UNIVERSIDAD DE CIENCIAS
Y ARTES DE CHIAPAS

INSTITUTO DE INVESTIGACIÓN E
INNOVACIÓN EN ENERGÍAS RENOVABLES

T E S I S

RECONOCIMIENTO DE FALLAS EN RODAMIENTOS MEDIANTE REDES NEURONALES

PARA OBTENER EL TÍTULO DE

LICENCIADA EN INGENIERÍA EN ENERGÍAS RENOVABLES

PRESENTA

KAREN ALONDRA GÓMEZ REYES

DIRECTOR DE TÉSIS

DR. JUAN LUIS PÉREZ RUIZ

Tuxtla Gutiérrez, Chiapas

12 de agosto de 2025

Universidad de Ciencias y Artes de Chiapas
Dirección de Servicios Escolares
Departamento de Certificación Escolar
Autorización de impresión

SISTEMA DE GESTIÓN DE CALIDAD

Lugar: Tuxtla Gutiérrez, Chiapas
Fecha: 12 de Agosto de 2025

C. Karen Alondra Gómez Reyes

Pasante del Programa Educativo de: Ingeniería en Energías Renovables

Realizado el análisis y revisión correspondiente a su trabajo recepcional denominado:
Reconocimiento de Fallas en Rodamientos mediante Redes Neuronales

En la modalidad de: Tesis Profesional

Nos permitimos hacer de su conocimiento que esta Comisión Revisora considera que dicho documento reúne los requisitos y méritos necesarios para que proceda a la impresión correspondiente, y de esta manera se encuentre en condiciones de proceder con el trámite que le permita sustentar su Examen Profesional.

ATENTAMENTE

Revisores	**Firmas:**
Mtra. María Fernanda Moguel Esteban	
Dr. Carlos Alonso Meza Avendaño	
Dr. Juan Luis Pérez Ruiz	

Ccp. Expediente

Dedicatoria

Este trabajo se lo dedico, con todo mi corazón, a mi madre. A la mujer humilde, honrada y trabajadora que, desde el principio, vio en mí lo que yo aún no podía ver. La que me protegió de mis dudas, me sostuvo en medio de mis temores y creyó en mis posibilidades cuando yo solo veía límites. A ti, que eres mi compañera incansable, mi consejera más sabia y la luz que me guía. Tú, que me inculcaste valores, fortaleza y el hábito de la perseverancia. Cada logro alcanzado es un reflejo de tu amor, tu fe y tu entrega. Entré con miedo y dudas; hoy salgo con esperanza, valentía y sueños renovados.

Todo esto, mamá, es también tuyo.

A ti te lo dedico.

Karen Alondra Gómez Reyes

Agradecimientos

Expreso mi más profundo agradecimiento al Instituto de Innovación e Investigación en Energías Renovables de la Universidad de Ciencias y Artes de Chiapas por haberme abierto sus puertas en un momento de gran incertidumbre personal. Gracias a la calidad educativa y al compromiso de sus docentes, he logrado culminar una carrera que alguna vez creí imposible siquiera comenzar.

Agradezco sinceramente a la *Dra. Aracely López Grijalva*, al *Dr. Heber Vilchis Bravo*, a la *Dra. Laura Elena Verea Valladares* y al *Dr. Antonio Verde Añorve*, por su guía, conocimiento y confianza. Sus enseñanzas han sido invaluables, no solo para mi formación profesional, sino también en lo personal.

Mi más sincero agradecimiento al *Dr. Juan Luis Pérez Ruiz*, director de esta tesis, por brindarme la oportunidad de explorar, cuestionar y aprender a lo largo de este trabajo. Su confianza al asignarme este tema fue clave para mi crecimiento académico y profesional.

Asimismo, extiendo un especial agradecimiento al *Arq. Ing. Joel Sánchez García* por su generosidad al compartir su experiencia en el área mecánica y por su apoyo constante durante el desarrollo de este proyecto.

Finalmente, agradezco con todo mi corazón a mi conejita Blanquita, quien fue mi consuelo y refugio en los días más difíciles, así como a mi padre y a mi hermano, cuyo amor y apoyo incondicional fueron mi mayor ejemplo de fuerza y resiliencia a lo largo de esta etapa.

Resumen

El presente trabajo se enfoca en la detección de fallas en rodamientos mediante redes neuronales LSTM, basadas en el conjunto de datos de la *Universidad Case Western Reserve*. Si bien el conjunto de datos proporciona una variedad adecuada de tipos de fallas, presenta desafíos como longitudes de señal variables y ruido significativo en ciertas series, lo que dificulta la clasificación sin un preprocesamiento adecuado.

Se observó que, en muchos casos, usar solo características del dominio temporal resulta insuficiente. La incorporación de información en el dominio de la frecuencia, combinada con técnicas de filtrado y segmentación de la señal, mejoró significativamente el rendimiento del modelo. En particular, la segmentación permitió una mejor caracterización local de las señales y aumentó la cantidad de datos útiles para el entrenamiento del modelo.

Finalmente, el ajuste cuidadoso de los hiperparámetros del modelo y la preparación adecuada de los datos resultaron esenciales para lograr una clasificación precisa de fallas, incluso en condiciones de ruido.

Palabras clave:

Diagnóstico de fallas en rodamiento, Maquinaria rotatoria, Procesamiento de señales, Análisis de señales en tiempo-frecuencia, Extracción de características, Permutación de la importancia de las características, Deep Learning, LSTM, RNN.

Abstract

This work focuses on fault detection in bearings using LSTM neural networks, based on the *Case Western Reserve University* Bearing Dataset. While the dataset provides a suitable variety of fault types, it poses challenges such as variable signal lengths and significant noise in certain series, which complicates classification without proper preprocessing.

It was found that using only time-domain features is insufficient in many cases. The inclusion of frequency-domain information, along with filtering and signal segmentation techniques, significantly improved model performance. In particular, segmentation enabled better local characterization of signals and increased the amount of useful training data.

Finally, careful tuning of model hyperparameters and appropriate data preparation proved essential to achieve accurate fault classification, even under noisy conditions.

Keywords:

Bearing Fault Diagnosis, Rotating Machinery, Signal Processing, Time-Frequency Signal Analysis, Feature Extraction, Permutation Feature Importance, Deep Learning, Long Short Memory, Recurrent Neural Net.

Tabla de contenidos

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

"Si alguien es capaz de diseñar una máquina de ajedrez exitosa, parecería haber penetrado hasta el núcleo del esfuerzo intelectual humano", 1950

La inteligencia artificial se define como *"Aquellos sistemas que muestran un comportamiento inteligente, analizando su entorno y tomando acciones, con cierto grado de autonomía para lograr objetivos específicos"* de acuerdo con el Grupo de Expertos de Alto Nivel en Inteligencia Artificial (AI HLEG) de la Comisión Europea (EC); otros autores catalogan a la inteligencia artificial como *"la imitación de la inteligencia inherente de los humanos por parte de las computadoras"*.

En la actualidad es difícil definir con exactitud el término de inteligencia artificial, al ser una simulación de algo que no se ha acabado de comprender: la inteligencia del ser humano. Si bien la sociedad contemporánea se ha encargado de estudiar el cerebro y su relación con la inteligencia, existe un sesgo que deja incompleto el rompecabezas sobre qué es exactamente la inteligencia humana. Hasta que dicho sesgo se termine de completar, es posible hablar de imitaciones exactas y artificiales de la inteligencia humana.

En 1965, el matemático ruso Alexander Kronrod enunció que *"el ajedrez es la clave para comprender la inteligencia humana"*. Por este motivo la gente quedo perpleja cuando en 1997, Garry Kasparov fue derrotado por Deep Blue, la computadora de ajedrez de

IBM.

Esta victoria dio origen a nuevas expectativas para el nuevo mileno, previendo que la inteligencia desarrollada por las máquinas supere a los humanos en todo tipo de actividades que consideramos más fáciles que el ajedrez. Sin embargo, entrando a los años 2000, esto no ocurrió de la forma que se esperaba.

Durante mucho tiempo el ajedrez fue considerado un juego extremadamente avanzado, pero años de investigaciones posteriores demostraron que algo tan aparentemente simple como reconocer un gato en una fotografía es mucho más complejo que el ganar en ajedrez. Este fenómeno se conoce como la *paradoja de Moravec*: ciertas cosas que son muy difíciles para los humanos como el ajedrez o el cálculo avanzado son fáciles para las computadoras, pero tareas muy simples para los humanos, como el percibir objetos o usar las habilidades motoras para ciertas tareas domésticas, resultan difíciles para las computadoras.

1.1. Antecedentes de la Inteligencia Artificial

En 1950 Alan Turing, matemático, lógico y criptógrafo británico, quien exploraba la posibilidad matemática de la inteligencia artificial, planteando el siguiente dilema en su trabajo "Computing Machinery and Intelligence": "Si los humanos utilizan la información disponible además de la razón para resolver problemas y tomar decisiones, entonces ¿Por qué las máquinas no podrían hacer lo mismo?".

Sin embargo, lo que detuvo a Turing de obtener algún avance relevante fueron las propias máquinas, las cuales para principios de la década de los 50 carecían de un importante requisito: almacenar comandos y posteriormente ejecutarlos.

El desarrollo de la inteligencia artificial tiene sus raíces en la antiguas fábulas griegas, donde ya se contemplaba la idea de crear vida artificial. No obstante, para los fines del presente trabajo, se abordará su evolución a partir del surgimiento de la computación

analógica. A partir de este punto, se propone una clasificación en dos fases, como se describe a continuación:

- Establecimiento de las bases de la computación
- Establecimiento de las bases teóricas de la IA

1.1.1. Establecimiento de las bases teóricas de la computación

Charles Babbage diseñó la máquina analítica en 1834, cuya finalidad era la de construir tablas matemáticas y astronómicas. Más tarde le daría el nombre de *"Máquina Diferencial"* a su invento, ya que el principio en el que se basaba era el *"Método de las Diferencias"*, que permite calcular tablas matemáticas sumando las diferencias entre los términos de una serie. Este método le permitía a Babbage simplificar el cálculo de series complejas, sustituyendo multiplicaciones difíciles por numerosas sumas sencillas pero monótonas.

Para el año de 1841 colaboraría con Ada Lovelace, inicialmente para la traducción de un artículo sobre la máquina analítica perteneciente al científico italiano Luigi Federico Menabrea al inglés, durante la traducción documentó su propio análisis sobre la máquina de Babbage y anexó la traducción del artículo, dotando así de mayor repercusión al invento.

En ninguno de los escritos de Babbage consideró las posibilidades a explotar de su nuevo invento en ningún campo diferente a las matemáticas. Por otro lado, Lovelace comprendió que la máquina analítica puede aplicarse a cualquier proceso que implicará datos, de allí se concibe la *"ciencia de las operaciones"*, una rama distinta de las matemáticas, a lo que actualmente le conocemos como informática.

La aportación de Ada Lovelace tomó importancia con el pasar de las décadas,

deduciendo y previendo los límites de las computadoras más allá de los cálculos matemáticos, prediciendo que en el futuro una máquina podría componer música y hacer gráficos. Charles creó la máquina y Ada Lovelace creó las reglas del juego.

Hasta la Segunda Guerra Mundial se empezó a invertir en el desarrollo y optimización de la informática para evitar la dependencia de los observadores humanos. Asimismo, otra razón que encabezó este movimiento fue el descifrar el sistema de comunicación secreto de los nazis, conocido como Enigma.

Francia se rindió ante Alemania en 1940, ocasionando que el Reino Unido se quede sin aliados en Europa. Temiendo la eventual derrota de su nación, fue fundamental interceptar y descifrar las comunicaciones alemanas para saber la posición de los submarinos y desviar los barcos que suministraban recursos armamentísticos, alimenticios y energéticos al Reino Unido. Sin embargo, estos submarinos se comunicaban con una máquina llamada *"Enigma"*.

Alan Turing diseñó una máquina, que más tarde nombraría como *"Bombe"*, la cual era una réplica de 36 máquinas Enigma con sus rotores y clavijeros. A esta máquina se le introducían manualmente las configuraciones, dichas configuraciones descartaba la opción si encontraba letras repetidas.

De forma simultánea se realizaron operaciones militares para hacerse de las configuraciones de la máquina Enigma, si bien no se obtuvo más que fragmentos, ya que los alemanes tenían como prioridad destruir la máquina en caso de ser atacados por el enemigo, Alan Turing fue capaz de deducir el resto de las configuraciones. Gracias a la participación de Turing, el Reino Unido pudo sostenerse durante el conflicto.

Luego de la derrota de Enigma por parte de Bombe, los alemanes idearon una máquina cuyo mecanismo es superior en comparación a Enigma, esta nueva máquina se llamaría más tarde *"Lorenz"* (conocido entre los británicos como *"Tunny"*). Tommy

Flowers, ingeniero británico, pasaría los siguientes 11 meses en idear y construir la primera computadora electrónica de la historia: *"Colossus"*, la cual entraría en funciones en 1943.

John von Neumann estaba particularmente interesado sobre el cerebro humano, ya que veía al sistema nervioso central como un centro de administración de información mediante un lenguaje que desconocíamos. Así que cuando los primeros computadores se desarrollaron, von Neumann notó la relación entre el funcionamiento de los computadores y el cerebro humano.

El ENIAC (acrónimo de *Electronic Numerical Integrator and Computer*), creada en 1943 en la Universidad de Pensilvania, tenía como propósito principal realizar cálculos numéricos para uso militar. Sus aplicaciones iban dirigidos al campo de la balística y al movimiento de los proyectiles. No obstante, el problema que se enfrentaba el ENIAC era que cada vez que se realizaba un cálculo había que rehacer su diseño electrónico original, es decir, que se debía modificar las conexiones entre sus válvulas, enchufar y desenchufar una vasta cantidad de cables.

Los computadores modernos utilizan la arquitectura von Neumann que usa una memoria para almacenar instrucciones y datos, esta arquitectura permite leer programas localizados en la memoria y escribir instrucciones. El encargado de ejecutar las instrucciones es la CPU mediante un ciclo de instrucciones.

Hoy no necesitamos rehacer el cableado para ejecutar distintos programas gracias a la contribución de von Neumann, a él se le atribuye separar los componentes físicos (dicho de otra forma, el Hardware) de los programas que dictaran las instrucciones (en otras palabras, el Software). Gracias a la programación flexible que ofrece el software separado del hardware se incrementó la capacidad de cálculo.

En 1948, el matemático y filosofo estadounidense Norbert Wiener, estableció el

término *"cibernética"* como el estudio del control y la comunicación en animales y máquinas. Él creía que los seres vivos, como los seres humanos y los animales, podían comunicarse con las máquinas bajo una serie de principios básicos. Siendo el control el primero de ellos, estableciendo que *"Todas esas entidades se esfuerzan por contrarrestar la entropía y controlar su entono utilizando el principio de retroalimentación"*, de acuerdo con lo que enunciaba, Wiener se refería a la retroalimentación como la *"capacidad de adaptar el comportamiento futuro a la experiencia pasada"* en la que, mediante un mecanismo de ajuste y retroalimentación constante, los seres vivos y las máquinas logren una comunicación fluida y constante.

1.1.2. Establecimiento de las bases teóricas de la IA

1.1.2.1. a) 1.ª Ola de la Inteligencia Artificial (1956-1973)

En 1956, el concepto de inteligencia artificial se materializó a través del programa *Logic Theorist*, creado por Allen Newell, Cliff Shaw y Herbert Simon. Logic Theorist fue un programa diseñado para imitar las habilidades de resolución de un humano y fue fundado por la Corporación de Investigación y Desarrollo (por sus siglas en inglés, RAND) [1]. Es considerado como el primer programa de inteligencia artificial y fue presentado en el Proyecto de Investigación de Verano sobre Inteligencia Artificial de Dartmouth (DSRPAI), organizado por John McCarthy y Marvin Minsky en el mismo año. En dicha reunión, McCarthy reunió a varios investigadores de diversos campos a un debate abierto sobre la inteligencia artificial. Sin embargo, durante la conferencia, hubo un desacuerdo sobre los métodos estandarizados adecuados para ese campo. Este suceso se denominaría como *"La Primera ola de la IA"*, dado que a partir de dicho evento se desarrollaron diversos programas primitivos que permitían competir contra el propio usuario, aunque

no eran del todo perfectas.

Las primeras demostraciones de *Machine Learning* tales como Newell, *Simon's General Problem Solver* y *ELIZA* de Joseph Weizenbaum se mostraron prometedoras hacia la resolución de problemas y la interpretación del lenguaje hablado.

Durante el periodo de la primera ola, prosperaron dos enfoques principales, que encaminarían más tarde el futuro de la IA. El primer enfoque, el Conexionista, se centra en el procesamiento de información mediante el modelo de redes neuronales artificiales; mientras que, el segundo, la Simbólica, se basa en el procesamiento y manipulación de símbolos o conceptos lógicos, prioriza los datos cualitativos y no los cuantitativos.

El primer enfoque se basa en una norma formulada por Donald Hebb en 1949 en su libro "The Organization of Behavior", donde postuló que el encendido paralelo de dos neuronas incrementaba la intensidad de la conexión entre ellas, más tarde esta teoría obtendría resultados experimentales favorables tras descubrir el fenómeno de "Potenciación a Largo Plazo (LTP)". A este enfoque se le denominaría más tarde como *"Conexionismo"*, término análogo al del cerebro y sus múltiples neuronas conectadas por sinapsis.

En julio de 1958, la Oficina de Investigación Naval de los Estados Unidos informó sobre un notable hallazgo que evolucionaría a las máquinas al siguiente nivel, pasando de complejos intérpretes binarios a aprendices computarizadas con cualidades humanas primitivas.

Una computadora IBM 704, fue alimentada con una serie de tarjetas perforadas, después de las 50 pruebas realizadas, la computadora aprendió a diferenciar las cartas ingresadas por la izquierda de las ingresadas por la derecha. Este mérito se le atribuye a Frank Rosenblatt, psicólogo estadounidense, quien ideo un algoritmo calificado para concebir una idea original conocido como *"Perceptrón"*.

Rosenblatt creó el perceptrón de la misma forma en que las neuronas operan el cerebro humano, con una estructura similar a este último, el algoritmo se compuso inicialmente de tres elementos: neuronas, enlaces y un parámetro llamado peso, el cual simula la fuerza de la conexión entre neuronas. El Perceptrón es una red neuronal de una sola capa, un algoritmo que clasifica la entrada de dos categorías posibles [2]. La red neuronal realiza una predicción, en caso de equivocarse, se ajusta automáticamente para una predicción más precisa la próxima vez. Alcanzando la precisión requerida después de miles o millones de iteraciones. A fin de estudiar el origen del aprendizaje, se estudió el perceptrón en su forma más simple y con ello el *"Teorema de Convergencia del Perceptrón"*. Aquello probaba que las redes neuronales tenían la capacidad de adaptación y, por tanto, de mejorar.

Las siguientes investigaciones en la rama de las neurociencias formularon modelos más precisos de aprendizaje asociativo entre las neuronas, tales como "Regla de la Covarianza", "Regla de la Oja", etc.

En la década de los 60 y 70 se realizaron múltiples ensayos para generalizar el aprendizaje y extenderlo a múltiples capas, resultando en la mayoría de los casos en resultados catastróficos. Uno de los experimentos dirigidos por Rosenblatt fue el diseño de una máquina para reconocer las diferencias entre hombres y mujeres en un álbum de fotos. Aunque inicialmente el perceptrón parecía prometedor, luego se demostró que no podían entrenarse para reconocer diversas clases de patrones. Rosenblatt y sus colegas desconocían que el origen de sus fallas se derivaba de la simplicidad del algoritmo; teniendo en cuenta que para reconocer patrones complejos se requería más neuronas ocultas.

Henry J. Kelly sienta las bases del *"Modelo de propagación hacia atrás"* (también conocido como *Backpropagation*) en 1961. Un año más tarde, Stuart Dreyfus realizó una

versión optimizada basándose en la regla de la cadena, adaptando las variables controladas en proporción al gradiente del error. En su artículo *"The numerical solution of variational problems"* expone un modelo de retropropagación utilizando la regla de la cadena en lugar de la programación dinámica, siendo este último el estándar en modelos anteriores de retropropagación. En el año de 1965, Alexey Grigoryevich Ivakhnenko y Valentin Grigorevich Lapa, llevaron a cabo los primeros intentos en desarrollar algoritmos de aprendizaje automático mediante modelos con funciones de activación polinómica.

Finalmente, en el año de 1969, Marvin Minsky junto a Seymour Papert publicaron *"Perceptrons: An Introduction to Computational Geometry"*, en ella presentaron pruebas matemáticas que demostraban las fortalezas y delimitantes del perceptrón. No obstante, Minsky y Papert criticaron duramente el rendimiento del perceptrón en el cálculo de la función XOR, lo que más tarde le valdría el término de la financiación por parte del gobierno estadounidense.

En la tesis de maestría de Seppo Ilmari Linnainmaa, presentada en la Universidad de Helsinki en 1970, se formuló el modo inverso de diferenciación automático, que más tarde sería conocido como *"Backpropgation"*, para calcular los coeficientes de la expansión de Taylor, primordialmente los coeficientes de primer y segundo orden, dejando fuera su potencial aplicación en las redes neuronales.

De forma simultánea al estudio del perceptrón surgieron, nuevos enfoques, incluida la Inteligencia Artificial Simbólica. El choque de ambos enfoques se debió a la competencia por la financiación, la aceptación del campo científico y la potencia informática. Dando lugar al enfoque simbólico como el nuevo vencedor, cuyos procesos y logros resultaban más fáciles de entender.

Desafortunadamente, las intenciones del libro se malinterpretaron como un ataque hacia el enfoque del conexionismo, apartándolo del reflector y dirigiendo la atención de

los científicos hacia la IA Simbólica. En años posteriores, los investigadores descubrieron que las redes neuronales tenían mayor ventaja en problemas donde la incertidumbre influya.

La Inteligencia Artificial Lógica nace en la década de los 70, esta rama comprende que las computadoras aprenden codificando reglas lógicas con fórmulas del tipo — Si X, entonces Y — [3], el principio de este enfoque está en seguir las reglas a fin de comunicarse mediante símbolos humanos.

Inicialmente, la forma de expresión para el conocimiento y resolver problemas era mediante simbolismos. Las computadoras han evolucionado diversas lógicas y algoritmos que permiten codificar reglas lógicas por medio de sintaxis del tipo — Si X, entonces Y —. Se ideó que la IA debería ejecutarse de la misma forma que lo realizan los computadores, a través de ceros y unos, a fin de lograr interferencias lógicas mediante esta innovadora herramienta.

Consecutivamente, se establecieron lenguajes para la operación de programación de expresiones simbólicas a fin de proporcionar datos estructurados, entre ellos podemos encontrar: LISP y Prologue [4].

A partir de la implementación de los lenguajes simbólicos, se formularon, más tarde, programas especiales denominados como *"Expert System"*. Citando la definición realizada por Stevens L. en su trabajo *Artificial Intelligence. The Search for the Perfect Machine (1984)* menciona que *"Los Sistemas Expertos son máquinas que razonan como un experto lo haría en una cierta especialidad o campo [...], no solo realiza las funciones tradicionales de manejar grandes cantidades de datos, sino que también manipula datos de forma tal que el resultado sea inteligible y tenga significado para responder a preguntar incluso no completamente especificadas"* [5]. Se puede decir que un sistema experto es, en pocas palabras, un sistema informático constituido por

hardware y software que simular el razonamiento de experto humano en un área específica.

Algunas razones para utilizar los sistemas expertos son las siguientes:

1. El conocimiento de diversos expertos en un determinado campo se puede compaginar, resultando en un sistema más fiable y uniforme.

2. El tiempo de respuesta de un sistema experto es más corta y eficiente hacia un problema en comparación a un experto humano.

3. Los sistemas expertos proporcionan respuestas rápidas y confiables en ocasiones donde los expertos humanos no logran encontrar. Gracias a la elevada capacidad de los ordenadores de procesar velozmente un sinfín de operaciones complejas.

4. También pueden ser utilizados para tareas y operaciones repetitivas

5. Su uso se sugiere en situaciones donde:

    a) Cuando el conocimiento es difícil de adquirir o se basa en reglas adquiridas de forma empírica.

    b) Cuando la optimización continua del conocimiento es fundamental y/o cuando el problema está relacionado a reglas o códigos.

    c) Cuando los expertos humanos son escasos.

    d) Cuando el conocimiento sobre el tema es reducido.

Los ejemplos más conocidos dentro de este tipo programas son DENDRAL y MYCIN, el primero tiene la finalidad de deducir estructuras moleculares químicas, mientras que el segundo se encarga de diagnosticar enfermedades de la sangre.

Las desventajas que presentan estos programas son las grandes capacidades de almacenamiento y razonamiento a requerir, por tanto, su desarrollo es de alto costo e ineficiente por la potencia informática necesaria para el almacenamiento y las operaciones [4].

Estos éxitos convencieron a agencias gubernamentales tales como la Agencia de proyectos de investigación avanzada de defensa (DARPA) a financiar la investigación. Sin embargo, el obstáculo más grande a lo que el equipo de McCarthy tuvo que enfrentarse fue el almacenamiento y procesamiento rápido de la información, ya que para comunicarse es necesario reconocer un amplio número de palabras y entender la complejidad de sus combinaciones, a este fenómeno se le conoce como *"Explosión Combinatoria"*. Para resolverlo se requería de enfoques más heurísticos basados en reglas generales, para reducir el número de combinaciones [3]. No obstante, dichas reglas no se habían creado aún, asimismo la falta de datos para alimentar y la capacidad limitada del hardware provocó que la expansión de la IA se pausará.

Para el año de 1966, el gobierno de los Estados Unidos realiza un informe donde concluye que los avances mostrados hasta el momento eran insuficientes e insatisfactorias, reduciendo la financiación progresivamente durante los próximos diez años siguientes.

En 1973 el informe Lighthill decepcionó a la comunidad científica respecto a la IA, comparando los buenos resultados ofrecidos por las ondas de radio para el aterrizaje de aviones frente a los resultados ineficientes utilizando IA. Ese informe fue el motivo final para terminar la financiación de la IA en Reino Unido.

Finalmente, para el año de 1979, Kunihiko Fukishima ideó el Neocognitron, la cual consiste en una red neuronal artificial que utiliza un diseño jerárquico de múltiples capas; las particularidades que ofrecía este algoritmo permitieron a la máquina aprender patrones visuales, así mismo, admitía ajustar manualmente las características al

aumentar el peso de ciertas conexiones. Durante su desarrollo, Fukushima desarrolló las primeras redes neuronales convolucionales.

1.1.2.2. b) 2.ª Ola de la Inteligencia Artificial (1982-1991)

La relevancia de la Inteligencia Artificial Simbólica y los Sistemas Expertos incrementó nuevamente a inicios de la década de los 80, esto gracias al milagro económico japonés y a la reciente abertura por parte de Japón al mercado global de la electrónica e informática. Dando lugar para el año de 1982, a un innovador sistema de IA basado en Prolog, dicho sistema sería ejecutado para una nueva categoría de computadoras que utilizarían las técnicas y tecnologías de la IA tanto para el hardware como del software [6]. A este proyecto se le denominó *"La quinta generación de computadoras"*, un plan de diez años cuyo objetivo radicaba en impulsar el desarrollo de máquinas competentes en análisis de imágenes, traducción de idiomas y razonamiento humano. Esta nueva tecnología integraría avances tales como la arquitectura VLSI (acrónimo de *Very Large Scale Integration*), sistemas de gestión de bases de datos (DBMS) e interfaz hombre-computadora (HCI).

Posteriormente, países del bloque occidental siguieron su ejemplo, crearon sus propias versiones del proyecto japonés. Estados Unidos por su parte instauró la *Microelectronics and Computer Technology Corporation (MCC)* así como la *Strategic Computing Initiative*, Reino Unido le siguió con el proyecto *ALVEY*, mientras que la Unión Europea impulsó su proyecto *ESPRIT (European Strategic Programme for Research in Information Technology)*. No obstante, los resultados arrojados por los distintos proyectos nacionales fueron desalentadores.

Simultáneamente, en el mismo año, John Hopfield desarrolló un nuevo tipo de red neuronal, compuesta de una sola capa, cuyas neuronas están conectadas entre sí, con

arcos dobles que van en ambas direcciones. Cada peso puede ser positivo o negativo. Esta topología de interconexión dual convierte a la red en una de tipo retroalimentada o recursiva. Entre las aplicaciones de la red Hopfield se encuentra la de la memoria asociativa no lineal. Una memoria asociativa es aquella en la que una parte de una señal es capaz de reproducir la señal completa. Este algoritmo es especialmente útil para regenerar señales incompletas o dañadas por ruido.

En 1984, John McCarthy criticó el pobre rendimiento que presentaba los sistemas expertos de la época. A modo de ejemplo, aplicó el sistema experto MYCIN en un caso real, exponiendo el caso de un paciente con *Cholerae Vibrio*. Los resultados obtenidos del MYCIN fueron la suministración de tetraciclina durante dos semanas. Sin embargo, esta opción ofrecida por la máquina no es viable, considerando que el medicamento mataría las bacterias también sería muy tarde para salvar al paciente, quien moriría sin terminar la medicación.

De acuerdo con el exdirector de la Agencia de Proyectos de Investigación Avanzada de Defensa y Oficina de Tecnología y Ciencia de la Información (por sus siglas en inglés, DARPA/ISTO), Jacob T. Schwartz, concluyó que el éxito de la IA fue moderado en ciertas áreas de investigación muy específicas, no obstante, su fracaso vino al querer expandir su uso a un público más general.

Sorpresivamente, las redes neuronales regresarían al foco público para octubre de 1986, cuando Rumelhart, Williams e Hinton concluyeron en su artículo *"Learning representations by back-propagating errors"* publicado en Nature, que la aplicación del algoritmo de retropropagación hacia diversas tareas demuestra que se puede construir representaciones internas mediante el descenso del gradiente del peso y por ende valdría la pena formular métodos más eficientes descender ese gradiente en las redes neuronales. Este documento daría nuevamente el auge a la investigación sobre redes neuronales.

Finalmente, para el año de 1989, dentro de los Laboratorios Bell, Yann LeCun evidencia la aplicación práctica de la retropropagación en conjunto con las redes neuronales convolucionales para el reconocimiento de números en fotografías, siendo una herramienta muy útil para la lectura de cheques escritos a mano.

Después del colapso de la Unión Soviética en diciembre de 1991, la Guerra Fría terminó y con ello la desactivación del programa de Ronald Reagan conocido "Star Wars", dando fin a la investigación de la IA con financiación militar.

A pesar de la decepción que significó la Inteligencia Artificial en el período del *"Segundo Invierno de la IA"*, Corinna Cortes y Vladimir Vapnik desarrollaron en 1995 una *máquina de soporte vectorial (SVM)*, el cual se encarga de mapear y reconocer la posición de cada dato y generar una separación entre ellos de acuerdo con el patrón presente. La particularidad de este sistema es que permite averiguar la división óptima de los puntos en infinitas dimensiones.

Antes de entrar al nuevo milenio, en 1997, Hochreiter y Schmidhuber introducen el *LSTM* (acrónimos de *Long Short-Term Memory*), siendo el primer modelo supervisado para el aprendizaje de *Redes Neuronales Recurrentes (RNN)*. Este nuevo modelo evita el problema de la pérdida de señal de error entre capas al hacer que las redes LSTM recuerden la información durante un periodo de tiempo más prolongado [7].

1.1.2.3. c) 3.ª Ola de la Inteligencia Artificial (2006-Actualidad)

Las redes convolucionales (CNN) fueron muy populares para el reconocimiento óptimo de caracteres en la década de los 90, sin embargo, para el año de 2006 tomaría un giro inesperado, cuando Jensen Huang, permitió a los desarrolladores acceder y programar las unidades de procesamiento gráfico (GPU) de NVIDIA, a través de una extensión del lenguaje C/C++ conocida como CUDA [8]. Con este salto en la potencia informática, se

permitiría llevar a cabo múltiples cálculos paralelos en sistemas de inteligencia artificial.

En el mismo año, Geoffrey Hinton presentó un algoritmo cuya particularidad radica en su entrenamiento, la cual se efectuaba con datos sin etiquetar. Geoffrey y su equipo denominaron a estos algoritmos como *"Deep Belief Networks (DBNs)"* y consistía en máquinas de Boltzmann. A diferencias de las redes convencionales, estas redes aprenden sin ninguna supervisión las propiedades estadísticas de los datos.

En 2009 se establece la ImageNet, una base de datos por excelencia para evaluar algoritmos de clasificación, localización y reconocimiento de imágenes. Este banco virtual nace de la colaboración entre la investigadora Fei Li, perteneciente de la Universidad de Stanford, y la profesora Christiane Fellbaum de la Universidad Princeton. El objetivo detrás del proyecto es el ser un referente para la investigación y desarrollo de software especializado en reconocimiento de imágenes. Para este proyecto, las imágenes se categorizan en 22,000 clases distintas, totalizando en unos 14 millones de imágenes.

En la competencia de 2012, Alex Krizhevsky en colaboración con Ilya Sutskever y Geoffrey Hinton, desarrollaron un sistema de aprendizaje profundo con una tasa de error del 10 % sobre el error del segundo lugar de la competencia. Dicho algoritmo se nombraría más tarde como AlexNet.

Una CPU estándar realiza de 109 a 1010 operaciones por segundo, mientras que un algoritmo de Deep Learning en un hardware distinto (tal como, GPU, TPU o FPGA) es capaz de ejecutar entre 1014 y 1017 operaciones por segundo. Las operaciones más usuales son matriciales y vectoriales altamente paralelizadas.

El informático Ian Goodfellow formuló en 2014 las *"Generative Adversarial Neural Network"* (por sus siglas en inglés, GAN), un revolucionario método cuyo entrenamiento consta de dos redes neuronales, una generadora y otra discriminatoria, donde la primera

intentará crear datos siguiendo el patrón de los datos de entrenamiento mientras que la segunda buscará optimizar su capacidad de distinguir los datos nuevos de los datos originales. Las GAN pueden aprender a imitar cualquier distribución de datos y pueden generar contenido como cualquier dominio: imágenes, música, voz, etc. [9].

La publicación del artículo *"Attention is all you need"* por un grupo de investigadores de Google y la Universidad de Toronto en 2017, idearon un procedimiento alternativo donde se excluye la recurrencia en su totalidad, en su lugar es posible el entrenamiento con grandes cantidades de datos secuenciales basándose únicamente en la atención. Esta nueva red sería nominada como *"Transformers o Transformadores"*, constituida por la combinación de módulos de autoatención, normalización y capas completamente conectadas.

La atención se considera un mecanismo que permite enfocar el interés en particiones específicas de datos para posteriormente realizar relaciones entre las palabras de una oración o secuencia de texto. En vez de efectuar un procesamiento de los datos de entrada de forma progresiva, tal cual lo realiza una RNN, una red Trasformer procesa la entrada de forma paralela; dicha característica le atribuiría de una enorme eficiencia informática a diferencia de otras redes.

Los beneficios que otorga esta nueva red es mayor eficiencia y precisión en comparación a los modelos basados en redes LSTMs y CNNs en tareas relativas al procesamiento de lenguaje natural, tales como la traducción de un texto o el resumen del mismo.

En junio de 2018, OpenAI creó la primera versión de GPT (acrónimo de *Generative Pre-Trained Transformer*), modelo basado en la arquitectura Transformer y entrenado con una gran base de datos de texto (incluyendo libros, artículos y páginas web), con el objetivo de generar respuestas coherentes y contextuales similares a la redacción humana. Gradualmente con cada versión se introducen nuevas características, como

generación de texto en el GPT-2, traducción y resumen presentado en el GPT-3, generación inmediata de respuestas en el GPT-3.5 y ahora con el GPT-4 se introducen las funciones multiidioma, razonamiento lógico y complementos API robustos (tales como Wolfram Alpha y Scholar AI).

El éxito exacerbado de esta tercera ola de la Inteligencia Artificial se debe en gran medida a la creciente disponibilidad de datos de prueba provenientes del internet. Este avance es significativo dado que, para el entrenamiento de modelos pasados, se necesita recopilar una base de datos de prueba para dichos modelos. El internet contribuyó en generar la información digital útil para el desarrollo de la IA. No obstante, la mayoría de los casos de éxito de las inteligencias artificiales son más eficaces al tener menos perturbaciones ambientales. Por consiguiente, se debe considerar al *"Deep Learning"* como una herramienta útil pero no infalible.

Dentro de las áreas de mejora del Deep Learning, está el de sobreajuste, que se produce al superponer objetos distintos entre sí, que impiden a la IA entender el objeto principal. Los modelos son entrenados con base en la probabilidad de ocurrencia para un aprendizaje eficiente.

La IA Neuro-simbólica se plantea para hacer frente a las deficiencias de la IA Simbólica y las Redes Neuronales, combinando las fortalezas de ambas. El informático Judea Pearl propone en su escrito *"The seven tools of causal inference, with reflections on machine learning (2019)"*, una jerarquía compuesta de tres niveles: asociación, intervención y razonamiento, donde además asegura que el único método para lograrlo es mediante el *"Machine Learning"*. La gestión de la próxima generación de inteligencia artificial se dirige hacia la generalización y razonamiento, aun con pocos datos disponibles.

1.2. Futuras implementaciones a realizar

La Inteligencia Artificial puede aumentar el valor de bienes personalizando productos, mejorar el servicio al cliente y crear nuevas categorías de productos según las necesidades de la sociedad. No obstante, se debe considerar los inconvenientes a su expansión, dado que, la IA reemplazará una gran cantidad de ocupantes humanos, liberando recursos humanos para trabajos con mayores requisitos.

1.2.1. Regulación en la IA

Michael Haenlein y Andreas Kaplan proponen que, para regular eficientemente el uso excesivo e indebido de la Inteligencia Artificial es necesario establecer dictámenes sociales, los cuales deberán ser aprobados por la sociedad civil, que funjan como estándar para el entrenamiento y prueba de algoritmos de la Inteligencia Artificial [10].

Se teme el monopolio en los sectores que involucren la Inteligencia Artificial, así como los verdaderos motivos del estado para regularlo, particularmente en Estados Unidos y China. Tres aspectos significativos son:

a) Fuentes potenciales de poder de mercado.

b) Mayor acaparamiento en áreas con aplicaciones de IA.

c) Competencia ilícita y la discriminación de precios.

1.2.1.1. a) Fuentes potenciales de poder de mercado:

El mercado con mayor riesgo al monopolio de la IA es la industria de diseño y producción de chips y semiconductores, así como de materias primas [11]. El primer desafío que suponen son los modelos de Inteligencia Artificial generativa, las cuales

demandan de chips especializados y una gran cantidad de potencia informática durante largos lapsos de tiempo. Una solución más rentable al punto anterior es la computación en la nube, alternativa barata y de costo variable en lugar de la costra infraestructura a invertir para un servidor óptimo.

Asimismo, otra problemática común es el acceso a los datos, requiriendo de grandes cantidades e infraestructura especializada para su almacenamiento, de forma que estos permanezcan vigentes como fuentes en otras actividades comerciales.

Similar al asunto anterior, las materias primas, herramientas y posición geográfica son factores decisivos en la fabricación de chips y GPUs, considerando que estas ventajas se pueden interpretar como posibles riesgos comerciales y geopolíticos, resultando en restricciones comerciales a nivel internacional a fin de proteger el propio desarrollo nacional o perjudicar los competidores internacionales al negar el acceso a insumos claves.

1.2.1.2. b) Acaparamiento en aplicaciones con IA:

En enero de 2024, la Comisión Europea promovió una convocatoria a fin de reunir información sobre el nivel de competencia en los mundos virtuales y la IA generativa, así como el posible rol a desempañar de las autoridades antimonopolio de la Unión Europea. El escrito cita la inversión realizada por Microsoft en OpenAI en sospecha de un caso de competencia desleal, asimismo la autoridad competente de Reino Unido se encuentra investigando el mismo caso.

1.2.1.3. c) Competencia desleal y discriminación de precios:

Las empresas utilizan Inteligencia Artificial para retirar excesos y manipular de los consumidores o destensar la competencia de precios. Esta herramienta permite a las

corporaciones conocer más a su público dirigido, favoreciendo la capacidad a discriminar precios a través de las manipulaciones mentales involuntarias. Acemoglu ha sugerido en diversos trabajos anteriores que la IA generativa puede dar soporte a técnicos al ofrecerles mayor conocimiento de un tema en campo y de esta forma, aumentar la experiencia del trabajador. La información fiable que pueden proporcionar rápidamente las herramientas de IA generativa puede conducir a mejoras significativas en la productividad [12]. Las ventajas que ofrece el análisis de datos hacia un público en particular podrían dirigir a las compañías a reducir la competencia de precios en toda la economía.

1.2.1.4. Acciones reguladoras por la comunidad internacional:

La velocidad, alcance y enfoque son los tres aspectos fundamentales por regir para alcanzar la regulación en materia de Inteligencia Artificial [13]. Si bien los desafíos son los mismos, la forma en como se abordan son diferentes para cada país.

La velocidad con la que avanza esta tecnología obliga a las autoridades a tomar medidas, ya sean flexibles (como el caso de Reino Unido y Japón, que optan por esta vía), rígidas (como sucede en la Unión Europea). Asimismo, el alcance varía enormemente entre los reguladores; aplicando desde prohibiciones temporales y sugerencias sobre buenas prácticas, hasta regulaciones más estrictas que combinan dos enfoques: la evaluación de riesgos antes de ocurrir (considerando las características del sistema o las áreas a utilizar) y la responsabilidad después de ocurrir el riesgo (dirigida a los desarrolladores y proveedores de IA).

Por ejemplo, las medidas regulatorias en Estados Unidos y la Unión Europea únicamente se aplican a tecnologías generales, mientras que China regula los algoritmos y su contenido de forma directa. Por último, en términos de enfoque, Estados Unidos apuesta por la descentralización de políticas a fin de su implementación en instituciones

federales, mientras que la Unión Europea adopta un modelo más centralizado para evaluar situaciones de riesgos y responsabilidad.

Si bien, la mayoría de regulaciones dentro de la comunidad internacional consideran los riesgos relacionados con el monopolio, la ética y la privacidad, Estados Unidos abarca la seguridad nacional y el uso militar de la IA. Sin embargo, en lo que respecta a las leyes sobre derechos de autor del material generado por una inteligencia artificial, las normativas sieguen siendo laxas, dejando en algunos casos la resolución en manos de los tribunales, como es el caso de Estados Unidos y China.

En la tabla 1.1 se especifica los aspectos considerados en la normativa de cada país respecto al uso y la explotación de la Inteligencia Artificial.

Tabla 1.1: Regulaciones Específicas en EU, UE y China

País	Regulación	Alcance	Resolución	Rivalidad	Privacidad	Copyright	Militar	Ético	Finanzas
UE	Adopción de la Ley Provisional de Inteligencia artificial	Basado en una evaluación antes de ocurrir	Aplicaciones, modelos para propósitos generales y sistemas con IA integrada	GN	✓	✓/GN	X	✓	✓/GN
US	Orden Ejecutiva sobre IA segura, protegida y confiable	Descentralizado basado en lineamientos y principios y prioridades nacionales	Modelos de IA incluido el uso dual en la versión base	✓	✓	✓	✓	✓	GN (incluyendo ciber-seguridad)
China	Medidas Provisionales para la gestión de los servicios de IA Generativa	Diseño y contenido ético y responsabilidad después de ocurrir un riesgo	Usos públicos	✓	✓	✓	X/GN (incluyendo la seguridad nacional)	✓	GN

[1] Esta tabla cubre las regulaciones específicas de la IA, mientras que señalamos con "GN"(como general) cuando otras regulaciones/leyes incluyen estas cuestiones en un contexto general. No se considera las leyes nacionales para cada país de la Unión Europea ni las regulaciones específicas de cada estado de los Estados Unidos.

2 Fuente: Extraído de [11].

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar y evaluar algoritmos de diagnóstico de fallas en rodamientos de motores eléctricos, basados en dos enfoques complementarios. El primero se centra en el análisis de vibraciones, que abarca el estudio de señales en el dominio del tiempo y la frecuencia, así como técnicas de filtrado, segmentación (ventaneo) y aplicación de funciones de ventana. El segundo enfoque se basa en las Redes Neuronales Artificiales, en particular en una arquitectura de red profunda conocida como *Long Short-Term Memory* (LSTM), orientada al reconocimiento de secuencias de defectos en rodamientos que combinan información temporal − espectral.

1.3.2. Objetivos Particulares

1. Comprender los fundamentos del análisis de vibraciones aplicado a turbomaquinaria.
2. Analizar los principios de las redes neuronales artificiales, que incluyen el perceptrón multicapa, el algoritmo de retropropagación y las redes profundas LSTM.
3. Utilizar un banco de datos experimentales para la detección de fallas en rodamientos de motores eléctricos.
4. Aplicar técnicas de procesamiento de señales y extracción de características, como filtros digitales, segmentación por ventanas y funciones de ventana, así como características estadísticas temporales y características del espectro.

5. Desarrollar, implementar y evaluar metodologías de diagnóstico de fallas en rodamientos mediante algoritmos y códigos computacionales.

6. Calcular métricas de rendimiento de la red LSTM, tales como *Accuracy, Loss, RMSE, Precision, Recall* y *F1-Score*.

7. Evaluar la importancia de cada característica en el rendimiento de la red LSTM mediante permutación.

1.4. Justificación

Los rodamientos son elementos fundamentales para el funcionamiento adecuado de muchas máquinas rotativas, como aerogeneradores, motores eléctricos, turbinas de vapor, turbinas de gas e hidráulicas, generadores, entre otros. Tener un sistema de diagnóstico que detecte e identifique fallas potenciales en estos componentes es esencial para prevenir fallas catastróficas y optimizar la operación industrial.

Si bien existen técnicas tradicionales de monitoreo y diagnóstico, como el análisis de vibraciones, el creciente desarrollo de la inteligencia artificial y su aplicación en ingeniería plantean nuevas oportunidades para mejorar estos sistemas de diagnóstico.

Por ello, el presente trabajo propone el desarrollo de metodologías basadas en redes neuronales profundas, implementadas mediante algoritmos computacionales, con el fin de optimizar la identificación de fallas en rodamientos a partir de señales de vibración.

Capítulo 2

Redes Neuronales

El *"Machine Learning (ML)"* es una rama de la inteligencia artificial que emplea un conjunto de técnicas basadas en algoritmos matemáticos (operaciones binarias o lógicas) y modelos estadísticos. Su objetivo es estudiar y desarrollar algoritmos computacionales capaces de procesar grandes volúmenes de datos, identificar patrones y extraer conclusiones o predicciones sin necesidad de programación previa. Esto permite a las máquinas aprender y mejorar su eficiencia en la realización de tareas sin detallar explícitamente las instrucciones para cada una de ellas.

2.1. Introducción a las Redes Neuronales Artificiales

Las *"Redes Neuronales Artificiales (RNA)"* son sistemas de procesamiento de información basados en la estructura y funcionamiento de las redes neuronales biológicas. Una red neuronal artificial está compuesta por la interconexión de unidades elementales, llamadas *neuronas*, que se organizan en capas. Cada una de las neuronas aplica una función determinada a los valores de sus entradas procedentes de las

conexiones con otras neuronas, y así se obtiene un valor nuevo que se convierte en la salida de la neurona [14].

Los sistemas computacionales tradicionales siguen una secuencia serial en su procesamiento, similar a un computador de un único procesador, encargado de ejecutar instrucciones una por una y procesar datos almacenados en la memoria. La arquitectura de estos algoritmos permite el procesamiento en paralelo, admitiendo la activación simultánea de múltiples neuronas y una distribución eficiente de la información a lo largo de las conexiones de la red. Por ende, favorece la tolerancia hacia los errores, ya que cada neurona lleva consigo un cálculo simple. De esta forma, aunque una pequeña parte de la red esté defectuosa, su rendimiento no se ve afectado, ya que el funcionamiento de la red depende de la intervención de todas las neuronas.

Además, las RNA se caracterizan por ser adaptativas, aprendiendo de la experiencia adquirida después de un proceso de entrenamiento exhaustivo. Este proceso radica en someter a la red neuronal a un conjunto de situaciones similares al problema a resolver. El aprendizaje finaliza una vez que la red ajusta los pesos de las conexiones entre neuronas al comprender conceptualmente el problema.

Otras ventajas que proporcionan las redes neuronales son las siguientes:

- Autoorganización. Capacidad de organizarse apropiadamente frente a la información dada en la etapa de aprendizaje mediante métodos matemáticos tales como *Adeline, Madeline, Perceptrón, etc.*
- Generalización. Facultad de responder adecuadamente al exponerse a nuevos datos o situaciones no vistos previamente.
- Tolerancia a fallos. Además de los puntos anteriormente mencionados, también poseen el reconocimiento de patrones para la detección de datos con distorsión,

ruido o incompletos.

- Posibilidad de implementación en VLSI. Habilidad de aplicarse a sistemas de tiempo real mediante simulaciones en computadores o dispositivos con hardware especial que integran esta tecnología.

- Fácil integración en la tecnología actual. Actualmente, las principales empresas de semiconductores invierten cada vez más en el diseño y fabricación de chips dirigidos al sector de la inteligencia artificial, mejorando la capacidad en ciertas tareas.

Se denomina al proceso de resolución de problemas empleando Redes Neuronales Artificiales como *"Cómputo Neuronal Artificial"*, el cual pretende reemplazar y optimizar el cómputo que se lleva a cabo por el cerebro humano, considerando que la velocidad de procesamiento del cerebro humano es lenta en comparación a un computador actual.

Una diferencia notable entre redes neuronales biológicas y artificiales es la organización de las neuronas y sus conexiones para la realización de tareas específicas. Por un lado, la estructura del cerebro está compuesta de alrededor de $10^{11}(100,000,000,000)$ neuronas; sin embargo, hasta la fecha se desconoce qué partes de la red neuronal se destinan para llevar a cabo ciertas actividades; por lo tanto, no es necesario conocer a qué subred neuronal destinar la razón, así como tampoco conocer el contenido transmitido por cada neurona. Es decir que utilizamos el cerebro desconociendo su estructura y operación interna.

Mientras que, en el caso de las redes neuronales artificiales, su construcción requiere solamente de cientos de neuronas, donde una decena es suficiente para aplicaciones con problemas sencillos. No obstante, es indispensable especificar la arquitectura, la conexión entre neuronas, los puntos de entrada y salida de la red; si el aprendizaje fue satisfactorio, entonces el usuario deberá cargar los datos de entrada y obtener la salida

esperada sin requerir atención a la organización interna. Adicionalmente, cada una de las neuronas que la componen usa operaciones computacionales simples, tales como sumas, multiplicaciones y operadores lógicos. Esta particularidad facilita la incorporación de las RNA tanto en software como en hardware, además de dar pie a su rápida ejecución.

Hoy en día, las RNA proporcionan a la computación clásica una alternativa que aborde aquellos problemas donde los métodos tradicionales son ineficientes. Sus aplicaciones más recurrentes en la industria son las siguientes:

- Procesamiento de imágenes y voz
- Reconocimiento de patrones
- Interfaces adaptables para sistemas hombre-máquina (HMI)
- Control y optimización
- Filtrado de señales
- Planeamiento y predicción

2.2. Analogía entre las redes neuronales biológicas y las artificiales

Los avances tecnológicos contemporáneos han sido inspirados en la naturaleza para diseñar soluciones innovadoras. Las aeronaves han adoptado principios aerodinámicos de las aves para desplazarse y las aletas de rana mejoran el desplazamiento de los buzos. De manera similar a los anteriores avances que surgen de la propia naturaleza, las *"Redes Neuronales Artificiales (RNA)"* comienzan como un prototipo de emular el funcionamiento de las neuronas mediante modelos matemáticos y estadísticos. Similar a las Redes Neuronales Artificiales, que replican el funcionamiento cerebral, la

comunicación neuronal en los organismos vivos se fundamenta en la compleja dinámica entre sistemas nerviosos y hormonales.

La comunicación neuronal del animal y del hombre está compuesta por el sistema nervioso y hormonal, conectados con los órganos responsables de recoger información proveniente de los estímulos (vista, oído, olfato, etc.), así como también de su transmisión, almacenamiento y envío en forma de señal. El sistema neuronal se compone de tres partes fundamentales:

1. **Los receptores**, ubicados en las células sensoriales. Encargados de recoger la información en forma de estímulos, ya sea procedente del exterior o interior del organismo.

2. **El sistema nervioso**. Responsable de recibir la información, decodificarla y transmitirla en forma de señal a los órganos responsables de la reacción y otras partes del sistema nervioso.

3. **Órganos efectores (músculos y glándulas)**. Receptoras de la información y traductoras de señales en acciones motoras, hormonales, etc.

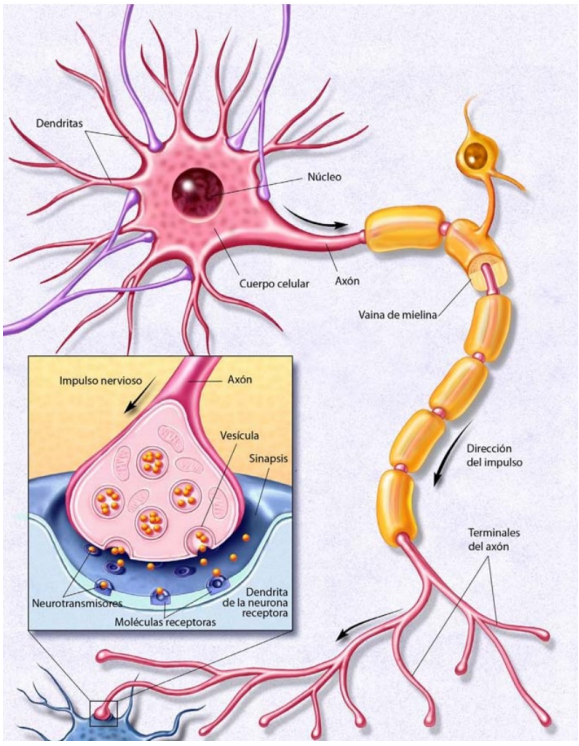
Las neuronas son la unidad estructural y funcional básica dentro del sistema nervioso. Estas son las responsables de realizar las funciones de recepción, procesamiento y emisión de información mediante mecanismos químicos y eléctricos en su membrana plasmática. Funciones que son realizadas de forma grupal, al conectarse en redes de comunicación mediante sinapsis.

Las neuronas se constituyen de

- **El Núcleo:** Donde la mayor parte de la computación neuronal se efectúa.

- **El soma:** Cuerpo celular que contiene el núcleo junto con la información genética (ADN) y genera la energía necesaria para que la neurona trabaje correctamente.
- **Las Dendritas:** Ramificaciones donde la neurona recibe las señales de entrada.
- **El Axón:** Estructura nerviosa con forma alargada y delgada, responsable de propagar el impulso nervioso desde el soma neuronal hacia otra célula nerviosa. La propagación puede ser unidireccional o bidireccional. En su zona final se ramifica en terminales y en cada una termina en una sinapsis, donde una neurona transmite su señal al propagarla con sus conexiones cercanas; por ello a esta zona se le conoce como *"Región Pre-Sináptica"*.

Figura 2.1: Morfología de una Neurona Biológica



Fuente: Ilustración extraída de [15]

De acuerdo a su función, las neuronas se pueden clasificar en tres tipos:

a) **Neuronas aferentes:** Neuronas que reciben la información de los receptores sensoriales (vista, tacto, olfato y gusto). Son las entradas provenientes del exterior.

b) **Neuronas eferentes:** Neuronas que conducen la información hacia las estructuras ejecutoras, provocando una reacción en los músculos. Además, se encarga de la secreción de hormonas por parte de las glándulas. Son las salidas enviadas al exterior.

c) **Interneuronas:** Neuronas que actúan como puentes que conectan con otras neuronas dentro del sistema nervioso central. Se pueden considerar como las entradas provenientes de otras neuronas.

En una red artificial, las neuronas aferentes son las entradas de la red, las interneuronas son las neuronas ocultas y las neuronas eferentes son las salidas de la red. Haciendo una comparación entre el cerebro humano y un computador, tenemos que las dendritas equivalen a los dispositivos de entrada, el núcleo de la neurona constituye el procesador y las sinapsis representan a los dispositivos de salida. Esta similitud entre el cerebro y un ordenador contribuye a una mejor concepción de la propagación de las señales en una neurona biológica.

Las señales que se encuentran en una neurona biológica son de carácter eléctrico y químico. La señal generada por la neurona y transmitida a lo largo del axón es eléctrica, mientras que la señal propagada entre las terminales del axón y las dendritas de las siguientes neuronas es química. Esta última se realiza específicamente mediante los neurotransmisores que fluyen a través de una región especial, llamada sinapsis, que se encuentra entre los terminales del axón y las dendritas de las neuronas siguientes.

El proceso electroquímico en una neurona se desarrolla en los siguientes pasos:

1. Las neuronas reciben las señales de entrada que alteran su comportamiento, y posiblemente, el de otras neuronas o músculos.

2. Las señales son alteradas por los pesos sinápticos. Estos se encargan de bloquear o liberar las señales derivadas de otras neuronas.

3. Las señales recibidas se almacenan en el cuerpo de la neurona, y si el potencial de acción alcanza un límite crítico, se propaga en forma de una señal de salida que es recibida por otras neuronas o actúa como estímulo en los órganos efectores.

El *"Potencial de Acción"* es el mecanismo responsable de la transmisión eléctrica a lo largo del axón. Este proceso se origina por el desplazamiento de los iones de sodio, cargados positivamente, desde el fluido extracelular hacia el interior de la célula, seguido del desplazamiento de iones de potasio, cargados negativamente, en el sentido contrario. Estos impulsos eléctricos alcanzan una amplitud máxima de 100 mV con una duración de un par de milisegundos. Estas señales son de baja frecuencia y no pueden viajar entre células directamente a causa del espacio sináptico, ocasionando que, al llegar al extremo del axón de una neurona, el impulso eléctrico provoque la liberación de *"Neurotransmisores"*. Posteriormente, se realiza un último procedimiento, conocido como *"Sinapsis"*, donde estos se liberan en el espacio sináptico y se encargan de llevar la señal hasta la neurona siguiente. Al unirse a los receptores de la neurona postsináptica, los neurotransmisores inducen cambios en la membrana de la nueva neurona, generando un nuevo impulso eléctrico.

Existen dos tipos de sinapsis:

a) **Sinapsis excitadoras:** Las cuales permiten que el potencial de acción disminuya la polarización de la membrana postsináptica, contribuyendo a la generación de nuevos impulsos.

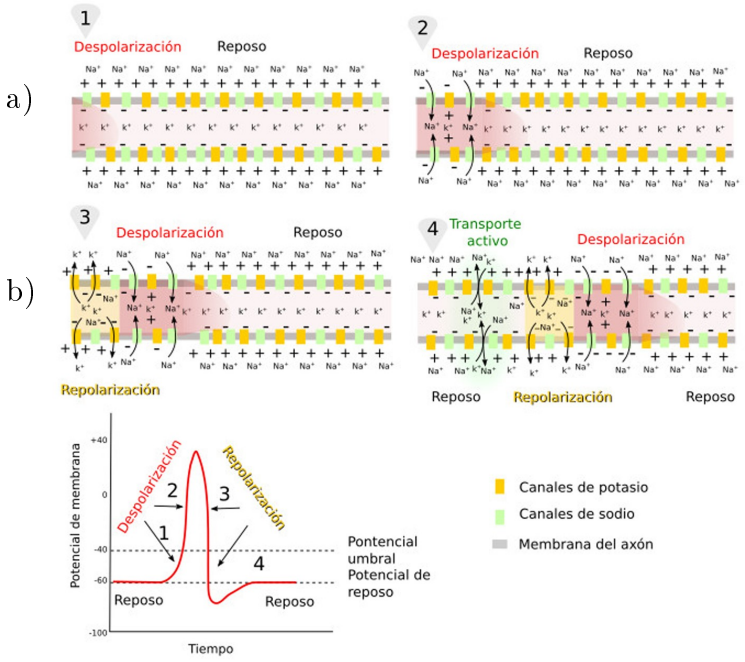
b) **Sinapsis inhibidoras:** Estas priorizan la estabilidad del potencial de acción en la membrana, perjudicando la transmisión de impulsos.

Los neurotransmisores son sustancias químicas encargadas de amplificar, transmitir y transformar las señales neuronales, esenciales para las funciones cerebrales de la persona, su conducta y los procesos de cognición. Desde 1921, aproximadamente 200 neurotransmisores se han identificado. No obstante, este número es impreciso ya que se desconoce la cantidad de neurotransmisores presentes en el cerebro. No obstante, además de los neurotransmisores, se debe considerar la operación típica de una neurona, la cual, a base de iones, define su habilidad de producir impulsos eléctricos.

La condición de una neurona varía en función de la concentración de iones en ambos lados de su membrana. En estado de reposo, la neurona se encuentra polarizada, con una elevada carga de iones de potasio en el interior e iones de sodio en el exterior. La permeabilidad selectiva de la membrana permite mantener este balance. Cuando la permeabilidad se altera, la neurona se despolariza, dando lugar a un impulso nervioso que se desplaza a través del axón. Posteriormente, la neurona se polariza durante un breve momento debido a la liberación de potasio. En este punto, la membrana interna deja de estar ionizada. Tras liberar el impulso, la neurona entra en un estado refractario, donde se reequilibran los iones en ambos lados de la membrana y no se pueden transmitir nuevos impulsos.

Figura 2.2: Fases del intercambio iónico en la membrana celular de la neurona

a) Fase de polarización. b) Fase de despolarización.



Fuente: Esquema extraído de [16].

La neurona biológica es una célula cerebral encargada de la función primordial de recolectar, procesar y emitir señales eléctricas. Se considera que la habilidad del cerebro para procesar información se deriva principalmente de las redes de estas neuronas. Inspirados por esta complejidad natural, los primeros acercamientos en materia de inteligencia artificial se dedicaron a replicar este proceso a través de modelos de redes neuronales artificiales. El perceptrón, modelo pionero de Rosenblatt presentado en 1958, tenía como fin imitar la sinapsis entre neuronas, introduciendo un enfoque revolucionario en la informática: un procesamiento paralelo, distribuido y adaptable. Fue este modelo de inteligencia artificial primitiva el progreso que marcaría el nacimiento de una nueva era tecnológica donde las máquinas serán capaces de emular

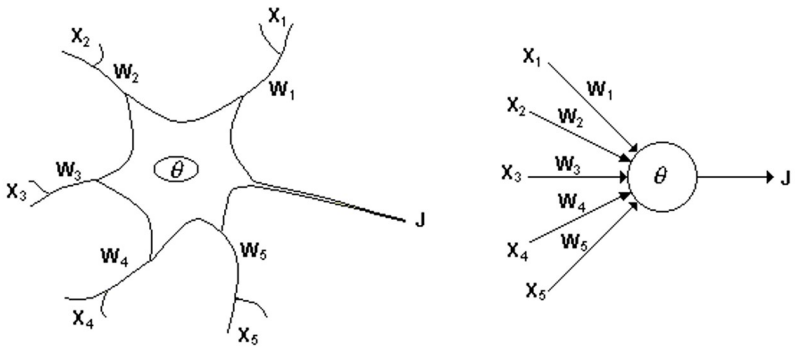
aspectos del raciocinio humano.

2.3. Modelos Neuronales

Una red neuronal artificial (RNA) consiste en la interconexión de un conjunto de unidades elementales llamadas neuronas. Una neurona artificial se puede denominar de distintas maneras, desde *"nodos, neuronodo, celda, unidad o elemento de procesamiento"*. Cada una de ellas aplica una función matemática a los valores de sus entradas procedentes de las conexiones con otras neuronas, y así se obtiene un valor nuevo que se convierte en la salida de la neurona [14].

Una forma de comprender este nuevo modelo es mediante la analogía entre una neurona biológica y una artificial, en la cual las entradas (x_n), las sinapsis o pesos (w_n) y el potencial de acción (J) puede compararse en ambas estructuras. Esta comparación se muestra en la Figura 2.1.

Figura 2.3: Diferencia entre una neurona biológica y artificial



Fuente: Ilustración extraída de [17]

Con base en lo expuesto, la Tabla 2.1 presenta una comparación detallada de los elementos biológicos y artificiales de cada neurona.

Tabla 2.1: Comparación entre elementos artificiales y biológicos de una neurona

Elemento artificial	**Elemento biológico equivalente**	**Función**
Entradas	Dendritas	Son los responsables de recibir las señales sin procesar derivadas de una fuente externa y transmitirlas al resto de elementos. En el caso biológico, las señales son estímulos provenientes de los sentidos, mientras que, en el caso artificial, las entradas representan datos o valores que la neurona recibe para su posterior procesamiento.
Pesos sinápticos	Sinapsis y Neurotransmisores	Determinan el paso o bloqueo de las señales hacia la siguiente neurona. En el caso biológico, dependiendo del tipo de sinapsis (excitadora o inhibidora), se determina si el impulso eléctrico se propaga o se detiene. En el caso artificial, al multiplicar las entradas con valores aleatorios, se priorizan unas señales sobre otras, las cuales, dependiendo del efecto (positivo o negativo), modifican la propagación hacia otras unidades.
Función de activación	Potencial de Acción	Producen la señal de propagación hacia sus semejantes o en forma de salida. En el caso biológico, se produce una señal eléctrica cuando la neurona se despolariza, al rebasar el límite de la permeabilidad de la membrana neuronal. En el caso artificial, la función matemática aplica cierto valor, el cual evalúa qué neuronas producen una salida o no.

Fuente: Elaboración propia

Los datos ingresan por medio de la *"capa de entrada"*; estos datos son procesados dentro de las *"capas ocultas"*. El número de capas ocultas está dado por la arquitectura de la red y, por último, los datos procesados salen por la *"capa de salida"* [18]. Generalmente,

se constituye dentro de 3 capas, aunque el número de capas ocultas puede variar de acuerdo al *"Pre-procesamiento"*.

Usualmente, se suelen encontrar tres tipos de unidades o capas en una red neuronal artificial:

1. **Unidades de entrada:** Es la primera de la red. Recibe los datos provenientes de fuentes externas a la red (datos de entrada).

2. **Unidades ocultas:** Son las capas internas de la red que no están en contacto con el ambiente exterior. Las neuronas ubicadas en las capas ocultas pueden estar interconectadas en distintos patrones, determinando distintas topologías de redes neuronales.

3. **Unidades de salida:** Es la última capa de la red, responsable de transferir el resultado hacia el exterior.

El procesamiento de las redes neuronales artificiales recae en los siguientes bloques:

1. **Arquitectura de la red:** Se refiere a la organización estructural de las neuronas en capas y las configuraciones de sus conexiones. Esta disposición cambia de acuerdo al tipo de red y afecta tanto su capacidad de resolución como la calidad de los resultados. Además, las funciones de activación utilizadas en cada capa también influyen en el rendimiento y comportamiento de la red. De acuerdo a la topología, las redes neuronales artificiales pueden clasificarse de acuerdo a las siguientes maneras:

   a) Según número de capas

   - Redes neuronales monocapas: Se trata de la red neuronal más simple, compuesta de tan solo una capa de neuronas. Las entradas se dirigen a la

capa de neuronas de salida, que realizan los cálculos. La capa de entrada, al no realizar ningún cálculo, no se considera como tal una capa.

- Redes neuronales multicapas: Se compone de una estructura cuya facultad permite la interconexión entre neuronas. En el primer nivel, las unidades reciben los valores cuyos patrones son vectores que alimentan a la red. Para cada vector de entrada, este se introducirá en la red al copiar cada valor de tal vector en las unidades de entrada correspondientes. Cada unidad de la red, tras obtener todas sus entradas, las procesa y produce una salida que se propaga por las conexiones entre las neuronas, llegando como entrada a la unidad final. Una vez que toda la red ha propagado completamente la entrada, se generará un vector de salida, cuyos elementos son cada uno de los valores de salida de las neuronas de salida.

b) Según el tipo de conexiones

- Redes neuronales no recurrentes: Son redes no repetitivas, con unidades básicas de procesamiento, todos ellos conectados con los nodos de las capas previas. Las conexiones tienen diferentes pesos en las neuronas. Este tipo de redes se caracteriza por no tener un ciclo de retroalimentación, por lo que la señal solo fluye hacia una sola dirección, desde la entrada hacia la salida.
- Redes neuronales recurrentes: A diferencia de las redes prealimentadas, la retropropagación posee vías hacia atrás, lo cual significa que la señal puede comunicarse en ambas direcciones usando ciclos, transformándolo en un sistema dinámico no lineal, cuyos cambios son continuos hasta alcanzar el estado de equilibrio.

2. **Ajuste de pesos o aprendizaje:** El aprendizaje es un método donde se modifican los pesos de las conexiones entre las neuronas en una red específica. El aprendizaje puede clasificarse en tres enfoques:

   a) Aprendizaje supervisado: El proceso de entrenamiento implica presentar al modelo una serie de ejemplos de entrada junto con las salidas correspondientes y ajustar los parámetros del modelo para minimizar la discrepancia entre las predicciones del modelo y las salidas reales [19]. Esta diferencia se evalúa mediante una función de pérdida o error, que cuantifica la diferencia entre la salida obtenida por el modelo y la salida que se desea.

   b) Aprendizaje no supervisado: La finalidad de este enfoque es identificar patrones, estructuras o relaciones inherentes en los datos sin la orientación directa de resultados obtenidos. Una de las técnicas más comunes en el aprendizaje no supervisado es el clustering (agrupamiento), donde el objetivo es dividir los datos en grupos basados en la similitud entre las instancias [19].

   c) Aprendizaje por refuerzo: Este enfoque recibe retroalimentación en forma de premios o castigos del entorno, ajustando su comportamiento en consecuencia. El concepto básico sostiene que el modelo adquiere conocimientos mediante la experiencia, examinando diferentes opciones y evaluando las repercusiones de cada acción. Gradualmente, el modelo neuronal idea una estrategia a fin de tomar las decisiones que se apeguen a la máxima gratificación.

3. **Función de entrada:** Tal como observamos en la Figura 2.5, cada entrada tiene su peso determinado. El objetivo de la función de entrada es combinar las distintas entradas con su peso y agregar los valores obtenidos de todas las conexiones de

entrada para obtener un único valor [14]. Las funciones más utilizadas son las siguientes:

- La función de suma ponderada:

$$z(x) = \sum_{j=1}^{n} x_j \cdot w_j^i \tag{2.1}$$

- La función máxima:

$$z(x) = \text{máx}(x_1 w_1^i, ..., x_n w_n^i) \tag{2.2}$$

- La función mínima:

$$z(x) = \text{mín}(x_1 w_1^i, ..., x_n w_n^i) \tag{2.3}$$

- La función lógica AND o OR:

$$z(x) = (x_1 w_1^i \wedge ... \wedge x_n w_n^i), \tag{2.4}$$
$$z(x) = (x_1 w_1^i \vee ... \vee x_n w_n^i).$$

4. **Función de activación o transferencia:** Una función de activación es una ecuación matemática que determina el nivel de excitación de cada neurona. Se representa como $\sigma(x)$. Generalmente, estas ecuaciones son no lineales y su importancia reside en mantener la salida dentro de ciertos límites, tal como lo haría un neurotransmisor. Las funciones de activación más utilizadas son las siguientes:

   - Función lineal: El comportamiento de la neurona es lineal (véase la Figura 2.4a), es decir que la entrada y salida son las mismas.

$$f(x) = x \tag{2.5}$$

- Función escalón: Determina la salida de la neurona en 0, si el argumento x es menor o igual que 0, o 1 si x es mayor o igual a 0 (véase la Figura 2.4b). La salida puede establecerse en los valores $\{-1, 1\}$ o $\{0, 1\}$ [14]. Estas salidas se utilizan para la clasificación de los datos de entrada en dos categorías distintas.

$$y(x) = \begin{cases} 1 & si \quad x \geq 0 \\ -1 & si \quad x < 0 \end{cases} \tag{2.6}$$

- Función logaritmo sigmoide: Esta función toma los valores de entrada n, los cuales oscilan entre $\pm\infty$, posee un gradiente suave y suele producir salidas entre cero y uno (véase la Figura 2.4c). Generalmente, la red se retrasa en realizar una predicción si los valores de entrada son muy altos o bajos debido al fenómeno del *"Problema del gradiente evanescente"*.

$$f(x) = \frac{1}{1 + e^{-n}} = \frac{e^n}{e^n + 1} \tag{2.7}$$

- Función tangente hiperbólica sigmoide (Tanh): Los posibles valores de salida están en el intervalo de $\{-1, 1\}$, donde para cualquier valor de entrada mayor a 0, genera una salida de 1, y en valores menores a 0, se obtiene una salida de -1 (véase la Figura 2.4d).

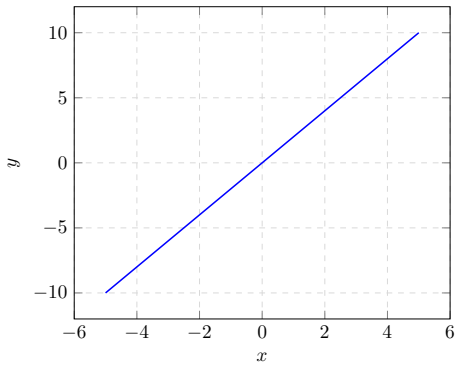
$$f(x) = \frac{e^n - e^{-n}}{e^n + e^{-n}} = \frac{2}{1 + e^{-2n}} - 1 \tag{2.8}$$

- Función rectificadora o ReLu: Se calcula obteniendo la parte positiva de su argumento (véase la Figura 2.4e). Se caracteriza por su alta eficiencia computacional, aunque no es capaz de procesar entradas cercanas a 0 o valores negativos.

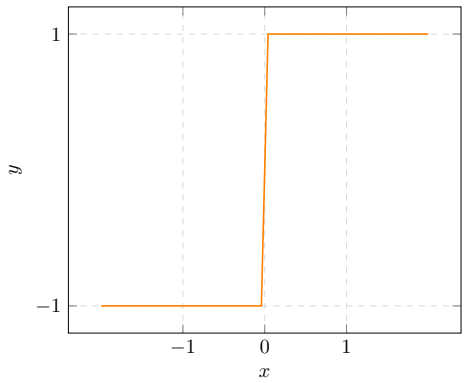
$$f(x) = \text{máx}(0, x) \tag{2.9}$$

Figura 2.4: Funciones de activación

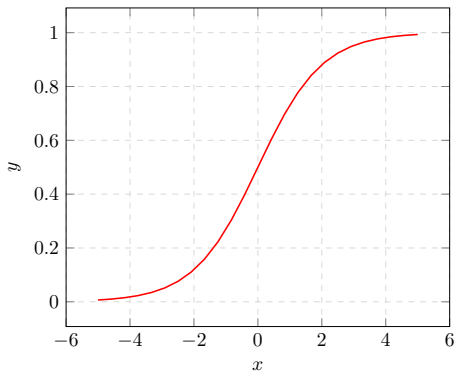
(a) Función Lineal



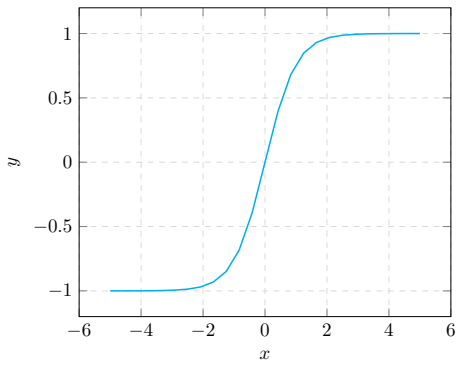
(b) Función escalón



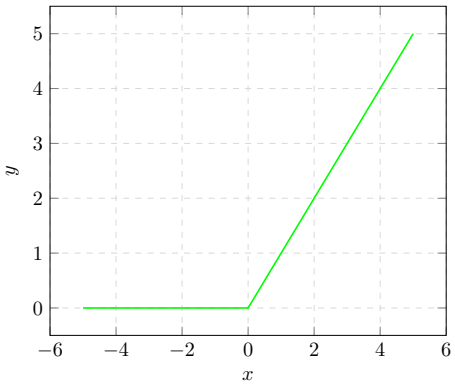
(c) Función Logaritmo Sigmoide



(d) Función Tanh



(e) Función ReLu



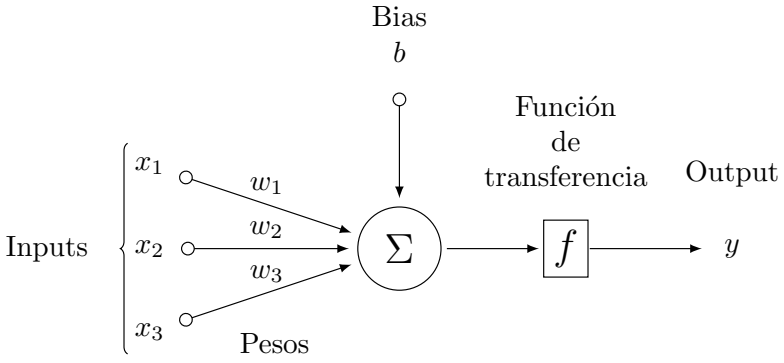
Fuente: Elaboración propia

2.4. Perceptrón

El perceptrón es la forma más simple de una red neuronal usada para la clasificación de un tipo especial de patrones, los linealmente separables (es decir, patrones que se encuentran en lados opuestos de un hiperplano) [20]. La arquitectura del perceptrón consiste en una neurona no lineal con pesos sinápticos ajustables, un nodo sumador y una función de transferencia a la salida, tal como se ilustra en la Figura 2.5.

Los *"pesos sinápticos"*, representados como w_n, son coeficientes asociados a cada entrada. Estos determinan la influencia de cada valor que entra en la neurona en la salida del modelo. Estos pesos se adecúan de acuerdo con el proceso de aprendizaje de la red.

Figura 2.5: Estructura de un perceptrón simple



Fuente: Elaboración propia

Por otra parte, el *"nodo sumador"* calcula una combinación lineal de las entradas junto a los pesos, además de incorporar un sesgo adicional (Ecuación 2.10), independiente de las unidades de entradas, denominado *"umbral o bias"*, representado como b o θ. Este sesgo se adiciona a fin de dotarle al modelo de mayor flexibilidad al momento de presentar nuevos datos. Al incluirse el bias, la neurona ajusta a la salida para clasificar correctamente

datos que de otra manera serían difíciles de distinguir (Ecuación 2.13).

Por último, se deberá especificar la *"tasa de aprendizaje"*, representada bajo el símbolo α, cuyo rol es el de facilitar el proceso de entrenamiento ponderando el delta utilizado para actualizar los pesos [21]. Es decir que, en lugar de reemplazar completamente los pesos anteriores, al sumar el Δw_k (Ecuación 2.14) se añade una proporción del error en la actualización de los nuevos pesos, ofreciendo una mayor estabilidad a largo plazo.

La única neurona de salida del perceptrón realiza la suma ponderada de las entradas, se suma el umbral y se transmite el resultado hacia una función de activación binaria. Debido a la naturaleza de la función de activación, la salida de la red tiene dos opciones para la señal de salida y. Donde la neurona produce una salida igual a $+1$ si la entrada a la función de transferencia es positiva y -1 si la entrada es negativa.

El proceso de aprendizaje del perceptrón se detalla más adelante:

1. Asignación de los valores iniciales de forma aleatoria a los pesos w_i y al umbral b. Se sugiere que estos valores estén en un rango entre -1 y $+1$.

2. Introducción del vector de entrada x_i a la red y definición del vector de salida deseada y.

3. Cálculo de los valores netos procedentes de las entradas mediante una función lineal. Donde z representa la salida que será usada como entrada a la función escalón, la cual se define de la siguiente manera:

$$z = b + \sum_{i=1}^{n} w_i x_i = w_0 x_0 + w_1 x_1 + ... + w_n x_n \tag{2.10}$$

   La flexibilidad que otorga a la red el añadir la constante b a la sumatoria de las entradas proviene a partir de la función escalón, la cual compara la sumatoria neta (z) con el umbral escalón θ:

$$z = \sum_i w_i x_i \geq \theta \tag{2.11}$$

Si restamos en ambos lados de la ecuación el valor θ, se obtiene:

$$z = \sum_i w_i x_i - \theta \geq \theta - \theta \tag{2.12}$$

$$z = \sum_i w_i x_i - \theta \geq 0 \tag{2.13}$$

Por último, se reemplaza el término $-\theta$ con b para indicar el bias; consecuentemente, se procede a mover el nuevo término al frente de la ecuación sumatoria de las entradas y obtenemos la Ecuación 2.10.

4. La función de escalón realiza el cálculo a la salida de la red (véase la fase 3 expuesta en la Figura 2.6).

5. La regla de aprendizaje correctiva de error consiste en calcular el desajuste entre el valor obtenido $\hat{o}$ y el valor esperado y para la fase de entrenamiento, considerando lo siguiente:

   - Si el valor obtenido y el esperado son iguales ($y = \hat{o}$), no se realiza nada.
   - Si el valor obtenido es diferente al esperado ($y \neq \hat{o}$), calcular la diferencia (Δw_k) entre ambos valores y utilizar una fracción de esta para actualizar los pesos de la red.

   Esta regla correctiva se puede expresar de la siguiente forma:

$$\Delta w_k = \alpha\,(y - \hat{o}) x_k \tag{2.14}$$

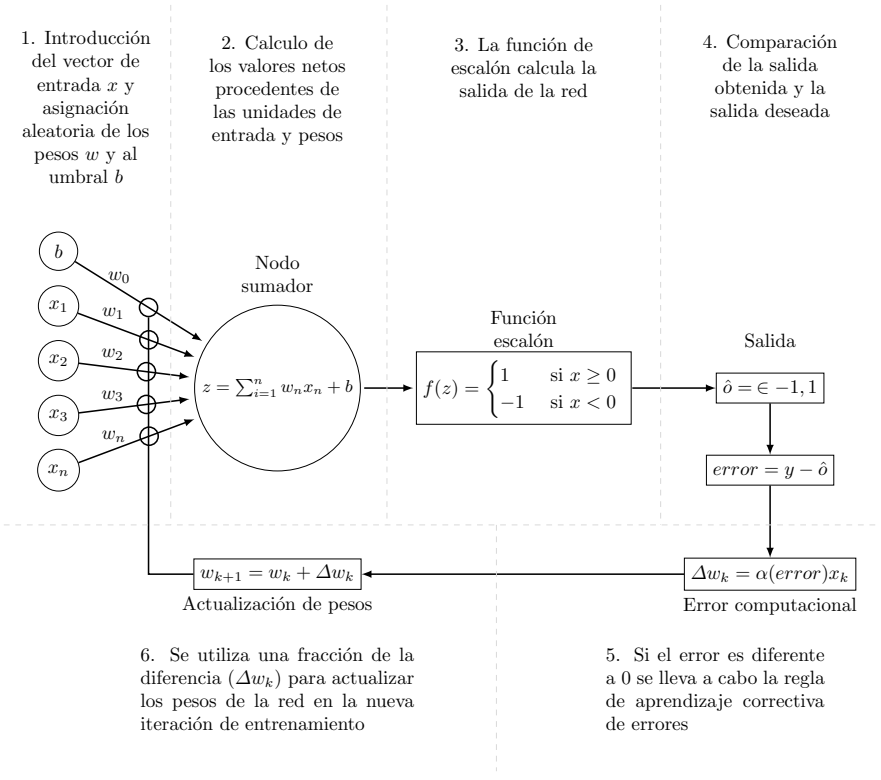
   Donde:

- w_k es el peso para el caso de entrenamiento k.
- η es la tasa de aprendizaje; este valor por lo usual se maneja entre $\{0, -1\}$.
- y es el valor deseado de salida.
- $\hat{o}$ es el valor obtenido por la red.
- x_k es el vector de entrada para el caso de entrenamiento k.

6. Posteriormente, se añade la diferencia (Δw_k) a los anteriores pesos presentes en la red bajo la siguiente expresión:

$$w_{k+1} = w_k + \Delta w_k \tag{2.15}$$

Figura 2.6: Fases de entrenamiento de un perceptrón simple



Fuente: Elaboración propia

2.4.1. Función de coste

La función de coste (también conocida como *"función de pérdida o función objetivo"*) es responsable de medir el desempeño de un modelo y de optimizar sus parámetros para minimizar la pérdida. Su propósito radica en capacitar al algoritmo para generar predicciones exactas. Las funciones de pérdida permiten definir y alcanzar este objetivo de forma matemática, midiendo la tasa de pérdida (o diferencia) de cada entrada al comparar la salida real con la salida obtenida por el modelo. Si las predicciones son exactas, la pérdida es pequeña. Si las predicciones son inexactas, la pérdida será grande.

Cada vez que aplicamos una instancia de entrenamiento al perceptrón, comparamos la salida obtenida con la esperada, y en el caso de diferir, deberemos modificar los pesos y el sesgo de las neuronas para que la red "aprenda" la función de salida deseada [14].

Las funciones de pérdida son exclusivas del aprendizaje supervisado, en el que se proporcionan las salidas correctas, catalogadas como *"axiomas"*. En contraste, los modelos de aprendizaje no supervisados, como la agrupación en clústeres o la asociación, no implican respuestas concretas, sino que buscan patrones intrínsecos en datos no etiquetados. La diferencia consiste en que el primer tipo necesita conjuntos de datos etiquetados, con directrices específicas que proveen información sobre cada ejemplo de entrenamiento.

Hay una variedad de funciones de pérdida, cada una adecuada a diferentes objetivos, tipos de datos y prioridades. Las funciones de pérdida más utilizadas se dividen en tres tipos:

- Funciones de pérdida de regresión: Estos se encargan de medir errores en predicciones de valores constantes o cuantificables. Este tipo puede subdividirse a su vez: de forma lineal o polinomial.

- Funciones de pérdida de clasificación binaria: Miden errores en predicciones cuya salida puede ser 0 o 1. La clasificación se realiza en función del valor umbral (generalmente entre 0.5), donde la $salida > umbral$, entonces se repartirá entre 0 y 1.

- Funciones de pérdida de clasificación multiclase: Miden errores en las predicciones que involucran *"valores discretos"*, es decir, que involucran más variables objetivo.

Las funciones de coste en problemas de regresión, tanto lineales como polinomiales, se encargan de modelar la relación entre una o más variables independientes. Su responsabilidad recae en — *predecir Y a partir de X* —, garantizando la mínima diferencia entre los valores reales y los predichos.

Para las ecuaciones presentadas a continuación, se utiliza la siguiente nomenclatura. Donde:

- n es el número de muestras en el conjunto de datos.
- i es el índice de muestra.
- y es valor real a la salida.
- $\hat{o}$ es valor obtenido a la salida.

Seguidamente, se expone las funciones de coste más relevantes empleadas en esta categoría:

- **Error cuadrático medio (MSE):** Esta función se caracteriza particularmente por la aplicación de la potenciación, resultando eficiente en el cálculo de su derivada. *"Cada error parcial es igual al área del cuadrado creado a partir de la distancia geométrica entre los puntos medidos"*. Todas las áreas de la región se suman y promedian.

No obstante, elevar el error al cuadrado puede provocar que los grandes errores tengan un impacto desmedido en la pérdida total, lo que condena severamente los valores atípicos e incentiva al modelo a disminuirlos. Por lo tanto, la MSE resulta apropiada cuando los resultados objetivos poseen una distribución normal (gaussiana).

Esta función es popular en modelos de regresión basados en el algoritmo de descenso de gradiente. Su ecuación es la siguiente:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{o})^2 \tag{2.16}$$

Algunas adaptaciones de la anterior función se presentan a continuación:

a) **Error logarítmico cuadrático medio (MSLE):** Para los problemas de regresión donde los resultados objetivos cubren un espectro muy extenso de valores potenciales, como aquellos que sugieren un crecimiento exponencial, una severa sanción a la pérdida total ocasionada por grandes errores podría resultar perjudicial.

   El MSLE resuelve lo anterior mediante la media de los cuadrados del logaritmo natural de las diferencias proveniente de los valores reales y los pronosticados. No obstante, el inconveniente del MSLE radica en que penaliza más las predicciones significativamente menores al valor real que las mayores, debido a que la diferencia logarítmica es mayor cuando se subestima que cuando se sobrestima un valor.

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(1+y) - \log(1+\hat{o}))^2 \tag{2.17}$$

b) **Error cuadrático medio raíz (RMSE):** Se trata de una función derivada

del MSE con la distinción de ofrecer una fácil interpretación al expresar las pérdidas en la misma unidad que el valor de salida en sí. Pese a ello, esta ventaja trae consigo un mayor cálculo computacional.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y - \hat{o})^2}{n}} \tag{2.18}$$

- **Error absoluto medio (MAE):** Es la media de diferencias absolutas entre la salida real y la obtenida, donde todas las desviaciones individuales tienen la misma importancia. Se calcula como la suma del valor absoluto de todos los errores dividido entre el tamaño del vector muestra. La influencia es menor en valores alejados de las salidas reales. Así pues, el MAE es óptimo cuando los datos pueden incluir ciertos valores extremos que no deberían influir excesivamente en el modelo.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |(y - \hat{o})| \tag{2.19}$$

Por otra parte, las funciones de coste destinadas a problemas de clasificación pueden ir desde la categorización binaria a la multiclase. La pérdida de clasificación se cataloga en términos de *"entropía"*.

El concepto de entropía cruzada proviene de un artículo titulado *"Una teoría matemática de la comunicación"* escrito por Claude Shannon en 1948, donde la entropía de la información (representada por la Ecuación 2.20) se aborda como la incertidumbre inherente a los posibles resultados. Donde *"Cuanto mayor sea el valor de la entropía, H(x), mayor será la incertidumbre para la distribución de probabilidad"*.

$$H(x) = -\sum p(x) \log p(x) \tag{2.20}$$

Este concepto es adaptado en el campo de las redes neuronales como un término utilizado para medir la incertidumbre dentro de un sistema. No es lo mismo comparar un

lanzamiento de moneda con dados, ya que el primero tiene menor entropía a diferencia del segundo, que tiene mayor entropía al ofrecer mayor número de posibles resultados.

En el aprendizaje supervisado, las predicciones de los modelos se comparan con las clasificaciones de verdad básicas proporcionadas por las etiquetas de datos. Esas etiquetas de verdad fundamental son ciertas y, por lo tanto, tienen baja o ninguna entropía [22].

No obstante, las dificultades a las que enfrenta este último se pueden solucionar de dos formas:

1. Calcular la probabilidad relativa para cada dato individual y su pertenencia a cada categoría posible, seleccionando al final la categoría con la probabilidad más alta. Este enfoque es el más usual a utilizar en redes neuronales con funciones de activación *softmax* en la capa de salida.

2. Descomponer el problema multiclase en varios problemas de clasificación binaria.

A continuación, se detallan las funciones de coste particulares para problemas relacionados con la clasificación.

- **Entropía cruzada:** Es una función de pérdida utilizada para medir el rendimiento de un modelo de clasificación [23]. Calcula la diferencia entre la probabilidad predicha y la probabilidad verdadera, considerando que cada muestra pertenece a una sola clase en particular, obteniendo como respuesta verdadera (1) para esa clase en particular y falso (0) para el resto de clases.

  Entre los tipos de entropía cruzada, se dividen por el número de posibles respuestas, como las binarias (sí/no, verdadero/falso, perro/gato, etc.) y las multiclase, que abarcan más de dos opciones disponibles en el modelo.

a) **Entropía cruzada binaria (Binary Cross Entropy):** En la clasificación binaria, existen dos clases; usualmente, la clase positiva corresponde a 1, mientras que la clase negativa a 0. Esta función mide qué tan lejos del valor verdadero se encuentra la predicción para cada una de las clases; posteriormente se promedia estos errores de clase, resultando la pérdida final presente en el sistema. Esta función se asemeja a la transferencia de bits y la cantidad de estos que se han perdido en el proceso. La siguiente ecuación expresa lo anterior:

$$BCE(y, p) = \frac{1}{N} \sum_{i=1}^{N} (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) \tag{2.21}$$

Donde:

- N es el número total de muestras en su conjunto de datos.
- y_i es el valor real a la salida, dicho de otra forma, la *"etiqueta"* real.
- p_i es la probabilidad prevista de que el punto de datos esté en la clase positiva.

La entropía cruzada funcionará mejor cuando los datos estén normalizados (forzados entre 0 y 1), ya que esto los representará como una probabilidad.

b) **Pérdida de entropía cruzada categórica (Categorical Cross Entropy):** Es una función de pérdida que se utiliza para la categorización de una sola etiqueta. Esto es cuando solo se aplica una categoría para cada punto de datos. En otras palabras, un ejemplo puede pertenecer a una sola clase.

La entropía cruzada categórica compara la distribución de las predicciones (las activaciones en la capa de salida, una para cada clase) con la distribución verdadera, donde la probabilidad de la clase verdadera se establece en 1 y 0

para las otras clases. La ecuación que lo representa es la siguiente:

$$CCE(y, \hat{o}) = -\frac{1}{N} \sum_{i}^{N} \sum_{j}^{C} y_{ij} \log(\hat{o}_{ij}) \tag{2.22}$$

Donde:

- N es el número de muestras en el conjunto de datos
- C es el número de clases
- y_{ij} es la salida real para la i-ésima muestra y la j-ésima clase
- $\hat{o}_{ij}$ es la salida prevista para la i-ésima muestra y la j-ésima clase

2.5. Backpropagation

El algoritmo de *"backpropagation"*, también consiste en dos etapas: la primera de funcionamiento y la segunda de entrenamiento. Donde en la primera se introduce un patrón de entrada X_{pN}, que es transmitido mediante las posteriores capas de neuronas ocultas hasta obtener una salida *"predicha"*. Por consiguiente, en la etapa de entrenamiento se modifican los pesos de la red, de forma que se consiga la salida deseada.

En las siguientes secciones de este documento se presentan de manera detallada ambas etapas del algoritmo.

2.5.1. Etapa de funcionamiento

Al introducir un patrón X_{pN} este se propaga con la ayuda de los pesos w_{ji} desde la capa de entrada hasta la oculta. Las unidades de esta capa intermedia se encargan de transformar las señales obtenidas a través de la implementación de una función de

activación, otorgando así un valor de salida. Esto se realiza gracias a los pesos w_{kj} hacia las neuronas de salida, donde replicando la función anterior se obtiene la salida de la red.

La entrada total a la red (net_{pj}), que recibe una neurona oculta j, se expresa de la siguiente forma:

$$net_{pj} = \sum_{i=1}^{N} w_{ji}\, x_{pi} + \theta_j \tag{2.23}$$

Donde:

- net_{pj} representa la entrada neta para la neurona j correspondiente al patrón p.
- N es el número total de entradas conectadas a la neurona j, donde i tomará valores desde 1 hasta N.
- w_{ji} se refiere al peso sináptico entre la entrada i y la neurona j.
- x_{pi} es el valor de la entrada i-ésima del patrón p.
- θ_j es el bias o umbral asociado a la neurona j.

El valor de salida de la neurona oculta j (i_{pj}) se obtiene al aplicar una función de activación $\sigma(x)$ (usualmente se opta por la función lineal o sigmoide) sobre su entrada neta, tal como se muestra a continuación:

$$i_{pj} = f_j(net_{pj}) \tag{2.24}$$

Asimismo, la entrada neta que recibe la neurona k (net_{pk}) es la siguiente:

$$net_{pk} = \sum_{j=1}^{L} w_{kj}\, i_{pj} + \theta_k \tag{2.25}$$

Finalmente, se concluye esta primera etapa al aplicar la función de activación a la suma ponderada de la neurona de salida k (o_{pk}) de la siguiente forma:

$$o_{pk} = f_k(net_{pk}) \tag{2.26}$$

2.5.2. Etapa de aprendizaje

En esta segunda etapa, se desea obtener la mínima diferencia o error entre la salida obtenida de la red (o) y la salida deseada (y). Dado que el usuario es quien determina la salida deseada, este tipo de entrenamiento que ofrece el backpropagation es de tipo supervisado.

2.5.2.1. Error en las capas ocultas y de salida

La función de error, encargada de reducir la discrepancia para cada patrón p en cierto número de neuronas de salida (M), viene dada de subsecuente forma:

$$E_p = \frac{1}{2} \sum_{k=1}^{M} (y_{pk} - o_{pk})^2 \tag{2.27}$$

Donde y_{pk} es la salida deseada para la neurona de salida k frente al patrón p. De esta expresión se deriva la tasa general del error:

$$E_T = \sum_{p=1}^{P} E_p / P \tag{2.28}$$

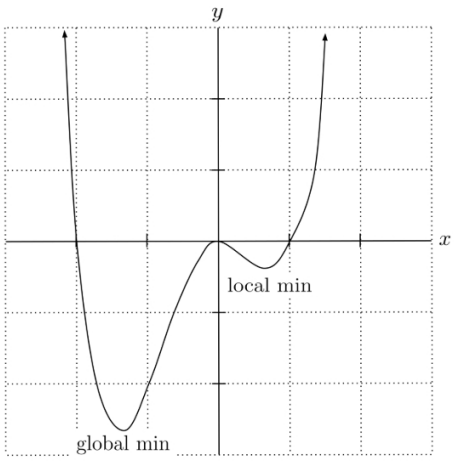
El *"Descenso del gradiente"* o error es una técnica aplicada a fin de reducir las funciones diferenciables de un modelo matemático. Partiendo de una suposición inicial evaluada como el *"mínimo local"*, para posteriormente manipular la diferencia a fin de alcanzar una suposición más real. Es decir, que el descenso del gradiente busca alcanzar el *"mínimo global o absoluto"* cercano a 0.

Para el descenso del error, son posibles dos casos:

a) En caso de que la derivada de la función sea positiva, se infiere que la función está inclinada *"hacia arriba"* y, por ende, el mínimo global está *"hacia la izquierda"*, es decir, restando a x la suposición mínima (x - *mínimo local*).

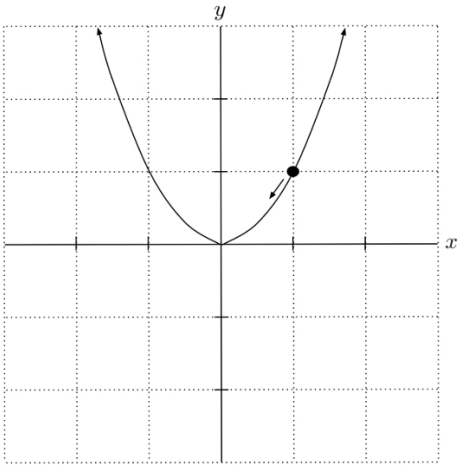
b) En caso de que la derivada de la función sea negativa, se interpreta que la función está inclinada *"hacia abajo"*; por tanto, el mínimo global está *"hacia la derecha"*, es decir, sumando a x la suposición mínima (x + *mínimo local*).

Figura 2.7: Descenso Local vs Descenso Global



Fuente: Gráfica extraída de [24]

Figura 2.8: Mínimo global de una derivada simple



Fuente: Gráfica extraída de [24]

Inicialmente, se calcula la derivada del error global respecto al peso w_{kj}, tal como se muestra a continuación:

$$\frac{\partial E}{\partial w_{kj}} = \left[\frac{\partial E}{\partial o_{pk}}\right] \cdot \frac{\partial o_{pk}}{\partial net_{pk}} \cdot \frac{\partial net_{pk}}{\partial w_{kj}}$$

$$\frac{\partial E_p}{\partial w_{kj}} = \left[\frac{1}{2}\sum_{k=1}^{M}(y_{pk} - o_{pk})^2\right] \cdot \frac{\partial o_{pk}}{\partial net_{pk}} \cdot \frac{\partial net_{pk}}{\partial w_{kj}}$$

$$\frac{\partial E_p}{\partial w_{kj}} = -\left[y_{pk} - o_{pk}\right] \cdot \frac{\partial o_{pk}}{\partial net_{pk}} \cdot \frac{\partial net_{pk}}{\partial w_{kj}}$$

De la anterior expresión se infiere que la función de activación $f_k(x)$ contenida dentro del término o_{pk} debe ser derivable, representada como $f'_k(net_{pk})$.

a) En caso de ser una función lineal (Ecuación 2.5), tanto en las capas ocultas como de salida, la derivada es:

$$f_k(x) = net_{pk}$$

$$f'_k(net_{pk}) = 1$$

b) En caso de ser una función sigmoidal (Ecuación 2.7), siendo que la variable x representa net_{pk}, tanto en las capas ocultas como de salida, la derivada es la que se presenta a continuación (consultar Anexo A.1 para más detalles):

$$f_k(x) = \frac{1}{1 + e^{-x}}$$

$$f'_k(net_{pk}) = o_{pk}\left(1 - o_{pk}\right)$$

Una vez que se obtiene el término de la derivada de la entrada neta (net_{pk}) a la neurona k correspondiente al peso w_{kj}, a partir de la Ecuación 2.25, se desarrolla lo siguiente:

$$\frac{\partial net_{pk}}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}}\left[\sum_{j=1}^{L} w_{kj} \cdot i_{pj} + \theta_k\right] = i_{pj}$$

Con base en el anterior resultado, es posible calcular el gradiente correspondiente a la derivada del error global respecto al peso w_{kj}.

$$\frac{\partial E_p}{\partial w_{kj}} = -(y_{pk} - o_{pk}) \cdot f'_k(net_{pk}) \cdot i_{pj} \tag{2.29}$$

El principio del gradiente descendente implica buscar gradualmente un punto de error mínimo, siempre orientado por la cifra negativa de la derivada del error global. Por ende, la expresión que se emplea se ajusta a la cifra negativa del gradiente de error ($-\nabla E_p$); buscando así reducir el error al modificar cada peso entre la capa de salida k y la capa oculta j en el sentido inverso, tal como se expresa en la siguiente ecuación:

$$-\nabla E_p = -\frac{\partial E_p}{\partial w_{kj}} = (y_{pk} - o_{pk}) \cdot f'_k(net_{pk}) \cdot i_{pj} \tag{2.30}$$

Se emplea un algoritmo recursivo desde las neuronas de salida hasta las neuronas de entrada, actualizando los pesos en sentido inverso (consulte el Anexo B.2.4 para más detalles), tal como se demuestra de la siguiente forma:

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t)$$

$$w_{kj}(t+1) = w_{kj}(t) + \alpha(-\nabla E_p)$$

$$w_{kj}(t+1) = w_{kj}(t) + \alpha(y_{pk} - o_{pk}) \cdot f'_k(net_{pk}) \cdot i_{pj}$$

Donde:

- $w_{kj}(t+1)$ es el valor actualizado del peso sináptico.
- $w_{kj}(t)$ es el valor actual del peso sináptico.
- $\Delta w_{kj}(t)$ es la variación del peso sináptico.
- α es la tasa o velocidad de aprendizaje.

De forma consecutiva, dependiendo del tipo de función de activación, el valor del gradiente cambia si:

a) Si la función es lineal

$$w_{kj}(t+1) = w_{kj}(t) + \alpha(y_{pk} - o_{pk}) \cdot i_{pj} \tag{2.31}$$

b) Si la función es sigmoide:

$$w_{kj}(t+1) = w_{kj}(t) + \alpha(y_{pk} - o_{pk}) \cdot o_{pk}(1 - o_{pk}) \cdot i_{pj} \tag{2.32}$$

2.5.2.2. Actualización de pesos en sentido inverso

Para la capa oculta no es posible calcular el error directamente, ya que no se conocen las salidas deseadas de esta capa. Sin embargo, si usamos el concepto de retropropagación, observamos que existe una manera de estimar este error de la capa oculta partiendo del error en la capa de salida [25].

Se aplica la regla de la cadena de forma subsecuente para el cálculo de la derivada interna del error (E_p) con respecto a los pesos w_{ji}, de manera similar a lo presentado en la Ecuación 2.29:

$$\frac{\partial E_p}{\partial w_{ji}} = \left[\frac{\partial E_p}{\partial o_{pk}} \cdot \frac{\partial o_{pk}}{\partial net_{pk}} \cdot \frac{\partial net_{pk}}{\partial i_{pj}}\right] \cdot \frac{\partial i_{pj}}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial w_{ji}}$$

$$\frac{\partial E_p}{\partial w_{ji}} = \left[\frac{\partial}{\partial o_{pk}}\left(\frac{1}{2}\sum_{k=1}^{M}(y_{pk} - o_{pk})^2\right) \cdot \frac{\partial}{\partial net_{pk}}\left(f_k(net_{pk})\right) \cdot \frac{\partial}{\partial i_{pj}}\left(w_{kj} \cdot i_{pj}\right)\right] \cdot \frac{\partial i_{pj}}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial w_{ji}}$$

$$\frac{\partial E_p}{\partial w_{ji}} = \left[-\sum_{k=1}^{M}(y_{pk} - o_{pk}) \cdot f'_k(net_{pk}) \cdot w_{kj}\right] \cdot \frac{\partial}{\partial net_{pj}}(f_j(net_{pj})) \cdot \frac{\partial}{\partial w_{ji}}(w_{ji} \cdot x_{pi})$$

$$\frac{\partial E_p}{\partial w_{ji}} = \left[-\sum_{k=1}^{M}(y_{pk} - o_{pk}) \cdot f'_k(net_{pk}) \cdot w_{kj}\right] \cdot f'_j(net_{pj}) \cdot x_{pi}$$

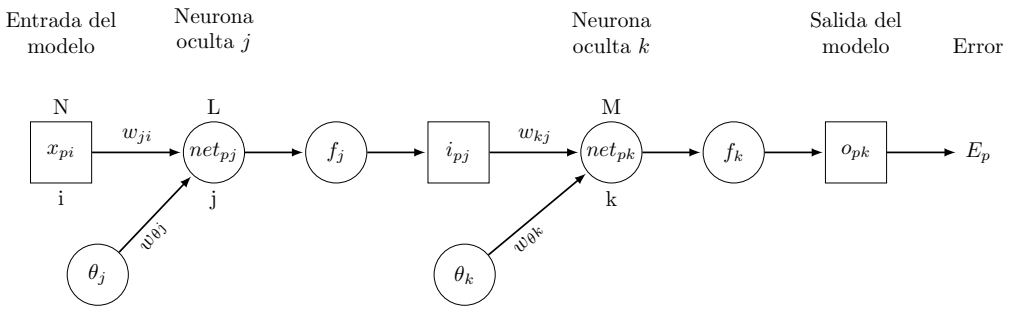
Basado en el proceso anterior demostrado en la Ecuación 2.30, se obtiene la ecuación de ajuste de peso (Δw_{ji}) para los pesos asociados a cada enlace de conexión entre la capa de entrada i y la capa oculta j, tal como se describe a continuación:

$$\Delta w_{ji}(t) = \alpha \left(-\sum_{k=1}^{M} (y_{pk} - o_{pk}) \cdot f'_k(net_{pk}) \cdot w_{kj} \cdot f'_j(net_{pj}) \cdot x_{pi} \right) \tag{2.33}$$

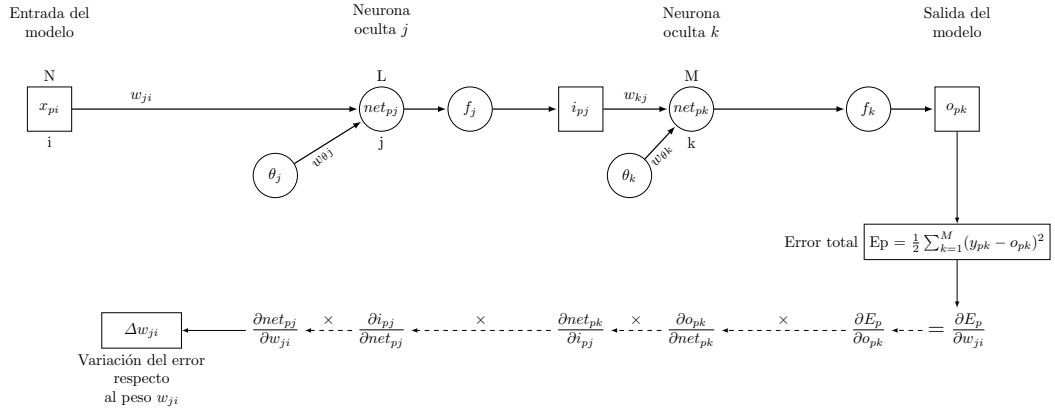
Al sustituir la variación de los pesos (obtenida en la Ecuación 2.33) en la expresión del gradiente, obtenemos la que nos permite actualizar los pesos de la capa oculta.

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t)$$
$$w_{ji}(t+1) = w_{ji}(t) + \alpha \left(-\frac{\partial E_p}{\partial w_{ji}} \right)$$
$$w_{ji}(t+1) = w_{ji}(t) + \alpha \left(-\sum_{k=1}^{M} (y_{pk} - o_{pk}) \cdot f'_k(net_{pk}) \cdot w_{kj} \cdot f'_j(net_{pj}) \cdot x_{pi} \right)$$

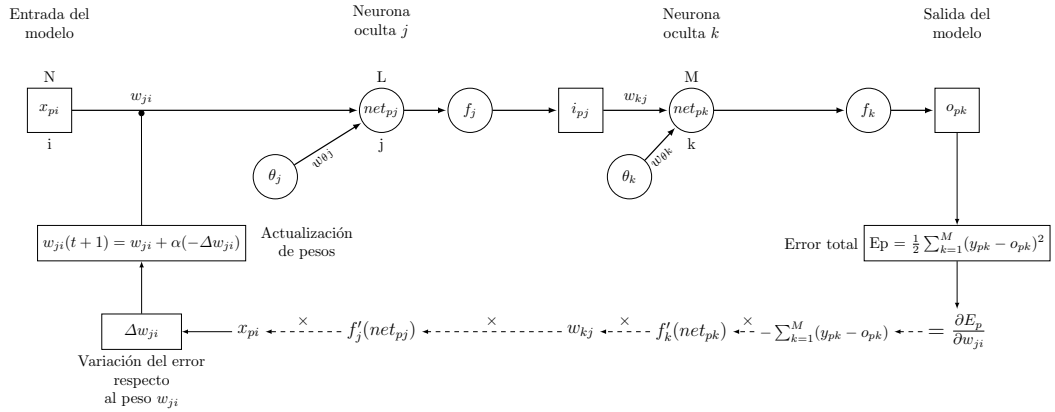
Figura 2.9: Fases del Backpropagation



(a) Fase de funcionamiento



(b) Fase de entrenamiento



(c) Actualización de los pesos hacia atrás

Fuente: Elaboración propia

El desafío que enfrentan las redes neuronales tradicionales (ANN) es una desventaja inherente a retener datos secuenciales, planteando limitaciones en tareas como la previsión de series temporales, procesamiento de lenguaje natural y otras tareas que requieren de una mayor memoria.

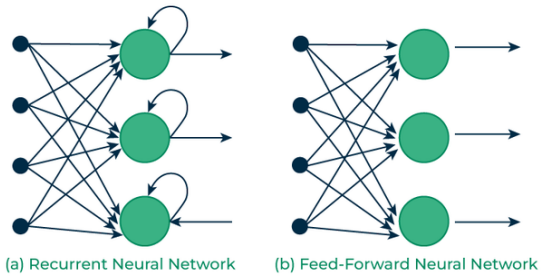
Las Redes Neuronales Recurrentes (por sus siglas RNN) son óptimas para datos secuenciales, donde cada neurona tiene conexiones que se entrelazan entre sí, lo que permite una mayor retención de la información.

Asimismo, las redes de tipo LSTM son una extensión de las RNN, ofreciendo una mayor retención y aprendizaje en datos temporales más largos, a diferencia de sus predecesores. A continuación se explicarán ambas arquitecturas.

2.6. Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes (RNN) incorporan ciclos que facilitan la retroalimentación de la información de los pasos previos a la red. Esta retroalimentación posibilita que las RNN recuerden entradas previas, lo que las convierte en perfectas para tareas donde el contexto es relevante.

Figura 2.10: Redes recurrentes vs Redes Feedforward



Fuente: Ilustración extraída de [26]

Por lo general, para llevar a cabo un análisis compartido de las redes neuronales recurrentes, se establece el término de *"celda"*. Una celda es simplemente una operación que contiene dos entradas y dos salidas, que se va formando a medida que la operación se aplica a los distintos valores de la secuencia.

Los puntos de entrada de la celda se relacionan, en un aspecto, con los valores de la secuencia para cada registro y, en otro aspecto, con la condición de la red neuronal en el paso previo. Los puntos de salida generalmente representan el estado de la red en un paso específico donde se lleva a cabo la celda y la reacción de la red a ese paso.

Según el problema a solucionar, la red podrá manejar diversos tipos de secuencias de entrada y salida, de las cuales se identifican cuatro topologías fundamentales:

- One to One RNN: En esta configuración, existe una única entrada y una única salida. Normalmente se emplea para labores de clasificación básicas donde los puntos de datos de entrada no están vinculados a los elementos previos.

- RNN de uno a muchos: En esta distribución, la red gestiona una sola entrada para producir múltiples salidas a lo largo del tiempo. Esta disposición es idónea cuando un solo elemento de entrada requiere generar una serie de pronósticos.

  Por ejemplo, para la tarea de subtitulado de imágenes, con una sola imagen como entrada, el modelo predice una secuencia de palabras como subtítulo.

- RNN de muchos a uno: Esta estructura es óptima para recepción múltiple de entradas y una única salida. Este tipo de configuración resulta apropiado para tareas donde se requiere el contexto global de la secuencia inicial para realizar una proyección.

  En el análisis de sentimiento, el modelo recibe una secuencia de palabras (una

oración) y produce una única salida, que es el sentimiento de la oración (positivo, negativo o neutro).

- RNN de muchos a muchos: Esta distribución de celdas procesa una serie de entradas y produce una serie de salidas, arquitectura ideal para tareas cuyas secuencias de entrada y salida necesitan estar en sincronía con el tiempo, frecuentemente en una distribución de uno a uno o de varios a varios.

  En la tarea de traducción de idiomas, se da una secuencia de palabras en un idioma como entrada y se genera una secuencia correspondiente en otro idioma como salida.

2.6.1. Limitaciones de las RNN

Si bien las RNN poseen la cualidad de la retención de la información, su memoria interna presenta un problema conocido como *"memoria a corto plazo"* (short-term memory en inglés), el cual se relaciona con el fenómeno del desvanecimiento del gradiente. Tal como se ha mencionado, las redes neuronales recurrentes analizan secuencias de información para posteriormente incorporar en cada paso el resultado del análisis del elemento anterior. Acumulando hasta el paso final el procesamiento de todos los elementos de la secuencia. Por ende, al incrementar los elementos en la secuencia, la red se enfrenta a dificultades en recordar los análisis de los elementos iniciales. Este problema se debe a que el gradiente, necesario para ajustar los pesos de la red, se reduce exponencialmente al alejarse de los elementos iniciales de la secuencia.

2.7. Long Short-Term Memory (LSTM)

La solución más ampliamente adoptada para evitar el desvanecimiento del gradiente consiste en implementar unidades o celdas especiales cuya función corresponde a incorporar una conexión directa entre el estado anterior y la salida de la celda. Esta conexión directa, mayormente conocida como *"conexión de identidad"*, permite que el gradiente se propague en lo largo de la red de forma estable, reduciendo el riesgo de su desvanecimiento o explotación.

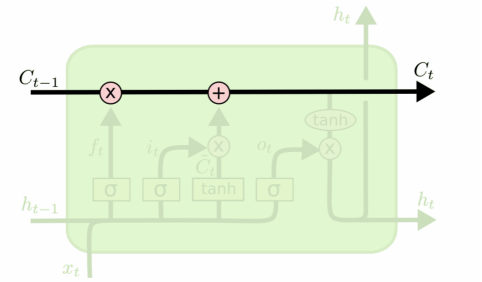
Para regular el flujo de información a través de este canal, se emplean mecanismos de compuertas lógicas; dichas compuertas son incorporadas mediante capas completamente conectadas. Estas compuertas funcionan de forma dinámica controlando qué información se retiene, se descarta o se transmite a lo largo de la secuencia, permitiendo el manejo eficiente de las dependencias temporales. Esta arquitectura avanzada fue propuesta en 1997 por Hochreiter y Schmidhuber, las cuales hasta hoy en día han demostrado ser altamente efectivas para modelar secuencias largas y complejas.

2.7.1. Estructura de una celda LSTM

El secreto detrás del LSTM se establece en controlar qué información se guarda en el *"estado"* (denominado también como *"memoria"*), el cual recorre la parte superior de la celda, y qué información se produce a la salida de la red.

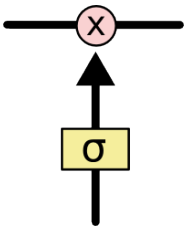
El LSTM tiene la capacidad de eliminar o agregar información al estado de la celda, cuidadosamente regulada por estructuras llamadas compuertas. Estas compuertas son una forma de dejar pasar la información de forma opcional. Están compuestos por una capa sigmoide y una operación de multiplicación puntual.

Figura 2.11: Estado de la celda del LSTM



Fuente: Ilustración extraída de [27]

Figura 2.12: Mecanismo de compuerta del LSTM



Fuente: Ilustración extraída de [27]

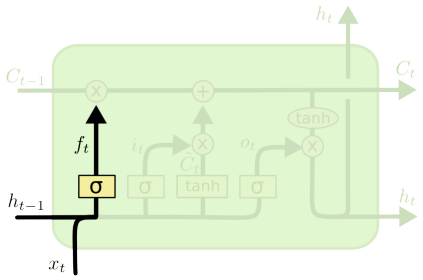
2.7.1.1. Compuerta de olvido

También conocida como *forget gates* en inglés, esta compuerta decide qué información se desecha del estado de la celda. Esta decisión la toma una capa sigmoide llamada *"capa de puerta de olvida"*. Los valores h_{t-1} y x_t se utilizan para generar un número entre 0 y 1 para cada número en el estado de la celda C_{t-1}. Donde un 1 representa *mantener información*, mientras que un 0 representa *deshacer información*.

La compuerta de olvido genera un vector f_t, que define la proporción de información del estado anterior C_{t-1} que se conserva o se descarta. Este vector tiene la misma longitud que los vectores de estado y se calcula mediante la siguiente fórmula:

$$f_t = \sigma\left(W_f, [h_{t-1} \cdot x_t] + b_f\right) \tag{2.34}$$

Figura 2.13: Compuerta *forget* del LSTM



Fuente: Ilustración extraída de [27]

Donde:

- h_{t-1} es la salida de la anterior celda LSTM.
- x_t es la nueva entrada a la actual celda LSTM.
- b_f es un vector de sesgos.
- W_f es una matriz de pesos aplicada a la concatenación de h_{t-1} con x_t, la cual se encarga de combinar la información de la salida anterior y la nueva entrada.

2.7.1.2. Compuerta de entrada

Esta compuerta controla qué información se añade a la memoria de la red. Se compone de dos partes.

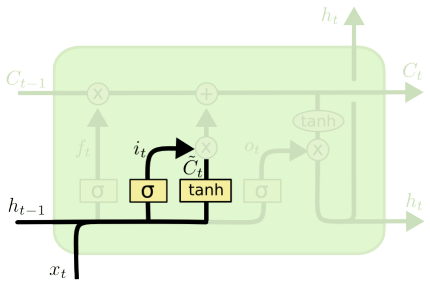
1. Una capa sigmoide llamada *"capa de puerta de entrada"*, denominada i_t, la cual controla cuánta información nueva se introduce en el estado de la celda.
2. Una capa tanh crea un vector de nuevos valores candidatos $\widetilde{C}_t$, el cual refiere a una versión ya procesada de la entrada que se podría agregar al estado de la celda.

Consecuentemente, se unen ambos vectores para crear la actualización del estado. Las ecuaciones que rigen ambas partes son las siguientes:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2.35a}$$

$$\widetilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{2.35b}$$

Figura 2.14: Compuerta *input* del LSTM

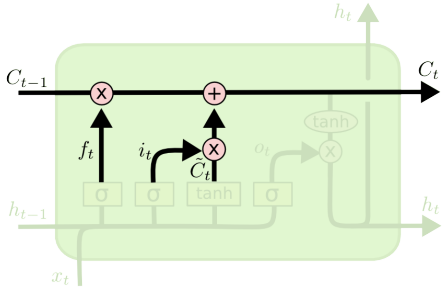


Fuente: Ilustración extraída de [27]

A fin de actualizar el estado de la celda anterior C_{t-1}, en el nuevo estado de celda C_t, se multiplica el vector f_t por el viejo estado C_{t-1}, a fin de desechar la información pasada, y se suma el producto de i_t por $\widetilde{C}_t$. El resultado obtenido son los nuevos valores candidatos, escalados por la cantidad que decidimos actualizar cada valor de estado. La ecuación que representa lo anterior es la siguiente (donde el operador $\odot$ denota multiplicación vectorial en este contexto):

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{2.36}$$

Figura 2.15: Actualización del estado del LSTM



Fuente: Ilustración extraída de [27]

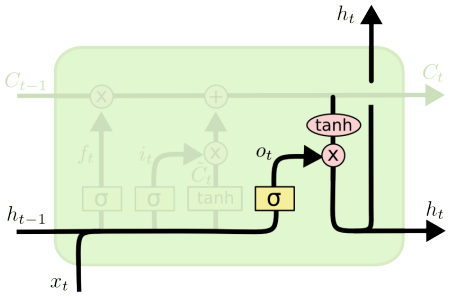
2.7.1.3. Compuerta de salida

A partir del nuevo estado calculado C_t, se debe decidir su salida h_t. Para ello se observa el valor de entrada x_t a la celda junto con la salida anterior de la red h_{t-1} y se ejecuta una capa sigmoide (Ecuación 2.37a) a fin de decidir qué partes del nuevo estado de la celda LSTM se va a generar. Posteriormente, mediante una capa Tanh, se somete el estado (Ecuación 2.37b), a fin de categorizar los valores en un rango de entre -1 y 1; finalmente, se multiplica la salida de la puerta sigmoide, de modo que solo obtenemos las partes que decidimos.

$$o_t = \sigma\ (W_o\,[h_{t-1}, x_t] + b_o) \tag{2.37a}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{2.37b}$$

Figura 2.16: Compuerta *output* del LSTM



Fuente: Ilustración extraída de [27]

Capítulo 3

Rodamientos y señales

3.1. Rodamientos

Los rodamientos, también conocidos comúnmente como *"cojinetes"*, son elementos mecánicos que permiten el movimiento de rotación y traslación entre dos piezas. En algunos tipos es posible transmitir fuerzas radiales y axiales (presentes en los rodamientos rotativos), así como fuerzas y momentos transversales a la dirección del movimiento (presentes en los rodamientos lisos). La importancia de la presencia de un rodamiento en máquinas se divide en tres puntos:

- Reducir la fricción.
- Transportar carga.
- Guiar piezas.

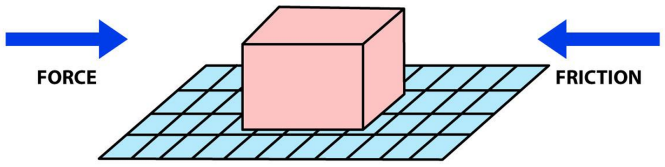
3.1.1. Fricción y carga

La fricción se puede definir como la *"resistencia al movimiento entre dos o más cuerpos"*. Esta puede presentarse en distintas formas, pero las más importantes son:

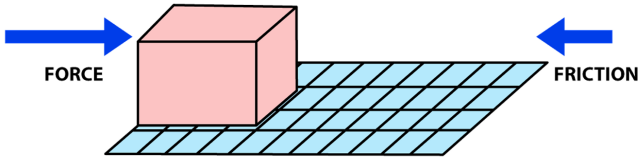
- Fricción estática: Fricción entre dos o más cuerpos sólidos que no se desplazan entre sí. Donde toda fuerza ejercida sobre un cuerpo debe sobrepasar la fricción estática antes de empezar su movimiento.
- Fricción cinética o deslizante: Fricción entre dos cuerpos producida al mover un sólido respecto al otro y que están en contacto entre sí.
- Fricción rodante: Fricción producida por un cuerpo al rodar sobre otro. Generalmente, este es menor que la fricción deslizante.

Figura 3.1: Tipos de fricción presentes en rodamientos

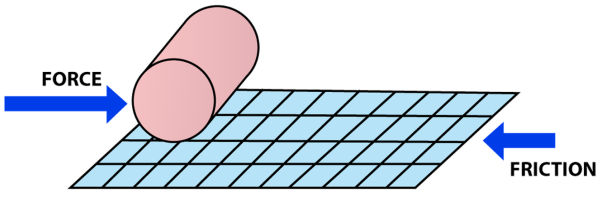
(a) Fricción Estática



(b) Fricción Deslizante



(c) Fricción Rodante



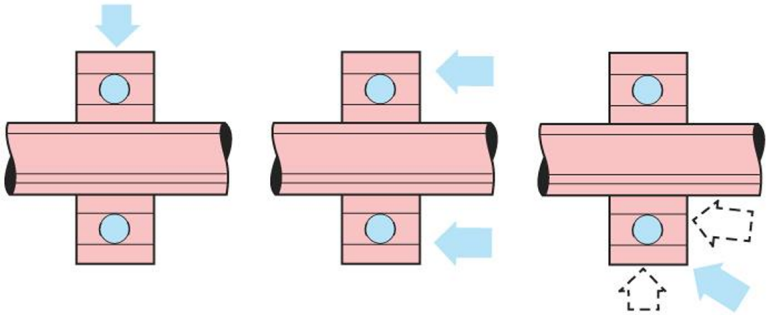
Fuente: Ilustración extraída de [28]

Por otro lado, los rodamientos se deberán seleccionar en función de las cargas que pueden soportar y las condiciones de servicio. Un dimensionamiento y selección incorrectos pueden provocar la falla del mismo. Las cargas en los rodamientos se clasifican de la siguiente forma:

- Cargas axiales: Cargas que se aplican de forma paralela al eje. También se les conoce como *"cargas de empuje"*.
- Cargas radiales: Cargas aplicadas en forma perpendicular al eje, es decir, a 90°.
- Cargas combinadas: Cargas que son la suma de cargas axiales y radiales. También son denominadas como *"cargas angulares"*.

Figura 3.2: Tipos de cargas presentes en rodamientos

a) Carga Radial. b) Carga Axial. c) Carga Mixta o combinada.



Fuente: Ilustración extraída de [28].

3.1.2. Componentes generales

En la Tabla 3.1 se desglosan los componentes presentes mayormente en los rodamientos, con ciertas variaciones dependiendo del tipo de rodamiento. En ella se explica los materiales que se utilizan para fabricar, su función dentro del rodamiento, los tipos de componentes y su aplicación.

Tabla 3.1: Componentes generales en rodamientos

Elementos generales	Materiales	Función	Tipos	Aplicación
Elementos rodantes	Acero	Elementos que permiten al rodamiento girar libremente entre las pistas interiores y exteriores	Bolas	Elementos rodantes son el tipo más común entre los rodamientos. Utilizan esferas que distribuyen la carga de manera uniforme, reduciendo la fricción y permitiendo una rotación suave y de baja fricción.
	Cerámica		Rodillo cilíndrico	Elemento rodante óptimo para cargas radiales o axiales más altas.
			Cónico	Diseñados para soportar cargas axiales en una dirección.
			Esférico	Adecuados para cargas axiales pesadas y cargas radiales moderadas. Son ideales para adaptarse en casos de delineación.
			Agujas	Se emplean para aplicaciones con espacio limitado y con altas cargas radiales.
Anillo interior	Acero	Garantiza una alineación adecuada y un funcionamiento suave del rodamiento.	Rodillos	La pista de rodadura es plana o cónica con una brida que mantiene los rodillos en su sitio
			Bolas	Se corta una ranura en su circunferencia exterior.
Anillo exterior	Acero	Protege los elementos rodantes y mantiene su posición dentro del rodamiento. Diseñado para soportar cargas estáticas y dinámicas.	Rodillos	La pista de rodadura exterior es plana, esférica o cónica con una brida que mantiene los rodillos en su sitio.
			Bolas	Se corta una ranura a lo largo de la circunferencia interior de la pista de rodadura para que las bolas se mantengan en su sitio.
Jaulas	Acero prensado	Las jaulas desempeñan el papel de mantener un espaciado adecuado entre bolas, mejorar la distribución de la carga, reducir la fricción y prolongar la vida útil general del rodamiento	Chapa metálica	Liviano, con alta resistencia y de baja fricción con un revestimiento para reducir el contacto deslizante con los elementos rodantes.
	Latón prensado		Acero tipo pasador	Perforados en toda su longitud del eje central. Permite colocar un rodillo adicional dentro del rodamiento.
	Latón mecaizado		Latón mecanizado	Amortiguan las vibraciones de los rodillos. Ideales para rodamientos de gran tamaño y en elementos rodantes de alta aceleración.
	Poliamida		Poliamida	Empleados para velocidades más altas dada su baja fricción por deslizamiento con elementos rodantes. Adecuadas para empuje inverso como en bombas debido a alta elasticidad.
Sellos	Caucho con refuerzo como NBR o FKM	Este elemento mantiene los contaminantes fuera del conjunto de elementos rodantes, de forma simultánea, evita que la lubricación salga del área de la pista.	Sellos de goma	Son eficaces para evitar la entrada de suciedad, polvo y agua, proporcionando una mejor protección en comparación con los protectores metálicos.
			Sellos de contacto	Similares a los de goma con la diferencia que tienen contacto directo con la pista interior del rodamiento, creando una barrera más eficaz contra los contaminantes.
			Sellos de fieltro	Compuestos de material de fieltro comprimido y se utilizan donde se requiere una protección moderada contra contaminantes.
			Sellos laberínticos	Consisten en múltiples barreras sin contacto que crean un acceso difícil para los contaminantes, al mismo tiempo, permiten cierto flujo de aire hacia el conjunto de elementos rodantes.
			Sellos dobles	Ofrece doble protección mediante dos barreras de sellos. Ideales para aplicaciones en entornos hostiles.

Fuente: Elaboración propia

3.1.3. Tipos de rodamientos

La selección de un sistema de apoyo para ejes en movimiento se basa en gran medida en la fricción generada por la interacción entre las superficies y del tipo de carga que debe soportar el mecanismo. Cuando una superficie se desplaza sobre otra, la fricción deslizante puede generar un desgaste significativo y una disminución de la eficiencia, particularmente en aplicaciones con altas velocidades o cargas variables. Por otro lado, si el movimiento se realiza a través de una rodadura, la fricción disminuye considerablemente, facilitando un funcionamiento eficaz y con menor uso de energía.

En función de la orientación y volumen de la carga ejercida, se requiere de un tipo específico de soporte que garantice estabilidad y longevidad. A fin de satisfacer las demandas, se exige una mayor resistencia hacia cargas radiales, cargas axiales o cargas mixtas; los sistemas de soporte se clasifican en dos grandes categorías.

Existen dos tipos de rodamientos:

a) **Rodamientos lisos:** Elemento con forma cilíndrica cuyo diseño permite encajar firmemente en la carcasa y en el eje. Generalmente, se emplean para guiar árboles y ejes, dotando de libertad de giro a las piezas mientras soportan cargas que actúan sobre estos. Son construidos con un material con bajo coeficiente de fricción, resultando en bajas caídas de presión y baja resistencia en superficies deslizantes (fricción por deslizamiento).

   Las ventajas que posee incluyen:

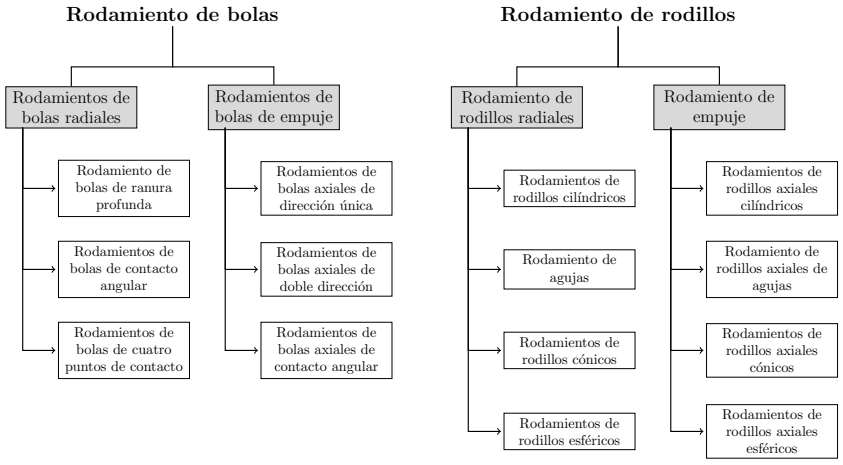
   - Diámetro exterior más pequeño.
   - Funcionamiento silencioso.
   - Movimiento oscilatorio duradero (movimiento repetitivo de ida y vuelta).
   - Bajo costo.

b) **Rodamientos de elementos rodantes:** Este tipo de rodamientos, si bien son de mayor complejidad que sus semejantes lisos y de mayor consideración para cada aplicación determinada, son los que satisfacen una mayor variedad de requisitos mecánicos complejos.

El tipo de rodamiento más común soporta un eje giratorio que resiste una combinación de cargas radiales y axiales. Si bien aumentar el tamaño y la cantidad de elementos rodantes incrementa la capacidad de carga del rodamiento, cambiar la forma del elemento rodante permite lograr el mismo efecto. La forma del elemento rodante varía desde bolas esféricas hasta cilindros y secciones cónicas. Los rodamientos de elementos rodantes se clasifican en dos clases, tal como se muestra en la Tabla 3.2, considerando los siguientes puntos:

- La dirección de la carga a soportar.
- El tipo de carga (estable o fluctuante).
- El tipo de elemento rodante (bolas o rodillos).

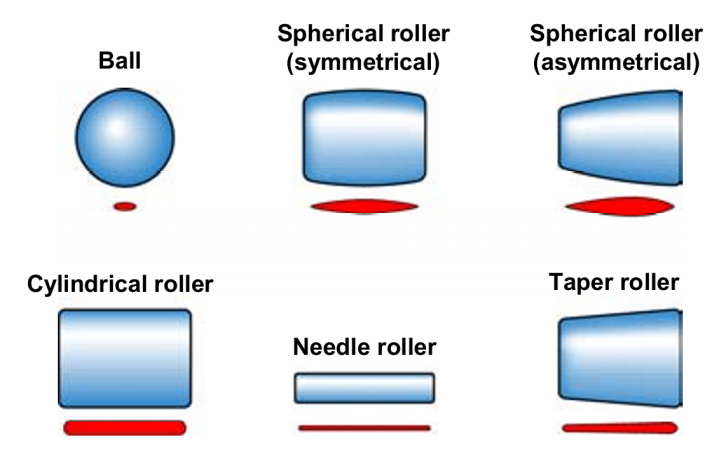
Figura 3.3: Clasificación de rodamientos de elementos rodantes



Fuente: Diagrama adaptado de [29]

El elemento rodante se encarga de llevar la carga mediante un *contacto puntual*, área por donde se transmite la carga desde el elemento rodante hasta la pista de rodadura del anillo. El contacto puntual varía dependiendo de la forma del elemento rodante, al igual que sus beneficios e inconvenientes.

Figura 3.4: Tipos de elementos rodantes y sus contactos puntuales



Fuente: Ilustración extraída de [30]

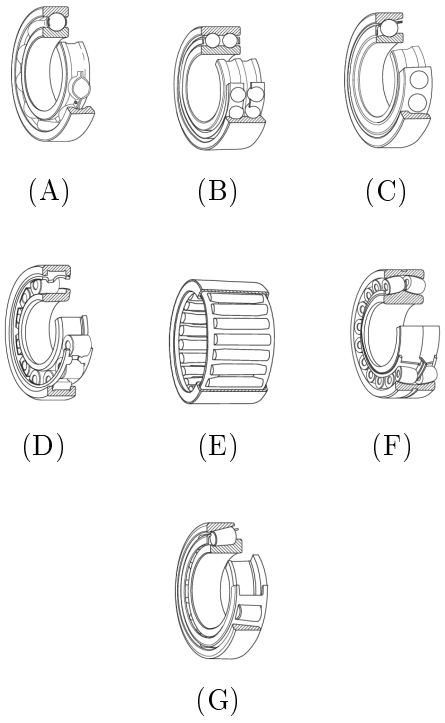
Aunque los elementos esféricos tienen la capacidad de trabajar a velocidades superiores y adaptarse a desalineaciones más bajas, su capacidad para soportar cargas es limitada. Por otro lado, los rodamientos de contacto lineal distribuyen la carga a lo largo de una superficie mayor entre las pistas de la rodadura del anillo, ofreciendo mayor resistencia y dispersión hacia una carga superior. El principal inconveniente de los elementos de contacto lineal es la reducción de la velocidad provocada por el incremento de la fricción y la producción de calor. La Tabla 3.2 resume sus diferencias.

Tabla 3.2: Diferencias entre los rodamientos de elementos rodantes

Tipos de rodamiento	Capacidad de carga radial	Capacidad de carga axial	Capacidad de desalineación	Figura 3.5
Fila única	Bien	Justo	Justo	(A)
Doble fila de bolas de ranura [a]	Excelente	Bien	Justo	(B)
Contacto angular	Bien	Excelente	Pobre	(C)
Rodillo cilíndrico	Excelente	Pobre	Justo	(D)
Rodillo de agujas	Excelente	Pobre	Pobre	(E)
Rodillo esférico	Excelente	Regular/Bueno	Excelente	(F)
Rodillo cónico	Excelente	Excelente	Pobre	(G)

[a] Los rodamientos de doble fila se utilizan para soportar cargas que actúan en direcciones opuestas o para aumentar la superficie de contacto y la capacidad total de carga del rodamiento.
[1] Fuente: Extraído de [31].

Figura 3.5: Tipos de rodamientos de elementos rodantes



Fuente: Ilustración extraída de [32]

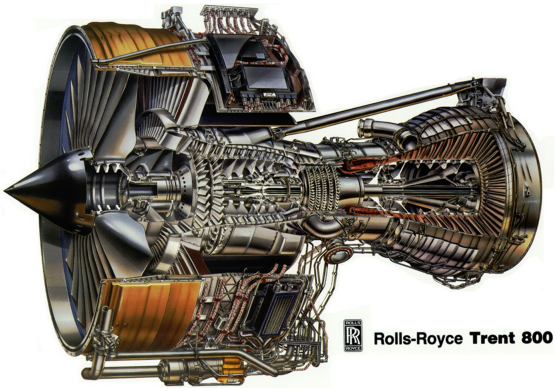
3.2. Fallas en rodamientos

La durabilidad de los rodamientos frente a la fatiga se establece por diversos factores, incluyendo las características del material, las propiedades del lubricante, la velocidad, la carga, el tamaño y la cantidad de elementos rodantes, entre otros.

Las teorías que predicen la duración de la vida por fatiga se formularon inicialmente en los años 40. A partir de ese momento, se han hecho avances notables en los materiales y en la elaboración de los rodamientos específicos, lo que resultó en un incremento notable de la durabilidad y la fiabilidad de los componentes rodantes.

Gran parte del trabajo de investigación y desarrollo que condujo a avances en la tecnología de los rodamientos fue impulsado por los requisitos de los motores de turbina de gas de aeronaves avanzados, donde ha habido una creciente necesidad de rodamientos con una vida útil más prolongada, mayor confiabilidad y mayor capacidad de velocidad.

Figura 3.6: Motor Rolls Royce Trent 800



Fuente: Ilustración extraída de [31]

En la literatura se han divulgado numerosas técnicas de procesamiento de señales de vibración para supervisar la condición de los rodamientos en toda la variedad de

maquinaria rotatoria.

La supervisión del estado puede segmentarse en tres áreas fundamentales:

- Identificación: En esta área determina que se ha producido una falla o alteración en el estado mecánico de la máquina.
- Diagnóstico: Durante esta etapa se precisa la ubicación y la naturaleza de la misma.
- Pronóstico: Finalmente, en esta etapa se realiza una estimación de la vida útil del rodamiento dañado.

Esta investigación se enfoca hacia el diagnóstico de fallas en rodamientos.

3.2.1. Vida nominal de un rodamiento

La vida nominal de un rodamiento, establecida por los estándares ISO y ABMA, se define como la duración real de un rodamiento bajo condiciones operativas antes de su fallo o eventual sustitución.

La norma ISO 281, actualizada en 2007, implementa técnicas avanzadas para estimar la vida útil de los rodamientos, considerando factores como la fatiga del material y un factor del efecto que ejerce la contaminación sólida con distintos sistemas de lubricación (grasa, circulación de aceite, baño de aceite y otros) sobre la vida de los rodamientos.

El método para calcular la vida nominal establecida bajo las normas ISO 281 y/o ABMA 9 y 11 está representado por la Ecuación 3.1:

$$L_{10h} = \frac{10^6}{60 \cdot n} \cdot \left[\frac{C}{P}\right]^P \tag{3.1}$$

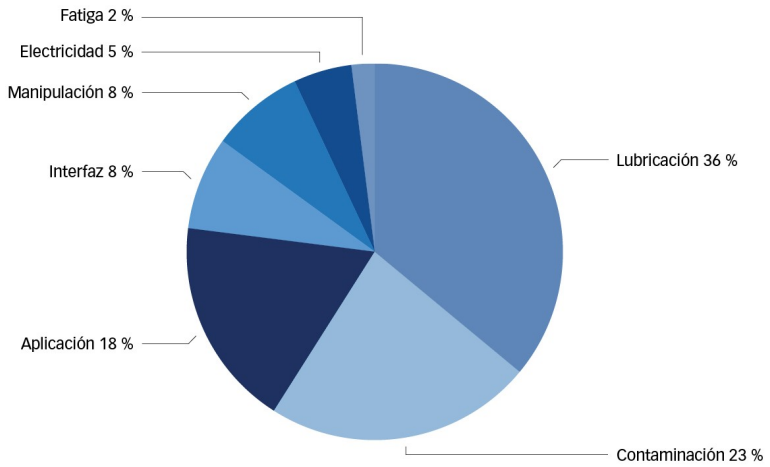
Donde:

- L_{10h} es la vida nominal en millones de revoluciones.

- C es la capacidad de carga dinámica del rodamiento.
- P es la carga dinámica equivalente del rodamiento.
- p es el exponente de la ecuación de la vida útil (3 para los rodamientos de bolas).
- n es la velocidad del rodamiento.

No obstante, según las estadísticas de fallas en rodamientos realizadas por el fabricante SKF, se describen en la Figura 3.7 las principales causas que afectan su condición funcional.

Figura 3.7: Causas comunes de daños en rodamientos



Fuente: Ilustración extraída de [33]

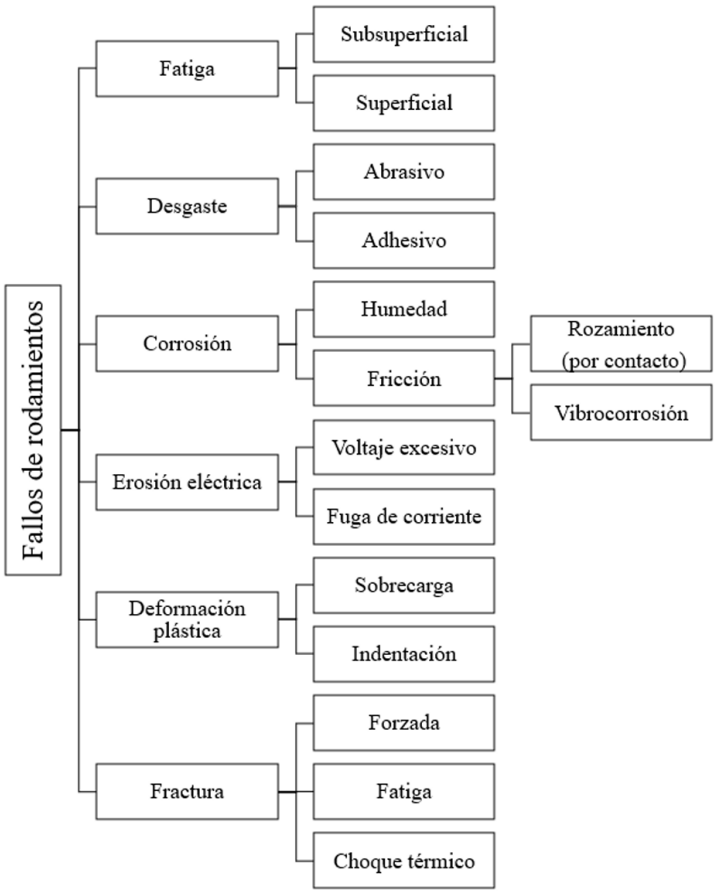
3.2.2. Fallas comunes en rodamientos

Para un análisis de fallas en rodamientos, es fundamental entender la clasificación básica de los diferentes tipos de fallas y sus respectivas fallas. La clasificación en fallas de rodamientos se basa en la norma ISO 15243:2017. Los tipos de falla se dividen en 6 clases

principales y subclases según la apariencia y las características visibles en las superficies de los componentes del rodamiento (véase la Figura 3.8).

Se denominan daños primarios a aquellos que provocan un tipo de fallo secundario o avanzado que implica descamación, grietas y daños en la jaula. Estos daños son el desgaste, indentación, deterioro superficial, corrosión y daños por corriente eléctrica. Un rodamiento defectuoso puede presentar con frecuencia una combinación de daños primarios y secundarios.

Figura 3.8: Clasificación de fallas de acuerdo a la ISO 15243:2017



Fuente: Diagrama extraído de [34]

Cada componente de un rodamiento presenta frecuencias características de fallo según el comportamiento de la señal. Inicialmente, se detectan solo picos de energía a altas frecuencias. A medida que la falla avanza, se destacan cambios alrededor de dichas frecuencias características. En etapas avanzadas, predominan frecuencias bajas excitadas y, finalmente, vibraciones aleatorias en gran parte del espectro.

3.2.2.1. Primera etapa: Detección incipiente

Se manifiestan picos de energía (Spike Energy), ondas de esfuerzo en frecuencias superiores a 2000 Hz, con amplitudes bajas y despreciables frente a las de la zona A (donde se ubican frecuencias operativas como la de giro o fallos como desbalance).

3.2.2.2. Segunda etapa: Modulación y desgaste inicial

Aparece excitación en la frecuencia natural del rodamiento, con bandas laterales que indican modulación de señales. Estas frecuencias se sitúan entre 500 Hz y 2000 Hz, junto a un incremento de Spike Energy. Es crítica: el rodamiento comienza a desgastarse.

3.2.2.3. Tercera etapa: Falla catastrófica

El desgaste es gradual pero severo. Emergen frecuencias naturales y sus armónicos. El daño es evidente, y el rodamiento debe reemplazarse tras un control de vibraciones.

Estas frecuencias están relacionadas con los elementos internos del rodamiento y su comportamiento durante el giro del eje. Para ello, se establece la siguiente nomenclatura. Donde:

- P_D es el diámetro de paso; se determina mediante la expresión $(D_1 + D_2)/2$.
- N_B es el número de elementos rodantes contenidos en el interior del rodamiento.
- B_D es el diámetro del elemento rodante.

- β es el ángulo de contacto del elemento rodante.
- D_1 es el diámetro del hombro del anillo exterior.
- D_2 es el diámetro del hombro del anillo interior.

Dichas frecuencias características, son las siguientes:

- **BPFO *(Ball Pass Frequency Outer)*:** Frecuencia asociada al paso de los elementos rodantes por un punto fijo de la pista exterior por cada revolución del eje. Su ecuación es la siguiente:

$$BPFO = RPM \cdot \frac{N_B}{2}\left(1 - \frac{B_D}{P_D}\cos(\beta)\right) \tag{3.2}$$

- **BPFI *(Ball Pass Frequency Inner)*:** Frecuencia correspondiente al paso de los elementos rodantes por un punto fijo de la pista interior en cada giro del eje. El cálculo de su frecuencia se determina a partir de la siguiente ecuación:

$$BPFI = RPM \cdot \frac{N_B}{2}\left(1 + \frac{B_D}{P_D}\cos(\beta)\right) \tag{3.3}$$

- **BSF *(Ball Spin Frequency)*:** Frecuencia de giro de cada elemento rodante (bola o rodillo) por cada vuelta del eje. Su frecuencia se rige por la ecuación a continuación:

$$BSF = RPM \cdot \frac{P_D}{B_D}\left[1 - \left(\frac{B_D}{P_D}\cos(\beta)\right)^2\right] \tag{3.4}$$

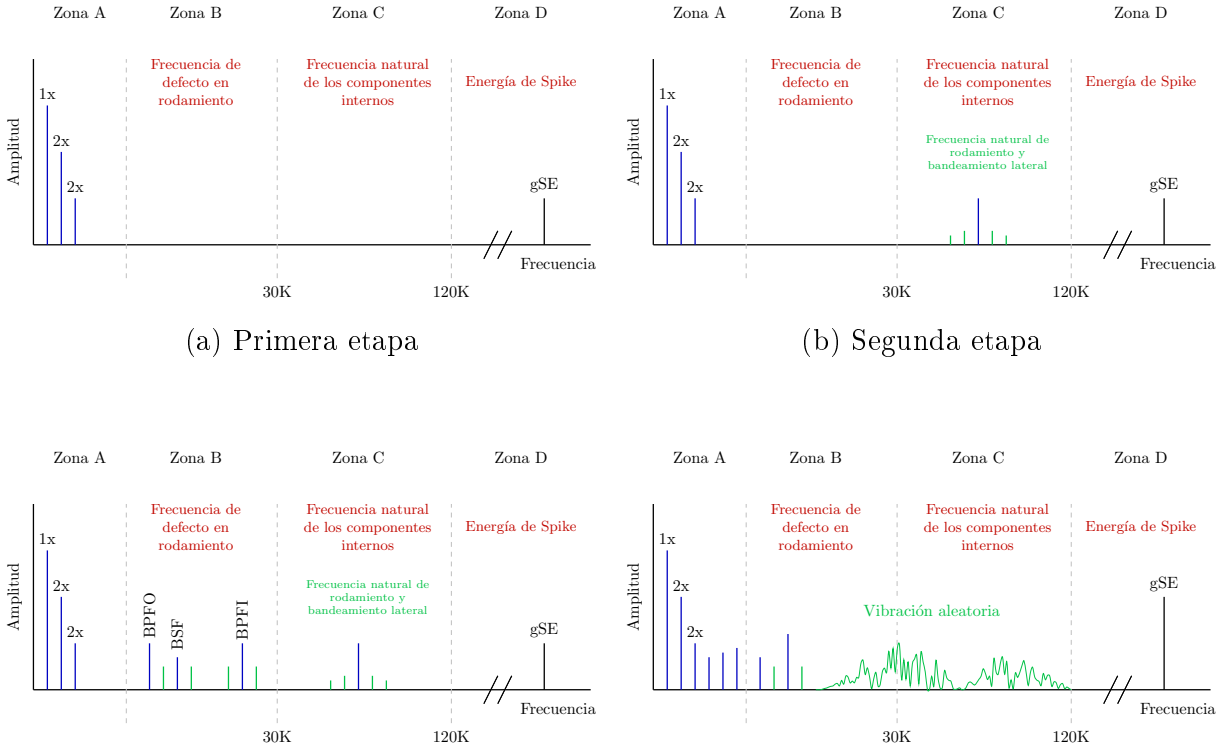
- **FTF *(Fundamental Train Frequency)*:** Frecuencia con la que gira la jaula del rodamiento por revolución del eje. Su frecuencia natural se expresa de la siguiente forma:

$$FTF = RPM \cdot \frac{1}{2}\left(1 - \frac{B_D}{P_D}\cos(\beta)\right) \tag{3.5}$$

3.2.2.4. Cuarta etapa: Falla generalizada

Las frecuencias características disminuyen, reemplazadas por vibración aleatoria y ruido ambiental. Esta fase debe evitarse: el esfuerzo excesivo daña otros componentes. El reemplazo del rodamiento ya no basta; se requiere revisión integral de la máquina.

Figura 3.9: Frecuencias en fallas de rodamientos



(a) Primera etapa

(b) Segunda etapa

(c) Tercera etapa

(d) Cuarta etapa

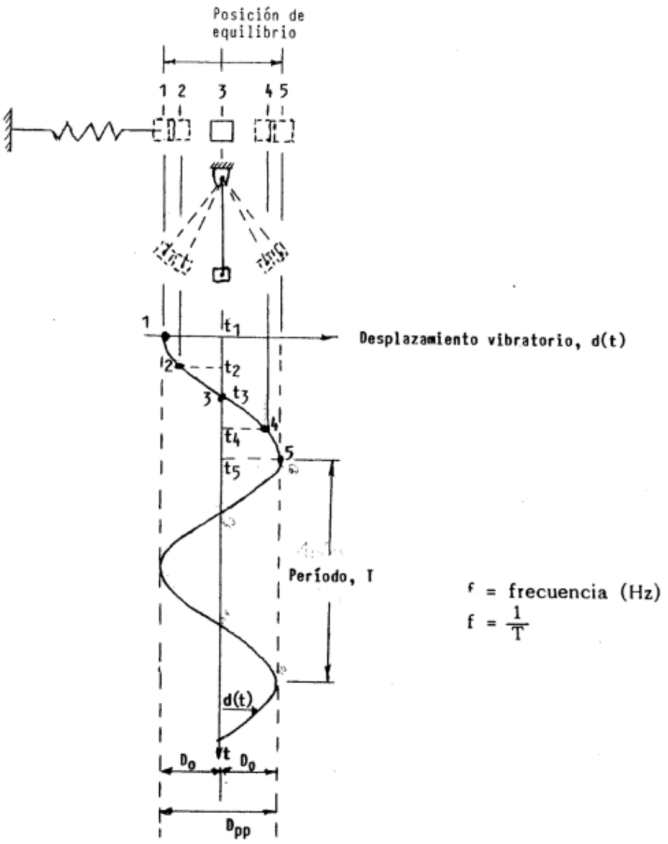
Fuente: Elaboración propia, basado en información de [34]

3.3. Vibrodiagnóstico

La vibración se define como *"el movimiento oscilatorio o repetitivo de un objeto alrededor de una posición de equilibrio"*. La vibración de un objeto puede ser causada

por una fuerza de excitación, la cual puede aplicarse de forma externa al objeto o interna al objeto. La posición de equilibrio se alcanza una vez que esta fuerza sea nula sobre el cuerpo. A este tipo de vibración se le conoce como *"vibración de cuerpo entero"*, lo que significa que todas las partes del cuerpo se desplazan en la misma dirección en cualquier instante.

Figura 3.10: Movimiento vibratorio en un cuerpo



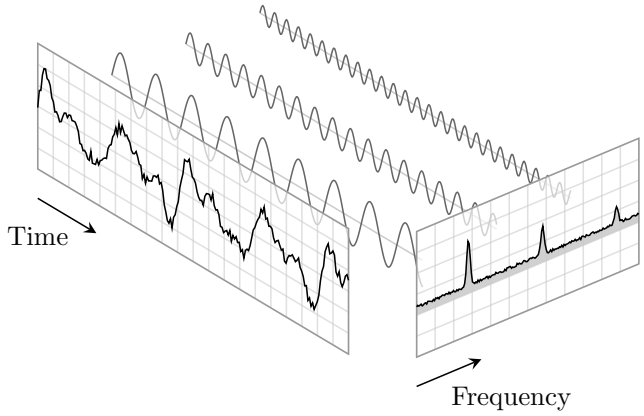
Fuente: Ilustración extraída de [35]

El movimiento vibratorio de un cuerpo entero se describe como una mezcla de movimientos individuales de seis categorías distintas. Estos van desde movimientos en las tres direcciones ortogonales a los ejes (X,Y,Z), así como de las rotaciones en torno a

los ejes (U,V,W) [36]. De esta forma, cualquier movimiento complejo que pueda realizar el cuerpo puede descomponerse en una mezcla de seis movimientos. Por lo tanto, se considera que un cuerpo puede tener seis grados de libertad.

Una *"señal compuesta"* se describe como la suma de varias señales sinusoidales, que pueden comprender desde información de cada uno de sus componentes internos, además de golpeteos externos y vibraciones aleatorias. Por tanto, una señal de vibración se especifica como *"la suma vectorial de la vibración de cada uno de sus elementos"*.

Figura 3.11: Descomposición de una señal compleja



Fuente: Ilustración extraída de [37]

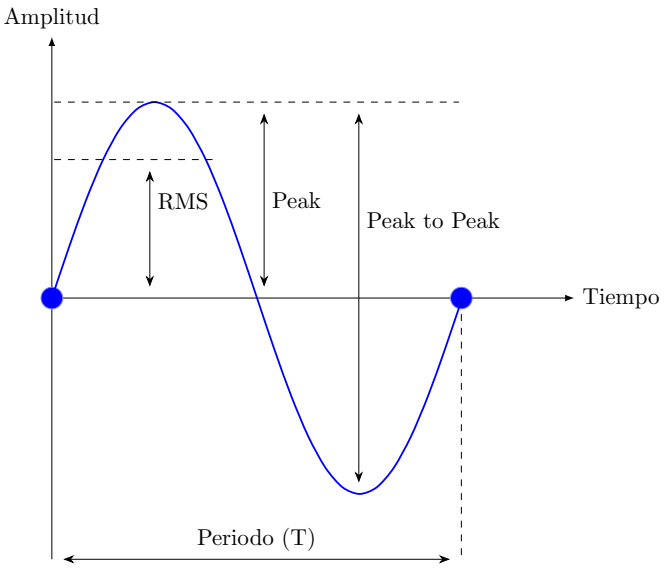
La vibración en los equipos rotatorios suele ser compleja, dado que se genera a partir de múltiples fuentes, dando lugar a una señal compuesta.

Para medir la vibración global, se utilizan tres magnitudes:

- **Valor pico (Peak):** Es el *"valor máximo alcanzado por la vibración durante un intervalo de tiempo T"*. Dicho valor sirve para describir vibraciones impulsivas o cuando se quiere evaluar la sobrecarga que generan las vibraciones del rotor en los descansos hidrodinámicos.

- **Valor pico a pico (Peak to Peak):** Se refiere a la *"máxima variación de la vibración durante un intervalo de tiempo T"*. Se encarga de medir los desplazamientos relativos.
- **Valor RMS (Root Mean Square):** Es aquella *"magnitud que estima la energía de la vibración en un período de tiempo T"*. Entrega el historial de la vibración.

Figura 3.12: Valores de medida en vibración



Fuente: Elaboración propia

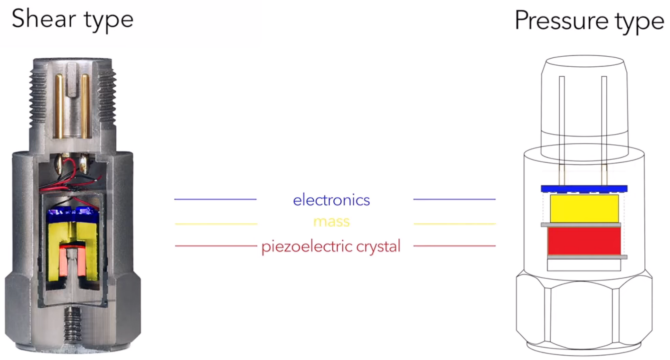
La aceleración es aquella magnitud que indica la proporción de cambio de velocidad en una unidad de tiempo. Esta magnitud es de tipo vectorial, indicando que posee una dirección. Su unidad de medida en el sistema internacional es el m/seg^2. Esta variable nos permite reconocer patrones asociados a contactos metal-metal y fricciones abrasivas mediante el resalte de picos de vibración en frecuencias medias y altas.

Para medir estas magnitudes, se usan acelerómetros, que convierten vibraciones en señales eléctricas, gracias a su precisión y capacidad para registrar aceleraciones en varias frecuencias.

3.3.1. Acelerómetro piezoeléctrico

El acelerómetro piezoeléctrico es un transductor estándar para la medición de vibración en máquinas. Este elemento produce el efecto piezoeléctrico, mediante el ajuste entre la base y la masa, por lo que, cuando una materia está sujeta a una fuerza, se produce una carga eléctrica entre sus superficies. De acuerdo a la segunda ley de Newton, esa fuerza se relaciona de forma directa con la aceleración de la materia. La fuerza sobre el cristal (usualmente cuarzo) produce la señal de salida, que es proporcional a la aceleración del transductor.

Figura 3.13: Elementos internos de un acelerómetro



Fuente: Ilustración extraída de [38]

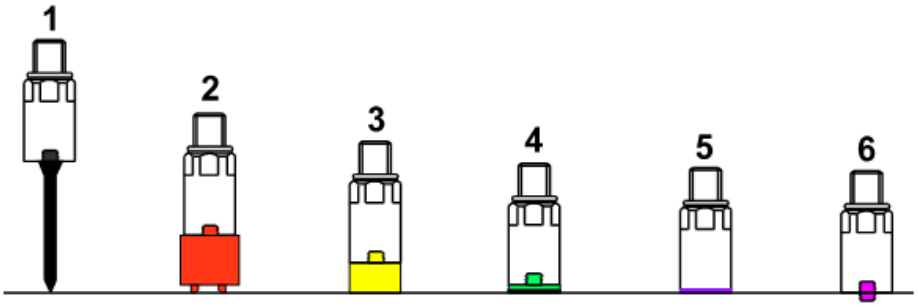
La frecuencia de resonancia del sensor (definida por su masa) debe ser al menos el doble que la frecuencia máxima a medir, para evitar amplificación distorsiva cerca de la resonancia. Además, el rango efectivo depende no solo del sensor, sino también de la rigidez de su montaje.

A fin de lograr la mejor respuesta de frecuencia posible del sensor montado, se han desarrollado seis métodos básicos de montaje:

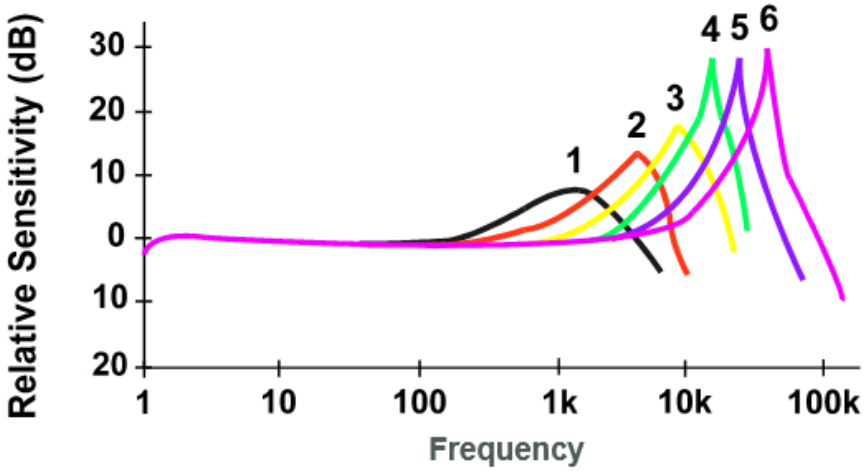
1. Punta de sonda.
2. Imán bipolar.
3. Imán plano.
4. Almohadilla de montaje adhesivo.
5. Únicamente adhesivo.
6. Montaje de perno.

El siguiente diagrama ejemplifica el rendimiento nominal en alta frecuencia de los anteriores métodos de montaje.

Figura 3.14: Métodos de montaje y sus frecuencias de resonancia



(a) Métodos de montaje del sensor



(b) Sensibilidad relativa de cada método

Fuente: Ilustración y gráfica extraída de [39]

Tal como se evidencia en la Figura 3.14b el montaje con pernos roscados ofrece el mejor rendimiento (similar a la calibración ideal), ideal en instalaciones donde se prevén

vibraciones de alta frecuencia (>10 kHz) y entornos hostiles, pero requiere perforar la máquina.

Como alternativa, el montaje adhesivo (cianoacrilato, cinta de doble cara, cera) evita perforaciones, pero su respuesta en frecuencia depende del grosor del adhesivo (entre más delgada sea la película, mejor será la propagación y su frecuencia) y de la preparación superficial.

Otras opciones son para usos temporales o zonas inaccesibles. Para superficies planas, se recomienda el uso de imanes planos; en cambio, en superficies irregulares o curvas, son más adecuados los imanes bipolares. La punta de sonda solo es viable en mediciones por debajo de 10 Hz y tiene la peor respuesta en frecuencia.

La calidad de los datos recolectados es relevante al examinar el tipo de vibración existente en el sistema: puede ser senoidal, con un espectro enfocado en una sola frecuencia (conocida como *"componente fundamental"*), o periódica, cuya descomposición muestra diversos componentes armónicos.

3.3.2. Procesamiento digital de la señal

Las señales contienen información esencial sobre la naturaleza o comportamiento de un fenómeno físico. Por otro lado, los sistemas son estructuras o elementos que procesan dichas señales. Es decir que reciben una señal a la entrada, aplican un conjunto de operaciones y generan una señal a la salida.

Una *"señal simple"* se define como cualquier cantidad que puede representarse matemáticamente como una función de una o más variables independientes. Asimismo, las señales analógicas generalmente presentan ruido, es decir, alteraciones indeseables que no contienen ninguna información relevante para el sistema y que, por el contrario,

introducen incertidumbre en la interpretación de la señal. A este tipo de señales se les denomina *"señales aleatorias"*, debido a su comportamiento impredecible en el dominio del tiempo.

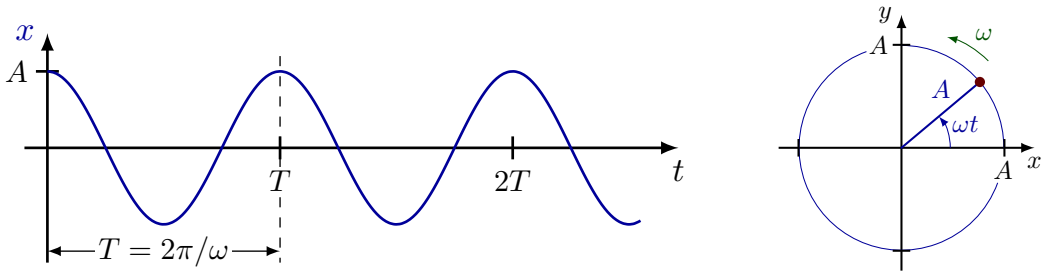
El modelo matemático de señales aleatorias no puede representarse mediante una única ecuación en función del tiempo. En su lugar, se recurre al uso de propiedades estadísticas como la media, la varianza o la densidad espectral de potencia para describir su comportamiento. A través del procesamiento digital de señales es posible aplicar algoritmos que permiten reducir o eliminar el efecto de las distorsiones y del ruido, mejorando la calidad y facilitando su análisis posterior.

El *"procesamiento digital de señales"* consiste en el conjunto de herramientas matemáticas, técnicas y algoritmos empleados para manipular y analizar estas señales una vez que han sido convertidas de forma analógica a digital mediante un proceso de muestreo y cuantización.

Una de las herramientas más empleadas en la actualidad para el procesamiento digital de señales es el análisis en el dominio de la frecuencia, el cual estudia las componentes espectrales de la señal. Este tipo de análisis otorga mayor comprensión del comportamiento de una señal de forma independiente al dominio del tiempo. Útil para la detección de patrones periódicos o extracción de características relevantes para su clasificación o diagnóstico.

En el análisis de señales de vibración, este se puede realizar en el dominio del tiempo mediante el nivel de amplitud o, alternativamente, en el dominio de frecuencia, ya sea por la *"Transformada de Fourier (TF)"*, u otras funciones relacionadas con ella, como son la *Densidad Espectral de Energía (DEE)* y la *Densidad Espectral de Potencia (DEP)* [40].

Figura 3.15: Distintas representaciones de una función



(a) Componente sinusoidal en el dominio del tiempo

(b) Representación factorial

Fuente: Ilustración y gráfica extraída de [41]

3.3.2.1. Transformada de Fourier

La *"Transformada de Fourier"* es una herramienta que permite descomponer una función en una combinación de componentes sinusoidales de distintas frecuencias. Esto proporciona una representación de la señal en el dominio de la frecuencia, es decir, describe cómo se distribuyen las diferentes frecuencias que componen la señal.

Este concepto fue introducido por primera vez en la obra *Théorie analytique de la chaleur* (1822), donde Joseph Fourier dedujo la ecuación de propagación del calor en medios materiales. En ese contexto, se introduce la *Serie de Fourier*, definida como *"Cualquier función periódica $f(t)$, con una frecuencia angular fundamental $\omega_o = 2\pi/T$, puede ser expresada como una suma de funciones sinusoidales con frecuencias múltiplos de ω_o, cada una con distinta amplitud y fase."*

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(nw_0 t\right) + b_n \sin\left(nw_0 t\right)\right] \tag{3.6}$$

Donde:

- a_0 representa la componente de frecuencia cero, equivalente al valor medio de la

función.

- a_n y b_n son los coeficientes de Fourier, determinan la amplitud de cada componente.
- n son los múltiplos de la frecuencia fundamental w_0.

Se emplean tanto funciones seno como coseno para representar posibles desfases en la señal. El coseno se utiliza especialmente para funciones cuyo valor es distinto de cero en $t = 0$. Los coeficientes a_n y b_n se calculan mediante integrales definidas, como se mostrará más adelante.

$$a_n = \frac{2}{T} \int_0^T f(t) \cos(nw_0 t)\, dt \tag{3.7a}$$

$$b_n = \frac{2}{T} \int_0^T f(t) \sin(nw_0 t)\, dt \tag{3.7b}$$

No obstante, si la señal no es periódica, se recurre a la *Transformada de Fourier* (TF), una extensión de la Serie de Fourier que permite analizar funciones no periódicas al considerarlas como periódicas con un periodo que tiende al infinito.

En este caso, a medida que el periodo se hace infinito, la separación entre las frecuencias componentes tiende a cero y la serie se transforma en una integral continua, como se verá más adelante.

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t} dt \tag{3.8}$$

En la práctica, registrar una función periódica en su totalidad no es posible, ya que implicaría un tiempo de observación infinito. Por lo tanto, el análisis se realiza a partir de un número finito de datos obtenidos durante un intervalo limitado.

Para abordar esta limitación, el concepto de Fourier se adapta a señales discretas mediante la *Transformada Discreta de Fourier* (DFT), la cual permite trabajar con señales muestreadas, como las digitales.

En lugar de una integral continua, como en la Transformada de Fourier (véase la

Ecuación 3.8), la DFT utiliza una suma finita para descomponer la señal en sus componentes de frecuencia, considerando N como el número de muestras de la señal discreta.

$$X\left[k\right] = \sum_{n=0}^{N-1} x\left[n\right]\ e^{-i2\pi kn/N} \tag{3.9}$$

Las características que ofrece la DFT son:

- **Linealidad:** La DFT de la suma de dos señales es la suma de las DFT de cada señal individual.
- **Periodicidad:** La DFT es periódica con periodo N, es decir, $X\left[k+N\right] = X\left[k\right]$.
- **Simetría:** Para señales reales, la DFT presenta una simetría hermética, donde las frecuencias positivas y negativas son espejos entre sí.

Estas características fundamentales de la DFT permiten el desarrollo de la *Transformada Rápida de Fourier* (FFT), considerada uno de los algoritmos más importantes en la historia del procesamiento de señales.

La FFT surge con el objetivo de reducir el elevado costo computacional de la DFT, cuyo orden de complejidad es $O(N^2)$ debido al número de multiplicaciones y sumas necesarias para calcular sus componentes. La FFT reduce este costo a $O(N \log_2 N)$, lo que permite un procesamiento mucho más eficiente y práctico de señales. Gracias a ello, la FFT se ha convertido en la base tecnológica para la transmisión y almacenamiento de datos en la actualidad.

3.3.2.2. Análisis de señales no estacionarias

Las *funciones estacionarias* son aquellas cuyas propiedades estadísticas, como la media y la varianza, permanecen constantes en el tiempo. Por ello, su análisis no

depende de un conjunto específico de muestras [42]. En contraste, las señales que no cumplen con estas condiciones, se denominan *"funciones no estacionarias"*, ya que carecen de una periodicidad definida que permita su análisis directo.

No obstante, es posible transformar una señal no estacionaria en una señal cuasi estacionaria para facilitar su estudio. Este proceso consiste en dividir la señal en segmentos de corta duración llamados *"ventanas de tiempo"*, en los que se asume que las propiedades de la señal se mantienen aproximadamente constantes. De este modo, se pueden analizar porciones específicas de interés dentro de la señal.

Al aplicar ventanas de tiempo, es necesario realizar el mismo procedimiento de análisis sobre cada una de ellas. Para evitar la pérdida de información importante entre ventanas adyacentes, se permite que estas se solapen, lo cual se conoce como *"overlap"*.

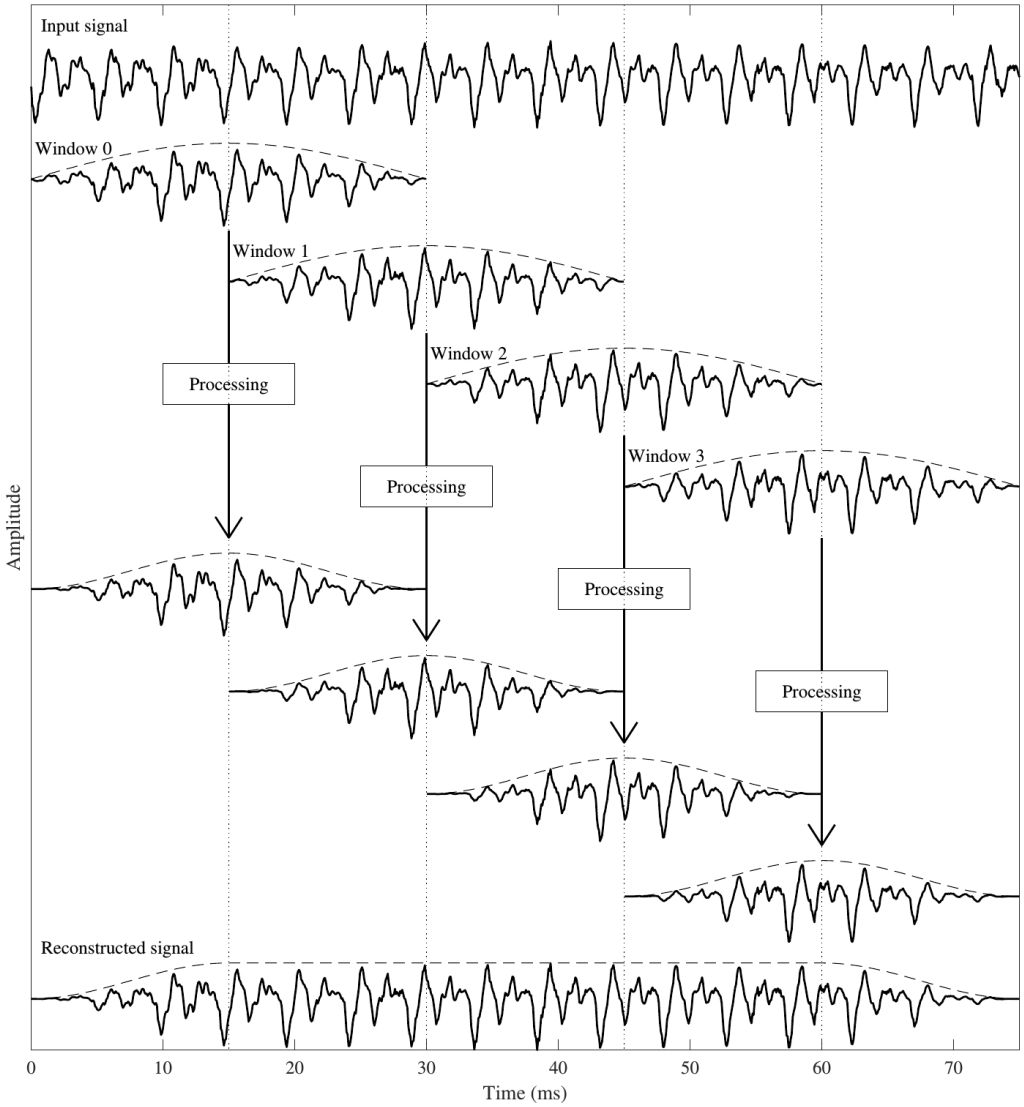
Se recomienda que los extremos de una ventana tomen valores mínimos y que estos coincidan con los valores máximos en el centro de la ventana siguiente. Este enfoque asegura una transición suave y se logra típicamente mediante un solapamiento del 50 % .

No obstante, dividir la señal en bloques que no necesariamente comienzan y terminan en cero puede introducir discontinuidades artificiales. Al aplicar la Transformada Rápida de Fourier (FFT) a estos bloques, estas discontinuidades generan componentes de alta frecuencia no presentes en la señal original. Como resultado, el espectro obtenido representa una versión distorsionada de la señal real, en la que la energía de ciertas frecuencias se dispersa hacia otras. Este fenómeno, conocido como *"Fuga espectral"*, degrada la precisión del análisis espectral al ensanchar las líneas espectrales, originalmente estrechas, hacia frecuencia cercanas a la original.

Para mitigar este efecto, se aplican funciones ventana, diseñadas matemáticamente para suavizar los extremos de cada segmento. Estas funciones suelen comenzar en cero, aumentar gradualmente hasta un valor máximo (típicamente uno), y luego decrecer

nuevamente hasta cero dentro del mismo marco temporal. Este perfil reduce los transitorios abruptos y, en consecuencia, disminuye la fuga espectral en el análisis de frecuencia.

Figura 3.16: Señal con ventana de tiempo cada 30ms y overlap de 50 %



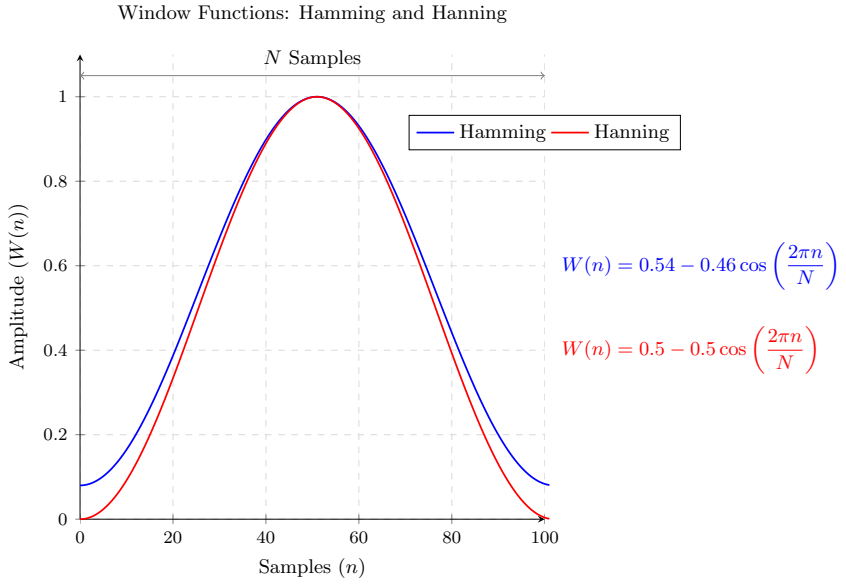
Fuente: Ilustración extraída de [43]

Hay dos requisitos importantes a cumplir al aplicar una función de ventana:

1. El ancho del lóbulo principal debe ser lo más estrecho posible para una mayor resolución.
2. Los lóbulos laterales deben ser lo más reducido posible, a fin de concentrar la mayor parte de energía en el lóbulo principal.

Las funciones de ventana de *"Hamming"* y *"Hanning"* (también conocida como Hann) tienen formas similares basadas en funciones cosenoidales, lo que les confiere una transición suave en el dominio del tiempo. Por un lado, la ventana de Hann toca cero en ambos extremos, eliminando completamente las discontinuidades en los bordes del segmento. En contraste, la ventana de Hamming no se reduce totalmente a cero en sus extremos, lo que puede introducir una leve discontinuidad, aunque a cambio reduce el nivel del lóbulo lateral máximo, es decir, la energía no deseada que se filtra hacia frecuencias adyacentes.

Figura 3.17: Ventana de Hamming y Hanning



Fuente: Elaboración propia

La ventana de Hamming se utiliza principalmente cuando se desea minimizar el nivel máximo de los lóbulos laterales más cercanos, lo que es útil cuando se quiere evitar que señales débiles sean enmascaradas por la fuga de señales fuertes cercanas. Esto la hace adecuada cuando hay señales de diferente amplitud cercanas en frecuencia, aunque sacrifica algo de resolución espectral debido a su lóbulo principal ligeramente más ancho.

Por otro lado, la ventana de Hann ofrece una mejor discriminación espectral gracias a su lóbulo principal más angosto, lo cual es beneficioso cuando se desea distinguir señales muy próximas entre sí en el dominio de la frecuencia. Sin embargo, sus lóbulos laterales son un poco más altos que los de la ventana de Hamming, lo que puede hacerla menos efectiva frente a interferencias fuertes cercanas.

3.3.2.3. Filtros digitales

Un filtro digital es un sistema lineal e invariante en el tiempo (LTI), que modifica la señal de entrada, dependiendo del espectro de sus frecuencias $X(\omega)$ (es decir, cuánta información de cada frecuencia contiene la señal). Cada frecuencia de la señal es procesada mediante una función especial denominada respuesta en frecuencia $H(\omega)$. Esta respuesta decide qué rango se deja pasar, se atenúa, se amplifica o se elimina. Se infiere que la señal de salida se define bajo la siguiente expresión:

$$Y(\omega) = H(\omega) \cdot X(\omega) \tag{3.10}$$

Los sistemas LTI se pueden clasificar mediante dos tipos; a continuación, se detallan ambos tipos:

- **Respuesta finita al impulso (FIR):** Son filtros cuya respuesta es finita al impulso; se caracterizan por ser un sistema no recursivo, dependiendo únicamente de las entradas pasadas, no de las salidas anteriores. Son filtros estables y de fácil

implementación (véase la Ecuación 3.11a).

- **Respuesta infinita al impulso (IIR):** Son filtros cuya respuesta es infinita al impulso; se caracteriza por tener retroalimentación a la salida, dependiendo de entradas y salidas pasadas. Son eficientes computacionalmente, logrando resultados similares con menos coeficientes (véase la Ecuación 3.11b).

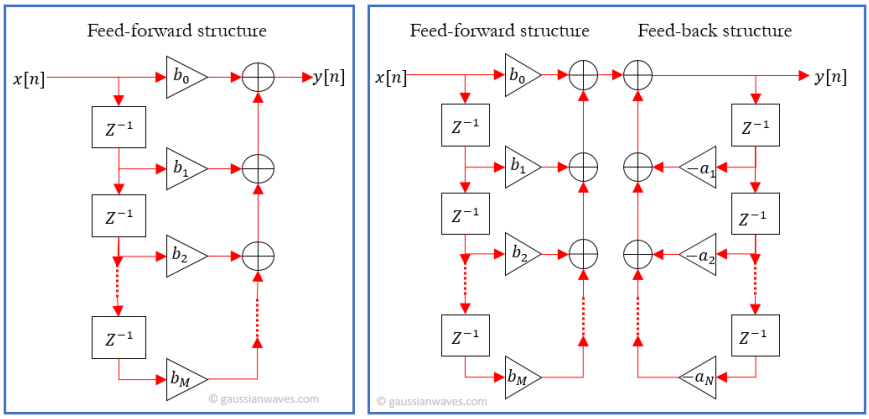
$$y_n(FIR) = b_0 \cdot x[n] + b_1 \cdot x[n-1] + \cdots + b_N \cdot x[n-N] = \sum_{k=0}^{M} b_k x(n-k) \tag{3.11a}$$

$$y_n(IIR) = b_0 \cdot x[n] + b_1 \cdot x[n-1] + \ldots a_1 \cdot y[n-1] + a_2 \cdot y[n-2] + \ldots$$

$$y_n(IIR) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{3.11b}$$

Figura 3.18: Estructura de Filtros Digitales LTI

a) Filtro Digital FIR. b) Filtro Digital IIR



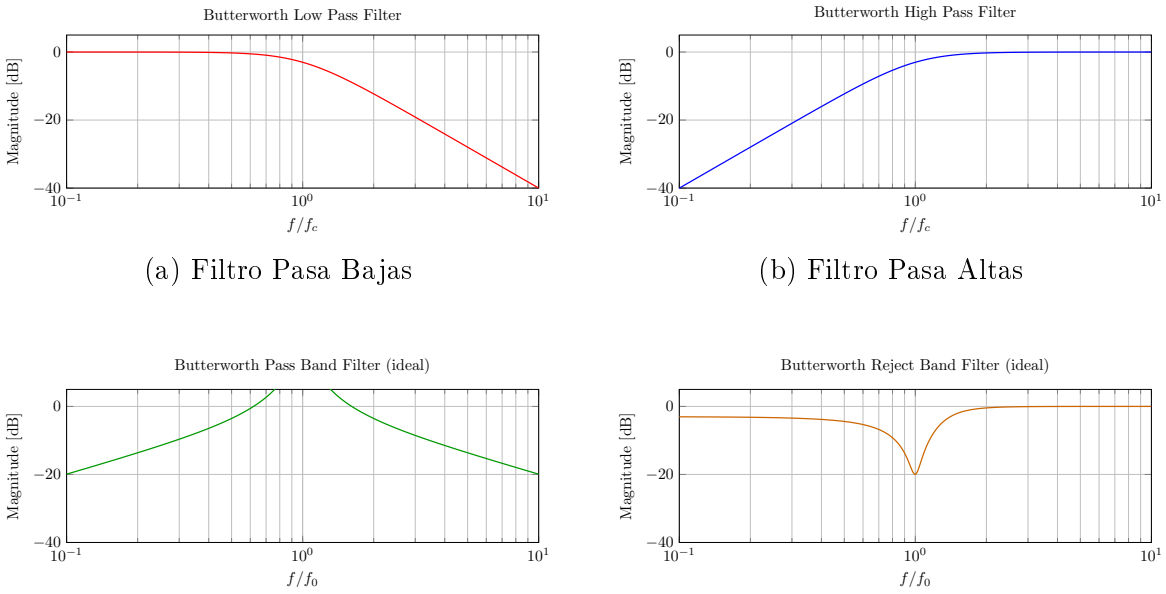
a) b)

Fuente: Ilustración extraída de [44].

Los filtros digitales poseen otra clasificación con base en su respuesta en el dominio de la frecuencia, de la cual se originan cuatro tipos básicos según la posición de las bandas de paso y las bandas atenuadas. Estos cuatro son:

- **Filtro Pasa-Bajo:** Permite el paso de la señal a baja frecuencia con una atenuación a las frecuencias superiores a la *"frecuencia de corte"*.
- **Filtro Pasa-Alto:** Contrario al pasa-bajas, permite el paso de la señal a altas frecuencias con una atenuación a las frecuencias inferiores a la frecuencia de corte.
- **Filtro Pasa-Banda:** Permite el paso de frecuencias dentro de un rango definido entre las frecuencias de corte inferior y superior.
- **Filtro Rechaza-Banda:** Contrario al pasa-banda, bloquea las frecuencias que se encuentran dentro de un rango, mientras permite el paso al resto de frecuencias fuera de ese rango.

Figura 3.19: Filtros Digitales Butterworth



(a) Filtro Pasa Bajas

(b) Filtro Pasa Altas

(c) Filtro Pasa Bandas

(d) Filtro Rechaza Bandas

Fuente: Elaboración propia

3.3.3. Extracción de características

El dominio del tiempo representa la evolución de la señal en función del tiempo. En este dominio, es posible distinguir de manera directa los cambios en la señal, lo que permite analizar su comportamiento transitorio. Sin embargo, su principal limitación es que no proporciona una representación directa de los componentes de frecuencia presentes en la señal.

El dominio de la frecuencia representa el contenido espectral de una señal, mostrando los distintos componentes sinusoidales que la conforman, junto con sus respectivas amplitudes y fases. Este dominio resulta especialmente útil para identificar frecuencias dominantes, armónicos y componentes de ruido presentes en la señal. Sin embargo, su principal limitación es que proporciona únicamente información sobre las frecuencias, sin reflejar el comportamiento temporal de la señal.

Tabla 3.3: Diferencia entre dominios presentes en una señal

Atributo	**Dominio de la frecuencia**	**Dominio del tiempo**
Representación	Señal representada como una suma de componentes sinusoidales	Señal en función del tiempo
Análisis	Se concentra en el contenido de frecuencia de una señal	Se centra en la amplitud y fase de una señal en puntos específicos en el tiempo
Transformar	Transformada de Fourier Transformada Discreta de Fourier	-
Unidades	Hercios (Hz)	Segundos (S)
Periodicidad	Las señales periódicas tienen componentes de frecuencia discretos	Las señales periódicas tienen un patrón que se repite en el tiempo
Fase	Se conserva información de la fase	La información de fase es inherente a la representación en el tiempo
Amplitud	Se conserva información de la amplitud	La información de la amplitud es inherente a la representación en el tiempo
Filtros	Filtros basados en frecuencia (Pasa Bajo, Pasa Alto, etc)	Filtros basados en el tiempo (Media, Móvil, Filtro de mediana)
Aplicaciones	Procesamiento de audio, procesamiento de imágenes, compresión de señal	Reconocimiento de voz, reconocimiento de patrones, reconstrucción de señales

Fuente: Elaboración propia

Para compensar ambas limitaciones, se realiza un análisis por separado: estadístico en el caso del dominio del tiempo y espectral en el dominio de la frecuencia.

En la Tabla 3.4 se visualiza la simbología aplicada a los parámetros estadísticos y espectrales utilizados en este trabajo, que se detallarán más adelante:

Tabla 3.4: Simbología de características

Símbolo	Significado	Símbolo	Significado
x_i	Valor de la i-ésima muestra de una señal	f_i	Frecuencia en el espectro
N	Número total de muestras	$P(f_i)$	Densidad espectral de potencia
$\|x_i\|$	Valor absoluto de la i-ésima muestra	$\overline{P(f)}$	Promedio de la densidad espectral
μ_x	Media aritmética de la señal	$\sigma_{P(f)}$	Desviación estándar del espectro
σ_x	Desviación estándar de la señal	σ_f	Desviación estándar de frecuencia
x_{rms}	Valor cuadrático medio (RMS)	f_{RMS}	Frecuencia cuadrática media
x_{max}	Valor máximo de la señal	f_{RV}	Frecuencia raíz de la varianza espectral
		$f_{\text{centroide}}$	Centroide de frecuencia
		f_{Fvar}	Factor de variación espectral
		f_{Var}	Varianza de frecuencia
		f_{Skew}	Asimetría de frecuencia

Fuente: Elaboración propia

3.3.3.1. Análisis estadístico

El *"Análisis Estadístico"* aplicada a señales se fundamenta bajo la capacidad de examinar conjuntos de datos que permitan identificar patrones, tendencias o comportamientos que no son detectables a simple vista. Un aspecto fundamental que otorga este enfoque es determinar la distribución de probabilidad que mejor describa el comportamiento de la señal. lo cuál permite extrapolar las propiedades observadas en una muestra hacia toda la población de datos.

Al aplicar este enfoque en señales discretas de N muestras, el análisis estadístico permite describir su comportamiento de forma cuantitativa mediante parámetros que capturen información relevante sobre su estructura. Este enfoque resulta especialmente útil en señales complejas, donde los métodos tradicionales pueden no ser suficientes.

En el dominio temporal, los parámetros estadísticos actúan como medidas

representativas de tendencia central, dispersión, forma y energía de la señal a lo largo del tiempo. Estos parámetros permiten resumir el comportamiento dinámico de la señal en métricas que facilitan su análisis, comparación y diagnóstico.

El análisis en el dominio temporal, desarrollado en el marco de esta investigación, considera los siguientes parámetros estadísticos:

Tabla 3.5: Características de una señal en el dominio del tiempo

Dominio del tiempo	**Definición**	**Ecuación**
Media	Valor medio de la señal a partir de N muestras presentes en un intervalo t.	$\mu_x = \frac{\sum_{i=1}^{N} x_i}{N}$
Desviación estándar	Describe la desviación de la i-ésima muestra respecto a la media.	$\sigma_x = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu_x)^2}{N}}$
Desviación absoluta media	Describe la distancia típica entre las muestras y la media.	$\lvert\mu_x\rvert = \frac{\sum_{i=1}^{N} \lvert x_i - \mu_x\rvert}{N}$
Raíz cuadrada media	Mide la energía efectiva y la potencia promedio presente en la señal.	$x_{\text{rms}} = \sqrt{\frac{\sum_{i=1}^{N} (x_i)^2}{N}}$
Valor absoluto máximo	Amplitud pico máximo absoluto presente en la señal.	$x_{\max} = \max \lvert x_i\rvert$
Asimetría	Medida estadística que cuantifica la desviación de la simetría en la distribución de probabilidad de N muestras, en relación a su media, mediante la dirección y grado de sesgo en la distribución.	$\text{Skewness} = \frac{\sum_{i=1}^{N} (x_i - \mu_x)^3}{(N-1)\,\sigma_x^3}$
Curtosis	Medida estadística que describe la forma de distribución de las muestras. Cuantifica el grado en que la distribución de las muestras difiere de la distribución normal (distribución gaussiana o curva de campana).	$\text{Kurtosis} = \frac{\sum_{i=1}^{N} (x_i - \mu_x)^4}{(N-1)\,\sigma_x^4}$
Factor de cresta	Determina algún indicio de impulsividad en la forma de la señal.	$\text{Crest Factor} = \frac{x_{\max}}{x_{\text{rms}}}$
Factor de forma	Relación entre el valor RMS y la media de los valores absolutos. Indica que tan ondulada es una señal. Valores altos a 1.11 (onda sinusoidal pura) indican picos más pronunciados o impulsividad en la señal.	$\text{Form Factor} = \frac{x_{\text{rms}}}{\frac{1}{N}\sum_{i=1}^{N} \lvert x_i - \mu_x\rvert}$
Factor de onda	Indica que tan agudo o ancho es el pico del espectro. Valores bajos pertenecen a una señal selectiva; valores altos pertenecen a señal más dispersa.	$\text{Shape Factor} = \frac{x_{\max}}{\frac{1}{N}\sum_{i=1}^{N} \lvert x_i - \mu_x\rvert}$
Factor de impulso	Relación entre la amplitud máxima y la media de los valores absolutos. Útil para la detección de picos anómalos o transitorios en una señal.	$\text{Impulse Factor} = \frac{x_{\text{rms}}^2}{\frac{\sum_{i=1}^{N} \lvert x_i\rvert}{N}}$

Fuente: Elaboración propia

3.3.3.2. Análisis espectral

El *"Análisis del Espectro"* consiste en descomponer una señal en sus componentes de frecuencia, a fin de identificar la amplitud, la fase y la distribución de energía en función de la frecuencia. Esta metodología es especialmente beneficiosa para la detección de patrones, anomalías o características específicas que no son evidentes en el dominio del tiempo.

A partir de este proceso se obtiene la *"Densidad Espectral de Potencia"* (PSD, por sus siglas en inglés), métrica que representa la distribución de potencia de la señal a lo largo de sus distintas frecuencias.

Para conseguir un cálculo eficaz y de alta resolución de la PSD, se segmenta la señal en segmentos solapados, luego se aplica una función de ventana a cada uno, se determina la FFT y finalmente se promedian los espectros obtenidos. Este procedimiento logra reducir significativamente la varianza y potenciar la resolución en frecuencia, atributos particularmente ventajosos para señales ventajosas o con presencia de ruido.

La caracterización espectral en el dominio de la frecuencia se realizó a partir de los siguientes parámetros:

Tabla 3.6: Características de una señal en el dominio de la frecuencia

Dominio de la frecuencia	**Definición**	**Ecuación**
Espectro de potencia media	Describe cómo se distribuye la potencia de una señal con N muestras a lo largo de múltiples frecuencias, en promedio, a lo largo del intervalo t.	$\overline{P(f)} = \frac{\sum_{i=1}^{K} P(f_i)(f)}{K}$
Espectro de potencia con desviación estándar	Mide cuánto varía el contenido de energía en cada frecuencia.	$\sigma_{P(f)} = \sqrt{\frac{\sum_{i=1}^{K} \left(P(f_i) - \overline{P(f)}\right)^2}{K-1}}$
Espectro de potencia con asimetría	Indica la asimetría del espectro, determinado donde se encuentra la mayor parte de potencia espectral, si a la izquierda o derecha del centroide.	$\text{Skew}_{P(f)} = \frac{\sum_{i=1}^{K} \left(P(f_i) - \overline{P(f)}\right)^3}{K \cdot \sigma_{P(f)}^3}$
Espectro de potencia con curtosis	Indica cuánta potencia del espectro se concentra en las colas de la distribución en comparación al centro.	$\text{Kurto}_{P(f)} = \frac{\sum_{i=1}^{M} \left(P(f_i) - \overline{P(f)}\right)^4}{M \cdot \sigma_{P(f)}^4} - 3$
Frecuencia media	Media de las frecuencias del espectro, de acuerdo a su densidad de potencia.	$\overline{f} = \frac{\sum_i f_i \cdot P(f_i)}{\sum_i P(f_i)}$
Frecuencia con desviación estándar	Cuantifica la dispersión de la energía espectral respecto a la frecuencia media, ponderada por la densidad de potencia.	$\sigma_f = \sqrt{\frac{\sum_i (f_i - \overline{f})^2 \cdot P(f_i)}{K}}$
Frecuencia cuadrática media	Valor que mide la energía global de la señal en el dominio de la frecuencia, con enfoque hacia las altas frecuencias.	$f_{RMS} = \sqrt{\frac{\sum_i f_i^2 \cdot P(f_i)}{\sum_i P(f_i)}}$
Frecuencia de variación de raíz	Mide la dispersión real presente en el espectro.	$f_{RV} = \sqrt{\frac{\sum_i \left(f_i^4 \cdot P(f_i)\right)}{\sum_i \left(f_i^2 \cdot P(f_i)\right)}}$
Centroide de frecuencia	Valor que indica el centro gravitacional de la energía espectral, ajustado por la desviación estándar de las potencias.	$f_{centroide} = \sqrt{\frac{\sum_i f_i \cdot P(f_i) - \sum_i P(f_i) \cdot \sigma_f}{\sum_i P(f_i)}}$
Factor de variación de frecuencia	Valor adimensional que indica la dispersión relativa del contenido espectral.	$f_{Fvar} = \frac{\sigma_f}{\overline{f}}$
Varianza de frecuencia	Evalúa la dispersión absoluta del espectro en función de las fluctuaciones cúbicas de la densidad de potencia respecto a la media, ponderadas por la frecuencia.	$f_{Var} = \frac{\sum_i f_i \cdot \left(P(f_i) - \overline{P(f)}\right)^3 \cdot P(f_i)}{K \cdot \sigma_{P(f)} \cdot \sigma_f}$
Asimetría de frecuencia	Indica el grado de sesgo espectral hacia frecuencias altas o bajas, considerando la curtosis de la densidad de potencia ponderada por la frecuencia.	$f_{Skew} = \frac{\sum_i f_i \cdot \left(P(f_i) - \overline{P(f)}\right)^4 \cdot P(f_i)}{K \cdot \sigma_{P(f)}^2 \cdot \sigma_f}$
Amplitud de pico	Amplitud máxima de una señal	
Ubicación de pico	Frecuencia donde se encuentra el valor máximo del espectro de la señal	

Fuente: Elaboración propia

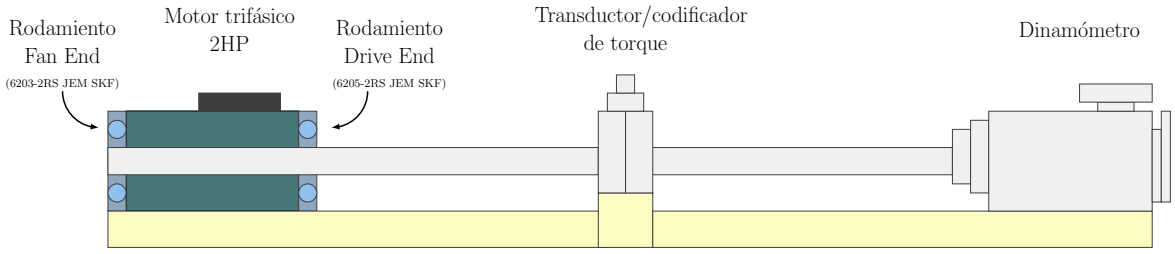
Capítulo 4

Metodología

La base de datos utilizada en la presente investigación corresponde al *Centro de Datos de Rodamientos de Elementos Rodantes de la Universidad Case Western Reserve* (CWRU, por sus siglas en inglés) [45]. Esta base de datos se caracteriza por aportar registros experimentales de vibraciones provocadas por fallas en los rodamientos.

El experimento se llevó a cabo en un banco de pruebas por un motor eléctrico trifásico de 2 HP, un transductor/codificador de torque y un dinamómetro que permite simular distintas cargas en el motor desde 0 HP hasta 3 HP. La velocidad del motor se encuentra en un rango de entre 1730 RPM y 1979 RPM.

Figura 4.1: Diagrama del banco de pruebas del CWRU



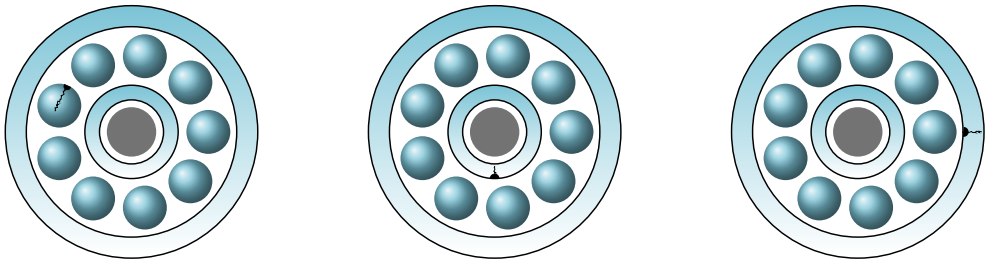
Fuente: Elaboración propia

Se emplearon dos tipos de rodamientos de la marca SKF, instalados en dos puntos del motor: el extremo de la transmisión (6205-2RS JEM SKF) y el extremo del ventilador (6203-2RS JEM SKF).

Las fallas se introducen de forma artificial mediante el mecanizado por electroerosión (EDM), afectando únicamente un componente del rodamiento a la vez. Se contemplan tres tipos de fallas:

- Falla en los elementos rodantes (RB, *Rolling Ball*).
- Falla en la pista interna (IR, *Inner Race*).
- Falla en la pista exterior (OR, *Outer Race*).

Figura 4.2: Fallas de rodamientos registradas



(a) Falla en elemento rodante

(b) Falla en pista interior

(c) Falla en pista exterior

Fuente: Elaboración propia

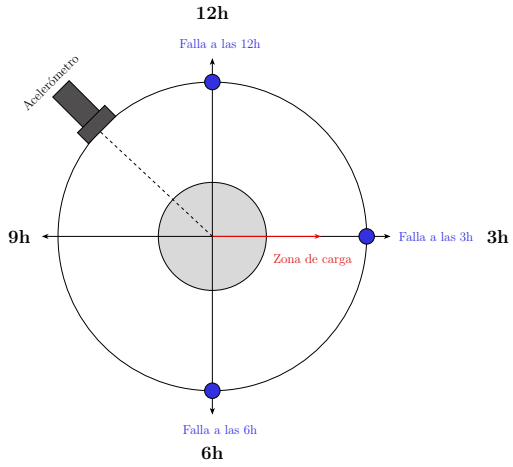
Las fallas en la pista externa de un rodamiento se consideran estacionarias, ya que esta pista es fija a la carcasa y no gira con el eje. Usualmente, se utiliza una referencia tipo reloj para describir la ubicación angular de la falla respecto a la carcasa, como se describe a continuación:

a) En 3:00 Hrs (carga máxima): Zona directa a la zona de carga principal, donde los elementos rodantes ejercen mayor presión sobre la falla.

b) En 6:00 Hrs (carga intermedia): Zona inferior del rodamiento, perpendicular a la carga.

c) En 12:00 Hrs (descarga): Zona superior del rodamiento, opuesta a la zona de carga principal, donde el contacto entre elementos rodantes y la pista externa es mínima.

Figura 4.3: Ubicación angular de la falla de la pista externa



Fuente: Elaboración propia

Las señales de vibración se recolectan mediante acelerómetros montados sobre la carcasa del motor mediante bases magnéticas. Estas señales, junto con las correspondientes a las condiciones normales de operación (rodamientos sanos), se almacenan en archivos con formato .mat, los cuales incluyen las siguientes series temporales:

- DE: Señal proveniente del acelerómetro ubicado en el extremo de la transmisión.
- FE: Señal proveniente del acelerómetro ubicado en el extremo del ventilador.
- BA: Señal proveniente de la base del acelerómetro.
- RPM: RPM durante la prueba.

Para el caso de la serie DE, se registraron en dos frecuencias de muestreo distintas: 12 kHz y 48 kHz. Mientras que la serie FE únicamente se recopiló con una tasa de muestreo de 12 kHz. Para el registro de datos en condiciones normales se emplea una tasa de muestreo de 48 kHz.

Asimismo, se variaron los diámetros y profundidades de las fallas, tal como se detalla en la siguiente tabla.

Tabla 4.1: Especificaciones de las fallas en rodamientos

Ubicación del rodamiento	Tipo de falla	Diámetro de la falla	Profundidad de la falla	Fabricante del rodamiento
Extremo de la transmisión (Drive End)	Pista Interna (IR)	0.007	0.011	SKF
Extremo de la transmisión (Drive End)	Pista Interna (IR)	0.014	0.011	SKF
Extremo de la transmisión (Drive End)	Pista Interna (IR)	0.021	0.011	SKF
Extremo de la transmisión (Drive End)	Pista Interna (IR)	0.028	0.050	NTN
Extremo de la transmisión (Drive End)	Pista Externa (OR)	0.007	0.011	SKF
Extremo de la transmisión (Drive End)	Pista Externa (OR)	0.014	0.011	SKF
Extremo de la transmisión (Drive End)	Pista Externa (OR)	0.021	0.011	SKF
Extremo de la transmisión (Drive End)	Pista Externa (OR)	0.040	0.050	NTN
Extremo de la transmisión (Drive End)	Elemento Rodante (B)	0.007	0.011	SKF
Extremo de la transmisión (Drive End)	Elemento Rodante (B)	0.014	0.011	SKF
Extremo de la transmisión (Drive End)	Elemento Rodante (B)	0.021	0.011	SKF
Extremo de la transmisión (Drive End)	Elemento Rodante (B)	0.028	0.150	NTN
Extremo del ventilador (Fan End)	Pista Interna (IR)	0.007	0.011	SKF
Extremo del ventilador (Fan End)	Pista Interna (IR)	0.014	0.011	SKF
Extremo del ventilador (Fan End)	Pista Interna (IR)	0.021	0.011	SKF
Extremo del ventilador (Fan End)	Pista Externa (OR)	0.007	0.011	SKF
Extremo del ventilador (Fan End)	Pista Externa (OR)	0.014	0.011	SKF
Extremo del ventilador (Fan End)	Pista Externa (OR)	0.021	0.011	SKF
Extremo del ventilador (Fan End)	Elemento Rodante (B)	0.007	0.011	SKF
Extremo del ventilador (Fan End)	Elemento Rodante (B)	0.014	0.011	SKF
Extremo del ventilador (Fan End)	Elemento Rodante (B)	0.021	0.011	SKF

Fuente: Elaboración propia

4.1. Metodología preliminar

Antes de definir la metodología actual para la segmentación de señales y extracción de características, descrita en la Figura 4.7, se exploraron dos enfoques preliminares con el fin de evaluar la viabilidad del análisis y comprender el comportamiento de las señales. Estas aproximaciones permitieron identificar las limitaciones de métodos sencillos y justificar el desarrollo de una solución más robusta y estable para el entrenamiento de la red neuronal.

a) **Extracción de características temporales sin filtrado:** Este primer enfoque consiste en extraer únicamente cuatro características temporales: media, desviación estándar, curtosis y asimetría. La extracción se aplica sobre las señales correspondientes a la estructura de Drive End y Healthy, sin realizar un filtrado previo, pero estableciendo una longitud estándar dependiendo de su condición.

b) **Extracción de características con funciones nativas de MATLAB y filtrado:** El segundo enfoque integra la etapa de filtrado y la aplicación del extractor de características nativa de MATLAB, mediante las funciones de `signalTimeFeatureExtractor` (disponible desde R2021a) y `signalFrequencyFeatureExtractor` (desde R2021b), sobre las señales correspondientes a las estructuras Drive End, Fan End y Healthy. Se extraen 16 características: 9 temporales y 5 frecuenciales (considerando la amplitud y ubicación de los dos picos máximos en el dominio de la frecuencia).

   Características proporcionadas por los extractores nativos de MATLAB como el *BandPower*, *SINAD* y *PowerBandwidth* fueron evaluadas pero descartadas en la metodología final por rebundancia y baja contribución al desempeño del modelo.

   Aunque esta metodología considera opciones para la segmentación por ventanas y solapamiento, en la práctica dichas configuraciones no se llevan a cabo, optando

por procesar la señal en su totalidad. Obteniendo un conjunto de datos limitado e insuficiente para el entrenamiento del modelo.

4.2. Extracción de datos

Para la extracción de datos desde los archivos .mat, se desarrolla un script en MATLAB que permite seleccionar los rangos de señales, definidos en la Tabla 4.2. El código identifica y extrae las series etiquetadas como DE, FE, BA y RPM, para ser almacenadas en celdas individuales que contienen cada vector correspondiente. Posteriormente, las celdas se agrupan en subconjuntos según el tipo de falla y su rango de muestras.

Tabla 4.2: Creación del Dataset para Extracción

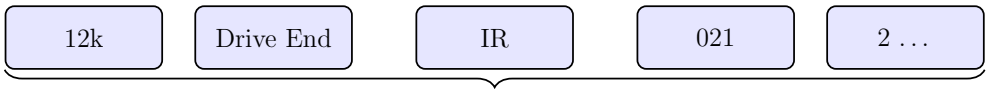
Ubicación del rodamiento	Tipo de falla	Frecuencia de muestreo	Rango del señales	Cantidad de señales
DRIVE END	Ball	12 KHz	01-16	16
DRIVE END	Inner	12 KHz	17-32	16
DRIVE END	Outer	12 KHz	33-60	28
FAN END	Ball	12 KHz	61-72	12
FAN END	Inner	12 KHz	73-84	12
FAN END	Outer	12 KHz	85-105	49
	Healthy	48 KHz	158-161	5
Total				**138**

Fuente: Elaboración propia

De acuerdo a la nomenclatura de los archivos mostrada en la Figura 4.4, las fallas en la pista externa se identifican por la carga aplicada, indicada mediante las etiquetas @12, @3 o @6 en el nombre del archivo. Cuando estas etiquetas están presentes, las señales se agrupan en subconjuntos separados bajo la nomenclatura: *Outer Race: 1H, 3H o 6H*,

según corresponda al nivel de carga [1].

Figura 4.4: Nomenclatura del CWRU Bearing Dataset



Nomenclatura del archivo de datos

Frecuencia de muestreo:	**Ubicación del rodamiento:**	**Tipo de falla del rodamiento:**	**Diámetro de la falla:**	**Condición de carga:**
12k = 12,000 muestras x 1s 48k = 48,000 muestras x 1s	Drive End Fan End	B: Falla en bola IR: Falla en pista interior OR: Falla en pista exterior ◦ @3: Carga máxima ◦ @6: Carga intermedia ◦ @12: Descarga	007: 0.0177 cm 014: 0.0355 cm 021: 0.0533 cm 028: 0.0711 cm 040: 0.1219 cm	0: 1797 rpm 1: 1772 rpm 2: 1750 rpm 3: 1730 rpm

Fuente: Elaboración propia

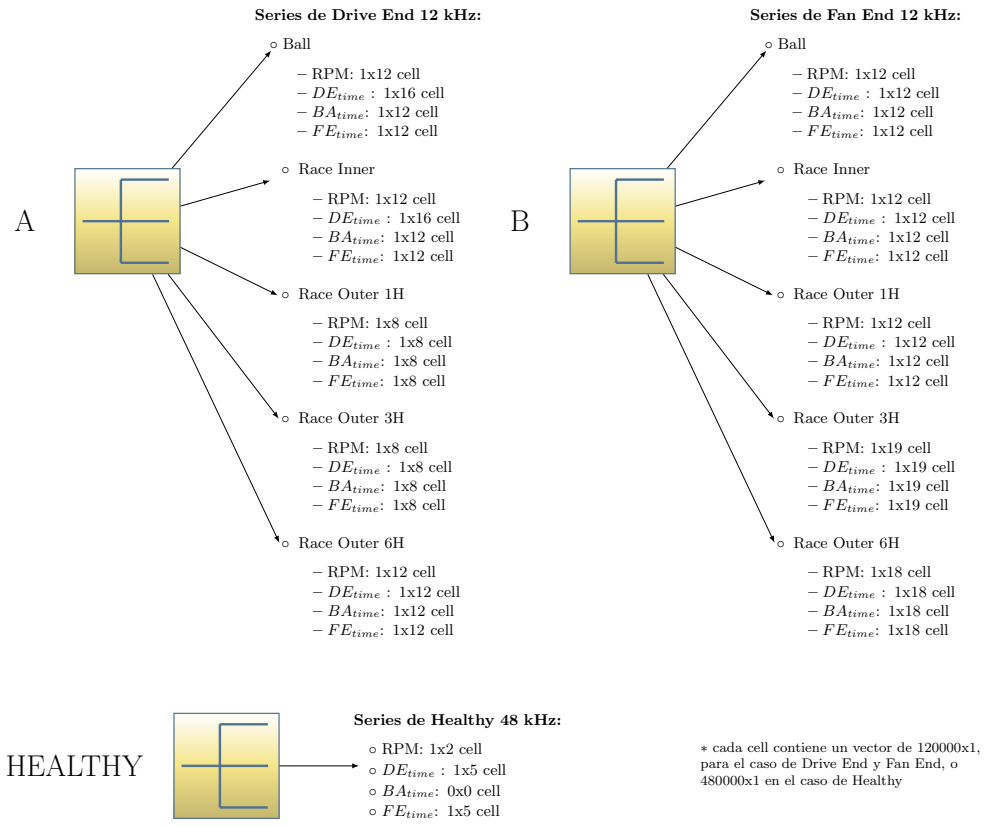
Para estandarizar la longitud de las señales, aquellas que corresponden a fallas en Drive End o Fan End se recortan a un máximo de 120,000 muestras. En el caso de señales correspondientes a condiciones normales de operación, se recortan a un máximo de 480,000 muestras con el fin de conservar la mayor cantidad posible de datos, dado que este tipo de condición tiene la menor representación en la base de datos.

Asimismo, algunos archivos contienen múltiples señales. Por ejemplo, los datos de Outer Race del Fan End reúnen 49 señales distribuidas en 21 archivos. Del mismo modo, las condiciones normales de operación agrupan 5 señales distribuidas en 3 archivos.

Finalmente, cada subconjunto de datos (Ball, Inner Race, Outer Race 1H, Outer Race 3H, Outer Race 6H) se guarda dentro de una estructura general según la ubicación del rodamiento como: A para Drive End y B para Fan End.

[1] La metodología empleada considera las fallas en la pista externa de las 12H equivalentes a la categoría 1H, con el objetivo de un manejo más eficiente de los datos, siguiendo la nomenclatura y el rango de selección de las señales.

Figura 4.5: Extracción de las series temporales



Fuente: Elaboración propia

Una vez que se obtienen las señales en crudo y con una longitud estandarizada, es más fácil procesarlas a través de filtros y ventanas.

4.3. Creación de Filtros Digitales

Dentro del campo del procesamiento digital de señales, la creación de filtros digitales adecuados es fundamental para aislar componentes espectrales específicos, especialmente si están asociados a defectos mecánicos. Para este análisis, se emplean filtros de tipo IIR, que consumen menos poder computacional en comparación a los filtros FIR..

4.3.1. Encontrar las frecuencias para cada falla

Con el objetivo de identificar las bandas espectrales asociadas a fallas características en los rodamientos, se calculan las frecuencias teóricas BPFO, BPFI y BSF, conforme a las expresiones 3.2, 3.3 y 3.4, respectivamente.

A partir del rango de revoluciones registrado en la base de datos (1718 rpm $-$ 1797 rpm), se construye un vector de RPM con incrementos unitarios. Posteriormente, se emplean los parámetros técnicos de los rodamientos en las ubicaciones Drive End y Fan End (véase la Tabla 4.3) para calcular las frecuencias BPFI y BSF. Para la estimación de BPFO, el vector de RPM se segmenta en tres partes iguales, distribuyéndose de la siguiente manera: primer tercio para 6H, segundo tercio para 3H y último tercio para 1H.

Con el fin de capturar la información espectral alrededor de dichas frecuencias, se definen intervalos con un margen de tolerancia de $\pm$ 250 Hz en torno a cada frecuencia central de defecto. Estos intervalos sirven como límites para diseñar filtros digitales que restringen el amplio espectro de la señal a un rango específico centrado en las frecuencias asociadas a fallas características y de esta forma evitar la superposición de frecuencias al momento de extraer las características.

Se opta por emplear un filtro pasabanda Butterworth de tipo IIR de orden 50° a todas las señales correspondientes a condiciones de falla. Para la condición de operación normal (Healthy), se aplica un filtro pasabajas Butterworth del mismo tipo y orden, con una frecuencia de corte de $f_c = 1200$ Hz. En este caso, se determina la frecuencia de corte considerando que la información relevante suele estar contenida en componentes de baja frecuencia, típicamente en el rango de 0 a 1000 Hz. La transición del filtro se diseña de modo que la atenuación comience antes de alcanzar las frecuencias asociadas a defectos

identificables, evitando así interferencias en la región crítica del espectro.

Las frecuencias listadas en la Tabla 4.4 corresponden a los intervalos estimados para cada modo de falla, incluyendo el margen de tolerancia indicado. En la Figura 4.6, se puede apreciar el espectro de potencia de las señales posterior a su filtrado.

Tabla 4.3: Parámetros técnicos de los rodamientos

Ubicación del sensor	Modelo y fabricante	Número de elementos rodantes	Diámetro del elemento rodante	Diámetro de paso	Ángulo de contacto (Beta)
Drive End	**6205-2RS JEM SKF**	9	0.7940 cm	3.9039 cm	0°
Fan End	**6203-2RS JEM SKF**	9	0.6746 cm	2.8498 cm	0°

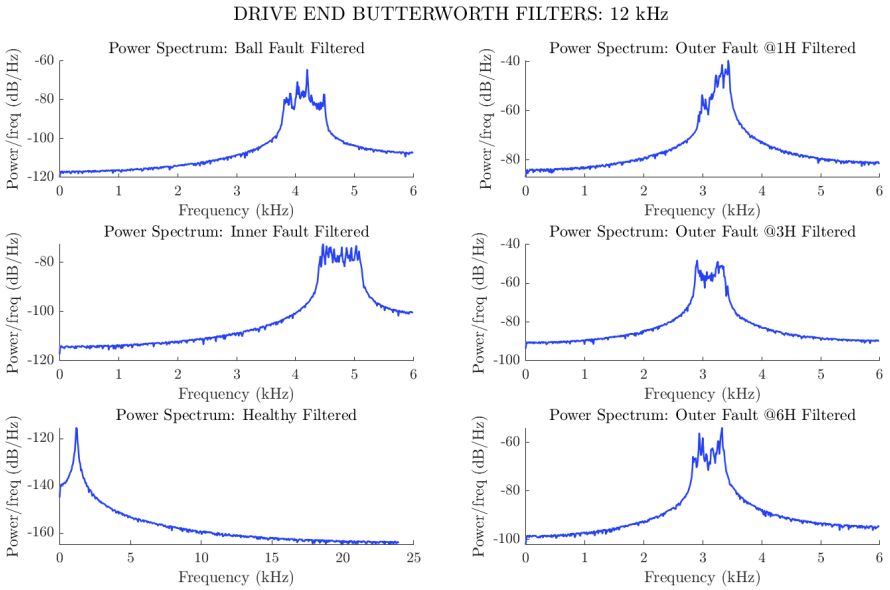
Fuente: Elaboración propia

Tabla 4.4: Frecuencias estimadas de defecto para Drive End y Fan End

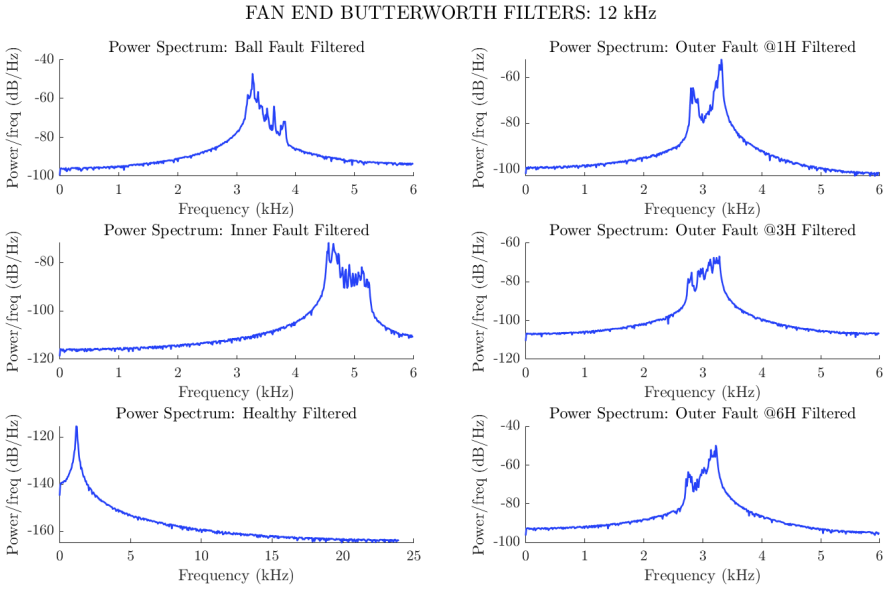
Drive End: Frecuencia asociada a falla		Frecuencia de defecto		Fan End: Frecuencia asociada a falla		Frecuencia de defecto	
BSF		4048.8479	Hz	BSF		3425.4149	Hz
		4235.0289	Hz			3582.9282	Hz
BPFI		4651.6778	Hz	BPFI		4780.5417	Hz
		4865.5792	Hz			5000.3687	Hz
BPFO	@1	3174.3187	Hz	BPFO	@1	3041.4794	Hz
		3220.9208	Hz			3086.1313	Hz
	@3	3125.9243	Hz		@3	2995.1102	Hz
		3172.5264	Hz			3039.762	Hz
	@6	3079.3222	Hz		@6	2950.4583	Hz
		3124.1319	Hz			2993.3928	Hz

Fuente: Elaboración propia

Figura 4.6: Espectro de potencia de las señales filtradas en Drive End y Fan End



(a) Filtrado de señales en Drive End



(b) Filtrado de señales en Fan End

Fuente: Elaboración propia

4.4. Segmentación de la señal en ventanas de tiempo

Con base en el diagrama de flujo mostrado en la Figura 4.7, se desarrolla un script para segmentar las señales en ventanas de 1.25 segundos, considerando la frecuencia de muestreo específica de cada muestra y aplicando un 50 % de solapamiento entre ventanas. A cada segmento se le aplica una función de ventana de Hanning, con el fin de reducir las discontinuidades en los extremos de la señal y mejorar la calidad del análisis espectral.

Durante la etapa de evaluación, se comparan diferentes funciones de ventana. Aunque la ventana de Hamming atenúa parcialmente las discontinuidades, genera un espectro con mayor nivel de ruido y presencia marcada de lóbulos secundarios, lo que dificulta la identificación precisa de componentes espectrales. En cambio, la ventana de Hanning ofrece un mejor rendimiento al suprimir de manera más efectiva los lóbulos laterales, produciendo espectros más limpios y consistentes, resultando en la opción más adecuada para esta aplicación.

Posteriormente, se programan manualmente las funciones para la extracción de características en el dominio temporal y frecuencial (véanse las Tablas 3.5 y 3.6). Aunque MATLAB, a partir de la versión R2021b, incluye un extractor de características con opciones de ventaneo y solapamiento, este realiza la extracción de forma global, sin segmentar la señal, lo que limita su utilidad en contextos donde se requiere un análisis detallado por ventana. Además, su conjunto de características es reducido y poco flexible para ser adaptado a necesidades específicas.

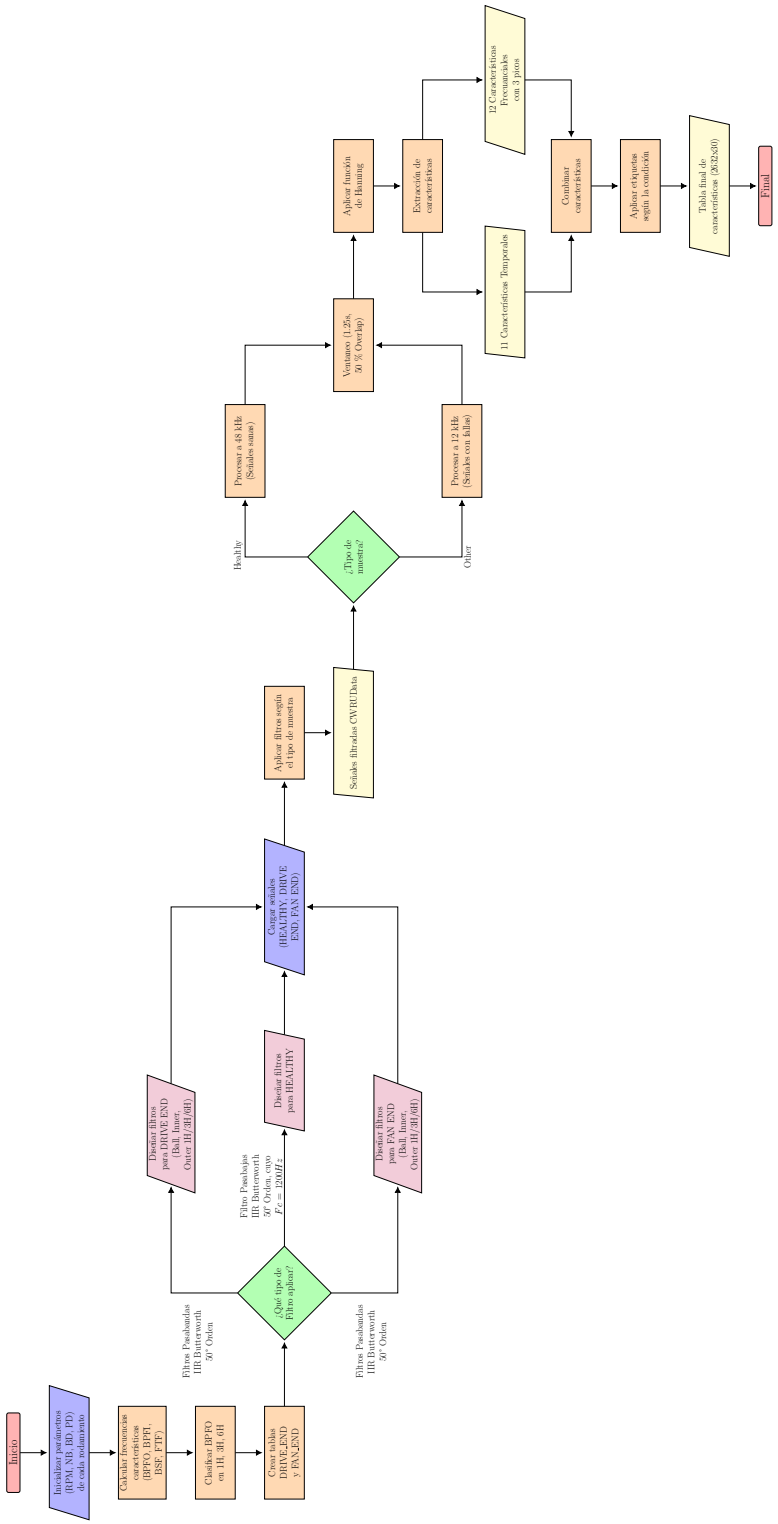
La programación manual permite garantizar la obtención de información temporal y espectral en cada una de las 15 ventanas generadas por señal, con el objetivo de enriquecer el conjunto de datos original de 138×29 valores a 2632×29, incrementando significativamente el nivel de detalle para cada muestra.

Entre las ventajas que ofrece la programación manual destaca un mayor control sobre el proceso de extracción, incluyendo la selección personalizada de 11 características temporales y 12 características frecuenciales, además de la amplitud y ubicación de los tres picos espectrales máximos, permitiendo una representación más completa y precisa de la señal a nivel local.

Finalmente, se asigna una etiqueta a cada ventana segmentada según la condición de la muestra correspondiente, utilizando la siguiente nomenclatura:

- Healthy: 0
- Ball Fault en DE: 1A
- Inner Race Fault en DE: 2A
- Outer Race Fault 1H en DE: 3A 12H
- Outer Race Fault 3H en DE: 3A 3H
- Outer Race Fault 6H en DE: 3A 6H
- Ball Fault en FE: 1B
- Inner Race Fault en FE: 2B
- Outer Race Fault 1H en FE: 3B 12H
- Outer Race Fault 3H en FE: 3B 3H
- Outer Race Fault 6H en FE: 3B 6H

Figura 4.7: Metodología de Pre-procesamiento de la señal



Fuente: Elaboración propia

4.5. Clasificación de señales mediante Redes Neuronales Profundas

La decisión de emplear redes neuronales LSTM fue motivada a través de una analogía conceptual con los codificadores de audio, ya que ambas tecnologías buscan preservar información esencial eliminando redundancias en datos secuenciales.

Los codificadores de audios son dispositivos comúnmente utilizados para la compresión de señales sonoras de alta calidad (como el WAV o FLAC) a formatos más ligeros y eficientes para su transmisión, como el MP3. Emplean transformaciones como la Transformada Discreta de Fourier (DFT) y la Transformada de Coseno Discreta (DCT) para retener solo los componentes espectrales relevantes.

De forma similar a los anteriores, una red LSTM analiza secuencias temporales mediante mecanismos de memoria a largo plazo y puertas de control, que le permiten retener patrones relevantes y atenuar la influencia de variaciones ruidosas o información redundante. Esta capacidad de modelar relaciones temporales y frecuenciales mediante los estados internos de la misma las convierte en potentes instrumentos para el análisis de señales mediante la extracción de características.

Asimismo, un ajuste adecuado de los hiperparámetros, como el tamaño de la memoria y la tasa de aprendizaje, permite al LSTM mejorar la precisión del modelo al disminuir la pérdida de información esencial.

Por estas razones, se adopta una arquitectura basada en redes LSTM para la clasificación de señales, debido a su capacidad de conservar información crítica a lo largo del tiempo. Este modelo incluye un conjunto auxiliar de Redes Neuronales Recurrentes (RNN) con el objetivo de reforzar la capacidad de memoria de corto plazo y capturar dinámicas temporales adicionales que podrían ser parcialmente ignoradas

por la arquitectura principal. Además, se integra una capa de *Dropout* durante el entrenamiento para reducir el riesgo de sobreajuste, desactivando aleatoriamente un porcentaje de unidades LSTM en cada iteración.

La Figura 4.8 presenta la estructura general de la red neuronal utilizada para la clasificación de señales, basada en 29 características de entrada y 11 clases de salida. Además, la Tabla 4.5 incluye el ajuste de hiperparámetros correspondiente al modelo con mejor desempeño.

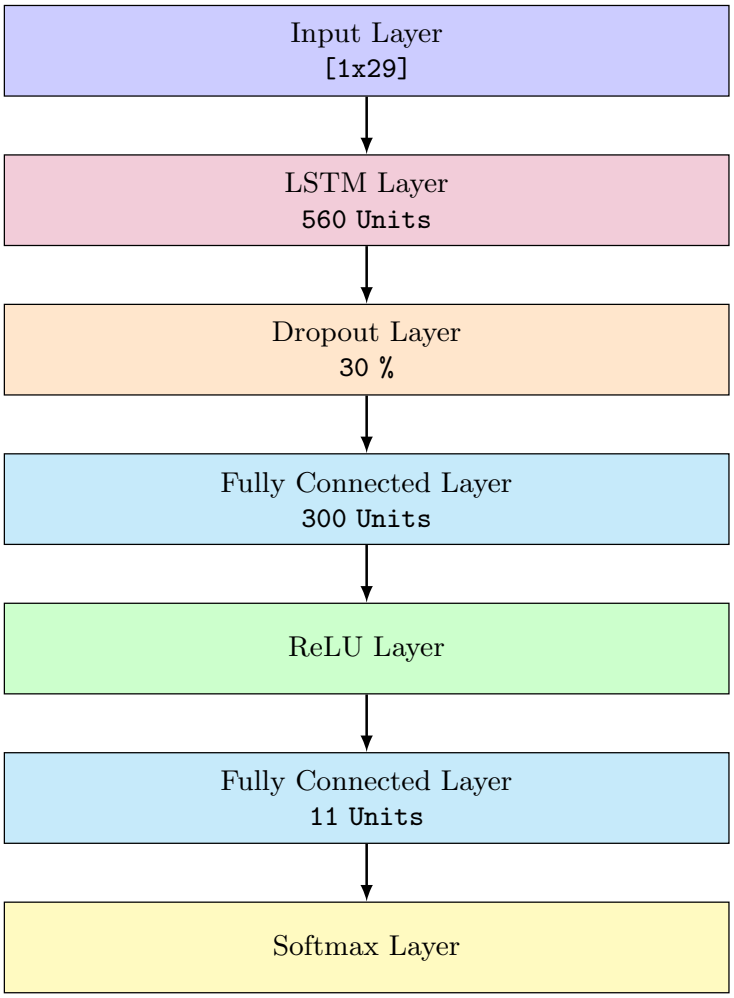
Tabla 4.5: Ajuste de hiperparámetros con el mejor desempeño

Mejor selección de hiperparámetros	
Solver	adam
MaxEpoch	2000
MiniBatchSize	900
Shuffle	every-epoch
InitialLearnRate	0.001
SequenceLength	shortest
ExecutionEnvironment	gpu

Fuente: Elaboración propia

Finalmente, se resalta que la configuración de los hiperparámetros de la red se realizó de forma manual, mediante un proceso iterativo basado en pruebas sucesivas. En cada iteración, se evaluó el desempeño del modelo comparando los resultados obtenidos con métricas de precisión y eficiencia computacional. Este procedimiento permitió identificar un equilibrio adecuado entre la precisión del modelo, el rendimiento del algoritmo en MATLAB y su correcta ejecución en el entorno de desarrollo.

Figura 4.8: Arquitectura del modelo LSTM para clasificación multiclase basada en 29 características de entrada



Fuente: Elaboración propia

Para calcular las características más relevantes del aprendizaje del modelo, es necesario realizar un análisis de la importancia de las características (*"Feature Importance"* en inglés). El cual permite reducir la dimensionalidad del conjunto de datos, preservando la mayor cantidad de información relevante del conjunto de datos original.

4.5.1. Importancia de las características

El propósito de este análisis es identificar qué características (o variables) introducidas al modelo son de mayor importancia en la generación de predicciones. Comprender la relevancia de cada característica permite evaluar la contribución individual de las variables al rendimiento del modelo, y por tanto, mejorar su diseño y eficacia.

Una de las técnicas más utilizadas para este fin es la permutación de la importancia de las características (*"Permutation Feature Importance"*, en inglés). Introducida inicialmente por Leo Breiman en 2001 para su uso en algoritmos de Random Forest, y posteriormente adaptada por Fischer en 2018 a una versión más general e independiente del modelo.

La técnica de importancia por permutación evalúa cuánto afecta al rendimiento del modelo alterar los valores de una característica específica de forma aleatoria. Al aplicar la permutación en una característica, se mezclan sus valores entre las filas al azar, dejando el resto de variables y etiquetas intactas. Esta alteración rompe la relación natural que existía entre dicha característica y el resto de los datos de cada observación.

Por ejemplificar:

- Antes de permutar:

  `Velocidad: 10, Temperatura: 45, Vibración: 0.03, Etiqueta: "Falla Interna"`.

- Después de permutar:

  `Velocidad: 10, Temperatura: 45, Vibración: 0.15, Etiqueta: "Falla Interna"`.

Ese nuevo valor de vibración ya no tiene sentido respecto a la velocidad y la temperatura, ni siquiera con esa etiqueta, porque pertenece a otro contexto original (es decir, otra fila).

Por tanto, al romper esa coherencia, evaluamos qué tanto dependía el modelo de ese valor específico de la característica. Si la permutación hace que el modelo se equivoque más, quiere decir que esa variable tenía un papel importante en la predicción.

Este método permite cuantificar la influencia de cada variable comparando el error de predicción del modelo antes y después de realizar la permutación aleatoria. El impacto en la predicción se expresa en forma de cociente entre ambos errores (Ecuación 4.1), donde:

- Si el cociente es igual a 1, significa que la permutación no afecta significativamente el rendimiento, por lo que dicha característica carece de relevancia al modelo.
- Si el cociente es menor a 1, la reducción del error tras la permutación infiere que esa característica tiene un efecto negativo en el modelo al introducir ruido o generar confusión en las predicciones.
- Si el cociente es mayor a 1, el desempeño del modelo se ve afectado, indicando que esa característica tiene un peso importante en las predicciones.

El principio de funcionamiento de este enfoque se resume a continuación:

1. **Preparación del entorno de evaluación:**
    - Utilizar un modelo previamente entrenado.
    - Contar con:
        - Matriz de entrada X, que contiene las características originales.
        - Vector objetivo Y, que presenta las etiquetas/clases reales a predecir.

- Definir una métrica de error para evaluar el rendimiento del modelo, empleando el Error Cuadrático Medio (RMSE).

2. Cálculo del error base:

- Evaluar el modelo usando el conjunto de datos original (X,Y).
- Calcular el error de referencia, denominado como $RMSE_{orig}$, que funge como punto de comparación para las permutaciones posteriores.

3. Evaluación por permutación para cada característica X_i en X:

a) Permutar aleatoriamente los valores de la característica X_i, con el objetivo de obtener una nueva versión de los datos de entrada X_{perm}.

b) Evaluar el modelo con X_{perm} como entrada y calcular $RMSE_{perm}$.

c) Calcular el cociente de importancia para la característica permutada:

$$\text{Importancia de la característica } \mathrm{X}_i = \frac{RMSE_{perm}}{RMSE_{orig}} \tag{4.1}$$

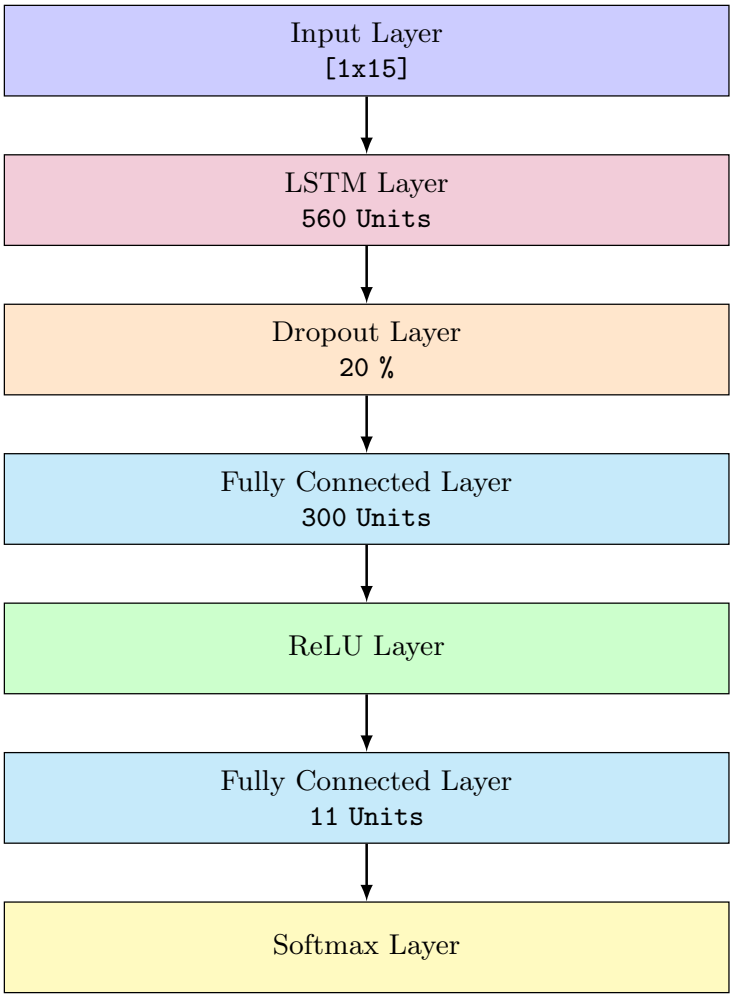
4. Interpretación de resultados:

- Organizar los cocientes de forma descendente.
- Las características con mayores cocientes representan un mayor impacto en el rendimiento del modelo.

Con base en este procedimiento, se identifican las 15 características más influyentes del conjunto de datos original. Estas se utilizan como entradas en un nuevo modelo neuronal,

con el objetivo de comparar su rendimiento frente al modelo que emplea la totalidad de las variables. Se describe la estructura de esta segunda red neuronal en la Figura 4.9.

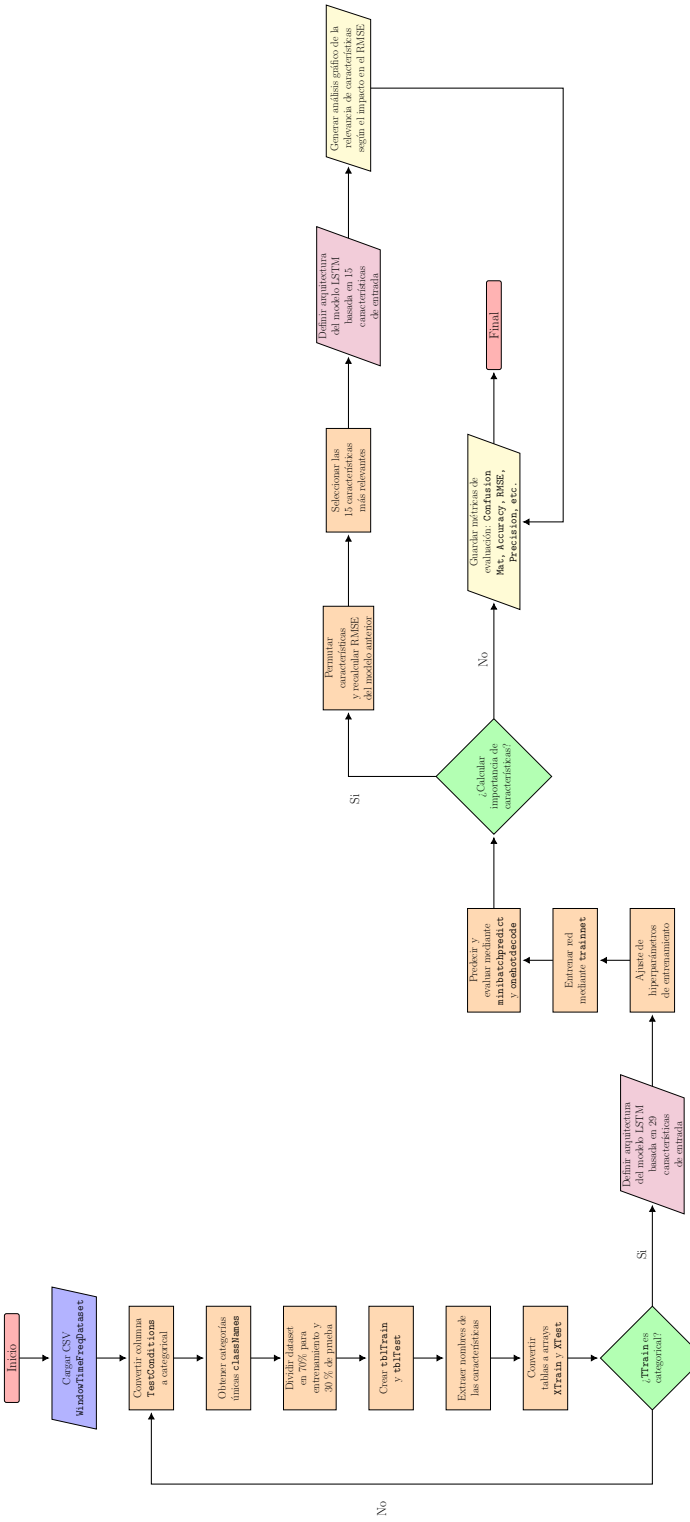
Figura 4.9: Arquitectura del modelo LSTM para clasificación multiclase basada en 15 características de entrada



Fuente: Elaboración propia

La Figura 4.10 presenta el diagrama de flujo de esta metodología, incluyendo el análisis de importancia de características, la selección de variables y la construcción del nuevo modelo.

Figura 4.10: Metodología de construcción, entrenamiento y evaluación de redes neuronales



Fuente: Elaboración propia

4.5.2. Métricas de evaluación

Para evaluar el rendimiento de un modelo neuronal de clasificación, una de las primeras métricas a generar es la *"Matriz de Confusión"*. Esta es una matriz cuadrada de dimensiones $N \times N$, donde N es el número de clases o categorías del problema a clasificar. Su propósito es comparar las predicciones del modelo con las etiquetas reales, permitiendo visualizar con claridad en qué casos el modelo acierta y en cuáles se equivoca.

Esta matriz se organiza como una tabla en la que:

- Las filas representan las clases reales (etiquetas verdaderas)
- Las columnas corresponden a las clases predichas por el modelo.

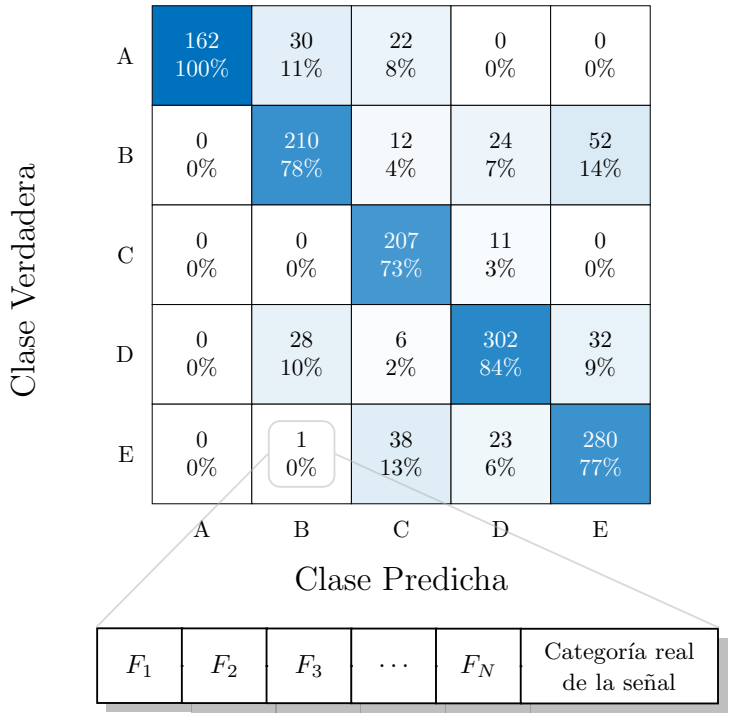
Cada celda de la matriz indica la cantidad de unidades o muestras que coinciden con una determinada combinación entre la clase real y la clase predicha. En un problema de clasificación binaria, esta disposición da lugar a cuatro categorías fundamentales:

a) **Verdadero positivo (TP):** Predicción correcta de un resultado positivo.
b) **Verdadero negativo (TN):** Predicción correcta de un resultado negativo.
c) **Falso positivo (FP):** Predicción incorrecta de un resultado positivo donde el resultado real es negativo.
d) **Falso negativo (FN):** Predicción incorrecta de un resultado negativo donde el resultado real es positivo.

Tal como se observa en la Figura 4.11, cada muestra del conjunto de datos se compone de una secuencia de N características cuantitativas, seguida por una etiqueta que indica la clase real a la que pertenece. El modelo LSTM procesa esta secuencia y, en función de los patrones que ha aprendido durante la etapa de entrenamiento, predice una única clase

para cada muestra. Esta predicción se registra como una unidad dentro de la matriz de confusión, permitiendo visualizar el desempeño del modelo: los aciertos se ubican sobre la diagonal principal (cuando la clase real y la predicha coinciden), mientras que los errores se representan fuera de ella (cuando hay discrepancia entre ambas).

Figura 4.11: Matriz de confusión multiclase



Fuente: Elaboración propia

A partir de esta matriz se derivan métricas que permiten evaluar el rendimiento del modelo de forma más específica:

1. **Accuracy:** Métrica que representa la proporción de predicciones correctas respecto al total de muestras evaluadas. Se utiliza comúnmente como una primera referencia del rendimiento general del modelo. Sin embargo, si el número de muestras para

cada clase no es igual, tiende a favorecer a la clase mayoritaria. En dichos casos, el modelo podría alcanzar una alta exactitud al clasificar todas las muestras bajo la clase dominante, mientras falla en identificar correctamente las clases minoritarias.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

2. **Precision:** Indica la proporción de verdaderos positivos sobre el total de predicciones clasificadas como positivas. Es una métrica útil para evaluar qué tan confiables son las predicciones positivas del modelo. *"De todos los casos que el modelo predijo como positivos, ¿cuántos realmente lo eran?"*.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

3. **Recall:** Mide la proporción de verdaderos positivos que fueron correctamente identificados respecto al total de positivos reales. Métrica que permite analizar la capacidad del modelo para detectar todos los casos positivos, lo que resulta fundamental en contextos donde las omisiones son críticas. *"De todos los casos que realmente eran positivos, ¿cuántos fueron identificados correctamente?"*.

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

4. **F1-Score**: Es la media armónica entre el precision y el recall. Resulta especialmente útil en contextos donde el número de muestras por clase es desbalanceado. El F1-Score penaliza tanto a los falsos positivos como a los falsos negativos, dando lugar a una evaluación más robusta del rendimiento. *"¿Qué tan equilibrado es el modelo entre identificar correctamente los positivos y evitar falsas alarmas?"*.

$$F1 - Score = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.5}$$

Aunque el RMSE no es una métrica habitual para problemas de clasificación, dentro de este contexto también se considera con el fin de observar la evolución del error general del modelo durante el entrenamiento. Del mismo modo, la pérdida de modelo (*Loss*) permite analizar el progreso de optimización del modelo a lo largo de las épocas, reflejando qué tan bien se adapta a los datos.

Las métricas obtenidas durante la etapa de entrenamiento del modelo son proporcionadas directamente por el entorno de MATLAB. Para la fase de evaluación, se emplea una función desarrollada por Preetham Manjunatha [46], la cual facilita el cálculo de las métricas de *Precision*, *Recall* y *F1-Score* a partir de la matriz de confusión generada por MATLAB.

Asimismo, se integra la técnica de Incrustación de Vecino Estocástico Distribuido en t *(t-SNE)* que reduce la dimensión de datos complejos, conservando sus relaciones de similitud. Funciona en dos etapas:

1. Calcula la probabilidad de que dos puntos sean similares: cuanto más parecidos son, mayor es esa probabilidad (usando la distancia euclidiana).
2. Intenta colocar esos mismos puntos en un espacio de menor dimensión (en 2D) de forma que las probabilidades de similitud se mantengan lo más parecidas posible.

La t-SNE minimiza una medida llamada *"Divergencia de Kullback − Leibler"* (KL), que cuantifica cuánto se parecen dos distribuciones de probabilidad. Si esta divergencia es mínima, significa que la estructura del espacio original se ha preservado bien en el nuevo espacio.

La visualización mediante t-SNE aplicada al segundo modelo LSTM permite observar la distribución de patrones tanto a la entrada como a la salida del modelo, con el objetivo de confirmar su capacidad para clasificar correctamente las muestras según su condición.

Para ello, se consideran únicamente las muestras de entrenamiento (es decir, `XTrain`) sin aplicar normalización para la visualización en la entrada, mientras que en la salida se incluyen tanto `XTrain` (70 %) y `XTest` (30 %), a fin de ofrecer una visión más completa del desempeño general del modelo. Esta visualización resulta útil para validar un rendimiento elevado, similar al obtenido en el primer modelo y complementar métricas como el *Accuracy*, que pueden puede variar en función al número de muestras por clase.

En la Tabla 4.6 se incluyen las características técnicas del hardware y software utilizado para la realización de las tres etapas que conforman este marco metodológico.

Tabla 4.6: Especificaciones del equipo empleado para pre-procesamiento

Especificaciones del equipo	
CPU	AMD Ryzen 9 7945HX (16 núcleos)
GPU	NVIDIA GeForce RTX 4070 Laptop (8GB GDDR6)
RAM	16 GB DDR5
Sistema Operativo	Windows 10 22H2
MATLAB Version	R2023b
Toolboxes	Deep Learning Toolbox, Signal Processing Toolbox
Tiempo estimado del entrenamiento	6 minutos (con aceleración del GPU)

Fuente: Elaboración propia

Capítulo 5

Resultados

5.1. Visualización de señales

En las Figuras 5.1 y 5.2 se muestra el comportamiento dinámico en el dominio temporal de las señales de vibración adquiridas en el extremo del motor (Drive End) y en el extremo del ventilador (Fan End), ambas registradas con una frecuencia de muestreo de 12 kHz. Se incluyen registros correspondientes a seis condiciones de operación: Ball Fault, Inner Fault, Outer Fault (medidas en las posiciones 1H, 3H y 6H)[2], así como la condición saludable (Healthy).

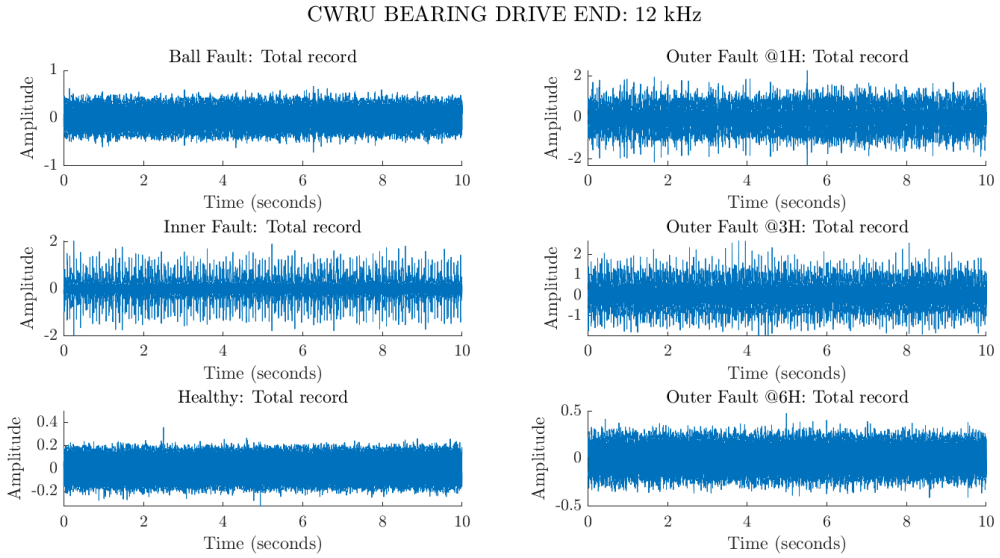
Tal como se aprecia en las señales del Drive End, es posible identificar diferencias cualitativas en la amplitud y el patrón de oscilación, lo cual permite una primera aproximación a la dinámica asociada a cada condición operativa. Sin embargo, estas diferencias se vuelven menos evidentes en las señales del Fan End, donde distintas

[2]A pesar de que Outer Fault representa una única falla, se considera como tres condiciones distintas debido a la posición del sensor y su posterior clasificación.

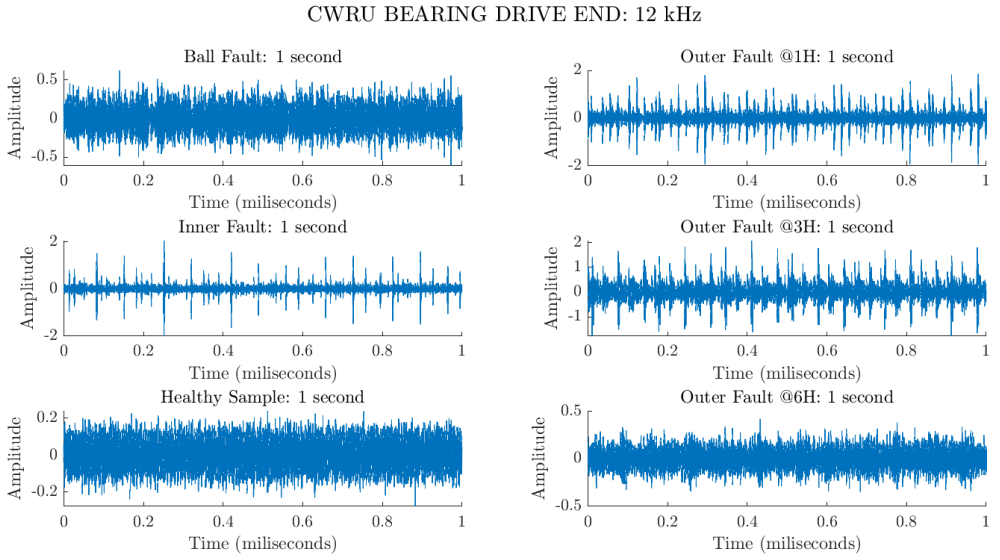
condiciones pueden presentar rangos de amplitud similares y patrones oscilatorios difícilmente distinguibles entre sí.

Para analizar el contenido espectral de las señales, se aplica la Transformada Rápida de Fourier (FFT) tanto a los registros del Drive End como del Fan End, obteniendo su representación en el dominio de la frecuencia (véase la Figura 5.3). Esta transformación permite identificar componentes frecuenciales dominantes, los cuales pueden estar asociados con defectos específicos en los rodamientos, lo cual resulta esencial para tareas de diagnóstico automático.

Figura 5.1: Señal Drive End en el tiempo



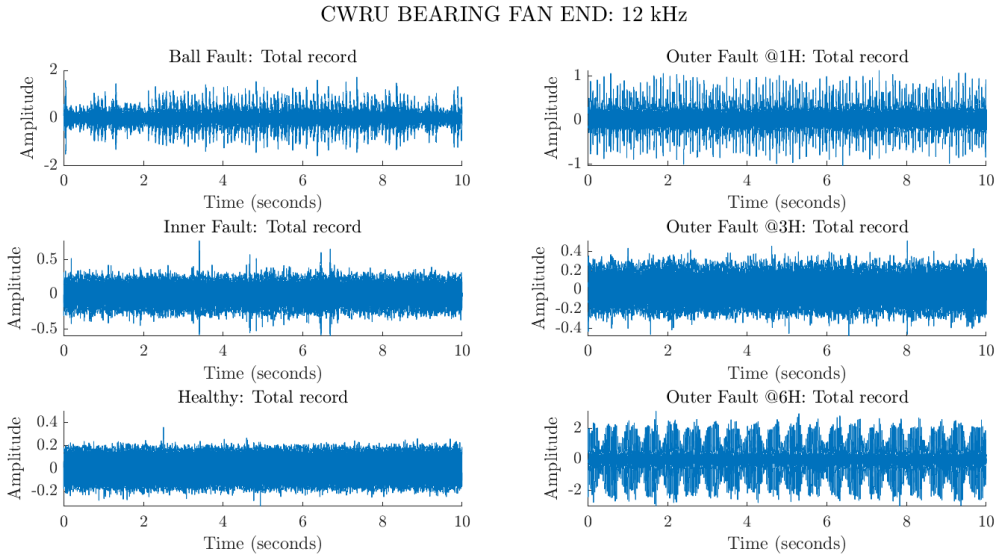
(a) Intervalo de 10 segundos



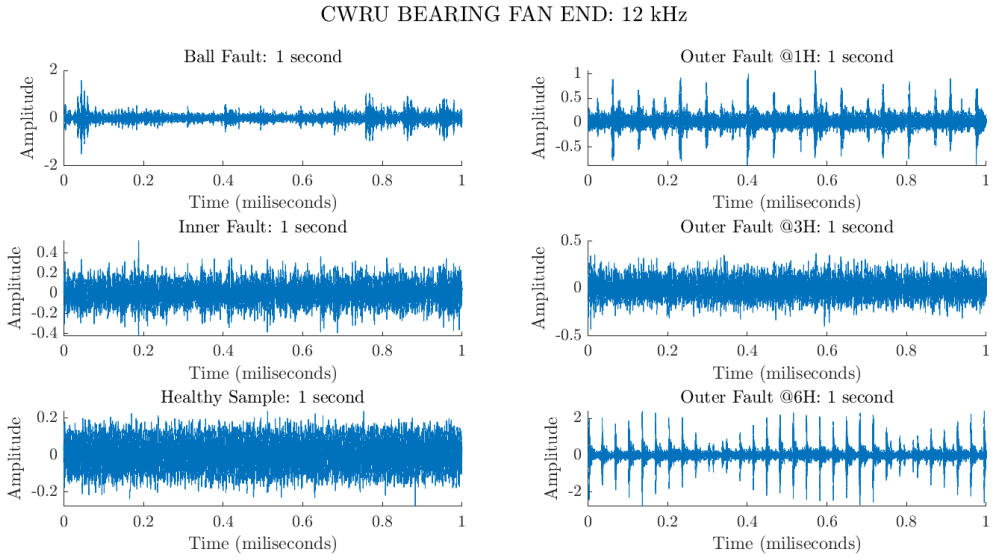
(b) Intervalo de 1 segundo

Fuente: Elaboración propia

Figura 5.2: Señal Fan End en el tiempo



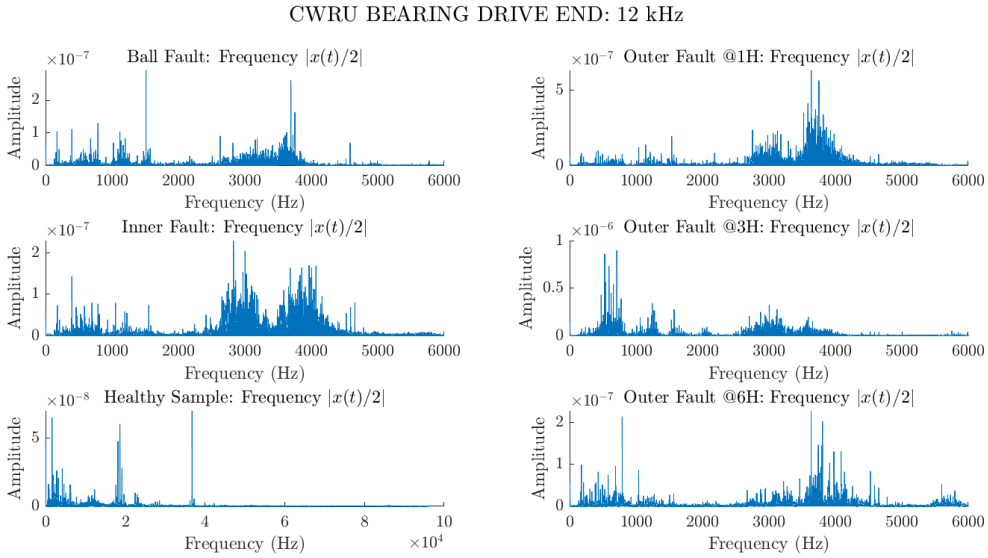
(a) Intervalo de 10 segundos



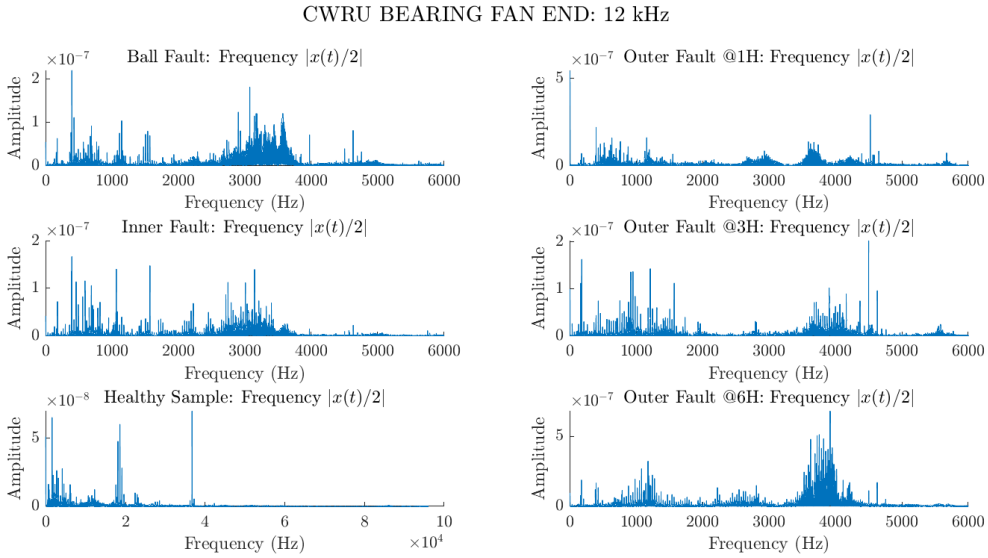
(b) Intervalo de 1 segundo

Fuente: Elaboración propia

Figura 5.3: Señales en el dominio de la frecuencia



(a) Frecuencias en Drive End



(b) Frecuencias en Fan End

Fuente: Elaboración propia

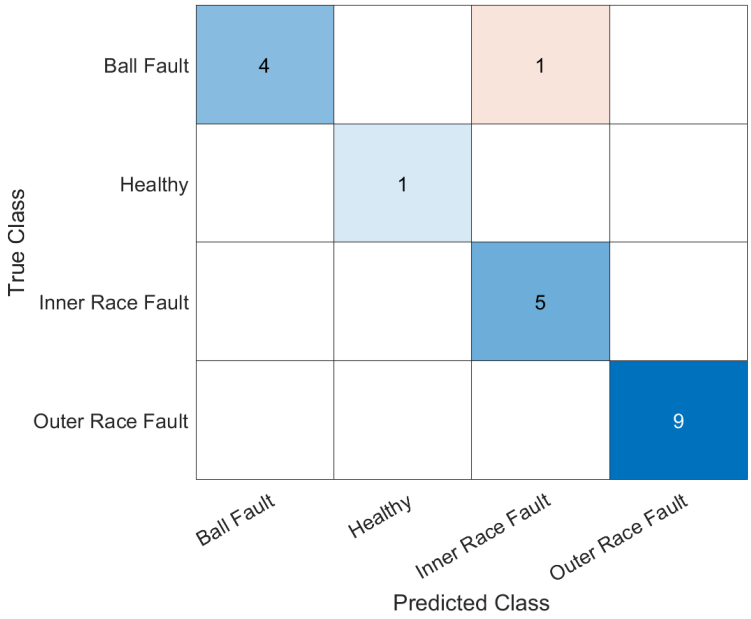
5.2. Resultados preliminares

El primer enfoque preliminar alcanzó una exactitud del 95 % al clasificar únicamente entre las señales DRIVE END y HEALTHY, como se muestra en la Figura 5.4. La arquitectura empleada se especifica en el título de dicha figura.

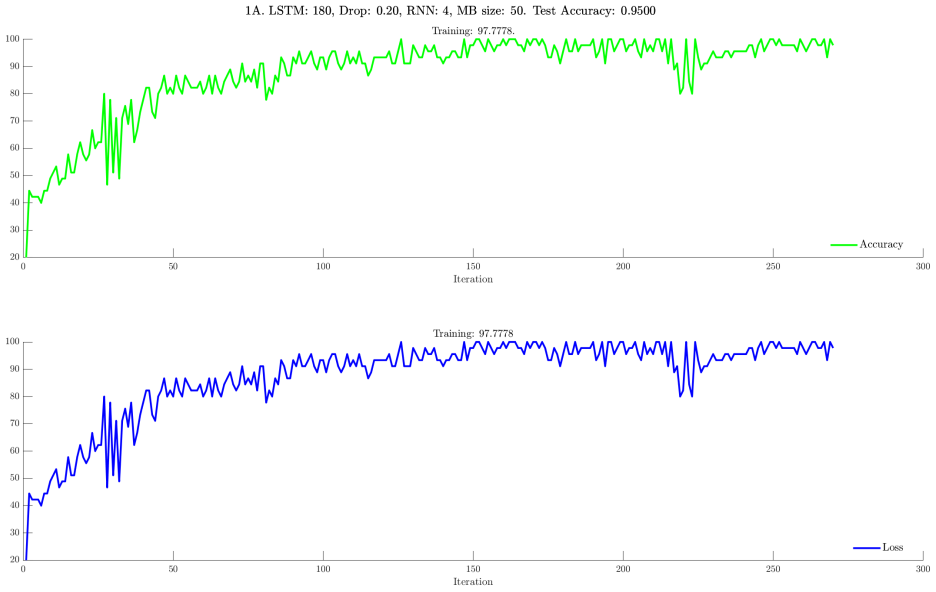
No obstante, al incorporar señales de la condición FAN END, el rendimiento del modelo se deterioró considerablemente (véase Figura 5.5). A pesar de ajustes significativos en la arquitectura, como el aumento del número de unidades LSTM, el tamaño del *MiniBatch* y el número de épocas de entrenamiento, no se logró mejorar sustancialmente su desempeño.

Esta inestabilidad se atribuye a la alta presencia de ruido en las señales provenientes del ventilador. La ubicación del sensor cerca del ventilador no es óptima para el diagnóstico de fallas en rodamientos, ya que el ruido aerodinámico generado por el flujo de aire turbulento puede enmascarar las frecuencias características de fallas. Además, esta ubicación se encuentra próxima a otras fuentes de vibración, como el desbalanceo, la acumulación de suciedad o daños en las aspas, las cuales introducen picos falsos o distorsiones en el análisis espectral al superponerse con las señales reales de defecto. Esto dificulta la detección precisa de defectos en su etapa temprana, ya que sus frecuencias características suelen tener baja amplitud y pueden quedar ocultas o mezcladas con el ruido proveniente de estas fuentes externas.

Figura 5.4: Clasificación DE & H con 4 características



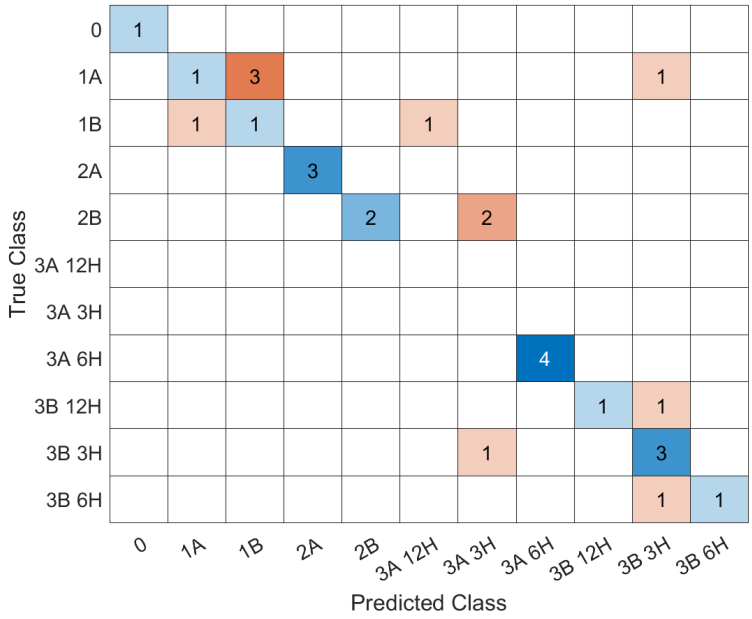
(a) Matriz de confusión (4 clases)



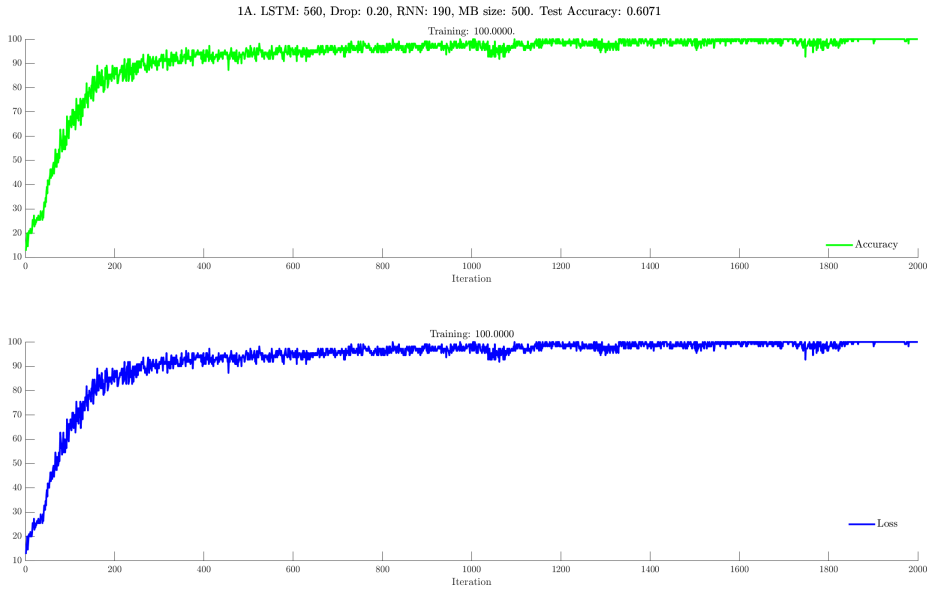
(b) Curvas de Accuracy y Loss

Fuente: Elaboración propia

Figura 5.5: Clasificación DE, FE & H con 4 características



(a) Matriz de confusión (11 clases)



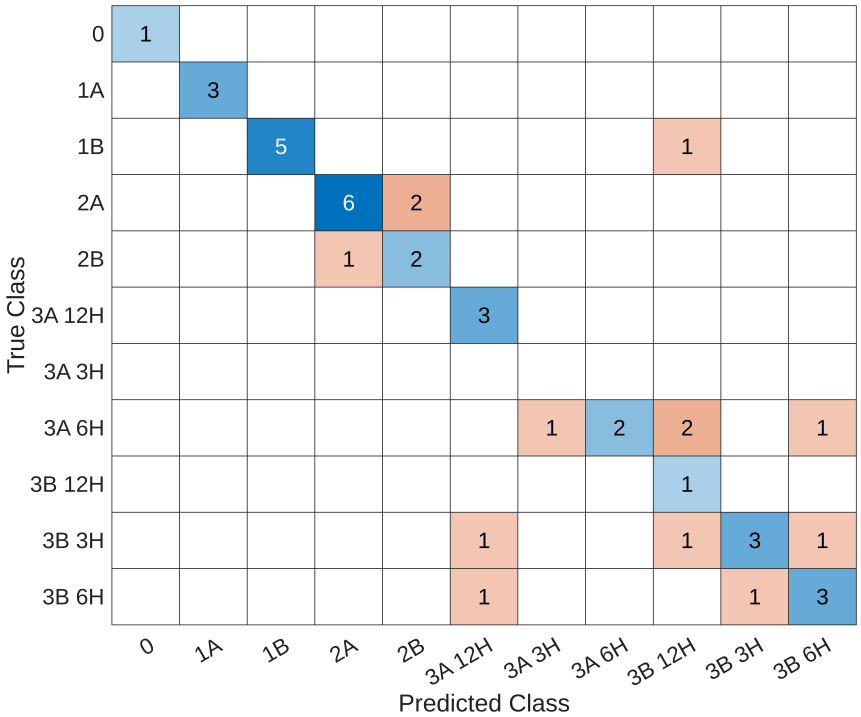
(b) Curvas de Accuracy y Loss

Fuente: Elaboración propia

En el segundo enfoque, se extrajeron 16 características a partir de 138 señales correspondientes a once condiciones de operación, generando un conjunto de datos de dimensiones 138×16. Sin embargo, la ausencia de segmentación y solapamiento impidió capturar información relevante a lo largo de la señal, incluso aplicando la función de ventana de Hamming. Esto derivó en una representación deficiente de las señales, generando un modelo inestable con resultados variables, oscilando entre un mínimo de 20 % y un máximo de 96 % de Accuracy en distintos entrenamientos.

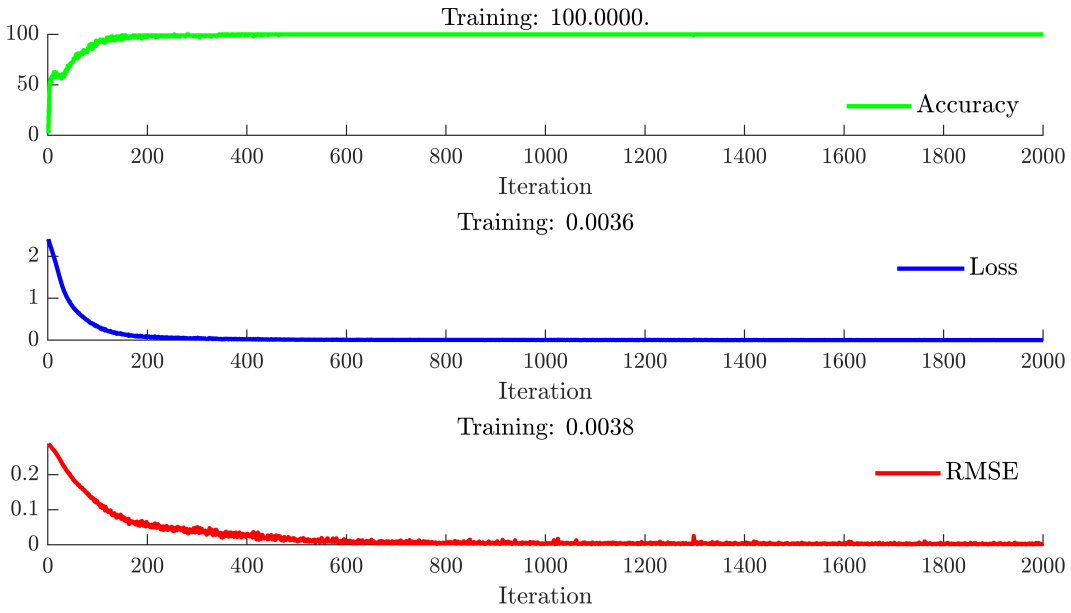
En la Figura 5.6c el valor correspondiente a *Training* representa el valor final de la métrica obtenida al concluir el entrenamiento del modelo. Por su parte, el valor de *Test* corresponde a la evaluación de dicha métrica aplicada al conjunto de prueba.

Figura 5.6: Clasificación DE, FE & H con 16 características

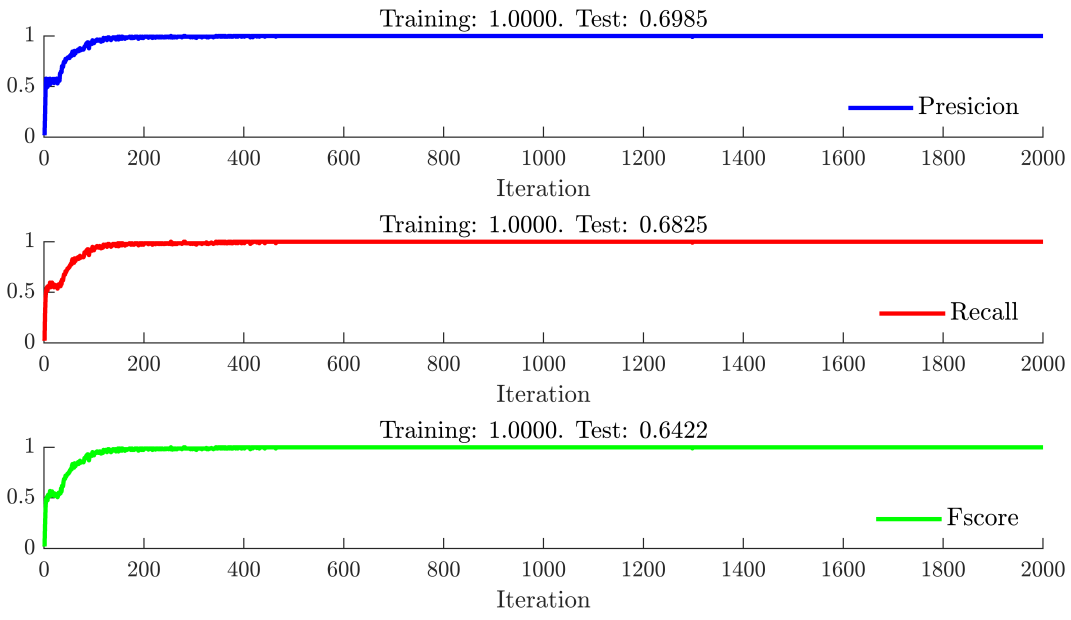


(a) Matriz de confusión (11 clases)

Fuente: Elaboración propia



(b) Curvas de Accuracy, Loss y RMSE

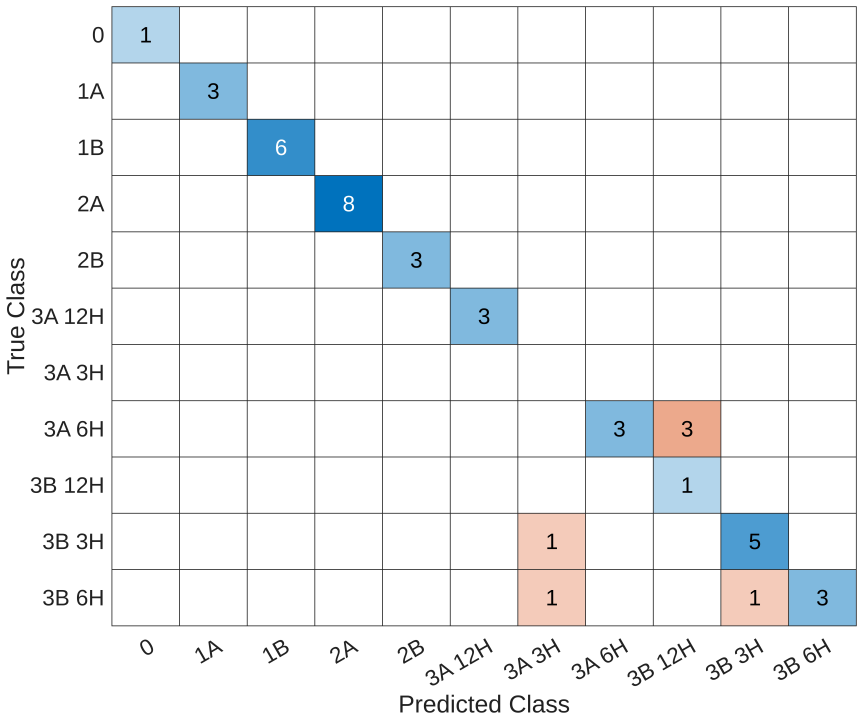


(c) Curvas de Precision, Recall y F1-Score

Fuente: Elaboración propia

Se incorpora además el análisis de importancia de características y la integración de un segundo modelo LSTM, utilizando siete características de entrada. A diferencia del primero, cuyo entrenamiento presenta cierta inestabilidad, este segundo modelo logró, en la mayoría de los casos, un desempeño igual o superior en términos de Accuracy. Los resultados sugieren una correlación positiva entre el desempeño del primer y segundo modelo, con una tendencia a mantener o superar la precisión. El primer modelo alcanzó un Accuracy del 69.05 % (Figura 5.6a), mientras que el segundo obtuvo un 85.71 % (Figura 5.7a). De forma adicional, la Figura 5.7d presenta las siete características más relevantes empleadas en este enfoque.

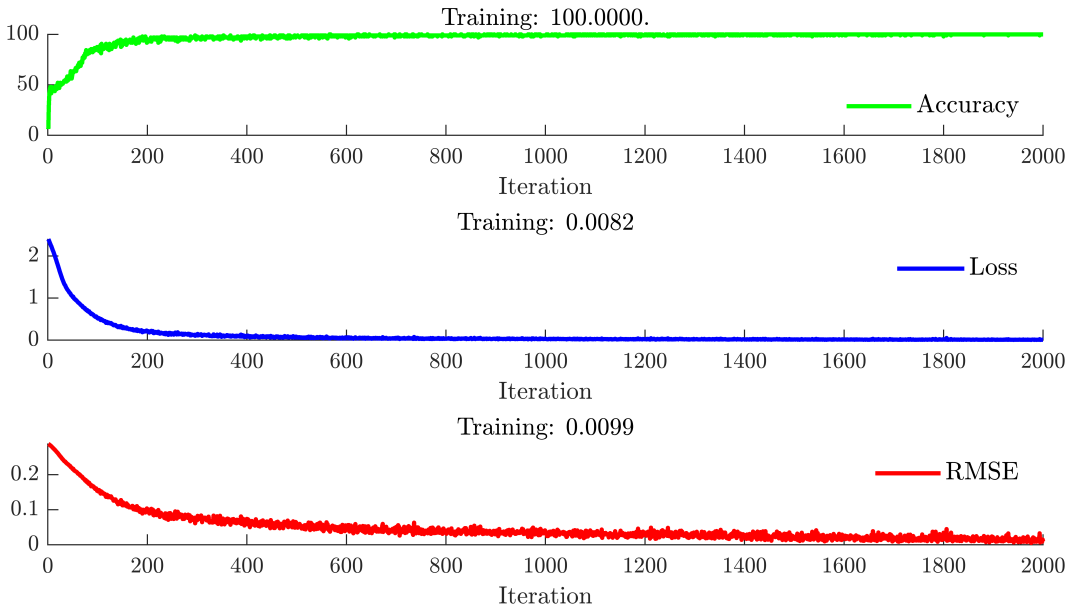
Figura 5.7: Clasificación DE, FE & H con 7 características seleccionadas



(a) Matriz de confusión (11 clases, 2do. modelo LSTM)

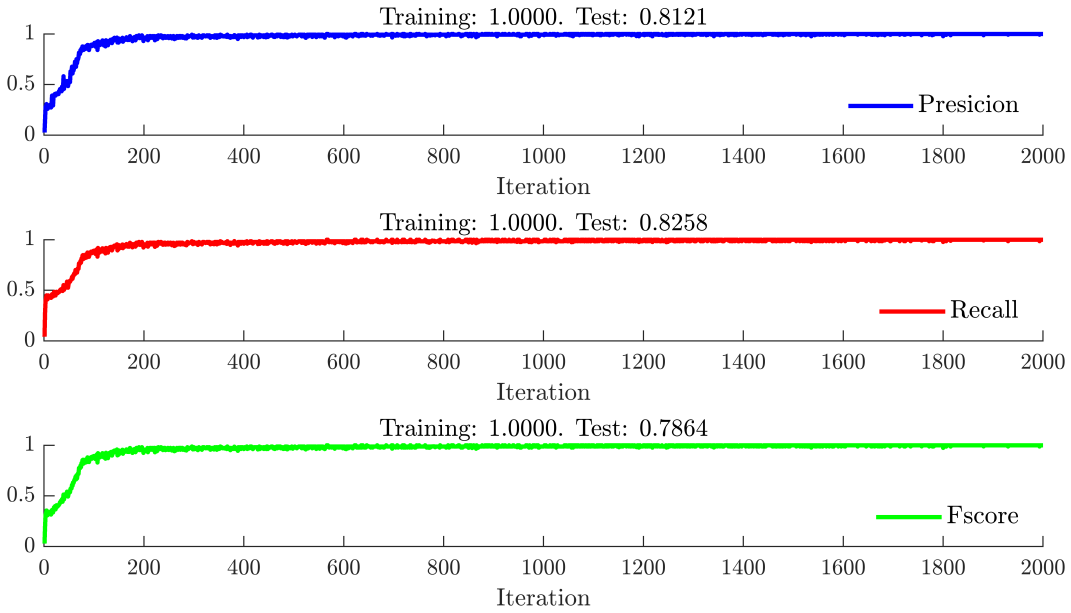
Fuente: Elaboración propia

2A. LSTM: 420, Drop: 0.30, RNN: 280, MB size: 400. Test Accuracy: 0.8571



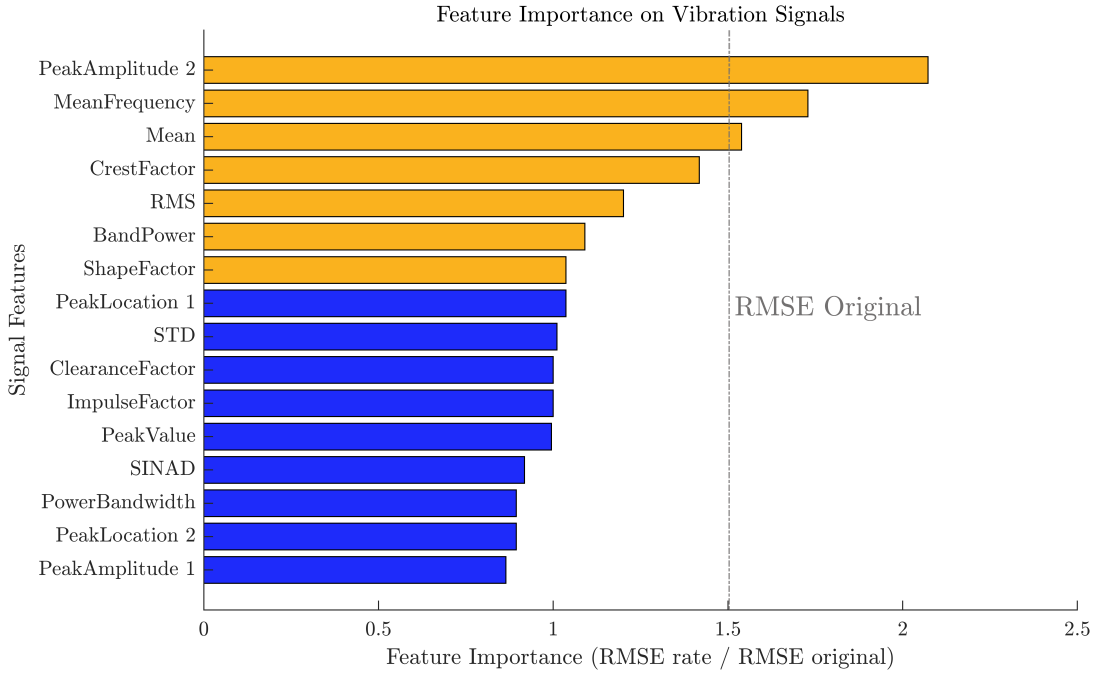
(b) Curvas de Accuracy, Loss y RMSE (2do. modelo LSTM)

2B. LSTM: 420, Drop: 0.30, RNN: 280, MB size: 400. Test Accuracy: 0.8571



(c) Curvas de Precision, Recall y F1-Score (2do. modelo LSTM)

Fuente: Elaboración propia



(d) 7 características más importantes ingresadas al 2do. modelo LSTM

Fuente: Elaboración propia

No obstante, a pesar del incremento en rendimiento al seleccionar las características más significativas, el modelo aún presenta inestabilidad. Esto se refleja en el RMSE presentado en la Figura 5.7d, donde el RMSE Original alcanza un valor ≥ 1.

Dado que la dimensionalidad del conjunto de datos ingresado a los dos modelos LSTM es baja, los tiempos de entrenamiento fueron relativamente cortos: el primer modelo, con 16 características, registró una duración aproximada de 65.05 segundos, mientras que el segundo, con 7 características, alcanzó los 60.80 segundos.

El bajo rendimiento observado en ambas metodologías evidencia la necesidad de un enfoque más robusto que ofrezca una mayor densidad informativa por señal y, por consecuencia, un aprendizaje más eficiente.

Estos resultados subrayan que las metodologías preliminares proporcionan una

representación parcial y limitada de la información contenida en las señales, insuficiente para un entrenamiento efectivo. En contraste, la metodología final se concibió como un sistema integral de procesamiento y representación de datos, capaz de capturar tanto la estructura temporal como espectral de las señales de vibración.

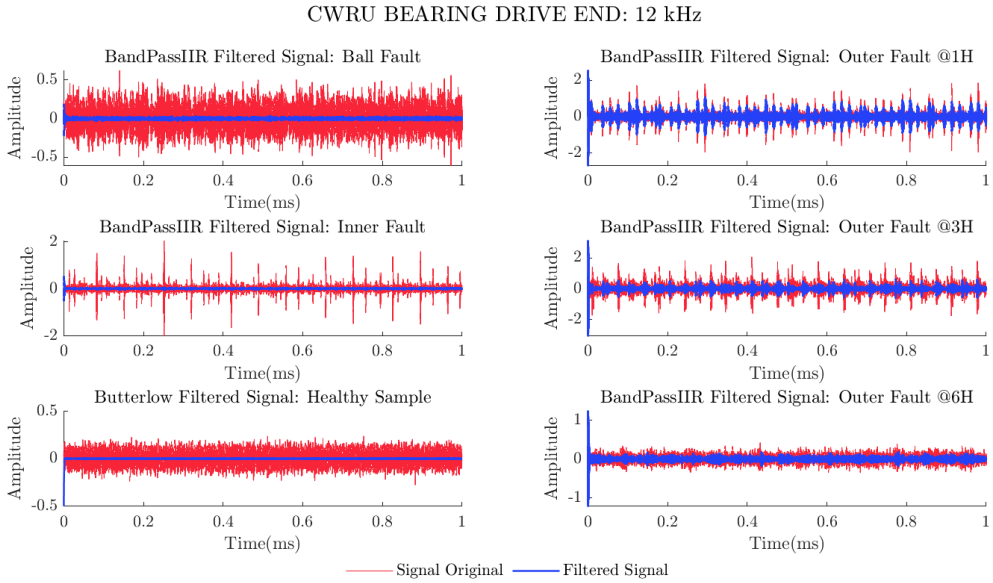
5.3. Filtrado de la señal

A fin de mejorar la calidad de la señal y atenuar componentes de alta frecuencia no relevantes, se aplica un filtro pasabanda IIR de orden 50° a todas las señales correspondientes a condiciones de falla.

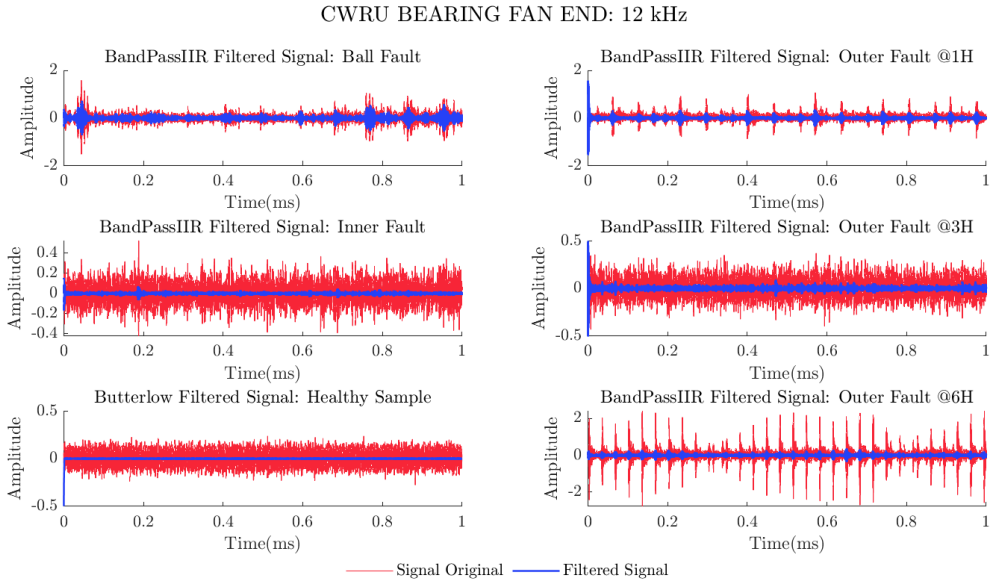
Para determinar las frecuencias de corte de los filtros pasabanda, se genera un vector discretizado de valores de RPM, que abarca desde el mínimo valor registrado (1718 rpm) hasta el máximo (1797 rpm). Este vector se divide en tres segmentos con el fin de calcular la frecuencia característica BPFO, correspondiente al nivel de carga en la falla Outer Fault. Se utiliza el mismo vector de RPM sin segmentar para el cálculo de las frecuencias características asociadas a los defectos BSF y BPFI. Las frecuencias centrales de cada tipo de falla se presentan en la Tabla 5.1 para Drive End y en la Tabla 5.2 para Fan End.

La Figura 5.8 ilustra el efecto del filtrado en señales de vibración para diferentes condiciones de fallo en rodamientos. Las señales filtradas permiten resaltar patrones característicos, como trenes de impulsos o resonancias mecánicas, que podrían estar superpuestas por el ruido en las señales originales. Particularmente, en fallas tipo Ball Fault y Outer Fault, el filtrado facilita la identificación visual de las recurrencias de daño. Además, la comparación entre Drive End y Fan End revela que la ubicación del sensor influye significativamente en la calidad de la señal captada, siendo el Drive End más sensible a las fallas en estos casos.

Figura 5.8: Filtrado en el dominio del tiempo (1 segundo)



(a) Filtrado en Drive End



(b) Filtrado en Fan End

Fuente: Elaboración propia

Tabla 5.1: Frecuencias características de fallo según RPM en el rodamiento Drive End

RPM	Ball	Inner	Outer1H	Outer3H	Outer6H	RPM	Ball	Inner	Outer1H	Outer3H	Outer6H	RPM	Ball	Inner	Outer1H	Outer3H	Outer6H
1718	8097.6958	9303.3556	0	0	6158.6444	1744	8220.2453	9444.1515	0	6251.8485	0	1771	8347.5083	9590.3625	6348.6375	0	0
1719	8102.4092	9308.7709	0	0	6162.2291	1745	8224.9587	9449.5667	0	6255.4333	0	1772	8352.2217	9595.7777	6352.2223	0	0
1720	8107.1227	9314.1861	0	0	6165.8139	1746	8229.6722	9454.9819	0	6259.0181	0	1773	8356.9352	9601.193	6355.807	0	0
1721	8111.8361	9319.6013	0	0	6169.3987	1747	8234.3856	9460.3971	0	6262.6029	0	1774	8361.6486	9606.6082	6359.3918	0	0
1722	8116.5495	9325.0165	0	0	6172.9835	1748	8239.0991	9465.8124	0	6266.1876	0	1775	8366.362	9612.0234	6362.9766	0	0
1723	8121.263	9330.4318	0	0	6176.5682	1749	8243.8125	9471.2276	0	6269.7724	0	1776	8371.0755	9617.4386	6366.5614	0	0
1724	8125.9764	9335.847	0	0	6180.153	1750	8248.526	9476.6428	0	6273.3572	0	1777	8375.7889	9622.8539	6370.1461	0	0
1725	8130.6899	9341.2622	0	0	6183.7378	1751	8253.2394	9482.058	0	6276.942	0	1778	8380.5024	9628.2691	6373.7309	0	0
1726	8135.4033	9346.6774	0	0	6187.3226	1752	8257.9528	9487.4733	0	6280.5267	0	1779	8385.2158	9633.6843	6377.3157	0	0
1727	8140.1168	9352.0926	0	0	6190.9074	1753	8262.6663	9492.8885	0	6284.1115	0	1780	8389.9293	9639.0995	6380.9005	0	0
1728	8144.8302	9357.5079	0	0	6194.4921	1754	8267.3797	9498.3037	0	6287.6963	0	1781	8394.6427	9644.5148	6384.4852	0	0
1729	8149.5436	9362.9231	0	0	6198.0769	1755	8272.0932	9503.7189	0	6291.2811	0	1782	8399.3561	9649.93	6388.07	0	0
1730	8154.2571	9368.3383	0	0	6201.6617	1756	8276.8066	9509.1342	0	6294.8658	0	1783	8404.0696	9655.3452	6391.6548	0	0
1731	8158.9705	9373.7535	0	0	6205.2465	1757	8281.5201	9514.5494	0	6298.4506	0	1784	8408.783	9660.7604	6395.2396	0	0
1732	8163.684	9379.1688	0	0	6208.8312	1758	8286.2335	9519.9646	0	6302.0354	0	1785	8413.4965	9666.1757	6398.8243	0	0
1733	8168.3974	9384.584	0	0	6212.416	1759	8290.9469	9525.3798	0	6305.6202	0	1786	8418.2099	9671.5909	6402.4091	0	0
1734	8173.1109	9389.9992	0	0	6216.0008	1760	8295.6604	9530.7951	0	6309.2049	0	1787	8422.9234	9677.0061	6405.9939	0	0
1735	8177.8243	9395.4144	0	0	6219.5856	1761	8300.3738	9536.2103	0	6312.7897	0	1788	8427.6368	9682.4213	6409.5787	0	0
1736	8182.5377	9400.8297	0	0	6223.1703	1762	8305.0873	9541.6255	0	6316.3745	0	1789	8432.3502	9687.8366	6413.1634	0	0
1737	8187.2512	9406.2449	0	0	6226.7551	1763	8309.8007	9547.0407	0	6319.9593	0	1790	8437.0637	9693.2518	6416.7482	0	0
1738	8191.9646	9411.6601	0	0	6230.3399	1764	8314.5142	9552.456	0	6323.544	0	1791	8441.7771	9698.667	6420.333	0	0
1739	8196.6781	9417.0753	0	0	6233.9247	1765	8319.2276	9557.8712	0	6327.1288	0	1792	8446.4906	9704.0822	6423.9178	0	0
1740	8201.3915	9422.4906	0	0	6237.5094	1766	8323.941	9563.2864	0	6330.7136	0	1793	8451.204	9709.4975	6427.5025	0	0
1741	8206.105	9427.9058	0	0	6241.0942	1767	8328.6545	9568.7016	0	6334.2984	0	1794	8455.9175	9714.9127	6431.0873	0	0
1742	8210.8184	9433.321	0	0	6244.679	1768	8333.3679	9574.1169	0	6337.8831	0	1795	8460.6309	9720.3279	6434.6721	0	0
1743	8215.5318	9438.7362	0	0	6248.2638	1769	8338.0814	9579.5321	0	6341.4679	0	1796	8465.3443	9725.7431	6438.2569	0	0
						1770	8342.7948	9584.9473	0	6345.0527	0	1797	8470.0578	9731.1584	6441.8416	0	0

Fuente: Elaboración propia

Tabla 5.2: Frecuencias características de fallo según RPM en el rodamiento Fan End

RPM	Ball	Inner	Outer1H	Outer3H	Outer6H	RPM	Ball	Inner	Outer1H	Outer3H	Outer6H	RPM	Ball	Inner	Outer1H	Outer3H	Outer6H
1718	6850.8299	9561.0834	0	0	5900.9166	1744	6954.5095	9705.7797	0	5990.2203	0	1771	7062.1768	9856.0412	6082.9588	0	0
1719	6854.8175	9566.6487	0	0	5904.3513	1745	6958.4971	9711.3449	0	5993.6551	0	1772	7066.1644	9861.6064	6086.3936	0	0
1720	6858.8052	9572.2139	0	0	5907.7861	1746	6962.4848	9716.9102	0	5997.0898	0	1773	7070.1521	9867.1717	6089.8283	0	0
1721	6862.7929	9577.7791	0	0	5911.2209	1747	6966.4725	9722.4754	0	6000.5246	0	1774	7074.1398	9872.7369	6093.2631	0	0
1722	6866.7806	9583.3444	0	0	5914.6556	1748	6970.4602	9728.0406	0	6003.9594	0	1775	7078.1275	9878.3021	6096.6979	0	0
1723	6870.7682	9588.9096	0	0	5918.0904	1749	6974.4479	9733.6059	0	6007.3941	0	1776	7082.1151	9883.8674	6100.1326	0	0
1724	6874.7559	9594.4749	0	0	5921.5251	1750	6978.4355	9739.1711	0	6010.8289	0	1777	7086.1028	9889.4326	6103.5674	0	0
1725	6878.7436	9600.0401	0	0	5924.9599	1751	6982.4232	9744.7364	0	6014.2636	0	1778	7090.0905	9894.9979	6107.0021	0	0
1726	6882.7313	9605.6053	0	0	5928.3947	1752	6986.4109	9750.3016	0	6017.6984	0	1779	7094.0782	9900.5631	6110.4369	0	0
1727	6886.719	9611.1706	0	0	5931.8294	1753	6990.3986	9755.8668	0	6021.1332	0	1780	7098.0659	9906.1283	6113.8717	0	0
1728	6890.7066	9616.7358	0	0	5935.2642	1754	6994.3862	9761.4321	0	6024.5679	0	1781	7102.0535	9911.6936	6117.3064	0	0
1729	6894.6943	9622.3011	0	0	5938.6989	1755	6998.3739	9766.9973	0	6028.0027	0	1782	7106.0412	9917.2588	6120.7412	0	0
1730	6898.682	9627.8663	0	0	5942.1337	1756	7002.3616	9772.5626	0	6031.4374	0	1783	7110.0289	9922.8241	6124.1759	0	0
1731	6902.6697	9633.4316	0	0	5945.5684	1757	7006.3493	9778.1278	0	6034.8722	0	1784	7114.0166	9928.3893	6127.6107	0	0
1732	6906.6573	9638.9968	0	0	5949.0032	1758	7010.337	9783.693	0	6038.307	0	1785	7118.0042	9933.9545	6131.0455	0	0
1733	6910.645	9644.562	0	0	5952.438	1759	7014.3246	9789.2583	0	6041.7417	0	1786	7121.9919	9939.5198	6134.4802	0	0
1734	6914.6327	9650.1273	0	0	5955.8727	1760	7018.3123	9794.8235	0	6045.1765	0	1787	7125.9796	9945.085	6137.915	0	0
1735	6918.6204	9655.6925	0	0	5959.3075	1761	7022.3	9800.3888	0	6048.6112	0	1788	7129.9673	9950.6503	6141.3497	0	0
1736	6922.608	9661.2578	0	0	5962.7422	1762	7026.2877	9805.954	0	6052.046	0	1789	7133.955	9956.2155	6144.7845	0	0
1737	6926.5957	9666.823	0	0	5966.177	1763	7030.2753	9811.5193	0	6055.4807	0	1790	7137.9426	9961.7807	6148.2193	0	0
1738	6930.5834	9672.3882	0	0	5969.6118	1764	7034.263	9817.0845	0	6058.9155	0	1791	7141.9303	9967.346	6151.654	0	0
1739	6934.5711	9677.9535	0	0	5973.0465	1765	7038.2507	9822.6497	0	6062.3503	0	1792	7145.918	9972.9112	6155.0888	0	0
1740	6938.5588	9683.5187	0	0	5976.4813	1766	7042.2384	9828.215	0	6065.785	0	1793	7149.9057	9978.4765	6158.5235	0	0
1741	6942.5464	9689.084	0	0	5979.916	1767	7046.2261	9833.7802	0	6069.2198	0	1794	7153.8933	9984.0417	6161.9583	0	0
1742	6946.5341	9694.6492	0	0	5983.3508	1768	7050.2137	9839.3455	0	6072.6545	0	1795	7157.881	9989.607	6165.393	0	0
1743	6950.5218	9700.2144	0	0	5986.7856	1769	7054.2014	9844.9107	0	6076.0893	0	1796	7161.8687	9995.1722	6168.8278	0	0
						1770	7058.1891	9850.4759	0	6079.5241	0	1797	7165.8564	10000.7374	6172.2626	0	0

Fuente: Elaboración propia

5.4. Segmentación de la señal

La Figura 5.9a ejemplifica la segmentación temporal en tres segmentos utilizando una ventana de Hanning, técnica comúnmente empleada para el preprocesamiento de señales. Esta ventana fuerza los extremos de la señal a cero en el dominio del tiempo, y en el dominio de la frecuencia atenúa los bordes de cada segmento, lo que reduce la aparición de discontinuidades espectrales. Este efecto contribuye a mejorar tanto el análisis espectral como el desempeño durante el entrenamiento de modelos supervisados.

De este modo, las características extraídas en el dominio de la frecuencia se aproximan más fielmente a las verdaderas frecuencias de defecto.

En las siguientes tablas se detallan las especificaciones técnicas del ventaneo individual aplicado a los primeros tres segmentos de la señal (Tabla 5.3), así como el ventaneo total aplicado al intervalo completo de 10 segundos (Tabla 5.4). Los resultados de ambas configuraciones se ilustran en la Figura 5.9.

Tabla 5.3: Especificaciones técnicas de ventaneo individual

Condiciones del ventaneo	Parámetros de configuración de ventaneo individual	En término de muestras de la señal	En término de tiempo de la señal
Parámetros de ventaneo	Longitud de ventana	1500 muestras	1.25 segundos
	Solapamiento (50%)	7500 muestras	
	Paso entre ventanas	7500 muestras	0.625 segundos
	Número de ventanas	3 ventanas	
Cobertura temporal	Primera ventana		0.00 segundos - 0.625 segundos
	Segunda ventana		0.625 segundos - 1.250 segundos
	Tercera ventana		1.250 segundos - 1.875 segundos
	Duración total cubierta		1.875 segundos
Eficiencia de cobertura	Porcentaje cubierto	20%	

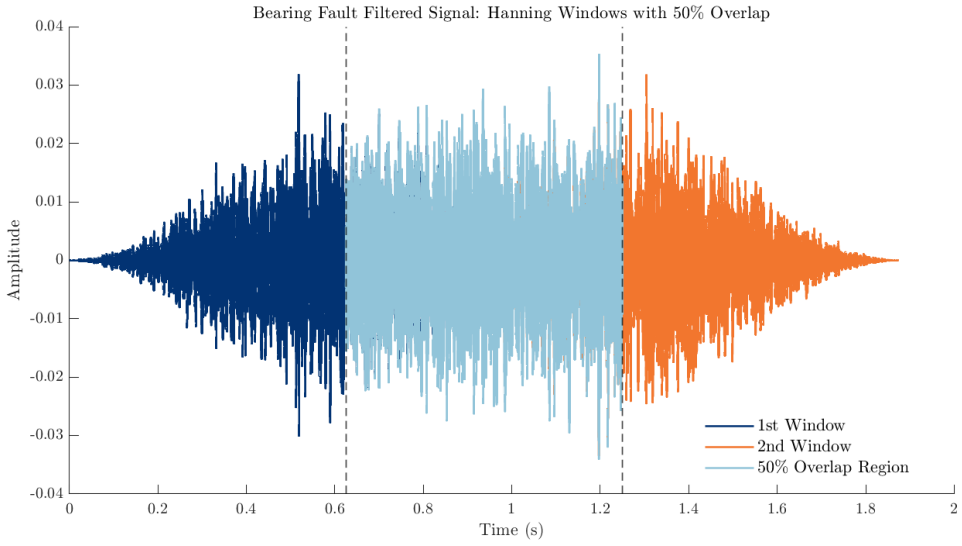
Fuente: Elaboración propia

Tabla 5.4: Especificaciones técnicas de ventaneo total

Condiciones del ventaneo	Parámetros de configuración de ventaneo total	En término de muestras de la señal	En término de tiempo de la señal
Información de la señal	Longitud total	12000 muestras	10 segundos
	Frecuencia de muestreo	12 kHz	
Parámetros de ventaneo	Longitud de ventana	15000 muestras	1.25 segundos
	Solapamiento (50%)	7500 muestras	
	Paso entre ventanas	7500 muestras	0.62 segundos
	Número total de ventanas	15 ventanas	
Cobertura temporal	Primera ventana	0.00 segundos - 1.25 segundos	
	Última ventana	8.75 segundos - 10.00 segundos	
	Duración total cubierta	10.00 segundos	
Eficiencia de cobertura	Porcentaje cubierto	100%	
	Tiempo no cubierto	0.00 segundos	

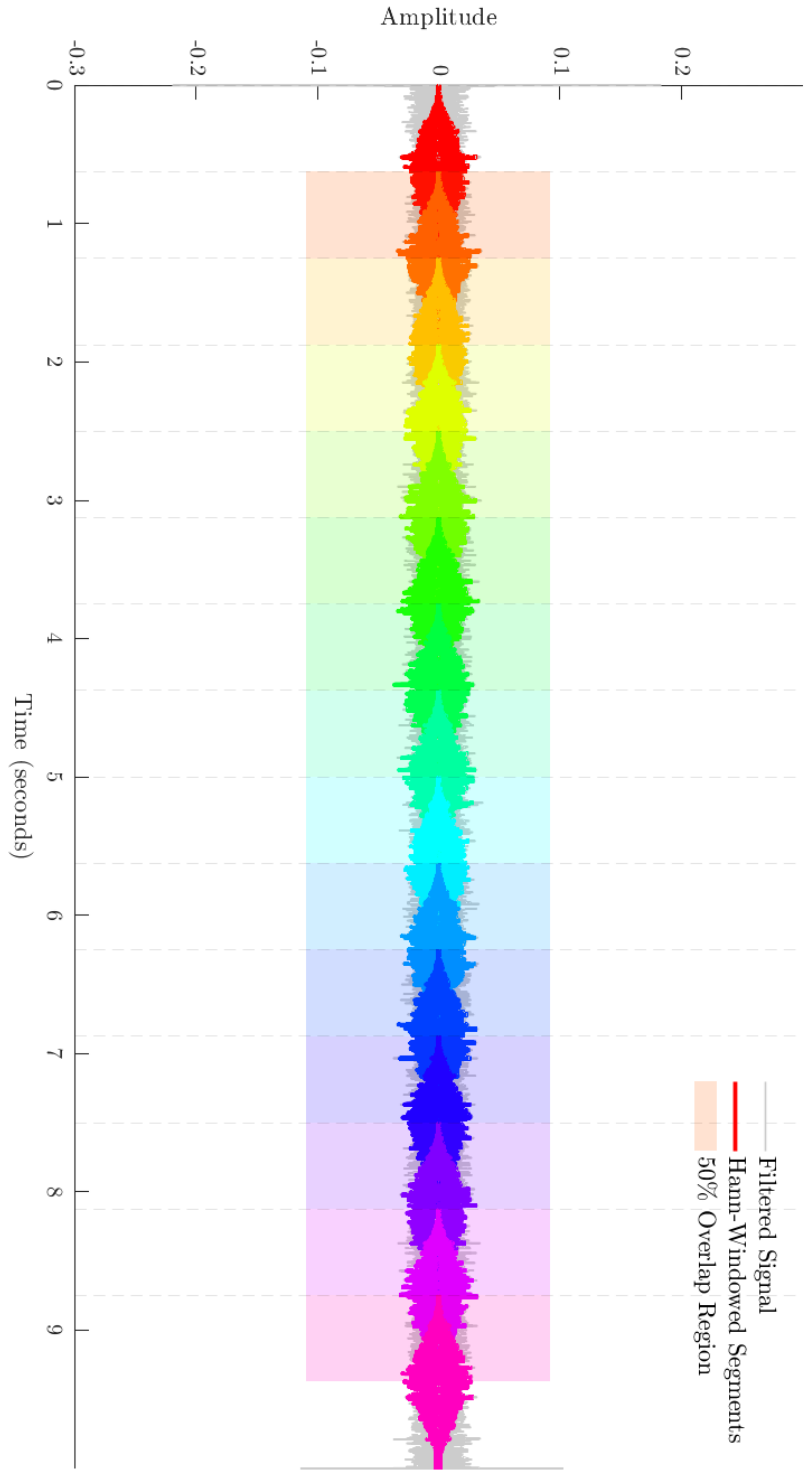
Fuente: Elaboración propia

Figura 5.9: Segmentación con ventanas Hanning



(a) Dos ventanas con 50 % de solapamiento

Fuente: Elaboración propia



(b) Conjunto de 15 ventanas en toda la señal

Fuente: Elaboración propia

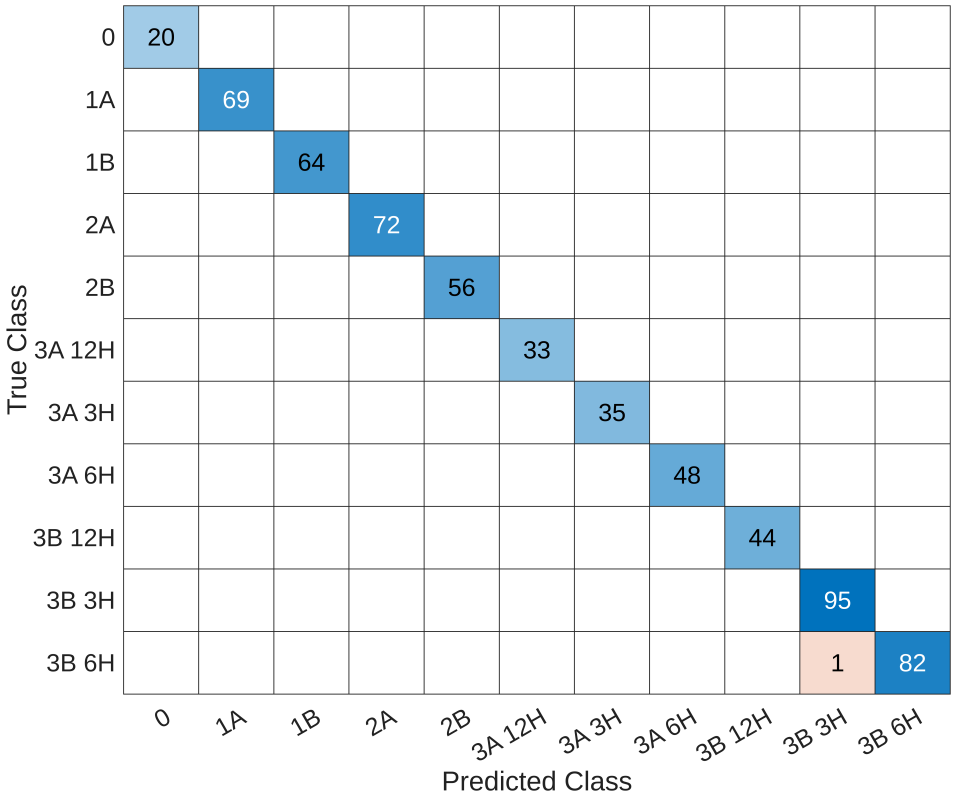
5.5. Clasificación de señales con la arquitectura final LSTM

Los resultados de clasificación de fallas en rodamientos, utilizando la arquitectura descrita en la Figura 4.8, constituyen el mejor desempeño alcanzado en esta investigación. El modelo logró una precisión (Accuracy) del 99.84 %, con un único falso positivo en la clase *3B 3H*. De un total de 619 secuencias evaluadas, 618 fueron clasificadas correctamente, como se muestra en la Figura 5.10a, lo cual refleja un rendimiento sobresaliente en la tarea de diagnóstico basado en características temporales y frecuencias.

Este primer modelo fue sometido a múltiples pruebas, obteniendo un rendimiento superior al 98 % de Accuracy en la totalidad de las pruebas, lo que permite concluir que esta arquitectura presenta una mayor estabilidad y fiabilidad en comparación con las redes propuestas en las metodologías preliminares. Este rendimiento del modelo se encuentra respaldado por los resultados mostrados en las Figuras 5.10b y 5.10c.

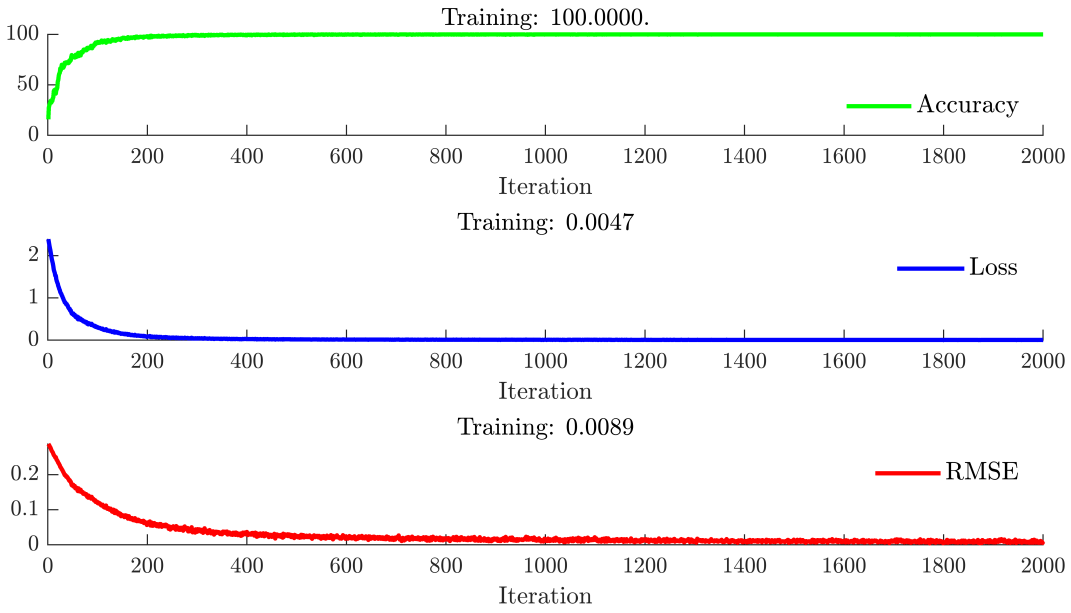
Se concluye que, a mayor densidad de información introducida a la red, mayor será su capacidad de aprendizaje y su desempeño frente a nuevas secuencias de evaluación. A modo de comparación, las metodologías anteriores ofrecían a la red únicamente una única capa de información temporal y espectral por cada una de las 138 señales y sus 11 clases. En contraste, la metodología actual proporciona un compendio más amplio y enriquecido de las mismas señales, permitiendo un aprendizaje más profundo.

Figura 5.10: Clasificación DE, FE & H con 29 características

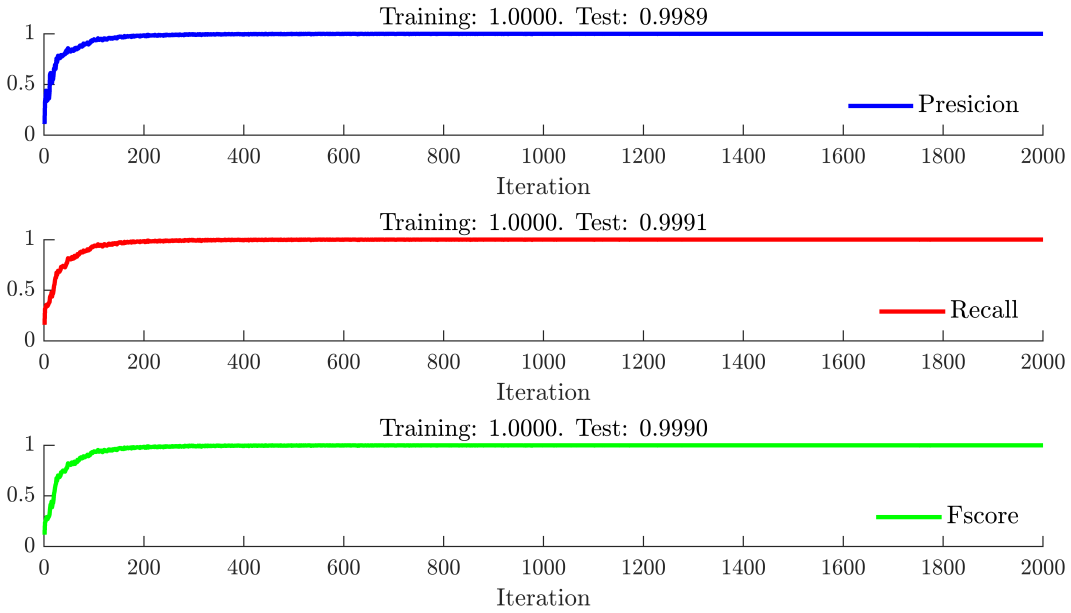


(a) Matriz de confusión (11 clases, 1er. modelo LSTM)

Fuente: Elaboración propia



(b) Curvas de Accuracy, Loss y RMSE (1er. modelo LSTM)



(c) Curvas de Precision, Recall y F1-Score (1er. modelo LSTM)

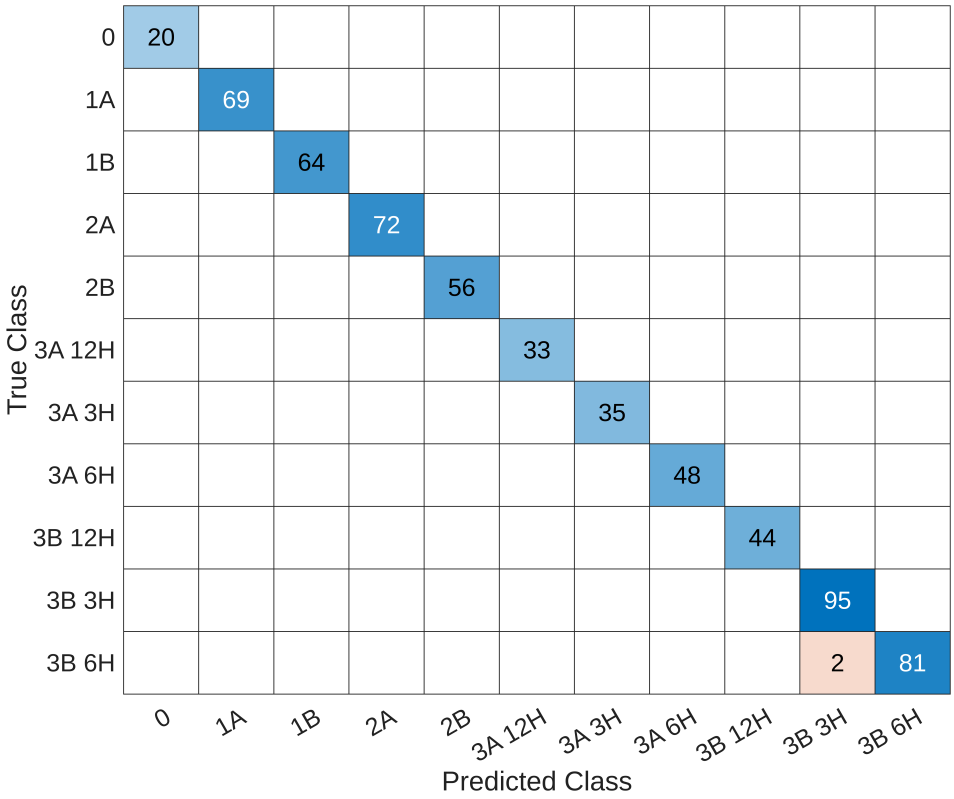
Fuente: Elaboración propia

Posteriormente, se aplica el análisis de importancia de características mediante el método de permutación con el objetivo de reducir la dimensionalidad del conjunto de datos. A partir de este análisis, se seleccionan las 15 características más relevantes en el proceso de aprendizaje de la primera red LSTM, las cuales se ilustran en la Figura 5.11d. Estas características marcadas en azul se utilizan como entrada para el segundo modelo LSTM.

El segundo modelo, descrito en la Figura 4.9, alcanzó el segundo mejor desempeño del presente trabajo, con un Accuracy del 99.68 %. Es decir que, con tan solo 15 características (temporales y frecuenciales), la red fue capaz de clasificar correctamente las 11 clases del conjunto de datos, con una diferencia mínima respecto a su predecesor (0.16 %), atribuida a dos falsos positivos en la misma clase de error del primer modelo.

Adicionalmente, el valor RMSE fue significativamente menor a 1, lo que confirma la estabilidad y consistencia de ambas arquitecturas.

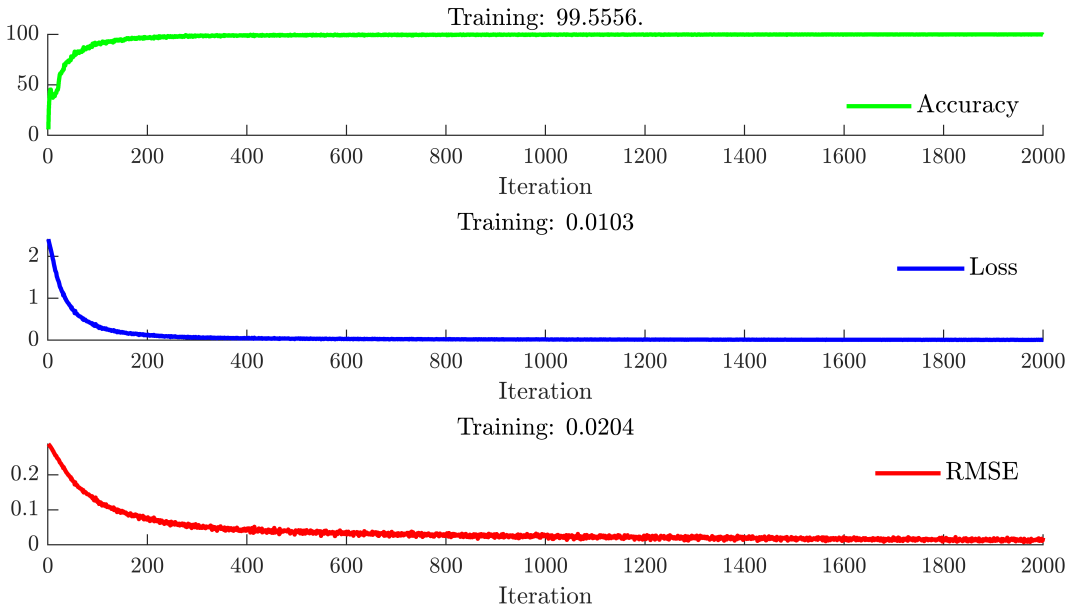
Figura 5.11: Clasificación DE, FE & H con 15 características seleccionadas



(a) Matriz de confusión (11 clases, 2do. modelo LSTM)

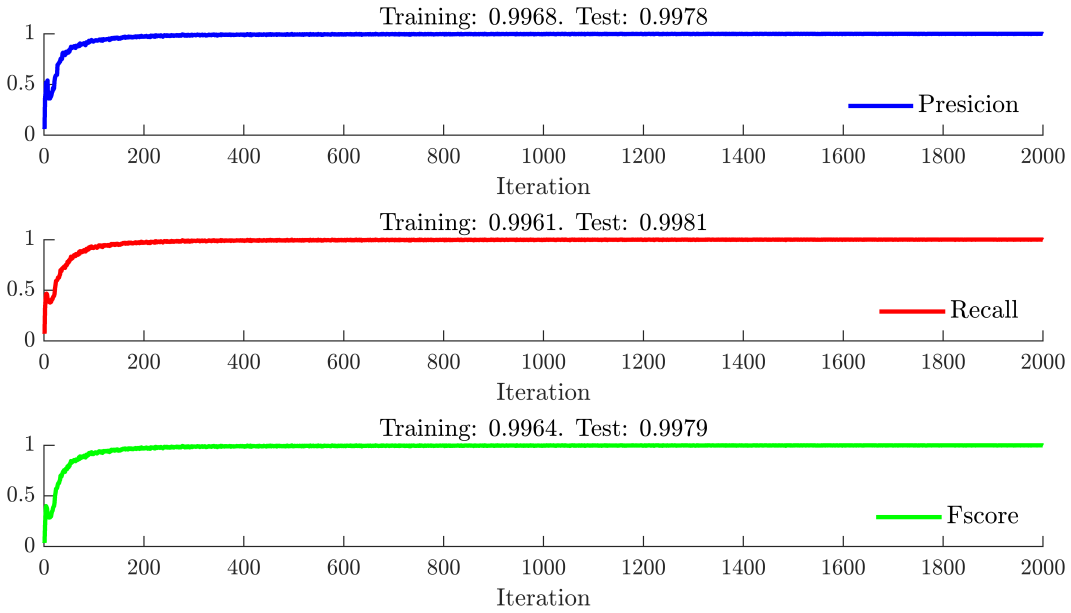
Fuente: Elaboración propia

2A. LSTM: 560, Drop: 0.20, RNN: 300, MB size: 900. Test Accuracy: 0.9968



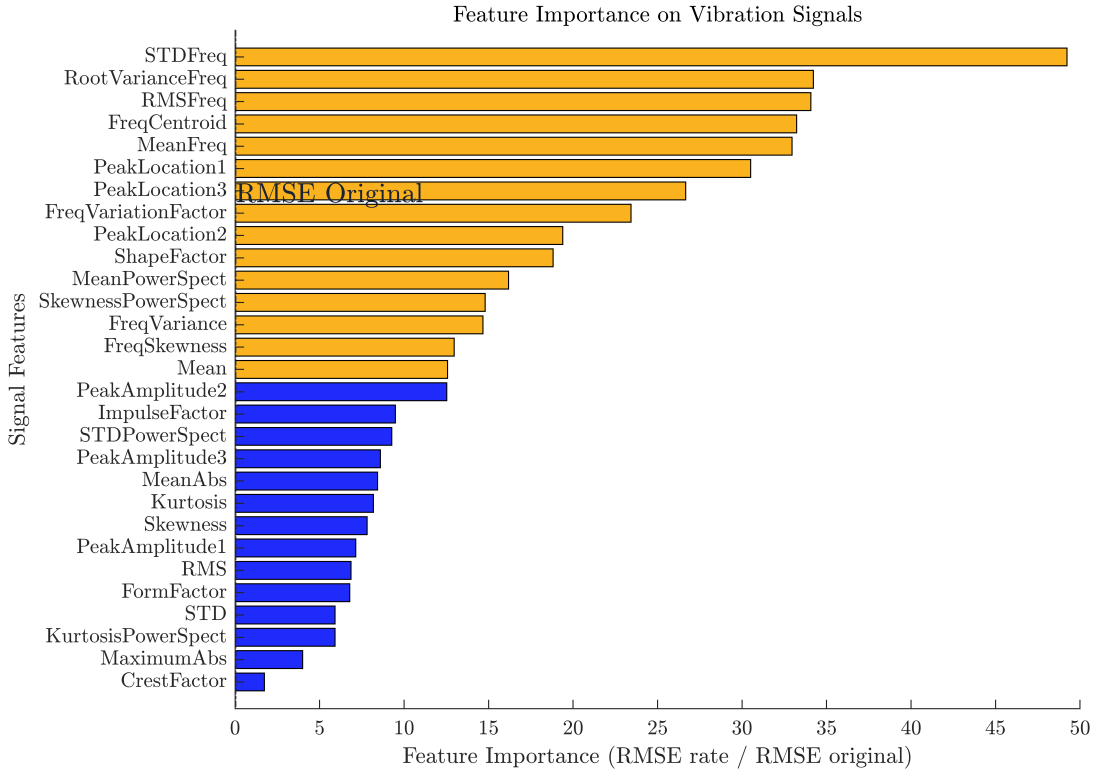
(b) Curvas de Accuracy, Loss y RMSE (2do. modelo LSTM)

2B. LSTM: 560, Drop: 0.20, RNN: 300, MB size: 900. Test Accuracy: 0.9968



(c) Curvas de Precision, Recall y F1-Score (2do. modelo LSTM)

Fuente: Elaboración propia



(d) 15 características más importantes ingresadas al 2do. modelo LSTM

Fuente: Elaboración propia

Finalmente, cabe destacar que, debido a una mayor dimensionalidad del conjunto de datos utilizados (2632×29 para el primer modelo y 2632×15 para el segundo), el tiempo de entrenamiento estimado para cada red LSTM fue de aproximadamente 6 minutos (360 segundos), lo cual se considera aceptable dado el volumen de información procesado por sus respectivas arquitecturas.

Es importante resaltar que el número de características extraídas (columnas del conjunto de datos) no guarda una relación directamente proporcional con el rendimiento del modelo; en cambio, este depende en mayor medida del número de muestras disponibles (filas). Reflejando que el rendimiento del modelo se ve más influenciado por el volumen de información disponible para el entrenamiento que por la

cantidad de características utilizadas.

Con el objetivo de validar el alto desempeño del segundo modelo LSTM, se emplea la técnica de reducción de dimensionalidad t-SNE, tal como se muestra en la Figura 5.12, donde cada punto representa una muestra y los colores corresponden a las distintas condiciones de operación. Cabe señalar que los componentes t-SNE 1 y 2 no poseen una interpretación directa, sino que representan una transformación lineal de los datos hacia un espacio bidimensional.

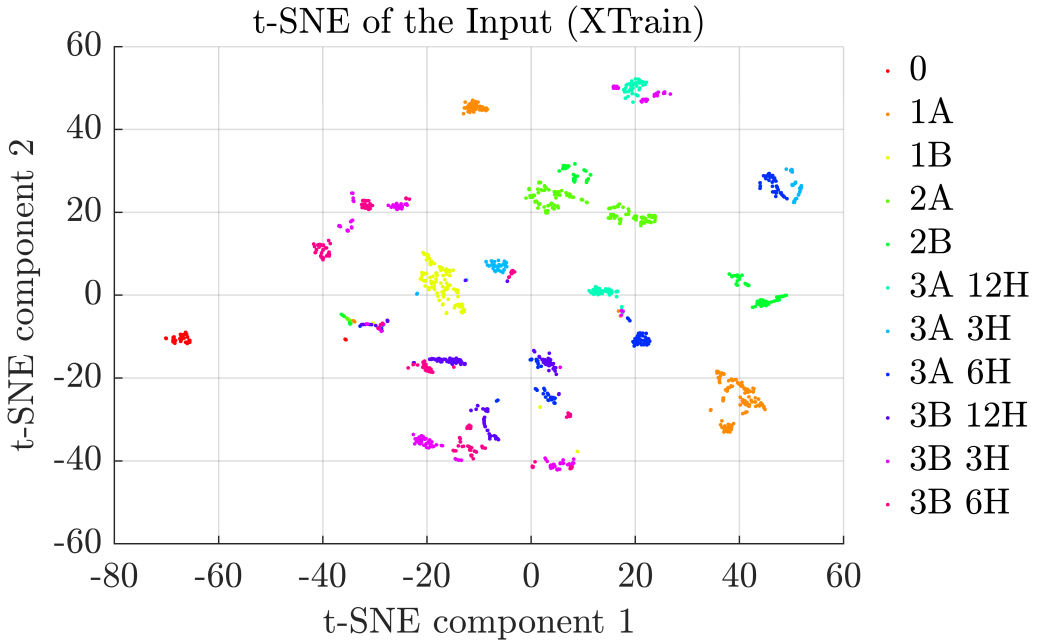
En la entrada del modelo (Figura 5.12a), se observa una separación evidente entre las clases, que incluyen las etapas temporales de fallo, en la pista externa (Outer Race: 3H, 6H, 12H), lo que señala que las características extraídas facilitan una correcta distinción entre las condiciones.

En la salida del modelo (Figura 5.12b), se confirma una agrupación compacta de muestras por clase. Las clases se encuentran adecuadamente separadas entre sí, lo que señala que los datos en su estado original poseen la suficiente información para diferenciar entre las condiciones correspondientes. Esto indica que el modelo ha conseguido adquirir representaciones internas que mantienen el equilibrio entre las condiciones operativas.

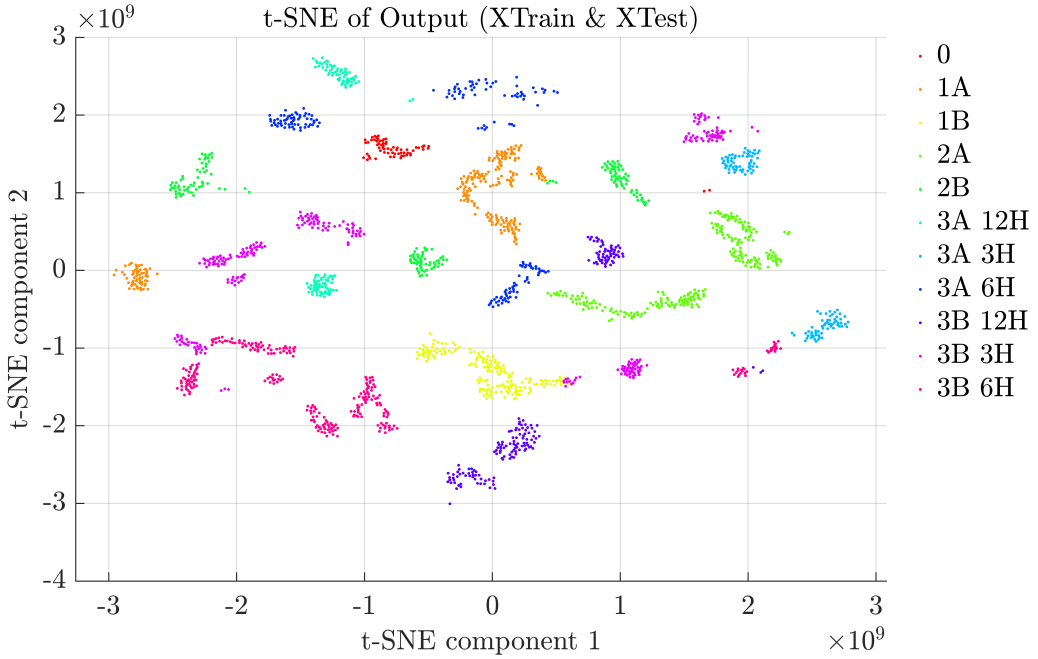
De esta forma, se deduce que el método t-SNE ofrece una validación visual efectiva del aprendizaje del modelo, evidenciando la habilidad del LSTM para distinguir entre condiciones de fallo.

En la Tabla 5.5 se resumen las metodologías aplicadas en los resultados presentados. Asimismo, se detallan las técnicas implementadas en cada caso, el número de características utilizadas para cada modelo LSTM (en caso de que la metodología contemple dos modelos) y los respectivos valores de precisión (Accuracy) obtenidos.

Figura 5.12: t-SNE a la entrada y salida de la red LSTM



(a) t-SNE a la entrada (2do. modelo LSTM)



(b) t-SNE a la salida (2do. modelo LSTM)

Fuente: Elaboración propia

Tabla 5.5: Resumen de las metodologías aplicadas y sus resultados

N° de secuencia	Etapa de la metodología	Técnicas implementadas	Series utilizadas	Nº de características hacia el 1er LSTM	Precisión del 1er LSTM (%)	Nº de características hacia el 2do LSTM	Precisión del 2do LSTM (%)
1a.	Preliminar	◦ Extracción de características.	◇ Drive End Series (12 kHz). ◇ Healthy Series (48 kHz).	4 características temporales	95	*N/A*	*N/A*
1b.	Preliminar	◦ Extracción de características.	◇ Drive End Series (12 kHz). ◇ Fan End Series (12 kHz). ◇ Healthy Series (48 kHz).	4 características temporales.	60.71	*N/A*	*N/A*
2a.	Preliminar	◦ Filtros Digitales. ◦ Aplicación de Hamming. ◦ Extracción de características. ◦ Análisis de importancia de características por permutación.	◇ Drive End Series (12 kHz). ◇ Fan End Series (12 kHz). ◇ Healthy Series (48 kHz).	16 características temporales y frecuenciales *.	69.05	7 características temporales y frecuenciales más relevantes.	85.71
3a.	**Final**	◦ Filtros Digitales. ◦ Segmentación de ventanas. ◦ Aplicación de Hanning. ◦ Extracción de características. ◦ Análisis de importancia de características por permutación.	◇ Drive End Series (12 kHz). ◇ Fan End Series (12 kHz). ◇ Healthy Series (48 kHz).	29 características temporales y frecuenciales †.	99.84	15 características temporales y frecuenciales más relevantes.	99.68

* incluyendo la amplitud y ubicación de los dos picos máximos en el dominio de la frecuencia.
† incluyendo la amplitud y ubicación de los tres picos máximos en el dominio de la frecuencia.

Fuente: Elaboración propia

Capítulo 6

Conclusiones

A continuación, se resaltan las partes relevantes encontradas en cada etapa de metodología:

Acerca de la base de datos CWRU Bearing Dataset

Respecto a la base de datos de la Universidad Case Western Reserve (CWRU), se reconoce que ofrece una cobertura adecuada respecto a los distintos tipos de fallas en rodamientos. No obstante, se identificó que las longitudes de las señales registradas no son uniformes entre todas las series, lo cuál, si bien es una característica común en el registro de señales, representa un inconveniente al procesar las mismas.

Aunque los archivos presentan un tamaño moderado en términos de almacenamiento, se observó que algunos conjuntos de datos, en particular, las series correspondientes al extremo del ventilador (Fan End), contienen un nivel significativo de ruido, representando una limitación al aplicar metodologías simples para el análisis de ambos rodamientos. Dificultando la evaluación directa del comportamiento de las fallas. Sin embargo, con el objetivo de abarcar la totalidad de los rodamientos disponibles en el banco de pruebas,

se consideraron ambas series en la presente investigación.

Se concluye que no toda la información disponible es adecuada para tareas de clasificación sin la implementación de técnicas de procesamiento sofisticadas como las que se exponen en este estudio. En ciertos casos, algunas señales de falla presentan características similares a otras señales de distinta clase, lo que puede provocar errores en los modelos de clasificación y perjudicar el desempeño del modelo.

Acerca del preprocesamiento de la señal y extracción de características

Se evidenció que, el uso exclusivo de características en el dominio del tiempo resulta insuficiente para lograr un reconocimiento de fallas eficiente al considerar más de un rodamiento en el proceso de clasificación. Sin embargo, en el caso particular del rodamiento ubicado en el extremo del rotor (Drive End), las características temporales son adecuadas para obtener un nivel de reconocimiento aceptable, debido al bajo nivel de ruido presente en las series.

La aplicación de filtros digitales pasabanda constituye una estrategia eficaz para restringir el amplio espectro de la señal a un rango específico centrado en las frecuencias asociadas a fallas características. Esta acción permite atenuar significativamente el ruido proveniente de componentes espectrales no relacionadas con los defectos, especialmente aquellas alejadas de la frecuencia central de interés.

Al realizar el análisis de señales en conjunto, tanto de las series en Drive End como de las series en Fan End, se observó que la adición de las características en el dominio frecuencial permite una mejora significativa en el rendimiento del modelo de clasificación, alcanzando un incremento de hasta un 9 %. Asimismo, la aplicación de la ventana de

Hanning resultó más eficaz para obtener una representación espectral más precisa, dado que atenúa los lóbulos laterales y reduce la amplitud del lóbulo principal, concentrando la energía en la frecuencia dominante. En contraste, la ventana de Hamming no impone que la señal tome un valor de cero en sus extremos, es decir, no fuerza a cero el inicio ni final del segmento analizado. Esta falta de suavizado en los bordes genera discontinuidades que, al transformarse en el dominio de la frecuencia, pueden traducirse en una mayor fuga espectral, incrementando la presencia de ruido y elevando la entropía en la representación espectral. Además, su menor atenuación en los lóbulos laterales y mayor anchura del lóbulo central contribuyen a una menor concentración en la frecuencia central.

Se debe resaltar que la función nativa de extracción de característica de MATLAB, aplica la ventana de Hamming sobre la totalidad de la señal sin segmentación ni solapamiento. Esto limita la capacidad de atenuación de bordes y puede inducir distorsiones espectrales. En cambio, el enfoque adoptado en este trabajo consistió en segmentar la señal en ventanas de tiempo más pequeñas, lo que permitió aplicar la función de Hanning de manera localizada a cada segmento y solapar las ventanas en un 50 %. Esta estrategia garantiza que cada fragmento de señal inicie y termine en cero en el dominio del tiempo, reduciendo el efecto de discontinuidades en el análisis de Fourier. La segmentación se consolidó así como una técnica eficaz para mejorar la caracterización de las señales, incrementar el volumen de información útil para el entrenamiento y evaluación de la red LSTM.

Acerca de la clasificación de las señales mediante la red LSTM

La utilización del Toolbox de Deep Learning de MATLAB representó una ventaja significativa al automatizar la construcción de arquitecturas de redes LSTM, eliminando

la posibilidad de cometer errores derivados de la programación manual capa por capa. Esta herramienta no solo facilita el diseño del modelo, sino que también optimiza el flujo de trabajo, permitiendo centrarse en el ajuste de los hiperparámetros y el análisis de resultados.

El aumento notable en el rendimiento observado en la tercera metodología puede atribuirse, en gran medida, a la segmentación de la señal en ventanas, que permite una mayor caracterización local de las señal, y por tanto, incrementa el volumen de datos disponibles para el entrenamiento del modelo.

Dentro del proceso de entrenamiento, se identificaron parámetros clave que influyen directamente en el desempeño del modelo, tales como la relación entre las unidades LSTM y las unidades RNN, el tamaño del *mini-batch* y el número de épocas. En particular, se observó que la cantidad de unidades LSTM y RNN no guarda una proporción directa, por lo que un ajuste abrupto o desbalanceado entre ambas puede derivar en un bajo rendimiento del modelo. Por esta razón, fue necesario realizar un ajuste manual y progresivo, mediante el cual se logró identificar un rango óptimo que equilibrara el desempeño del modelo, el tiempo de entrenamiento y la complejidad computacional asociada al número de unidades en cada capa. Por otra parte, se observó que un aumento en el tamaño del *mini-batch* tiende a suavizar la función de pérdida, favoreciendo un descenso en la función de pérdida de forma más suave, sin cambios abruptos o estancamientos. De igual forma, la reducción de la tasa de aprendizaje inicial *InitialLearnRate* puede beneficiar la precisión del entrenamiento, aunque esto requiere compensarse con un mayor número de épocas, ya que una tasa más baja implica un aprendizaje más lento.

El tiempo de entrenamiento estimado para esta metodología, bajo los hiperparámetros empleados en el modelo con mejor desempeño, oscila entre los 6 a 7 minutos. No obstante,

este tiempo puede extenderse hasta los 10 minutos si se incrementa el número de unidades LSTM y RNN.

Finalmente, una ventaja adicional del uso del entorno de MATLAB radica en la posibilidad de especificar el entorno de ejecución (CPU o GPU), lo cual influye de manera directa en la reducción del tiempo de entrenamiento, siendo especialmente relevante para modelos de mayor complejidad.

Bibliografía

[1] A. Rockwell, *The History of Artificial Intelligence*, 2017.

[2] M. Lefkowitz, *Professor's Perceptron Paved the Way for AI – 60 Years Too Soon*, 2019.

[3] H. Sheikh, C. Prins y E. Schrijvers, "Mission AI: The New System Technology," en *Springer*, 2023, págs. 1-385.

[4] K. H. Choy, "Neuro Symbolic Artificial Intelligence Pioneer to Overcome the Limits of Machine Learn," Research Paper, Arizona State University, 2021, págs. 1-23.

[5] E. Castillo, J. M. Gutiérrez y A. Hadi, *Sistemas Expertos y Modelos de Redes Probabilísticas*. Madrid: Monografías de la Academia de Ingeniería, 1997, pág. 639.

[6] W. B. Rauch-Hindin, *Aplicaciones de La Inteligencia Artificial En La Actividad Empresarial La Ciencia y La Industria*, Díaz de Santos, ed. 1989, pág. 644.

[7] C. Olah, *Understanding LSTM Networks*, 2015.

[8] L. Lahoud, *Forever Artificial Intelligence Summer*, 2024.

[9] J. de la Torre, "Redes Generativas Adversarias (GAN) Fundamentos Teóricos y Aplicaciones," *arXiv*, 2023.

[10] M. Haenlein y A. Kaplan, "A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence," *California Management Review*, vol. 61, n.º 4, págs. 5-14, ago. de 2019. visitado 8 de jul. de 2025.

[11] M. Comunale, "The Economic Impacts and the Regulation of AI: A Review of the Academic Literature and Policy Actions," *IMF Working Papers*, vol. 2024, n.º 065, pág. 1, 2024.

[12] D. Acemoglu et al., "The Simple Macroeconomics of AI," *National Bureau of Economic Research*, págs. 43-46, mayo de 2024.

[13] T. Wheeler, *The Three Challenges of AI Regulation*, 2023. visitado 28 de sep. de 2024.

[14] A. Bosch Rue, J. Casas Roma y T. Lozano Bagen, *Deep learning: principios y fundamentos*, UOC. Barcelona: Editorial UOC, 2019, pág. 259.

[15] C. A. Perez Estudillo et al., "Apuntes Sobre El Cerebro Y El Sistema Nervioso," *Society for Neuroscience*, M. Millar, T. Bentsen, D. D. Clendenning, S. Harris y D. Speert, eds., págs. 1-16, 2011.

[16] M. Megías, "Tipos Celulares," *Universidad de Vigo*, págs. 4-11, nov. de 2023.

[17] F. Tanco, D. Grinberg, J. Roitman y C. Verrastro, "Implementación de Redes Neuronales Artificiales En Hardware Para Aplicación En Detección Automática de Fulguraciones Solares," Universidad Tecnológica Nacional, Buenos Aires, inf. téc. November, 2014, pág. 2.

[18] C. F. Orozco Solis, "Integración y Desarrollo de Un Nuevo Modelo de Perceptrón Multicapa Para La Clasificación de Patrones No Lineales.," Tesis doct., Tecnológico Nacional de México, 2022, pág. 30.

[19] J. Velasco Rebolledo, *Machine Learning: fundamentos, algoritmos y aplicaciones para los negocios, industria y finanzas*. Diaz de Santos, 2024, pág. 252.

[20] S. Haykin, "Neural Networks and Learning Machines," en *Neural Networks*, A. Dworkin, ed., 3rd ed., vol. 1–3, Hamilton, Ontario: Pearson Education Ltd., 2009, págs. 612-620.

[21] P. Caceres, *The Perceptron - a Guided Tutorial through Its History and Implementation in Python*, 2020.

[22] C. Sryker y D. Bergmann, *¿Qué es la función de Pérdida?* https://www.ibm.com/mx-es/think/topics/loss-function, jul. de 2024. visitado 2 de dic. de 2024.

[23] K. Pykes, *Cross-Entropy Loss Function in Machine Learning: Enhancing Model Accuracy*, https://www.datacamp.com/tutorial/the-cross-entropy-loss-function-in-machine-learning, ago. de 2024. visitado 20 de dic. de 2024.

[24] J. Skycak, *Single-Variable Gradient Descent*, https://justinmath.com/single-variable-gradient-descent/, feb. de 2021. visitado 23 de ene. de 2025.

[25] E. F. Caicedo Bravo y J. A. López Sotelo, *Una aproximación práctica a las redes neuronales artificiales*. Calí, Colombia: Programa Editorial Universidad del Valle, 2020.

[26] GeeksforGeeks, *Introduction to Recurrent Neural Networks*, https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/, nov. de 2024. visitado 12 de feb. de 2025.

[27] C. Olah, *Understanding LSTM Networks*, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, ago. de 2015. visitado 19 de mayo de 2025.

[28] J. Russell, "Anti-Friction Bearing Fundamentals," *Continuing Eduacation and Development Inc.*, n.º M01-201, 2020.

[29] NBC, *Basics of Roller Bearings*, Jaipur, India, 2021.

[30] Aptitude Xchange, "Bearing Basic: An Overviiew," *SKF Reliability Systems*, n.º RB02002, ene. de 2004.

[31] P. R. N. Childs, "6 - Rolling Element Bearings," en *Mechanical Design Engineering Handbook (Second Edition)*, P. R. N. Childs, ed., Butterworth-Heinemann, nov. de 2018, págs. 231-294. visitado 8 de mar. de 2025.

[32] NSK, "Rolling Bearings," n.º E1102m, A10-A12, 2013.

[33] D. Rodríguez y K. E. Meyers, *Análisis de daños en los rodamientos: ISO 15243 está aquí para ayudarle*, ago. de 2022. visitado 19 de mar. de 2025.

[34] A. Grande Gutiérrez, "Desarrollo de Un Sistema de Diagnóstico de Fallo de Rodamientos En Maquinaria Agrícola Mediante Análisis de Señales Acústicas.," Tesis doct., Universidad Politécnica de Madrid, Madrid, jul. de 2022.

[35] S. M. Namara Retamal, "Análisis de desbalanceo dinámico mediante el software Skf Aptitude Analyst en conjunto con un estudio de vibraciones en Rotor Machinery Fault Simulator," Tesis doct., Universidad de Talca, Chile, 2022.

[36] G. White, *Introducción al Análisis de Vibraciones*, Azima DLI. North Woburn, USA, 2010.

[37] T. Flatscher, *Schematic Fourier Analysis*, oct. de 2019. visitado 22 de mar. de 2025.

[38] ADASH, *Vibration Analysis : Vibration Limits, Types of Measurements, Acceleration Sensor*. Mar. de 2019. visitado 31 de mar. de 2025.

[39] Wilcoxon Sensing Technologies, *Accelerometer mounting methods blog*, 2012. visitado 31 de mar. de 2025.

[40] E. Florez, S. Cardona y L. Jordi, "Selección de La Ventana Temporal En La Transformada de Fourier En Tiempos Cortos Utilizada En El Análisis de Señales de Vibración Para Determinar Planos En Las Ruedas de Un Tren," *Revista Facultad de Ingeniería*, dic. de 2009.

[41] I. Neutelings, *Harmonic oscillator plots*, jul. de 2021. visitado 13 de abr. de 2025.

[42] M. d. R. Prieto García, "Vibraciones de Máquinas Rotativas: Análisis de Órdenes," Tesis doct., Universidad de Sevilla, Sevilla, España, 2017.

[43] T. Bäckström et al., *Introduction to Speech Processing*, 2nd Edition. 2022. visitado 24 de abr. de 2025.

[44] Mathuranathan Viswanathan, *Choosing FIR or IIR? Understand design perspective*, feb. de 2017. visitado 25 de abr. de 2025.

[45] Case Western Reserve University, *Apparatus & Procedures | Case School of Engineering*,
https://engineering.case.edu/bearingdatacenter/apparatus-and-procedures.
visitado 5 de mayo de 2025.

[46] M. Preetham, *Multiclass Metrics of a Confusion Matrix*, nov. de 2021. visitado 21 de mayo de 2025.

Anexos

A.1. Derivada de la función de activación sigmoidal

Se reescribe la ecuación 2.7 a fin de obtener su derivada, tal como se muestra a continuación:

$$\begin{aligned} f(x) &= \frac{1}{1+e^{-x}} \\ f(x) &= (1)(1+e^{-x})^{-1} \\ f(x) &= (1+e^{-x})^{-1} \end{aligned} \tag{1}$$

Posteriormente, con ayuda de la regla de la cadena, se deriva de la siguiente forma:

$$\begin{aligned} \frac{d}{dx}f(x) &= \frac{d}{dx}((1+e^{-x})^{-1}) \\ \frac{d}{dx}f(x) &= -1\left[(1+e^{-x})^{(-1-1)}\cdot\frac{d}{dx}(1+e^{-x})\right] \\ \frac{d}{dx}f(x) &= -1\left[(1+e^{-x})^{-2}\cdot\frac{d}{dx}(1)+\frac{d}{dx}(e^{-x})\right] \\ \frac{d}{dx}f(x) &= -1\left[(1+e^{-x})^{-2}\cdot(0+e^{-x}\frac{d}{dx}(-x))\right] \\ \frac{d}{dx}f(x) &= -1\left[(1+e^{-x})^{-2}\cdot e^{-x}\cdot(-1)\right] \\ \frac{d}{dx}f(x) &= (1+e^{-x})^{-2}\cdot(e^{-x}) \\ \frac{d}{dx}f(x) &= \frac{e^{-x}}{(1+e^{-x})^2} \end{aligned} \tag{2}$$

La principal característica de esta función es que su derivada puede ser expresada en término de si misma:

$$\begin{aligned} f'(x) &= \frac{e^{-x}}{(1+e^{-x})^2} \\ f'(x) &= \frac{e^{-x}+1-1}{(1+e^{-x})^2} \\ f'(x) &= \frac{(1+e^{-x})}{(1+e^{-x})^2}-\frac{1}{(1+e^{-x})^2} \\ f'(x) &= \frac{1}{(1+e^{-x})}-\frac{1}{(1+e^{-x})^2} \\ f'(x) &= \frac{1}{(1+e^{-x})}-\left[\frac{1}{(1+e^{-x})}\cdot\frac{1}{(1+e^{-x})}\right] \\ f'(x) &= \frac{1}{(1+e^{-x})}\left(1-\frac{1}{(1+e^{-x})}\right) \\ f'(x) &= f(x)\,(1-f(x)) \end{aligned} \tag{3}$$

$$f'(net_{pk}) = o_{pk}\,(1-o_{pk})$$

B.2. Derivadas del descenso del gradiente

B.2.1. Derivada del error con respecto a la función de activación

$$\frac{\partial E_p}{\partial o_{pk}} = \frac{\partial \left[\frac{1}{2}\left(y_{pk} - o_{pk}\right)^2\right]}{\partial o_{pk}} = -(y_{pk} - o_{pk}) \tag{4}$$

B.2.2. Derivada de la función de activación sigmoidal con respecto a la suma ponderada

$$\begin{aligned}
\frac{\partial o_{pk}}{\partial net_{pk}} &= \frac{\partial (1 + e^{-net_{pk}})^{-1}}{\partial net_{pk}} \\
\frac{\partial o_{pk}}{\partial net_{pk}} &= \frac{e^{-net_{pk}}}{(1 + e^{-net_{pk}})^2} + 1 - 1 \\
\frac{\partial o_{pk}}{\partial net_{pk}} &= \frac{(1 + e^{-net_{pk}})}{(1 + e^{-net_{pk}})^2} - \frac{1}{(1 + e^{-net_{pk}})^2} \\
\frac{\partial o_{pk}}{\partial net_{pk}} &= \frac{1}{(1 + e^{-net_{pk}})} - \frac{1}{(1 + e^{-net_{pk}})^2} \\
\frac{\partial o_{pk}}{\partial net_{pk}} &= \frac{1}{(1 + e^{-net_{pk}})} - \left[\frac{1}{(1 + e^{-net_{pk}})} \cdot \frac{1}{(1 + e^{-net_{pk}})}\right] \\
\frac{\partial o_{pk}}{\partial net_{pk}} &= \frac{1}{(1 + e^{-net_{pk}})}\left(1 - \frac{1}{(1 + e^{-net_{pk}})}\right) \\
\frac{\partial o_{pk}}{\partial net_{pk}} &= f(net_{pk})\left(1 - f(net_{pk})\right) \\
\frac{\partial o_{pk}}{\partial net_{pk}} &= o_{pk}(1 - o_{pk})
\end{aligned} \tag{5}$$

B.2.3. Derivada de la suma ponderada con respecto a los pesos ocultos

$$\frac{\partial net_{pk}}{\partial w_{kj}} = \frac{\partial \left(w_{kj} \cdot i_j\right)}{\partial w_{kj}} = i_j \tag{6}$$

B.2.4. Actualización de los pesos en la capa oculta

Sustituyendo los resultados anteriores, obtenemos la diferencia de error en los pesos que conforman la capa oculta

$$\Delta w_{kj} = \alpha \left(y_{pk} - o_{pk}\right) \cdot o_{pk}(1 - o_{pk}) \cdot i_{pj} \tag{7}$$

B.2.5. Actualización de los pesos en la capa de entrada

$$\Delta w_{ji} = -\left[\sum_{k=1} \frac{\partial E_p}{\partial o_{pk}} \cdot \frac{\partial o_{pk}}{net_{pk}} \cdot \frac{\partial net_{pk}}{\partial i_{pj}}\right] \cdot \frac{\partial i_{pj}}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial w_{ji}}$$

$$\Delta w_{ji} = \alpha\left[-(y_{pk} - o_{pk}) \cdot o_{pk}(1 - o_{pk}) \cdot w_{kj}\right] \cdot f_j(1 - f_j) \cdot x_{pi}$$