



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



INSTITUTO TECNOLÓGICO DE LEÓN

TESIS

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA BCI CON
CLASIFICACIÓN DE SEÑALES EMPLEANDO MACHINE LEARNING

Que presenta:

LUIS GERARDO VILCHIS SOTO

MAESTRÍA EN CIENCIAS
DE LA COMPUTACIÓN

Con la dirección de:

DRA. MARÍA DEL ROSARIO BALTAZAR FLORES

Con la co-dirección de:

DR. ARNULFO ALANIS GARZA

León, Guanajuato, México

Junio 2024

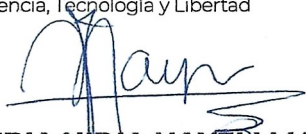
León, Guanajuato, **19/junio/2024**
Memorándum No. DEPI-070-2024

**C. VILCHIS SOTO LUIS GERARDO
PRESENTE**

De acuerdo al fallo emitido por la Comisión Revisora, integrada por: Dra. María del Rosario Baltazar Flores, Dr. Arnulfo Alanís Garza, Dr. Juan Francisco Mosiño y Dr. Raúl Santiago Montero; y considerando que cubre todos los requisitos establecidos en los Lineamientos Generales para la Operación del Posgrado del Tecnológico Nacional de México, se autoriza la impresión del trabajo de tesis titulado: "Desarrollo e implementación de un sistema BCI con clasificación de señales empleando machine learning". Lo que hacemos de su conocimiento para los efectos y fines correspondientes.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Ciencia, Tecnología y Libertad



**LIBIA NIDIA NAYELI MARTÍNEZ AGUILAR
JEFA DE DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN.**

LNNMA/barc.



Av. Tecnológico s/n Fracc. Industrial Julián de Obregón C.P.37290 León, Guanajuato.
Tel.: 477 710 5200 e-mail: tecleon@leon.tecnm.mx tecnm.mx | leon.tecnm.mx





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

León, Guanajuato, a 11 de junio del 2024

C. ING. LUIS ROBERTO GALLEGOS MUÑOZ
JEFE DE SERVICIOS ESCOLARES
P R E S E N T E

Por este medio hacemos de su conocimiento que la tesis titulada **"Desarrollo e implementación de un sistema BCI con clasificación de señales empleando machine learning"**, ha sido leída y aprobada por los miembros del Comité Tutorial para su evaluación por el jurado del acto de examen de grado al alumno (a) **C. Luis Gerardo Vilchis Soto**, con número de control **M22240200** como parte de los requisitos para obtener el grado de Maestro(a) en Ciencias de la Computación (MCCOM-2011-05).

Sin otro particular por el momento, quedamos de Usted.

ATENTAMENTE
COMITÉ TUTORIAL

Dra. María del Rosario Baltazar Flores

DIRECTOR

Dr. Arnulfo Alanís Garza

CODIRECTOR O REVISOR

Dr. Juan Francisco Mosiño

REVISOR (a)

Dr. Raúl Santiago Montero

REVISOR (a)



DECLARACION DE AUTENTICIDAD Y DE NO PLAGIO

Yo, **Luis Gerardo Vilchis Soto**, identificado con No. Control **M22240200**, alumno (a) del programa de la **Maestría en Ciencias de la Computación o Ingeniería**, autor (a) de la Tesis titulada: **“Desarrollo e implementación de un sistema BCI con clasificación de señales empleando machine learning”**, DECLARO QUE:

1.- El presente trabajo de investigación, tema de la tesis presentada para la obtención del título de **MAESTRO (A) EN CIENCIAS DE LA COMPUTACIÓN O INGENIERÍA** es original, siendo resultado de mi trabajo personal, el cual no he copiado de otro trabajo de investigación, ni utilizado ideas, fórmulas, ni citas completas “stricto sensu”, así como ilustraciones, fotografías u otros materiales audiovisuales, obtenidas de cualquier tesis, obra, artículo, memoria, etc. en su versión digital o impresa.

2.- Declaro que el trabajo de investigación que pongo a consideración para evaluación no ha sido presentado anteriormente para obtener algún grado académico o título, ni ha sido publicado en sitio alguno.

3.- Declaro que las pruebas o experimentos derivados de esta investigación fueron realizados bajo el consentimiento de los involucrados y con fines estrictamente académicos conforme a criterios éticos de confidencialidad.

Soy consciente de que el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades de la tesis, así como de los derechos sobre la obra presentada.

Asimismo, me hago responsable ante el Tecnológico Nacional de México/Instituto Tecnológico de León o terceros, de cualquier irregularidad o daño que pudiera ocasionar por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el trabajo de investigación haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas pecuniarias o legales que se deriven de ello sometiéndome a las normas establecidas en los Lineamientos y Disposiciones de la Operación de Estudios de Posgrado en el Tecnológico Nacional de México.

León, Guanajuato a 14 de junio de 2024



Luis Gerardo Vilchis Soto

ACUERDO PARA USO DE OBRA (TESIS DE GRADO)

A QUIEN CORRESPONDA

PRESENTE

Por medio del presente escrito, **Luis Gerardo Vilchis Soto** (en lo sucesivo el AUTOR) hace constar que es titular intelectual de la obra denominada: **“Desarrollo e implementación de un sistema BCI con clasificación de señales empleando machine learning”**, (en lo sucesivo la OBRA) en virtud de lo cual autoriza al Tecnológico Nacional de México/Instituto Tecnológico de León (en lo sucesivo TECN/IT León) para que efectúe resguardo físico y/o electrónico mediante copia digital o impresa para asegurar su disponibilidad, divulgación, comunicación pública, distribución, transmisión, reproducción, así como digitalización de la misma con fines académicos y sin fines de lucro como parte del Repositorio Institucional del TECN/ITLeón.

De igual manera, es deseo del AUTOR establecer que esta autorización es voluntaria y gratuita, y que de acuerdo a lo señalado en la Ley Federal del Derecho de Autor y la Ley de Propiedad Industrial el TECN/IT León cuenta con mi autorización para la utilización de la información antes señalada, estableciendo que se utilizará única y exclusivamente para los fines antes señalados. El AUTOR autoriza al TECN/IT León a utilizar la obra en los términos y condiciones aquí expresados, sin que ello implique se le conceda licencia o autorización alguna o algún tipo de derecho distinto al mencionada respecto a la “propiedad intelectual” de la misma OBRA; incluyendo todo tipo de derechos patrimoniales sobre obras y creaciones protegidas por derechos de autor y demás formas de propiedad intelectual reconocida o que lleguen a reconocer las leyes correspondientes. Al reutilizar, reproducir, transmitir y/o distribuir la OBRA se deberá reconocer y dar créditos de autoría de la obra intelectual en los términos especificados por el propio autor, y el no hacerlo implica el término de uso de esta licencia para los fines estipulados. Nada de esta licencia menoscaba o restringe los derechos patrimoniales y morales del AUTOR.

De la misma manera, se hace manifiesto que el contenido académico, literario, la edición y en general de cualquier parte de la OBRA son responsabilidad de AUTOR, por lo que se deslinda al (TECN/ITLeón) por cualquier violación a los derechos de autor y/o propiedad intelectual, así como cualquier responsabilidad relacionada con la misma frente a terceros. Finalmente, el AUTOR manifiesta que estará depositando la versión final de su documento de Tesis, OBRA, y cuenta con los derechos morales y patrimoniales correspondientes para otorgar la presente autorización de uso.

En la ciudad de León, del estado de Guanajuato a los 14 días del mes de junio de 2024.

Atentamente,

Luis Gerardo Vilchis Soto



Índice general

Índice general	VI
1. Introducción	4
1.1. Planteamiento del problema	5
1.1.1. Preguntas de investigación	6
1.2. Justificación	7
1.3. Objetivos	8
1.3.1. Objetivo general	8
1.3.2. Objetivos específicos	8
1.4. Hipótesis	9
1.5. Metodología	9
1.5.1. Alcance y limitaciones	12
2. Estado del Arte	14
3. Marco teórico	17
3.1. Trastorno del Espectro Autista (TEA)	17
3.2. Brain-computer interface (BCI)	19
3.3. Electroencefalograma (EEG)	22
3.4. Procesamiento Digital de Señales (DSP)	24
3.4.1. Teorema de Nyquist-Shannon	25
3.4.2. Transformada de Fourier	26

3.4.2.1.	Transformada Discreta de Fourier (DFT)	28
3.4.2.2.	Transformada Rápida de Fourier (FFT)	28
3.4.3.	Transformada Wavelet	29
3.4.3.1.	Transformada Wavelet Continua (CWT)	31
3.4.3.2.	Transformada Wavelet Discreta (DWT)	33
3.4.4.	Clasificadores (Machine Learning)	34
3.4.4.1.	Clasificador ANN (Artificial Neural Network)	35
3.4.5.	Herramientas de validación estadística	42
3.4.5.1.	Análisis de componentes principales (PCA)	42
3.4.5.2.	Validación cruzada k-fold	43
3.4.6.	Evalúadores de modelo	44
3.4.6.1.	Matriz de confusión	44
3.4.6.2.	Métricas de clasificación	45
3.5.	Placa de Circuito Impreso (PCB)	47
4.	Desarrollo	50
4.1.	Sistema EEG-BCI basado en ESP32 (propuesta de sistema embebido)	50
4.1.1.	Formulación del problema	50
4.1.2.	Especificaciones del sistema	50
4.1.3.	Arquitectura y desarrollo de hardware	51
4.1.4.	Implementación del Firmware	59
4.2.	Diseño de la red neuronal	62
4.3.	Implementación de la red neuronal en MCU	71
4.3.1.	Edge Impulse	71
4.4.	Colocación de electrodos EEG	73
4.4.1.	Funciones de las áreas del cerebro	73
4.4.2.	Colocación de electrodos EEG	74

5. Experimentación y Resultados	77
5.1. Medición y análisis de señales a través del BCI diseñado	77
5.1.1. Diseño experimental	77
5.1.2. Resultados del análisis de las señales EEG	83
5.2. Análisis y clasificación de dataset con etiquetas y pruebas en sistema embebido	87
5.2.1. Exploratory Data Analysis (EDA)	87
5.2.1.1. Análisis estadístico descriptivo	87
5.2.2. Clasificadores en PC	95
5.2.2.1. K-Nearest Neighbors (KNN)	95
5.2.2.2. Random Forests	96
5.2.2.3. Support Vector Machine (SVM)	98
5.2.2.4. Naive Bayes	99
5.2.2.5. Logistic Regression	101
5.2.2.6. Multilayer Perceptron (MLP)	102
5.2.2.7. LSTM (Long Short Term Memory)	104
5.2.3. Clasificadores en sistema embebido	105
5.2.3.1. K-Nearest Neighbors (KNN)	105
5.2.3.2. Random Forests	107
5.2.3.3. Support Vector Machine (SVM)	108
5.2.3.4. Naive Bayes	110
5.2.3.5. Logistic Regression	111
5.2.3.6. Multilayer Perceptron (MLP)	113
5.2.3.7. LSTM (Long Short Term Memory)	114
5.2.4. Comparativa de clasificadores en PC	116
5.2.4.1. Exactitud	116
6. Conclusiones	117
6.1. Limitaciones de la investigación	118

6.2. Trabajo a futuro	119
---------------------------------	-----

Dedicatoria

A mis queridos padres, cuyo amor incondicional y apoyo inquebrantable han sido la columna vertebral de mi vida y mi educación. A mi mentora, Rosario Baltazar, cuya sabiduría y enseñanzas han sido cruciales no solo en la formación de esta investigación, sino también en mi desarrollo personal y profesional. A mis amigos, fieles compañeros en cada etapa, cada desafío, y cada triunfo, que han enriquecido este viaje con su presencia y amistad. Esta tesis es un tributo a todos aquellos que han dejado una huella imborrable en mi camino: es un reflejo de su amor, sacrificio y fe en mis capacidades.

Agradecimientos

Mi más profundo agradecimiento al Tecnológico Nacional de México/Instituto Tecnológico de León, cuya excelencia académica me brindó la oportunidad de conocer y ser guiado por mi asesora, Dra. Rosario Baltazar. Su apoyo incansable y motivación han sido fundamentales en mi camino académico. Extiendo mi gratitud a mi comité tutorial: al Dr. Arnulfo Alanís, al Dr. Francisco Mosiño y al Dr. Raúl Santiago, por su fe en mis capacidades y sus valiosas recomendaciones y consejos. Un agradecimiento especial al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el apoyo económico a través de su programa de becas, facilitando así mi formación y desarrollo profesional.

Resumen

Este trabajo de tesis presenta un enfoque técnico para el diseño de una plataforma BCI rentable desde la construcción de hardware basado en componentes comerciales hasta el firmware. Este dispositivo puede incluir controladores para la adquisición de señales, preprocesamiento, eliminación de artefactos, separación de bandas, extracción de características y clasificación mediante ANN comprimida con el marco de trabajo TensorFlow Lite. Consiste en una plataforma EEG basada en STM32 orientada como una herramienta para experimentos BCI en el ámbito educativo y de investigación, capaz de enviar datos a PC o dispositivos móviles a través de una conexión inalámbrica, permitiendo capacidades de procesamiento externo junto con la principal característica de clasificación de señales integradas (TinyML) enfocada en la detección temprana mediante la discriminación de niños con TEA de los que no lo tienen.

Capítulo 1

Introducción

De acuerdo con la Organización Mundial de la Salud (OMS) (World Health Organization (WHO), 2022), el autismo, también llamado Trastorno del Espectro Autista (TEA), constituye un conjunto de condiciones diversas relacionadas con el desarrollo cerebral. Las personas con autismo a menudo tienen condiciones comórbidas, como epilepsia, depresión, ansiedad, trastorno por déficit de atención e hiperactividad y comportamientos problemáticos, como problemas para dormir y autolesiones. El nivel intelectual varía enormemente de un caso a otro, abarcando desde el deterioro severo hasta casos con altas capacidades cognitivas.

Las capacidades y necesidades de las personas con autismo varían y pueden evolucionar con el tiempo. Aunque algunas personas con autismo pueden vivir de forma independiente, hay otras con discapacidades graves que necesitan cuidados y apoyo constantes durante toda su vida. El autismo influye a menudo en las oportunidades educativas y laborales.

Los resultados de la investigación muestran que ha habido un aumento en la prevalencia registrada del autismo en todo el mundo. Esto puede atribuirse a varios factores, como el creciente conocimiento de la condición en las comunidades y la respuesta global de iniciativas de salud pública. Revisamos estudios sobre la prevalencia del autismo en todo el mundo, considerando el impacto de factores geográficos, étnicos y socioeconómicos en las estimaciones de prevalencia. Aproximadamente 1 de cada 100 niños es diagnosticado con trastorno del espectro autista en todo el mundo (Zeidan et al., 2022); esta estimación representa una cifra

promedio ya que la prevalencia observada varía considerablemente entre los diferentes estudios. Sin embargo, algunos estudios bien controlados han reportado cifras significativamente más altas. Algunas personas con autismo pueden vivir de forma independiente, mientras que otras están gravemente discapacitadas y necesitan cuidado y apoyo constantes durante toda su vida.

En los últimos diez años, una cantidad creciente de evidencia ha resaltado la importancia de la detección temprana para el tratamiento exitoso del autismo. Esto significa que la comunidad médica debería priorizar el desarrollo de un método diagnóstico asequible y preciso que pueda identificar la condición lo más pronto posible.

Tecnologías como los EEG (electroencefalogramas) y la inteligencia artificial son herramientas clave para lograr mejorar la calidad de vida de las personas con estas afecciones, demostrándose a lo largo de los años con el análisis de las señales de la actividad cerebral y el uso de las neurociencias para brindarle neuroretroalimentación al paciente y rehabilitarle poco a poco su cognición. Y de la misma manera, los sistemas embebidos, como el más reciente paradigma tecnológico de la humanidad, juega un papel relevante en el proceso de poder llevar todas las tecnologías mencionadas al ámbito del cómputo portátil personal con un costo asequible; como un nodo más en el ecosistema de los ambientes inteligentes.

1.1. Planteamiento del problema

Si bien actualmente existen un sinnúmero de dispositivos y técnicas (académicas y de aplicación en campo) para la exploración de las señales de EEG en la rama de las neurociencias, aún no se cuenta con un dispositivo portátil, de bajo costo y enfocado al diagnóstico de pacientes en el espectro neurodivergente.

El reto planteado yace en la elaboración de un modelo de procesamiento de señales y una ANN capaz de caracterizar los EEG en el espectro neurodivergente, todo circunscrito a una población por edades específicas (puesto que el desarrollo y estado neurológico dista mucho entre grupos de edad) y sexo específico dentro del rango del espectro que se desee

clasificar (que puede ir desde el autismo hasta trastornos de ansiedad y paranoicos). A su vez, lo anterior debe ser optimizado correctamente (asumiendo una pérdida en la resolución, eficiencia y precisión sin mermar la exactitud y validez), descargado y ejecutado en una plataforma (sistema embebido) con las capacidades y recursos necesarios para computar las señales y clasificarlas (a partir del modelo principal comprimido) y otorgar un veredicto sobre el paciente, con base de un esquema que apele a la movilidad, portabilidad y *free roaming*. Es importante recalcar la relación que existe entre el hardware (equipo de medición) y la calidad de señales recabadas para un diagnóstico certero del paciente (ya sea por una red neuronal o un agente humano experto en el área); por lo que es imprescindible indagar en las diferencias notables que puedes existir entre un dispositivo de laboratorio profesional (de banco) contra uno académico (portátil) que posean calidades y capacidades de cómputo muy similares y las variables que puedan influir en detrimento en la calidad de la información a partir de la presuposición de uno móvil mientras que el otro no.

1.1.1. Preguntas de investigación

Teniendo en cuenta el problema y el uso de la tecnología dirigida en pro de la calidad de vida, surgen algunas preguntas relacionadas directamente con las funcionalidades y características de las herramientas más novedosas y la clasificación e interpretación de las señales EEG:

- ¿Cuál es la relación que se da entre el número de canales y el grado de portabilidad de la herramienta de medición para garantizar calidad e integridad de las señales?
- ¿Cuáles son las ventajas y limitaciones de las diferentes técnicas de procesamiento (FFT, DWT, HHT, etc.) y extracción de características de señales EEG (DNN y SVM) y cuánta precisión tienen en la obtención de información relevante de actividad cerebral de personas neurodivergentes?
- ¿Cómo han venido dándose los patrones de interés en la actividad cerebral para la caracterización y cómo se construye un modelo de actividad cerebral específico para los pacientes neurodivergentes?

- ¿Cómo pueden afectar al estudio una alta dispersión en el sexo y edad de los sujetos de pruebas para el análisis de señales del cerebro en el espectro autista?

1.2. Justificación

Los trastornos neurológicos y neurocognitivos han incrementado considerablemente en las últimas décadas en todo el mundo y, a su vez, el paradigma de esperanza de vida se ha estado reemplazando poco a poco con el de calidad de vida, pues, de acuerdo a la perspectiva generacional, es mejor vivir poco tiempo, pero con un cuerpo funcional que vivir mucho tiempo con un cuerpo doliente (Chaste & Leboyer, 2012). Con ello, la accesibilidad a mejores y más novedosas tecnologías para el cuidado de la salud están en la mira de los países más desarrollados.

La finalidad de este estudio es obtener información clara sobre los mejores métodos de extracción y caracterización a partir de señales de EEG orientadas a la atención médica para el diagnóstico de personas neurodivergentes y un equipo portátil con el poder de cómputo suficiente y algoritmos *ad hoc* para fungir como una herramienta embebida capaz de soportar las funciones necesarias para clasificar pacientes.

Asimismo, es importante señalar que no hay estudios previos en esta materia con el diseño de cómputo embebido portátil para atención de pacientes de esta naturaleza. La mayoría de los equipos portátiles están orientados a la accesibilidad a través de BCI, o bien, como equipo de laboratorio para estudios en el campo de la neurociencia para facilitar el *free roaming* en las pruebas pertinentes. Y con la propuesta, posicionada en ese nivel de accesibilidad, es factible alcanzar un mayor número de clínicas para que éstas se hagan de su propio aparato.

A partir de la premisa argumentativa, se pretende cambiar la forma en la que los profesionales de la salud brindan diagnósticos, facilitando la acotación del espectro neurodivergente, con el propósito de representar un apoyo más objetivo en el veredicto del padecimiento; favoreciendo directamente al paciente y allegados, puesto que se podría proveer de una atención clínica más precisa que aporte mucho más en su rehabilitación.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar e implementar un sistema BCI embebido, portátil y de bajo costo como herramienta de apoyo para el diagnóstico de personas neurodivergentes con la ayuda de un modelo de análisis para caracterización de señales de actividad cerebral.

1.3.2. Objetivos específicos

- Clasificar las señales a partir de machine learning utilizando SVM con aprendizaje supervisado para la detección y neuroretroalimentación de actividad cerebral atípica.
- Procesar digitalmente las señales, valiéndose de la FFT y la DWT a través del uso de redes neuronales optimizadas para procesadores ARM Cortex-M.
- Modelar y caracterizar con EEG la actividad cerebral de pacientes neurodivergentes con una red neuronal de bajo costo de cómputo para su encauzamiento a procesamiento portátil.
- Diseñar una tarjeta electrónica de adquisición de señales EEG (BCI) de 16 canales con alimentación por batería recargable y comunicación inalámbrica a 5 GHz con protocolo BLE (*Bluetooth Low Energy*).
- Programar el sistema embebido enfocado a la integración de firmware (adquisición y procesamiento de señales) y software (*machine learning*).
- Reportar el impacto en neuroretroalimentación de los desórdenes neurológicos del sujeto de pruebas neurodivergente.
- Reportar, de manera general, los resultados del dispositivo en cuanto a eficiencia eléctrica, poder de cómputo, portabilidad, precisión y velocidad de adquisición, velocidad de clasificación y procesamiento de señales.

1.4. Hipótesis

Es posible generar un sistema que permita clasificar personas neurodivergentes empleando redes neuronales embebidas.

1.5. Metodología

El proceso de diseño y ejecución experimentales serán llevados a cabo en el Laboratorio de Ambientes Inteligentes perteneciente al DEPI, bajo el asesoramiento de la Dra. María del Rosario Baltazar Flores, en el Instituto Tecnológico de León dirigido por el siguiente esquema de desarrollo:

1. Revisión de bibliografías e investigación:

- a)* Consultar fuentes bibliográficas válidas en el ámbito de la ingeniería, ciencias computacionales y medicina que se ajusten al enfoque del proyecto.
- b)* Reportar en una hoja de cálculo el estado del arte con sus respectivas bibliografías citadas en APA.
- c)* Discutir la validez y relevancia de las fuentes para la investigación.
- d)* Reportar toda la información recabada al coasesor y brindar un espacio de asesoría y retroalimentación de los avances.
- e)* Visitar centros de neurociencia (De La Salle como tentativo) para consultar información objetiva de los protocolos éticos y procedurales para las lecturas de EEG.
- f)* Visitar centros de psiquiatría y psicología para consultar información objetiva de los protocolos éticos y procedurales para el diagnóstico de personas neurodivergentes.

- g) A partir del punto anterior: Identificar los aspectos neurocomportamentales relevantes que son indispensables para el diagnóstico clínico del espectro neurodivergente.
2. Diseño de hardware embebido para experimentación (equipo de laboratorio):
- a) Consultar fuentes bibliográficas válidas de proveedores de semiconductores y equipos electrónicos a fin de actualizar el estado del arte en el desarrollo de SoC (*System on a Chip*) y *turnkey solutions* relevantes para la construcción del equipo.
 - b) Diseñar en tarjetas de prototipado los circuitos electrónicos más delicados (hablando de integridad de señales) con tal de poder hacer pruebas y mediciones eléctricas importantes para obtener un panorama de validez de las señales.
 - c) Programar firmware de pruebas para el procesamiento digital de señales en el ambiente de sistemas embebidos. Esto a través del uso de tarjetas de desarrollo de evaluación provistas por los fabricantes de microprocesadores.
 - d) Programar firmware de pruebas para redes neuronales comprimidas en el ambiente de sistemas embebidos. Esto a través del uso de tarjetas de desarrollo de evaluación provistas por los fabricantes de microprocesadores.
 - e) Diseñar y ensamblar la tarjeta electrónica bajo los estándares del IPC (*Institute of Printed Circuits*) para cumplir las normas internacionales de integridad de señales analógica y digitales de alta velocidad.
 - f) Crear un banco de pruebas para la tarjeta con la finalidad de probar todas las funcionalidades y verificar el correcto funcionamiento de los circuitos y *framework* del firmware.
3. Diseño de software de *machine learning* para el procesamiento de las señales (base experimental):
- a) El algoritmo base para lograr la clasificación de las señales es el siguiente: captura de señales, preprocesamiento digital de señales (tratamiento con filtros), procesa-

- miento digital de señales (aplicación de transformadas para extraer características relevantes), inserción de los datos relevantes en el clasificador, interpretación automatizada del vector de características resultante de la DNN.
- b)* Seleccionar bases de datos congruentes con el sujeto de estudio y descargarlas en el servidor de inteligencia artificial.
 - c)* Programar las funciones indispensables para el tratamiento de señales digitales (transformadas y filtros).
 - d)* Definir los vectores característicos y programar su ejecución en Keras y TensorFlow.
 - e)* Emplear las técnicas (algoritmos) de clasificación de señales pertinentes (SVM) y analizar otras propuestas para su implementación en el sistema según la eficiencia, precisión y exactitud del resultado del procesamiento.
 - f)* Diseñar y programar la red neuronal de clasificación de señales y entrenarla a través del esquema del *deep learning*.
 - g)* Emplear las herramientas del ecosistema de inteligencia artificial de STMicroelectronics (AlgoBuild y XCUBENN) para comprimir, optimizar y cargar la red neuronal entrenada en la memoria del microcontrolador ARM de la tarjeta electrónica desarrollada.

4. Diseño experimental:

- a)* Definir el funcionamiento del dispositivo en una habitación aislada no esterilizada con baja radiación electromagnética de señales inalámbricas (Wi-Fi, Bluetooth, etc.)
- b)* Definir el protocolo de medición del paciente dados los siguientes pasos: Enunciación de políticas de privacidad y ética experimental, encauzamiento de los paciente a la zona de pruebas, definición de ambiente inmobiliario de interacción para el

- paciente, definición de forma de medición de señales y captura (guardado en memoria) de las mismas y captura del expediente de la persona medida.
- c)* Profilaxis obligatoria de la diadema con los electrocodos en el intercambio de sujetos de prueba y previo al inicio del procedimiento.
 - d)* Automatizar la captura de señales para garantizar la integridad de las mismas al restringir el contacto directo con seres humanos u otros objetos.
 - e)* Visualizar en tiempo real en una pantalla de PC el EEG contra el procesamiento digital de señales y el clasificador.
 - f)* Realizar el entrenamiento redundante posterior a la captura de las señales.

5. Validación de resultados:

- a)* Colaborar directamente con los centros de neurociencias y profesionales de la psicología y psiquiatría para darle seguimiento al dispositivo.
- b)* Comparar los resultados del dispositivo contra los diagnósticos clínicos de los profesionales al posicionar a los pacientes dentro del espectro.
- c)* Validar estadísticamente los falsos positivos o negativos que se pueden dar y etiquetar al aparato con un porcentaje de confianza subjetivo (puesto que no se puede suplir éticamente ni funcionalmente a un profesional de la salud).
- d)* Validar todas las posibilidades de clasificación en el espectro neurodivergente a través de pruebas exhaustivas.
- e)* Corroborar la capacidad de análisis del equipo dentro de su rango poblacional y explorar empíricamente su eficacia con otros grupos.
- f)* Abrir la posibilidad de su aplicación y validación a través de su implementación práctica en las clínicas psiquiátricas para apoyar en el diagnóstico.

1.5.1. Alcance y limitaciones

El alcance del proyecto queda definido en los puntos expuestos a continuación:

- El presente estudio explorará la clasificación en el espectro neurodivergente (autista) para los jóvenes de sexo masculino en edades preuniversitarias y universitarias.
- La investigación abarcará únicamente al alumnado del ITL Campus 1 y Campus 2 y la Universidad De La Salle Bajío Campus Campestre.
- El hardware BCI sólo tendrá una capacidad de lectura en tiempo real de 16 canales y su transmisión por BLE a una tasa de 9600 baudios (embebido a PC) sincronizada con una tasa de muestreo a 250 Hz.
- El hardware sólo estará enfocado como herramienta de laboratorio en su diseño industrial , mas no como un producto de consumo electrónico.
- La experiencia de usuario en el uso de los electrodos para la recolección de señales estará acotada a una diadema o casco (no de grado médico).

Las limitaciones del proyecto quedan definidos en los puntos expuestos a continuación:

- Disponibilidad y disposición de los centros de neurociencias y profesionales de la psicología y psiquiatría en tiempos y capacidad de asesoramiento.
- Calidad (en frecuencia de muestreo y resolución) y cantidad de señales EEG diferentes (pacientes) en las bases de datos a consultar.
- Banco de datos incompletos y no actualizados de los centros de investigación en neurociencias.
- El período de tiempo de la caracterización de las señales comprende un año y medio de duración a partir de agosto de 2022.
- Disponibilidad de semiconductores en el rubro de microcontroladores y convertidores analógicos a digitales.
- Calidad y certificados de estándares austeros o nulos en el suministro de insumos.

Capítulo 2

Estado del Arte

Esta parte del texto presentará varios métodos para reconocer tareas cognitivas mediante el análisis de señales EEG usando el DWT para la descomposición de señales. Estos métodos se han desarrollado basándose en trabajos previos de los autores. Los resultados de sus experimentos han confirmado la efectividad de su enfoque propuesto. Esto sugiere que tiene el potencial de ser una herramienta útil para identificar patrones EEG anormales en datos clínicos, ya que puede ayudar a distinguir entre patrones normales y anormales.

Respecto a la posibilidad de usar datos EEG para estudiar el TEA, Bosl et al. (Bosl et al., 2018) presentan un enfoque basado en datos para demostrar que los valores no lineales computados de segmentos relativamente cortos de señales EEG en estado de reposo contienen información que puede usarse como perfiles de biomarcadores para posibilitar una predicción temprana de un futuro desenlace de TEA. Esto propone que la medición EEG usando el método presentado aquí es una tecnología prometedora para monitorear el desarrollo neural en una amplia población de niños.

El principal objetivo de Amin et al. (Amin et al., 2017) es proponer un esquema de reconocimiento de patrones para la extracción de características de modo que puedan comparar los mejores algoritmos de aprendizaje automático para la clasificación de señales EEG bajo diferentes condiciones cognitivas. Los siguientes pasos definen la propuesta de procesamiento para la interpretación de señales EEG postulada por los autores: Extracción de Caracte-

rísticas, Visualización y Estandarización de Características, Selección de Características y Clasificación de Características.

En la clasificación de características, ejecutaron los algoritmos propuestos con resultados alentadores en pequeñas bases de datos públicas de EEG, revelando que la combinación de Transformada Wavelet Discreta (DWT) con la Ratio Discriminante de Fisher (FDA) y técnicas de Análisis de Componentes Principales (PCA) proporcionan un enfoque robusto de extracción de características para la clasificación con Máquinas de Vectores de Soporte (SVM), k-Vecinos más Cercanos (KNN), Perceptrón Multicapa (MLP) ya que esos algoritmos mostraron altas precisiones en bajas frecuencias y por encima.

Por otro lado, Bularka et al. (Bularka & Gontean, 2015) profundizaron no solo en los algoritmos de clasificación sino también en su eficiencia y precisión en aplicaciones en tiempo real para el procesamiento interno y externo de datos EEG. El Método 1 hace posible el desarrollo de un sistema BCI simple y barato con buenos rendimientos de clasificación, ya que el prototipo consta de un headset EEG Emotiv COPD junto con su API interfazada con un programa LabVIEW. La desventaja de este método es su pobre tiempo de respuesta para aplicaciones en tiempo real. El segundo método tiene una mejor precisión de clasificación y es adecuado para implementación en hardware, de esta manera lograron mejor tiempo de respuesta usando un Sistema Operativo en Tiempo Real (RTOS) montado en un Sistema En Un Chip (SOC), explotando la capacidad de procesamiento al máximo concurrente digital del Array de Puertas Programables en Campo (FPGA). Por consiguiente, establece un sólido precedente que hace plausible la integración de un clasificador de señales eficiente en un sistema embebido.

Djmal et al. (Djmal et al., 2017) presentaron un sistema de Diagnóstico Asistido por Computadora (CAD) que diferencia automáticamente entre individuos autistas y no autistas mediante el análisis de señales EEG. Inicialmente, los autores solo usaron transformada wavelet discreta (DWT) y características estadísticas, como media, desviación estándar, varianza, asimetría y curtosis, para extraer las características de la señal EEG. Utilizaron una Red Neuronal Artificial (ANN) para clasificar a los individuos basándose en las características

extraídas. Sin embargo, la precisión de clasificación fue baja utilizando solo DWT y características estadísticas para la extracción de características. Los autores descubrieron que la mayor precisión de clasificación se logró cuando combinaron DWT con la entropía de Shannon para la extracción de características.

La aplicación del método de regularización norma L1 como extractor de características para el cribado temprano de niños con TEA es el foco de la investigación realizada por Peng et al. (Peng et al., 2021), que propone un SVM lineal que utiliza este método para clasificar a niños con autismo usando características EEG seleccionadas que fueron recogidas mientras los niños estaban expuestos a estímulos emocionales positivos y negativos. Los experimentos demostraron que la norma L1 fue efectiva en eliminar características redundantes, identificar mejores y mejorar la precisión del cribado. Así que hay otras metodologías que son más adecuadas para emociones que para procesos cognitivos.

Una de las investigaciones más cercanas al alcance de este proyecto es la de Grossi et al. (Grossi et al., 2020) ya que desarrollaron un clasificador capaz de identificar signos de TEA solo con señales EEG de dos electrodos. El Pick and Squash Tracking (PST), un sistema de aprendizaje automático no supervisado basado en un algoritmo evolutivo, se utilizó para agrupar registros según las características seleccionadas por el sistema Training with Input Selection and Testing (TWIST). El trabajo establece otro importante precedente abriendo el camino a un MLP optimizado capaz de automatizar la clasificación de señales EEG, que puede desarrollarse para ser embebido en un microprocesador.

En resumen, Ibrahim et al. (Ibrahim et al., 2018) desarrollaron un experimento en el que engloban las metodologías y algoritmos previamente expuestos para asistir en el diagnóstico de epilepsia y trastorno del espectro autista. Inferieron que la capacidad para analizar automáticamente datos EEG mejorarían la velocidad y precisión de los procesos de diagnóstico, basándose en la combinación de DWT (extracción de características) y el clasificador KNN obteniendo una alta tasa de precisión.

Capítulo 3

Marco teórico

3.1. Trastorno del Espectro Autista (TEA)

El trastorno del espectro autista (TEA) es un trastorno del desarrollo neurológico caracterizado por dificultades en la comunicación e interacción social, así como por patrones de comportamiento, intereses y actividades restringidos, repetitivos y estereotipados. Aunque los síntomas del autismo pueden variar significativamente entre individuos, estos generalmente se manifiestan antes de los tres años de edad.

El "espectro" en TEA refleja la diversidad y variedad de síntomas y grados de discapacidad que pueden presentar las personas con este trastorno.

1. Dificultades de Comunicación e Interacción Social:

- Problemas para desarrollar y mantener relaciones sociales.
- Dificultad para participar en conversaciones bidireccionales.
- Dificultades para comprender las normas sociales, como turnarse para hablar o mantener el contacto visual.
- Luchar para entender y procesar sus propios sentimientos y los de los demás.

2. Patrones de Comportamiento, Intereses o Actividades Restringidos y Repetitivos:

- Movimientos, uso de objetos o habla repetitiva.
- Necesidad extrema de rutina y resistencia al cambio.
- Intereses restringidos y fijación en ellos.
- Hipersensibilidad o hiposensibilidad a los estímulos sensoriales.

Causas y Factores de Riesgo

Aunque la causa exacta del autismo es desconocida, se cree que es el resultado de una combinación de factores genéticos y ambientales. Los factores de riesgo incluyen tener un hermano con autismo, tener padres mayores, y la exposición a ciertos factores ambientales durante el embarazo.

Diagnóstico y Tratamiento

El diagnóstico se realiza principalmente a través de la observación del comportamiento y desarrollo del niño. Es crucial detectar y empezar a tratar el TEA a una edad temprana para mejorar el pronóstico a largo plazo. Aunque no existe una cura para el TEA, existen tratamientos y terapias, como terapia del comportamiento, terapia ocupacional y medicamentos, que pueden ayudar a manejar los síntomas y mejorar las habilidades sociales y comunicativas. La intervención temprana y el apoyo continuo pueden mejorar significativamente la calidad de vida de las personas con TEA.

Variedad en el Espectro

El TEA es muy variado, y cada persona con autismo tiene un conjunto único de fortalezas y desafíos. Algunas personas pueden necesitar apoyo significativo en su vida diaria, mientras

que otras pueden necesitar menos apoyo y, en algunos casos, vivir de forma independiente. En la Figura 3.1 se muestran algunos de los indicadores.

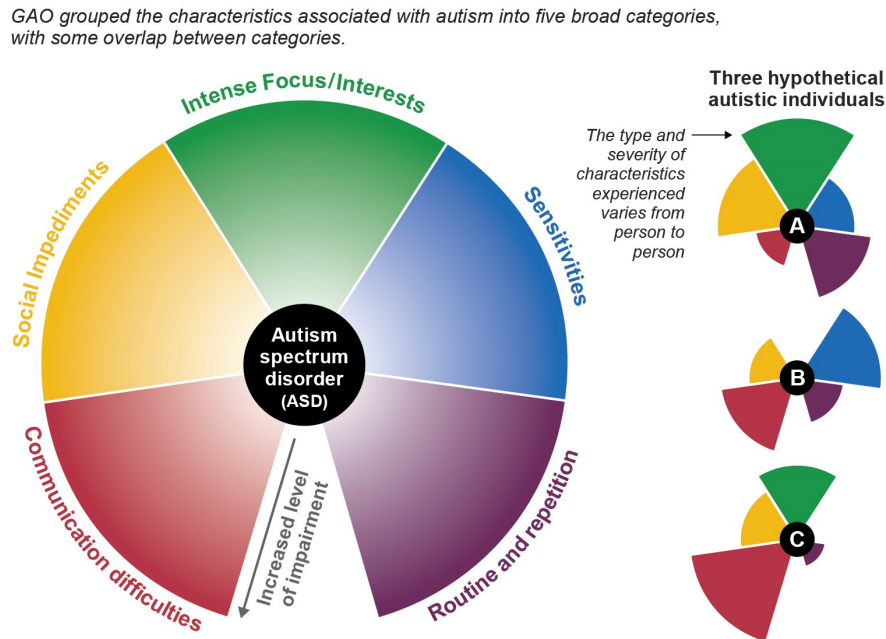


Figura 3.1: Variación en las características del TEA.

(Office, 2023)

3.2. Brain–computer interface (BCI)

Una interfaz cerebro-computadora (BCI, por sus siglas en inglés, "Brain-Computer Interface") es una tecnología avanzada que permite la comunicación directa entre el cerebro y los dispositivos externos, como computadoras o prótesis, sin necesidad de intervención muscular, como se puede observar en la Figura 3.2. Esta tecnología se basa en la conversión de señales cerebrales en comandos digitales que pueden ser interpretados por un dispositivo electrónico.

Causas y Factores de Riesgo

1. Adquisición de Señales:

- Se colocan electrodos en el cuero cabelludo para captar la actividad eléctrica del cerebro (EEG).
- También pueden usarse técnicas más invasivas que implican la implantación de electrodos directamente en el cerebro.

2. Procesamiento de Señales:

- Las señales cerebrales crudas se procesan y se filtran para extraer la información relevante.
- Se traducen las señales en comandos específicos mediante algoritmos de decodificación.

3. Ejecución de Comandos:

- Los comandos decodificados se usan para controlar dispositivos externos, como computadoras, sillas de ruedas, o extremidades robóticas.

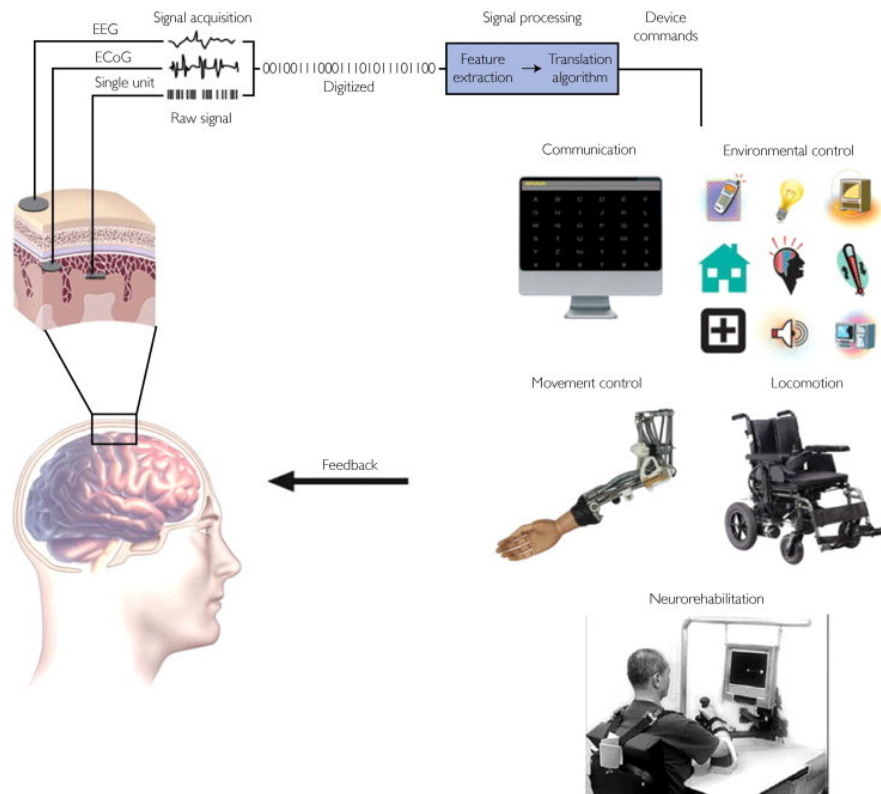


Figura 3.2: Componentes de un sistema BCI.

(Shih et al., 2012)

Aplicaciones de las BCI

1. Medicina:

- Ayudar a las personas con parálisis a mover prótesis robóticas o a comunicarse.
- Rehabilitación de pacientes con daño cerebral o accidentes cerebrovasculares.

2. Investigación:

- Estudio del funcionamiento cerebral y desarrollo de nuevas terapias neurológicas.
- Investigación en neurociencia y comprensión de procesos cognitivos.

3. Tecnología y Entretenimiento:

- Desarrollo de videojuegos controlados por la mente.
- Creación de experiencias inmersivas en realidad virtual.

Retos y Consideraciones Éticas

Las interfaces cerebro-computadora presentan desafíos significativos, tanto técnicos como éticos. El desarrollo de BCIs más precisas y menos invasivas es un reto en curso. Además, la posibilidad de acceder y modificar la actividad cerebral plantea preocupaciones éticas sobre la privacidad, la autonomía, y la identidad de los individuos.

3.3. Electroencefalograma (EEG)

El electroencefalograma (EEG) es una prueba diagnóstica no invasiva que mide y registra la actividad eléctrica del cerebro. Es comúnmente usado para investigar trastornos neurológicos y condiciones como la epilepsia, el trastorno del sueño, y otras alteraciones cerebrales. El EEG es una herramienta crucial en la neurología, la psiquiatría, y la investigación del cerebro.

Procedimiento del EEG

1. Colocación de Electroodos:

- Se colocan electrodos en el cuero cabelludo del paciente.
- Los electrodos son pequeños discos metálicos que detectan la actividad eléctrica del cerebro.
- A menudo se utiliza un gorro especial para asegurar la correcta ubicación de los electrodos.

2. Registro de Actividad Cerebral:

- Los electrodos captan las señales eléctricas generadas por las neuronas del cerebro.

- Las señales son amplificadas y registradas gráficamente, usualmente en una computadora.
- La actividad cerebral se representa mediante líneas onduladas, cada una correspondiente a un electrodo.

3. Análisis de Datos:

- Un neurofisiólogo o neurologista interpreta los patrones y anomalías en el EEG.
- Los hallazgos pueden ayudar a diagnosticar o monitorear diversas condiciones neurológicas.

Aplicaciones del EEG

1. Diagnóstico de Epilepsia:

- Identificación de anomalías eléctricas asociadas a crisis epilépticas.
- Ayuda en la localización del foco epiléptico.

2. Estudio del Sueño:

- Evaluación de los patrones de sueño y diagnóstico de trastornos como el insomnio y la apnea del sueño.
- Análisis de las fases del sueño.

3. Investigación del Cerebro:

- Estudio de la función cerebral y los procesos cognitivos.
- Investigación en neurociencia cognitiva y desarrollo de terapias neurológicas.

4. Interfaces Cerebro-Computadora (BCIs).

Limitaciones del EEG

- Resolución Espacial Limitada: El EEG tiene una resolución espacial (distancia que puede detectar un sensor de imágenes digitales como los satélites o radares) relativamente baja comparada con otras técnicas de imagen cerebral, como la resonancia magnética funcional (fMRI).
- Limitado a la Superficie Cortical: El EEG detecta principalmente la actividad eléctrica de las regiones superficiales del cerebro, por lo que puede no ser efectivo para evaluar estructuras cerebrales más profundas.

3.4. Procesamiento Digital de Señales (DSP)

El Procesamiento Digital de Señales (PDS) es un área fundamental en la ciencia y la ingeniería que se ocupa de la representación, transformación, y manipulación de señales de una manera digital. En el PDS, las señales analógicas, que son representaciones continuas de variables, son convertidas en señales digitales, que son representaciones discretas, mediante un proceso de muestreo y cuantificación, y luego codificadas en formato binario para el análisis y procesamiento subsiguiente.

Este proceso comienza con la conversión de una señal continua en una señal discreta, tomando muestras de la señal a intervalos regulares de tiempo. La frecuencia de estas muestras debe ser adecuada para capturar con precisión la información contenida en la señal original, de acuerdo con el Teorema de Nyquist, para evitar distorsiones.

Una vez que las señales han sido convertidas en una forma digital, se pueden aplicar diversas técnicas y operaciones para analizar y manipular estas señales. La Transformada de Fourier es una técnica crucial en PDS que permite descomponer una señal en sus componentes frecuenciales, facilitando el análisis y filtrado en el dominio de la frecuencia. El filtrado digital, por su parte, es una operación común que permite la atenuación o amplificación de ciertas frecuencias dentro de una señal.

El PDS tiene una amplia gama de aplicaciones en diversos campos como la medicina, para el análisis de señales biológicas y de imagen médica; en las comunicaciones, para la mejora y modulación de señales; en el entretenimiento, para el procesamiento de audio e imágenes; y en la robótica y control de sistemas, para el diseño de sistemas de control. Además, el reconocimiento de patrones y características en señales es fundamental para el desarrollo de tecnologías como el reconocimiento de voz y de imagen.

Las ventajas del PDS sobre el procesamiento analógico de señales incluyen su mayor flexibilidad y precisión, su robustez frente al ruido y otras distorsiones, y la facilidad con la que las señales digitales pueden ser almacenadas, transmitidas y manipuladas utilizando computadoras. La relevancia del Procesamiento Digital de Señales probablemente seguirá creciendo en nuestra era digital contemporánea, impulsando avances en ciencia, tecnología, y medicina.

3.4.1. Teorema de Nyquist-Shannon

El Teorema de Nyquist-Shannon establece que una señal continua en tiempo puede ser completamente reconstruida a partir de sus muestras si la señal ha sido muestreada a una tasa suficientemente alta. Para evitar el aliasing (una distorsión que ocurre cuando se submuestra una señal), la señal debe ser muestreada a una frecuencia de muestreo f_s que sea al menos el doble de la frecuencia máxima, f_{max} , contenida en la señal. Matemáticamente, esto se expresa como:

$$f_s \geq 2 \cdot f_{max} \quad (3.1)$$

Este valor, $2 \cdot f_{max}$, se conoce como la Frecuencia de Nyquist. Si se muestrea una señal a una frecuencia inferior a la Frecuencia de Nyquist, se produce aliasing, donde las frecuencias más altas de la señal se reflejan como frecuencias más bajas en la señal muestreada, resultando en una reconstrucción incorrecta de la señal original.

Proceso de Muestreo

El proceso de muestreo consiste en tomar valores de la señal en intervalos discretos de tiempo. Estos valores son representaciones de la amplitud de la señal en puntos específicos en el tiempo. Después de muestrear, cada muestra se cuantifica y se codifica en formato binario para su procesamiento y almacenamiento digital.

Importancia y Aplicaciones

La Teoría de Muestreo es fundamental para diversas aplicaciones y campos, como las telecomunicaciones, la medicina, la música, y la ingeniería. Permite la conversión precisa de señales analógicas a digitales para el procesamiento, análisis, transmisión, y almacenamiento en sistemas digitales. Es especialmente crucial en el desarrollo de tecnologías de comunicación digital, procesamiento de audio e imagen, y en el diseño de sistemas de control digital.

3.4.2. Transformada de Fourier

La Transformada de Fourier es una técnica matemática vital que tiene la capacidad de descomponer una señal o función en componentes de frecuencia, revelando las frecuencias individuales que conforman la señal original. Esta transformación es fundamental en el análisis de señales ya que convierte una señal del dominio del tiempo al dominio de la frecuencia, proporcionando una perspectiva diferente y útil de la señal.

La premisa básica de la Transformada de Fourier es que cualquier señal, incluso las más complejas, pueden representarse como la suma de funciones sinusoidales, es decir, senos y cosenos, cada uno con su propia frecuencia, amplitud y fase. Entonces, una señal en el tiempo, o una función temporal, se transforma en una suma de sinusoides, representando así su composición frecuencial, como se muestra en la Figura 3.3.

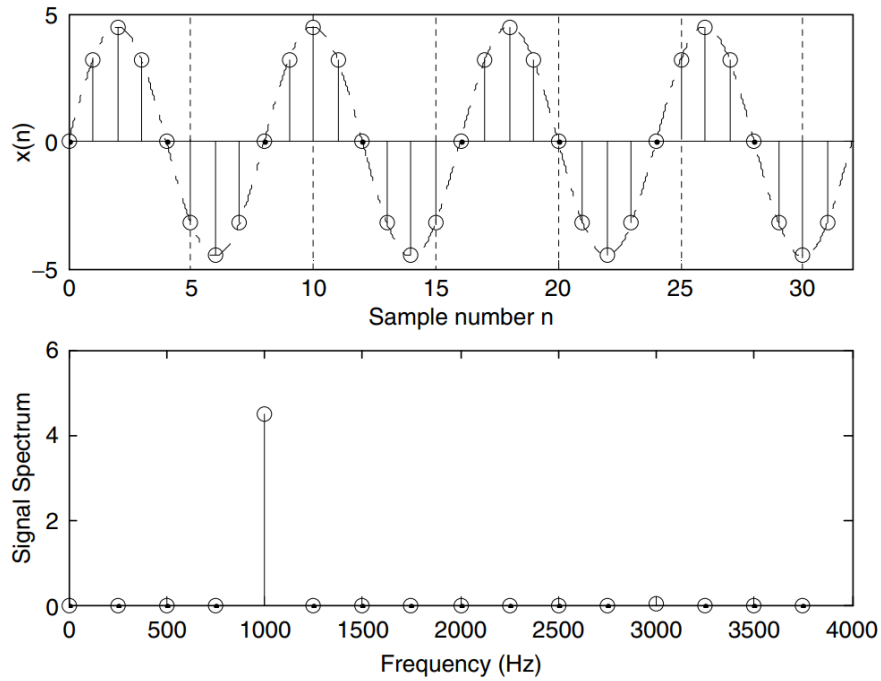


Figura 3.3: Ejemplo de una señal digital y su espectro de amplitud.

(Tan, 2008)

En el caso de señales continuas, la Transformada de Fourier se define mediante una integral que transforma la función o señal en una función de frecuencia compleja, donde cada punto en el dominio de frecuencia representa la amplitud y la fase de una sinusoidal de esa frecuencia particular. Para señales discretas, se utiliza la Transformada Discreta de Fourier (DFT), que realiza una operación similar para secuencias de números, proporcionando así una representación frecuencial de señales discretas o digitalizadas.

En términos prácticos, la implementación eficiente de la DFT, conocida como la Transformada Rápida de Fourier (FFT), es esencialmente un pilar en el procesamiento digital de señales, ya que reduce significativamente el tiempo de cómputo necesario para calcular los componentes frecuenciales de una señal, siendo crucial en aplicaciones en tiempo real y en sistemas con recursos limitados.

Las aplicaciones de la Transformada de Fourier son vastas y diversas, abarcando desde el análisis espectral, donde se estudian las componentes frecuenciales de señales, hasta el pro-

cesamiento de imágenes y audio, donde se realiza filtrado y modificación de señales en el dominio de la frecuencia. También es fundamental en el campo de las comunicaciones, donde la modulación y demodulación de señales se basa en gran medida en la manipulación de componentes frecuenciales de las señales.

3.4.2.1. Transformada Discreta de Fourier (DFT)

La Transformada Discreta de Fourier (DFT) es un método matemático utilizado para transformar una secuencia de números complejos (señal discreta) del dominio del tiempo al dominio de la frecuencia. Es una versión discreta de la Transformada de Fourier y es especialmente útil en el análisis y procesamiento de señales digitales.

La DFT convierte una secuencia de valores complejos, que usualmente representan amplitudes en puntos discretos en el tiempo, en otro conjunto de valores complejos que representan amplitudes y fases de sinusoides de frecuencias discretas. En otras palabras, permite expresar una señal en términos de sus componentes de frecuencia, revelando la presencia de sinusoides de diferentes frecuencias en la señal original.

Matemáticamente, la DFT de una secuencia $x[n]$ de N números se define como:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j(2\pi/N) \cdot k \cdot n} \quad (3.2)$$

Aquí, $X[k]$ representa los coeficientes de la DFT, cada uno correspondiente a una frecuencia diferente. Cada coeficiente $X[k]$ es un número complejo que indica la amplitud y la fase de una sinusoidal de una frecuencia específica.

3.4.2.2. Transformada Rápida de Fourier (FFT)

La Transformada Rápida de Fourier (FFT, por sus siglas en inglés Fast Fourier Transform) es un algoritmo que calcula de manera eficiente la Transformada Discreta de Fourier (DFT) y su inversa. La DFT transforma una secuencia de números complejos (señal en el dominio del tiempo) a otra secuencia de números complejos (señal en el dominio de la frecuencia),

permitiendo el análisis de las componentes frecuenciales de una señal.

Eficiencia Computacional

Mientras que la DFT tiene una complejidad computacional de $O(N^2)$, donde N es el número de puntos de la señal, la FFT reduce esta complejidad a $O(N \log N)$, permitiendo la transformación de señales más largas de manera eficiente. Esto la hace particularmente valiosa en una gran variedad de aplicaciones prácticas, como el procesamiento de señales, imágenes y sonidos, y en la solución de ecuaciones diferenciales parciales, entre otras.

Desarrollo del Algoritmo

El algoritmo FFT fue popularizado por James W. Cooley y John W. Tukey en 1965, aunque algunas versiones ya habían sido propuestas anteriormente. El algoritmo de Cooley-Tukey, también conocido como la FFT Radix-2, es quizás la versión más conocida de la FFT, y es especialmente eficiente cuando el número de puntos de la señal, N , es una potencia de 2.

Funcionamiento del Algoritmo

La FFT explota la simetría y periodicidad de las funciones trigonométricas para reducir el número de cálculos necesarios para obtener la DFT. Divide la secuencia de entrada en subsecuencias más pequeñas y calcula sus DFT de manera recursiva. Estas DFT más pequeñas se combinan para obtener la DFT de la secuencia completa.

3.4.3. Transformada Wavelet

La Transformada Wavelet es una técnica matemática utilizada para descomponer señales, muy similar en propósito a la Transformada de Fourier, pero con algunas diferencias clave. Mientras que la Transformada de Fourier descompone la señal en sinusoides de diferentes frecuencias, la Transformada Wavelet descompone la señal en “wavelets”, que son funciones

matemáticas construidas de tal manera que tienen un valor medio de cero, como se muestra en la Figura 3.4.

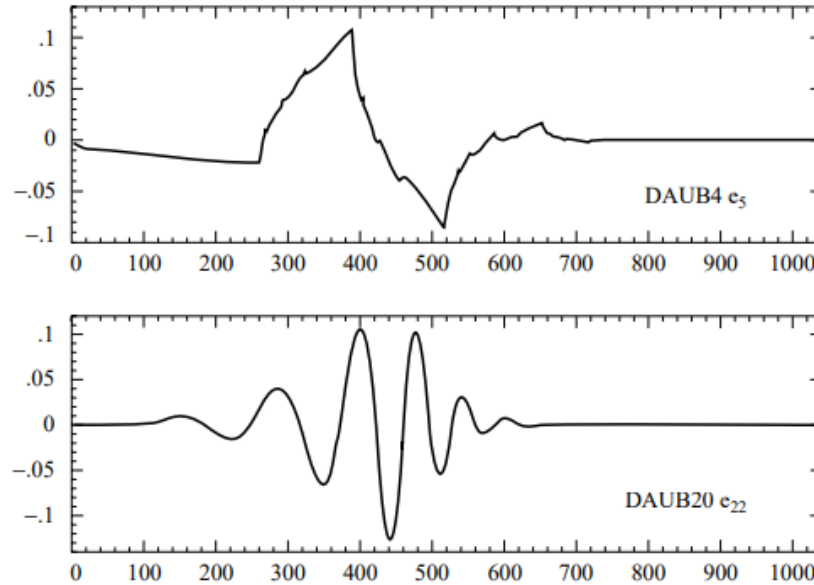


Figura 3.4: Funciones Wavelet, es decir, funciones de base única de las familias DAUB4 y DAUB20.

(Press, 2009)

Características de la Transformada Wavelet

Una de las principales características de la Transformada Wavelet es su capacidad para analizar señales a diferentes escalas o resoluciones, proporcionando así una representación multiresolución de la señal. Esto permite un análisis más detallado de la señal en diferentes puntos temporales y frecuenciales, siendo muy útil para analizar señales no estacionarias, es decir, señales cuyas características frecuenciales cambian con el tiempo.

Funcionamiento del Wavelet

La Transformada Wavelet utiliza funciones llamadas wavelets, que son generadas por dilataciones y traslaciones de una función madre wavelet. Esto permite representar una señal

como una suma de wavelets dilatados y trasladados. El proceso de transformada produce coeficientes que representan la correlación entre la señal y el wavelet en diferentes escalas y posiciones, ofreciendo información sobre la presencia de características en la señal a diferentes niveles de detalle.

Aplicaciones de la Transformada Wavelet

La Transformada Wavelet tiene una variedad de aplicaciones, incluyendo:

1. Procesamiento de Imágenes y Señales: Para la detección de bordes, reducción de ruido, y compresión.
2. Medicina: En el análisis de señales biomédicas como los electrocardiogramas.
3. Análisis de Audio: Para la detección y análisis de características musicales y de sonido.
4. Finanzas: En el análisis de series temporales financieras para la detección de tendencias y anomalías.

Ventajas sobre la Transformada de Fourier

La principal ventaja de la Transformada Wavelet sobre la Transformada de Fourier es su capacidad para proporcionar información temporal y frecuencial simultáneamente. La Transformada de Fourier, especialmente en su forma básica, proporciona información solo en el dominio de la frecuencia, sin información sobre el momento exacto en el que ocurren los eventos en la señal. La Transformada Wavelet, en cambio, puede mostrar cómo el contenido frecuencial de una señal cambia con el tiempo.

3.4.3.1. Transformada Wavelet Continua (CWT)

La Transformada Wavelet Continua (CWT, por sus siglas en inglés) es una técnica utilizada para descomponer una señal en función de wavelets a diferentes escalas y traslaciones. A diferencia de su contraparte discreta (DWT), que opera en escalas y traslaciones discretas,

la CWT analiza la señal a lo largo de un continuo de escalas y traslaciones. Esto proporciona una vista más detallada y granular del contenido de la señal.

La CWT de una señal $x(t)$ se define mediante la siguiente fórmula:

$$CWT_x(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t - b}{a} \right) dt \quad (3.3)$$

Donde:

- a es el parámetro de escala. Cuando a es mayor que 1, estamos observando una dilatación del wavelet (es decir, una versión estirada), y cuando a es menor que 1, estamos observando una compresión.
- b es el parámetro de traslación, que desplaza la función wavelet a lo largo de la señal.
- $\psi^*(t)$ es el complejo conjugado de la función wavelet madre, una función que cumple ciertas condiciones (como tener un promedio de cero).

Interpretación

La idea básica detrás de la CWT es "deslizar" la función wavelet a través de la señal $x(t)$ y calcular la similitud entre la señal y el wavelet en cada punto. Esta operación se realiza a diferentes escalas (es decir, diferentes frecuencias) y traslaciones. Cuando el coeficiente de la CWT es grande en magnitud en un punto particular (a, b) , indica que hay una fuerte correlación entre la señal y la wavelet madre dilatada y trasladada en esa escala y posición específica. Esto ayuda a detectar características o eventos en la señal que tienen una forma similar a la wavelet en una escala específica.

Aplicaciones

Debido a su naturaleza continua y su capacidad para proporcionar información detallada a diferentes escalas, la CWT es particularmente útil en aplicaciones donde se requiere una alta resolución temporal y frecuencial, como:

1. Detección de eventos transitorios: Puedes identificar eventos que tienen una duración corta en una señal.
2. Análisis de señales no estacionarias: Cuando las características de la señal cambian con el tiempo.

3.4.3.2. Transformada Wavelet Discreta (DWT)

La Transformada Wavelet Discreta (DWT, por sus siglas en inglés) es una técnica de procesamiento de señales que descompone una señal en diferentes niveles de detalle, utilizando un conjunto de funciones matemáticas conocidas como wavelets. A diferencia de la Transformada Wavelet Continua (CWT), la DWT opera sobre escalas y posiciones discretas.

La Transformada Wavelet Discreta de una señal se puede definir mediante la relación:

$$DWT_x[j, k] = \sum_{n=0}^{N-1} x[n] \psi^* (2^j n - k) \quad (3.4)$$

Aquí, j y k son enteros que representan la escala y la posición, respectivamente, y N es el número total de muestras en la señal. El término ψ^* representa el complejo conjugado de la función wavelet madre.

Proceso de Descomposición

En la DWT, una señal se descompone en dos partes en cada nivel de la transformada: los coeficientes de aproximación y los coeficientes de detalle.

- Los coeficientes de aproximación representan la parte de baja frecuencia de la señal, proporcionando una visión general o aproximada de la señal.
- Los coeficientes de detalle representan la parte de alta frecuencia de la señal, proporcionando información sobre los cambios rápidos y detalles finos en la señal.

Este proceso de descomposición se puede repetir en los coeficientes de aproximación para obtener una representación multiresolución de la señal original.

3.4.4. Clasificadores (Machine Learning)

En Machine Learning, un clasificador es un modelo que se utiliza para asignar una etiqueta de clase a un objeto en función de sus características. En otras palabras, un clasificador es un algoritmo de aprendizaje automático que se entrena para distinguir entre diferentes clases de objetos en un conjunto de datos.

Los clasificadores son un tipo común de modelo de aprendizaje supervisado en el que se proporciona al modelo un conjunto de datos etiquetados, es decir, un conjunto de datos en el que cada objeto ya tiene una etiqueta de clase conocida. El modelo utiliza estos datos etiquetados para aprender a distinguir entre las diferentes clases.

Por ejemplo, en un problema de clasificación de correo electrónico, el modelo podría aprender a distinguir entre correos electrónicos legítimos y spam. El modelo se entrena utilizando un conjunto de datos etiquetados que incluye correos electrónicos conocidos que son legítimos y correos electrónicos conocidos que son spam. El modelo utiliza las características de los correos electrónicos, como el remitente, el asunto y el contenido, para determinar si un correo electrónico nuevo es legítimo o spam.

Los clasificadores pueden ser de diferentes tipos, como el algoritmo de k-vecinos más cercanos (k-NN), árboles de decisión, redes neuronales, etc. La elección del clasificador depende de la naturaleza del problema de clasificación y de las características de los datos.

Una vez que se ha entrenado el clasificador, se puede utilizar para clasificar objetos no etiquetados. El modelo examina las características de los objetos no etiquetados y utiliza la información que ha aprendido durante el entrenamiento para asignar una etiqueta de clase a cada objeto.

A continuación se describen algunos elementos importantes que deben considerarse en un clasificador:

- Exploratory Data Analysis (EDA): Es un enfoque en el análisis de datos que se enfoca en descubrir patrones, relaciones y tendencias en los datos. El EDA se realiza típicamente en una etapa temprana del proceso de análisis de datos y se utiliza para comprender

mejor las características del conjunto de datos. El EDA implica la exploración y visualización de los datos para identificar patrones y anomalías. Algunas de las técnicas comunes utilizadas en el EDA incluyen la creación de gráficos de dispersión, histogramas, diagramas de caja, matrices de correlación y diagramas de densidad, entre otros. El EDA también puede incluir la identificación de valores atípicos, la imputación de datos faltantes y la detección de valores extremos. También puede ayudar a identificar las características importantes o relevantes en el conjunto de datos, lo que puede ser útil para seleccionar las variables adecuadas para el análisis. El EDA se considera una práctica importante en el análisis de datos porque ayuda a los analistas a comprender mejor los datos que están analizando y a tomar decisiones más informadas sobre cómo procesar y modelar los datos. Además, el EDA puede ayudar a identificar problemas en los datos, como datos faltantes o errores de medición, que pueden afectar la calidad del análisis.

- **Model Evaluation:** El model evaluation (evaluación del modelo) es un proceso crítico en el aprendizaje automático, que implica la evaluación del rendimiento del modelo de clasificación en términos de su capacidad para hacer predicciones precisas en datos no vistos. En el contexto de un algoritmo de clasificación, el model evaluation implica la evaluación del modelo en un conjunto de datos de prueba independiente, que no se ha utilizado para entrenar el modelo. El objetivo es evaluar la capacidad del modelo para generalizar a datos no vistos y prevenir el sobreajuste al conjunto de datos de entrenamiento.

3.4.4.1. Clasificador ANN (Artificial Neural Network)

Las redes neuronales artificiales (ANN) son una familia de modelos utilizados en el aprendizaje de máquina. Aunque existen muchos tipos de ANN, todas se basan en el mismo principio básico: el procesamiento paralelo de información a través de unidades de cálculo interconectadas, llamadas "neuronas". En la Figura 3.5 se puede apreciar el proceso básico.

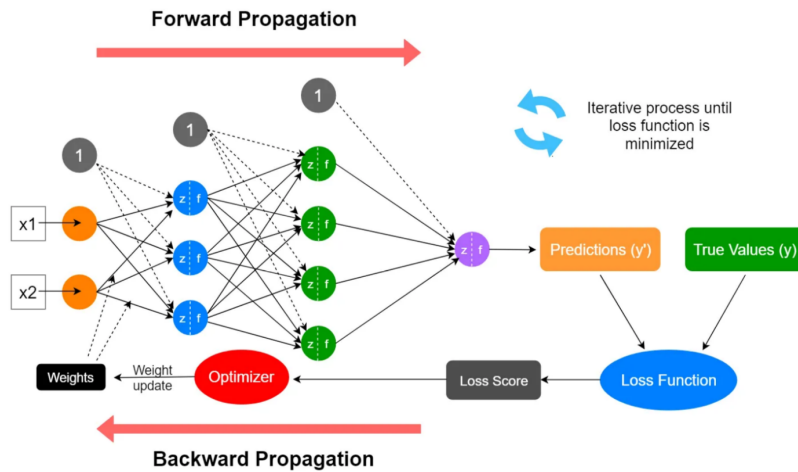


Figura 3.5: Proceso iterativo de aprendizaje (entrenamiento) de una red neuronal.

(Pramoditha, 2022)

El algoritmo de clasificación en una ANN funciona en esencia a través de dos etapas principales: la fase de entrenamiento y la fase de clasificación o predicción (**ANN_1**).

1. Fase de entrenamiento: La red se entrena a través de un conjunto de datos de entrenamiento, que consta de entradas y salidas correspondientes. Cada neurona recibe una entrada, la pondera en relación con su peso (que inicialmente se establece de manera aleatoria), aplica una función de activación y luego pasa el resultado a la siguiente neurona. Este proceso se repite a través de todas las capas de la red hasta llegar a la salida. A continuación, se compara la salida de la red con la salida deseada y se calcula el error. Este error se propaga de vuelta a través de la red en un proceso llamado "back-propagation", y los pesos de la red se ajustan para minimizar el error. Este proceso se repite para cada ejemplo en el conjunto de datos de entrenamiento, a menudo múltiples veces, hasta que el error de la red sobre el conjunto de datos de entrenamiento es lo suficientemente bajo.
2. Fase de clasificación o predicción: Una vez que la red ha sido entrenada, puede utilizarse para clasificar nuevas entradas. Cada nueva entrada se pasa a través de la red de la misma manera que durante el entrenamiento, pero esta vez no se realiza ninguna

backpropagation ni ajuste de los pesos. La salida de la red es la clasificación de la entrada.

Es importante mencionar que existen varias técnicas y enfoques para entrenar una red neuronal artificial y también diferentes tipos de redes neuronales artificiales, como las redes perceptrón multicapa, las redes convolucionales, las redes recurrentes, etc. Cada una de estas redes puede ser más adecuada para ciertos tipos de problemas de clasificación que otras (**ANN_2**). Por ejemplo, las redes convolucionales son ampliamente utilizadas para la clasificación de imágenes, mientras que las redes recurrentes son a menudo la elección preferida para problemas de secuencias, como el procesamiento del lenguaje natural.

MLP (Multilayer Perceptron)

El perceptrón multicapa (MLP, por sus siglas en inglés), es una clase de red neuronal artificial que se compone de al menos tres capas de nodos: una capa de entrada, una o más capas ocultas y una capa de salida. A diferencia de un perceptrón simple, que sólo puede resolver problemas linealmente separables, un MLP puede resolver problemas que son no linealmente separables.

1. Capa de entrada: La primera capa de la red es la capa de entrada. Cada nodo en esta capa representa una característica de entrada. La capa de entrada toma el conjunto de datos y los pasa a la siguiente capa (la primera capa oculta) sin modificarlos.
2. Capas ocultas: Las capas ocultas son donde ocurre la mayor parte del cálculo en un MLP. Cada nodo en una capa oculta representa una combinación lineal de las salidas de la capa anterior, seguida de una transformación no lineal, o función de activación. Existen varias funciones de activación que se utilizan comúnmente, como la función sigmoideal, la tangente hiperbólica y la función ReLU (unidad lineal rectificadora).
3. Capa de salida: La última capa de la red es la capa de salida. Cada nodo en la capa de salida representa una combinación lineal de las salidas de la última capa oculta. En un problema de clasificación, cada nodo podría representar una clase diferente, y la red

podría ser entrenada para activar el nodo correspondiente a la clase correcta para cada entrada.

Durante el entrenamiento, el MLP utiliza un algoritmo llamado backpropagation para ajustar los pesos de la red. La red comienza con pesos aleatorios, y utiliza la retropropagación para actualizar iterativamente los pesos para minimizar la diferencia entre la salida prevista de la red y la salida real para cada entrada en el conjunto de entrenamiento.

El MLP es un tipo de red "feedforward", lo que significa que la información fluye a través de la red en una sola dirección: de la capa de entrada, a través de las capas ocultas, y finalmente a la capa de salida, como se observa en la Figura 3.6. No hay conexiones de retroalimentación en las que las salidas de la red se retroalimenten en sí mismas. Esto es diferente de otros tipos de redes neuronales, como las redes neuronales recurrentes, que sí tienen conexiones de retroalimentación.

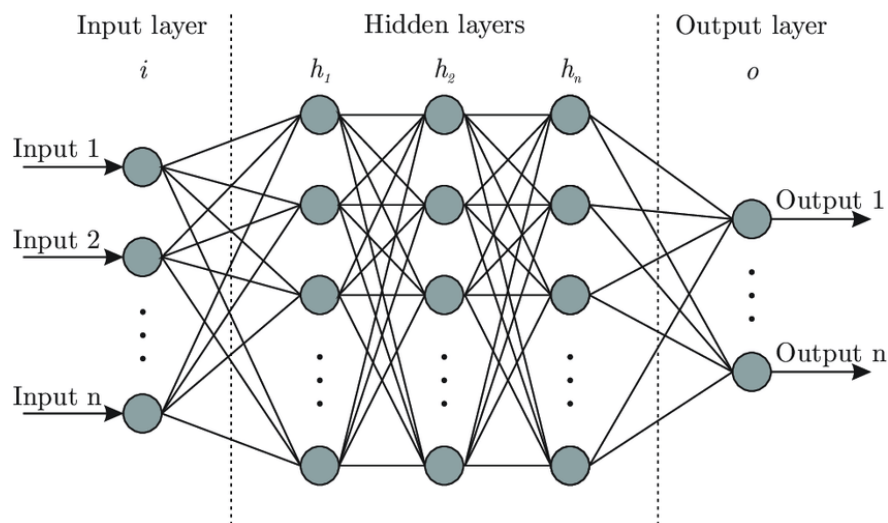


Figura 3.6: Ejemplificación de la arquitectura del perceptrón multicapa (Bre et al., 2017).

Convolutional Neural Network (CNN)

Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés de Convolutional Neural Networks) son una categoría especializada de redes neuronales que han demostrado ser particularmente eficaces en tareas relacionadas con el procesamiento de imágenes. Están diseñadas

para reconocer patrones visuales directamente desde imágenes con mínima preprocesamiento. Aunque inicialmente fueron desarrolladas para el análisis de imágenes, se han adaptado y utilizado en una amplia variedad de aplicaciones, desde el análisis de secuencias hasta el procesamiento de audio. A continuación se describen las principales características de este tipo de red:

1. Capas Convolucionales: Esta es la piedra angular de las CNNs. Durante la convolución, pequeñas matrices (llamadas filtros o núcleos) se desplazan a lo largo de la imagen para producir un mapa de características. Estos filtros son entrenables, lo que significa que la red aprende los filtros más adecuados para extraer características relevantes de una imagen.
2. Función de Activación: Por lo general, después de la convolución, se aplica una función de activación, comúnmente la función ReLU (Rectified Linear Unit), que introduce no linealidades en el modelo.
3. Capas de Agrupamiento (Pooling): Estas capas reducen la dimensionalidad espacial de la entrada para disminuir la cantidad de parámetros y cálculos en la red. Un tipo común de agrupamiento es el "max pooling", que selecciona el valor máximo de un grupo particular de valores.
4. Capas Totalmente Conectadas (Fully Connected): Estas capas son similares a las capas en una red neuronal tradicional. Una vez que las características han sido extraídas por las capas anteriores, las capas totalmente conectadas interpretan estas características y producen la salida final de la red.
5. Capa de Aplanamiento (Flatten): Antes de pasar los mapas de características a las capas totalmente conectadas, es común aplanarlos o convertirlos en un vector.

El entrenamiento de redes neuronales suele ser la parte más prolongada y desafiante en la creación de CNNs (Redes Neuronales Convolucionales) para el aprendizaje profundo. En la

etapa de aprendizaje supervisado, se enseña a la red a realizar una tarea específica, como la clasificación de imágenes. Este proceso implica la recolección de un conjunto de datos extenso, que puede contener miles o millones de imágenes. Estas imágenes se suministran a la red, permitiéndole predecir lo que cada imagen representa. Si la predicción es incorrecta, es necesario actualizar las neuronas con la respuesta correcta para asegurar que las predicciones futuras para la misma imagen sean precisas. Dicho proceso se repite hasta que los desarrolladores consideren que la exactitud de las predicciones de la red neuronal es satisfactoria.

LSTM (Long Short Term Memory)

La Memoria a Largo y Corto Plazo (LSTM) es una variante de las redes neuronales recurrentes. En las RNN, la salida de un paso anterior se utiliza como entrada en el paso actual. La LSTM fue propuesta por Hochreiter y Schmidhuber para solucionar el problema de las dependencias a largo plazo en las RNN. Estas redes a menudo no pueden predecir palabras que dependen de información almacenada hace mucho tiempo, pero son más precisas con datos recientes. Las RNN convencionales no son eficientes cuando el intervalo entre datos relevantes es muy largo. Por otro lado, las LSTM están diseñadas para retener información durante períodos extensos y son útiles para procesar, predecir y clasificar datos secuenciales como series temporales.

La Memoria a Largo y Corto Plazo (LSTM) es una red neuronal recurrente especializada en manejar datos secuenciales como series de tiempo, voz y texto. Estas redes pueden aprender dependencias a largo plazo en datos secuenciales, lo que las hace aptas para tareas como traducción de idiomas, reconocimiento de voz y predicción de series temporales.

Una RNN convencional tiene un único estado oculto que se transmite a lo largo del tiempo, lo cual puede complicar el aprendizaje de dependencias de largo alcance. Las LSTM resuelven esto con una celda de memoria, que puede mantener información durante mucho tiempo. Esta celda es gestionada por tres compuertas: de entrada, de olvido y de salida. Estas compuertas determinan qué información se añade, se elimina o se emite desde la celda de memoria.

La compuerta de entrada decide qué información se introduce en la celda. La de olvido

determina qué datos se descartan de la celda, y la de salida establece qué información se extrae de ella. Gracias a esta estructura, las LSTM pueden decidir qué información conservar o eliminar, permitiéndoles aprender dependencias a largo plazo.

Las LSTM se pueden apilar para formar redes LSTM profundas, lo que permite aprender patrones aún más complejos en datos secuenciales. Además, las LSTM pueden combinarse con otras arquitecturas de redes neuronales, como las Redes Neuronales Convolucionales (CNN) para el análisis de imágenes y videos.

Ventajas de LSTM:

1. Las redes LSTM son capaces de capturar dependencias a largo plazo gracias a su celda de memoria especializada en almacenar información a lo largo del tiempo.
2. A diferencia de las RNN tradicionales, que enfrentan problemas de gradientes que desaparecen o explotan al entrenarse con secuencias largas, las LSTM solucionan este inconveniente a través de un mecanismo de compuertas que selecciona o descarta información de manera selectiva.
3. Las LSTM permiten que el modelo capture y recuerde contextos importantes, incluso cuando existe un intervalo de tiempo considerable entre eventos relevantes en una secuencia. Por esto, se utilizan en tareas donde entender el contexto es crucial, como en la traducción automática.

Desventajas de LSTM:

1. En comparación con arquitecturas más sencillas, como las redes neuronales feed-forward, las redes LSTM son computacionalmente más costosas. Esto puede limitar su escalabilidad para conjuntos de datos de gran tamaño o en entornos con recursos limitados.
2. El entrenamiento de las redes LSTM puede ser más prolongado en comparación con modelos más simples, debido a su complejidad computacional. Por lo tanto, entrenar LSTM a menudo requiere más datos y tiempos de entrenamiento prolongados para lograr un alto rendimiento.

3. Dado que procesan las palabras de manera secuencial, es difícil paralelizar el trabajo al procesar oraciones.

3.4.5. Herramientas de validación estadística

3.4.5.1. Análisis de componentes principales (PCA)

El análisis de componentes principales (PCA, por sus siglas en inglés) es una técnica de reducción de dimensionalidad utilizada en estadística y aprendizaje automático para identificar las características más importantes o relevantes en un conjunto de datos. PCA se utiliza para transformar un conjunto de datos de alta dimensionalidad en un conjunto de datos de menor dimensionalidad, mientras se mantiene la mayor parte de la variabilidad en los datos.

El objetivo principal de PCA es encontrar una transformación lineal de los datos originales en un nuevo conjunto de variables (componentes principales) que son ortogonales entre sí y que capturan la mayor cantidad posible de variabilidad en los datos originales. La primera componente principal es la dirección en la que los datos tienen la mayor varianza, la segunda componente principal es la dirección con la mayor varianza restante, y así sucesivamente. En general, las componentes principales se ordenan de acuerdo a su varianza, de mayor a menor.

(MDC_6; MDC_7)

Para calcular las componentes principales, se realiza una descomposición de valores propios de la matriz de covarianza o de la matriz de correlación de los datos originales. Esta descomposición de valores propios identifica los vectores propios y los valores propios de la matriz de covarianza o correlación, que a su vez definen las componentes principales.

La reducción de dimensionalidad que proporciona PCA puede tener varios beneficios, como una mayor eficiencia computacional en algoritmos de aprendizaje automático, una mejor visualización de los datos y una reducción del ruido y redundancia en los datos.

Es importante destacar que PCA asume que los datos están linealmente correlacionados y que las variables tienen igual importancia en la transformación. En algunos casos, donde las relaciones no son lineales o donde algunas variables son más importantes que otras, puede ser

necesario utilizar técnicas de reducción de dimensionalidad más avanzadas y personalizadas.

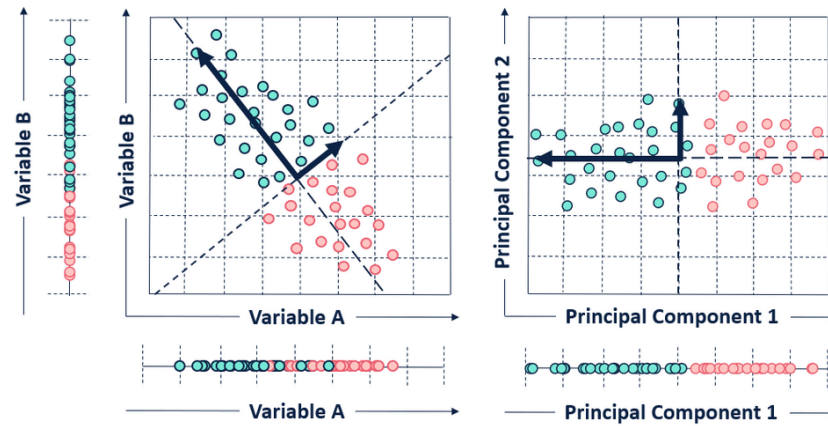


Figura 3.7: Eigenvectores del proceso de reducción de dimensionalidad. El componente principal 1 corresponde a la varianza de ambas variables, A y B (Alumni, 2022).

3.4.5.2. Validación cruzada k-fold

La validación k-fold es una técnica de validación cruzada que se utiliza para evaluar la capacidad de generalización de un modelo de aprendizaje automático. Esta técnica se utiliza para estimar el rendimiento del modelo en datos no vistos, es decir, en datos que no se han utilizado para entrenar el modelo.

El proceso de validación k-fold implica dividir el conjunto de datos en k subconjuntos o "pliegues" de tamaño similar. A continuación, se entrena el modelo en k-1 de los subconjuntos y se evalúa en el subconjunto restante. Este proceso se repite k veces, utilizando cada uno de los subconjuntos como conjunto de validación exactamente una vez. En cada iteración, se evalúa el desempeño del modelo en el subconjunto de validación y se registra una medida de rendimiento, como la precisión o el error de predicción, como se observa en la Figura 3.8(MDC_8).

Al final de las k iteraciones, se promedian las medidas de rendimiento para obtener una estimación general del rendimiento del modelo en datos no vistos. Esta técnica de validación cruzada ayuda a evitar el sobreajuste del modelo, ya que se entrena y evalúa en diferentes

subconjuntos de datos en cada iteración.

La validación k-fold se utiliza comúnmente para ajustar los hiperparámetros del modelo, ya que permite una evaluación más precisa y robusta de los hiperparámetros. Por ejemplo, se pueden probar diferentes valores de hiperparámetros en cada iteración y evaluar el rendimiento del modelo en datos no vistos.

En resumen, la validación k-fold es una técnica de validación cruzada que se utiliza para evaluar el rendimiento de un modelo de aprendizaje automático en datos no vistos, evitando el sobreajuste y permitiendo una evaluación más precisa y robusta de los hiperparámetros del modelo. (MDC_1; MDC_2)

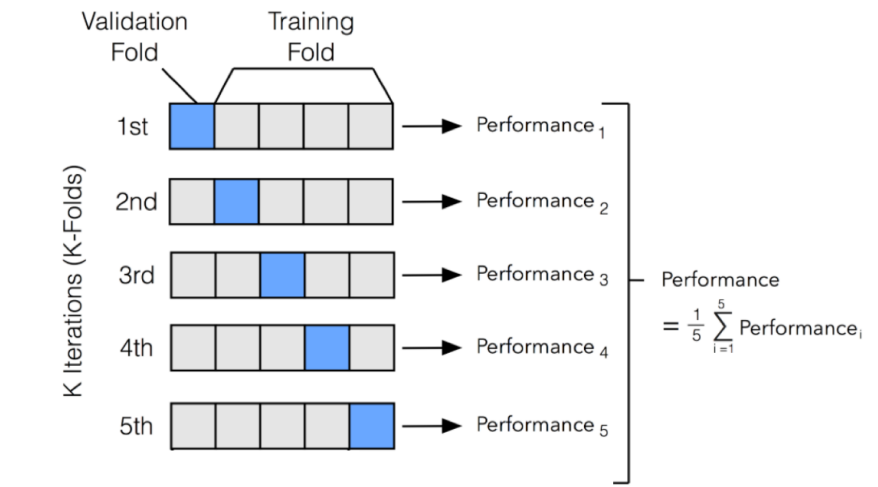


Figura 3.8: Ejemplificación del funcionamiento de la validación cruzada k-fold (Guerrero, 2021).

3.4.6. Evaluadores de modelo

3.4.6.1. Matriz de confusión

Una matriz de confusión es una herramienta que se utiliza para evaluar el desempeño de un modelo de clasificación. Esta matriz muestra el número de predicciones correctas e incorrectas que hace un modelo en cada una de las clases. (MDC_3)

La matriz de confusión se construye comparando las predicciones del modelo con las clases reales de los datos. En la matriz, cada fila representa la clase real, mientras que cada columna representa la clase predicha por el modelo. Los elementos de la diagonal principal de la matriz representan las predicciones correctas, mientras que los elementos fuera de la diagonal principal representan las predicciones incorrectas. (**MDC_4**)

Por ejemplo, en una matriz de confusión de dos clases (Tabla 3.1), la primera fila podría representar la clase "positivo" y la segunda fila podría representar la clase "negativo". Si el modelo clasifica correctamente 20 ejemplos como positivos y 30 ejemplos como negativos, mientras que clasifica incorrectamente 5 ejemplos positivos como negativos y 10 ejemplos negativos como positivos, la matriz de confusión se vería así:

Cuadro 3.1: Ejemplo de matriz de confusión.

	Clase predicha: "Positivo"	Clase predicha: "Negativo"
Clase real: "Positivo"	20	5
Clase real: "Negativo"	10	30

A partir de la matriz de confusión, se pueden calcular diversas métricas de evaluación del modelo, como la precisión, la sensibilidad y la especificidad, entre otras. Estas métricas son útiles para entender cómo se desempeña el modelo en cada clase y para comparar diferentes modelos de clasificación. En general, una buena matriz de confusión se caracteriza por tener valores altos en la diagonal principal y valores bajos fuera de la diagonal principal.

3.4.6.2. Métricas de clasificación

Las métricas de clasificación son medidas que se utilizan para evaluar el rendimiento de un modelo de clasificación. Las métricas más comunes son las siguientes:

1. Exactitud (accuracy): mide el porcentaje de predicciones correctas del modelo en el conjunto de datos de prueba. Es la proporción de casos totales que se han clasificado

correctamente.

$$Accuracy = \frac{(VP + VN)}{(VP + FP + VN + FN)} \quad (3.5)$$

2. Precisión (precision): mide la proporción de predicciones positivas verdaderas entre todas las predicciones positivas, es decir, cuántas de las predicciones positivas son realmente verdaderas. La precisión se utiliza para evaluar la exactitud del modelo cuando predice una clase positiva.

$$Precision = \frac{VP}{(VP + FP)} \quad (3.6)$$

3. Recall (sensibilidad): mide la proporción de predicciones positivas verdaderas entre todos los casos positivos reales en el conjunto de datos de prueba, es decir, cuántos de los casos positivos fueron detectados correctamente por el modelo. El recall se utiliza para evaluar la capacidad del modelo para detectar correctamente los casos positivos.

$$Recall = \frac{VP}{(VP + FN)} \quad (3.7)$$

4. Especificidad: mide la proporción de predicciones negativas verdaderas entre todos los casos negativos reales en el conjunto de datos de prueba, es decir, cuántos de los casos negativos fueron detectados correctamente por el modelo.

$$Especificidad = \frac{VN}{(FP + VN)} \quad (3.8)$$

5. F1 Score: es una medida que combina la precisión y el recall en una sola métrica. El F1 Score es la media armónica de la precisión y el recall, y es una buena medida de rendimiento cuando las clases están desequilibradas.

$$F1Score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (3.9)$$

En general, se utiliza una combinación de estas métricas para evaluar el rendimiento del modelo de clasificación. Por ejemplo, en un problema de detección de fraudes, el recall puede ser más importante que la precisión, ya que es más importante detectar correctamente los

casos de fraude, incluso si se generan algunos falsos positivos. Por otro lado, en un problema de diagnóstico médico, la precisión puede ser más importante que el recall, ya que es más importante no diagnosticar erróneamente a un paciente sano.

3.5. Placa de Circuito Impreso (PCB)

Una Placa de Circuito Impreso (PCB, por sus siglas en inglés de Printed Circuit Board) es un componente fundamental en electrónica que se utiliza para conectar física y eléctricamente componentes electrónicos mediante pistas conductivas, pad, y otras características grabadas desde láminas de cobre laminadas sobre un sustrato no conductor, generalmente de material de fibra de vidrio (véase Figura 3.9).

Estructura de un PCB

1. Capas de Cobre: Un PCB contiene una o más capas de cobre, que transportan corriente eléctrica. En PCBs multicapa, estas capas de cobre están separadas por capas de material aislante.
2. Sustrato (o Substrato): Es el material base de la PCB, y suele ser de fibra de vidrio (FR4). Proporciona rigidez mecánica a la PCB.
3. Máscara de Soldadura: Esta capa, usualmente de color verde, se coloca sobre la capa de cobre y ayuda a prevenir cortocircuitos, protege contra la corrosión y facilita el proceso de soldadura.
4. Serigrafía (o leyenda/silk-screen): Es la capa que contiene las indicaciones, números y letras, para facilitar la colocación y soldadura de los componentes.
5. Vías: Son pequeños agujeros llenos de metal que conectan las diferentes capas de cobre de un PCB multicapa.

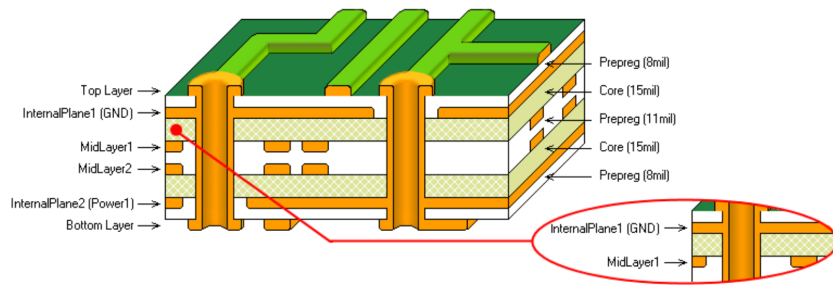


Figura 3.9: Esquema de apilamiento de capas de un PCB (AllPCB, 2017).

Diseño y fabricación

El diseño de un PCB se realiza mediante software especializado, y una vez diseñado, se envía a una empresa de fabricación de PCB para su producción. El proceso de fabricación implica la creación de las pistas de cobre, la perforación de los agujeros para los componentes y las vías, y la aplicación de la máscara de soldadura y serigrafía.

Aplicaciones

Los PCBs se utilizan en prácticamente todos los dispositivos electrónicos, desde dispositivos simples como calculadoras y relojes, hasta dispositivos más complejos como computadoras y teléfonos móviles. Permiten la conexión de componentes de manera ordenada y compacta, y son fundamentales en la creación de circuitos electrónicos fiables y robustos.

Ventajas

1. **Fiabilidad y Durabilidad:** Los PCBs son altamente eficaces y confiables, con una vida útil larga comparada con otros métodos de conexión de componentes.
2. **Compacto y Organizado:** Permiten una conexión densa y organizada de componentes, haciendo posible la creación de dispositivos electrónicos compactos y portátiles.
3. **Facilidad de Reparación y Mantenimiento:** Las conexiones son claras y los componentes

están fijos, facilitando la identificación de problemas y reparaciones.

4. Producción en Masa: El proceso de fabricación de PCBs está altamente automatizado, lo que facilita la producción en masa de circuitos de forma rápida y económica.

Consideraciones ambientales

Los PCBs, debido a su contenido de metales y otros materiales, deben ser desechados adecuadamente para evitar la contaminación ambiental. Existen procesos de reciclaje que permiten recuperar metales y otros materiales de las placas de circuito impreso.

Capítulo 4

Desarrollo

4.1. Sistema EEG-BCI basado en ESP32 (propuesta de sistema embebido)

4.1.1. Formulación del problema

A continuación, se presentan las pautas de diseño del sistema, que cuentan con el fuerte respaldo de la metodología desarrollada para la invención de la plataforma Creamino (Chiesi et al., 2019), así como las consideraciones más importantes a tener en cuenta al seleccionar chips de Convertidor Analógico a Digital (ADC) para EEG y medición de biopotenciales, postuladas en la investigación de Frey (Frey, 2016), y microprocesadores ARM compatibles con la herramienta X-CUBE-AI para la generación de bibliotecas a partir de redes neuronales pre-entrenadas y modelos clásicos de aprendizaje automático (STMicroelectronics, 2023).

4.1.2. Especificaciones del sistema

El punto crítico inicial reside en el paradigma de la filosofía "Hazlo tú mismo"(DIY, por sus siglas en inglés) frente a la compra de equipos de laboratorio comerciales (Frey, 2016), ya que la brecha entre ambos dispositivos (características, precisión de lectura, inmunidad al ruido,

software propietario optimizado, capacidades, certificaciones, normas, etc.) son abismales. Por lo tanto, es necesario seguir y adherirse a las pautas mínimas para cumplir con las restricciones de especificaciones de calidad de señal EEG comúnmente aceptadas en la práctica clínica (es decir, los estándares de la Federación Internacional de Neurofisiología Clínica [IFCN]) (Chiesi et al., 2019), para que el dispositivo que se cree pueda ser considerado como un equipo viable para la investigación en neurociencias.

En este caso, es conveniente adherirse a las capacidades proporcionadas por el Front End Analógico (AFE) TI ADS1299, que se utiliza en la placa de desarrollo OpenBCI Cyton para biosensores (OpenBCI, 2023). Las propiedades más notables se enumeran a continuación: 8 canales, resolución de 24 bits por canal, tasa de datos: 250 SPS a 16 kSPS y ADC delta-sigma de muestreo simultáneo integrado.

4.1.3. Arquitectura y desarrollo de hardware

La Figura 4.1 propone un diagrama de bloques que indica la arquitectura de hardware mínima que garantiza un sistema BCI funcional capaz de capturar señales EEG y procesarlas localmente:

Los autores de este trabajo actualmente están en el diseño de una plataforma BCI (bajo el esquema presente en este documento), teniendo hasta ahora, el diseño del PCB y su fabricación y ensamblaje en proceso de ser realizados para comenzar la medición de la integridad de la señal, cargar el firmware y depurar en general.

Para el diseño del PCB, la casa de manufactura y ensamblaje comenzó desde el principio a tener las características dieléctricas del apilamiento de capas como una forma de mantener el control de las impedancias a lo largo del enrutamiento de las conexiones en todo momento.

La Figura 4.2 muestra el frente de la tarjeta, que incluye una entrada micro-USB para cargar la batería y alimentar el sistema en general (no se recomienda usar esta opción debido a los armónicos presentes en la fuente de alimentación desde este puerto, que pueden degradar la calidad de la señal y añadir artefactos, por lo que se sugiere cargar solo la batería y usarla como fuente principal) con su respectiva circuitería para acondicionamiento de voltaje de bus,

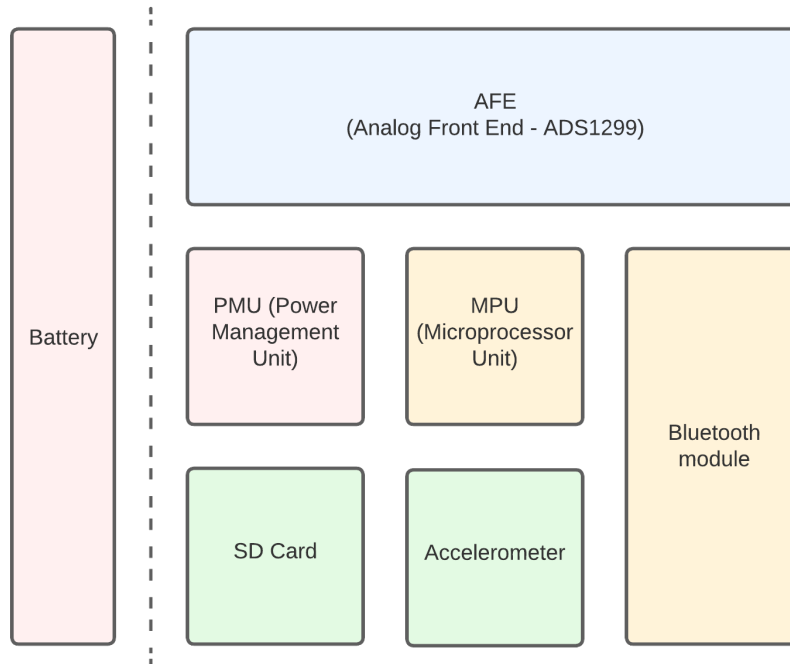


Figura 4.1: El diagrama de bloques de un diseño de sistema embebido que incorpora un RTOS.

protección contra sobrecorriente y un puente USB a UART. Asimismo, se incluye un conector JST para conexión a una batería Li-Ion (18650) y reguladores de voltaje que alimentan la sección digital (microprocesador y módulo Bluetooth) y el AFE con sus entradas conectadas a cabezales de pines que corresponden a los electrodos. Finalmente, es importante resaltar la presencia de dos opciones de acelerómetro debido a la escasez en la cadena de suministro de semiconductores.

Además, la Figura 4.2 muestra la parte posterior de la tarjeta, que solo tiene un conector para memoria micro-SD y el microprocesador principal del dispositivo (responsable de la adquisición, filtrado, procesamiento y clasificación de señales con TinyML).

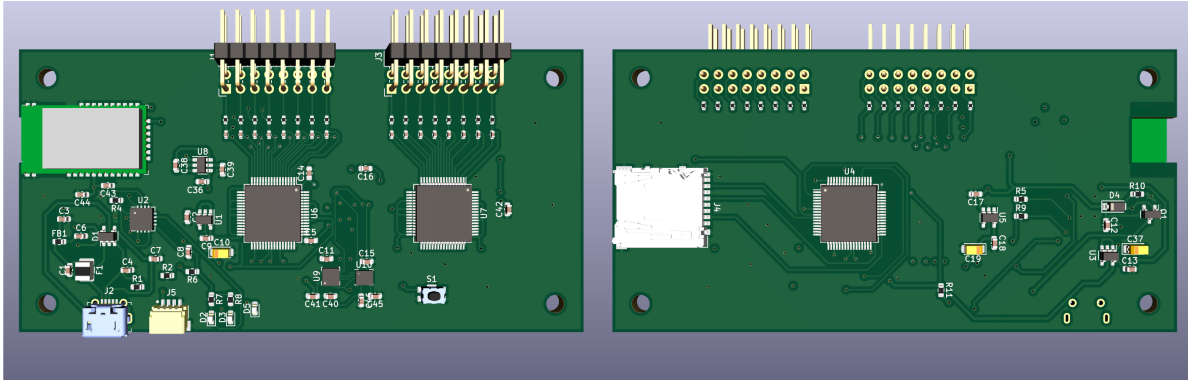


Figura 4.2: Vista 3D desde el frente (render izquierdo) y la parte posterior del PCB (render derecho).

No hay que dejar de lado la necesidad subyacente (para todo tipo de desarrollo de producto electrónico) de seguir una filosofía de modularización del sistema. Eventualmente se proporciona la capacidad de probar cada etapa o sección, por separado, de un dispositivo complejo, facilitando el proceso de aislamiento de errores y acceso puntualizado de los periféricos para llevar a cabo la creación de *drivers* de bajo nivel. Lo anterior se ve reflejado en las Figuras 4.3 y 4.4:

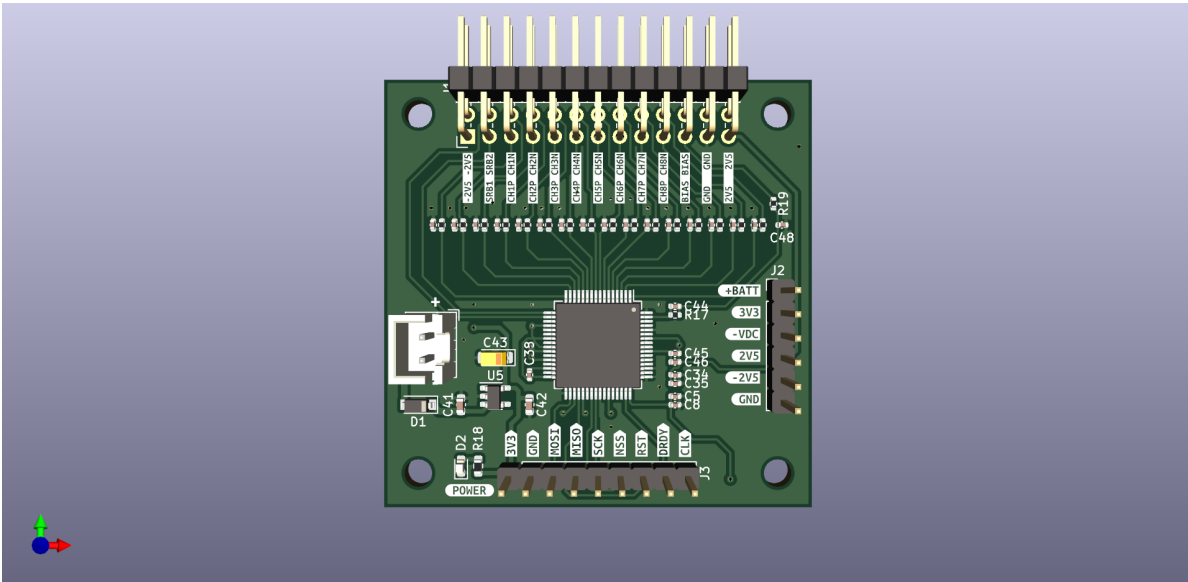


Figura 4.3: Vista 3D frontal del módulo de AFE ADS1299.

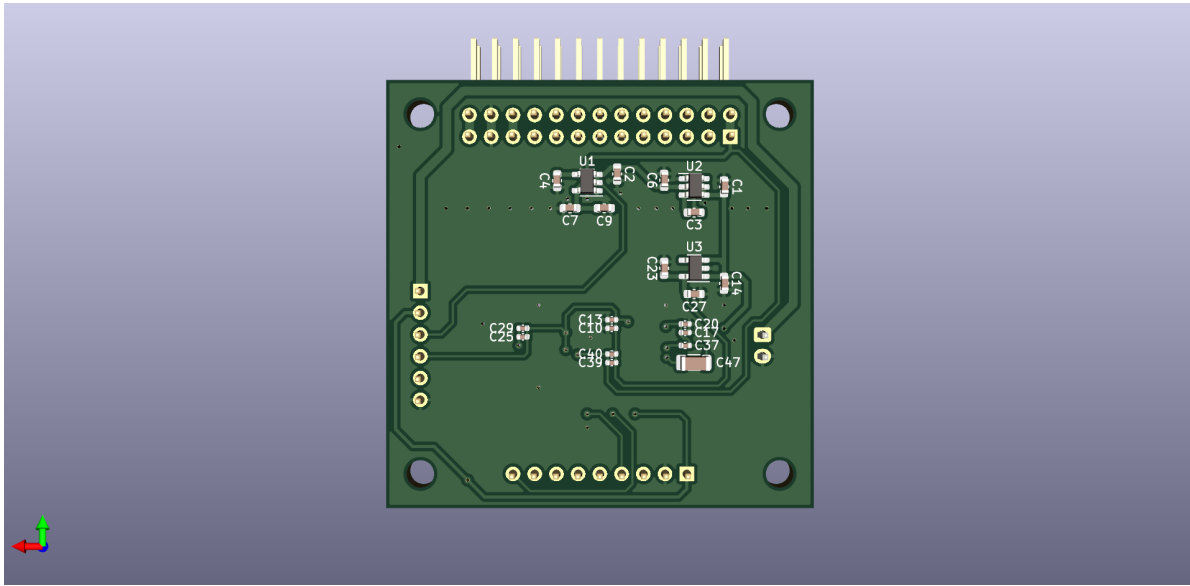


Figura 4.4: Vista 3D trasera del módulo de AFE ADS1299.

Con relación al desarrollo del *hardware* que condujo al diseño final de los PCB presentados anteriormente, se expone la guía de diseño que se construyó y siguió durante todo el proceso:

1. Entendimiento del problema

- Requisitos del proyecto: Reunir toda la información necesaria sobre los requisitos del proyecto. Esto incluye las especificaciones del sistema, limitaciones de tamaño, presupuesto, y plazos.
- Investigación preliminar: Comprender la aplicación del PCB, el entorno en el que operará y las normativas aplicables.

2. Definición de especificaciones

- Especificaciones del sistema: Definir las especificaciones técnicas como voltajes de operación, corrientes, frecuencias de las señales, interfaces de comunicación, requisitos de potencia, etc.
- Dimensiones y forma del PCB: Determinar el tamaño y la forma del PCB según

el dispositivo en el que se integrará. Este paso es importante dada la limitante principal de portabilidad e inmunidad a ruido eléctrico.

3. Búsqueda de CI (Circuitos Integrados) y componentes

- Selección de CI: Buscar componentes que cumplan con las especificaciones del sistema. Considerar disponibilidad, costo y soporte técnico.
- Componentes pasivos y conectores: Seleccionar resistencias, condensadores, inductores, conectores y otros componentes necesarios.
- Bibliotecas de componentes: Asegurarse de tener las librerías adecuadas para los componentes seleccionados en el software de diseño de PCB.

4. Diseño del esquema

- Diagramas de bloques: Crear un diagrama de bloques del sistema.
- Esquemático: Dibujar el esquemático detallado utilizando un software de diseño (en este caso fue KiCad EDA 7.0.2).
- Revisión del esquemático: Revisar el esquemático para verificar la coherencia y correcta integración de los diseños recomendados y aplicaciones típicas de los circuitos integrados seleccionados.

5. Diseño del PCB

- Asignación de componentes: Colocar los componentes en el PCB considerando las restricciones de espacio y las rutas de señal.
- Enrutamiento de señales: Enrutar las señales siguiendo las mejores prácticas de diseño para minimizar el ruido y las interferencias.
- Capas del PCB: Definir el número de capas necesarias para el PCB (y en el proyecto se optó por un diseño multicapa con un apilado de 4 capas).

- Planos de tierra y potencia: Crear planos de tierra y potencia para asegurar una distribución adecuada de energía y minimizar problemas de ruido.

6. Simulación y verificación

- Simulación del circuito (SPICE): Utilizar herramientas de simulación para verificar el comportamiento de la sección analógica del circuito.
- Análisis DFM (Diseño para Manufacturabilidad): Asegurarse de que el diseño del PCB sea manufacturable, considerando las capacidades del fabricante.
- Revisión por pares: Realizar una revisión por pares del diseño para identificar posibles errores o mejoras.

7. Generación de archivos de fabricación

- Gerbers y Excellon: Generar los archivos Gerber y Excellon necesarios para la fabricación del PCB.
- BOM (Bill of Materials): Crear una lista de materiales detallada para la adquisición de componentes.
- Documentación: Preparar toda la documentación necesaria, incluyendo diagramas, especificaciones y guías de ensamblaje.

8. Fabricación

- Selección de fabricante: Elegir un fabricante de PCB adecuado basado en capacidades, costos y tiempos de entrega.
- Envío de archivos: Enviar los archivos de fabricación al fabricante.

9. Montaje y soldadura

- Adquisición de componentes: Comprar todos los componentes necesarios.

- Montaje del PCB: Ensamblar los componentes en el PCB (la opción manual a través de herramientas como un cautín de punta fina y cama caliente para soldar fue la solución seleccionada para el proyecto).

10. Pruebas y verificación

- Pruebas funcionales: Realizar pruebas funcionales del PCB para asegurarse de que cumple con las especificaciones.
- Corrección de errores: Identificar y corregir cualquier error o defecto en el diseño.
- Pruebas de validación: Realizar pruebas adicionales bajo diferentes condiciones para validar el diseño.

El diseño de PCB de señal mixta requiere adherirse a las normas fundamentales de diseño tanto digital como analógico, y es crucial evitar la interferencia de señales. Establecer estas directrices puede ser complicado sin las herramientas adecuadas para el diseño de PCB. Además, la ubicación de los componentes es tan relevante como el diseño de las rutas y la estructura de las capas. Un diseño eficiente de PCB de señal mixta combina las funcionalidades de circuitos digitales y analógicos. Utilizando KiCad EDA, un software avanzado para el diseño de PCB, es posible desarrollar productos electrónicos de señal mixta desde su conceptualización hasta su producción final en una única plataforma. Este programa ofrece todas las herramientas necesarias para la creación de esquemas, el diseño de PCB y la fabricación, entre otros. Consecuentemente, la necesidad de una guía de diseño es indispensable para garantizar la integridad de señales y el funcionamiento general del sistema, y para ello los artículos de Analog Devices correspondientes a buenas prácticas de diseño para PCB de señales mixtas (Chesser & Porley, s.f.) y diseño de PCB para señales de alta velocidad (Ardizzoni, 2005). Seguidamente se mencionan algunos de los puntos que fueron aplicados en el proyecto con el objetivo de salvaguardar la integridad de las señales provenientes de los electrodos:

- Mantenimiento de la aislación entre las secciones digitales y analógicas de una PCB: Para minimizar la interferencia y el ruido entre las secciones digitales y analógicas de

una PCB (Placa de Circuito Impreso), es fundamental mantener una adecuada aislación entre ellas. Las señales digitales, que suelen ser de alta velocidad y tener transiciones rápidas, pueden inducir ruido en las señales analógicas más sensibles. Por tanto, la disposición física de los componentes debe planificarse cuidadosamente para garantizar que las rutas de señal digital y analógica no se crucen ni se solapen, evitando así la contaminación de ruido.

- Separación de los componentes digitales y analógicos en sus propios bloques de circuito: Una técnica efectiva para gestionar la integridad de las señales es agrupar los componentes digitales y analógicos en bloques de circuito separados. Esto implica diseñar la PCB de manera que las secciones digitales y analógicas ocupen regiones distintas y definidas. Al mantener estos bloques separados, se reduce el riesgo de que las señales digitales perturben las señales analógicas. Además, facilita el diseño y la implementación de rutas de señal dedicadas y optimizadas para cada tipo de componente.
- Creación de canales de enrutamiento dedicados entre las secciones analógicas y digitales: Los canales de enrutamiento dedicados son esenciales para gestionar el flujo de señales entre las secciones analógicas y digitales. Estos canales deben diseñarse de manera que minimicen el acoplamiento de ruido y preserven la integridad de las señales. El uso de planos de tierra y de alimentación adecuados, junto con el enrutamiento estratégico de las pistas, puede ayudar a aislar las señales críticas y reducir la interferencia mutua.
- Prevención del crosstalk y la interferencia entre estas secciones: El crosstalk (diafonía) ocurre cuando las señales de una pista de la PCB se acoplan con otra pista adyacente, causando interferencia. Para prevenir este fenómeno, es crucial mantener una distancia adecuada entre las pistas digitales y analógicas y utilizar técnicas como el apantallamiento y el uso de pistas diferenciales. Además, se deben considerar los efectos de los planos de tierra y de alimentación, asegurándose de que estén bien diseñados para minimizar las corrientes de retorno y las interferencias.

- Establecimiento de un sistema de tierra consistente a lo largo de la placa: Un sistema de tierra consistente es fundamental para el correcto funcionamiento de una PCB que contiene tanto secciones digitales como analógicas. Esto implica la creación de planos de tierra dedicados que proporcionen un camino de retorno estable para las señales y minimicen las diferencias de potencial entre diferentes puntos de la placa. Un plano de tierra bien diseñado ayuda a reducir el ruido y a mejorar la inmunidad de las señales frente a interferencias externas.
- Asegurar el control de impedancia en las interconexiones críticas analógicas y digitales: El control de impedancia es crucial para mantener la integridad de las señales, especialmente en interconexiones críticas tanto analógicas como digitales. Esto implica diseñar las pistas de la PCB con un ancho y una separación adecuados, y considerar el uso de materiales dieléctricos específicos que mantengan la impedancia característica deseada. El control de impedancia ayuda a prevenir reflexiones de señal y a mantener la calidad de las transmisiones de datos, lo cual es esencial para el rendimiento general del sistema.

4.1.4. Implementación del Firmware

La Figura 4.5 propone un diagrama de bloques que indica un *stack* de firmware que abarca todos los requisitos para un sistema embebido orientado al procesamiento de señales y conexión IoT. El siguiente contenido describe la funcionalidad esperada en el firmware (correlacionándolos con los bloques) para la realización del procesamiento y clasificación de señales en tiempo real:

Aplicación de Usuario Embebida

Aquí yacen la mayoría de las funcionalidades del sistema y cómo se comunican entre sí para lograr la concepción fundamental del BCI a diseñar: adquisición y filtrado de señales EEG para su posterior compresión y envío por medios inalámbricos a una computadora o

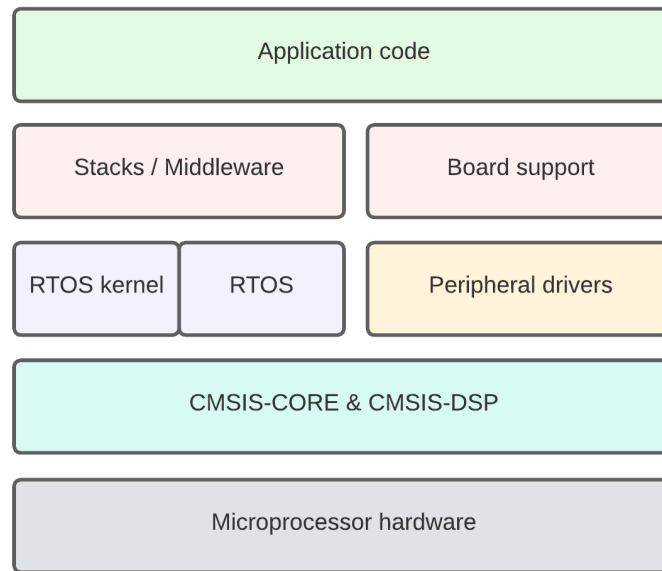


Figura 4.5: Vista 3D de la parte posterior del PCB.

dispositivo móvil encargado de entrenar la red neuronal que se cargará como solución al problema de clasificación de señales integrado; o solo un registro local de datos para guardar directamente en memoria flash (tarjeta SD).

TinyML

La extracción de características es un paso requerido en el proceso de clasificación de señales de electroencefalograma (EEG) (Amin et al., 2017). Por lo tanto, si se busca desarrollar una RNA para realizar esta actividad, es imperativo que el modelo deba ser comprimido para caber en la memoria del microprocesador y trabajar bajo las restricciones de velocidad de reloj (traducido en operaciones por segundo) y energía. TinyML es, de hecho, una disciplina en desarrollo de tecnología embebida que intenta aplicar técnicas de aprendizaje automático dentro de microprocesadores con recursos restringidos (Han & Siebert, 2022). El marco TensorFlow Lite es una plataforma gratuita y públicamente accesible para el aprendizaje profundo que permite la inferencia de aprendizaje automático en el dispositivo, enfocándose

en el aprendizaje consciente del borde. Este marco puede ayudar a abordar la necesidad de que el aprendizaje automático se realice directamente en el dispositivo, en lugar de depender únicamente de recursos basados en la nube (Ray, 2022). Un diseño de tubería de datos para TinyML relacionado con el enfoque técnico se muestra en la Figura 4.6.

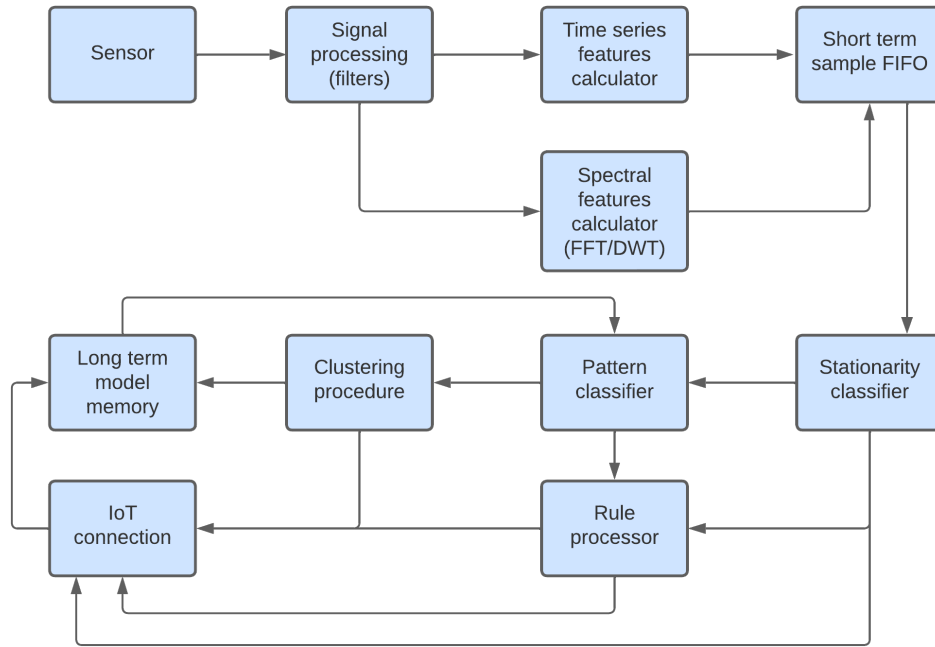


Figura 4.6: Tubería de borde para TinyML para datos de series temporales (Ray, 2022).

Middleware

Dada la naturaleza de las señales EEG y los ensayos clínicos en el campo de las neurociencias, es imperativo contar con un sistema de adquisición y procesamiento capaz de ejecutar operaciones y procesos en tiempo real con el objetivo de acelerar las instrucciones que el procesador tiene que llevar a cabo. Es esencial garantizar que los datos obtenidos se organicen de manera paralela, relacionándolos uno a uno independientemente de su origen; que los datos estén directamente relacionados entre sí a través de series temporales correctamente sincronizadas. Esto representa el éxito o fracaso del experimento, ya que sensores adicionales, como

el acelerómetro, sirven para obtener información relevante que ayuda a discriminar acciones mecánicas del sujeto de prueba (es decir, movimiento de los ojos o inclinación de la cabeza) de la actividad cerebral que está relacionada solo con tareas cognitivas (Esqueda-Elizondo et al., 2022). Gracias a esto, el uso de un sistema operativo en tiempo real (por ejemplo, FreeRTOS (Services, 2023)) para la paralelización de tareas de adquisición, procesamiento, compresión y registro de datos se vuelve muy importante para el desarrollo.

Drivers de bajo nivel

Es crucial confiar en las bibliotecas oficiales del fabricante del chip (en este caso: Texas Instruments) para acelerar la integración del AFE con el microprocesador a través de la comunicación de Interfaz Serie a Paralelo (SPI), ya que estos paquetes de firmware ya tienen las capas de abstracción de hardware necesarias y mapas de registro extensos para controlar completamente el circuito integrado, hablando en términos de configuración de tasas de muestreo, filtros internos, Amplificadores de Ganancia Programables (PGAs), etc. Asimismo, es necesario involucrar la comunicación serial mediante comandos AT hacia el módulo Bluetooth, dado su rol decisivo en la transmisión de datos inalámbrica. Finalmente, otros periféricos o sensores existentes dentro del hardware, como acelerómetros, inclinómetros, micrófonos, memorias de tarjeta SD, etc., deben ser incluidos dentro de los límites de esta sección.

4.2. Diseño de la red neuronal

Crear un programa de aprendizaje automático se diferencia del proceso habitual de programación. En un software tradicional, un programador diseña un algoritmo que recibe una entrada, aplica diversas reglas y entrega una salida. Las operaciones internas del algoritmo son definidas por el programador y se establecen de forma explícita mediante líneas de código. Para prever fallos en una máquina de fábrica, el programador tendría que reconocer qué indicadores en los datos señalan un problema y desarrollar un código que específicamente los

detecte.

Para crear un programa de aprendizaje automático, un programador introduce datos en un tipo especial de algoritmo y permite que éste descubra las reglas. Esto implica que los programadores pueden desarrollar códigos que realicen predicciones basadas en datos complejos sin necesidad de entender a profundidad el problema a resolver. El algoritmo de aprendizaje automático construye un modelo del sistema con base en los datos que se proporcionan, mediante un proceso denominado entrenamiento. El modelo es una especie de programa informático. Se utiliza este modelo para realizar predicciones, en un proceso llamado inferencia.

Existen muchos enfoques distintos para el aprendizaje automático. Uno de los más populares es el aprendizaje profundo, que se basa en una idea simplificada de cómo podría funcionar el cerebro humano. En el aprendizaje profundo, una red de neuronas simuladas (representadas por matrices de números) se entrena para modelar las relaciones entre distintas entradas y salidas. Diferentes arquitecturas, o disposiciones de neuronas simuladas, son útiles para diferentes tareas. Por ejemplo, algunas arquitecturas son excelentes para extraer significado de datos de imágenes, mientras que otras son ideales para predecir el siguiente valor en una secuencia.

Flujo de trabajo para la programación e implementación de la red neuronal embebida

1. Definición del problema a resolver por inferencia: Al diseñar un algoritmo, es esencial determinar con precisión su objetivo principal. Esto es aplicable también al aprendizaje automático. Se debe establecer qué se desea predecir para así definir qué datos son necesarios recolectar y qué arquitectura de modelo es más adecuada para utilizar.
2. Recopilación del conjunto de datos:
 - a) Selección de datos: Los modelos de aprendizaje profundo pueden aprender a ignorar datos ruidosos o irrelevantes. Sin embargo, es recomendable entrenar el modelo utilizando solo información relevante para resolver el problema. Es aconsejable

- combinar la experiencia en el dominio con la experimentación al decidir si incluir datos. También es posible emplear técnicas estadísticas para intentar identificar qué datos son significativos.
- b)* Recopilación de datos: Determinar la cantidad exacta de datos necesarios para entrenar un modelo efectivo es complejo. Esto depende de varios factores, como la complejidad de las relaciones entre las variables, la cantidad de ruido en los datos y la facilidad con la que se pueden distinguir las clases. Es recomendable recopilar datos que representen el espectro completo de condiciones y eventos que puedan surgir en el sistema.
 - c)* Etiquetado de datos: Es el proceso mediante el cual se asignan categorías o “etiquetas” específicas a conjuntos de datos para indicar ciertas características o atributos. Estas etiquetas sirven como respuestas o soluciones para que el algoritmo de aprendizaje pueda identificar patrones y aprender de ellos.
3. Diseño de la arquitectura de la red neuronal: Existen numerosas arquitecturas de modelos de aprendizaje profundo, diseñadas para resolver una amplia variedad de problemas. Al entrenar un modelo, se puede optar por diseñar una arquitectura propia o basarse en una arquitectura existente desarrollada por investigadores. Al decidir sobre una arquitectura, es necesario considerar el tipo de problema que se intenta resolver, el tipo de datos al que se tiene acceso y las formas en que se puede transformar esos datos antes de introducirlos en un modelo (la transformación de datos se discutirá más adelante). La realidad es que, dado que la arquitectura más efectiva varía según el tipo de datos con los que se trabaja, los datos y la arquitectura del modelo están estrechamente vinculados. También es esencial considerar las limitaciones del dispositivo en el que se ejecutará el modelo. Por lo general, los microcontroladores tienen memoria limitada y procesadores lentos, y los modelos más grandes requieren más memoria y tardan más en ejecutarse. El tamaño de un modelo depende del número de neuronas que contiene y de cómo están conectadas estas neuronas. Además, algunos dispositivos cuentan con

aceleración de hardware que puede agilizar la ejecución de ciertos tipos de arquitecturas de modelos. Por lo tanto, podría ser conveniente adaptar el modelo a las características del dispositivo que se tiene en consideración.

- a)* Extracción de características: Implica convertir datos en bruto de series temporales en un conjunto de características (a menudo llamadas “variables” o “atributos”) que representen la información subyacente de los datos de manera más compacta y útil para tareas específicas como el análisis predictivo, la clasificación o el clustering. La idea es capturar patrones, tendencias y propiedades inherentes en la serie temporal para usarlos en modelos analíticos. Al transformar una serie temporal al dominio de la frecuencia (por ejemplo, usando la Transformada Rápida de Fourier), se pueden identificar componentes periódicos dominantes o frecuencias fundamentales en los datos.
 - b)* Ventaneo: Esta técnica implica dividir la señal o serie temporal en segmentos o ventanas más pequeños y consecutivos de datos. Cada ventana puede superponerse o no con la siguiente. Hay diversas razones para emplear ventaneo, desde reducir la complejidad de los datos hasta preparar los datos para una transformada o extracción de características. Se extraen características de cada ventana para representar la información en ese segmento.
 - c)* Normalización: Es un paso crucial para asegurarse de que las características derivadas de los datos sean comparables y que contribuyan de manera equitativa al rendimiento de un modelo o algoritmo. Si las características extraídas tienen diferentes escalas o magnitudes, pueden influir de manera desproporcionada en la función de un modelo, lo que podría llevar a interpretaciones erróneas o a un rendimiento subóptimo.
4. Entrenamiento del modelo: Un modelo es una red de neuronas simuladas representadas por matrices de números organizadas en capas. Estos números se conocen como pesos y sesgos, o colectivamente como los parámetros de la red. Cuando se introduce data en la

red, esta se transforma mediante operaciones matemáticas sucesivas que involucran los pesos y sesgos de cada capa. La salida del modelo es el resultado de procesar la entrada a través de estas operaciones. El modelo inicia con pesos asignados aleatoriamente y los sesgos típicamente comienzan con un valor de 0. Durante el entrenamiento, se introducen lotes de datos en el modelo, y la salida del modelo se compara con la salida deseada (que en nuestro caso es la etiqueta correcta, “normal” o “anormal”). Un algoritmo llamado retropropagación ajusta los pesos y sesgos de manera incremental, de modo que, con el tiempo, la salida del modelo se acerque más al valor deseado. El entrenamiento, que se mide en épocas (es decir, iteraciones), continúa hasta que se decide detenerlo. Generalmente se detiene el entrenamiento cuando el rendimiento de un modelo deja de mejorar. Cuando comienza a realizar predicciones precisas, se dice que ha convergido. Para determinar si un modelo ha convergido, se pueden analizar gráficos de su rendimiento durante el entrenamiento. Dos métricas de rendimiento comunes son la pérdida y la precisión. La métrica de pérdida nos ofrece una estimación numérica de cuán lejos está el modelo de producir las respuestas esperadas, y la métrica de precisión nos indica el porcentaje de veces que elige la predicción correcta. Un modelo perfecto tendría una pérdida de 0.0 y una precisión del 100 %, pero los modelos reales rara vez son perfectos. Para intentar mejorar el rendimiento del modelo, es posible modificar la arquitectura del modelo y ajustar diversos valores utilizados para configurar el modelo y moderar el proceso de entrenamiento. Estos valores se denominan colectivamente como hiperparámetros e incluyen variables como el número de épocas de entrenamiento a realizar y el número de neuronas en cada capa. Cada vez que se realiza un cambio, se puede volver a entrenar el modelo, observar las métricas y decidir si se optimiza aún más.

- a) Sobreajuste y subajuste: Las dos razones más comunes por las que un modelo no logra converger son el subajuste y el sobreajuste. Un red neuronal aprende a ajustar su comportamiento a los patrones que reconoce en los datos. Si un modelo está correctamente ajustado, producirá la salida correcta para un conjunto dado

de entradas. Cuando un modelo está subajustado, aún no ha logrado aprender una representación lo suficientemente fuerte de estos patrones para poder hacer buenas predicciones. Esto puede ocurrir por varias razones, siendo las más comunes que la arquitectura es demasiado pequeña para capturar la complejidad del sistema que se supone debe modelar o que no ha sido entrenado con suficientes datos. Cuando un modelo está sobreajustado, ha aprendido sus datos de entrenamiento demasiado bien. El modelo es capaz de predecir exactamente los detalles menudos de sus datos de entrenamiento, pero no puede generalizar su aprendizaje a datos que no ha visto previamente. Esto suele suceder porque el modelo ha logrado memorizar completamente los datos de entrenamiento, o ha aprendido a depender de un atajo presente en los datos de entrenamiento pero no en el mundo real. Existen varias estrategias para combatir el sobreajuste. Una de ellas es reducir el tamaño del modelo para que no tenga la capacidad suficiente para aprender una representación exacta de su conjunto de entrenamiento. Un conjunto de técnicas conocidas como regularización se pueden aplicar durante el entrenamiento para reducir el grado de sobreajuste. Para aprovechar al máximo datos limitados, se puede utilizar una técnica llamada aumentación de datos, la cual genera nuevos puntos de datos artificiales cortando y mezclando los datos existentes. Pero la mejor manera de vencer el sobreajuste, cuando es posible, es obtener un conjunto de datos más grande y variado.

- b) Entrenamiento, validación y testeo: Para la evaluación del rendimiento de un modelo, es esencial considerar su desempeño con los datos de entrenamiento. No obstante, esta métrica proporciona una visión parcial. Durante el entrenamiento, los modelos se optimizan para adecuarse lo más posible a sus datos de entrenamiento, lo cual puede llevar a un fenómeno conocido como sobreajuste. Éste se caracteriza porque el modelo muestra un alto rendimiento con los datos de entrenamiento, pero una eficacia reducida en aplicaciones prácticas. Para detectar y comprender el fenómeno, resulta crucial validar el modelo con datos que no se hayan utilizado

durante el proceso de entrenamiento. En la práctica, se acostumbra dividir el conjunto de datos en tres segmentos: entrenamiento, validación y prueba. Un esquema de división común asigna un 60 % de los datos para entrenamiento, y un 20 % para cada uno de los conjuntos de validación y prueba. Esta división debe realizarse asegurando que cada segmento contenga una representación equitativa y variada del conjunto de datos total, para garantizar una evaluación precisa del modelo. La división de los datos en los conjuntos de entrenamiento, validación y prueba debe realizarse de manera que cada parte contenga la misma distribución de información, preservando la estructura de los datos. Por ejemplo, si los datos constituyen una serie temporal, podrían dividirse potencialmente en tres segmentos contiguos en el tiempo. En cambio, si los datos no forman una serie temporal, se podrían muestrear aleatoriamente los puntos de datos. Durante el entrenamiento, se utiliza el conjunto de entrenamiento para entrenar el modelo. Periódicamente, se introduce en el modelo datos del conjunto de validación y se calcula la pérdida. Dado que el modelo no ha visto estos datos anteriormente, su puntuación de pérdida es una medida más fiable de su rendimiento. Comparando las pérdidas de entrenamiento y validación (y la precisión u otras métricas disponibles) a lo largo del tiempo, es posible determinar si el modelo está sobreajustándose. Al llevar a cabo este proceso, los modelos y procesos de entrenamiento para mejorar el rendimiento y evitar el sobreajuste, idealmente se observará una mejora en las métricas de validación. Sin embargo, este proceso puede tener un efecto secundario desafortunado. Al optimizar para mejorar las métricas de validación, se podría estar induciendo al modelo a sobreajustarse tanto a los datos de entrenamiento como a los de validación. Cada ajuste realizado podría adaptar el modelo ligeramente mejor a los datos de validación, resultando finalmente en el mismo problema de sobreajuste que inicialmente. Para verificar que esto no haya ocurrido, el paso final en el entrenamiento de un modelo es ejecutarlo con los datos de prueba y confirmar que su rendimiento sea tan bueno como en la validación. Si no es así, significa que el

modelo se ha optimizado para sobreajustarse tanto a los datos de entrenamiento como a los de validación. En este caso, podría ser necesario volver al principio y plantear una nueva arquitectura de modelo, ya que si se continúa ajustando para mejorar el rendimiento en los datos de prueba, se terminará sobreajustando también a estos. Una vez que se tiene un modelo que funciona aceptablemente bien con los datos de entrenamiento, validación y prueba, la parte de entrenamiento de este proceso concluye.

5. Conversión del modelo a biblioteca optimizada para sistemas embebidos: Un modelo de TensorFlow es esencialmente un conjunto de instrucciones que indican a un intérprete cómo transformar los datos para producir una salida. Cuando se desea utilizar el modelo, simplemente se carga en la memoria y se ejecuta utilizando el intérprete de TensorFlow. Sin embargo, el intérprete de TensorFlow está diseñado para ejecutar modelos en potentes computadoras de escritorio y servidores. Dado que los modelos se ejecutarán en microcontroladores de pequeño tamaño y baja potencia, es necesario un intérprete diferente, diseñado específicamente para este propósito. Afortunadamente, TensorFlow ofrece un intérprete y herramientas complementarias para ejecutar modelos en dispositivos pequeños y de baja potencia. Este conjunto de herramientas se conoce como TensorFlow Lite. Antes de que TensorFlow Lite pueda ejecutar un modelo, primero debe convertirse al formato de TensorFlow Lite y luego guardarse en el disco como un archivo. Este proceso se lleva a cabo utilizando una herramienta llamada el Convertidor de TensorFlow Lite. El convertidor también puede aplicar optimizaciones especiales destinadas a reducir el tamaño del modelo y ayudar a que se ejecute más rápido, a menudo sin sacrificar el rendimiento.
6. Ejecución de la inferencia: Una vez que el modelo ha sido convertido, está listo para su despliegue. Ahora se utilizará la biblioteca C++ de TensorFlow Lite para Microcontroladores para cargar el modelo y realizar predicciones. Dado que esta es la etapa en la que el modelo se integra con el código de la aplicación, es necesario escribir código

que tome los datos brutos de los sensores y los transforme al mismo formato en el que fue entrenado el modelo. Posteriormente, estos datos transformados se introducen en el modelo para ejecutar la inferencia. Este proceso implica adaptar los datos sensoriales de entrada para que coincidan con las especificaciones de entrada del modelo, incluyendo escalado, normalización o cualquier otra transformación necesaria. Una vez que los datos están en el formato correcto, se pueden pasar al modelo para realizar la inferencia. Esta etapa es crucial, ya que la precisión y eficacia de las predicciones del modelo dependen en gran medida de la calidad y la precisión del preprocesamiento de los datos.

7. Métricas de evaluación y solución de problemas: Una vez que el modelo ha sido desplegado y está en funcionamiento en el dispositivo, se podrá comenzar a evaluar si su rendimiento en el mundo real se acerca a lo esperado. A pesar de que ya se ha demostrado que el modelo realiza predicciones precisas con sus datos de prueba, el rendimiento en el problema real podría ser diferente. Existen varias razones por las que esto podría suceder. Por ejemplo, los datos utilizados en el entrenamiento podrían no ser completamente representativos de los datos disponibles en operación real. Tal vez, debido al clima local, la temperatura de la máquina en la que se ejecuta el modelo es generalmente más baja que la de la máquina de donde se recopiló el conjunto de datos. Esto podría afectar las predicciones realizadas por el modelo, de manera que ya no sean tan precisas como se esperaba. Esta discrepancia resalta la importancia de la validación continua y la adaptación del modelo a las condiciones cambiantes o a variaciones en los datos de entrada. Es crucial realizar un seguimiento del rendimiento del modelo en el entorno real y estar preparado para ajustarlo según sea necesario, lo cual puede incluir la re-entrenamiento con datos más representativos o la modificación de ciertos parámetros para adaptar mejor el modelo a las condiciones específicas de su aplicación.

4.3. Implementación de la red neuronal en MCU

4.3.1. Edge Impulse

Edge Impulse es una plataforma que facilita la creación de aplicaciones de aprendizaje automático para dispositivos de borde (edge devices) como microcontroladores y sistemas integrados.

Aquí está el texto redactado con un tono científico y neutral, evitando referencias personales:

1. Creación de una cuenta en Edge Impulse: Para el inicio del proceso, es esencial disponer de una cuenta en Edge Impulse. En caso de no existir previamente, dicha cuenta puede ser creada en el sitio web oficial de Edge Impulse.
2. Selección de dispositivo: Tras el acceso a la plataforma, se selecciona un dispositivo compatible. Edge Impulse brinda soporte a una variedad de dispositivos, y la información detallada sobre compatibilidad se halla disponible en el sitio web oficial. La conexión del dispositivo y su configuración con Edge Impulse es un paso subsecuente.
3. Recopilación de datos:
 - Captura de datos: El dispositivo permite la adquisición de datos, que pueden incluir registros de sensores, imágenes, sonidos, entre otros, dependiendo de la naturaleza del proyecto.
 - Carga de datos: En casos donde exista un conjunto de datos previamente recolectado, es posible cargar dicho conjunto en la plataforma de Edge Impulse.
4. Diseño y entrenamiento de la red neuronal:
 - Procesamiento de señales: Es crucial definir el tratamiento de señales o imágenes con el objetivo de extraer características relevantes. Edge Impulse dispone de bloques predefinidos, como el FFT para datos temporales o bloques especializados para el tratamiento de imágenes.

- Diseño de la red neuronal: La definición de la arquitectura de la red neuronal es un paso indispensable. Existen redes predefinidas que pueden ser empleadas o, alternativamente, se puede diseñar una red específica.
 - Entrenamiento: Con la red definida, se procede al entrenamiento utilizando el conjunto de datos disponible.
5. Evaluación del modelo: Posterior al entrenamiento, se evalúa el rendimiento del modelo mediante un conjunto de test. Si los resultados obtenidos no cumplen con los criterios establecidos, es posible retornar a fases previas como diseño o recopilación de datos para realizar ajustes.
6. Implementación:
- Generación del código: Edge Impulse permite la generación de un modelo en formato de código C++ listo para ser desplegado en microcontroladores.
 - Descarga del código: Se realiza la descarga del código generado junto con las bibliotecas requeridas.
7. Integración en el microcontrolador:
- Integración de bibliotecas: Las bibliotecas descargadas se incorporan al entorno de desarrollo del microcontrolador.
 - Integración del modelo: Se procede a la incorporación del código C++ del modelo en la aplicación correspondiente.
 - Compilación y despliegue: Se compila la aplicación y se transfiere al microcontrolador.
8. Verificación en tiempo real: Con el modelo desplegado en el microcontrolador, se alimenta con datos en tiempo real y se monitoriza su comportamiento.

4.4. Colocación de electrodos EEG

4.4.1. Funciones de las áreas del cerebro

El ámbito de la neurotecnología ha experimentado un notable crecimiento debido a la introducción de numerosas herramientas innovadoras, lo que ha incrementado la complejidad en la selección de sistemas de electroencefalograma (EEG) adecuados para aplicaciones específicas. Una de las consideraciones primordiales en esta selección implica determinar si el EEG posee un diseño de electrodos fijo o variable. Los diseños variables ofrecen una mayor flexibilidad en la colocación de electrodos, permitiendo adaptaciones específicas a diferentes estudios y sujetos, aunque esto puede complicar su configuración y uso. Por contraste, los sistemas con diseños fijos facilitan la usabilidad y la estandarización entre diferentes sesiones e individuos, pero con menor adaptabilidad a necesidades particulares (Bitbrain, 2024).

En el contexto de la anatomía cerebral, la corteza cerebral, que constituye la capa externa del cerebro, desempeña funciones cruciales para el sistema nervioso. Esta región está organizada en cuatro lóbulos principales: frontal, parietal, temporal y occipital, cada uno asociado con distintas funciones cerebrales. El lóbulo frontal es crucial para el razonamiento, la regulación del habla y del movimiento, las emociones y la resolución de problemas. El área central, principalmente sensoriomotora, interactúa con la percepción y respuesta a estímulos físicos. El lóbulo parietal se involucra en la atención y el procesamiento sensorial, incluyendo la percepción de temperatura, tacto, presión y dolor. El lóbulo occipital se especializa en el procesamiento visual, mientras que el lóbulo temporal juega un papel vital en la memoria, así como en la interpretación y el procesamiento de los estímulos auditivos.

Estos conocimientos sobre la organización funcional del cerebro son fundamentales para la correcta aplicación de tecnologías como los EEG, donde la colocación de electrodos puede optimizarse basándose en los lóbulos y funciones específicas a investigar.

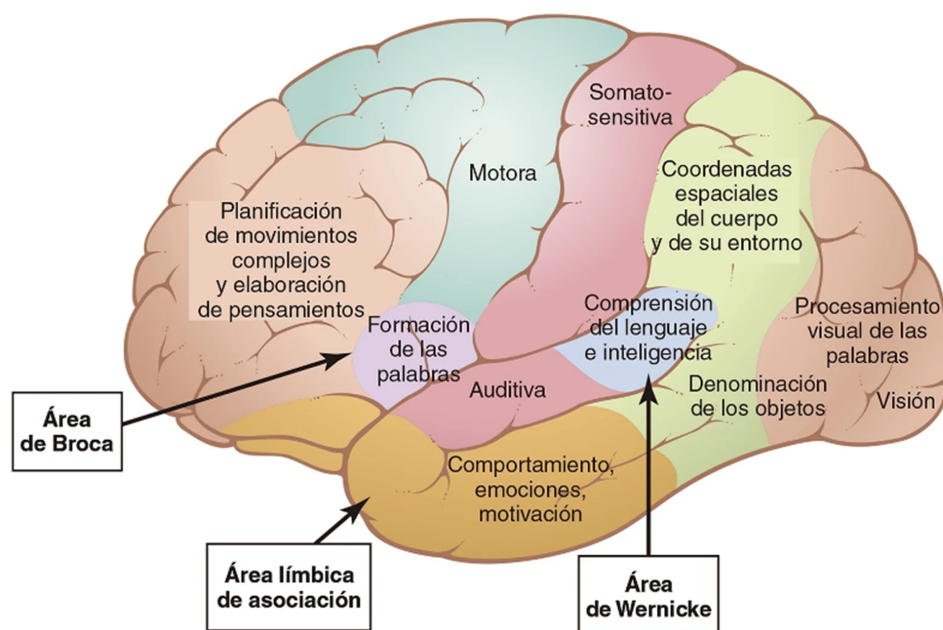


Figura 4.7: Áreas funcionales de la corteza cerebral humana (Connect, 2021).

4.4.2. Colocación de electrodos EEG

Los procesos bioquímicos que ocurren entre células neuronales generan una actividad eléctrica mínima como resultado de la comunicación neuronal. Individualmente, estas señales eléctricas son imperceptibles, pero al sincronizarse millones de neuronas, el campo eléctrico resultante es suficientemente robusto para ser detectado mediante electrodos posicionados en el cuero cabelludo. Estas señales electroencefalográficas, conocidas como EEG, atraviesan tejidos, huesos y cabello, perdiendo significativamente su amplitud antes de ser finalmente registradas (Adeli et al., 2007).

La precisión en la ubicación de los electrodos en el cuero cabelludo es crucial para asegurar la reproducibilidad de los experimentos y la comparabilidad de las grabaciones obtenidas por distintos investigadores. Con el fin de estandarizar la colocación de electrodos y la metodología de registro de EEG, en 1947 se estableció un comité encargado de desarrollar un sistema uniforme para estos procesos. Este comité formuló el sistema 10-20 de EEG, que dicta cómo posicionar y etiquetar los canales de EEG, recomendando un mínimo de 21 electrodos para

la exploración adecuada del cerebro adulto (Zeidan et al., 2022).

El sistema internacional para la colocación de electrodos de electroencefalograma (EEG) se basa en cuatro puntos de referencia craneales estándar: el nasion, el inion y ambos puntos preauriculares. Los electrodos se distribuyen de manera proporcional sobre la superficie de la cabeza siguiendo este sistema. Esta metodología asigna a cada electrodo una etiqueta que corresponde a la región cortical específica que se está monitorizando.

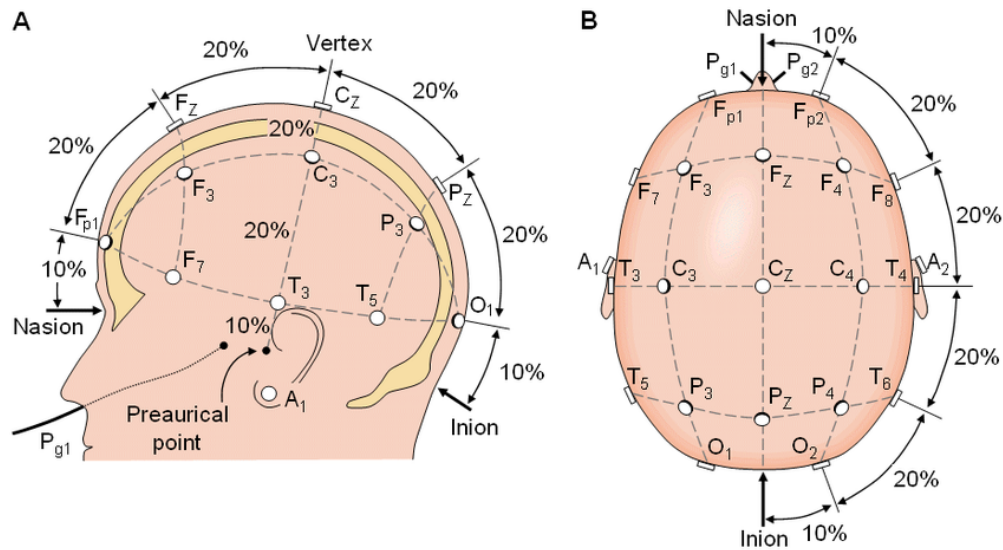


Figura 4.8: Sistema Internacional 10-20 para la colocación de los electrodos extracraneales (Novo-Olivas et al., 2010).

Las principales áreas cerebrales registradas incluyen la región frontopolar (Fp), la frontal (F), la central (C), la temporal (T), la parietal (P) y la occipital (O). Además, la nomenclatura del sistema incorpora una designación numérica para indicar la lateralidad de los electrodos: los números impares (1, 3, 5, 7) identifican los electrodos ubicados en el hemisferio izquierdo, mientras que los números pares (2, 4, 6, 8) señalan aquellos en el hemisferio derecho. Por otro lado, los electrodos situados directamente sobre la línea media del cerebro se marcan con la letra “z”, indicando su posición central.

El diseño de la colocación de electrodos en electroencefalograma (EEG) se basa en porcentajes específicos de la distancia entre sensores, siguiendo las líneas de referencia central sagital

y coronal del cráneo. El sistema 10-20, que comprende un total de 21 sensores, utiliza una distribución basada en distancias que representan el 10 % y el 20 % de estas líneas de referencia. Para una mayor densidad de muestreo, el sistema 10-10 se implementa con 81 sensores, subdividiendo estas mismas líneas en segmentos del 10 %. En una escala aún más detallada, el sistema 10-5 incluye 320 electrodos, estableciendo las divisiones a intervalos del 5 % (Tawhid et al., 2020).

Este estándar en la colocación de electrodos permite una correlación directa entre la actividad registrada en un sensor específico y las funciones cerebrales asociadas, dependiendo de su posición dentro del diseño del sensor. Por ejemplo, el sistema de electrodos 10-20 facilita la identificación y descripción de áreas cerebrales específicas y sus funciones correspondientes en estudios de EEG. Esta estandarización no solo asegura que los resultados puedan ser comparables con otros estudios en la literatura científica, sino que también simplifica la descripción de las áreas y funciones cerebrales que cada sensor puede registrar.

Capítulo 5

Experimentación y Resultados

5.1. Medición y análisis de señales a través del BCI diseñado

5.1.1. Diseño experimental

El diseño experimental aquí presentado tiene como objetivo describir el método para la medición de señales electroencefalográficas (EEG) empleando únicamente tres electrodos, conforme al sistema internacional 10-20. Este esquema se ha estandarizado para la colocación de electrodos en el cuero cabelludo y permite una ubicación precisa de los mismos en relación con las áreas del cerebro subyacentes. El estudio se centra en la evaluación del sistema BCI diseñado en el presente trabajo y la eficacia de utilizar una configuración minimalista para la recopilación de datos EEG, en el desarrollo de aplicaciones portátiles y accesibles para el monitoreo cerebral.

1. Objetivos y diseño del estudio: El estudio se diseñó con el objetivo de investigar la calidad de adquisición de señales de EEG de adultos jóvenes, en un sistema embebido, mientras se resuelve una versión en línea adaptada de una prueba de coeficiente intelectual WAIS (Escala de Inteligencia de Weshsler para Adultos) (Pradas & Vicente, 2022)

en el sitio web de Psicología-Online. Se optó por un enfoque descriptivo, seleccionando una muestra de tres participantes, según criterios de inclusión y exclusión definidos previamente en el protocolo de investigación. El análisis de los datos se estructuró para evaluar las hipótesis planteadas, utilizando la transformada rápida de Fourier y la transformada Wavelet.

2. Selección de participantes: Los participantes fueron seleccionados mediante un proceso de selección aleatoria estratificada para garantizar la representatividad de diversas características demográficas relevantes para el estudio, asegurando que todos cumplieran con los criterios establecidos de edad, género y nivel académico (licenciatura). Se obtuvo el consentimiento informado de cada participante, garantizando la comprensión de los procedimientos, riesgos y beneficios del estudio.
3. Preparación del entorno: El entorno experimental fue acondicionado para minimizar interferencias y ruidos externos, utilizando una habitación con condiciones controladas de iluminación y temperatura. Este ambiente optimizado fue crucial para asegurar la integridad de las señales de EEG adquiridas.
4. Preparación del sujeto: La preparación del sujeto incluyó varias etapas esenciales:
 - *Limpieza de la piel*: Se limpió la piel de la frente y el lóbulo de la oreja, empleando algodón y alcohol etílico en los puntos de colocación de los electrodos para eliminar impurezas que pudieran afectar la calidad de la señal (grasa y células muertas).
 - *Colocación de electrodos*: Se empleó el sistema de colocación de electrodos 10-20, ocupando las posiciones FP2 (medición de la señal en el córtex prefrontal), A2 (referencia de la señal en el lóbulo de la oreja) y A1 (referencia a tierra del cuerpo en el lóbulo de la oreja opuesta).
 - *Aplicación de pasta conductiva*: Se aplicó una pasta conductiva marca Ten20 en los electrodos para mejorar la conexión eléctrica y reducir la impedancia.

- *Fijación del electrodo en la posición deseada*: Se aplicó un trozo de cinta quirúrgica 3M en los electrodos para garantizar la fijación y adherencia en la piel.

5. Configuración del equipo de EEG de diseño propietario en el presente trabajo: El equipo de EEG fue calibrado conforme a las especificaciones del fabricante del AFE antes de iniciar la sesión. Se realizaron pruebas preliminares para asegurar la funcionalidad de cada electrodo y la claridad de las señales EEG. A continuación, se presentan los valores empleados en cada registro de configuración del AFE, los cuales establecen las características de discretización de la señal a medir:

CONFIG1: Configuration Register 1 (address = 01h) (reset = 96h)		
Bit	Campo	Descripción y valor
7	Reserved	Reservado Siempre vale 1h Daisy-chain o modo de lecturas múltiples
6	DAISY_EN	El bit determina el modo habilitado 1: Modo de lecturas múltiples Conexión de CLK
5	CLK_EN	El bit determina si la señal del oscilador interno de CLK se a través de un pin. 0: Salida de CLK deshabilitado
4:3	Reserved	Reservado Siempre vale 2h Frecuencia de muestreo
2:0	DR[2:0]	Estos bits determinan la frecuencia de muestreo del dispositivo. $f_{MOD} = f_{CLK} / 2$ 110: $f_{MOD} / 4096$ (250 SPS)

Cuadro 5.1: Tabla de configuración del registro CONFIG1. El valor de escritura equivale a: 0b11010110. Este apartado establece una tasa de muestreo de 250 muestras por segundo.

CONFIG2: Configuration Register 2 (address = 02h) (reset = C0h)

Bit	Campo	Descripción y valor
7:5	Reserved	Reservado Siempre vale 6h
4	INT_CAL	Señal de TEST El bit determina el origen de la señal de prueba. 1: La señal se genera internamente
3	Reserved	Reservado Siempre vale 0h
2	CAL_AMP	Amplitud de TEST El bit determina la calibración de la amplitud de la señal de prueba. $0 : 1 \times -(VREFP - VREFN) / 2400$
1:0	CAL_FREQ[1:0]	Frecuencia de TEST Estos bits determinan la calibración de la frecuencia de la señal de prueba. $00 : \text{Pulsos a } f_{CLK} / 2^{21}$

Cuadro 5.2: Tabla de configuración del registro CONFIG2. El valor establecido equivale a: 0b11010000. Este registro sólo es necesario configurarlo cuando se requiere de una señal de prueba cuadrada para garantizar el funcionamiento del canal del AFE.

CHnSET: Individual Channel Settings (address = 05h to 0Ch) (reset = 61h)

Bit	Campo	Descripción y valor
7	PDn	Apagado El bit determina el modo de energía para el canal correspondiente. 0: Operación normal
6:4	GAINn[2:0]	Ganancia PGA Estos bits determinan la configuración de ganancia del PGA. 110 : 24
3	SRB2	Conexión SRB2 El bit determina la conexión de SRB2 para el canal correspondiente. 0: Abierto
2:0	MUXn[2:0]	Entrada del canal Estos bits determinan la selección de entrada de señal del canal correspondiente. 0 : Entrada normal de electrodo

Cuadro 5.3: Tabla de configuración del registro CHnSET. El valor establecido equivale a: 0b01100000. Este registro debe ser configurado por cada canal que se requiera utilizar. Este apartado establece la operación normal del electrodo con una ganancia de x24 y conexión a la referencia SRB2. El multiplexor permite seleccionar entre la señal que proviene de los electrodos y la de prueba que se genera de forma interna.

6. Recopilación de datos: Durante la recogida de datos, se proporcionaron instrucciones claras al sujeto para garantizar la consistencia en la respuesta a las tareas asignadas

y para minimizar artefactos. Las señales fueron monitorizadas en tiempo real en un lapso de 20 minutos (duración del examen de IQ) en la interfaz diseñada en LabVIEW para identificar y corregir cualquier irregularidad. Al detener la adquisición, la serie de tiempo discretizada se guarda en un archivo .data para su procesamiento posterior.

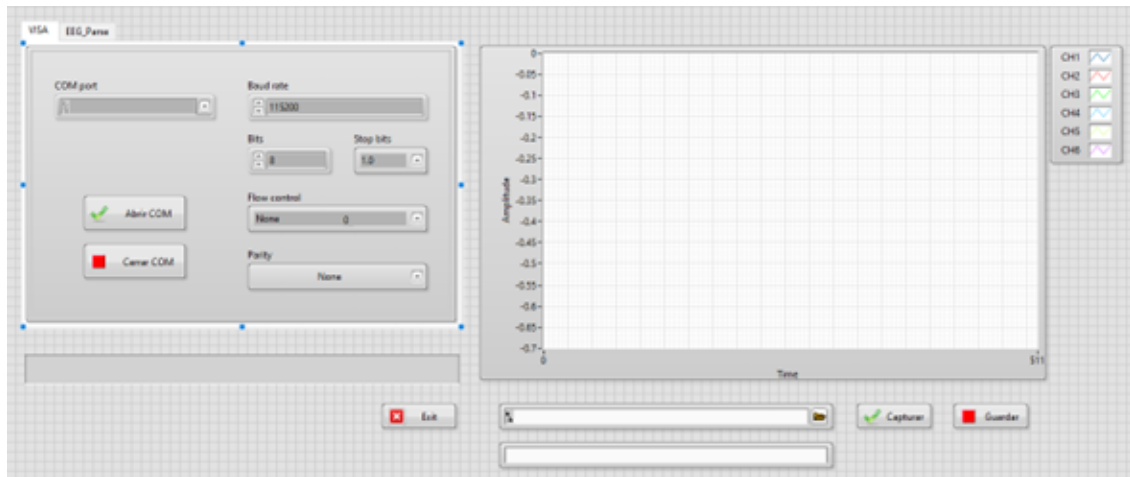


Figura 5.1: Interfaz de adquisición de datos en LabVIEW de la tarjeta electrónica del BCI.

7. Finalización del experimento y desmontaje: Al concluir la recogida de datos, se procedió a retirar los electrodos y a limpiar la piel del sujeto, agradeciendo su participación y proporcionando instrucciones para cualquier atención posterior necesaria.
8. Análisis de datos: El procesamiento de las señales de EEG incluyó un filtro digital FIR (Finite Impulse Response) de primer orden con frecuencia de corte en 100 Hz, además de técnicas segmentación para preparar los datos para el análisis. El análisis se llevó a cabo utilizando la FFT (2000 muestras) y la DWT (empleando la wavelet daubechies 3).
9. Interpretación y reporte de resultados: Los datos obtenidos fueron interpretados en el contexto de la observación de la presencia de actividad cerebral relacionada con el razonamiento lógico en la zona del córtex prefrontal.

Este diseño meticuloso aseguró la recogida de datos de alta calidad, fundamentales para validar las hipótesis de investigación y para la reproducibilidad de los resultados en estudios futuros.

5.1.2. Resultados del análisis de las señales EEG

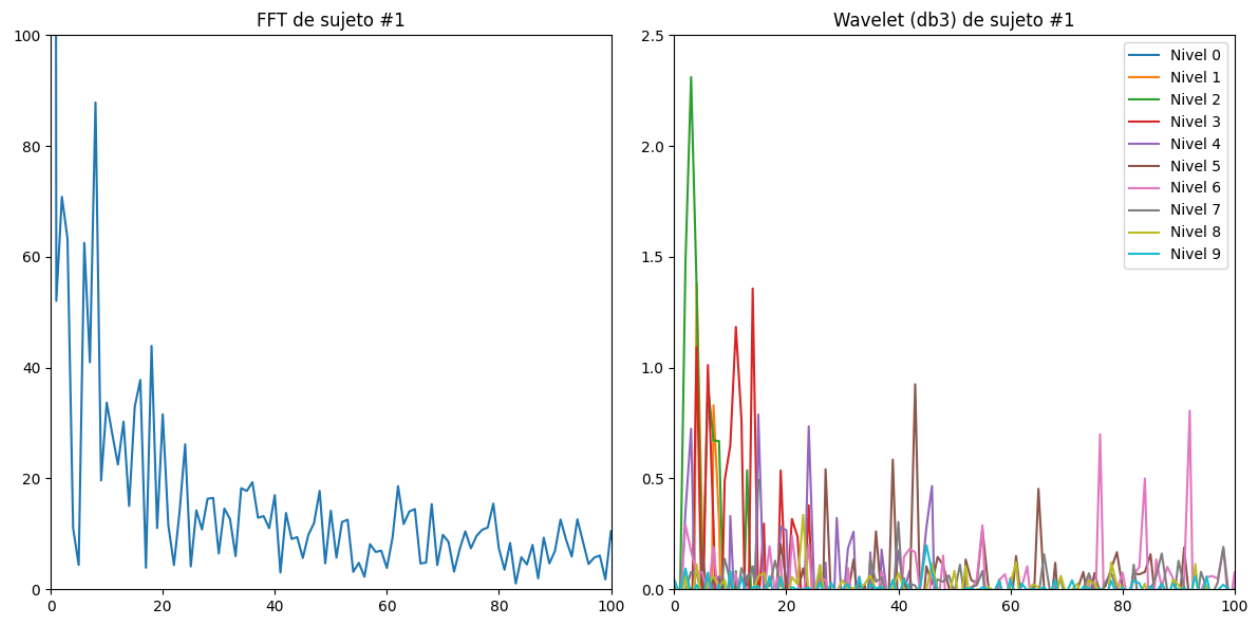


Figura 5.2: Análisis de señal EEG del sujeto #1 (hombre de 25 años del área académica de posgrado en ciencias).

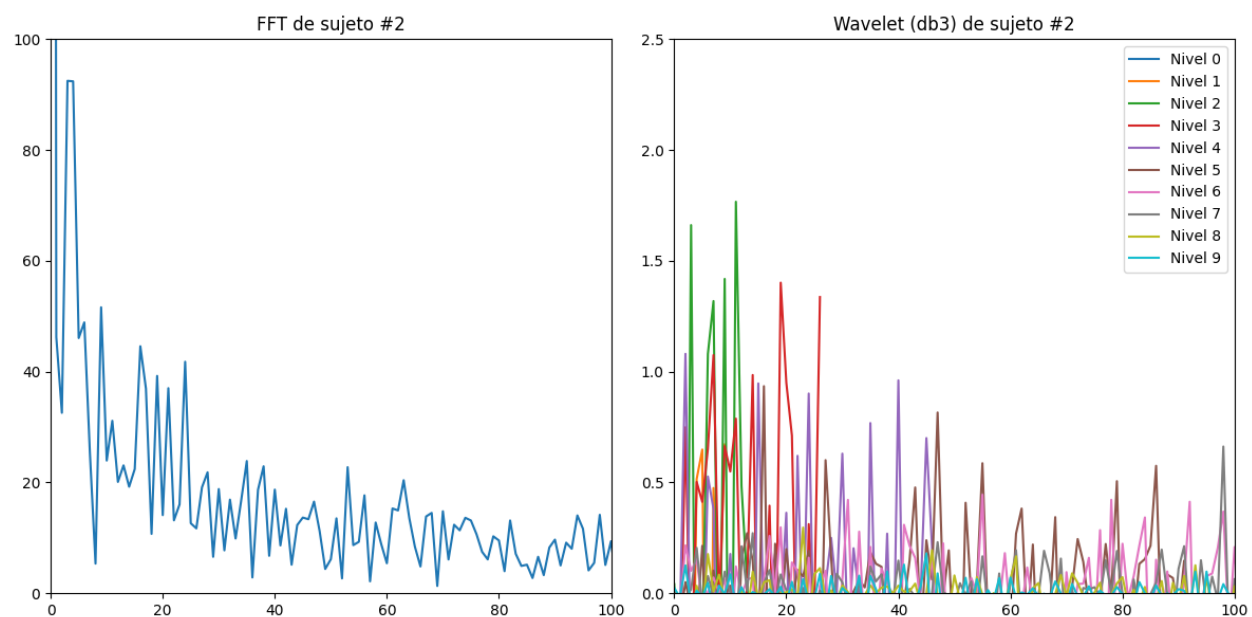


Figura 5.3: Análisis de señal EEG del sujeto #2 (hombre de 26 años del área académica de posgrado en ciencias).

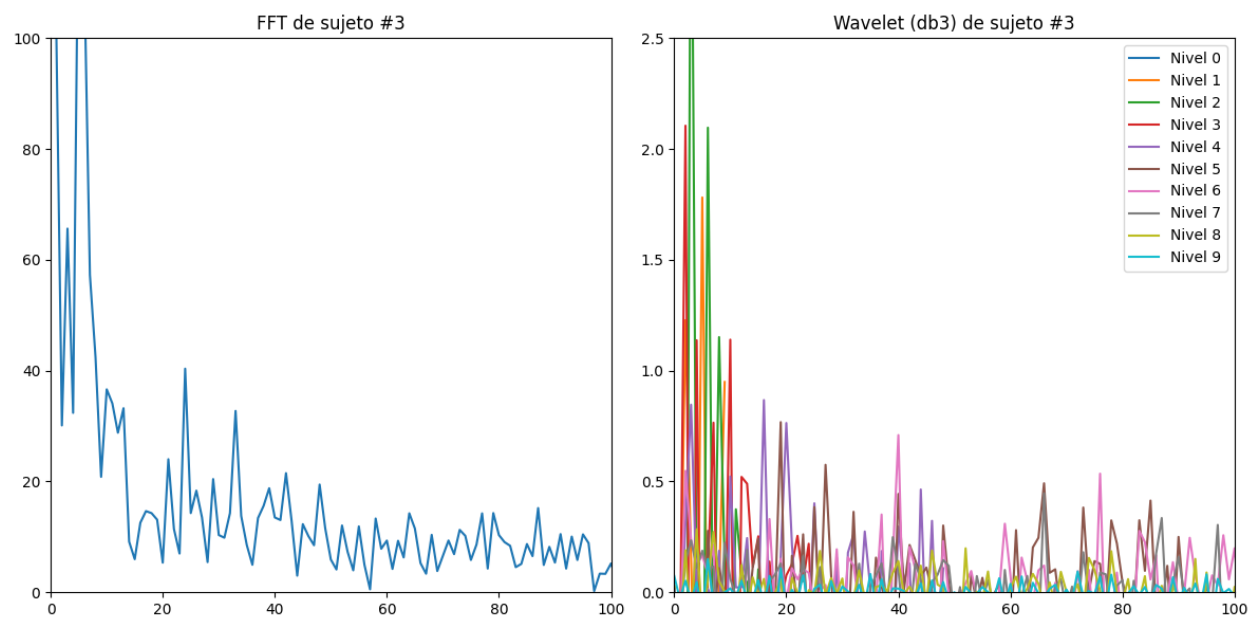


Figura 5.4: Análisis de señal EEG del sujeto #3 (mujer de 27 años del área académica de posgrado en ciencias).

Diseño experimental para algoritmos embebidos

1. Definición del problema: El objetivo es evaluar el rendimiento de varios algoritmos de machine learning en la base de datos EEG Eye State, utilizando validación cruzada k-folds, Análisis de Componentes Principales (PCA) para la reducción de dimensionalidad y la correspondiente conversión del dataset de formato float a unsigned char.
2. Diseño del experimento: Se realizará un diseño factorial completo, con diferentes métricas de distancia, diferentes valores de k para PCA y validación cruzada k-folds. Para cada combinación de factores, se realizará una serie de experimentos. Cada experimento se repetirá varias veces para garantizar la estabilidad de los resultados.
3. Preprocesamiento de datos: Antes de aplicar los algoritmos, es necesario normalizar los datos de la base de datos EEG Eye State y aplicar PCA para reducir la dimensionalidad.
4. Ejecución de los experimentos: Para cada algoritmo, se entrenará y evaluará utilizando validación cruzada k-folds en 100 bajo el esquema de una computadora de escritorio y el sistema embebido. Se trabajará únicamente con un solo hiperparámetro (con valor arbitrario) para cada algoritmo. Se calcularán métricas de rendimiento como precisión, exactitud, sensibilidad y especificidad.
5. Análisis de los resultados: Se analizarán los resultados y se llevarán a cabo pruebas post-hoc para identificar diferencias estadísticamente significativas en el desempeño de los algoritmos. Además, se pueden utilizar gráficos de interacción y análisis de componentes principales para visualizar y comprender mejor las relaciones entre los factores y su impacto en el rendimiento.
6. Conclusiones: Basándose en los resultados del análisis, se identificarán las condiciones bajo las cuales los algoritmos embebidos presentan un rendimiento óptimo (porcentaje de clasificación por arriba del 70 % y bajo uso de memoria flash y RAM dentro del microprocesador) en la base de datos EEG Eye State y se discutirán las implicaciones de estos resultados para la práctica y futuras investigaciones.

Diagrama de flujo del desarrollo experimental

1. Inicio
2. Cargar la base de datos (EEG Eye State)
3. Preprocesar los datos:
 - a) Eliminar outliers
 - b) Normalización
 - c) Conversión de float a unsigned char.
4. Aplicar K-Folds en 100
5. Aplicar PCA:
 - Seleccionar el valor de k (número de componentes principales representativos)
 - Reducir la dimensionalidad
6. Para cada algoritmo de machine learning con hiperparámetro principal arbitrario:
 - Con PCA y sin PCA:
 - Sistema embebido
 - Computadora de escritorio
7. Analizar resultados:
 - Realizar análisis y pruebas post-hoc
 - Visualizar resultados y analizar interacciones entre factores
8. Sacar conclusiones
9. Fin

5.2. Análisis y clasificación de dataset con etiquetas y pruebas en sistema embebido

El conjunto de datos utilizado en este estudio fue recopilado por Roesler y está disponible para su uso público en el repositorio de datos de la UCI. Este conjunto de datos se utiliza frecuentemente para la identificación del estado de los ojos. Se compone de grabaciones electromagnéticas no procesadas tomadas del cuero cabelludo de un participante, así como información sobre si sus ojos estaban abiertos o cerrados en ese momento. Utilizando el BCI Emotiv EEG, se realizó una medición continua de EEG para producir todos los datos. El conjunto incluye 14980 patrones y 14 características, donde las últimas corresponden a los datos recogidos por los sensores. Las correlaciones entre estas características se presentan en la Figura ???. La medición duró 117 segundos. Durante la medición del EEG, se registró el estado de los ojos con una cámara, y tras revisar los fotogramas del video, se añadió manualmente esta información al archivo.

5.2.1. Exploratory Data Analysis (EDA)

5.2.1.1. Análisis estadístico descriptivo

Ojos abiertos

- Las medias de las características varían alrededor de 4300, con desviaciones estándar en torno a 25.
- Los valores mínimos y máximos están dentro de un rango razonable de la media, lo que indica una distribución normal sin valores atípicos extremos.

Ojos cerrados

- Las medias de las características varían alrededor de 4100, con desviaciones estándar ligeramente más altas que para el estado de ojos abiertos, en torno a 30.

- Al igual que con los ojos abiertos, los valores mínimos y máximos están dentro de un rango esperado de la media.

T-Tests

Los resultados del t-test muestran t-estadísticas significativas para todas las características, con p-valores extremadamente bajos (prácticamente cero), lo que indica que hay diferencias estadísticamente significativas entre los estados de ojos abiertos y cerrados para todas las características medidas.

- AF3, FC5, T7, P7, O1, P8, AF4: Estos tienen un valor p alto ($p > 0.05$), lo que indica que no hay una diferencia estadísticamente significativa en estos casos.
- F7, F3, O2, T8, FC6, F4: Estos muestran un valor p muy bajo ($p < 0.05$), indicando una diferencia estadísticamente significativa entre los grupos para estas variables.
- F8: Tiene un valor p que está cerca del umbral de 0.05, lo que podría sugerir una posible diferencia, pero no es concluyente.

Estadística descriptiva (Ojos abiertos)					
	N	Mínimo	Máximo	Media	Desviación estándar
	Estadístico	Estadístico	Estadístico	Estadístico	Estadístico
AF3	8257	4197.95	4504.1	4298.0686	40.652
F7	8257	3924.1	4156.92	4012.7937	31.635
F3	8257	4197.44	4386.15	4262.8903	22.20171
FC5	8257	4073.33	4250.26	4123.3197	20.56495
T7	8257	4304.62	4463.59	4341.5613	16.08134
P7	8257	4566.15	4756.92	4621.1798	19.27816
O1	8257	4027.18	4178.46	4071.9733	17.80437
O2	8257	4567.2	4731.8	4614.925	18.2165
P8	8257	4152.3	4320	4200.308	17.1726
T8	8257	4152.82	4362.56	4229.4507	19.92239
FC6	8257	4100	4332.31	4200.0248	24.88416
F4	8257	4201.03	4397.95	4276.93	21.00663
F8	8257	4443.08	4833.85	4602.1036	33.39582
AF4	8257	4205.64	4573.33	4356.6255	40.61546

Cuadro 5.4: Ojos abiertos

Estadística descriptiva (Ojo cerrado)					
	N	Mínimo	Máximo	Media	Desviación estándar
	Estadístico	Estadístico	Estadístico	Estadístico	Estadístico
AF3	6722	4198.97	4445.13	4305.4429	33.45631
F7	6722	3905.64	4138.97	4005.4726	27.54161
F3	6722	4212.31	4367.18	4265.5488	20.12595
FC5	6722	4058.46	4214.36	4121.221	21.31324
T7	6722	4309.74	4435.38	4341.5613	18.08085
P7	6722	4574.87	4708.72	4618.6865	17.44129
O1	6722	4026.15	4167.18	4073.8663	24.14483
O2	6722	4567.7	4695.9	4616.849	18.4396
P8	6722	4147.7	4287.7	4202.611	18.5517
T8	6722	4174.36	4323.08	4233.3547	19.35768
FC6	6722	4130.77	4319.49	4204.7597	23.70626
F4	6722	4225.64	4368.72	4281.742	18.57421
F8	6722	4510.26	4811.28	4610.806	32.78862
AF4	6722	4246.15	4552.82	4367.05	34.81985

Cuadro 5.5: Ojo cerrado

Prueba T		
	Estadístico T	Valor P
AF3	-1.279921	1.064632e-22
F7	9.821569	2.005926e-01
F3	-4.764648	1.909585e-06
FC5	0.921654	3.567241e-01
T7	0.045125	9.640081e-01
P7	0.960143	3.369986e-01
O1	0.884043	3.766872e-01
O2	-3.072788	2.124499e-03
P8	-1.171993	2.412187e-01
T8	-5.785225	7.386151e-09
FC6	-7.884877	3.364595e-15
F4	-5.876886	4.269353e-09
F8	-1.605869	1.083238e-01
AF4	0.923971	3.555164e-01

Cuadro 5.6: Prueba T

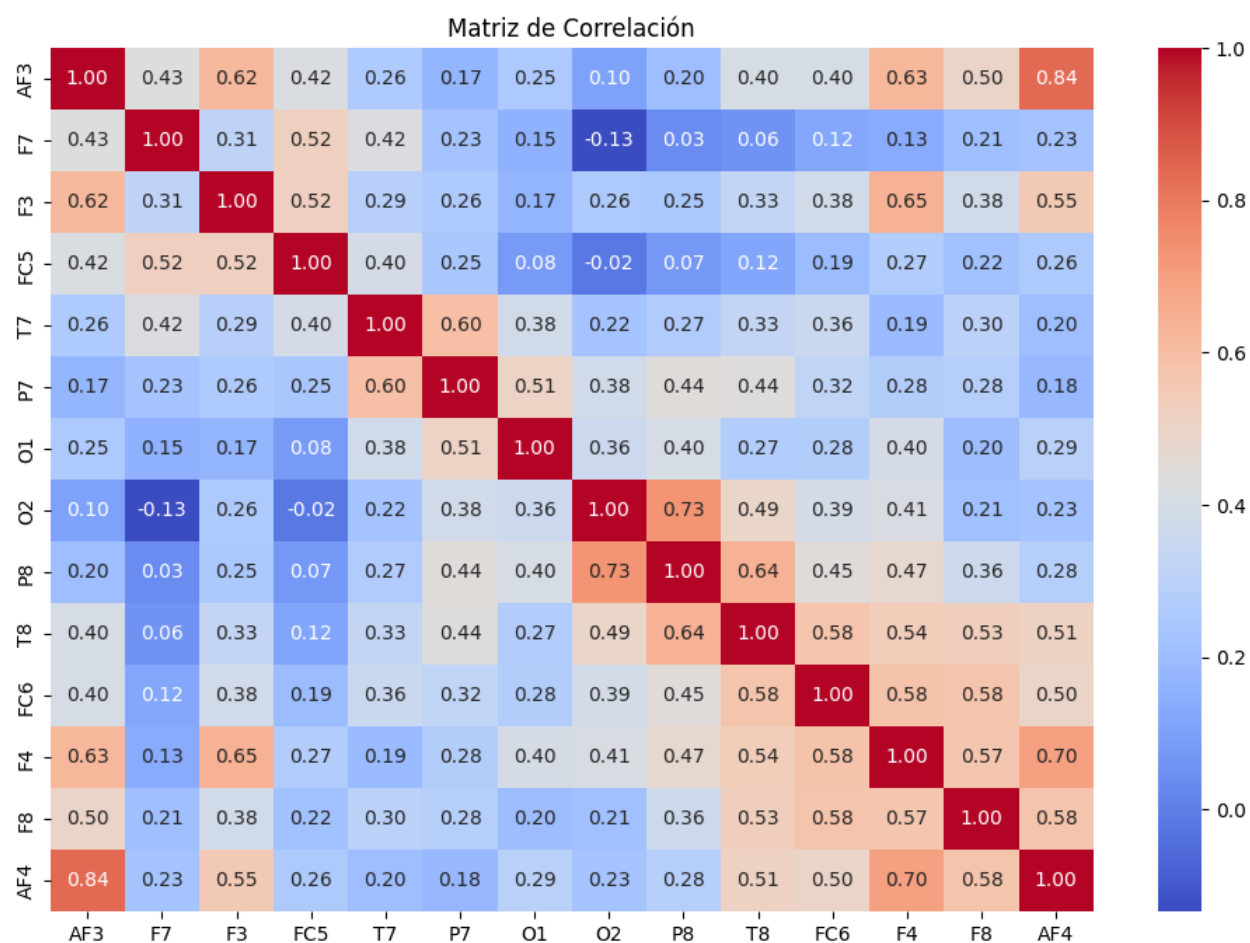


Figura 5.5: Matriz de correlación.

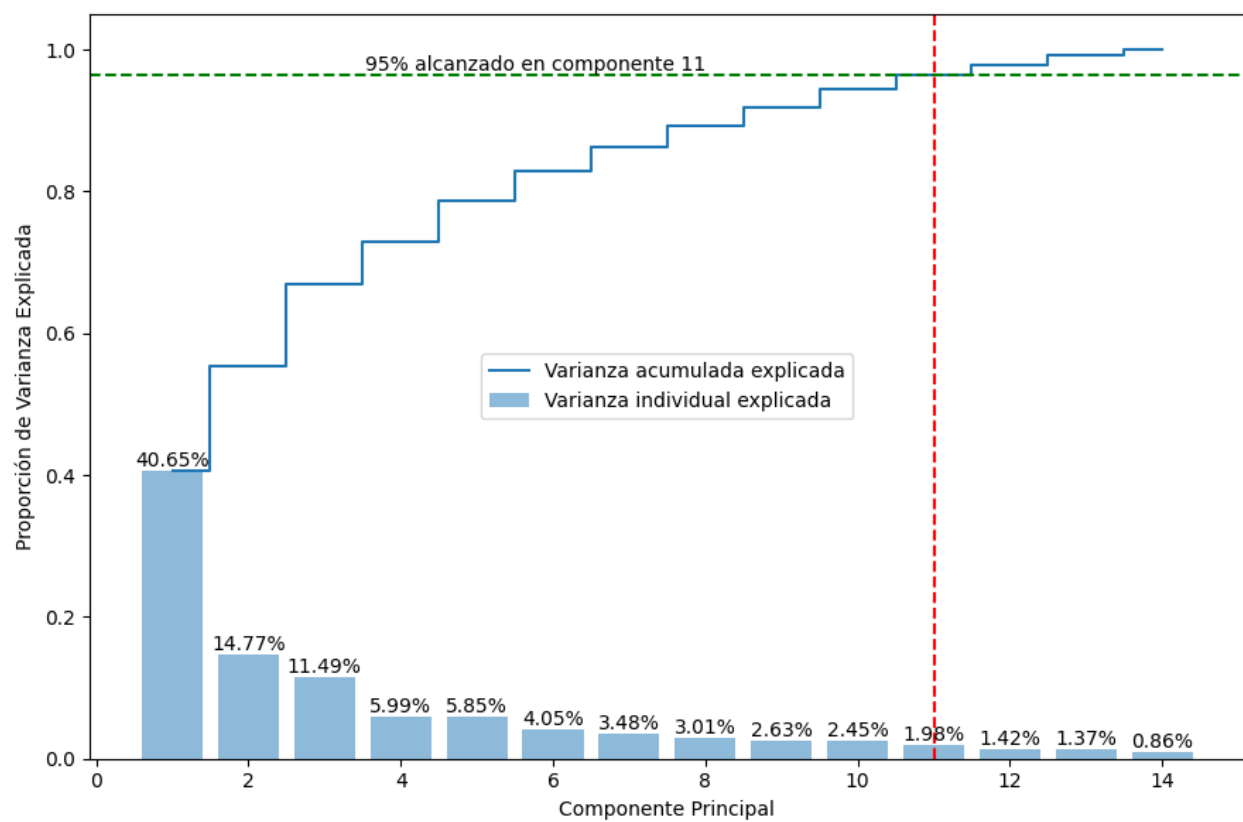


Figura 5.6: Gráfica de varianza explicada.

PCA				
<i>Eje</i>	<i>Eigen valor</i>	<i>Diferencia</i>	<i>Proporción (%)</i>	<i>Acumulado (%)</i>
1	5.691817	0.000000	40.652086	40.652086
2	2.068591	-3.623225	14.774289	55.426375
3	1.608906	-0.459685	11.491126	66.917501
4	0.838206	-0.770700	5.986635	72.904136
5	0.818681	-0.019525	5.847181	78.751317
6	0.567136	-0.251545	4.050597	82.801914
7	0.486600	-0.080536	3.475391	86.277305
8	0.422093	-0.064507	3.014671	89.291976
9	0.368738	-0.053355	2.633602	91.925578
10	0.343119	-0.025619	2.450628	94.376206
11	0.276706	-0.066413	1.976290	96.352496
12	0.198496	-0.078211	1.417694	97.770191
13	0.192328	-0.006167	1.373645	99.143836
14	0.119874	-0.072454	0.856164	100.000000

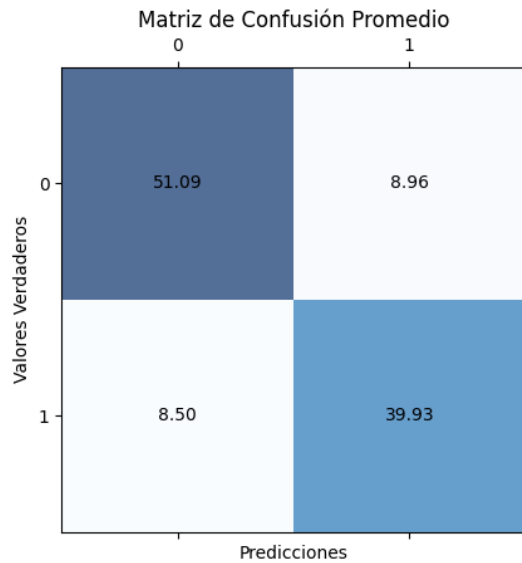
Cuadro 5.7: Análisis de PCA.

5.2.2. Clasificadores en PC

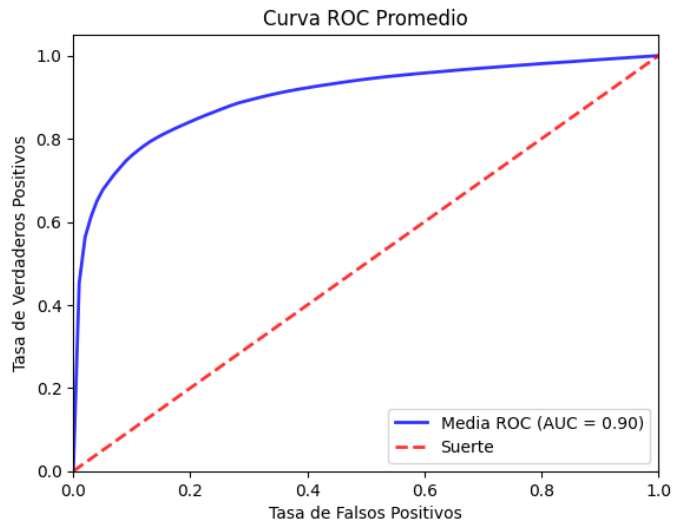
5.2.2.1. K-Nearest Neighbors (KNN)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.8390383961943597	0.8034429153924567
<i>Precisión</i>	0.8342393710754686	0.7991518068503037
<i>Sensibilidad (Recall)</i>	0.8243792517006803	0.7834566326530613
<i>Especificidad</i>	0.8508579234972677	0.8195628415300547
<i>F1 Score</i>	0.8197549220455397	0.7786946768566260

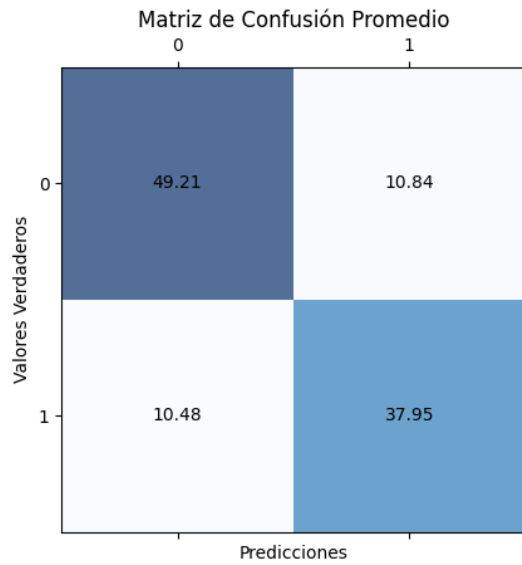
Cuadro 5.8: Resultados de KNN en PC.



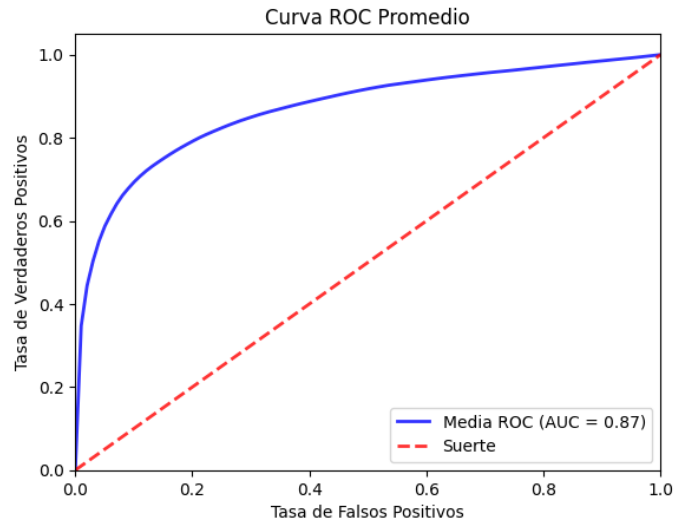
(a) Matriz de confusión de KNN sin PCA.



(b) Curva ROC de KNN sin PCA.



(a) Matriz de confusión de KNN con PCA.

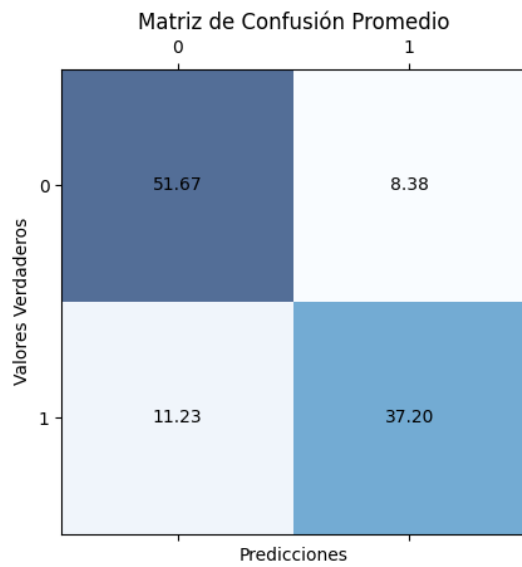


(b) Curva ROC de KNN con PCA.

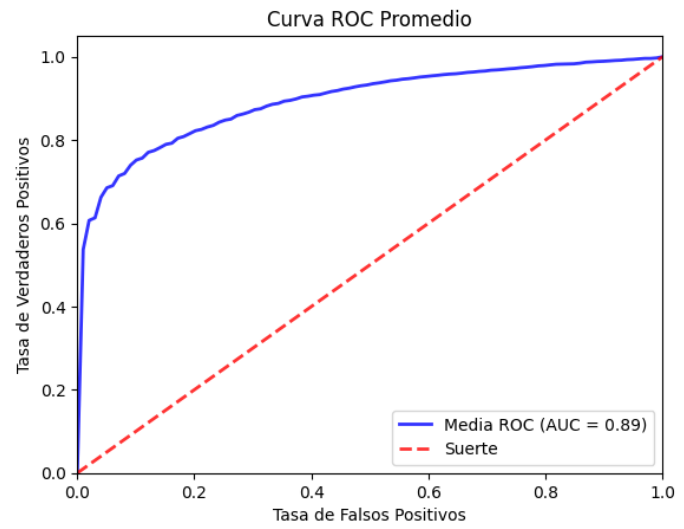
5.2.2.2. Random Forests

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.8192218824328917	0.8068730886850153
<i>Precisión</i>	0.8309417585499748	0.8176704446306732
<i>Sensibilidad (Recall)</i>	0.7679549319727890	0.7656292517006803
<i>Especificidad</i>	0.8606311475409835	0.8402322404371586
<i>F1 Score</i>	0.7803498200727564	0.7712321113371016

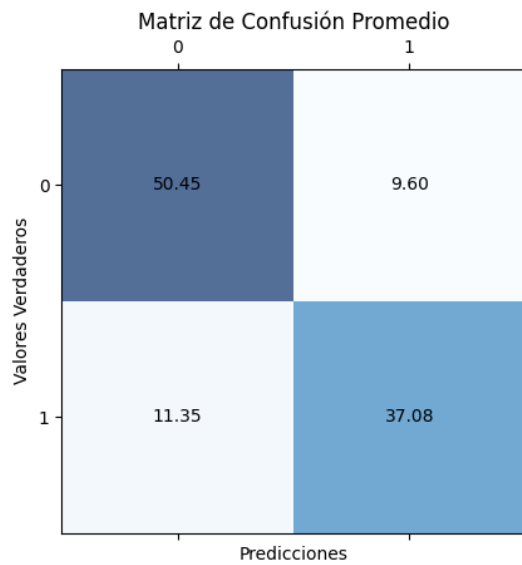
Cuadro 5.9: Resultados de Random Forests en PC.



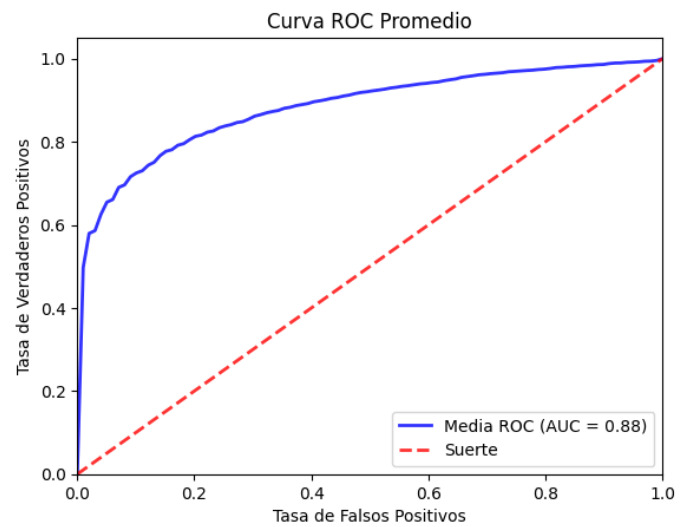
(a) Matriz de confusión de Random Forests sin PCA.



(b) Curva ROC de Random Forests sin PCA.



(a) Matriz de confusión de Random Forests con PCA.

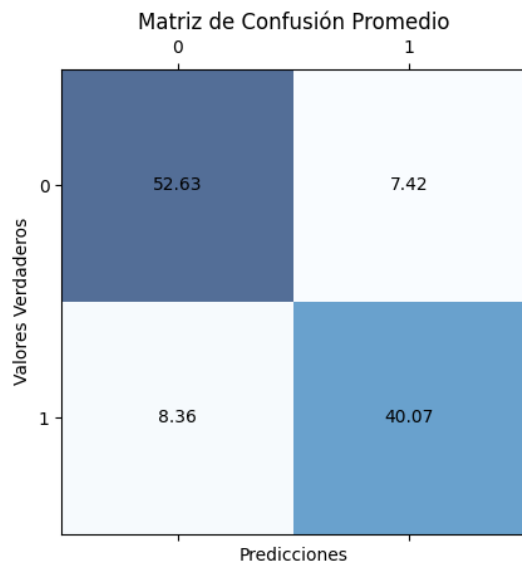


(b) Curva ROC de Random Forests con PCA.

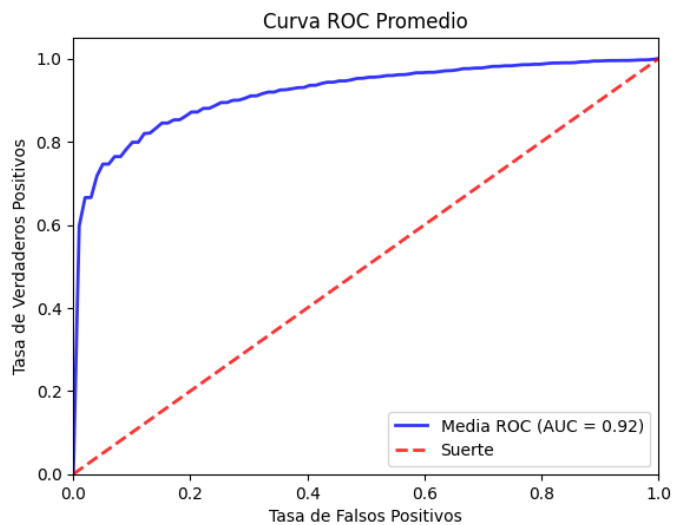
5.2.2.3. Support Vector Machine (SVM)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.8545633707101596	0.8350798504926946
<i>Precisión</i>	0.8590420230250011	0.8370264271714976
<i>Sensibilidad (Recall)</i>	0.8272704081632654	0.8136819727891155
<i>Especificidad</i>	0.8765928961748634	0.8522868852459016
<i>F1 Score</i>	0.8295621231118866	0.8057587492357361

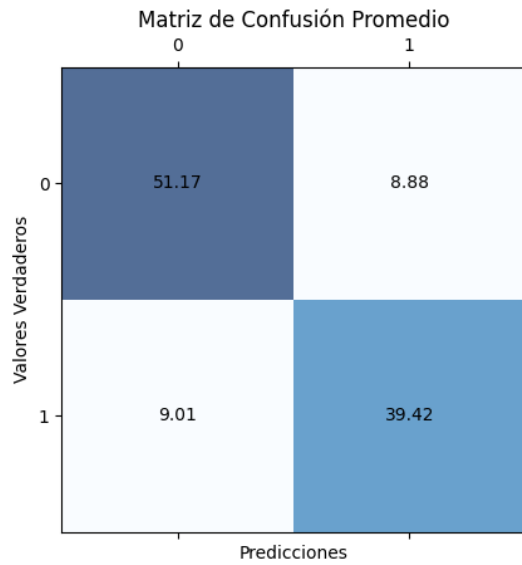
Cuadro 5.10: Resultados de SVM en PC.



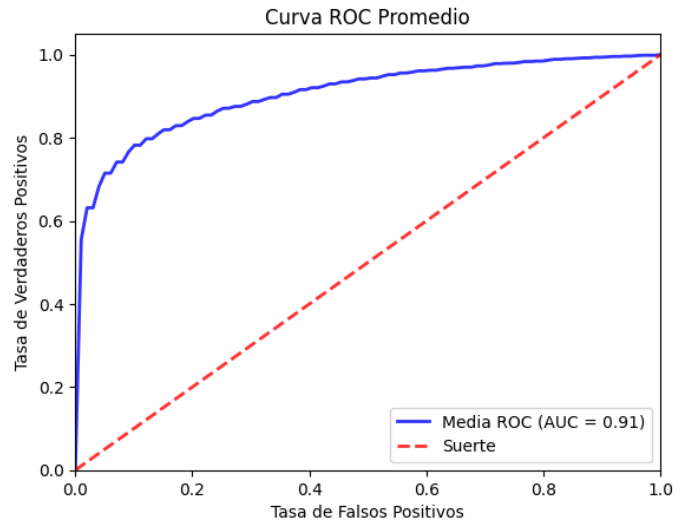
(a) Matriz de confusión de SVM sin PCA.



(b) Curva ROC de SVM sin PCA.



(a) Matriz de confusión de SVM con PCA.

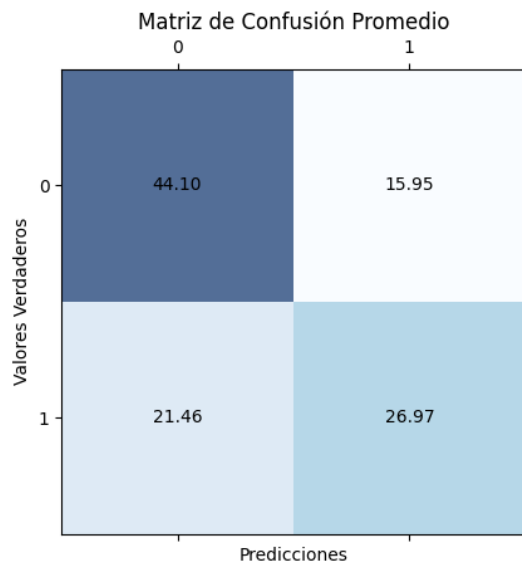


(b) Curva ROC de SVM con PCA.

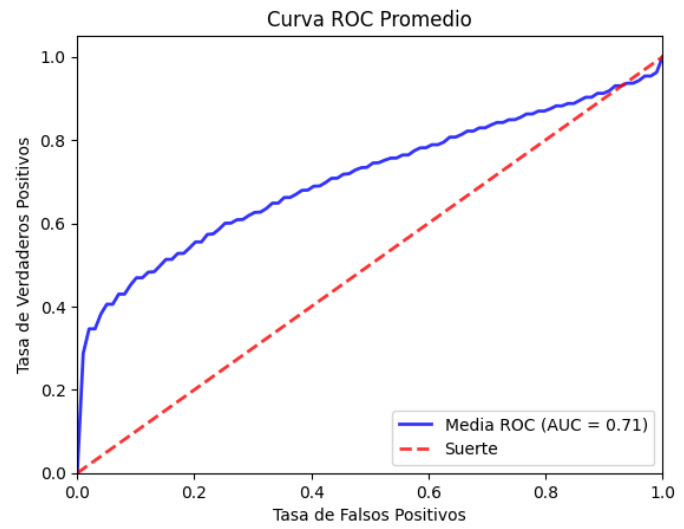
5.2.2.4. Naive Bayes

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6552191641182467	0.6931498470948013
<i>Precisión</i>	0.6489611439579829	0.6919013604162462
<i>Sensibilidad (Recall)</i>	0.5570748299319728	0.5750255102040817
<i>Especificidad</i>	0.7345901639344262	0.7883715846994537
<i>F1 Score</i>	0.5643739757555213	0.5954950300428695

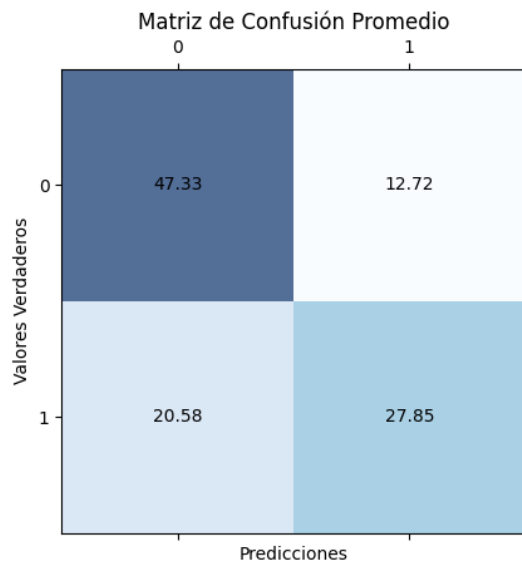
Cuadro 5.11: Resultados de Naive Bayes en PC.



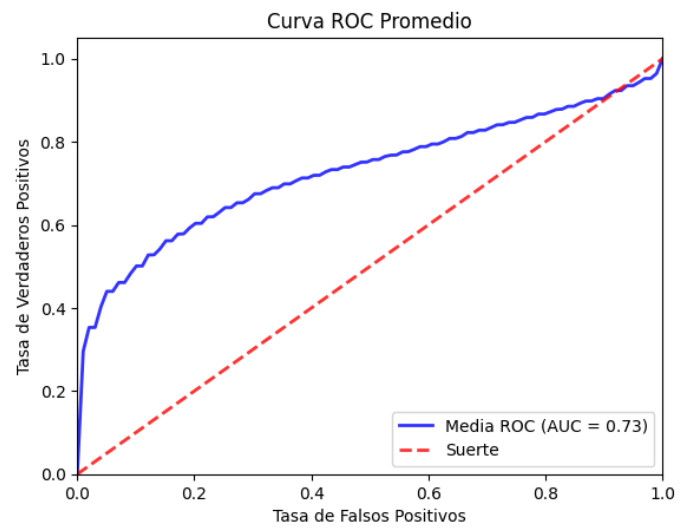
(a) Matriz de confusión de Naive Bayes sin PCA.



(b) Curva ROC de Naive Bayes sin PCA.



(a) Matriz de confusión de Naive Bayes con PCA.

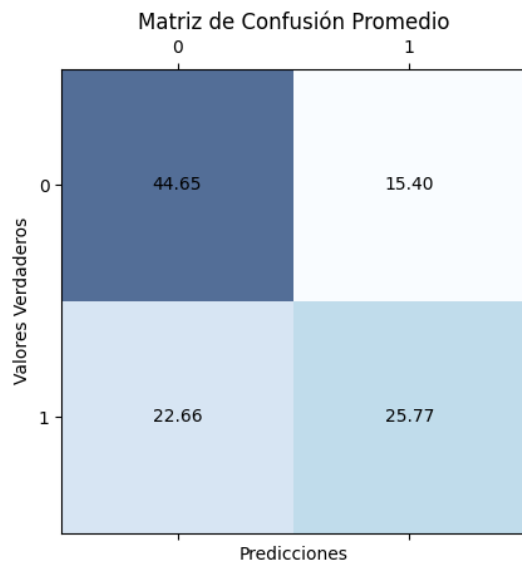


(b) Curva ROC de Naive Bayes con PCA.

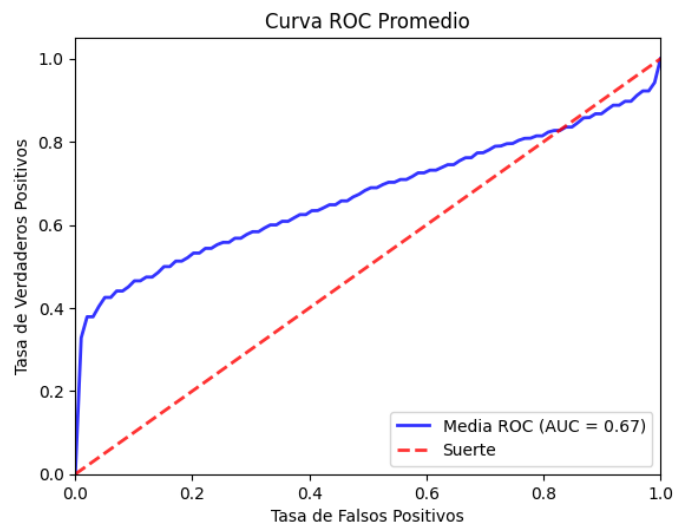
5.2.2.5. Logistic Regression

	K-Folds = 100	
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6492414203194020	0.6570914033299354
<i>Precisión</i>	0.6331943560637892	0.6360603251050079
<i>Sensibilidad (Recall)</i>	0.5326445578231292	0.5333886054421769
<i>Especificidad</i>	0.7436939890710383	0.7571939890710384
<i>F1 Score</i>	0.5442810271246374	0.5485682651163265

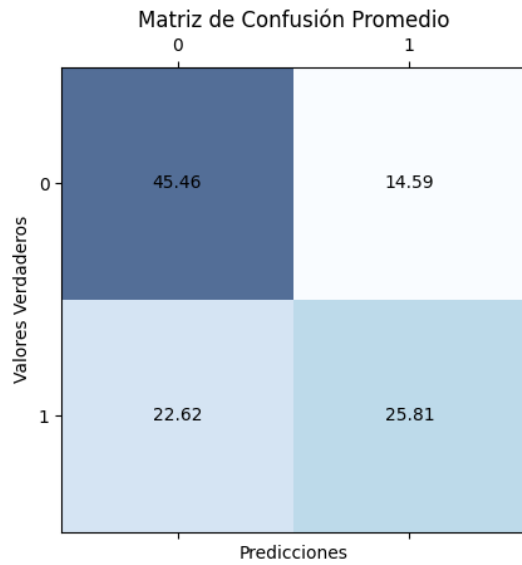
Cuadro 5.12: Resultados de Logistic Regression en PC.



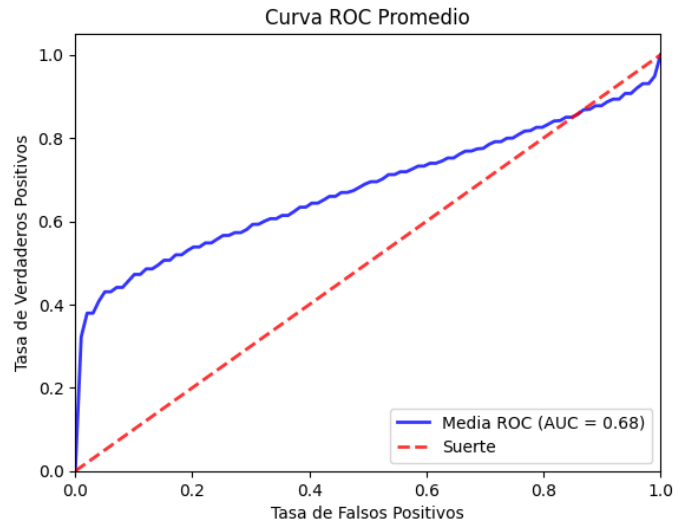
(a) Matriz de confusión de Logistic Regression sin PCA.



(b) Curva ROC de Logistic Regression sin PCA.



(a) Matriz de confusión de Logistic Regression con PCA.

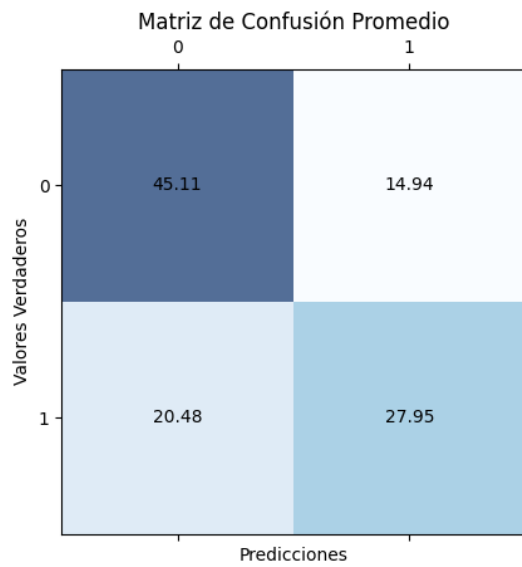


(b) Curva ROC de Logistic Regression con PCA.

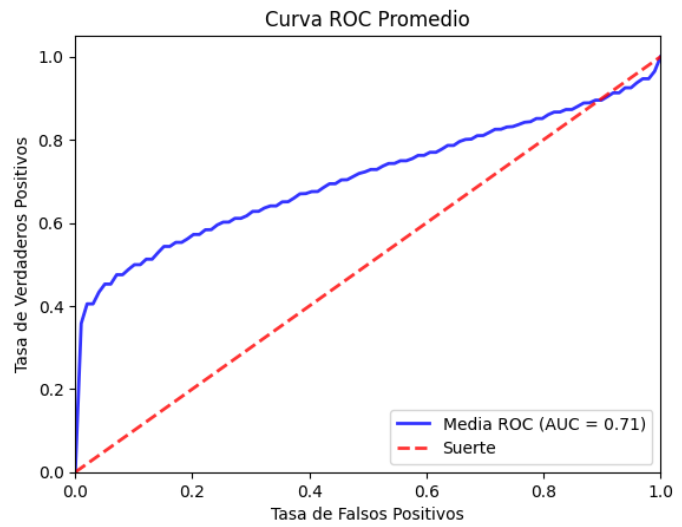
5.2.2.6. Multilayer Perceptron (MLP)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6736408426775398	0.7183171933401291
<i>Precisión</i>	0.6604437106392873	0.7258884147101442
<i>Sensibilidad (Recall)</i>	0.5776870748299320	0.6317389455782313
<i>Especificidad</i>	0.7513606557377048	0.7883961748633882
<i>F1 Score</i>	0.5837261066450803	0.6434568954418545

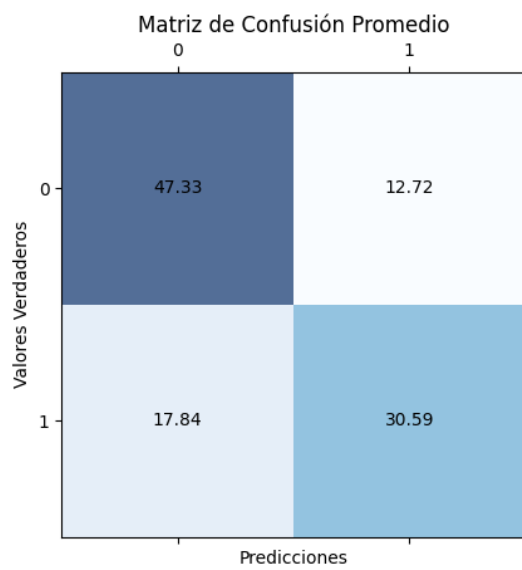
Cuadro 5.13: Resultados de MLP en PC.



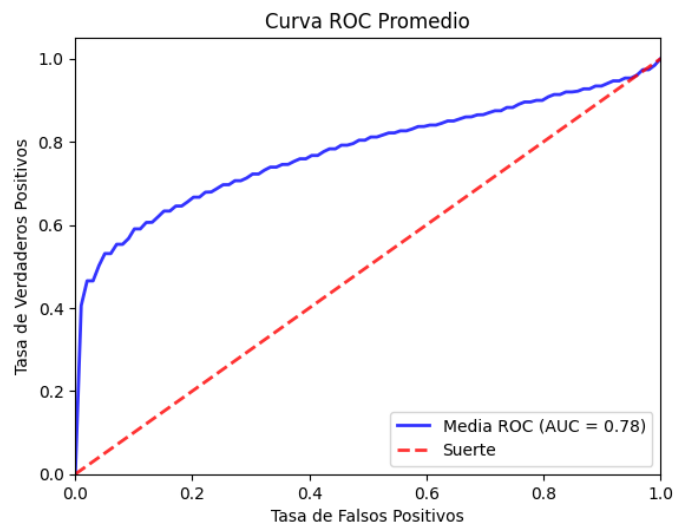
(a) Matriz de confusión de MLP sin PCA.



(b) Curva ROC de MLP sin PCA.



(a) Matriz de confusión de MLP con PCA.

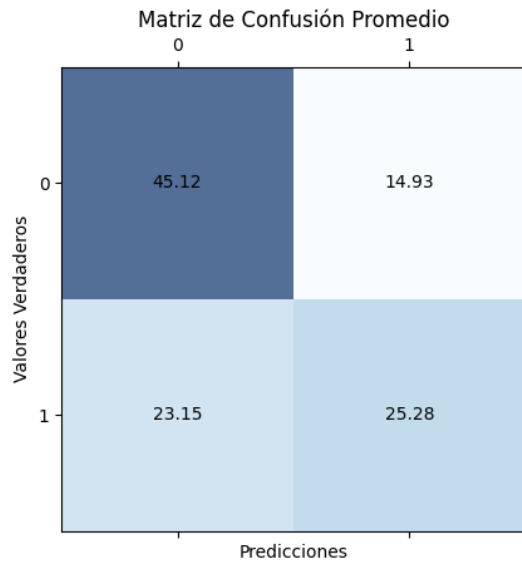


(b) Curva ROC de MLP con PCA.

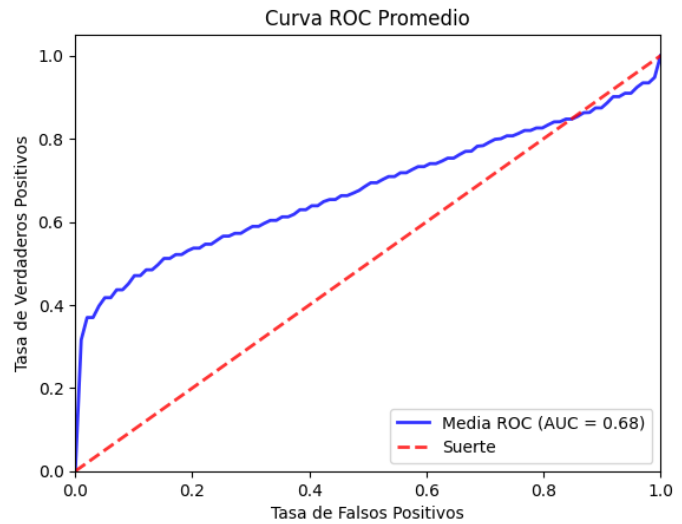
5.2.2.7. LSTM (Long Short Term Memory)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6490817193340129	0.7237368331634387
<i>Precisión</i>	0.6309142338213851	0.7149740663378755
<i>Sensibilidad (Recall)</i>	0.5223596938775510	0.6706717687074830
<i>Especificidad</i>	0.7515273224043715	0.7667786885245904
<i>F1 Score</i>	0.5395438997300921	0.6667862950515898

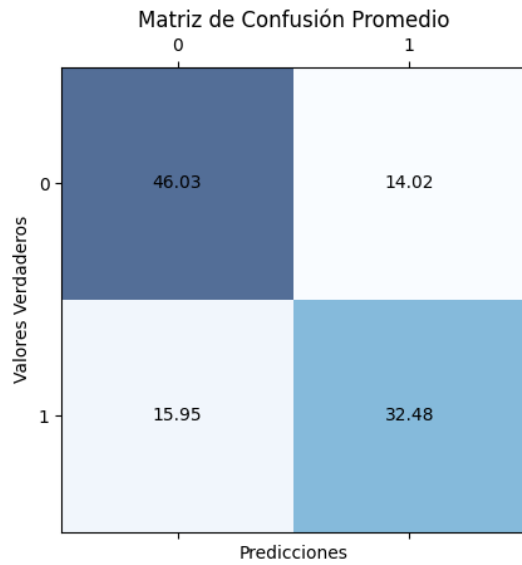
Cuadro 5.14: Resultados de red neuronal LSTM en PC.



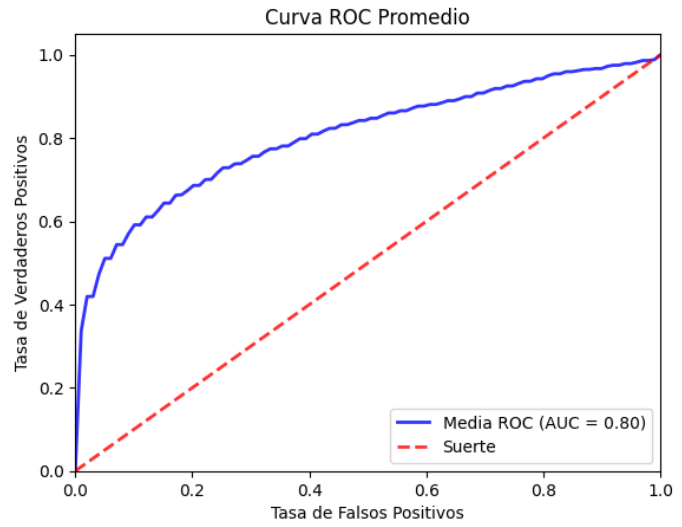
(a) Matriz de confusión de red neuronal LSTM sin PCA.



(b) Curva ROC de red neuronal LSTM sin PCA.



(a) Matriz de confusión de red neuronal LSTM con PCA.



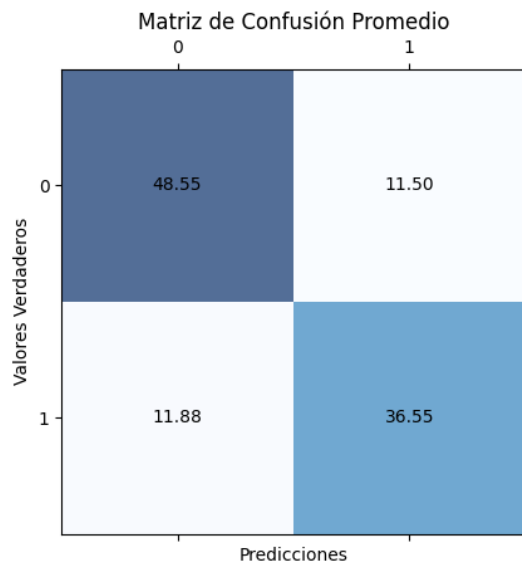
(b) Curva ROC de red neuronal LSTM con PCA.

5.2.3. Clasificadores en sistema embebido

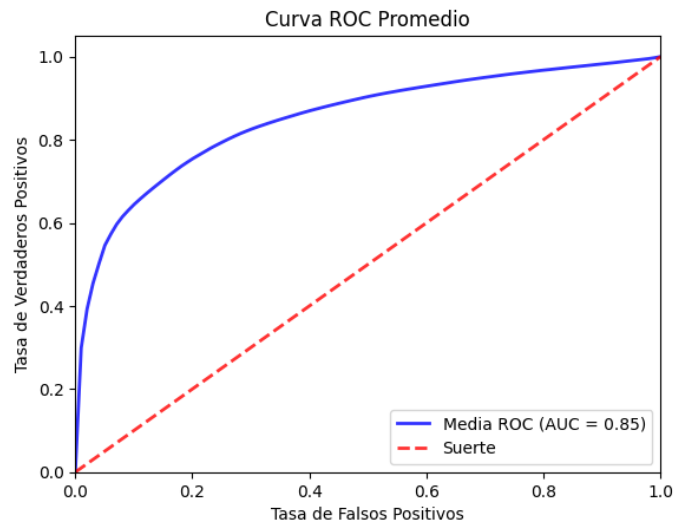
5.2.3.1. K-Nearest Neighbors (KNN)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.7844283044512403	0.7638829425756031
<i>Precisión</i>	0.7728277102281553	0.7575759308852175
<i>Sensibilidad (Recall)</i>	0.7546258503401362	0.7161394557823130
<i>Especificidad</i>	0.8085300546448089	0.8024125683060109
<i>F1 Score</i>	0.7549176329055395	0.7241812371894177

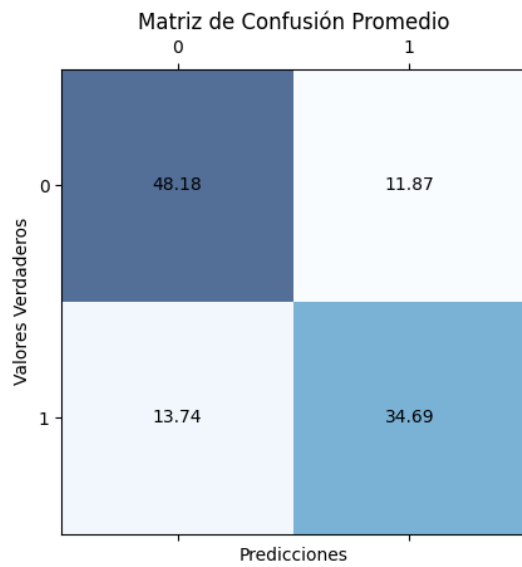
Cuadro 5.15: Resultados de KNN en sistema embebido.



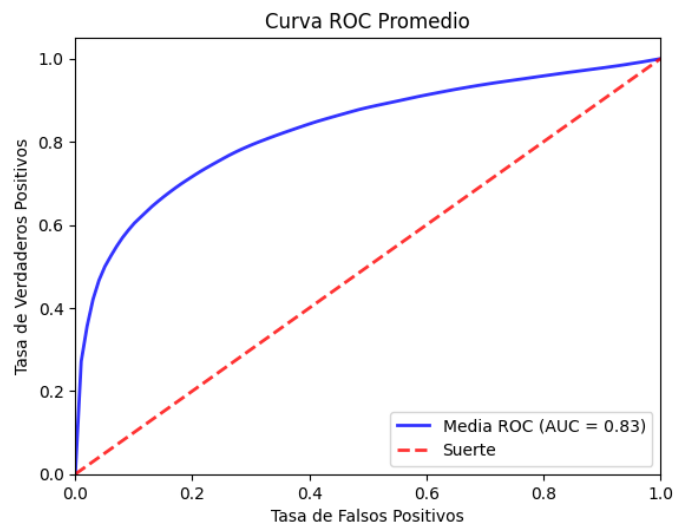
(a) Matriz de confusión de KNN sin PCA.



(b) Curva ROC de KNN sin PCA.



(a) Matriz de confusión de KNN con PCA.

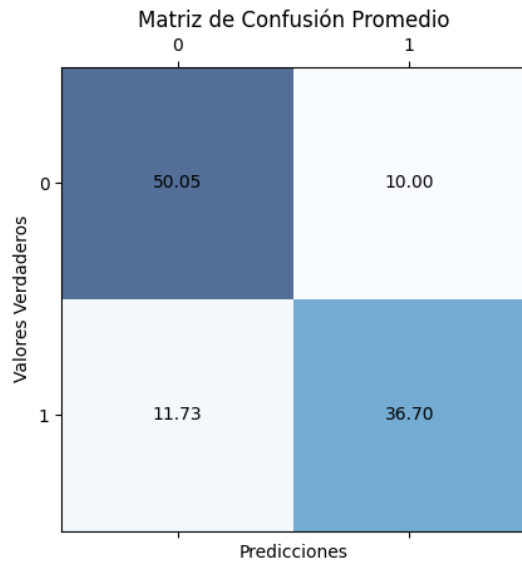


(b) Curva ROC de KNN con PCA.

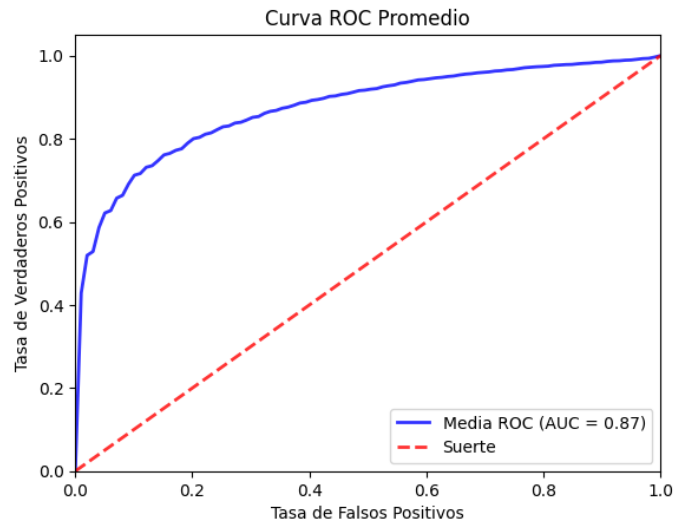
5.2.3.2. Random Forests

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.7996704043493034	0.7651010873258577
<i>Precisión</i>	0.8016429700429061	0.7709679167157439
<i>Sensibilidad (Recall)</i>	0.7576360544217688	0.7050510204081633
<i>Especificidad</i>	0.8336666666666668	0.8136202185792349
<i>F1 Score</i>	0.7651208625088713	0.7182225881609356

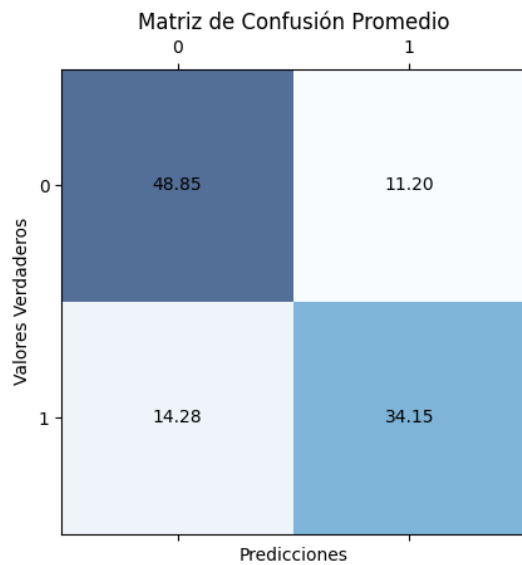
Cuadro 5.16: Resultados de Random Forests en sistema embebido.



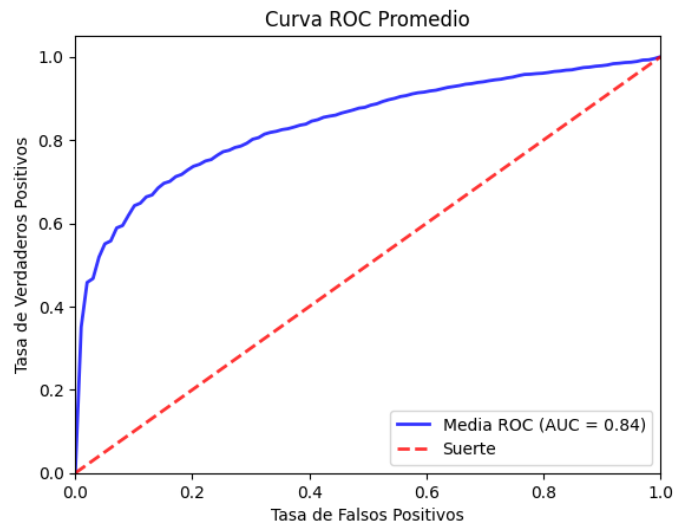
(a) Matriz de confusión de Random Forests sin PCA.



(b) Curva ROC de Random Forests sin PCA.



(a) Matriz de confusión de Random Forests con PCA.

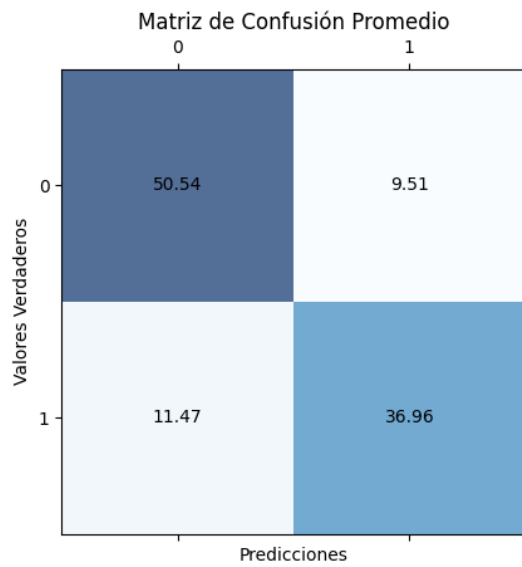


(b) Curva ROC de Random Forests con PCA.

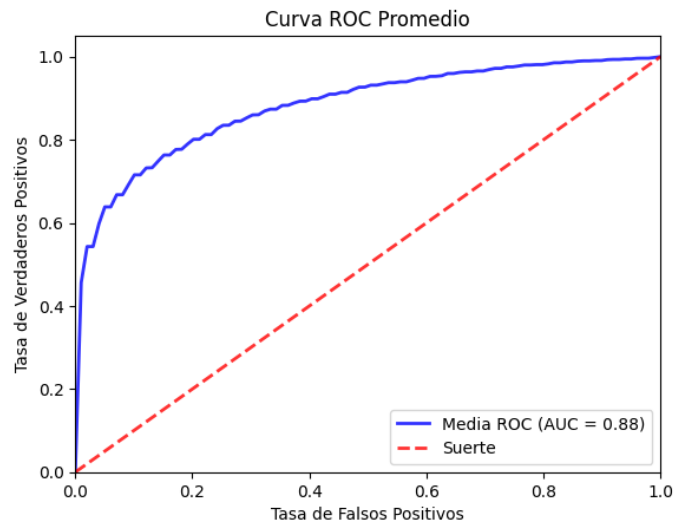
5.2.3.3. Support Vector Machine (SVM)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.8065902140672782	0.7927370030581041
<i>Precisión</i>	0.8109608074900245	0.7929522403045800
<i>Sensibilidad (Recall)</i>	0.7629294217687074	0.7478188775510205
<i>Especificidad</i>	0.8417896174863386	0.8289016393442624
<i>F1 Score</i>	0.7715369848613224	0.7527955845310649

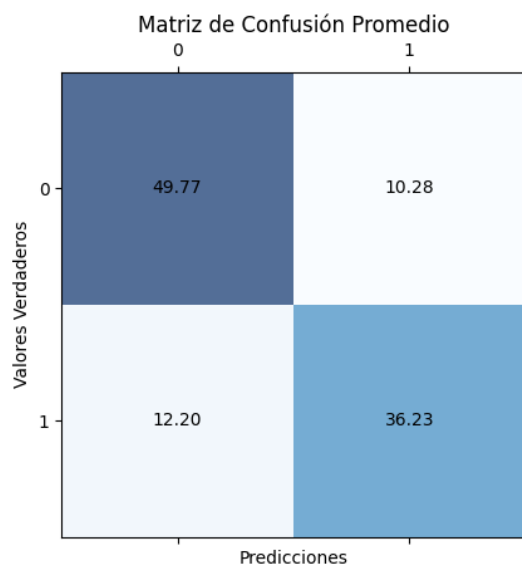
Cuadro 5.17: Resultados de SVM en sistema embebido.



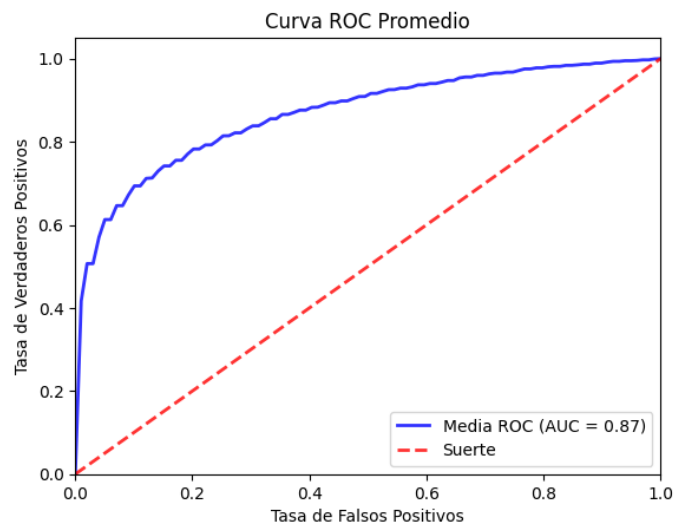
(a) Matriz de confusión de SVM sin PCA.



(b) Curva ROC de SVM sin PCA.



(a) Matriz de confusión de SVM con PCA.

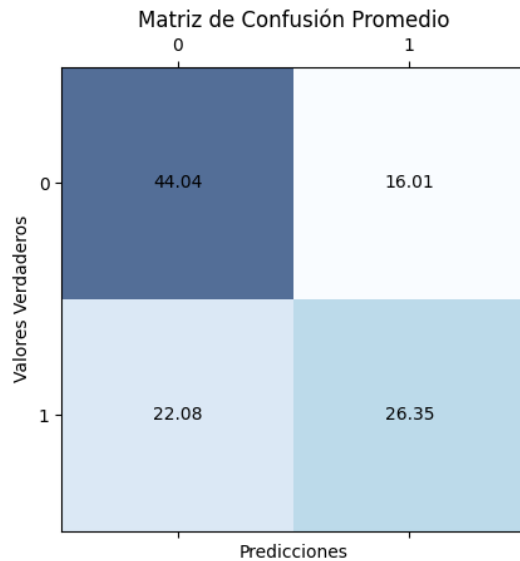


(b) Curva ROC de SVM con PCA.

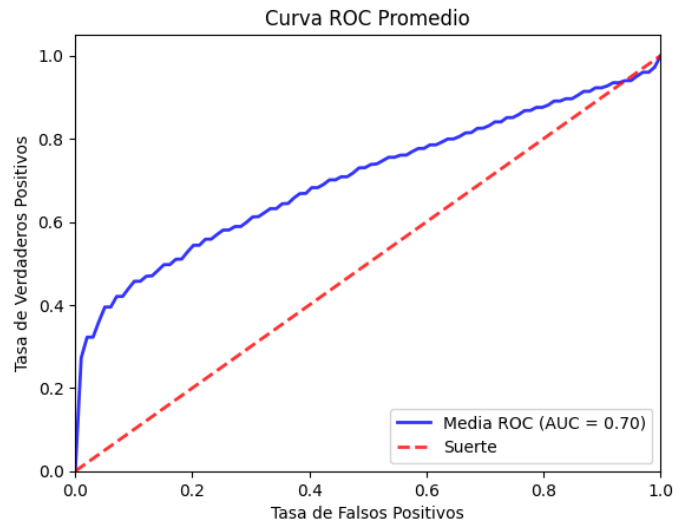
5.2.3.4. Naive Bayes

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6489424057084608	0.6763769962623172
<i>Precisión</i>	0.6348835455919706	0.6750862292922490
<i>Sensibilidad (Recall)</i>	0.5442091836734694	0.5408078231292517
<i>Especificidad</i>	0.7335846994535520	0.7856994535519126
<i>F1 Score</i>	0.5530165641311912	0.5732984691574837

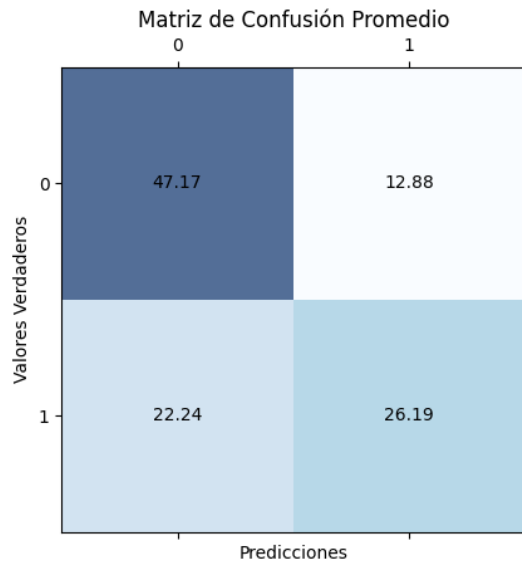
Cuadro 5.18: Resultados Naive Bayes en sistema embebido.



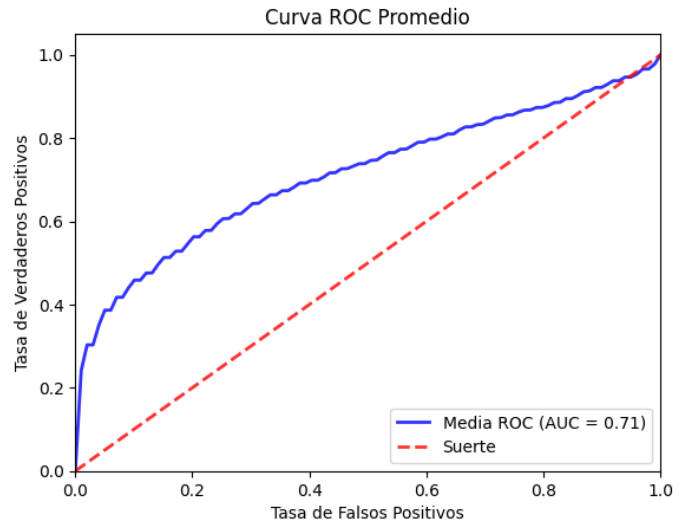
(a) Matriz de confusión de Naive Bayes sin PCA.



(b) Curva ROC de Naive Bayes sin PCA.



(a) Matriz de confusión de Naive Bayes con PCA.

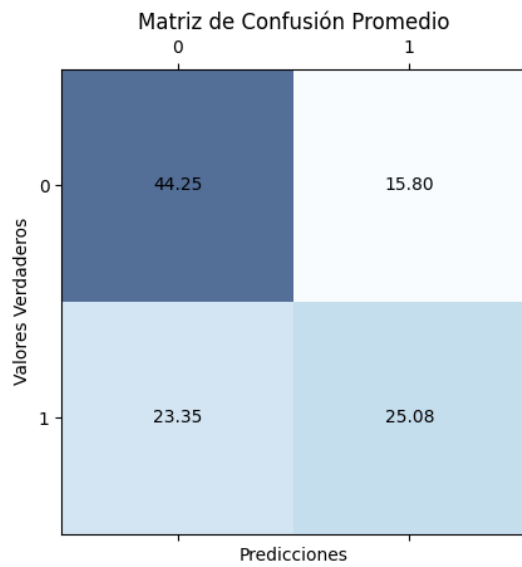


(b) Curva ROC de Naive Bayes con PCA.

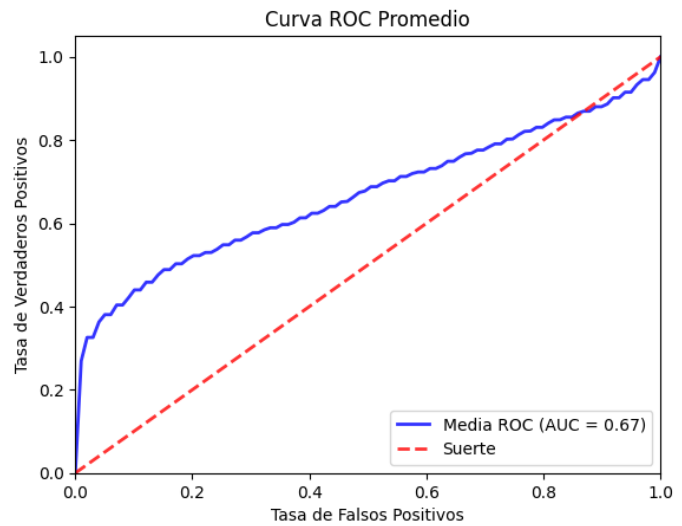
5.2.3.5. Logistic Regression

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6392099898063200	0.6464152225620114
<i>Precisión</i>	0.6132387806648687	0.618744333304205
<i>Sensibilidad (Recall)</i>	0.5182908163265306	0.5190646258503402
<i>Especificidad</i>	0.7370273224043715	0.7493579234972677
<i>F1 Score</i>	0.5314697761543392	0.5363448250691095

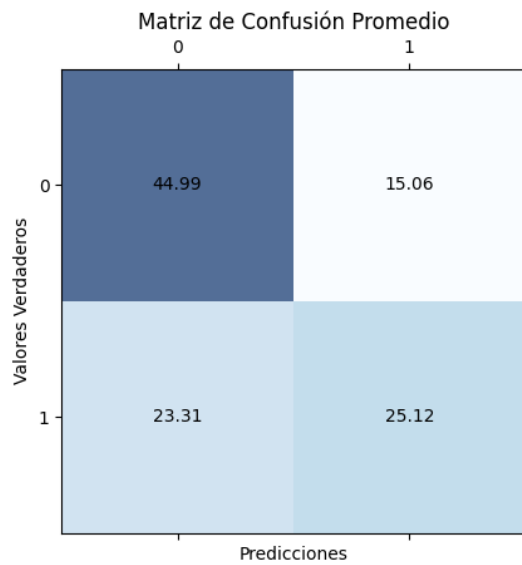
Cuadro 5.19: Resultados de Logistic Regression en sistema embebido.



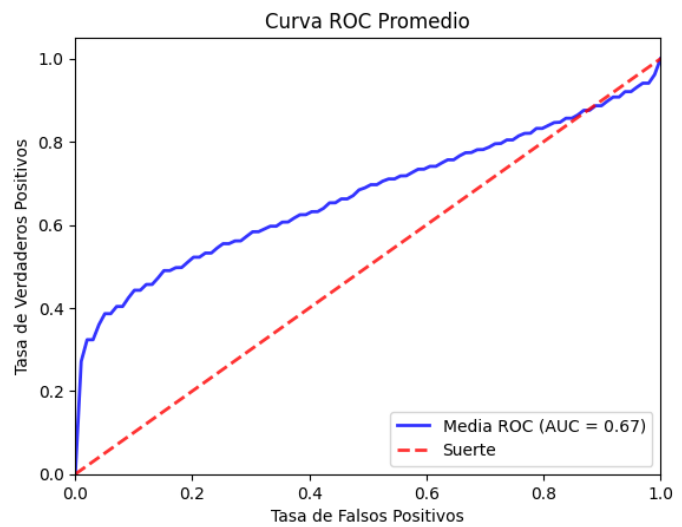
(a) Matriz de confusión de Logistic Regression sin PCA.



(b) Curva ROC de Logistic Regression sin PCA.



(a) Matriz de confusión de Logistic Regression con PCA.

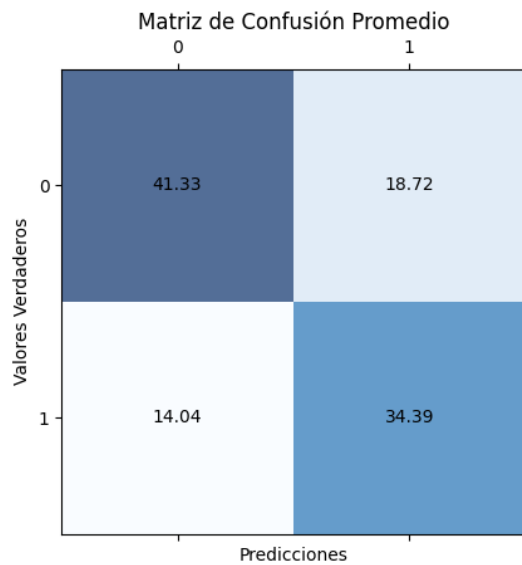


(b) Curva ROC de Logistic Regression con PCA.

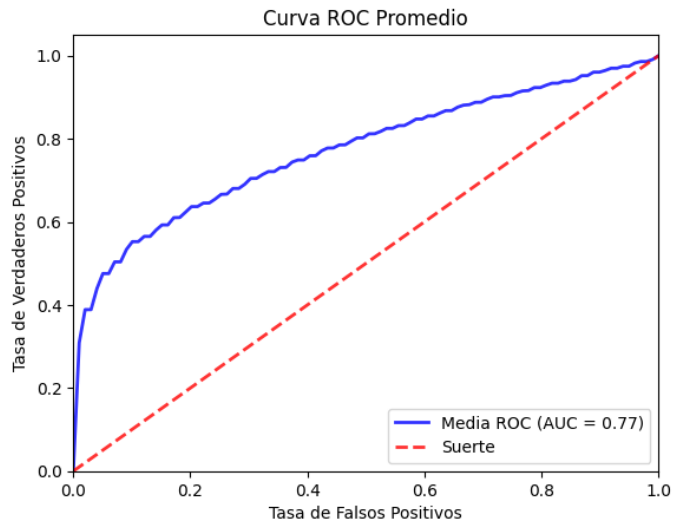
5.2.3.6. Multilayer Perceptron (MLP)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6981099218484540	0.7259089364593951
<i>Precisión</i>	0.6780750067984692	0.7289327923152800
<i>Sensibilidad (Recall)</i>	0.7097406462585033	0.6543877551020407
<i>Especificidad</i>	0.6883825136612022	0.7836311475409836
<i>F1 Score</i>	0.6742101441060372	0.6679300928135344

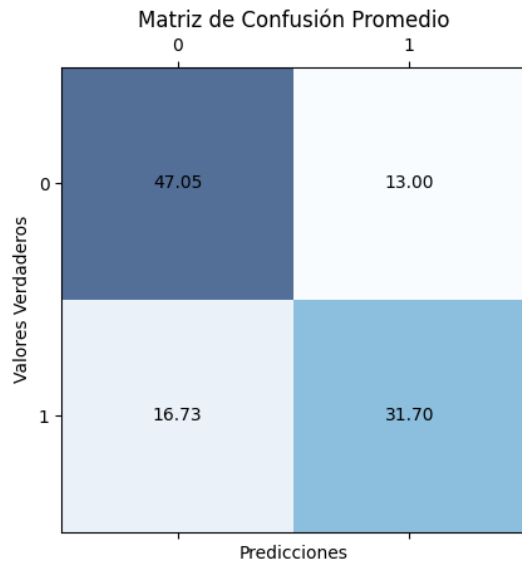
Cuadro 5.20: Resultados de MLP en sistema embebido.



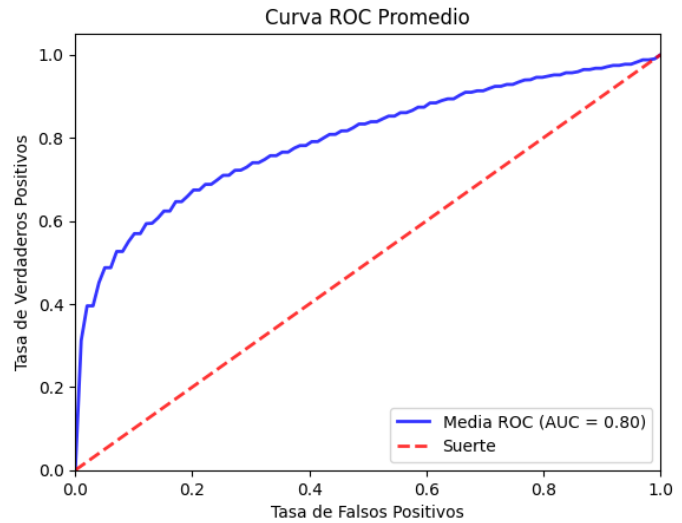
(a) Matriz de confusión de MLP sin PCA.



(b) Curva ROC de MLP sin PCA.



(a) Matriz de confusión de MLP con PCA.

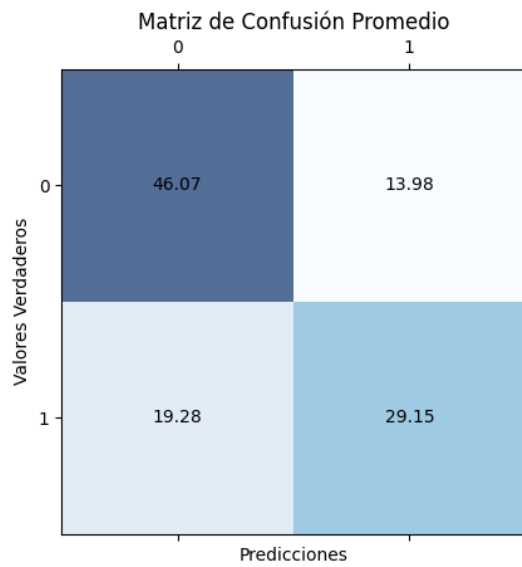


(b) Curva ROC de MLP con PCA.

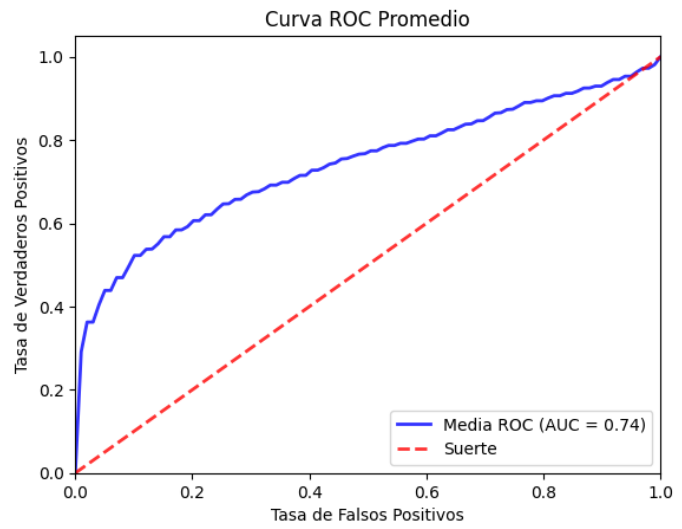
5.2.3.7. LSTM (Long Short Term Memory)

K-Folds = 100		
Métrica promedio	Sin PCA	Con PCA
<i>Exactitud</i>	0.6935074753652737	0.7308486238532110
<i>Precisión</i>	0.6892946034724382	0.7200153260039770
<i>Sensibilidad (Recall)</i>	0.6021386054421769	0.6772151360544219
<i>Especificidad</i>	0.7673633879781423	0.7742978142076504
<i>F1 Score</i>	0.6194652517525924	0.6737325652164441

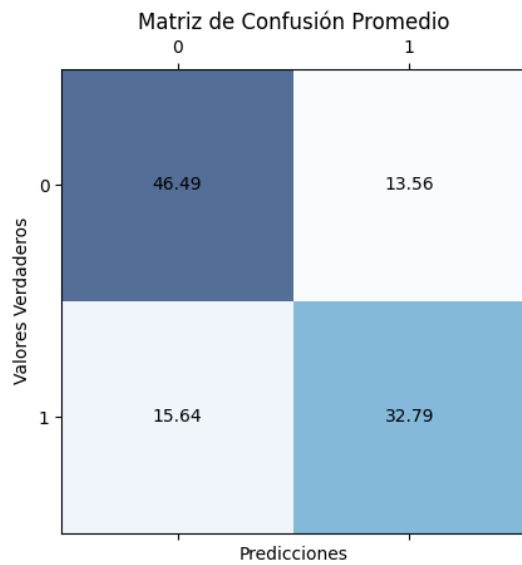
Cuadro 5.21: Resultados de red neuronal LSTM en sistema embebido.



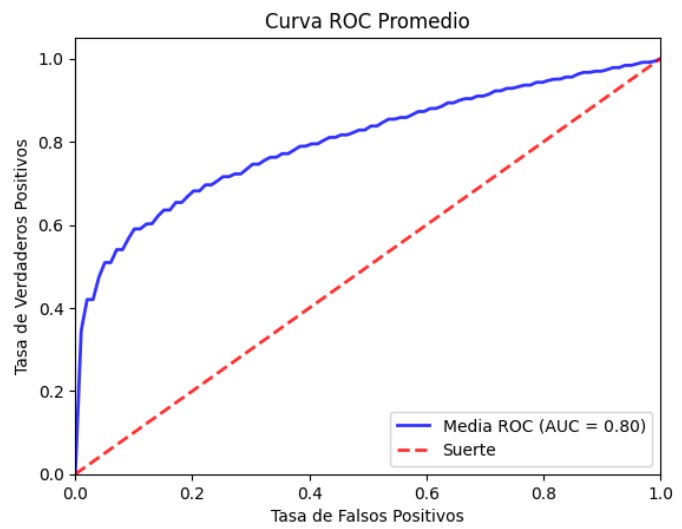
(a) Matriz de confusión de red neuronal LSTM sin PCA.



(b) Curva ROC de red neuronal LSTM sin PCA.



(a) Matriz de confusión de red neuronal LSTM con PCA.



(b) Curva ROC de red neuronal LSTM con PCA.

5.2.4. Comparativa de clasificadores en PC

5.2.4.1. Exactitud

(Exactitud)	PC		Sistema Embebido	
Algoritmo	Sin PCA	Con PCA	Sin PCA	Con PCA
<i>K-Nearest Neighbors</i>	0.8390	0.8034	0.7844	0.7638
<i>Random Forests</i>	0.8192	0.8068	0.7996	0.7651
<i>Support Vector Machine</i>	0.8545	0.8350	0.8065	0.7927
<i>Naive Bayes</i>	0.6552	0.6931	0.6489	0.6763
<i>Logistic Regression</i>	0.6492	0.6570	0.6392	0.6464
<i>Multilayer Perceptron (MLP)</i>	0.6736	0.7183	0.6981	0.7259
<i>Long Short Term Memory</i>	0.6490	0.7237	0.6935	0.7308

Cuadro 5.22: Comparativa entre los resultados truncados a 4 decimales de los clasificadores en PC. Se resalta en color verde el resultado con el mejor resultado, mientras que en rojo, el peor.

Capítulo 6

Conclusiones

En este trabajo de investigación se presentaron los resultados de la primera etapa de desarrollo de un dispositivo BCI portátil, el cual permitirá la adquisición y procesamiento embebido de señales EEG en un contexto académico, fuera del clínico, facilitando el acceso a una herramienta de bajo costo con funcionalidades de análisis de señales (incluyendo técnicas de *Machine Learning*) para obtener información relevante de los procesos cognitivos de los sujetos de estudio. Para el desarrollo del proyecto se propuso implementarlo en un microprocesador ARM de última generación de la marca STMicroelectronics modelo STM32F7, bajo el esquema de un sistema embebido, el cual dispone de los periféricos necesarios para la adquisición de las señales analógicas, además del debido procesamiento digital y la clasificación de las mismas con algoritmos tanto comprimidos como optimizados para su funcionamiento dentro de éste, a pesar de las restricciones de capacidad de cómputo inherentes.

Como resultado, se logró la implementación modular de un sistema incipiente de BCI como preludio de la tarjeta principal del presente trabajo. A través del uso de la tarjeta modular del AFE, fue viable integrar el microprocesador y el *driver* de bajo nivel para la adquisición de las señales de EEG con un electrodo, donde estas últimas lograron procesarse exitosamente dentro del sistema embebido. La iteración de *hardware* del módulo del AFE comprueba que el diseño del PCB de señales mixtas garantiza una integridad de señales EEG admisible para su procesamiento y clasificación; dejando la puerta abierta a la replicación del *layout* del

PCB en la tarjeta final.

Aunado a lo anterior, los resultados de la ejecución de modelos de clasificación, enfocados a señales EEG y optimizados para sistemas embebidos en el microprocesador ESP32, manifiestan porcentajes de clasificación significativamente más bajos en comparación a lo que puede procesar una computadora portátil de gama media, pero, a su vez, siendo más restringidos en el consumo de recursos computacionales, haciéndolos accesibles para su implementación y ejecución paralela a la adquisición de señales. Las redes neuronales resultaron ser más efectivas para su inclusión en el sistema embebido a comparación de otros modelos. Lo ya dicho se puede apreciar en el Cuadro 5.22.

Con base de lo establecido en las hipótesis y en los objetivos, sí fue posible el diseño de un sistema BCI capaz de procesar bioseñales (EEG) bajo la integración de la etapa de adquisición, procesamiento digital de señales y clasificación con *machine learning* en un sistema embebido, teniendo en cuenta que lo expuesto anteriormente se concibió, por cuestiones de tiempo de desarrollo, de forma modular, dejando a un lado uno de los objetivos principales que correspondía a la integración y paralelización de todas las funciones (colocándolo como parte del trabajo a futuro). Incluso los objetivos de creación del *dataset* y clasificación de acuerdo a la condición del trastorno del espectro autista quedan pendientes de corroborarse.

6.1. Limitaciones de la investigación

- Se hicieron pruebas de las lecturas de los electrodos conectados al AFE y funcionan correctamente, pero tiene la limitante que en esta etapa del proyecto todavía no ejecuta, paralelamente, inferencias de algún modelo de clasificación.
- El dispositivo sólo puede enviar información codificada en ASCII UTF-8 (8 canales a la velocidad de muestreo del AFE más baja) a través de comunicación serial a la PC con el software de adquisición que se diseñó en LabVIEW.
- Actualmente se puede experimentar con dos versiones de *firmware* por separado: uno

que se encarga de la adquisición del AFE y otro que ejecuta inferencias de modelos (cuyos vectores de características se ingresan a través de comunicación serial) de clasificación exportados desde EdgeImpulse.

- El *driver* del AFE no está desarrollado conforme a los lineamientos de una HAL (*Hardware Abstraction Layer*), sino emplea directamente funciones de bajo nivel de comunicación SPI, entre otras específicas del microprocesador empleado para la experimentación (ESP32). Lo anterior limita la portabilidad a otras marcas de microprocesadores y legibilidad del código.
- El *firmware* de procesamiento digital de señales (FFT y DWT) no está debidamente adaptado al microprocesador, dado que no se está aprovechando el núcleo de DSP que provee la arquitectura ARM, incurriendo en operaciones aritméticas, trigonométricas y *bitwise* hechas directamente en la ALU (*Arithmetic Logic Unit*) del CPU.
- Dado el contrato de licencias de EdgeImpulse y LabVIEW, el proyecto no está desarrollado en totalidad por licencias de código abierto, mermando por completo la posibilidad de llevar a la venta y distribución la plataforma BCI.
- Tomando en cuenta la escasez de semiconductores a nivel mundial como resultado de la emergencia sanitaria de COVID-19, se optó por llevar a cabo las experimentaciones con el módulo del AFE desarrollado en paralelo a la tarjeta principal del proyecto.

6.2. Trabajo a futuro

En futuras etapas de la propuesta, se prevé robustecer la arquitectura del *firmware*, a fin de facilitar la conexión y detección de más AFE que incrementen la cantidad de canales disponibles. Asimismo, es relevante la inclusión de los *drivers* necesarios para el funcionamiento del puerto microSD que garantice la habilitación de la característica de almacenamiento de las muestras en tiempo real. De igual manera, la comunicación inalámbrica es indispensable para

enviar los datos efectivamente a una computadora sin mermar la comodidad y libertad de movimiento del sujeto, lo cual, concretamente, minimiza la adición de artefactos en la señal muestreada a la par de la ventaja de la disminución de carga computacional al sistema embebido, puesto que el procesamiento y guardado de la información queda resuelto por parte del agente externo; que eventualmente resulta beneficioso en el sentido estricto de creación y diseño de *datasets* correctamente etiquetados que servirán a futuro para el entrenamiento de modelos de clasificación. También es imperativo el cambio de uso de códigos cerrados a aquellos que sean de código abierto; migrar por completo y desarrollar utilidades y herramientas en ecosistemas que sean más apropiados para el mantenimiento a largo plazo. Incluso es necesario pensar en el diseño de un datagrama codificado en hexadecimal, enfocado en apelar a la tasa de muestreo del AFE y la cantidad de canales que se deseen adquirir, evitando por completo la codificación ASCII UTF-8 que entorpece y limita la cantidad máxima de información que se puede enviar. Todo lo anterior es crucial que tenga una arquitectura de tiempo real (RTOS) con la intención de satisfacer la necesidad de distribución de recursos internos del sistema embebido que genere un procesamiento paralelo con una adquisición continua en el fondo y procesamiento de señales en el frente.

Por otro lado, es imperativo contar con un diseño 3D funcional de una diadema y gabinete para la tarjeta electrónica. Esto mejoraría considerablemente la simplicidad del sistema entero a partir de una colocación de electrodos más precisa con la debida fuerza de contacto y consistencia durante toda la toma de la muestra, al igual que la movilidad dejaría de ser un inconveniente con base de una buena sujeción del PCB a la espalda del sujeto.

Finalmente, el siguiente paso a tomar, dentro del paradigma de desarrollo, es darle seguimiento a la construcción de un *dataset* sólido y sustancial de señales EEG de sujetos neurodivergentes en condiciones de evaluaciones cognitivas, con el propósito de etiquetar las muestras con la condición diagnosticada y, de esta manera, entrenar a posteriori los modelos de *machine learning* (preferiblemente redes neuronales) para que provean de un porcentaje de probabilidad la existencia de un trastorno del espectro autista. Es primordial no perder de vista el objetivo principal de la creación de un BCI que procese señales cerebrales en tiempo

real sin depender de computadoras externas, lo que mejora la autonomía y la usabilidad del sistema.

Bibliografía

- Adeli, H., Ghosh-Dastidar, S., & Dadmehr, N. (2007). A wavelet-chaos methodology for analysis of eegs and EEG subbands to detect seizure and epilepsy. *IEEE Transactions on Biomedical Engineering*, 54(2), 205-211. <https://doi.org/10.1109/tbme.2006.886855>
- AllPCB. (2017). The 7 working layers in printed circuit board. https://www.allpcb.com/pcb/pcb_layers.html
- Alumni, S. A. (2022, octubre). Tidying up with PCA: An Introduction to Principal Components Analysis. <https://community.alteryx.com/t5/Data-Science/Tidying-up-with-PCA-An-Introduction-to-Principal-Components/ba-p/382557>
- Amin, H., Mumtaz, W., Subhani, A., Saad, M., & Malik, A. (2017). Classification of EEG signals based on Pattern Recognition Approach. *Frontiers in Computational Neuroscience*, 11. <https://doi.org/10.3389/fncom.2017.00103>
- Ardizzoni, J. (2005, septiembre). A practical guide to high-speed printed-circuit-board layout. <https://www.analog.com/en/resources/analog-dialogue/articles/high-speed-printed-circuit-board-layout.html>
- Bitbrain, E. e. d. (2024, abril). Localización de Electrodo del Eeg: Layout Fijo vs. variable. <https://www.bitbrain.com/es/blog/colocacion-electrodos-eeg>
- Bosl, W., Tager-Flusberg, H., & Nelson, C. (2018). EEG analytics for early detection of autism spectrum disorder: A data-driven approach. *Scientific Reports*, 8(1). <https://doi.org/10.1038/s41598-018-24318-x>

- Bre, F., Gimenez, J., & Fachinotti, V. (2017). Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. *Energy and Buildings*, 158. <https://doi.org/10.1016/j.enbuild.2017.11.045>
- Bularka, S., & Gontean, A. (2015). EEG Pattern Recognition Techniques Review. *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)*. <https://doi.org/10.1109/siitme.2015.7342339>
- Chaste, P., & Leboyer, M. (2012). Autism risk factors: Genes, environment, and gene-environment interactions. *Dialogues in Clinical Neuroscience*, 14(3), 281-292. <https://doi.org/10.31887/dcns.2012.14.3/pchaste>
- Chesser, K., & Porley, M. (s.f.). What are the basic guidelines for layout design of mixed-signal PCBs? <https://www.analog.com/en/resources/analog-dialogue/articles/what-are-the-basic-guidelines-for-layout-design-of-mixed-signal-pcbs.html>
- Chiesi, M., Guermandi, M., Placati, S., Scarselli, E., & Guerrieri, R. (2019). Creamino: A cost-effective, open-source EEG-based BCI system. *IEEE Transactions on Biomedical Engineering*, 66(4), 900-909. <https://doi.org/10.1109/tbme.2018.2863198>
- Connect, E. (2021, julio). Áreas funcionales de la corteza cerebral humana. <https://www.elsevier.com/es-es/connect/areas-funcionales-de-la-corteza-cerebral-humana>
- Djemal, R., AlSharabi, K., Ibrahim, S., & Alsuwailem, A. (2017). EEG-based computer aided diagnosis of autism spectrum disorder using wavelet, entropy, and ann. *BioMed Research International*, 2017, 1-9. <https://doi.org/10.1155/2017/9816591>
- Esqueda-Elizondo, J., Juárez-Ramírez, R., López-Bonilla, O., García-Guerrero, E., Galindo-Aldana, G., Jiménez-Beristáin, L., Serrano-Trujillo, A., Tlelo-Cuautle, E., & Inzunza-González, E. (2022). Attention measurement of an autism spectrum disorder user using EEG signals: A case study. *Mathematical and Computational Applications*, 27(2), 21. <https://doi.org/10.3390/mca27020021>
- Frey, J. (2016). Comparison of an open-hardware electroencephalography amplifier with medical grade device in brain-computer interface applications. *Proceedings of the 3rd*

- International Conference on Physiological Computing Systems*. <https://doi.org/10.5220/0005954501050114>
- Grossi, E., Valbusa, G., & Buscema, M. (2020). Detection of an autism EEG signature from only two EEG channels through features extraction and Advanced Machine Learning Analysis. *Clinical EEG and Neuroscience*, 52(5), 330-337. <https://doi.org/10.1177/1550059420982424>
- Guerrero, G. A. R. (2021, junio). Cross-validation. <https://gladysandrea-rodriguez.medium.com/cross-validation-11e9ea688506>
- Han, H., & Siebert, J. (2022). TinyML: A systematic review and synthesis of existing research. *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*. <https://doi.org/10.1109/icaaic54071.2022.9722636>
- Ibrahim, S., Djemal, R., & Alsuwailem, A. (2018). Electroencephalography (EEG) signal processing for epilepsy and autism spectrum disorder diagnosis. *Biocybernetics and Biomedical Engineering*, 38(1), 16-26. <https://doi.org/10.1016/j.bbe.2017.08.006>
- Novo-Olivas, C., Guitiérrez, L., & Bribiesca, J. (2010, febrero). Mapeo Electroencefalográfico y Neurofeedback.
- Office, U. G. A. (2023). Helping youth with autism. <https://www.gao.gov/blog/2017/04/18/helping-youth-with-autism>
- OpenBCI. (2023). *Open Source Tools for Neuroscience*. <https://openbci.com/>
- Peng, S., Xu, R., Yi, X., Hu, X., Liu, L., & Liu, L. (2021). Early screening of children with autism spectrum disorder based on electroencephalogram signal feature selection with L1-norm regularization. *Frontiers in Human Neuroscience*, 15. <https://doi.org/10.3389/fnhum.2021.656578>
- Pradas, C., & Vicente, M. (2022, julio). Test de inteligencia - test de IQ gratis, online y fiable. <https://www.psicologia-online.com/test-de-inteligencia-3317.html>
- Pramoditha, R. (2022, febrero). Overview of a neural network's learning process. <https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa>

- Press, W. H. (2009). Wavelet Transforms. En *Numerical recipes in C: The art of scientific computing*. Cambridge University Press.
- Ray, P. (2022). A review on tinyml: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences*, 34(4), 1595-1623. <https://doi.org/10.1016/j.jksuci.2021.11.019>
- Services, A. W. (2023). *Market leading RTOS (real time operating system) for embedded systems with internet of things extensions*. <https://www.freertos.org/>
- Shih, J. J., Krusienski, D. J., & Wolpaw, J. R. (2012, marzo). Brain-computer interfaces in medicine. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3497935/>
- STMicroelectronics. (2023). *AI expansion pack for STM32CubeMX (X-CUBE-AI)*. <https://www.st.com/en/embedded-software/x-cube-ai.html>
- Tan, L. (2008). *Digital Signal Processing: Fundamentals and Applications*. Elsevier/Academic Press.
- Tawhid, M., Siuly, S., & Wang, H. (2020). Diagnosis of autism spectrum disorder from EEG using a time–Frequency Spectrogram image-based approach. *Electronics Letters*, 56(25), 1372-1375. <https://doi.org/10.1049/el.2020.2646>
- World Health Organization (WHO). (2022). *Autism* [Accessed: 30 march 2022]. <https://www.who.int/es/news-room/fact-sheets/detail/autism-spectrum-disorders>
- Zeidan, J., Fombonne, E., Scoriah, J., Ibrahim, A., Durkin, M., Saxena, S., Yusuf, A., Shih, A., & Elsabbagh, M. (2022). Global prevalence of autism: A systematic review update. *Autism Research*, 15(5), 778-790. <https://doi.org/10.1002/aur.2696>