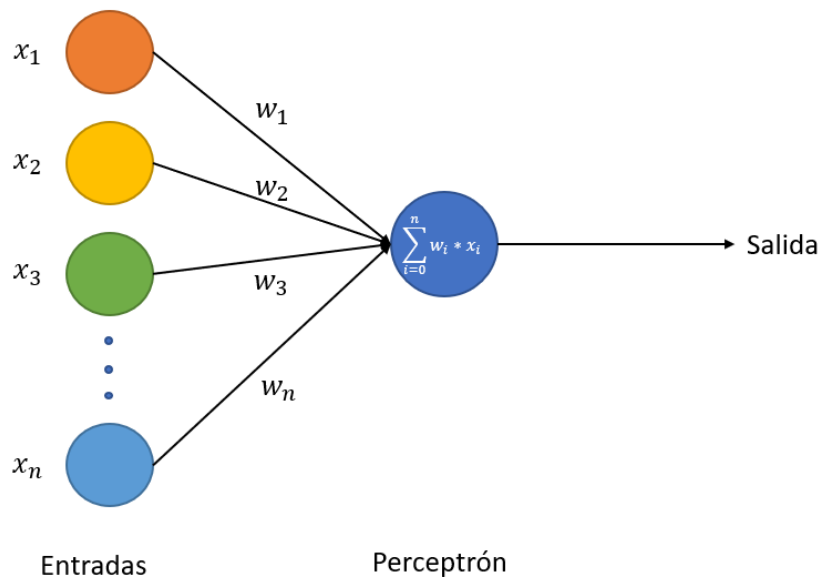


Figura 5.5. Perceptrón de una red neuronal. Fuente: Elaboración propia.

La forma en la que la salida es calculada se da mediante la suma de cada entrada con su respectivo peso, luego de que se realiza esta operación se debe colocar un valor umbral o límite en donde si la salida es mayor, se convertirá en un 1, mientras que si es inferior se convertirá en un 0. De este modo, al realizar cambios en las constantes y el valor límite de decisión, se puede aplicar a otro problema ya que el modelo ha cambiado. Otra forma de llamar a un perceptrón es neurona ya que

toma una gran cantidad de entradas, interpreta su importancia (mediante los pesos) y, finalmente, hace una elección.

Si se desea incrementar la capacidad y complejidad del análisis de los datos de entrada, se puede emplear múltiples perceptrones para formar una red donde las entradas serán las salidas de la neurona anterior, se le llama capa cuando en un mismo orden (se muestran en columnas para mayor facilidad) existen una o más neuronas, como se muestra en la figura 5.6, en esta imagen, los primeros tres perceptrones harán decisiones relativamente simples, mientras que los 4 siguientes como toman los datos ya procesados de la primera capa de neuronas, logrando formar análisis más complejos y abstractos, por realizar un segundo refinamiento de las entradas.

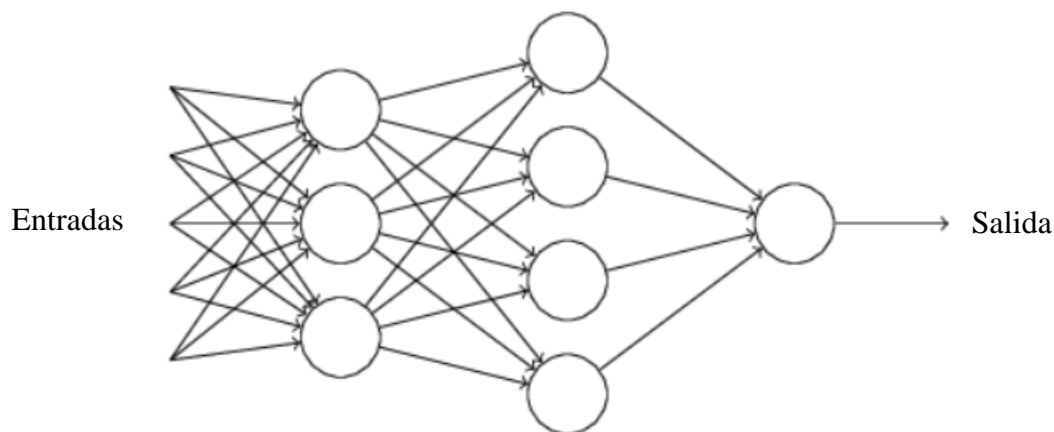


Figura 5.6. Múltiples neuronas conectadas en forma de red. Fuente: Nielsen, 2019.

Si se tuviese solo la expresión de multiplicar la entrada por la constante de peso, la convergencia sería un tanto lenta y la forma de definir el umbral para cada una de las neuronas tendría que ser manual, ya que se debe ajustar al valor que encuentre luego del proceso iterativo, sin embargo, esto no es del todo fiable, porque, en ocasiones, no se conoce el valor que se ajusta de la mejor forma al comportamiento del problema, y lo que se busca es que el modelo logre comprender por

su cuenta las tendencias y el comportamiento que define en mayor medida los datos provistos de aprendizaje, razón por lo que la ecuación pasa a ser la siguiente:

$$O = w_i \cdot x_i + b \quad (2)$$

Donde, O corresponde a la salida de la neurona, w_i es el peso respecto a la entrada i , x_i es la entrada i , y b será una constante que ayuda a definir qué tan fácil es hacer que la salida del perceptrón sea un 1, por ejemplo, un valor de b muy alto positivo significaría que prácticamente con cualquier valor de entrada o de constante w la salida será 1, ahora por el contrario si el valor es muy negativo es prácticamente un hecho que la salida será un 0. El hecho de que estas variables puedan ser modificadas (w y b) para cada una de las neuronas hace que el proceso de aprendizaje sea iterativo y más completo sin la necesidad de la intervención del programador.

De igual forma, como se observa en la figura 5.4, la salida de las neuronas será la suma de la multiplicación de cada una de las entradas con su respectivo peso, pero ahora se le debe sumar la constante b de esa neurona para lograr formular una salida más representativa y de fácil convergencia

5.5.Red neuronal profunda (Deep Neural Network)

El uso del aprendizaje profundo (*Deep Learning*), como se muestra en la figura 5.7, comienza a aplicarse cerca del 2010 esto, no significa que la idea surgiese por esta época, de hecho, las bases se desarrollan mucho tiempo atrás, el problema que surgió es que como se mencionó en la sección anterior si se desea desarrollar un proceso iterativo con una gran cantidad de neuronas, cada una modificando su propio peso y parámetros requiere una gran capacidad de procesamiento que antes no se tenía.

El desarrollo de las tarjetas gráficas capaces de realizar un alto número de cálculos aritméticos por segundo hace que no solo sea el procesador el encargado de procesar y modificar el algoritmo hasta alcanzar el modelo y, en recientes años, cada vez más personas son capaces de costearse una computadora con altas prestaciones posibilitando el desarrollo de nuevas herramientas y métodos mucho más sofisticados.

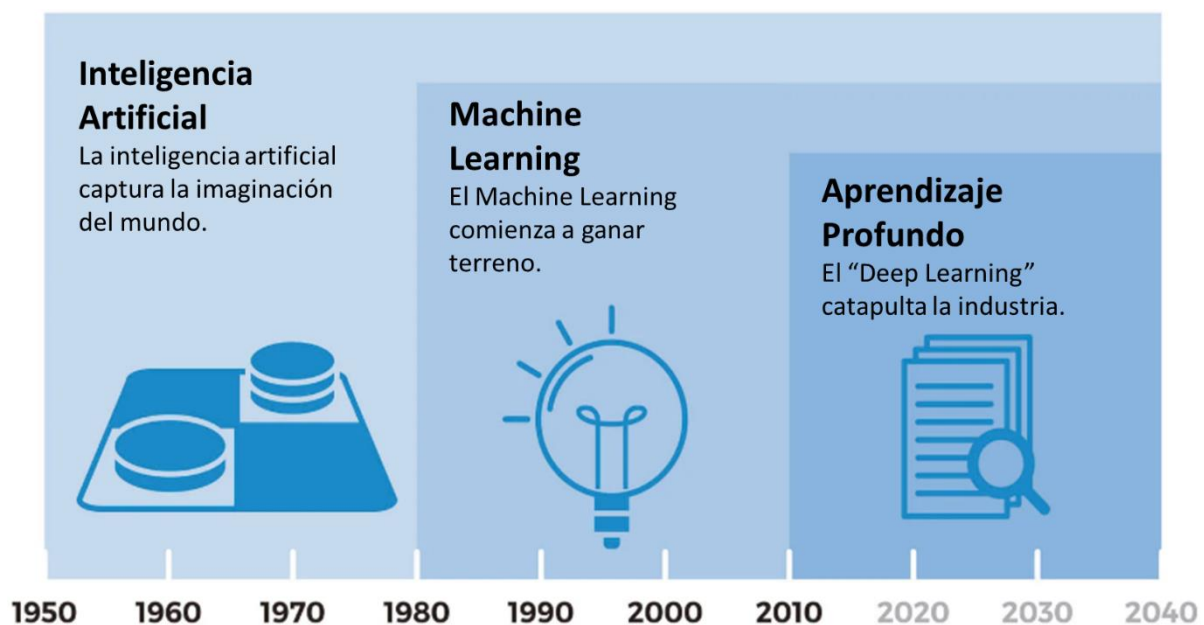


Figura 5.7. Surgimiento del Aprendizaje Profundo. (Traducido por Antony Alfaro.) Fuente: Gorini, 2019.

La principal diferencia entre una red neuronal convencional y una profunda consiste en la cantidad de capas ocultas que posee, se les llama así dado que la salida de las neuronas no se logra observar, de modo que funcionan como una caja negra donde se realizan los cálculos y solo se pueden identificar las entradas y la salida de todo el sistema, una red para que se considere

profunda debe poseer 3 o más capas ocultas, en la figura 5.8, se muestra una representación donde estas capas están identificadas con la leyenda “capas ocultas”.

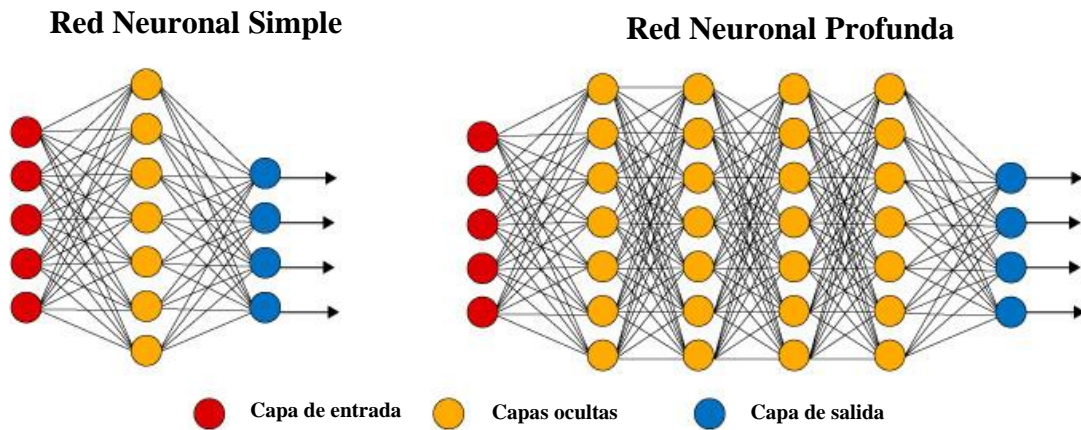


Figura 5.8. Red Neuronal Profunda y Red Neuronal simple. Fuente: Kampakis, 2018.

Ahora bien, si se desea hacer que un sistema aprenda por la modificación de variables como pesos o las constantes “b”, dado que la salida de un perceptrón solo puede ser 2 valores (0 o 1) resulta difícil que una pequeña variación produzca una salida más o menos aproximada, ya que un pequeño cambio en una de las primeras neuronas en las primeras capas podría afectar al resto mediante una reacción en cadena logrando que otros de los conocimientos sean modificados completamente, perdiendo la precisión.

Con el fin de erradicar el problema binario en la salida de un perceptrón, se formulan las funciones de activación, estas toman el valor de la ecuación lineal del perceptrón, pero en lugar de aplicar el valor umbral para hacer la salida un 1 o 0, la normalizan para que tome ciertos valores no binarios.

Según MissingLink (2020), las funciones de activación también proveen al sistema la capacidad de determinar la precisión y la eficiencia computacional de entrenamiento del modelo, logrando mejorar la capacidad y velocidad de convergencia, haciendo posible que altos números de

neuronas e iteraciones puedan llevarse a cabo con mayor practicidad. Los valores de salida de una función de activación tomarán valores entre -1 y 1 o 0 y 1, dependiendo de la ecuación elegida, donde incluso pueden variar estos rangos.

5.5.1. Función sigmoide

La primera de las funciones es la de tipo sigmoide, su representación se muestra en la figura 5.9, como se observa, este tipo de ecuación está restringida entre 0 y 1 con la ventaja de que no existen saltos súbitos entre los dos estados, sino que, por el contrario, existe una pendiente que abarca cualquier número de este rango, lo que lo hace sumamente efectivo cuando de clasificar se tratase. La forma en que crece desde 0 hasta 1 hace que las predicciones sean bastante claras (con diferencia entre probabilidades), ya que lo hace de forma relativamente acelerada, lo cual también deriva en el problema de pequeñas variaciones en valores altos de z no se ven reflejados en a .

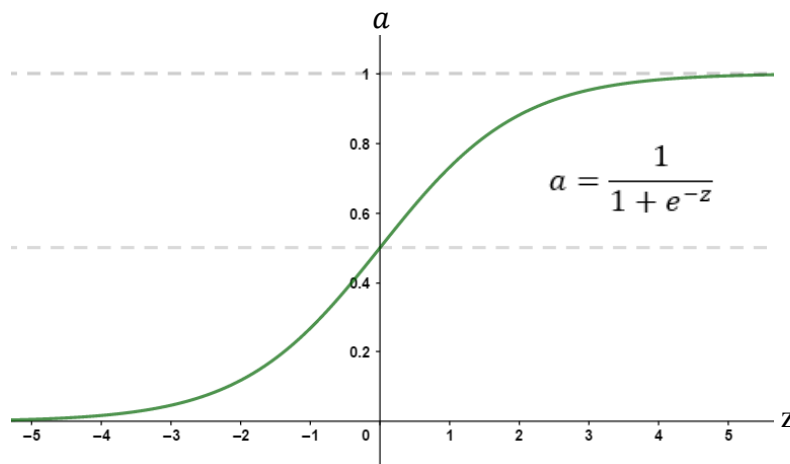


Figura 5.9 Función Sigmoide. Fuente: Elaboración propia.

5.5.2. Función ReLU

Se le llama ReLU por el acrónimo de su nombre en inglés (*Rectified Linear Unit*), se basa en la función que se muestra en la figura 5.10, en ella, si el valor de la salida de la neurona es negativo

ReLU lo convierte en un 0, mientras que si el valor es positivo, el resultado, después de aplicar esta función de activación será el mismo valor, de este modo, es posible cambiar la linealidad de un sistema y hacerlo no lineal, si bien es cierto que solo existen dos tipos de pendientes en esta función basta para que cada neurona sea diferente, porque esta varía en múltiples puntos del algoritmo y logre adecuarse al comportamiento, como resultado el sistema aprende.

Esta función tiene como inconveniente el punto de inflexión que se encuentra en el cero absoluto pero la probabilidad de que una salida de una neurona sea exactamente 0 es muy baja, si en los datos existen muchas posibilidades en la entrada de la función de activación sea negativa, el sistema no podrá aprender y la red no tendrá funcionalidad, para ello, existe una versión modificada llamada “Leaky ReLU”, donde la parte negativa no es cero, será una constante menor que 1 multiplicada por el valor de entrada (z en la figura 5.10).

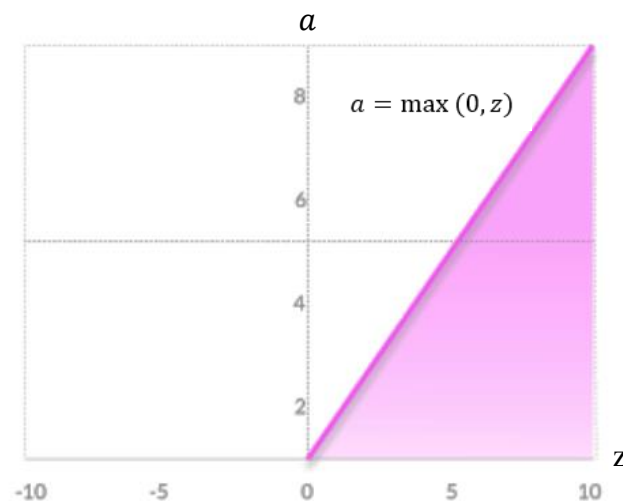


Figura 5.10. Función ReLU. Fuente: MissingLink, 2020.

5.5.3. Función tangente hiperbólica

El problema que surgía con la función sigmoide con los datos que son negativos, así como la forma en que está centrado (0.5) no son viables para algunos sets de entrenamiento, a modo de

resolver estos inconvenientes, surge la tangente hiperbólica mostrada en color rojo en la figura 5.11, en ella, también se detalla la expresión matemática que la identifica. En este caso, al estar centrada en cero hace que los valores que son muy negativos, muy positivos o neutrales sean más fáciles de caracterizar, ya que, además, los valores pueden estar entre -1 y 1, ampliando el rango respecto al sigmoide.

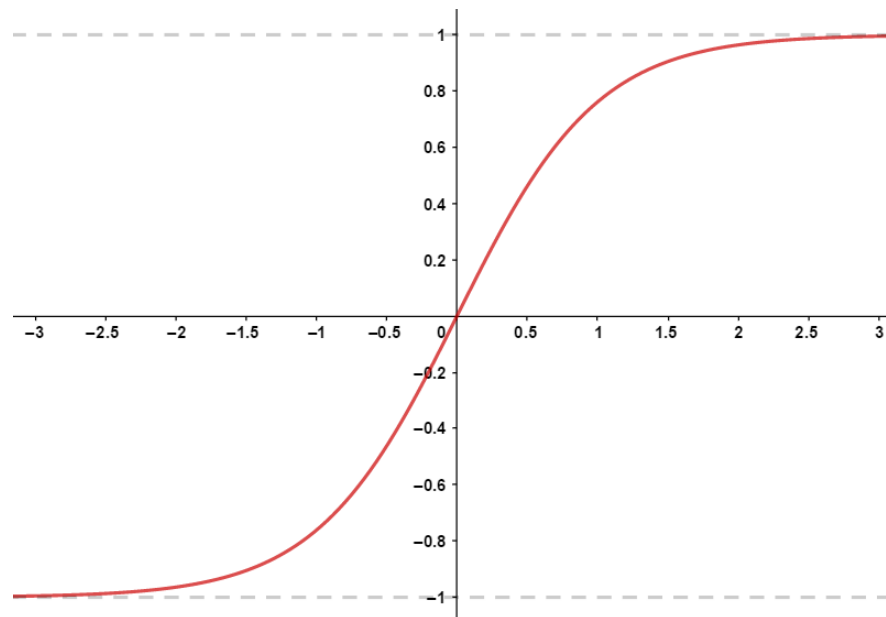


Figura 5.11 Función tangente hiperbólica. Fuente: Elaboración propia.

En algunas ocasiones, cuando se tiene un modelo donde requiere una clasificación en distintas salidas es conveniente la función llamada Softmax, la cual está fuera de los límites del presente trabajo, pero se menciona para siguientes mejoras que al incluir más datos relacionados y buscar una clasificación múltiple, sin embargo, se recomienda que solo sea empleada en la capa de salida.

5.5.4. Función de error (Cost and Loss Function)

Cuando el modelo realiza una predicción, la salida de la última neurona luego de la función de activación a la que se denominará con el símbolo \hat{y} , debe compararse con la salida real o esperada (y), con el fin de verificar la precisión del modelo, para realizar este procedimiento se define la

función de costo o error. A menudo, se utiliza la diferencia de cuadrados como ecuación, sin embargo, tal y como lo menciona Andrew Ng en su curso de Redes Neuronales no es viable el uso de esta, ya que, cuando el sistema comienza a iterar el comportamiento no es del todo convexo y podría dar lugar a errores en el cálculo, porque se forman muchos puntos óptimos locales y no globales, impidiendo el correcto funcionamiento de la función de descenso de gradiente (*Gradient Descent*).

Dado lo anterior, es necesario encontrar una ecuación que permita encontrar un mínimo global para que exista la menor diferencia posible entre los datos reales y los predichos por el modelo, dando como resultado la siguiente ecuación.

$$L(\hat{y}, y) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \quad (3)$$

La ecuación anterior es válida para un solo ejemplo de aprendizaje, si, por el contrario, se tiene más de un set de ejemplos se debe realizar un promedio de cada una de las salidas para cada ejemplo, transformando la ecuación anterior como se muestra a continuación y pasará a llamarse función de costo J .

$$J(w, b) = \frac{-1}{m} \sum_{i=1}^m (y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})) \quad (4)$$

En ella, el superíndice i representa el número de ejemplo y la letra m corresponde a la cantidad de ejemplos que se emplearán para el aprendizaje del modelo.

La ecuación de costo se encarga de medir la eficiencia de los parámetros w y b en la predicción del modelo, sin embargo, si la función de costo es alta, se deben modificar los valores de las constantes anteriores para mejorar la predicción y que el costo sea el menor posible, como se observa en la figura 5.12, en ella, se comienza con un punto rojo aleatorio y el principio es

modificar w y b de tal forma que se llegue al punto mínimo global indicado con una flecha de color rojo en la parte baja del plano.

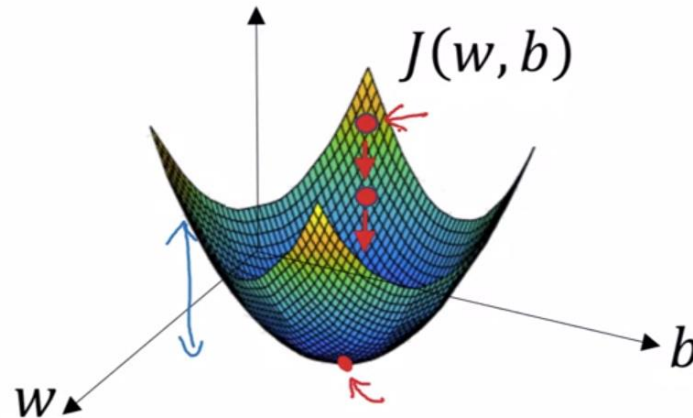


Figura 5.12 Representación de la función de costo. Fuente: Ng et al., 2019.

5.5.5. Función de descenso de gradiente (Gradient Descent)

Para realizar la modificación de las variables, como se mencionó anteriormente, se plantea el uso de una función donde se le resta al valor original ya sea de w o de b , una constante que denominará razón de aprendizaje, el valor de esta se introducirá en la ejecución del programa y no variará en las iteraciones, multiplicada por la derivada parcial de la función de costo respecto a la variable que se desea modificar, las ecuaciones se muestran a continuación:

$$w := w - \alpha \frac{\partial J(w, b)}{\partial w} \quad (5)$$

$$b := b - \alpha \frac{\partial J(w, b)}{\partial b} \quad (6)$$

La idea de emplear la derivada es encontrar la pendiente para que comience el descenso del valor de J hasta encontrar el mínimo global, dada la condición de la derivada, el sistema puede comenzar ya sea en la parte derecha o izquierda, ya que siempre irá descendiendo porque el valor de la

pendiente ya sea positiva o negativa se ajustará con el signo introducido en la ecuación lo que hará que multiplicando el factor de aprendizaje logre reducirse.

Esta ecuación del descenso de gradiente se aplicará en un paso que se denominará propagación hacia atrás, donde basados en la salida final y su error se derivará respecto a la salida de la función de activación, a su vez, se derivará la salida de la función lineal de la neurona. También, se debe buscar el término de la derivada parcial descrita en las ecuaciones anteriores. Debido a la complejidad que se puede tener con las variables, sus nombres y simbología se hará un pequeño repaso en orden hasta llegar a la propagación hacia atrás.

En primer lugar, la salida de la neurona de la primera capa de entradas se puede definir como la multiplicación de cada una de las entradas con su respectivo peso y, al final, se suma la constante b , el superíndice indicado mediante paréntesis cuadrados corresponde al número de capa.

$$z^{[1]} = w_1^{[1]}x_1 + w_2^{[1]}x_2 + \dots + w_n^{[1]}x_n + b^{[1]} \quad (7)$$

Ahora bien, al resultado de la ecuación anterior se le debe aplicar la función de activación (g), la cual puede ser cualquiera de las antes mencionadas, a esta salida se le asignara la letra a , como se pueden aplicar distintas funciones de activación a distintas capas se le incluye el paréntesis cuadrado para identificarse con mayor facilidad.

$$a^{[1]} = g^{[1]}(z^{[1]}) \quad (8)$$

Posterior a este paso, se aplica la misma ecuación de z antes mencionada, pero, con la diferencia de que ahora sus entradas no corresponden a las del sistema, sino que, por el contrario, serán las salidas de la capa anterior, este proceso se repetirá a lo largo de toda la red neuronal hasta llegar a

la capa de salida, la ecuación genérica quedaría de la siguiente forma, con la salvedad de que ahora en los superíndices se colocó la letra n para que sea sustituida por el número de capa que se desee.

$$z^{[n]} = w_1^{[n]} a_1^{[n-1]} + w_2^{[n]} a_2^{[n-1]} + \dots + w_n^{[n]} a_n^{[n-1]} + b^{[n]} \quad (9)$$

La función de error (L) tomará el valor de (a) en la última de las capas y lo comparará con la salida esperada y, cuando se comienza en la propagación hacia atrás se debe derivar respecto a (a), en este caso, como se muestra en la siguiente ecuación.

$$\frac{dL(a, y)}{da} = -\frac{y}{a} + \frac{1-y}{1-a} \quad (10)$$

Si se continúa hacia atrás, ahora se debería derivar la función z, pero, debido a que la función debe pasar por su respectiva activación, se debe derivar dicha ecuación respecto a z y basado en las reglas de derivadas se multiplicará $\frac{dL}{da}$ con la derivada de la función de activación que se denominará $g'(z)$. Las derivadas de las funciones de activación antes mencionadas se describen a continuación.

Sigmoide

$$g'(z) = \frac{1}{1 + e^{-z}} \left(1 - \frac{1}{1 + e^{-z}} \right) \quad (11)$$

ReLU

$$g'(z) = \begin{cases} 0 & \text{si } z < 0 \\ z & \text{si } z > 0 \end{cases} \quad (12)$$

Tangente hiperbólica

$$g'(z) = 1 - \tanh^2(z) \quad (13)$$

Finalmente, la expresión de dz en la última capa de neuronas sería la siguiente:

$$dz = \frac{dL(a, y)}{da} \cdot g'(z) \quad (14)$$

Al continuar la iteración hacia atrás el valor del dz que se acaba de calcular servirá como entrada para la derivada de z en la capa anterior a esta, pero se le debe multiplicar la constante w quedando de la siguiente forma genérica, utilizando las mismas reglas de los subíndices y superíndices antes mencionados.

$$dz^{[l]} = w^{[l+1]} \cdot dz^{[l+1]} * g^{[l]'}(z) \quad (15)$$

Cabe destacar que el asterisco que se indica en la ecuación anterior pasaría a ser un producto de elementos si se trabajase con matrices, y no una multiplicación normal de matriz como si lo fuera el producto entre w y dz.

En todas las neuronas para determinar la derivada de las constantes, lo cual es el fin de la propagación hacia atrás para modificar el valor de estas y así ajustarse al modelo serían de la siguiente forma.

$$db = dz \quad (16)$$

$$dw = dz \cdot a \quad (17)$$

Si se tienen múltiples ejemplos para el entrenamiento es importante hacer un promedio de los valores encontrados, dividiendo entre el número de ejemplos cuya variable será la letra m. Una vez obtenidos los valores de las ecuaciones 16 y 17, se deben actualizar los valores de w y b con las ecuaciones 5 y 6, respectivamente.

El proceso anterior debe realizarse en reiteradas ocasiones, con el fin de ajustar lo mejor posible los parámetros w y b, cuando se da este proceso iterativo es importante verificar que la razón de aprendizaje sea un valor adecuado, ya que este le permite al modelo encontrar el mínimo global con la cantidad mínima de iteraciones, si es un valor muy grande, puede pasarse del valor y divergir

o necesitar un número significativamente mayor de recursividades, tal y como se muestra en la figura 5.13 donde en la sección a) mediante los puntos rojos se muestra que cada iteración se acercaba más al punto con el coeficiente de aprendizaje de 0.005, pero en la parte b) en ocasiones se pasaba y ocurrían saltos muy grandes, lo que complica más encontrar el punto óptimo, con un coeficiente de aprendizaje de 0.05, ambos con 50 iteraciones totales

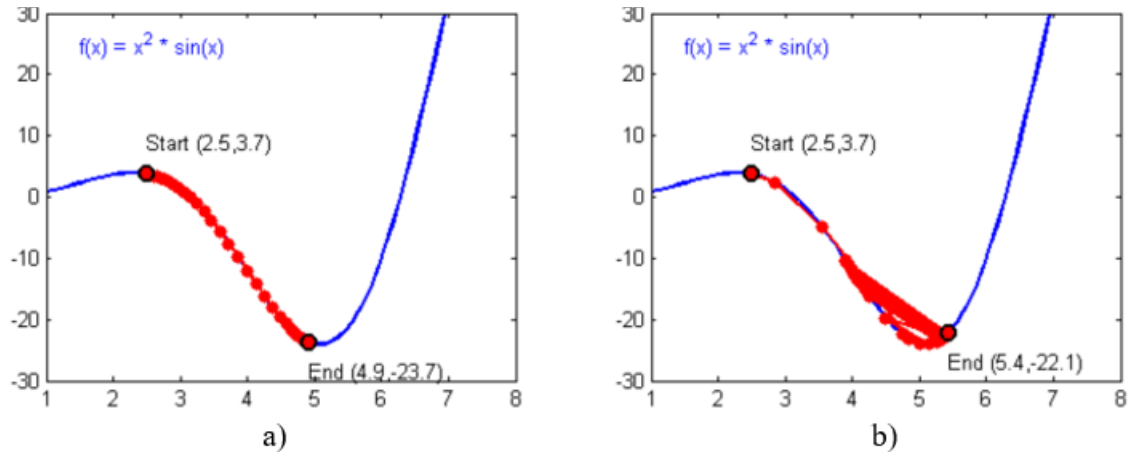


Figura 5.13 Razón de aprendizaje. Fuente: Ng et al., 2019.

En el desarrollo de las redes neuronales, se le puede definir como hiperparámetros a las características que son modificables para llevar a cabo el modelo, estas son: la razón de aprendizaje, la cantidad de iteraciones, la selección de la función de activación, el número de capas y el número de unidades ocultas. La combinación de estas funciones logra que la red neuronal sea capaz de ajustarse al modelo y permite descubrir o analizar con mayor o menor profundidad las características del conjunto de datos y que el aprendizaje sea el mejor posible, la forma de elegir las será con el criterio experto y una serie de pruebas hasta encontrar los mejores parámetros.

5.6. Sistemas expertos

Como se mencionó anteriormente y puntualizado por Badaró et al. (2014), los sistemas expertos (SE) se pueden considerar como un subconjunto de la inteligencia artificial, pues sus acciones emulan el pensar de una persona mediante la atribución de conocimiento y formular un aprendizaje

para proporcionar una respuesta acorde. La idea principal de estos sistemas es que cualquier persona pueda hacer uso de estos programas sin la necesidad de ser un experto en el tema, ya que esto lo proveería el software.

La complejidad de un sistema experto se puede extender tanto como se requiera llegando, incluso, a la capacidad de tomar decisiones involucrando la experiencia de múltiples expertos y sets de conocimientos que se incorporan al método de aprendizaje del programa, logrando funcionar mejor que un simple experto en un determinado tema.

El primero de los sistemas expertos, fue desarrollado entre los años sesenta y setenta por Edward Feigenbaum y Joshua Lederberg en la Universidad de Stanford en la búsqueda de nuevas aplicaciones de la IA y, desde ese momento, han comenzado a convertirse en un sistema cada vez más complejo, logrando predecir y funcionar como una persona experta en los distintos ámbitos donde se aplica, el desempeño de estos SE; se basa en el conocimiento almacenado en su base de datos, cuanto mayor sea el tamaño de este mejor será el comportamiento del sistema.

Algunos de los ejemplos más característicos son, la sugerencia de Google, cuando se escribe mal algún termino en su cuadro de búsqueda, mostrando los resultados aproximados de lo que se deseaba buscar, el sistema DENDRAL fue realizado por expertos en la química, este es usado para detectar moléculas orgánicas desconocidas con ayuda del espectro de masa, MYCIN se encarga de detectar la bacteria que causa meningitis y bacteriemia, llegando a recomendar antibióticos.

Para que un sistema experto funcione apropiadamente debe contener una serie de características, las cuales se logran a través de un desarrollo adecuado y con los conocimientos claros y específicos para resolver el problema, y dado que este software será utilizado por personas que no son expertos, debe ser capaz de generar respuestas que puedan entenderse fácilmente y sin ambigüedades de

modo que se puedan proporcionar datos en lenguaje cotidiano y pueda devolver una respuesta en ese mismo lenguaje, la confiabilidad debe ser alta para evitar errores de aplicación y posibles problemas de un mal diagnóstico y por último, el SE tendrá la capacidad de generar una respuesta en el menor tiempo posible cuando se le da un problema indistintamente de la complejidad.

Con el fin de lograr las capacidades antes mencionadas, el sistema experto se basa, además de la base de conocimiento, en un motor de inferencias, el cual se encarga de evaluar e interpretar los hechos que se introducen al SE para proveer una respuesta adecuada, a menudo, este motor involucra tareas de clasificación, monitoreo, diseño, programación y planeamiento. Zwass (2016) menciona en su estudio que la base de datos de este sistema se completa mediante observaciones y entrevistas usualmente representadas de la forma “Si, entonces”, como se muestra en la figura 5.14, a las cuales se les conoce como reglas de producción, estas suelen ser de un tamaño considerable (en ocasiones más de mil), también deben añadirse las famosas reglas de dedo antes mencionadas y la heurística que son usadas por los expertos para determinar un problema.

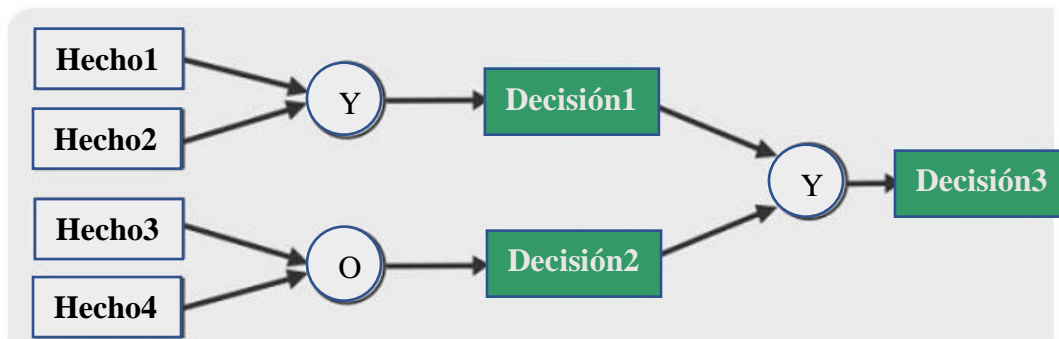


Figura 5.14 Base de condiciones de un sistema experto. Fuente: Tutorials Point India Limited, 2020.

Para desarrollar un sistema experto, se requieren una serie de pasos con el fin de desarrollarlo de la forma más ordenada posible y sin contratiempos que puedan entorpecer el procedimiento o el desempeño del programa estos son citados por (Tutorials Point India Limited,2020).

- **Identificar el problema:** el problema debe ser adecuado para que un modelo de sistema experto lo pueda resolver, es decir, debe tener la capacidad de poder manejarse mediante reglas de si-entonces. También, deben encontrarse los expertos en el tema para poder obtener la mayor cantidad de datos y experiencias posibles para entrenar al sistema. Y, por último, debe establecerse con claridad el costo beneficio del proyecto con el fin de atraer al cliente y determinar correctamente la viabilidad del sistema.
- **Diseño del sistema:** debido a que existen muchas formas de programarse, es importante identificar la tecnología de sistema experto que mejor se adecúe al problema, en conjunto con la integración y el porcentaje de este con respecto a otros modelos preexistentes o con bases de datos y finalmente, determinar cómo los conceptos representan mejor el dominio de conocimientos atribuidos en el paso anterior.
- **Desarrollo del prototipo:** una vez obtenidos los datos y con el modelo claro, se debe tomar el conocimiento y transformarlo digitalmente a la vez que se convierten en reglas de si-entonces para entrenar al modelo.
- **Pruebas y refinamiento del prototipo:** cuando el modelo es desarrollado se deben proveer una serie de ejemplos con una salida conocida por el usuario con el fin de determinar la precisión y el desempeño, si la salida es satisfactoria se puede seguir con su puesta en marcha, pero si por el contrario la salida no es correcta o precisa se deben modificar los parámetros y las reglas para mejorar su predicción y recomendación.
- **Puesta en marcha:** una vez que la salida es satisfactoria y se tiene certeza que el modelo funciona adecuadamente y toma los datos de una forma eficaz logrando establecer comunicaciones con los diversos módulos y bases de datos se puede hacer un despliegue

del programa, documentándolo para posteriores intervenciones y finalmente entrenar al usuario para su uso, no se debe olvidar que una aplicación o software se vuelve obsoleto si no se le da su respectivo mantenimiento.

Basados en el último punto, es importante destacar que el software no lo utilizará una persona con amplios conocimientos o experta en el tema, de modo que es necesario utilizar una interfaz lo más intuitiva posible, con el fin de evitar ambigüedades, generalmente se emplea el lenguaje natural basado en caracteres como el presente documento, narraciones verbales y una lista de números desplegados en pantalla, con botones y demás componentes que hagan de la interfaz de usuario (UI), lo más amigable, que se ajuste a las necesidades y capacidades del usuario de forma que sea lo más eficiente posible.

Finalmente, estos sistemas pueden incluir *Machine Learning* o Redes Neuronales, con el fin de mejorar las capacidades del modelo formando un criterio y asemejando el pensamiento humano como lo ha hecho IBM con su sistema ROSS, el cual emplea el “*Deep Learning*” y procesamiento de lenguaje en conjunto con un aprendizaje no supervisado y autónomo y con un reconocimiento de patrones que lo hacen un sistema muy similar a como lo haría el funcionamiento del cerebro humano (Rouse, 2016).

5.7. Python

En el desarrollo de las aplicaciones tanto de Redes Neuronales como *Machine Learning* uno de los lenguajes que se emplean con mayor frecuencia es el entorno de Python. Una de las principales ventajas de este sistema es que es de código abierto (*Open Source*). En la página oficial, se menciona que Python es un intérprete, orientado a objetos, cuyo lenguaje de programación es de categoría alto nivel, la sintaxis es amigable con el programador y debido a que no existe ningún

paso de compilación, el ciclo de edición, prueba y depuración es increíblemente rápido, un error o una entrada incorrecta nunca provocará una falla de segmentación, en cambio cuando se descubre el error genera una excepción.

Es común encontrar en múltiples foros secciones de código que permiten al programador buscar ayuda para resolver un determinado problema algunos se encuentra dentro de la página oficial del desarrollador, incluso, al ser de código abierto, logra que múltiples personas fabriquen secciones de código que pueden ser llamadas por otros para facilitar y evitar la escritura de largas líneas de código en forma reiterada. La flexibilidad es otro punto fuerte de este sistema, ya que en este se puede crear tanto aplicaciones de escritorio como aplicaciones web o interfaces de usuario todo bajo el mismo programa y lenguaje de programación.

Algunas de las ventajas que hacen a Python un lenguaje tan popular es la cantidad de librerías, tipos de datos para trabajar y funciones incorporadas dentro del mismo lenguaje que ayudan como se mencionó con anterioridad a realizar múltiples tareas evitando su programación desde un espacio en blanco. Otro punto a favor es la sencillez y velocidad con la que son creados los programas en comparación con otros lenguajes como C o Java, los cuales necesitan una cantidad considerablemente mayor en algunas funciones. Finalmente, las plataformas o entornos que permiten su uso son Windows, Mac, Linux, entre otros más.

5.8.Librerías y módulos en Python

Como se mencionó anteriormente, las librerías y módulos en el lenguaje de programación permiten organizar lógicamente el código, cuando unas líneas con comandos son agrupadas dentro de un módulo hace que cuando se desee programar pueda ser más fácil su uso. Como tal, un módulo es un objeto de Python, en este caso, que tiene un conjunto de funciones con nombres variados y

arbitrarios que se pueden emplear en cualquier código haciendo uso de una importación o con referencias.

Python posee una librería estándar con un gran número de módulos los cuales son los más comunes dentro del entorno de programación, que van desde comandos para archivos del sistema operativo (OS) hasta interfaces gráficas como Tkinter. Esto facilita el no tener que buscar por la red el módulo deseado para su instalación mediante comandos pip u otros, sino que, por el contrario, cuando se instala Python desde su página oficial o si se usara algún otro programa que corra este lenguaje ya traería preestablecido y descargado estos códigos.

Para hacer un llamado o incorporación al código que se esté programando de uno de los módulos se debe realizar por medio de la función “*import*” seguido del nombre del módulo que se desea emplear como por ejemplo “*math*”, “*request*”, “*time*”, por citar algunos, en la figura 5.14 se puede observar este procedimiento, dependiendo del editor de texto que se esté utilizando los comandos como “*import*” pueden cambiar de color para diferenciarse de otras instancias o texto (en este caso es de color morado).

```
8 import math
9 import request
10 import time
```

Figura 5.15 Importado de módulos en Python. Fuente: Elaboración propia.

Otra forma de importar los módulos sin la necesidad de tomar tantas líneas de código es colocar una coma luego del nombre de cada uno de ellos, también, si no se coloca la dirección el sistema buscará en la carpeta por defecto dentro del entorno Python, si se tiene algún programa o módulo que se encuentre en otro lugar se deberá especificar la ubicación por medio de el comando “PYTHONPATH = C:/.....”, donde los puntos indican la carpeta, subcarpeta, etc.

En algunas ocasiones, resulta importante unir varias programaciones por lo que surge el comando “from x import y” en este se le especifica que desde el archivo “x” se tomen las funciones “y” o por el contrario si se desea tomar todas las clases y funciones se debe sustituir la letra “y” por un asterisco “*”. Algunas de los módulos más utilizados son los que se describen a continuación.

5.8.1. Numpy

El primer módulo que se mencionará posee el nombre de NumPy, sin embargo, para hacer un llamado de este dentro de la programación se escribirá en minúsculas, este paquete es fundamental cuando se desea trabajar con cálculos científicos o matemática avanzada, esta librería permite la manipulación de objetos multidimensionales como matrices, vectores, transformadas de Fourier, algebra básica, simulaciones aleatorias, operaciones estadísticas entre otros.

Numpy es un módulo que incrementa la velocidad de cálculo de operaciones matemáticas complejas mediante el método de vectorización, este describe la ausencia de cualquier ciclo explícito, ya que, por ejemplo, si se tiene una matriz de un tamaño considerable, el método anterior consistía en hacer dos ciclos mediante un “for”, con este, se buscaban las posiciones y se hacía el cálculo correspondiente, luego, continuaba a la siguiente posición y se repetía el procedimiento, este método es una pérdida de tiempo significativa que se manifiesta en mayores proporciones cuando el tamaño de los datos incrementa.

Dentro de las ventajas de utilizar un código vectorizado, además de la velocidad, es la facilidad de entendimiento y lectura, ya que, si se emplearan ciclos, es difícil saber con seguridad el procedimiento que se está realizando mientras que cuando se vectoriza la misma línea “dice” qué es lo que se está ejecutando dado que se asemeja a la notación matemática estándar, por lo general,

se necesitan mucho menos líneas de código y, por consiguiente, existe menos probabilidad de errores de escritura.

Un término que es importante conocer es el de “*Broadcasting*” este se emplea para describir el comportamiento implícito elemento por elemento, lo que permite aplicar operaciones a elementos con distintos tamaños, por ejemplo, si se desea realizar una multiplicación, suma u otro operación entre dos matrices llámese “a” y “b”, estas pueden ser del mismo tamaño, o un escalar y un vector o incluso dos vectores de distinto tamaños, lo que realiza este procedimiento es expandir el tamaño en una dimensión para hacerlo coincidir con el más grande, esto facilita mucho la programación cuando involucra operaciones entre componentes con distintos tamaños, sin embargo se debe tener cuidado, ya que puede aplicarse en ocasiones donde no se necesite (NumPy, 2020).

Para hacer uso de la librería, luego de importarse el módulo, se debe escribir el comando seguido de un punto y después la función a emplear. En la figura 5.16, se muestra un ejemplo de un comando denominado “zeros”, el cual fabrica una matriz del tamaño 2 filas y 3 columnas y llena cada espacio con ceros, también se puede especificar el tipo de datos y demás, pero por ahora solo se mencionará como se escribe el código. La última línea contiene un comando llamado “*print*”, el cual se utiliza para mostrar la operación en pantalla que se muestra en la parte de la derecha de la misma figura.

```
8 import numpy
9
10 a=numpy.zeros((2,3))
11 print(a)
```



Figura 5.16 Ejemplo de uso de la librería NumPy. Fuente: Elaboración propia.

5.8.2. Tkinter

Otro de los módulos más populares y, por tanto, más utilizados es el famoso Tkinter, este corresponde al paquete por defecto de Python para fabricar Interfaces Gráficas de Usuario (GUI),

esta es una importante adición al código, ya que permite al usuario no hacer modificaciones o interactuar con líneas de escritura del algoritmo, ya que una persona sin el conocimiento o experiencia en el área podría provocar problemas de ejecución, esto posibilita a que cualquiera pueda hacer uso del programa.

Para crear la interfaz, basta con importar el módulo Tkinter con los pasos que antes se han mencionado, luego, se tiene que crear la aplicación mediante la ventana principal, posterior a esto, se comienzan a agregar los objetos con los cuales interactuará el usuario de la aplicación y, finalmente, se coloca la instrucción de “*main event loop*”, la cual estará en constante ejecución y tomará acciones que sean introducidas en la GUI por el usuario. Algunos de los objetos que tiene la posibilidad de integrar esta librería son los siguientes.

- a) **Botones simples:** esta función permite incorporar botones para que sean presionados y desencadenar algún tipo de acción dentro de la aplicación, de modo que funcionan como detonantes de actividades, pueden configurarse tantos como el tamaño de la pantalla lo permita.
- b) **Canvas:** permite al programador establecer un espacio para formar o dibujar líneas, óvalos, polígonos y rectángulos donde pueden contener una imagen u otro objeto que se desee.
- c) **Botones tipo check:** cuando se pueden seleccionar múltiples objetos generalmente se colocan un cuadro donde en su interior si se pulsa con un clic se rellenará con un símbolo de *check* (✓)
- d) **Entrada de texto:** necesaria para todo tipo de aplicaciones donde sea necesario que el usuario o el programa introduzca texto en un espacio en concreto.

- e) **Cuadros:** este comando es especialmente útil cuando se busca organizar o agrupar otros objetos, funciona como un contenedor donde dentro de él alberga a otros rubros en un área rectangular.
- f) **Etiquetas de texto:** permite colocar texto que no es modificable por el usuario, sin embargo, el programa puede editarlo sin problema las veces que sea necesario
- g) **Listado de objetos:** empleado para desplegar una lista de ítems donde el usuario puede seleccionarlos.
- h) **Botón de menú y menú:** estos botones cuando son presionados muestran una serie de opciones para que el usuario pueda elegir y realizar una determinada acción o seleccionar un objeto de la lista que se despliega, se utiliza en conjunto con la función menú.
- i) **Mensajes:** este comando es similar a la etiqueta sin embargo este tiene la posibilidad de ser multilínea y puede amplificar su tamaño para que se ajuste al texto que tiene en su interior justificándolo también.
- j) **Botones tipo circulares de opción:** son utilizados con mayor frecuencia cuando de un grupo de opciones solo es posible elegir una de ellas, tienen la forma de un círculo y cuando es presionado en su centro se forma otro círculo de color negro, indicando la selección.
- k) **Escalas:** este comando funciona como un deslizador, cuando se mueve en alguna dirección cambia el valor, este se puede tomar para realizar alguna otra acción que se desee.
- l) **Barra de desplazamiento:** cuando la ventana o algún componente es muy grande para mostrarse se coloca una barra en alguno de los laterales con el fin de mostrar la parte

restante que estaba oculta, puede colocarse en la ventana y los objetos que permitan su uso.

m) Cuadro de selección: este es un espacio donde se muestran valores en un espacio y al final de este se encuentran dos flechas una hacia arriba y otra hacia abajo las cuales cuando son presionadas muestran el valor anterior o posterior según corresponda.

5.8.3. Pandas

El módulo Pandas corresponde a una librería la cual permite al programador manipular datos de forma eficiente mediante el indexado integrado, logra también leer y escribir datos en distintos formatos que incluyen el .csv (valores separados por comas), el cual puede leer sin problemas programas sencillos como Microsoft Excel o archivos .xlsx propietarios de este último programa, bases de datos SQL, y uno de los formatos con mayor rapidez para su lectura y escritura como el HDF5.

Cuando se trabajan con datos, suele ser un problema su lectura o escritura debido a que en ocasiones cuando se ejecutan estas tareas los datos no se alinean correctamente o pueden cortarse haciendo particiones no deseadas, este módulo permite tener un control completo y evitar estos problemas, logrando, incluso, unir varios sets de datos en diferentes locaciones en una sola. Posee la versatilidad de aplicarse en series de tiempo o intervalos que se establezcan. Lo que lo hace una herramienta de gran importancia en el análisis de datos en tiempo real, por lo que es ampliamente usado en finanzas, neurociencia, economías, estadísticas y muchos otros servicios más. Y lo más importante es que es gratuita por ser de tipo *Open Source*, tanto para fines comerciales como didácticos (Pandas, 2020).

5.9.Arduino

Arduino como tal es una plataforma de tipo *Open Source*, basada en un hardware de fácil uso y con un software intuitivo, cuenta con tarjetas impresas, las cuales emplean un microcontrolador que varía dependiendo del tamaño capacidades que se requieran, donde el más común de las placas es el Arduino UNO, como se muestra en la figura 5.17, en ella, se detalla que en la parte superior numeradas del 0 al 13 se encuentran las salidas/entradas de este dispositivo, las cuales pueden ser de tipo PWM (~) o digitales. En la parte inferior, mediante la etiqueta A0 hasta A5, se encuentran las entradas de tipo analógico puras, por lo que tiene 6 pines tipo PWM, 6 analógicos y 20 digitales.

Esta placa se alimenta mediante la entrada de USB tipo A (cuadro color gris en la parte superior izquierda de la figura 5.17), lugar donde también sirve de puerto de programación, o bien, si se requiere más corriente, se puede emplear el puerto de color negro ubicado en la parte inferior derecha de la figura 5.17, el cual es de tipo barril de 5.5mm/2.1mm, donde se pueden conectar de 9V a 12V de corriente directa.



Figura 5.17. Arduino UNO. Fuente: Arduino, 2018.

La parte del Arduino que se encarga de interpretar las entradas y salidas, realizar acciones y establecer comunicaciones se denomina microcontrolador, este es un circuito integrado, el cual funciona como procesador de una computadora con la salvedad que el utilizado por el modelo de Arduino posee otros componentes en su interior, como registros, unidades control, unidades lógico-aritméticas y demás.

Se podría decir que un microcontrolador es una computadora de propósito especial estos están dedicados a una tarea en específico a diferencia de los ordenadores, el cual está almacenado en una ROM (memoria de solo lectura) y, generalmente, no suele cambiar con mucha frecuencia, usualmente estos circuitos integrados no requieren de grandes cantidades de potencia para funcionar, en el caso del Arduino UNO emplea el ATMEGA 328P, un chip de 28 pines con memoria para el programa de hasta 32kB, y con la capacidad de monitorear o escribir hasta 23 entradas de forma simultánea.

La placa de Arduino UNO también tiene salidas de 3.3V y 5V, con el fin de alimentar diversos sensores o actuadores de bajo consumo (no más de 50mA), si se requiere conectar dispositivos con mayor consumo, se debe proveer de una fuente externa, pero los terminales negativos (GND) deben unirse, con el fin de mantener una referencia común.

Bajo el mismo concepto de muchos de los componentes y software mencionados, al ser de tipo *Open Source*, existen muchos desarrolladores independientes que logran realizar placas o componentes para este sistema a muy bajo costo, donde se pueden mencionar sensores de todo tipo, actuadores, módulos de comunicación ya sea por bluetooth, internet alámbrico, inalámbrico, y demás, para programarse el microcontrolador emplea un software y lenguaje propietario de alto nivel y bastante intuitivos totalmente gratuitos (Arduino IDE), bajo el cual también existen módulos y librerías creadas por terceros que se pueden instalar sin costo alguno.

5.10. Módulo Wifi ESP8266

Uno de los módulos con mayor utilidad y más empleados es el ESP8266 fabricada por la empresa china Espressif, este es un circuito impreso capaz de comunicar al Arduino UNO a una red de internet inalámbrica, este se conecta mediante los pines que se muestran en la figura 5.18. Posee una tensión de alimentación de 3.3V, la cual puede ser alimentada por las salidas específicas de la placa Arduino.

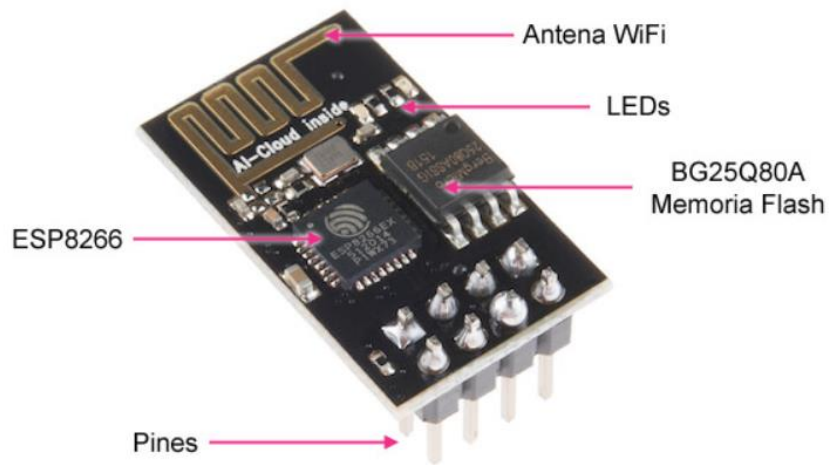


Figura 5.18. Módulo Wifi ESP8266-01. Fuente: del Valle Hernández, 2017.

Cuando salió al mercado hace algunos años, este módulo era bastante complicado de programar ya que este también posee un microcontrolador y requiere de ciertos comandos, el firmware es un software de bajo nivel que permite la comunicación y el control de las partes eléctricas de un circuito integrado. Como menciona Del Valle Hernández (2017), este modelo posee una versión con la que se puede comunicar con este módulo mediante comandos AT, estos son instrucciones codificadas que forman un lenguaje de programación entre el hombre y un terminal modem para lograr codificarlos y ejecutar acciones, fue desarrollado por Dennis Hayes en 1977, (la lista de comandos se muestra en la sección de apéndices [Chen,2015]).

Para que un módulo sea capaz de compartir información con otro necesita un cierto tipo de protocolo para que ambos logren “entenderse”, en el caso del Arduino posee protocolos como UART, I2C y SPI, en concreto el 8266 emplea el tipo UART (Receptor y Transmisor Asíncrono Universal), el cual utiliza una comunicación tipo asíncrona en serie, esto quiere decir que el número de cables se reduce a 2 los famosos pines con etiquetas Tx como transmisor y Rx como receptor, aquí no se hace necesario el uso de algún tipo de reloj externo para enviar pulsos y hacer fluir la información, sino que por el contrario hacen uso de sus relojes internos.

Sin embargo, dado que los relojes antes mencionados pueden tener distintas velocidades de oscilación ocurrirán problemas, si estas no son iguales, para ello existen ciertos valores estándar y alguno de módulos deberá ser modificado para ajustarse al otro, este término de velocidad en la transmisión de datos se le conoce como Tasa de Baudios y hace referencia a la cantidad de bits que son enviados o recibidos por segundo, Arduino tiene la posibilidad de tener valores de 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200.

Este protocolo tiene como inconveniente que solo dos dispositivos pueden conectarse entre sí, no se admiten más, ya que, para la transmisión de datos, como se mencionó anteriormente, se emplean dos terminales Rx y Tx, de modo que cuando uno envía los datos el otro los debe de recibir, por ello, se conecta el Tx de un módulo con el Rx de otro y viceversa, como se muestra en la figura 5.18.

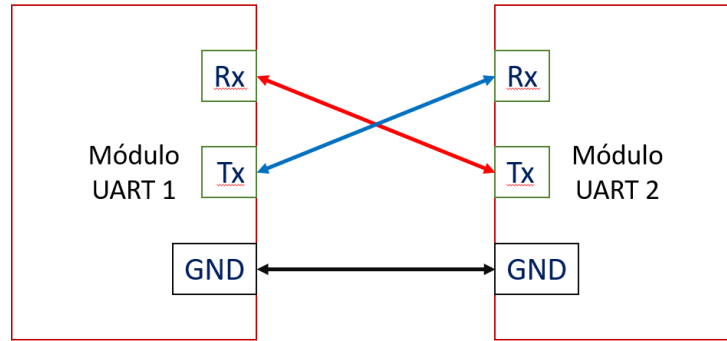


Figura 5.19 Conexión del protocolo UART. Fuente: Elaboración propia.

5.11. Bombas centrífugas

Una bomba es una máquina, la cual transforma la energía mecánica en la entrada de este sistema hacia el fluido en forma de presión o movimiento, como expresa Garro (2014), en muchos de los campos de aplicación de una bomba incluye tanto posición, velocidad y presión en la conversión hacia el fluido, las cuales se rigen por medio de las leyes fundamentales de la mecánica de fluidos.

A menudo, la energía mecánica en la entrada es provista por un motor eléctrico capaz de tomar la energía eléctrica y convertirla en movimiento, en muchos casos la unión entre el motor y el dispositivo de bombeo ese mediante un acople directo como se muestra en la figura 5.20. Con el avance en el desarrollo de los motores eléctricos, estas máquinas se han convertido en una opción fiable debido a su reducido peso y a una mejora en sus prestaciones.



Figura 5.20. Bomba centrífuga con acople directo. Fuente: Jacuzzi, 2020.

Las bombas centrífugas reciben su nombre debido a su principio de funcionamiento, donde el fluido es movido gracias a la fuerza centrífuga que provocan sus componentes, en la figura 5.21 se detalla este principio, donde el fluido entra por la parte central, luego, mediante el movimiento proveniente de su fuente mecánica, hace girar el impulsor, el cual posee una serie de aspas o álabes que, cuando este gira hace que el fluido sea impulsado fuera de este hacia la carcasa o voluta hasta que es expulsado fuera de la bomba por el difusor. En la figura 5.21, se muestra mediante flechas de color azul el movimiento del fluido.

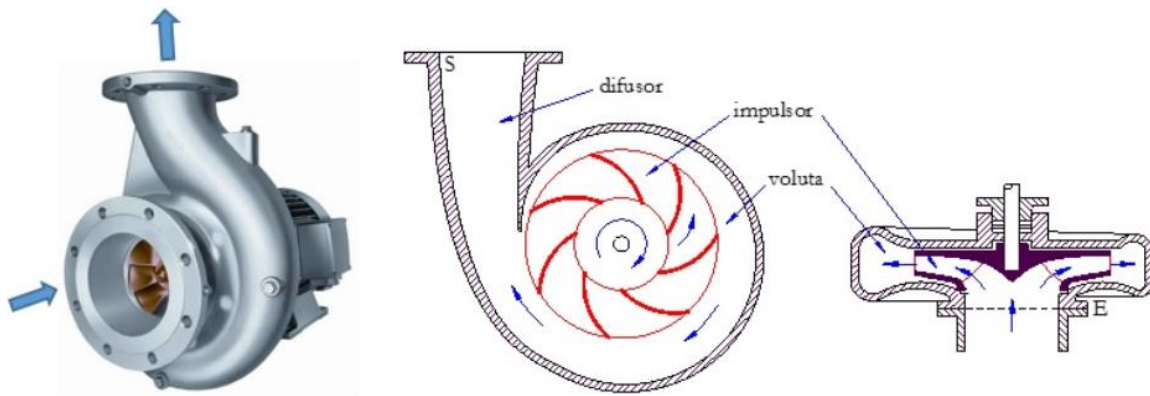


Figura 5.21. Principio de funcionamiento de una bomba centrífuga. Fuente: Monge, 2017.

Estas bombas pueden ser de diversos tipos, algunas de flujo radial, axial o mixto dependiendo de la dirección del fluido, la que se muestra en la figura 5.21 es de tipo radial, pues la diferencia entre la entrada y la salida es de 90° mientras que las de tipo axial se encuentran a 180° , es decir se puede observar que el fluido se dirige en la misma dirección. Ahora bien, la capacidad de presión la define la voluta y, en ocasiones, no es suficiente para alcanzar la demanda por lo que coloca una rodete o impulsor detrás de otro lo que se le conoce como una bomba de varias etapas. También puede tener diversos tipos de succión como simple, doble, positiva (nivel de líquido superior a la bomba), negativo (nivel de líquido inferior a la bomba) o puede provenir de un tanque a presión.

Ahora bien, la bomba centrífuga posee más componentes que son vitales para su funcionamiento, los cuales se muestran en la figura 5.22, cada una de ellas posee una función en específico las cuales se detallan a continuación, algunas de ellas pueden variar respecto a otras bombas, debido a los diversos tipos, esta que se presenta es centrífuga radial con succión simple.

- 1. Tubería de succión.**
- 2. Impulsor o rodete.**
- 3. Rotor (transmite el movimiento del motor al impulsor).**

4. Voluta o carcasa.
5. Rodamientos.
6. Caja de rodamientos.
7. Sellos mecánicos para evitar las fugas del líquido.
8. Tubería de descarga.

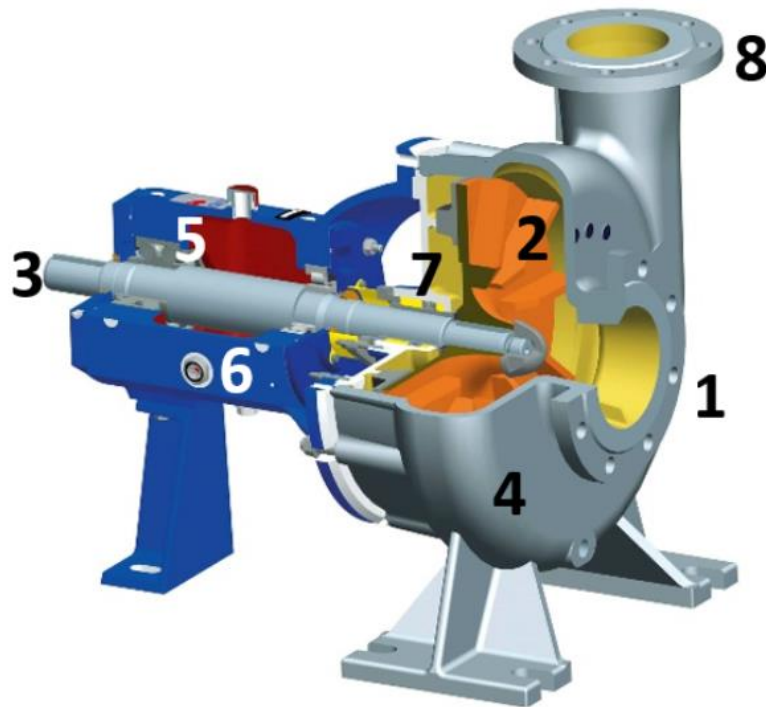


Figura 5.22. Partes de una bomba centrífuga. Fuente: Trocel, 2019.

5.12. Motor eléctrico

Como parte de entender los componentes de una bomba centrífuga, la principal fuente de energía proviene de un motor eléctrico, el cual es una máquina capaz de convertir energía eléctrica en mecánica por medio de las fuerzas magnéticas, estos motores pueden ser de diversos tipos dependiendo de su alimentación, corriente continua (CC) o corriente alterna (CA) o algunos que logran funcionar con cualquiera como los universales.

En el caso de los motores de corriente alterna, estos poseen diversas formas de construirse para provocar el movimiento, y pueden ser síncronos o asíncronos, a estos últimos también se les conoce como motores de inducción y poseen la capacidad de ser alimentados ya sea por corriente trifásica o monofásica y su rotor puede ser devanado o de tipo jaula de ardilla, en donde sea posible es preferible el uso de tres fases en lugar de una debido a su simple y robusta construcción y sus factores de confiabilidad, las partes principales de un motor de inducción de jaula de ardilla se muestran en la figura 5.23.

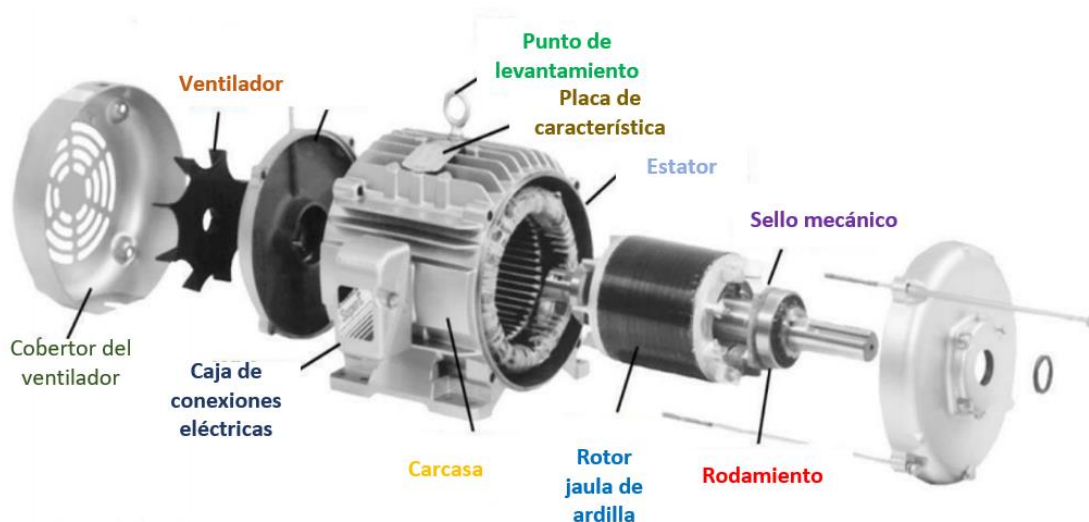


Figura 5.23. Motor de inducción de jaula de ardilla. Fuente: Alkadhim, 2020 (Traducido).

Para el motor de inducción el principio de funcionamiento, recae en un flujo de corriente hacia el estator para establecer un flujo magnético el cual pasa a través del espacio de aire que existe entre el estator y el rotor, el cual, a su vez, induce una corriente eléctrica en el rotor y formando un flujo magnético en el rotor, la interacción entre ambos flujos produce que una parte busque alinearse magnéticamente con la otra, por tanto, el rotor comenzará a girar tratando de acoplarse al estator.

El rotor puede poseer una construcción de tipo jaula de ardilla, llamada así por la forma que tiene, en los extremos se encuentran dos placas que son unidas mediante barras de aluminio o cobre, esta configuración es muy empleada, debido a la robustez, ya que, al no poseer contactos no existe desgaste progresivo tan acelerado como su contraparte, su construcción es simple y fácil de realizar y posee mucha eficiencia, debido a su baja resistencia y bajas pérdidas en el cobre.

Todo motor eléctrico debe llevar consigo en su carcasa una placa de características, con el fin de conocer las capacidades de la máquina, así como los estándares bajo los cuales fue fabricado y muy importante si es compatible con la red eléctrica a la cual se puede conectar.

Algunos de los apartados que se muestran son el tipo de motor, si es de alta eficiencia, uso general o demás, el tipo de carcasa (este dato es útil para solicitar refacciones), la capacidad o potencia del motor la cual generalmente viene dada en caballos de fuerza (HP) o en kilowatts (kW), el factor de servicio es un detalle de cuánto puede ser sobrecargado el motor bajo carga continua sin dañarse, características eléctricas como la tensión, consumo de corriente en su régimen nominal, frecuencia, fases y formas de conexión, la cual puede ser en estrella o delta, año de fabricación y normas de fabricación (aislamiento, diseño y eficiencia)


<Name of Manufacturer>					
ORD. No.	1N4560981324				
TYPE	HIGH EFFICIENCY		FRAME	286T	
H.P.	42		SERVICE FACTOR	1.10	3 PH
AMPS	42		VOLTS	415	Y
R.P.M.	1790		HERTZ	60	4 POLE
DUTY	CONT		DATE	01/15/2003	
CLASS INSUL	F	NEMA DESIGN	B	NEMA NOM. EFF.	95
<Address of Manufacturer> 					

Figura 5.24. Placa de características de un motor eléctrico de inducción. Fuente: Alkadhim, 2020.

Cuando se trata de motores un apartado importante a tomar en cuenta, es la temperatura que soportan los devanados sin sufrir daños permanentes, esto se demuestra con la clase de aislamiento, la cual, por ejemplo, en la figura 5.24, corresponde a la letra F, estas se encuentran regidas mediante la norma NEMA MG-1 12.43, la cual establece el límite permitido de temperatura para un motor eléctrico tanto monofásico como polifásico, los valores en grados Celsius se muestran en la figura 5.25, donde se hace referencia al factor de servicio (SF) como parámetros a considerar y en la sección derecha están las letras que se encuentran en la placa de características.

Clase de aislamiento (ver 1.65	A	B	F*	H†
Ratio de tiempo (debería ser continuo o cualquier corto tiempo dado en 10.36)				
Incremento de temperatura (basado en una temperatura ambiente máxima de 40°C)				
a. Devanados, por método de resistencia				
1. Motores con factor de servicio de 1.0 o los mencionados en los apartados				
a.3 y a.4	60	80	105	125
2. Todos los motores con factor de servicio igual o mayor a 1.15	70	90	115	...

Figura 5.25. Clases de aislamiento. (Traducido por Antony Alfaro). Fuente: NEMA, 2010.

Por tanto, y a modo de ejemplo, si se toma la letra F de la placa de características de mostrada en la figura 5.24, y basado en que el factor de servicio es inferior a 1.15, se puede establecer que la temperatura no deberá poseer un incremento de más de 105°C respecto a la temperatura ambiente la cual deberá ser como máximo de 40°C.

5.12.1. Vibración

Debido a que las máquinas antes mencionadas son dinámicas (pues posee partes en movimiento), una de las características que se manifiestan en su funcionamiento es la vibración, la cual no es más que una oscilación respecto a un punto de referencia de forma cíclica, debido a las fuerzas internas en el interior de la máquina. Las causas de una vibración pueden ser muchas como un desbalanceo en partes rotativas, desalineamiento, fallas en rodamientos entre otras más.

La vibración medida en la práctica no solo depende de las fuerzas internas antes mencionadas, sino que también depende de la estabilidad y robustez de la máquina y, si esta cuenta con dispositivos para atenuar estas oscilaciones como resortes, contrapesas dinámicas y demás.

Cuando se realizan mediciones y análisis los parámetros de las vibraciones en máquinas rotativas son una parte fundamental que se debe conocer, estos se muestran en la figura 5.26, en ella, se representa una onda con las distintas variables de amplitud (magnitud de la onda o vibración) y pueden ser, según Pruftechnik (2016):

- **Amplitud pico:** amplitud máxima desde el punto de equilibrio.
- **Amplitud pico-pico:** diferencia entre el punto negativo y positivo máximo alcanzado.
- **Amplitud RMS:** corresponde a la raíz cuadrada del promedio de los cuadrados de la señal, si fuera una onda senoidal pura el valor será 0.707 veces el valor pico.
- **Período (T):** tiempo que se necesita para completar un ciclo, lo que es lo mismo a decir cuando el sistema alcanza la posición inicial del análisis bajo el mismo comportamiento y viniendo de la misma dirección.

A menudo, se emplea más el término de frecuencia en lugar del período, generalmente dado en Hertz, este se calcula como 1 dividido entre el período, o en ciclos por minuto (multiplicar por 60).

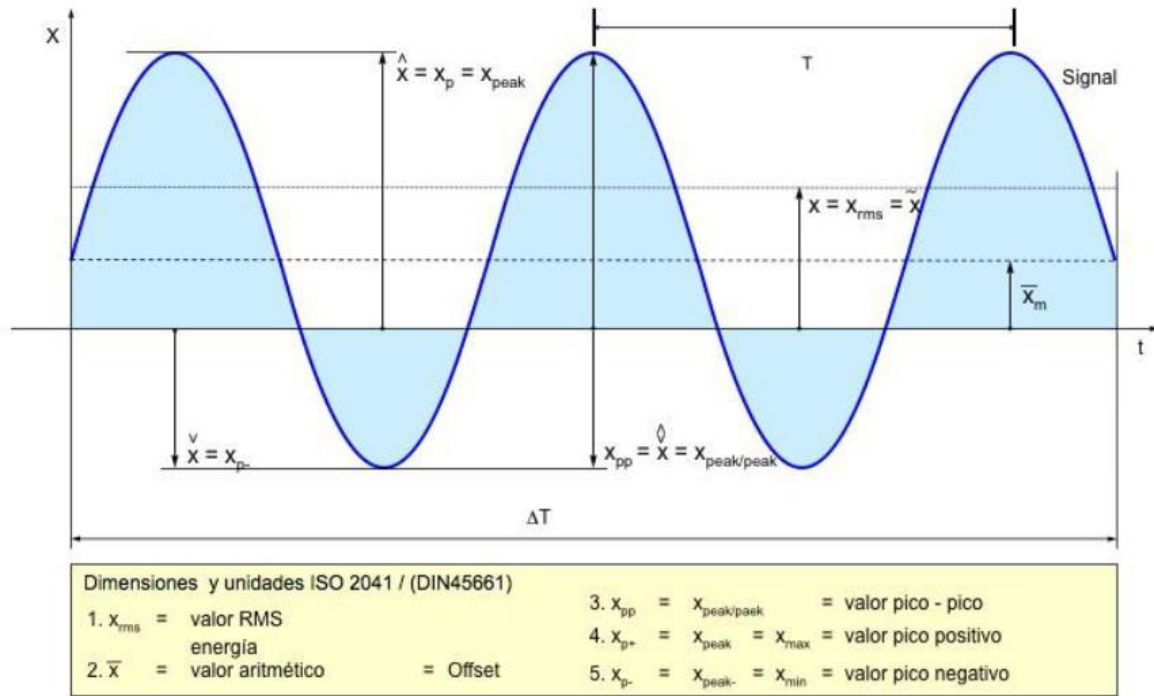


Figura 5.26 Parámetros de vibración. Fuente: Pruftechnik, 2016.

Cuando se desea determinar la vibración, se debe colocar un sensor que permita tomar estas medidas, la amplitud puede poseer distintas magnitudes como aceleración (medido en g), velocidad (medido en mm/s o en in/s) o desplazamiento (mm o in). Los sensores deben colocarse en puntos estratégicos, pero, a menudo, estarán en lugares propensos a sufrir desperfectos como cajas de rodamientos o cajas de muñoneras, siempre sobre un lugar firme.

Algunos de los sensores o acelerómetros empleados pueden ser de tipo piezoeléctrico, donde una oscilación provoca que su lámina piezoeléctrica se flexione y provoque una tensión proporcional a la magnitud de la vibración, aunque la sensibilidad disminuye con el aumento de la frecuencia de resonancia. Para un mejor análisis de las señales producidas por el sensor, se emplea una transformada de Fourier, con el fin pasar la gráfica desde magnitud vs tiempo hacia magnitud vs frecuencia, lo cual hace que en lugar de observarse ondas se formen picos, lo que hace que su observación sea mucho más eficiente.

Con el fin de mejorar la repetitividad, la constancia y las buenas prácticas en las mediciones para tomar datos significativos, surgen las normas, estas están basadas en la experiencia de muchos profesionales y diversos estudios logrando establecer zonas con valores estándar y el estado de la maquinaria. Así, tomando mediciones y comparándolas con los parámetros colocados en tablas dentro de normas, se establecen pautas para generar alertas, órdenes de mantenimiento o reemplazos inmediatos en partes.

Una norma que es de bastante utilidad es la ISO-10816 en la cual su contenido es referente a la evaluación de vibraciones de máquinas de diversos tipos ya sean en partes fijas o móviles, además de establecer las condiciones y procedimientos de medición, este estándar cuenta con 5 secciones: indicaciones generales, turbinas de vapor y generadores que superan los 50MW, maquinaria industrial con potencia nominal mayor a 15kW, conjuntos movidos por turbinas de gas excepto en aeronáutica y, finalmente, conjuntos de máquinas en plantas de hidro generación y bombeo. Un ejemplo de los valores establecidos y sus alertas se muestran en la figura 5.27.

								v r.m.s. mm/s	v r.m.s. inch/s	Velocidad Vibración (2 - 1000 Hz n > 600 1/min n > 120 1/min)
								11	0.433	
								7.1	0.280	
								4.5	0.177	
								3.5	0.138	
								2.8	0.110	
								2.3	0.091	
								1.4	0.055	
								0.71	0.028	
rigida	flexible	rigida	flexible	rigida	flexible	rigida	flexible	Fundación		
Bombas > 15 kW radial, axial, diagonal				Máquinas medianas 15 kW < P ≤ 300 kW		Máquinas Grandes 300 kW < P < 50 MW		Tipo de Máquina		
Acople directo		Eje intermedio / Poleas.		Motores 160 mm ≤ H < 315 mm		Motores 315 mm ≤ H				
Grupo 4		Grupo 3		Grupo 2		Grupo 1		Grupo		

A Puesta en operación recientemente.

B Operación para largo plazo.

C Operación para corto plazo

D Vibración causando daños

Figura 5.27. Vibración en maquinaria y su estado, basado en ISO 10816-3. Fuente: Pruftechnik, 2011.

6. Desarrollo del proyecto (En relación con el objetivo general)

Para el desarrollo, se empleó como guía de los estatutos de la norma ISO 17359:2018, la cual hace referencia al procedimiento adecuado para establecer un monitoreo en la condición de los equipos, como se observa en la figura 6.1. Este flujograma está basado en un problema, ya especificado, en el caso de este proyecto, se tiene como inconveniente la falta de monitoreo y de información respecto al funcionamiento del sistema de bombeo de agua potable del campus central del Tecnológico de Costa Rica, tal y como se mencionó en la sección 4.3 del presente documento.

El monitoreo en la condición de los equipos debe ser direccionado a identificar y evitar las causas de los modos de fallas de las máquinas en estudio, con el fin de evitar las paradas imprevistas en el funcionamiento de los equipos y, a la vez, extender la vida útil de estos la mayor cantidad de tiempo posible. El método ideal consiste en pronosticar una falla con suficiente antelación para prever al equipo de mantenimiento sobre las acciones que deberán desarrollarse en las próximas tareas de preservación del bien.

Para el primer punto de la metodología expuesta por la norma ISO 17359:2018, donde se debe determinar la relación de costo beneficio, se plantea para este proyecto desarrollar un prototipo totalmente funcional de las variables a medir junto con el muestreo de los datos físicos (sensores), puesto que es necesario minimizar los costos asociados para el desarrollo, así, se determina que los captadores serán lo más accesibles posible, con la ventaja que pueden sustituirse a lo largo del tiempo por otros donde la precisión y el grado industrial sea mayor conservando el mismo sistema de toma de decisiones y software empleado.

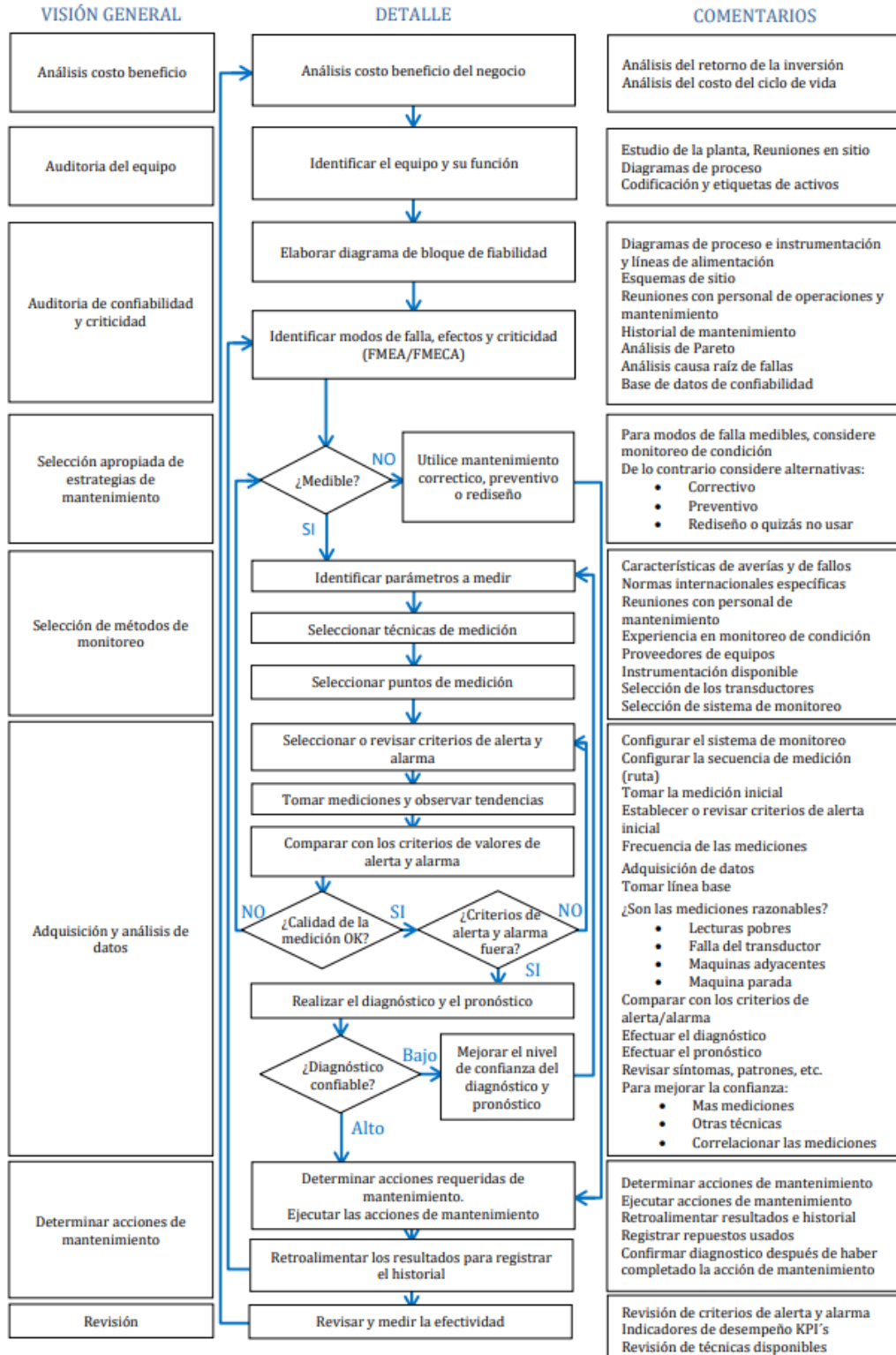


Figura 6.1 Procedimiento para el monitoreo de condición (Traducida por MSc. Ing. Harold Alconz).
Fuente: ISO 17359, 2018.

6.1. Auditoría de equipos (Primer objetivo específico)

Antes de aplicar un sistema de monitoreo como tal, es importante identificar las máquinas que se encuentran instaladas, el sistema de bombeo del TEC posee tres bombas centrífugas de tipo radial de la línea Jacuzzi de la serie DB2 cuyo fabricante es Franklin Pumps, posee un impulsor de bronce y su cuerpo es de hierro fundido, estas son impulsadas por motores de 7.5 caballos de potencia a 3500 revoluciones por minuto. Estas bombas funcionan para llenar un tanque elevado el cual almacena el agua que, luego, pasa a la etapa de filtrado y purificación mediante lámparas ultravioleta.

Las bombas funcionan bajo el método de alternancia, es decir, solo una de ellas estará en funcionamiento cuando se necesita llenar bajo volumen de agua en el tanque elevado, el cual cuando se encuentra lleno posee cerca de 530 metros cúbicos de agua, si se requiere más caudal, pueden trabajar 2 de ellas y si se necesita la máxima capacidad de llenado pueden funcionar las 3 de forma simultánea.



Figura 6.2. Bombas centrífugas instaladas. Fuente: Elaboración propia.



Figura 6.3 Bombas centrífugas y motores instalados. Fuente: Elaboración propia.

Como se observa en la figura 6.3, los motores corresponden a dos marcas distintas, Baldor y US Electrical Motors, sin embargo, ambos poseen las mismas capacidades y están fabricados bajo los mismos estándares, funcionan a una tensión nominal de 460V y consumen en su régimen normal de trabajo 9,6 Amperes los motores de US Electrical, y 9 Amperes el Baldor, ambos poseen una clase de aislamiento tipo F en los devanados y un factor de servicio de 1.15 lo que implica que según la figura 5.25 basado en la norma NEMA MG-1 12.43 la temperatura no deberá sobrepasar un incremento de 115 grados respecto a la temperatura ambiente.

Las bombas poseen una toma de succión de un tamaño de 3 pulgadas con conexión tipo NPT y su salida corresponde a un diámetro de 2 pulgadas, de igual forma, mediante uniones NPT, los sellos son de un elastómero de tipo BUNA, el cual es adecuado para aplicaciones, donde la limpieza es importante, así como para presiones considerables y algunos de sus componentes son de acero inoxidable tipo 300, el cual es bastante resistente frente a la corrosión.

Respecto a la criticidad de los equipos, estos resultan ser de suma importancia, ya que permiten el suministro de agua potable hacia el campus mediante el llenado del tanque elevado, además, esta es la única forma de llevar a cabo esta acción, por el momento y en el tiempo de las visitas el TEC se encuentra laborando de forma diferenciada y bajo condiciones atípicas, por lo que no existe tanto consumo como si lo habría en su estado normal, ya que muchos estudiantes y funcionarios estarían en las instalaciones.

El hecho de ser redundantes y funcionar bajo alternancia hace que la monitorización y los sensores deban colocarse en todas las máquinas, puesto que si, por el contrario, una sería respaldo de otra, bastaría con colocar sensores en la principal, así cuando exista algún fallo o alerta, se pueda sacar de funcionamiento y reparar sin necesidad de instalar sensores de más.

Basado en la entrevista al personal de mantenimiento, cuando una bomba se daña, la empresa tercerizada encargada de esta labor de revisión informa al departamento para que sea trasladada a un sitio para su reparación, lo que implica costos altos, aunado a una pérdida en la capacidad de bombeo y, en casos de extremo consumo, cabría la posibilidad de un faltante del preciado líquido.

Además, el costo de reemplazo de una bomba centrífuga es alto, comparado con la inversión necesaria para brindar un mantenimiento adecuado, una falla súbita en alguno de los componentes podrá repercutir fuertemente en el desempeño de la máquina y, dado que el mantenimiento es tercerizado, no existen rutinas para el control dentro de los 2 meses de cada visita, por lo que, si ocurriese alguna imperfección, no se sabrá hasta tiempo después, siendo el único responsable de detener el flujo de corriente un dispositivo protector de sobrecarga, instalado en el panel de control.

6.2.Modos de falla (Segundo objetivo específico)

Una vez que se han detectado los equipos involucrados en realizar su estudio correspondiente, es importante definir cuáles son los posibles fallos que se puedan presentar, así como su posible causa, es importante aclarar que se deben tomar en cuenta las dos secciones de una bomba, como lo son la parte mecánica y la parte eléctrica (motor), pues son parte de una misma máquina. Karassik (2001) menciona en su libro *Pump Handbook* que los problemas en la bomba pueden ser tanto hidráulicos como mecánicos.

Para el primer rubro de fallos, la bomba puede dejar de entregar líquido en su salida, funcionar con una presión insuficiente o un caudal inferior al nominal, o en el peor de los casos perder el cebado luego de iniciar. Toda bomba centrífuga debe estar con líquido en su interior en todo momento, ya que, de lo contrario, provocará daños en el impulsor y sobrecalentará los empaques y rodamientos los cuales son enfriados mediante el líquido que se trasiega, y la capacidad de succión se pierde también.

Para la segunda categoría, también se pueden incluir como síntomas un consumo excesivo de corriente ya que algunas partes o problemas pueden ocasionar oposición al giro, lo que implicaría más torque necesario para moverse, la forma de compensar esto es mediante un mayor consumo de energía del motor, lo cual provocará fallos en el aislamiento, debido al calentamiento por altas corrientes. Vibraciones, sonidos o roturas en la carcasa o piezas mecánicas son otros de los síntomas de posibles fallos mecánicos.

Es importante recalcar que existe una interdependencia entre algunas de los problemas que se pueden presentar, por ejemplo, una falla en los rodamientos puede causar un desbalance que provoca un espacio extra en el impeler, lo que implica una disminución en la capacidad de entrega.

Los síntomas que se podrían presentar en una bomba centrífuga, Karassik (2001) los ha clasificado en una tabla, la cual se muestra en la Tabla 6.1, en ella, se han asociado a su respectiva causa del problema, mediante números que se pueden comprobar en la tabla 6.2.

Tabla 6.1. Síntomas de problemas en bombas centrífugas.

Síntomas	Posible causa de problemas (cada número se define en la Tabla 6.2)
1. La bomba no entrega líquido	1, 2, 3, 5, 10, 12, 13, 14, 16, 21, 22, 25, 30, 32, 38, 40
2. Flujo de salida insuficiente	2, 3, 4, 5, 6, 7, 7a, 10, 11, 12, 13, 14, 15, 16, 17, 18, 21, 22, 23, 24, 25, 31, 32, 40, 41, 44, 63, 64
3. Presión de salida insuficiente	4, 6, 7, 7a, 10, 11, 12, 13, 14, 15, 16, 18, 21, 22, 23, 24, 25, 34, 39, 40, 41, 63, 64
4. La bomba pierde el cebado después de arrancar.	2, 4, 6, 7, 7a, 8, 9, 10, 11
5. La bomba requiere una potencia excesiva para funcionar	20, 22, 23, 24, 26, 32, 33, 34, 35, 39, 40, 41, 44, 45, 61, 69, 70, 71
6. La bomba vibra o hace ruido en todo momento	2, 16, 37, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 67, 78, 79, 80, 81, 82, 83, 84, 85
7. La bomba vibra o hace ruido a bajos caudales	2, 3, 17, 19, 27, 28, 29, 35, 38, 77
8. La bomba vibra o hace ruido a altos caudales	2, 3, 10, 11, 12, 13, 14, 15, 16, 17, 18, 33, 34, 41
9. El eje oscila axialmente	17, 18, 19, 27, 29, 35, 38
10. Las paletas del impulsor están erosionadas en el lado visible	3, 12, 13, 14, 15, 17, 41
11. Las paletas del impulsor están erosionadas en el lado invisible	12, 17, 19, 29
12. Las paletas del impulsor se erosionan en la descarga cerca del centro	37
13. Las paletas del impulsor se erosionan en la descarga cerca de obenques o en filetes de obenques / aspás	27, 29
14. Las cubiertas del impulsor están arqueadas o fracturadas	27, 29
15. La bomba se sobrecalienta y se atasca	1, 3, 12, 28, 29, 38, 42, 43, 45, 50, 51, 52, 53, 54, 55, 57, 58, 59, 60, 61, 62, 77, 78, 82
16. Las partes internas se corroen prematuramente	66
17. Las holguras internas se desgastan demasiado rápido	3, 28, 29, 45, 50, 51, 52, 53, 54, 55, 57, 59, 61, 62, 66, 77
18. La carcasa partida axialmente se corta	63, 64, 65
19. Las juntas estacionarias internas se cortan	53, 63, 64, 65
20. La caja empaquetada gotea excesivamente o empaqueta tiene corta vida	8, 9, 45, 55, 57, 68, 69, 70, 71, 72, 73, 74
21. Caja empaquetada: manga ranurada	8, 9
22. El sello mecánico tiene fugas excesivas	45, 54, 55, 57, 58, 62, 75, 76
23. Sello mecánico: caras dañadas, manguito, fuelle	45, 54, 55, 57, 58, 62, 75, 76
24. Los rodamientos tienen una vida corta	3, 29, 41, 42, 45, 50, 51, 54, 55, 58, 77, 78, 79, 80, 81, 82, 83, 84, 85
25. El acoplamiento falla	45, 50, 51, 54, 67

Fuente: Karassik, 2001.

Tabla 6.2. Causas de problemas en bombas centrífugas.

Problemas de succión	Otros problemas hidráulicos (continuación)
1. Bomba no cebada	31. Pérdidas por fricción en descarga superiores a las calculadas.
2. La tubería de succión de la bomba no está completamente llena de líquido	32. Cabeza de presión del sistema mayor que el diseño de la bomba.
3. Insuficiente NPSH disponible	33. Cabeza de presión del sistema inferior al diseño de la bomba.
4. Cantidad excesiva de aire o gas en el líquido	34. Hacer funcionar la bomba a un flujo demasiado alto (para bombas de baja velocidad específica).
5. Bolsa de aire en la línea de succión	35. Hacer funcionar la bomba a un flujo demasiado bajo (para bombas de alta velocidad específica).
6. Fugas de aire en la línea de succión	36. Fugas por válvula check atascada.
7. Fugas de aire en la bomba a través de prensaestopas o sello mecánico.	37. Espacio cerrado entre las paletas del impulsor y la lengüeta de la voluta o las paletas del difusor.
7a. Aire en la fuente del líquido sellante	38. Funcionamiento en paralelo de bombas inadecuadas para ese propósito.
8. Tubería de sello de agua obstruida	39. La gravedad específica del líquido difiere de las condiciones de diseño.
9. La jaula del sello está mal montada en los prensaestopas.	40. Viscosidad del líquido difiere del diseño.
10. Tubería de succión insuficientemente sumergida.	41. Desgaste excesivo en las holguras de funcionamiento internas.
11. Formación de vórtices en la succión	42. Obstrucción en la línea de la línea de balanceo.
12. Bomba operada con válvula de succión cerrada o parcialmente cerrada.	43. Transitorios en la fuente de succión (desequilibrio entre la presión en la superficie del líquido y la presión de vapor en la brida de succión).
13. Filtro de succión obstruido	
14. Obstrucción en la línea de succión	
15. Pérdidas excesivas por fricción en la línea de succión.	
16. Impulsor trabado.	
17. Codo de aspiración en plano paralelo al eje (para bombas de doble aspiración)	
18. Dos codos en la tubería de succión a 90 ° entre sí, creando remolinos y rotaciones previas.	
19. Selección de bomba con una velocidad específica de succión demasiado alta.	
	Problemas mecánicos: general
	44. Materiales extraños en impulsores
	45. Desalineamiento.
	46. Fijación insuficientemente rígida.
	47. Pernos de fijación al suelo sueltos.
	48. Pernos sueltos de la bomba o del motor.
	49. Fijación al piso inadecuada.
	50. Fuerzas en tubería y momentos excesivos en las boquillas de la bomba.
	51. Juntas de expansión mal montadas.
	52. Arranque de la bomba sin un calentamiento.
	53. Superficies de montaje de ajustes internos (en anillos de desgaste, impulsores, manguitos de eje, tuercas de eje, carcasas de cojinetes, etc.) no perpendiculares al eje.
	54. Eje doblado.
	55. Rotor desbalanceado.
	56. Piezas sueltas en el eje.
	57. Eje descentrado debido a cojinetes desgastados
	54. Eje doblado.
	55. Rotor desbalanceado.
	56. Piezas sueltas en el eje.
	57. Eje descentrado debido a cojinetes desgastados

Otros problemas hidráulicos

20. Velocidad de la bomba demasiado alta.
21. Velocidad de la bomba demasiado baja.
22. Sentido de giro incorrecto.

23. Montaje en sentido inverso del impulsor de doble aspiración.
24. Instrumentos no calibrados.
25. Diámetro del impulsor menor que el especificado.
26. Diámetro del impulsor mayor que el especificado.
27. Selección del impulsor con coeficiente de carga anormalmente alto.
28. Hacer funcionar la bomba con la válvula de descarga cerrada sin abrir una derivación (by-pass).
29. Funcionamiento de la bomba con menos del caudal mínimo recomendado.
30. Cabeza de presión más alta que la presión de apagado.

Fuente: Karassik, 2001.

6.3. Parámetros por observar (Segundo objetivo específico)

Tras analizar las tablas 6.1 y 6.2, es posible establecer algunas interdependencias entre algunos de los problemas y su forma de manifestarse, por ejemplo, mediante la vibración se pueden comprobar irregularidades como, rotor desbalanceado, fijación defectuosa, cavitación, rodamientos en mal estado, entre otros, por ello se debe tener un buen criterio a la hora de seleccionar cuáles parámetros se pueden observar, con el fin de abarcar la mayor cantidad de fallas que se puedan presentar con la menor cantidad de sensores posibles.

En concreto, en la tabla 6.1, se menciona en los apartados 6, 7, 8 y 24 causas referentes a síntomas relacionados con la vibración de la máquina, y estos corresponden a la mayoría de las causas mencionadas en la tabla 6.2, algunos otros síntomas también poseen muchas causas asociadas como los puntos 2 y 3, sin embargo, muchos de ellos también se encuentran en los puntos de vibración antes mencionados, por lo que este punto no corresponde a un parámetro con una importancia tan alta como otros.

Otro punto basado en la tabla 6.1 con muchas causas asociadas representa el número 15, este corresponde a problemas asociados con altas temperaturas presentes tanto en parte del motor eléctrico como en la voluta o partes internas de la bomba, por lo que este apartado sí es un parámetro por considerar en las mediciones que puedan desencadenar en problemas de funcionamiento de la bomba.

Por último, otro de los puntos importante es el tema de consumo eléctrico (punto 5, corriente), aunque, si bien es cierto que estos síntomas se pueden transformar en otros como vibración o temperatura, no está de más tener un control y monitoreo sobre este punto, ya sea por el hecho de verificar con suficiente anterioridad problemas eléctricos, así como la carga que posee la bomba.

En el marco de la norma ISO 17359, existen algunas variables recomendadas a monitorear con el fin de realizar un proceso adecuado para velar por el correcto estado de la máquina, como se observa en la figura 6.4, los parámetros seleccionados anteriormente se encuentran también en ella validando así la selección realizada, se debe hacer la salvedad que en el apartado de bombas solo se hace referencia al conjunto dentro de la voluta, por ello, debe tomarse en cuenta esta máquina, así como el motor eléctrico.

Parámetro	Tipo de máquina									
	Motor eléctrico	Turbina de vapor	Turbina aero gas	Turbina de gas industrial	Bombas	Compresor	Generador Eléctrico	Motor RIC	Ventiladores	Transformador de potencia
Temperatura	●	●	●	●	●	●	●	●	●	●
Presión		●	●	●	●	●		●	●	●
Cabeza de presión					●					
Ratio de presión			●	●		●				
Presión de vacío		●			●					
Flujo de aire			●	●		●		●	●	
Flujo de combustible			●	●				●		
Flujo de fluidos		●			●	●				
Corriente	●						●			●
Tensión	●						●			●
Resistencia	●						●			●
Fase eléctrica	●						●			
Potencia de entrada	●				●	●	●		●	●
Potencia de salida	●	●	●	●			●	●		●
Ruido	●	●	●	●	●	●	●	●	●	●
Vibración	●	●	●	●	●	●	●	●	●	●
Emisión acústica	●	●	●	●	●	●	●	●	●	●
Ultrasonido	●	●	●	●	●	●	●	●	●	●
Presión de aceite	●	●	●	●	●	●	●	●	●	
Consumo de aceite	●	●	●	●	●	●	●	●	●	
Tribología del aceite	●	●	●	●	●	●	●	●	●	●
Termografía	●	●	●	●	●	●	●	●	●	●
Torque	●	●	●	●		●	●	●		
Velocidad	●	●	●	●	●	●	●	●	●	
Longitud		●								
Posición angular		●	●	●		●				
Eficiencia		●	●	●	●	●		●		

● indica parámetro de monitoreo aplicable

Figura 6.4. Parámetros por monitorear para evitar fallas en los equipos. Fuente: ISO 17359, 2018.

6.4. Formas de muestreo y ubicaciones (Tercer objetivo específico)

Para realizar un muestreo de forma correcta es importante tomar en cuenta la sensibilidad y tipo de sensores a instalar, ya que, en ocasiones, se pasa por alto las magnitudes de salida y su forma de transformarlas a un valor estándar conocido, muchos de los sensores producen como señal una onda análoga de corriente de entre 4 y 20 miliamperes, aunque también existen con salidas de tensión análogas que varían entre 0 y 10 volts.

En el caso del parámetro de vibración, su medición se realizará por medio de sensores piezoeléctricos de bajo costo, fabricados por Sparkfun, el uso de este tipo de dispositivo recae en la necesidad de emplear sensores de bajo costo, pero que provean la capacidad de tomar mediciones confiables y sin tener que realizar intervenciones al equipo de bombeo instalado. Si bien es cierto que este tipo de sensores puede tener dificultades o interferencias en sus medidas, funciona como un buen punto de partida en la creación de un prototipo el cual puede ser escalable mediante la instalación de nuevos sensores, más del tipo industrial.

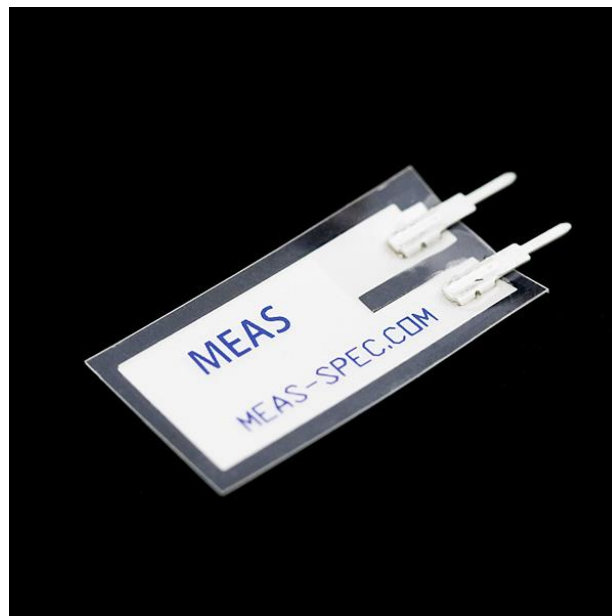


Figura 6.5. Sensor de vibración piezoeléctrico. Fuente: Sparkfun, 2020.

Estos sensores piezoeléctricos se colocan mediante el uso de algún adhesivo al punto de medición deseado, dejando en voladizo las terminales para dejar un punto en el cual pueda oscilar cuando la bomba entre en funcionamiento, las flexiones que se producen en el cuerpo del sensor provocan una tensión, la cual puede ser registrada mediante algún dispositivo con capacidad de obtener mediciones analógicas, no requiere de ninguna fuente externa para su puesta en marcha, una de las terminales se coloca en algún terminal a tierra (GND), mientras que la otra es la que posee la señal.

Como cualquier sensor que emplee como principio de funcionamiento un piezoeléctrico, entre mayor sea la flexión mayor será la tensión que se producirá, en el caso de este sensor LTD0, las diferencias de potencial entre sus terminales pueden llegar a ser peligrosas y dañar la entrada de medición de otros dispositivo por lo que se recomienda instalar una resistencia con un valor de 1 mega Ohm entre sus terminales, de modo que la salida esté en un rango accesible y sin riesgo (-0,5V hasta 5V).

En muchas ocasiones, se puede variar la sensibilidad de este sensor añadiendo pesos a uno de sus extremos facilitando una flexión con bajas fuerzas u oscilaciones en la máquina, sin embargo, para las bombas tratadas en este proyecto, no es necesario añadirse de modo que la sensibilidad final indicada por el fabricante será de 50mV/g, es decir, que cada 50mV en la señal equivale a un g de aceleración, cuando se tiene una resonancia (incremento de la amplitud por acoplamiento de las oscilaciones) la sensibilidad pasa a ser de 1,4V/g.

Si bien es cierto que, para dar una indicación y valoración adecuada se recomienda instalar al menos dos sensores de vibración para cada máquina, con el fin de abarcar las oscilaciones en dos direcciones para este caso, se colocará solamente uno para tomar mediciones iniciales y formar un punto de partida hacia un criterio adecuado de la toma de datos y su posterior procesamiento.

La ubicación de los sensores no podrá determinarse mediante prueba y error, de hecho, con el fin de formular un buen criterio y buenas prácticas en el montaje y calidad en las mediciones obtenidas existen puntos de medición estandarizados, la norma ISO 10816 establece que, para una bomba centrífuga, se deben colocar en puntos alineados con los ejes cartesianos y, por tanto, desplazados 90° entre sí, en la figura 6.6, se muestran estos puntos identificados con las letras A,H y V, para el caso de las bombas instaladas se procederá a instalarlo en el punto V, en el lugar de acople entre el motor eléctrico y la bomba.

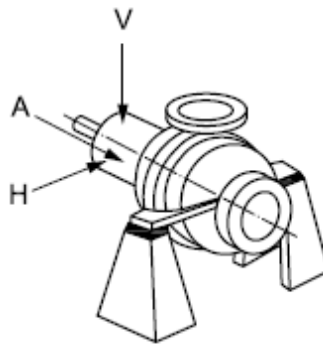


Figura 6.6. Puntos recomendados de medición de vibraciones. Fuente: ISO, 2009.

Otro de los parámetros que se monitorearán corresponde a la temperatura, para ella y siguiendo el modelo de bajo costo antes mencionado, se emplearán sensores que utilizarán el principio de los diodos cuando sufren cambios en la temperatura, cambiará la tensión en cierta proporción establecida, este sensor posee el código TMP36, de encapsulado similar a un transistor, dado su reducido tamaño hace fácil su instalación. Posee un rango de medición de temperatura bastante amplio, en concreto logra captar entre -40°C y 125°C , lo cual lo hace ideal para esta aplicación. Este sensor posee tres terminales donde una de ellas deberá conectarse a una fuente de 5V otra estará conectada a tierra (GND) y la última será la salida de la señal correspondiente a su medición. La precisión de este dispositivo es de $\pm 1^\circ\text{C}$, y su salida varía entre 0,1V (-40°C) y 2V (125°C).

Respecto a la instalación de este sensor, se debe tener en cuenta que la temperatura normada y detallada en los datos de placa de los motores (dada mediante un código) es respecto a los devanados internos del estator, medir temperatura en estos puntos es bastante difícil y los equipos para realizar esto son costosos, sin embargo, una práctica consiste en tomar mediciones de la superficie del motor, con la salvedad que la temperatura en esta ubicación usualmente difiere por concepto de transferencia de calor respecto a los devanados internos entre 15°C y 20°C, dependiendo de la construcción y tipo de motor (US Motors, 2020).

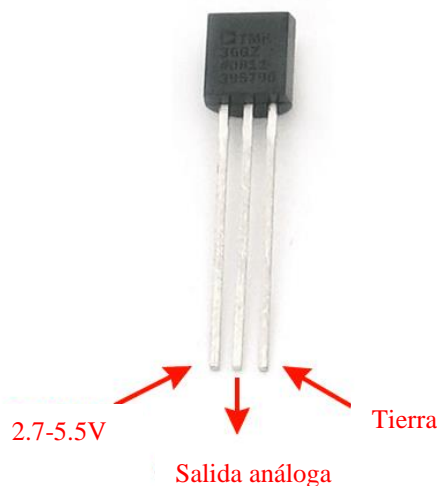


Figura 6.7. Sensor de temperatura TMP36. Fuente: Adafruit, 2020.

Estos sensores se colocarán en la superficie superior de cada motor, de modo que exista un contacto físico entre el dispositivo y la máquina para mejorar la percepción y que la desviación respecto a la temperatura real del motor sea la menor posible, haciendo que sus datos sean confiables, se buscarán puntos calientes en la carcasa para colocarlos en los espacios más críticos.

Aunque se recomienda realizar mediciones del consumo de corriente del motor, para la primera etapa de monitoreo se plantea usar solo las variables de temperatura y vibraciones, con la posibilidad de instalar a futuro más sensores para observar más parámetros, mejorando la estimación en el estado de la máquina.

6.5. Establecimiento de alarmas iniciales

Debido a que la mayoría de los métodos de inteligencia artificial que se pueden desarrollar, requieren de un set de datos de entrenamiento bajo los cuales se puede establecer un aprendizaje automático, donde el mismo modelo define las regiones y límites para alarmas y las bombas que se encuentran instaladas en el campus central no cuentan con un histórico de datos, es necesario recurrir a normas que permitan definir valores umbrales sobre los cuales el sistema alertará de funcionamientos anómalos.

Con respecto a la vibración, existen normas que definen umbrales aceptables de trabajo y puntos donde establecer alertas y, finalmente, valores donde la máquina debería apagarse para evitar daños más significativos, por ejemplo, la ISO 10816-7, la cual posee un detallado escrutinio en el establecimiento de valores de vibración para determinar zonas de operación de la máquina (A, B, C y D).

Bajo la norma ISO recién mencionada, se hará un detalle de cada zona para mejorar su comprensión, para la zona A, se establecen estados para bombas nuevas en un rango operativo aceptado, B es usado para establecer condiciones anormales, pero bajo las cuales la bomba puede operar en largos periodos de tiempo, en C las bombas deben trabajar limitadamente y en D ya existe un riesgo inminente de daño. Las zonas de alarma y de alerta poseen rangos con un máximo de 1,25 veces la zona de referencia, para la primera de ellas se debe usar la zona B y para la alerta la zona C.

También existen distintas categorías de bombas I y II, para la primera categoría se incluyen bombas con un alto nivel de confiabilidad, disponibilidad y para razones de seguridad o aplicaciones críticas, ciertamente, las bombas de agua potable recaen en esta categoría.

Una vez claros los puntos a considerar para un uso correcto de las tablas presentadas se procede a aplicarlos en la tabla A.1 (mostrada en apéndices del presente documento), en ella, se establece que el valor máximo para determinar una alarma es de 5mm/s el cual es 1,25 veces el valor de la zona B (4mm/s) y, además, debe establecerse una alerta donde se debe detener la máquina a la mayor brevedad posible cuando el valor de las vibraciones iguale o sobrepase los 8,3mm/s.

Sin embargo, dado que los sensores empleados para el muestreo de datos serán piezoeléctricos, cuya salida estará en unidades de aceleración (g's), se deberá pensar en algunas tablas que sean equivalentes a la ISO antes mencionada, pero, con las unidades apropiadas, así, según la publicación de Graney y Starry (2012), se establecen 4 zonas, al igual que la norma, pero con unidades en (g's).

Si bien es cierto que la alarma debería establecerse en la zona 3, debido a que en la zona 2 existe una probabilidad moderada de daño crítico y ya se presenta un pequeño desgaste en la banda de rodadura, es preferible avisar al usuario en esta zona, para que se vayan tomando previsiones de mantenimiento y, aunado a que en una bomba se pueden presentar otros daños por distintas causas mecánicas e hidráulicas, es preferible establecer un valor de 3,5G como límite para declarar una alerta por causa de vibraciones anómalas.

El valor global de aceleración

Amplitud de aceleración Pk-Pk (g)

RPM	Etapa 1	Etapa 2	Etapa 3	Etapa 4
1200	2,40	4,80	9,60	19,20
1800	4,10	8,20	16,40	32,80
3600	10,20	20,40	40,80	81,60

Amplitud de aceleración RMS (g)

RPM	Etapa 1	Etapa 2	Etapa 3	Etapa 4
1200	1,3	2,6	5,2	10,4
1800	1,5	3,0	6,0	12,0
3600	1,75	3,5	7,0	14,0

Valores publicados por Graney y Starry, "Rolling Element Bearing Analysis" Revista Material Evaluation Vol 70, pp. 78-85. ASNT

Figura 6.8. Valores de alerta por vibraciones. Fuente: Graney y Starry, 2012.

En el caso de la temperatura, debido a que los motores presentan una clase de aislamiento tipo F, el máximo valor que pueden soportar será un incremento de 115°C respecto a la temperatura ambiente que para valores máximos según normas será de 40° dando como resultado un valor de 155°C, sin embargo, debido a lo comentado en apartados anteriores por US Electric, cuando se realizan mediciones desde la superficie pueden existir variaciones de entre 15°C y 20°C.

Si se toma como temperatura ambiente máxima 27°C (basado en estadísticas de WeatherSpark) y se suma el máximo incremento permitido se obtiene 142°C y restando la máxima desviación resulta en 122°C, pero, debido al lugar donde se encuentran las bombas aunado a la incertidumbre del sensor de temperatura a emplear, es recomendable poseer un rango con un margen de error, es por este motivo que se establece como valor de umbral para mostrar una alarma de 100°C.

Para el caso de las características de corriente y temperatura, se emplean recomendaciones de fabricantes como US Electric, donde mencionan que, de acuerdo con normas como NEMA MG 1 12-44, solo se permiten variaciones de no más de 10 % de la tensión de alimentación del motor en régimen continuo de trabajo, así, si el motor está conectado a 460V, cualquier valor fuera del rango comprendido entre 414V y 506V, sin embargo, estos son los valores máximos, por lo que se

plantea el reducir el rango para establecer posibles fallas de medición u otros aspectos, por lo que se establece en 440V el mínimo y 480V el máximo.

Respecto a la corriente que pueda consumir la bomba centrífuga, si se desea aplicar un criterio para determinar cuándo los valores son anormales, debe realizarse en conjunto con otros parámetros ,ya que, cuando se da el arranque, se producirán valores altos por el torque inicial, pero en un régimen de trabajo nominal los valores no deberán cambiar o estar fuera de un rango entre 8,5A y 10A, tomando como corriente nominal 9A, según los datos de placa.

6.6.Modelo para el procesamiento y análisis de datos (Tercer objetivo específico)

6.6.1. Arduino

Como medio para la obtención de los datos provenientes de los sensores se plantea el uso de un Arduino UNO R3, el cual posee la capacidad de tomar los valores de 6 dispositivos análogos, con ello, se pueden conectar los tres sensores previstos de temperatura y los otros 3 restantes para la vibración, para ello, es necesario antes de definir cual terminal corresponde a qué sensor, especificar los nombres que se emplearán para identificarlos dentro del código, de ese modo, resulta más intuitivo el entendimiento y la programación.

Para ello, se hace uso del comando “*const*”, el cual se utiliza para definir una constante en el programa, luego, la palabra “*int*” hace indicar que corresponde a un valor de tipo entero y, finalmente, se hace la asignación del nombre que se va a dar, como por ejemplo, vib1 para representar al sensor de vibración del motor 1 y se le adjudica la constante A0, el resto de las asignaciones se muestran en la figura 6.8. La sección inferior es con el propósito de establecer variables de tipo texto para que los datos transformados de los sensores se guarden en este formato.

```

#include <SoftwareSerial.h>
const int vib1=A0;
const int temp1=A1;
const int vib2=A2;
const int temp2=A3;
const int vib3=A4;
const int temp3=A5;
String a="";
String vb1="";
String vb2="";
String vb3="";
String tmp1="";
String tmp2="";
String tmp3="";
String coma=",";

```

Figura 6.9. Definición de variables Arduino UNO. Fuente: Elaboración propia.

Posteriormente, se le programa al Arduino como instrucción que las terminales A0 hasta A5, ahora etiquetadas con el nombre del sensor correspondiente serán entradas, el comando “pinMode” permite definir a las terminales como entradas o salidas según se requiera, luego, entre paréntesis se especifica el nombre de la terminal y, por último, se indica entrada como “*INPUT*” o salida como “*OUTPUT*”, en la figura 6.9 se muestra este procedimiento, en ella también se ejecuta el comando “Serial.begin(9600)” esta instrucción se usa para iniciar la comunicación Serial con otros dispositivos a una velocidad de 9600 baudios.

```

void setup()
{
  pinMode(vib1, INPUT);
  pinMode(temp1, INPUT);
  pinMode(vib2, INPUT);
  pinMode(temp2, INPUT);
  pinMode(vib3, INPUT);
  pinMode(temp3, INPUT);
  Serial.begin(9600);
}

```

Figura 6.10. Configuración de sensores. Fuente: Elaboración propia.

Ahora bien, una vez definidas las variables que ejecutará el sistema, se debe indicar al programa que lea la entrada, sin embargo, la forma de funcionar de un Arduino al leer una señal de tipo analógica es enviarla a un convertidor de tipo analógico a digital (ADC) y, a su vez, lo convierte en número entre 0 y 1023, de modo que estos serán los datos que se observarán mediante la salida serial.

Para evitar este inconveniente y mostrar el valor con sus respectivas unidades, se debe realizar una transformación mediante ecuaciones matemáticas para tomar el valor numérico de la salida del ADC y basado en los rangos y sensibilidad establecida para cada sensor se hará el cálculo correspondiente.

En el ciclo de ejecución del programa de Arduino “*void loop*”, se efectúan las instrucciones de forma cíclica, mientras el programa esté en funcionamiento y alimentada la placa la indicación “*double*” que se muestra en la figura 6.10 hace relación al tipo de dato que se obtendrá de la operación indicada, la ventaja de utilizar esta recae en lugar de otras como “*float*” recae en la precisión de los datos que se pueden guardar, puesto que “*double*” permite guardar hasta 8 bytes de información en comparación con los 4 que posee el segundo tipo de datos.

Para la modificación de los datos al valor en unidades correctas, se emplea la ecuación, según el tipo de sensor. Para el sensor de vibración piezoeléctrico se utilizan factores de conversión entre unidades, primero, se toma el valor de salida del convertidor ADC, el cual es adimensional y se divide entre el valor máximo que pueda alcanzar (1023), de forma que ahora está normalizado a valores entre 0 y 1, posterior a ello, dado que el sensor posee una resistencia de $1M\Omega$ entre sus terminales la tensión máxima que se podrá medir en el Arduino será de 5V y se multiplica por 1000 para obtener un valor en milivolts, una vez obtenido este valor, ahora se puede utilizar la sensibilidad del dispositivo (1g equivale a 50mV) para dar el valor final en unidades de g.


```

void loop() {
  double vib_bomba1= ((analogRead(vib1)) * 5 * 1000) / (1023.0 * 50); // convierte a g's el valor de entrada
  vb1=String(vib_bomba1,2);
  double vib_bomba2= ((analogRead(vib2)) * 5 * 1000) / (1023.0 * 50); // convierte a g's el valor de entrada
  vb2=String(vib_bomba2,2);
  double vib_bomba3= ((analogRead(vib3)) * 5 * 1000) / (1023.0 * 50); // convierte a g's el valor de entrada
  vb3=String(vib_bomba3,2);
  double temp_bomba1 = ((analogRead(temp1) * 5/1024)-0.5)*100; // temperatura en °C
  tmp1=String(temp_bomba1,2);
  double temp_bomba2 = ((analogRead(temp1) * 5/1024)-0.5)*100; // temperatura en °C
  tmp2=String(temp_bomba2,2);
  double temp_bomba3 = ((analogRead(temp1) * 5/1024)-0.5)*100; // temperatura en °C
  tmp3=String(temp_bomba3,2);
  a=(vb1+coma+tmp1+coma+vb2+coma+tmp2+coma+vb3+coma+tmp3);
  Serial.println(a);
}

```

Figura 6.11. Ciclo de ejecución Arduino. Fuente: Elaboración propia.

En el caso de la temperatura, primero, se debe hacer la normalización del valor de salida del convertidor a un estado entre 0 y 1, luego, se debe convertir a una tensión representativa, la cual corresponda al máximo valor que pueda presentar (5V), ahora bien, en la tensión de salida que se produce existe un valor de offset de 0.5V, por lo tanto, es necesario restarlo del valor mínimo para obtener el valor en milivolts proporcional a la temperatura percibida así un valor de 25°C estará representado como 750mV cuando se hace la resta de 0.5V quedará en 250mV, por ello, para obtener el valor en grados Celsius, es necesario realizar una multiplicación de 100 por el valor obtenido en el paso anterior.

Estos valores se convertirán de un valor de tipo numérico a texto, mediante el comando “*String*”, que se muestra en la línea siguiente de cada operación matemática, luego de obtener la medición de cada uno de los sensores se procede a realizar una concatenación (unión) de texto para formar una línea donde se contengan los valores separados por comas y se asignará a la variable “a”, esta se imprimirá o enviará por medio de la comunicación serial mediante el comando “*Serial.println*” la terminación de “*ln*” le permite escribir dentro de la función Monitor Serie del editor de programación de Arduino mostrar cada revisión en una línea nueva.

La razón de convertir los datos de tipo numérico a tipo texto es que estos se enviarán a través de una red de internet inalámbrica mediante un servidor web capaz de escribir en una dirección IP específica el texto con los valores de cada sensor en forma de lista mediante la variable “a”, para que, posteriormente, se pueda acceder y convertir a números, donde será analizada en la computadora destinada al monitoreo mediante software de inteligencia artificial que se mencionará más adelante, y así mostrar los datos bajo ciertos intervalos y producir alertas si fuese necesario.

Para que los datos puedan ser enviados a una dirección de red, es necesario que el Arduino posea la capacidad de conectarse a la red, para ello, se hará uso del antes mencionado módulo ESP 8266-01, el cual posee la capacidad de funcionar como un servidor web, sin embargo, para su programación, deberán realizarse algunas conexiones en específico, tal y como se muestra en la figura 6.11.

Se empleará una placa de pruebas (*protoboard*) para ejemplificar mejor las conexiones, las terminales VCC y CH_PD del módulo ESP8266-01 deberán conectarse a 3.3V, mientras que GPIO_0 (cuando está puesta en la terminal negativa activa el modo de programación del módulo) y GND deberán conectarse a la terminal negativa del Arduino (GND), las terminales Tx y Rx del módulo deben estar conectadas con las terminales 1 y 0 del Arduino, ya que estas son los puntos de comunicación Serial y por donde se enviarán los datos de programación.

Finalmente, para hacer un “by-pass” en el Arduino, se debe hacer un puente entre la terminal RESET y GND (representado con un cable de color amarillo). Una que la programación se ha cargado al módulo, la terminal de RESET y GPIO_0 se deben desconectar, y las terminales en el Arduino deben intercambiarse como lo indica la figura 5.19.

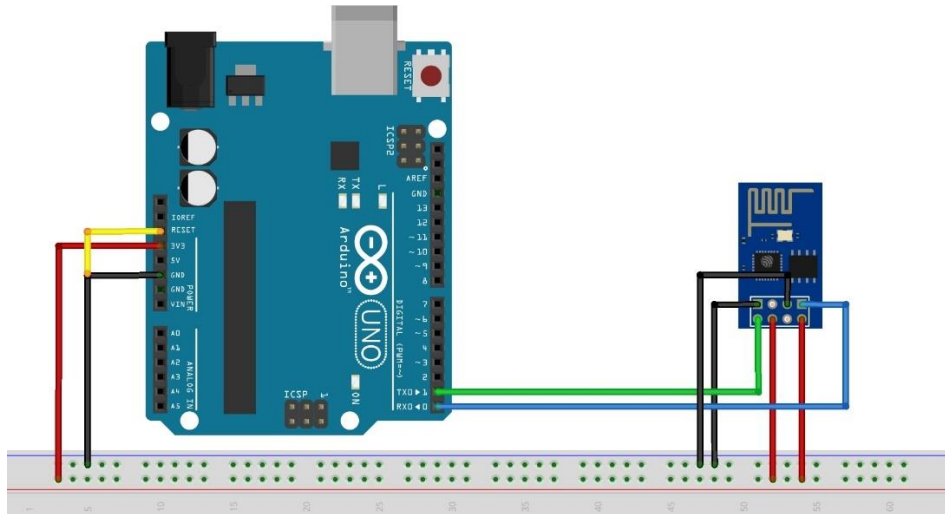


Figura 6.12. Conexión del módulo WIFI ESP8266-01. Fuente: Elaboración propia.

Una vez realizada la conexión se procede a escribir el código para conectar el módulo a la red, para ello, se incluyen librerías de código abierto, con el fin de evitar la programación mediante los códigos AT antes mencionados, en concreto, serían ESP8266WiFi.h y ESP8266WebServer.h, las cuales se instalan mediante el IDE de Arduino, luego, se procede a configurar una dirección de red estática y puerto (80) para el módulo de este modo si ocurriese alguna falla eléctrica o desconexión del servicio, cuando se reconecte volverá a la misma dirección, esto se muestra en la figura 6.12.

```

#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
extern "C" {
#include "user_interface.h"
#include "wpa2_enterprise.h"
#include "c_types.h"
}
// SSID de la red a conectar
char ssid[] = "WTEC";
char username[] = "alfaroanthon@estudiantes.tec.cr";
char identity[] = "alfaroanthon@estudiantes.tec.cr";
char password[] = "2016112120";

ESP8266WebServer server(80);
//Configuración de IP estática
IPAddress staticIP(172, 18, 225, 149); //ESP IP estática
IPAddress gateway(172, 18, 225, 1); //Dirección IP del Router WiFi (Gateway)
IPAddress subnet(255, 255, 255, 0); //Máscara de la subred

```

Figura 6.13. Configuración del módulo ESP8266-01. Fuente: Elaboración propia.

La velocidad de conexión para comprobar en el monitor serial será de 9600, la cual deberá ser igual que la programación del Arduino, para que la información no se corrompa, luego, se hace un

pequeño bucle, donde el sistema se intentará conectar y mostrará un mensaje que el módulo está en este proceso de conexión.

Una vez ingresado a la red, se imprime en pantalla la dirección IP, a modo de confirmación, pues esta debe coincidir con la especificada en pasos anteriores, y se asigna una dirección extra donde los datos serán mostrados, de modo que alguien logra colocar la dirección IP no podrá ver la información hasta que se especifique el apartado “/Python”. En esa misma línea de código bajo el comando “*server.on*”, aunado a la dirección, se especifica la función “*handlePath*”, para que dicho servidor realice ciertas acciones que después se programarán. Una vez establecidos todos los pasos anteriores, se puede arrancar el servidor mediante la instrucción “*server.begin()*”.

En el bucle de ejecución se especifica la instrucción “*server.handleClient()*” esta se encarga de recibir las peticiones de los clientes y lanzar las funciones asociadas al ruteo, en otras palabras, cuando alguno usuario accede a la dirección del servidor, se produce una solicitud de información y se carga un contenido en específico, de forma similar a cuando se accede a una página web desde el navegador, de modo que el sistema estará siempre en monitoreo y a la espera de una conexión.

Por último, dentro de la función “*handlePath*” que realizará el servidor web, se ingresa un comando con el nombre “*server.send*”, el cual sirve para enviar datos al cliente que ingrese a la dirección, seguido, se indica el tipo de contenido, el cual será texto para que pueda ser comprendido por las personas y adicional el tipo de texto como tal, en este caso será plano “*plain*”, ya que este es un subtipo de contenido genérico que puede ser interpretado por la mayoría de programas y navegadores, en la figura 6.13 se muestra este código.

Finalmente, bastará con compilar el código en busca de errores de escritura y luego descargarlo en la memoria ROM del módulo ESP8266-01, donde se debe especificar la placa y el puerto COM donde está conectado el Arduino.

```
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
server.on("/Python", handlePath); //Se asigna una dirección a la IP donde estarán los datos
server.begin(); //Arranca el servidor web
Serial.println("Server listening");
}

void loop() {
  server.handleClient(); // Muestreo de solicitudes de datos
}

void handlePath() { // Qué hacer cuando se soliciten datos
  if (Serial.available() > 0) {
    server.send(200, "text/plain", String(Serial.readString()));
  }
}
```

Figura 6.14. Programación del servidor web. Fuente: Elaboración propia.

Una vez que se han obtenido las mediciones de los sensores, se han convertido a unidades estándar y se han colocado en el servidor web, es necesario tomar los datos y pasarlos a un entorno donde se encuentre la red neuronal que administrará y supervisará el comportamiento de los parámetros con el fin de determinar alertas de fallas.

Para realizar esta adquisición de información, se procede a programar en Python un código simple basado en la librería llamada “*requests*”, una vez importado este módulo se establece mediante una variable llamada “*path*” la dirección en la que se encuentran los datos, la cual debe ser igual al que se especificó en la figura 6.12, luego, la instrucción “*requests.get(path)*” permite ingresar como un cliente y solicitar el contenido que se encuentre disponible y se guarda en la variable “*g*”.

En apartados anteriores, se mencionó el hecho de escribir los datos en una lista separada por comas, esto es con la idea de aplicar el comando “*split*”, donde se indica que la variable “*a*” se

dividirá en secciones cuando se encuentre una coma (,), estos fragmentos pasarán a ocupar posiciones en una lista, por ello, para hacer la asignación correspondiente el dato, se convierte a número con decimales (tipo *float* por comodidad) y se indica la posición correspondiente al dato dentro de la lista “c”.

```
import requests
path='http://192.168.43.90/Python' #si configuro un número de puerto colocar :puerto luego de la ip y antes de /Python
datos= requests.get(path)
a=datos.text
c=a.split(',')
vib1=float(c[0])
temp1=float(c[1])
vib2=float(c[2])
temp2=float(c[3])
vib3=float(c[4])
temp3=float(c[5])
```

Figura 6.15. Adquisición de datos del servidor web. Fuente: Elaboración propia.

6.6.2. Redes Neuronales

Ahora bien, parte de la idea principal en este proyecto es la de presentar algún algoritmo capaz de afrontar y tomar decisiones sobre un histórico de datos en bruto (no están modificados por el empleo de algún método), resulta un tanto complicado la escogencia del mejor método donde sea capaz de aprender con la mínima intervención del usuario, pero brindando resultados claros y precisos.

Al realizar un análisis de los datos que se pueden obtener, como, por ejemplo, vibraciones donde las mediciones pueden oscilar entre valores negativos y positivos, los cuales serán afectados en distintas etapas y presentar comportamientos un tanto difusos y repetibles solo en periodos, resulta complicado en un entorno como el que se tiene en el campus del TEC, donde las bombas centrífugas instaladas no cuentan con ningún histórico de mediciones o datos, definir un método en específico.

Aunque existen múltiples formas de analizar datos mediante *Machine Learning* como los mencionados en apartados anteriores, si no se tiene claro el comportamiento de los datos, aplicar uno de ellos resultaría en incongruencias y la precisión sería bastante baja. El determinar en específico el daño que podría presentar una bomba sin un histórico adecuado resulta muy difícil, por ello, es mejor definir zonas donde existe una probabilidad bastante alta de ocurrencia de daños con el tiempo suficiente para lograr actuar.

Un método que permite clasificar los datos en zonas es el de K-vecinos más cercanos, pero el inconveniente es que, dado que está constantemente analizando la distancia de un dato nuevo con cada uno de los ya existentes para largos set de información, se vuelve un poco inconveniente por los tiempos de procesamiento o por la cantidad de recursos que podría necesitar.

El uso de redes neuronales profundas, por otro lado, no se ve afectado por un gran número de datos, ya que en el inicio son usados para entrenar al sistema, con el fin de encontrar los pesos y las desviaciones de cada una de las salidas de cada neurona y las entradas de datos, luego de ese proceso, ya no son necesarios y simplemente un dato nuevo ingresa y es multiplicado por cada peso ya establecido y, al final, se establece la probabilidad de pertenencia hacia alguno de los rubros aprendidos como salidas.

Una de las principales ventajas que posee este método es que no se necesita determinar, antes de aplicar el modelo, el comportamiento y tendencias de los datos ya que esto se integra dentro de la misma red neuronal, tal y como se observa en la figura 6.15, donde si existen un gran número de entradas, en el ML es imperativo determinar la trascendencia de cada uno, para así filtrarlos antes de aplicar el modelo, mientras que la red neuronal profunda se encarga de determinarlos por él

mismo, por ello, este será el algoritmo que se empleará para determinar las regiones de fallas en las bombas centrífugas de la estación de bombeo del campus central del TEC en Cartago.

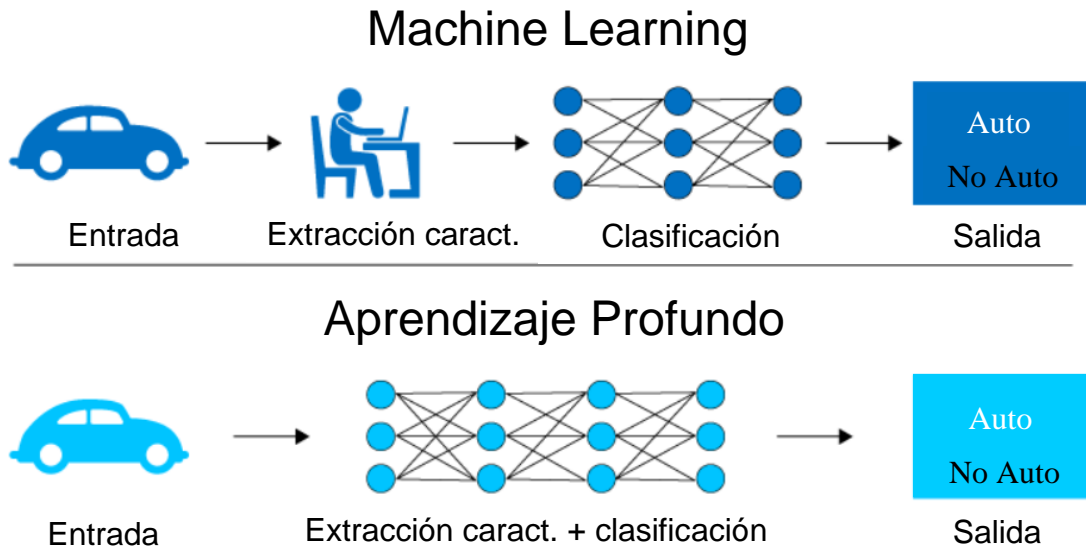


Figura 6.16 Diferencias entre el Machine Learning y Aprendizaje Profundo. (Traducido por Antony Alfaro) Fuente: Kampakis, 2018.

Para llevar a cabo la programación del algoritmo, se hizo uso del convenio existente entre el TEC y la plataforma de cursos virtuales Coursera, para profundizar más en los principios que se mostraron en el apartado 5.4 del presente documento, con él se logra establecer criterios con los cuales sirven como un buen punto de partida en el desarrollo de este algoritmo, algunas de las partes involucradas incluso fueron parte del examen final para optar por el certificado del curso.

Como primer paso, el sistema debe cargar los datos que usará para su entrenamiento y aprendizaje, los cuales se encuentran en un archivo de Microsoft Excel, el cual contiene variables como temperatura, vibración, tensión y corriente, aunado a columnas asociadas a errores en cada rubro, en el caso de las oscilaciones mecánicas se tomaron de un repositorio “*Prognostics Center of Excellence*” de la NASA, en específico el donado por la Universidad de Cincinnati, donde mediante el Centro para la Inteligencia de Sistemas de Mantenimiento (IMS) desarrollaron un

experimento, en el que colocaron un motor para mover un eje con cuatro rodamientos mediante correas.

Los datos del IMS poseen una amplitud dada en aceleración, ya que se los acelerómetros instalados corresponden a la marca PCB modelo 353B33, los cuales poseen una sensibilidad de 100mV/g, la información se tomó del Set de pruebas No. 3, donde se probó a falla los componentes dando como resultado que el rodamiento No. 3 fallara por causa del aro externo.

Sin embargo, dado que la información corresponde a una unión distinta a la que se tiene en las bombas y la aplicación difiere de la presentada en este documento, los valores de alertas se establecerán según el apartado 6.5, del presente documento, tomando los valores globales en valor absoluto, ya que, debido a la oscilación, los movimientos se pueden presentar en direcciones tanto positivas como negativas.

En total, se sustrajeron 122880 mediciones en tres fechas distintas, al inicio de la vida, a nivel de media vida con transición a falla y cuando el rodamiento ya presentaba evidencias de fallas. En la figura 6.17, se muestran los datos elegidos en forma de gráfica, donde se puede observar un aumento paulatino de la amplitud conforme la vida útil del rodamiento llegaba a su fin. Asimismo, cuando existía un valor por encima del umbral, se asignaba un 1 como indicador de alarma y un 0 cuando la medición estaba inferior al límite.

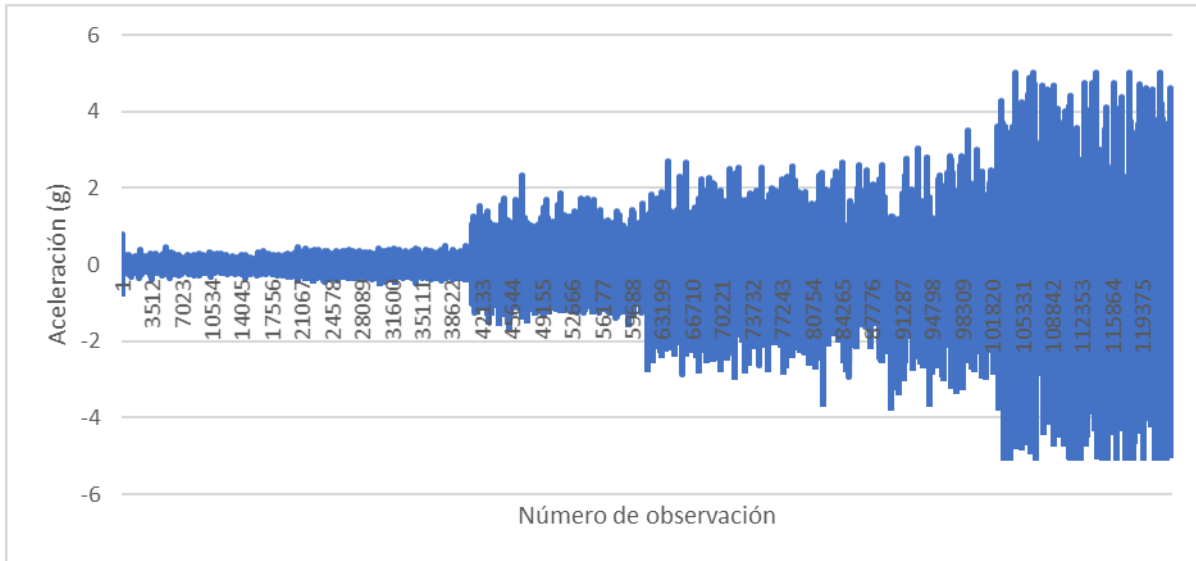


Figura 6.17. Representación de datos de vibración para el entrenamiento. Fuente: Elaboración propia.

Con respecto a la temperatura y, a la vez, mantener una consistencia con la cantidad de datos de entrenamiento, cuando se planteó la información que se empleará para el entrenamiento y, basado en la revisión de temperaturas de WeatherSpark, junto con los valores máximos permitidos, según la norma NEMA MG 1, se estableció el mismo número de datos que la vibración (122880) con un valor aleatorio entre 10°C y 130°C, estableciendo valores de falla (mayor a 100°C) con un 1 y en su estado normal mediante un 0.

De la misma forma, se implementó la base de datos para las variables de tensión y corriente donde se estableció un rango en el cual abarcara los límites mencionados en secciones anteriores y también los sobrepase un poco, con el fin de obtener valores, donde exista una posible falla indicados con un 1 y los que no con un 0. En el caso de la tensión, el rango será desde 428V hasta los 495V y para la corriente será de 8A a 11A.

Una vez cargados los datos de entrenamiento, se almacenarán en variables respectivas, tomando como valores “x” las mediciones de los parámetros antes mencionados y cómo “y” o salida la indicación de 1 o 0, dependiendo si existe una alerta o no, como, en este caso, solo se tiene una

variable de entrada la matriz, en la que se guardan será de tamaño (122880,1), sin embargo, es necesario redimensionar la misma para que quede de la forma (1,122880).

De igual forma, se hace una sección de los datos de entrenamiento, incluyendo las indicaciones o “y”, los cuales servirán como prueba para cuando el modelo finalice su aprendizaje donde se comparará el resultado previsto por el modelo y el real, así, se obtendrá la precisión a modo de evaluar el rendimiento de la red neuronal, estos también se cargarán al programa y se debe realizar los ajustes de la matriz, como se mencionó en el párrafo anterior.

Seguidamente, se deberán introducir los hiper parámetros que usará la red neuronal como configuración, el primero de ellos será la cantidad de neuronas, así como el número de capas que se usarán, estas se guardarán en una lista llamada “*layer_dims*”, donde el primer elemento deberá coincidir con la cantidad de datos de entrada “x”, por ello, se realiza la función “*length*” para averiguar su tamaño y ese dato se introduce bajo el nombre “*numero_x*”. Los demás números corresponden a la cantidad de neuronas en cada capa y los campos representan las capas, como se muestra en la figura 6.18, donde posee la capa de entrada (*numero_x*), 6 capas ocultas y la capa de salida (1), la primera capa oculta será de 20 neuronas, la segunda de 35 y así sucesivamente.

```
layers_dims = [numero_x, 20, 35, 30,12,8,4, 1]
```

Figura 6.18. Dimensiones de la red neuronal para vibraciones. Fuente: Elaboración propia.

Una vez definido el tamaño de la red neuronal, se procede a hacer el llamado al modelo que contiene todas las funciones programadas para llevar a cabo la iteración en la búsqueda de los parámetros o constantes antes mencionadas, en esta función se introducen: los datos de entrada (*train_x*), las salidas reales para realizar la comparación al final (*train_y*), las dimensiones de la red (*layer_dims*), el ratio de aprendizaje (*learning_rate*), la cantidad de iteraciones que se desean realizar (iteraciones) y, por último, una instrucción para decirle al programa que imprima el costo

luego de ciertas iteraciones con el fin de verificar que se está encontrando un mínimo, el esquema de llamado se muestra en la figura 6.19.

```
parametros = modelo_L_capas(train_x, train_y, layers_dims, learning_rate = 0.75, iteraciones = 350, print_costo = True)
```

Figura 6.19. Llamado a la función principal de la red neuronal. Fuente: Elaboración propia.

Posteriormente, la función “*modelo_L_capas*”, toma los valores introducidos en la figura 6.19, y llama a la función “*inicializar_parametros_profundos*”, con el fin de proveer las primeras constantes que se emplearán para establecer las variables “w” y “b”, es importante que sean distintas entre sí, ya que, de lo contrario, el sistema no podrá actualizarse mediante iteraciones, pues todas cambiarían bajo el mismo valor, por ello, se hace uso de una comando que provee valores aleatorios a las variables, con la salvedad que para editar los hiperparámetros se hace que los número es este paso sean los mismos, lo cual se logra mediante la instrucción “*seed*”.

Cuando se establece un “*seed*” con un valor numérico por ejemplo 1, cada vez que se produzcan los valores aleatorios serán los mismos, a modo de ejemplo, si se quieren 3 valores aleatorios entre 0 y 100, cuando la función es llamada, se obtendrán 10,65,30, si no se tuviese el “*seed*” y se vuelve a llamar los números serán 50,30,80, pero si se establece “*seed*” volverán a ser 10,65,30.

Dentro de la función de inicializar los parámetros de forma aleatoria, se tiene que primero se debe verificar la cantidad de neuronas y capas, para ello, se debe tomar el comando “*len*”, el cual devuelve un número entero, posteriormente, se debe realizar un bucle mediante un “*for*” en un rango entre 1 y el tamaño de la red (L), luego, se le etiqueta la constante seguida del número de la neurona, y se le asigna una matriz de dimensiones (tamaño de la capa actual, tamaño de la capa anterior) y se divide entre la raíz cuadrada del tamaño de la capa anterior, ya que la varianza de los pesos de una capa depende de la cantidad de neuronas de la capa anterior, cuanto más grande sea la anterior, menor será la varianza de los pesos de la actual.

La variable “b” se determina, formando una matriz del tamaño (cantidad de neuronas de la capa actual,1) siempre de esa manera, y se llena de ceros, esta no importa que sean iguales, ya que no presenta los problemas de similitud entre neuronas, como si sucedía con los pesos, estos se guardarán en un diccionario y, finalmente, se realizar una verificación de los tamaños de las matrices para ello se hace uso del comando “*assert*”, si existiera algún problema, el programa se detiene e indica el error por diferencias en la forma de la matriz, los datos se guardarán en un diccionario llamado parámetros, con las etiquetas W_1, W_2, \dots, W_n , y b_1, b_2, \dots, b_n . La programación de esta función se muestra en la figura 6.20.

```
np.random.seed(1)
parametros = {}
L = len(layer_dims) # numero de capas de la red

for l in range(1, L):

    parametros['W' + str(l)] = np.random.randn(layer_dims[l], layer_dims[l-1]) / np.sqrt(layer_dims[l-1])
    parametros['b' + str(l)] = np.zeros((layer_dims[l], 1))

    assert(parametros['W' + str(l)].shape == (layer_dims[l], layer_dims[l-1]))
    assert(parametros['b' + str(l)].shape == (layer_dims[l], 1))

return parametros
```

Figura 6.20. Iniciar las constantes aleatoriamente. Fuente: Elaboración propia.

Una vez definidas las constantes, el sistema deberá comenzar a calcular las salidas de cada neurona, las cuales para simplicidad en el tratamiento de datos se trabajarán mediante matrices, aplicando la función ReLU para las capas ocultas y, finalmente, la función sigmoide, para dar la clasificación si existe la alarma o no.

Cuando la salida de la última capa es dada, se calcula la función de costo mediante la ecuación 4 mencionada en el apartado 5.4.4 del presente documento, y se guarda en un vector. Luego, comienza la propagación hacia atrás calculando las derivadas de las funciones de activación y de

las salidas de cada neurona, primero con la etapa de sigmoide y, después, las ReLU, hasta llegar a la primera capa, haciendo uso de las ecuaciones 10 a la 17.

Finalizada la propagación hacia atrás calculando dW y db de cada neurona, se puede aplicar la función de descenso de gradiente multiplicando la derivada de los parámetros anteriores por la ratio de aprendizaje y restándolo a su valor original, todo esto se aplica dentro de una función programada llamada “actualizar_parametros”, luego, se repite el ciclo calculando hacia adelante con los nuevos valores, esto se realiza la cantidad de iteraciones programada, los resultados se muestran en la figura 6.21, en ella, se observa como el costo desciende y, al final, se da la precisión tanto para los datos de aprendizaje como para el set de prueba con resultados excelentes.

```
Número de datos de entrenamiento: 122880
Número de datos de prueba: 537
Forma de la matriz x de entrenamiento original: (122880, 1)
Forma de la matriz y de entrenamiento: (1, 122880)
Forma de la matriz x de prueba original: (537, 1)
Forma de la matriz y de prueba: (1, 537)
Forma de la matriz x de entrenamiento redimensionada:(1, 122880)
Forma de la matriz x de prueba redimensionada: (1, 537)
Costo luego de la iteración 0: 0.750951
Costo luego de la iteración 100: 0.004178
Costo luego de la iteración 200: 0.001899
Costo luego de la iteración 300: 0.001076
Precisión: 0.9998860677083332
Precisión: 1.0000000000000002
```

Figura 6.21. Resultados de la red neuronal para vibración. Fuente: Elaboración propia.

Utilizando la misma estructura, se realiza el aprendizaje para la variable de temperatura, en esta ocasión se ha utilizado un ratio de aprendizaje y cantidad de iteraciones distinta, las capas de la red se han reducido con el fin de aumentar la velocidad de convergencia sin sacrificar la precisión de este, los resultados se muestran en la figura 6.22, para el caso de las variables de tensión y corriente, la red neuronal queda lista, pero, debido a que en la primera etapa no se incluirán, no se aplicaron los métodos para encontrar los parámetros.

```
Número de datos de entrenamiento: 122880
Número de datos de prueba: 537
Forma de la matriz x de entrenamiento original: (122880, 1)
Forma de la matriz y de entrenamiento: (1, 122880)
Forma de la matriz x de prueba original: (537, 1)
Forma de la matriz y de prueba: (1, 537)
Forma de la matriz x de entrenamiento redimensionada:(1, 122880)
Forma de la matriz x de prueba redimensionada: (1, 537)
Costo luego de la iteración 0: 0.702204
Costo luego de la iteración 100: 0.126470
Costo luego de la iteración 200: 0.364334
Costo luego de la iteración 300: 0.045054
Costo luego de la iteración 400: 0.047042
Costo luego de la iteración 500: 0.032107
Precisión: 0.9997151692708334
Precisión: 1.0000000000000002
```

Figura 6.22 Resultados de la red neuronal para temperatura. Fuente: Elaboración propia.

Cuando se concluye con el aprendizaje, los parámetros encontrados se guardan en un archivo de modo que cuando ingrese un dato nuevo la predicción sea lo más rápido posible, para ello, se hace uso de la librería “*pickle*” el cual es un comando para serializar los objetos y transformarlos en archivos para su uso posterior, se utiliza este y no “*json*”, debido a que la cantidad de datos que se debe cargar es alta y se requiere mayor velocidad. En la figura 6.23, se muestra el código para temperatura, si no existe el archivo es creado y guarda los parámetros mediante “*dump*”.

```
with open('Parámetros Temperatura.pickle', 'wb') as f:
    pickle.dump(parametros, f)
```

Figura 6.23 Almacenado de parámetros mediante pickle. Fuente: Elaboración propia.

6.6.3. Almacenado de datos

Por otra parte, es importante que el sistema aprenda de los datos reales que se están tomando de las bombas instaladas, para ello es necesario realizar un guardado de datos bajo intervalos predefinidos, ya que, de lo contrario, sería una cantidad de datos muy grande y se podrían tener problemas de espacio, para ello, se hace uso de la librería “*os*”, la cual permite acceder a funcionalidades dependientes del sistema operativo, como leer y guardar archivos.

Una vez importado el módulo, se establece una dirección donde se guardarán los datos, pero, para evitar que el programa interrumpa su actividad, debido a que la carpeta fue borrada o movida, se hace una función de “*try*” en ella el código intenta hacer una carpeta con el nombre especificado, si no existiese la crea, pero, si ya fue creada, no lo intenta y envía un mensaje informativo que “la creación del directorio fallo”, sin perjudicar el código y, dado que se hará una interfaz gráfica, este mensaje no se verá.

Luego, se establece la dirección de la carpeta que se acaba de mencionar, junto con el nombre y tipo de archivo, el cual será .CSV, por motivos de su flexibilidad y compresión de datos (en comparación a JSON, TSV y otros), pues este puede ser abierto mediante programas comunes como Microsoft Excel, logrando si así se desea graficar u observar las tendencias para otros usos. Se especifican los nombres de las columnas para identificar los números guardados, en la primera columna estará la fecha, luego, la vibración y, por último, la temperatura.

La escritura del nuevo archivo se hará mediante la librería Pandas, donde, si no existe el archivo, lo crea, pero, si ya está, realiza un añadido a los datos existentes, es decir, medición se guardará en la fila siguiente del último dato medido, de esta forma, la información anterior no se borrará y se logrará tener un histórico de cada una de las bombas. El código se muestra en la figura 6.24.


```

import pandas as pd
import os

directorio='datasets' #nombre de la carpeta
try:
    os.mkdir(directorio)
except OSError:
    print("La creación del directorio %s falló" % directorio)
else:
    print("Se ha creado el directorio: %s " % directorio)

path1 = "datasets/Bomba1.csv"
path2 = "datasets/Bomba2.csv"
path3 = "datasets/Bomba3.csv"
columnas = ['Fecha', 'Vibracion', 'Temperatura']
def guardado1(X):
    df1 = pd.DataFrame(X, columns=columnas)
    df1.to_csv(path1, index=None, mode="a", header=not os.path.isfile(path1))

def guardado2(X):
    df2 = pd.DataFrame(X, columns=columnas)
    df2.to_csv(path2, index=None, mode="a", header=not os.path.isfile(path2))

def guardado3(X):
    df3 = pd.DataFrame(X, columns=columnas)
    df3.to_csv(path3, index=None, mode="a", header=not os.path.isfile(path3))

```

Figura 6.24. Guardado de datos. Fuente: Elaboración propia.

6.6.4. Interfaz gráfica

Ahora bien, todo programa, y en específico un sistema experto, debe poseer una interfaz gráfica bajo la cual le permita al usuario, comprender de una forma fácil e intuitiva la información que genere el programa, el código debe quedar oculto, puesto que, de lo contrario, si existe una pequeña modificación en él, hará que el algoritmo deje de funcionar. La GUI, como se le conoce comúnmente por sus siglas en inglés (*Graphical User Interface*), consiste en un conjunto de objetos e imágenes bajo los cuales el cliente podrá interactuar, consultar datos y realizar acciones.

Python posee diversas formas de realizar una GUI, ya sea mediante librerías predeterminadas o mediante la instalación de terceros mediante módulos de acceso gratuito, sin embargo, para la programación, se hará uso de Tkinter, la cual provee los objetos necesarios para esta aplicación.

Como primer paso, se debe definir la pantalla principal, sobre la cual se aplicarán todos los objetos, la geometría será a pantalla completa (1920x1080 píxeles) y el estado por defecto será pantalla grande, de igual forma, se puede minimizar o reducir la ventana como tradicionalmente funciona en la mayoría de las aplicaciones. Es importante mencionar que, para mostrar un dato numérico en la GUI, se debe realizar un cambio en el tipo de variable, dado que la vibración y la temperatura podrían tener decimales se debe cambiar de tipo “float” a “DoubleVar”.

Además, se incluyen una serie de etiquetas donde se especifica el número de bomba y el parámetro a mostrar ya sea vibración o temperatura, al lado de ellas se indica en un recuadro de color blanco el parámetro proveniente de la medición de los sensores instalados.

Un espacio por debajo de las etiquetas anteriores, se instala otro rótulo el cual mientras no exista un valor que se detecte como alarma, tendrá color verde con la leyenda en blanco “Bomba N en buen estado” donde la N representa el número de máquina. Si, por el contrario, existiese alguna alerta, la etiqueta antes mencionada cambiará a color rojo y el texto incluirá el nombre del parámetro que causó de la falla (vibración o temperatura),, así como la fecha y hora en la que ocurrió.

Como parte de la sección de recomendaciones de un sistema experto, se incluye un cuadro de texto, donde cada vez que se detecte una falla el recuadro se llenará con una lista de acciones a realizar, así como las posibles causas que provocaron la alarma, si, por alguna razón, se detectan anomalías en vibración y, posterior a ello ,existen problemas con la temperatura (o viceversa), el cuadro se actualiza con las recomendaciones de ambas fallas, así como la inclusión del valor que detonó la alarma.

Finalmente, se incluye un botón debajo del cuadro anterior, el cual tiene como función eliminar la alerta y borrar el texto del recuadro, indicando que la bomba ya fue intervenida y el problema fue solucionado, si el sistema detecta que aún existen problemas se activará la alarma nuevamente, es importante aclarar que en ningún momento el sistema deja de monitorear y mostrar los valores enviados por el Arduino.

Debido a que existe la posibilidad de obtener datos significativos para activar la alarma y, luego, bajar a valores normales bajo, los cuales la indicación será de bomba en buen estado, se plantea un sistema de memorización, donde en el momento que se detecte una falla el sistema activará la alarma y la dejará en esa posición, aunque los valores se estabilicen en el rango normal de funcionamiento, ya que podrían pasarse por alto problemas en momentos específicos.

Dentro de este código, se establece que los tiempos de refresco y, por ende, de toma de datos serán cada segundo, de modo que este sea lo más en tiempo real posible sin mostrar valores en pantalla con variaciones tan rápidas que no las pueda captar el ojo humano o no sean reconocibles, de la misma manera se estipula un tiempo de guardado de datos cada 10 segundos a modo de no saturar la memoria del sistema, ambos valores pueden ser regulados según se desee.

A modo de mejorar la comprensión de las instrucciones antes mencionadas se muestra en la figura 6.25 el resultado de la programación de la GUI, en ella, se observa como en la Bomba 1 y 2 existen problemas, pues el sistema ha detectado una posible falla o avería, en concreto la primera de ellas estipula un defecto por temperatura mientras que la segunda se debe a vibración, la Bomba 3 se encuentra en buen estado, por lo que no requiere ninguna acción. En el recuadro de texto, se menciona entre asteriscos el valor que provocó la alerta, 103,37°C para la primera máquina y 4.0g para la segunda.

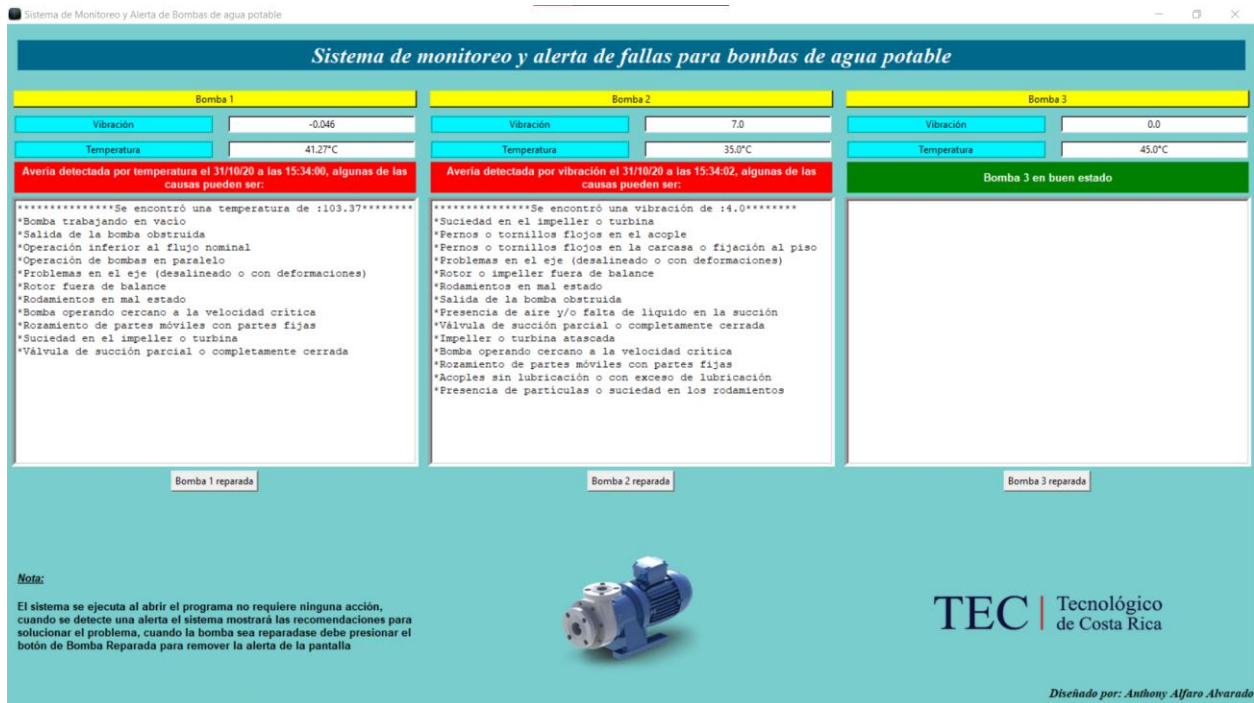


Figura 6.25. Interfaz gráfica del sistema experto. Fuente: Elaboración propia.

6.7. Determinación de las recomendaciones y acciones de mantenimiento

Debido a que, como se mencionó en anteriores ocasiones, no existe un historial bajo el cual se puedan detallar los procedimientos realizados en cada intervención se deberán establecer con base a normas, en concreto se hará uso del libro *Pump Handbook* de Igor Karassik, en el cual el capítulo 12 menciona las tablas 6.1 y 6.2 presentadas en este documento, bajo las cuales, tras analizar la última de estas y relacionarla con los parámetros a medir, se logran establecer recomendaciones adecuadas.

La forma en la que fueron programadas corresponde a un diccionario con etiquetas de llamado indicando el nombre de la variable, por ejemplo, si se realiza un llamado, mediante el nombre “Vibración”, el código devolverá todas las recomendaciones que estén relacionadas a este parámetro, la lista se muestra en la figura 6.26, en ella, también se incluyó la parte eléctrica, aunque no se pondrán a prueba, quedarán como base para futuras ampliaciones.

```

recomendaciones={}
recomendaciones['Vibración']=['Suciedad en el impeller o turbina',
                                'Pernos o tornillos flojos en el acople',
                                'Pernos o tornillos flojos en la carcasa o fijación al piso',
                                'Problemas en el eje (desalineado o con deformaciones)',
                                'Rotor o impeller fuera de balance',
                                'Rodamientos en mal estado',
                                'Salida de la bomba obstruida',
                                'Presencia de aire y/o falta de líquido en la succión',
                                'Válvula de succión parcial o completamente cerrada',
                                'Impeller o turbina atascada',
                                'Bomba operando cercano a la velocidad crítica',
                                'Rozamiento de partes móviles con partes fijas',
                                'Acoples sin lubricación o con exceso de lubricación',
                                'Presencia de partículas o suciedad en los rodamientos'
                                ]

recomendaciones['Temperatura']=['Bomba trabajando en vacío',
                                  'Salida de la bomba obstruida',
                                  'Operación inferior al flujo nominal',
                                  'Operación de bombas en paralelo',
                                  'Problemas en el eje (desalineado o con deformaciones)',
                                  'Rotor fuera de balance',
                                  'Rodamientos en mal estado',
                                  'Bomba operando cercano a la velocidad crítica',
                                  'Rozamiento de partes móviles con partes fijas',
                                  'Suciedad en el impeller o turbina',
                                  'Válvula de succión parcial o completamente cerrada'
                                  ]

recomendaciones['Eléctrico']=['Dirección inversa de rotación',
                                'Tamaño inadecuado del impeller o turbina',
                                'Rozamiento de partes móviles con partes fijas',
                                'Empaques instalados incorrectamente',
                                'Apriete demasiado fuerte (impide lubricación a partes móviles)',
                                'Pérdida de fase',
                                'Impeller o turbina atascada',
                                'Salida de la bomba obstruida',
                                'Válvula de succión parcial o completamente cerrada'
                                ]

```

Figura 6.26 Recomendaciones para alertas de falla. Fuente: Elaboración propia.

6.8. Comparación económica (Cuarto objetivo específico)

Respecto a la parte económica del proyecto, las intervenciones realizadas por el personal de mantenimiento se encuentran tercerizadas por la empresa RQL Ingeniería, mediante visitas para realizar mantenimiento preventivo de forma bimensual con un costo de ¢165 000 por cada visita, el contrato se realiza anualmente con un costo total de ¢825 000. Si en la visita se encuentran con una bomba dañada, el reporte se pasa al Departamento de Mantenimiento para intervenir el equipo.

Con la propuesta del cambio hacia un sistema experto, como el desarrollado en el presente documento, se pueden reducir drásticamente la cantidad de visitas hasta, incluso, llegar al punto donde solo se requieran cuando se presente una intervención alertado por el sistema computacional para el monitoreo y alerta de fallas, lo que supone un ahorro significativo en el costo de mantenimiento de los equipos instalados.

Por otra parte, con el sistema experto y la programación aplicada, se logra formar una base de datos de los parámetros medidos en cada bomba (vibración y temperatura), lo cual posee un gran valor ya que estos se almacenarán periódicamente en una computadora logrando transferirse, si fuese necesario a un servidor, los datos en la actualidad y en el futuro representan un gran costo económico si se desean adquirir y no muchos están dispuestos a compartirlos.

El costo total asociado al desarrollo y la instalación del sistema experto, aunado a un monitoreo de variables se especifica en la figura 6.27, en ella, se describen cada uno de los componentes necesarios para la puesta en marcha del mismo, como se mencionó anteriormente se hará uso de un Arduino UNO R3, el cual junto con su fuente de poder es el componente de mayor valor en el desglose de precios, los sensores involucrados serán de bajo costo con costo unitario de \$3.95 el más costoso.

Si bien es cierto que los sensores pueden cambiarse por otros de grado más industrial y con mejores prestaciones, el sistema no sufrirá modificaciones, se mantendrá el Arduino como el medio para la interpretación de las señales y el módulo ESP8266-01 como servidor web para el envío de datos a través de la red, hacia los clientes que posean el programa de monitoreo y alertas de fallas.

En relación con los costos asociados a los recursos computacionales, para el desarrollo e implementación del sistema, se tiene que todas las herramientas empleadas son de código libre (*Open Source*), cuyo valor de adquisición es nulo, tanto los módulos de la aplicación como Tkinter, hasta los usados en la programación de la red neuronal y la adquisición de datos desde un servidor (Numpy, Pandas, Requests y demás) son gratuitos.

El entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés *Integrated Development Environment* (IDE), usado para programar los comandos o instrucciones del programa también es de libre uso, sin ningún costo asociado, para este proyecto se utilizó Spyder, el cual es una IDE para el desarrollo del lenguaje Python, también para facilitar la escritura del código y evitar errores se usó Kite, que brinda recomendaciones para completar la línea de forma correcta, facilitando la programación.

Dado lo anterior, se puede concluir que el costo total de desarrollo e instalación del sistema experto es bastante asequible y más económico que el método actual, por lo que la viabilidad económica queda demostrada, además del beneficio de obtener la propia base de datos, la cual servirá para que el programa pueda continuar aprendiendo y mejorando con el tiempo.







Cart Items	Qty	Item Price	Item Total
 <p>Official Arduino UNO REV 3 Power Wall Adapter Power Supply Supply: - 9V 1A Change</p>	<input type="text" value="1"/> Remove	\$27.90	\$27.90
 <p>Piezo Vibration Sensor - Large</p>	<input type="text" value="3"/> Remove	\$3.95	\$11.85
 <p>TMP36 - Temperature Sensor</p>	<input type="text" value="3"/> Remove	\$1.95	\$5.85
 <p>ESP8266 Serial WIFI Module</p>	<input type="text" value="1"/> Remove	\$8.95	\$8.95
 <p>Ribbon Cable - 6 Wire (15ft)</p>	<input type="text" value="1"/> Remove	\$4.50	\$4.50
 <p>Acrylic Case for Arduino UNO</p>	<input type="text" value="1"/> Remove	\$2.49	\$2.49
Subtotal:			\$61.54
Sales Tax:			\$8.00
Grand Total:			\$69.54

Figura 6.27. Costo total de dispositivos para el Sistema de monitoreo. Fuente: CRCibernética, 2020.

6.9. Beneficios de la aplicación de un sistema experto

Como se ha mencionado a lo largo del documento, una de las secciones más representativas y tangibles de la incorporación de sistemas inteligentes asociadas al mantenimiento es la falta de intervención en la predicción y análisis de datos del personal, evitando así problemas de subjetividad en la búsqueda de tendencias y valores umbrales para determinar alertas y el momento adecuado para realizar intervenciones en el equipo.

En relación con el aspecto económico, se tiene que las intervenciones de una empresa tercerizada se darán siempre y cuando el sistema anuncie que requiere una visita o aplicar métodos preventivos para evitar una falla inminente, lo que supone un ahorro de alrededor de \$1350 anuales, debido a que se podría prescindir del contrato, el costo de instalación de los componentes ronda los \$70, si se desea cambiar los sensores en un futuro se podrá realizar, sin embargo, para efectos del presente prototipo es suficiente con los mostrados en la figura 6.27.

Mediante el almacenado de los datos, es posible aplicar distintos métodos para el análisis de comportamiento (ya sean sistemas expertos, redes neuronales, *Machine Learning*), con el fin de lograr pronosticar el deterioro o comportamiento de una máquina, esto es de suma importancia en comparación con los datos actuales de funcionamiento los cuales son nulos y no le permiten al departamento tomar decisiones respecto a la vida útil y desempeño de las bombas.

Un beneficio que no está relacionado al departamento de mantenimiento es la utilización de los datos y el sistema experto como medio de ejemplo en los cursos que se brindan en el área administrativa, como se mencionó en apartados anteriores respecto al curso de Métodos cuantitativos para la toma de decisiones, así se tendrá un conjunto de datos reales del comportamiento y tendencia del equipo respecto a fallas.

Debido a que en esta etapa se cuenta con un prototipado y una fase de investigación, aún no se puede comentar respecto al prolongamiento de la vida útil del bien, muchos expertos en el tema recomiendan una puesta en marcha del equipo de algunos meses, con el fin de almacenar los datos y, posterior a ello, el sistema pueda aprender y adecuarse a estas tendencias. No obstante, se tomarán datos de máquinas similares como primer acercamiento y aprendizaje para alertar de fallas en las bombas.

Una vez que los datos han sido almacenados y se tiene un conjunto suficiente para representar la mayor parte del comportamiento de la máquina, al aplicar métodos de predicción de fallas, se logra extender el periodo de funcionamiento de los equipos al máximo, dado que cada avería supone un riesgo de no volver a poner en marcha la máquina, debido a averías en cadena, por no atender la primera que se manifestó o por no contar con sistemas que permitan monitorear el estado de los parámetros de funcionamiento, logrando garantizar el suministro de agua en todo momento dentro del campus.

7. Conclusiones

El desarrollo de un sistema experto requiere de un historial adecuado para su puesta en marcha, aunque se puede realizar mediante la aplicación de normas como la ISO 17359 y la ISO 10816 o mediante recomendaciones de libros especializados en las máquinas en estudio.

La programación del sistema se puede realizar mediante métodos inteligentes, como *Machine Learning* o Redes Neuronales para mejorar la capacidad de análisis de datos y reducir lo más posible la intervención humana en el procesamiento y estudio de comportamientos y tendencias.

La utilización de normas como ISO 17359 resulta de gran apoyo en la determinación de parámetros a considerar, esto si no se cuenta con ningún equipo instalado y para evitar una inadecuada inversión de fondos en datos no significativos bajo el contexto de detección de fallas.

El uso de la red neuronal supone una gran ventaja en su implementación, cuando no se conoce con exactitud el comportamiento y tendencia de los datos, ya que el mismo sistema se encarga de buscarlos y formular áreas mediante el agrupamiento de los datos y así determinar cuando existe una alerta o no.

El costo asociado a la implementación y desarrollo del sistema experto es mucho menor al modelo de gestión de mantenimiento actual (\$69.54 totales respecto a los casi \$1350 anuales), debido a la simplicidad y tipo de dispositivos empleados, y a los componentes y módulos del programa por ser código abierto o libre, el sistema una vez establecido con los parámetros adecuados y con una inteligencia apta, no harán necesarias las visitas recurrentes, solo se harán efectivas cuando el sistema alerte sobre una posible falla.

8. Recomendaciones

Cuando se realice la instalación, se debe colocar el Arduino en una posición alejada de un posible rocío o lugar donde pueda mojarse, además de evitar golpes serios que puedan comprometer la integridad de este, se incluirá una carcasa para evitar daños físicos ligeros y cabe resaltar que esta no es impermeable. Asimismo, los sensores de temperatura son susceptibles al agua, por lo que deberán colocarse en la parte superior del motor mediante algún dispositivo de sujeción o material adhesivo, lo suficientemente firme como para evitar mediciones erróneas, ya que deberá tener en la medida de lo posible contacto directo con la superficie a medir.

Con respecto a los sensores de vibración, estos deberán colocarse en las posiciones especificadas por la norma ISO 10816, con la salvedad de que la parte de los contactos deberá quedar al aire (voladizo), para permitir que las fuerzas de oscilación doblen el material piezoeléctrico del sensor, el otro extremo se debe fijar lo más posible a la sección de la bomba, mediante materiales adhesivos que no dañen la integridad de este, pero que logren sujetarse sin permitir movimientos.

Los cables de señal provenientes de los sensores deberán colocarse lo más distanciado posible de los conductores de alimentación de los motores, ya que podrían provocar interferencias y los datos que recopila el Arduino serán erróneos conduciendo a un mal funcionamiento del sistema.

Se debe coordinar con el Departamento de Tecnologías de la Información (DATIC) para la instalación de un repetidor o extensor de rango de red para que la potencia de esta sea adecuada en la estación de bombeo, por parte del guardado de los datos en la computadora, solo se realiza mientras el programa esté en ejecución, por lo que se recomienda que esté en abierto la mayor cantidad de tiempo posible, los recursos necesarios para su ejecución no comprometen al equipo ni ralentiza otras tareas, por lo que puede colocarse en segundo plano sin problemas.

9. Bibliografía

- Adafruit. (2020). *TMP36 Temperature Sensor*. Adafruit Learning System.
<https://learn.adafruit.com/tmp36-temperature-sensor>.
- Alkadhim, Saif Aldeen. (2020). Three-phase Induction Motor: Types and Structure.
10.13140/RG.2.2.11796.12162.
https://www.researchgate.net/publication/342819630_Three-phase_Induction_Motor_Types_and_Structure?channel=doiylinkId=5f075c68a6fdcc4ca459d543yshowFulltext=true
- Amat Rodrigo, J. (2017). Máquinas de Vector Soporte (Support Vector Machines, SVMs).
Cienciadedatos.net.
https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines.
- Arduino. (2018). *What is Arduino?* Arduino.cc. <https://www.arduino.cc/en/Guide/Introduction>.
- Badaró, S., Ibáñez, L., y Agüero, M. (2014). *Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones*. Universidad de Palermo.
https://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf.
- Basco, A., Beliz, G., Coats, D., y Garnero, P. (2018). *Industria 4.0: Fabricando el Futuro*
<https://publications.iadb.org/publications/spanish/document/Industria-40-Fabricando-el-Futuro.pdf>.
- Beekman, G. (1999). *Introducción a la computación*. Addison-Wesley.
- Bloch, H. (2011). *Pump wisdom* (1st ed.). Wiley.

- CFIA. (2017). *Código de Instalaciones Hidráulicas y Sanitarias en Edificaciones* [PDF]. San José, Costa Rica: Colegio Federado de Ingenieros y Arquitectos. http://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTCynValor1=1ynValor2=83561ynValor3=107558ystrTipM=TC
- Chen, D. (2015). *Wireless Communication with ESP8266 | How to Make Something that Makes (almost) Anything*. MIT. <http://fab.cba.mit.edu/classes/865.15/people/dan.chen/esp8266/>.
- Cordero, M. (2013). ¿Cómo Costa Rica creó una universidad tecnológica? <https://www.elfinancierocr.com/tecnologia/como-costa-rica-creo-una-universidad-tecnologica/TOMUEI576NB5RPG4E3ZZTACYHM/story/>
- CRCibernética. (2020). CRCibernética Open Source Hardware. Crcibernetica.com. Retrieved 16 November 2020, from <https://www.crcibernetica.com/>.
- del Valle Hernández, L. (2017). *Guía para configurar un ESP-01, el módulo WiFi basado en ESP8266*. <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>.
- Departamento de Salud y Servicios Humanos. (2016). Acerca de los parásitos. <https://www.cdc.gov/parasites/es/about.html>
- EVTECH. (2020). Industria y mantenimiento 4.0 <https://evtech.cl/industria-y-mantenimiento-4-0/>.
- Fang, Xianjin y Yu, Fangchao y Yang, Gaoming y Qu, Youyang. (2019). Regression Analysis with Differential Privacy Preserving. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2940714.
- Garro, A. (2014). Bombas Hidráulicas [PDF].

Ghazali, Z., y Ridhuan, M. (2018). Maintenance performance improvement analysis using Fuzzy Delphi method A case of an international lube [PDF] (1st ed.). Malaysia: University of Petronas.

Gorini, M. (2019). *¿Cuál es la diferencia entre el machine learning y el deep learning?* Bismart. <https://blog.bismart.com/es/diferencia-machine-learning-deep-learning>.

Graney, B., y Starry, K. (2012). *Rolling Element Bearing Analysis*. Maintenance.org. https://www.maintenance.org/fileSendAction/fcType/0/fcOid/399590942964610394/filePointer/399590942964867002/fodoid/399590942964867000/Rolling_Element_Bearing_Analysis_graney.pdf.

IBM Educational Group. (2020). *What is Deep Learning?* IBM. <https://www.ibm.com/cloud/learn/deep-learning>.

IBM Educational Group. (2020). *What is Machine Learning?* IBM. <https://www.ibm.com/cloud/learn/machine-learning>.

International Organization for Standardization. (2009). *Mechanical vibration — Evaluation of machine vibration by measurements on non-rotating parts — Part 7: Rotodynamic pumps for industrial applications, including measurements on rotating shafts* (ISO/DIS Standard No. 10816-7). <https://www.iso.org/standard/41726.html>

International Organization for Standardization. (2018). *Condition monitoring and diagnostics of machines — General guidelines* (ISO/DIS Standard No. 17359). <https://www.iso.org/standard/71194.html>

- JavaTpoint. (2018). *Expert Systems in Artificial Intelligence - Javatpoint*. www.javatpoint.com.
<https://www.javatpoint.com/expert-systems-in-artificial-intelligence>.
- Jacuzzi. (2020). Bombas Centrífugas de Hierro Fundido Serie G - Jacuzzi. Jacuzzi.
<https://www.jacuzzi.cl/produto-detalhes/bombas-centrifugas-de-hierro-fundido-serie-g/>.
- Kampakis, S. (2018). *What deep learning is and isn't? - The Data Scientist*. The Data Scientist.
<https://thedata scientist.com/what-deep-learning-is-and-isnt/>.
- Karassik, I. (2001). Pump handbook (3rd ed.). McGraw-Hill.
- Kavlakoglu, E. (2020). *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?* IBM. <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>.
- Koutoudjian, J., s.f. *Estaciones De Bombeo*. 1st ed. [PDF] Buenos Aires, Argentina: Instituto de Ingeniería Sanitaria y Ambiental. http://www.fi.uba.ar/archivos/Estaciones_de_Bombeo.pdf
- La Nucía TV. (2011). Estación Bombeo de Agua Potable, puesta en marcha. [Video]. YouTube: Ayuntamiento de La Nucía, España.
- Londoño-Franco, Luis Fernando, Londoño-Muñoz, Paula Tatiana, y Muñoz-García, Fabián Gerardo. (2016). Los Riesgos de los metales pesados en la salud humana y animal. *Biotecnología en el Sector Agropecuario y Agroindustrial*, 14(2), 145-153. [https://dx.doi.org/10.18684/BSAA\(14\)145-153](https://dx.doi.org/10.18684/BSAA(14)145-153)
- Mahmoud Afifi (2020). kNN classifier. MATLAB Central File Exchange.
<https://www.mathworks.com/matlabcentral/fileexchange/63621-knn-classifier>
- Mendoza, G. F. (2007). Bombas centrífugas. <https://ebookcentral.proquest.com>

Microsoft. (2020). *Selección de un algoritmo de Machine Learning - Azure Machine Learning*.

Docs.microsoft.com. <https://docs.microsoft.com/es-es/azure/machine-learning/how-to-select-algorithms>.

MissingLink. (2020). *7 Types of Activation Functions in Neural Networks: How to Choose?*

<https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/>.

Mobius Institute. (2020). *Vibration Analysis Calculators, simulations, y Severity Charts - Mobius*.

Mobius Institute. <https://www.mobiusinstitute.com/calculators-simulations-severity-charts/>.

Molden, D. (2008). Resumen. agua para la alimentación, agua para la vida.

<https://ebookcentral.proquest.com>

Monge, M. (2017). Qué es y cómo se calcula el NPSHdisp de una bomba centrífuga. iAgua.

<https://www.iagua.es/blogs/miguel-angel-monge-redondo/que-es-y-como-se-calcula-npshdisp-bomba-centrifuga>.

Mora Alvarado, D., Orozco Gutiérrez, J., Solís Castro, Y., Rivera Navarro, P., Cambronero

Bolaños, D., Zúñiga Zúñiga, L., y García Aguilar, J. (2018). Índice de Riesgo de la Calidad del Agua para Consumo Humano en Costa Rica (IRCACH). *Tecnología En Marcha*, (31-3), 3-14.

https://www.researchgate.net/publication/328240674_Indice_de_Riesgo_de_la_Calidad_del_Agua_para_Consumo_Humano_en_Costa_Rica_IRCACH

Mora Pérez, K. (2020). Tres universidades públicas costarricenses destacan en la lista de las

mejores universidades del mundo. <https://www.tec.ac.cr/hoyeneltec/2020/06/15/tres-universidades-publicas-costarricenses-destacan-lista-mejores-universidades-mundo>

- NASA. (2020). *Prognostics Center of Excellence - Data Repository*. Ti.arc.nasa.gov.
<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>.
- National Electrical Manufacturers Association (NEMA), 2010, Standard NEMA MG-1-2010: Motors and generators.
- Ng, A., Katanforoosh, K., y Mourri, B. (2019). *Neural Networks and Deep Learning*. Coursera.
<https://www.coursera.org/learn/neural-networks-deep-learning>.
- Nielsen, M. (2019). *Neural Networks and Deep Learning*.
<http://neuralnetworksanddeeplearning.com/>
- NumPy. (2020). *What is NumPy? — NumPy v1.20.dev0 Manual*. Numpy.org.
<https://numpy.org/devdocs/user/whatisnumpy.html>.
- OMS. (2006). *Guías para la calidad del agua potable (3rd ed.)*. Ginebra: Organización Mundial de la Salud.
- OMS. (2019). *Saneamiento*. <https://www.who.int/es/news-room/fact-sheets/detail/sanitation>
- Oppy, G., y Dowe, D. (2020). *The Turing Test (Stanford Encyclopedia of Philosophy)*. Plato.stanford.edu. <https://plato.stanford.edu/entries/turing-test/>.
- Palma, M. J. T., y Marín, M. R. (2008). *Inteligencia artificial: Métodos, técnicas y aplicaciones*.
<https://ebookcentral.proquest.com>
- Pandas. (2020). *pandas - Python Data Analysis Library*. Pandas.pydata.org.
<https://pandas.pydata.org/about/>.
- Pastor, C. (2019). *El mantenimiento como herramienta para conseguir infraestructura de alta calidad y durabilidad*. Banco Interamericano de Desarrollo (BID).

Pruftechnik. (2011). La vibración confirma la Norma. pruftechnik.
<https://pruftechnik.wordpress.com/2011/10/13/la-vibracion-confirma-la-norma/>.

Pruftechnik. (2016). Conceptos del análisis de vibraciones mecánicas en máquinas rotativas [PDF]. www.blog.pruftechnik.com

Rouse, M. (2016). *What is expert system?* SearchEnterpriseAI.
<https://searchenterpriseai.techtarget.com/definition/expert-system>.

Sparkfun. (2020). *Piezo Vibration Sensor - Large - SEN-09196 - SparkFun Electronics*.
Sparkfun.com. <https://www.sparkfun.com/products/9196>.

Tecnológico de Costa Rica. (2020). Reseña del TEC. <https://www.tec.ac.cr/resena-tec>

Thermo-Watt Industria. (2020). Mantenimiento industrial de la Industria 4.0. <https://www.thermo-watt.com/thermo-watt-empresa/blog-actualidad/86-mantenimiento-industrial-de-la-industria-4-0>.

Trocel, D. (2019). Análisis de vibraciones en bombas centrífugas. Power-MI. <https://power-mi.com/es/content/an%C3%A1lisis-de-vibraciones-en-bombas-centr%C3%ADfugas>.

Turban, E. (2007). *Decision Support Systems and Intelligent Systems* (7th ed.). Prentice-Hall.

Tutorials Point India Limited. (2020). *Artificial Intelligence - Expert Systems - Tutorialspoint*.
Tutorialspoint.com.
https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_expert_systems.htm.

US Electric. (2020). Voltage Frequency Variation. Acim.nidec.com.
<https://acim.nidec.com/motors/usmotors/techdocs/profacts/voltage-frequency-variation>.

US Motors. (2020). *Frequently Asked Questions*. Acim.nidec.com. <https://acim.nidec.com/fr-fr/motors/usmotors/service-support/faq>.

WeatherSpark. (2020). *Clima promedio en Cartago, Costa Rica, durante todo el año - Weather Spark*. Es.weatherspark.com. [https://es.weatherspark.com/y/16154/Clima-promedio-en-Cartago-Costa-Rica-durante-todo-el-a %C3 %B1o](https://es.weatherspark.com/y/16154/Clima-promedio-en-Cartago-Costa-Rica-durante-todo-el-a%C3%B1o).

Zwass, V. (2016). *Expert system / computer science*. Encyclopedia Britannica <https://www.britannica.com/technology/expert-system>.

10. Apéndices

Commands	Description	Set/Execute	Parameters
AT+RST	restart the module	-	-
AT+CWMODE	wifi mode	AT+CWMODE=<mode>	1= Sta, 2= AP, 3=both
AT+CWJAP	join the AP	AT+ CWJAP =<ssid>,< pwd >	ssid = ssid, pwd = wifi password
AT+CWLAP	list the AP	AT+CWLAP	
AT+CWQAP	quit the AP	AT+CWQAP	
AT+ CWSAP	set the parameters of AP	AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn>	ssid, pwd, chl = channel, ecn = encryption
AT+ CIPSTATUS	get the connection status	AT+ CIPSTATUS	
AT+CIPSTART	set up TCP or UDP connection	1)single connection (+CIPMUX=0) AT+CIPSTART= <type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	id = 0-4, type = TCP/UDP, addr = IP address, port= port
AT+CIPSEND	send data	1)single connection(+CIPMUX=0) AT+CIPSEND=<length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND= <id>,<length>	
AT+CIPCLOSE	close TCP or UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	
AT+CIFSR	Get IP address	AT+CIFSR	
AT+ CIPMUX	set mutiple connection	AT+ CIPMUX=<mode>	0 for single connection 1 for mutiple connection
AT+ CIPSERVER	set as server	AT+ CIPSERVER= <mode>[,<port>]	mode 0 to close server mode, mode 1 to open; port = port
+IPD	received data		

Figura 10.1 Comandos AT. Fuente: Chen, D, 2015.



Figura 10.2 Datos de placa del motor Baldor.



Figura 10.3 Datos de placa de motor US Electric

Table A.1 — Zone limits for vibration of non-rotating parts of rotodynamic pumps with power above 1 kW, applicable for impellers with number of blades $z_i \geq 3$

Zone	Description (see 5.2 for details of zone definitions)	Vibration velocity limit r.m.s. value mm/s			
		Category ^a I		Category ^a II	
		≤ 200 kW	> 200 kW	≤ 200 kW	> 200 kW
A	Newly commissioned machines in preferred operating range	2,5	3,5	3,2	4,2
B	Unrestricted long-term operation in allowable operating range	4,0	5,0	5,1	6,1
C	Limited operation	6,6	7,6	8,5	9,5
D	Risk of damage	> 6,6	> 7,6	> 8,5	> 9,5
Maximum ALARM limit (= 1,25 times the upper limit of zone B) ^b		5,0	6,3	6,4	7,6
Maximum TRIP limit (= 1,25 times the upper limit of zone C) ^b		8,3	9,5	10,6	11,9
<i>In situ</i> acceptance test	Preferred operating range	2,5	3,5	3,2	4,2
	Allowable operating range	3,4	4,4	4,2	5,2
Factory acceptance test	Preferred operating range	3,3	4,3	4,2	5,2
	Allowable operating range	4,0	5,0	5,1	6,1
For all acceptance tests in the preferred operating range (see 3.4), each of the filtered values ^c for rotational frequency (f_n) and blade-passing frequency ($f_n \cdot z_i$) should be		≤ 2	≤ 2	≤ 3	≤ 3
^a For definition, see 5.1. ^b Recommended values. The vibration magnitudes should be above these limits for about 10 s before an ALARM or TRIP is released to avoid false alarms and trips. ^c For acceptance tests in the allowable but outside the preferred operating range, the filtered values for rotational frequency (f_n) and blade-passing frequency ($f_n \cdot z_i$) may be expected to be higher (1,3 times) than the values for the preferred operating range.					

Figura 10.4 Zonas de alerta en mm/s para vibración. Fuente: ISO 10816, 2009



Figura 10.5 Bombas instaladas (se muestran 2 de 3).



Figura 10.6 Relé inteligente para el control de las bombas.

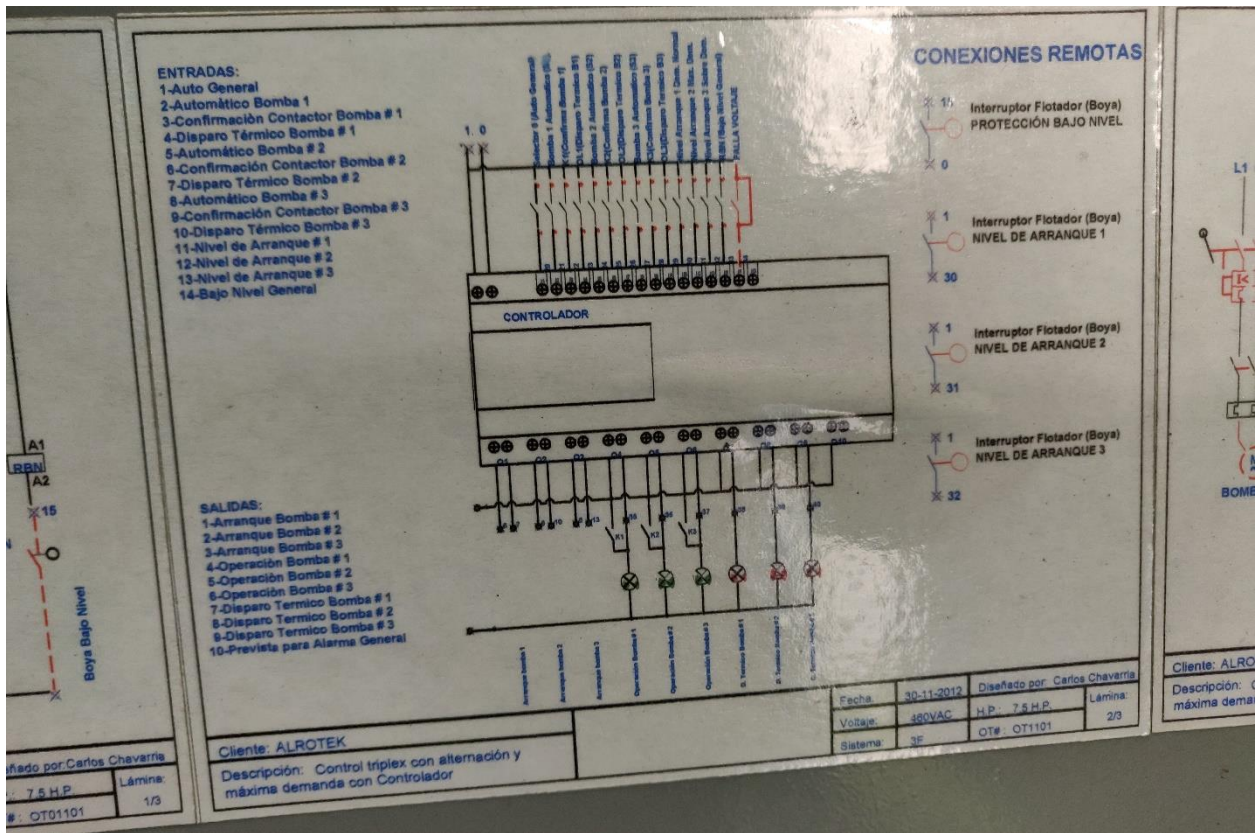


Figura 10.7 Conexión del Relé inteligente.