



INTERFAZ CEREBRO COMPUTADORA CON SEÑALES DE INFRARROJO CERCANO

Carlos José Mora
202043476

ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA - 3744
FACULTAD DE INGENIERÍA
UNIVERSIDAD DEL VALLE

SANTIAGO DE CALI
2026

INTERFAZ CEREBRO COMPUTADORA CON SEÑALES DE INFRARROJO CERCANO

Carlos José Mora
202043476

TRABAJO PRESENTADO PARA OPTAR
POR EL TÍTULO DE INGENIERO
ELECTRÓNICO

DIRECTOR
HUMBERTO LOAIZA CORREA, Ph.D.

Grupo de Investigación Percepción y Sistemas Inteligentes

ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA - 3744
FACULTAD DE INGENIERÍA
UNIVERSIDAD DEL VALLE

SANTIAGO DE CALI
2026

Tabla de contenido

Nota de Aceptación ..... ***¡Error! Marcador no definido.***

RESUMEN I

ABSTRACT I

1. INTRODUCCIÓN 1

1.1 CONTEXTUALIZACIÓN 1
1.2 PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS 2
1.2.1 Planteamiento del problema. *2*
1.2.2 Objetivo General. *2*
1.2.3 Objetivos Específicos *2*
1.3 DESCRIPCIÓN DEL TRABAJO 2
1.4 ORGANIZACIÓN DE LOS CAPÍTULOS 3

2. ANTECEDENTES Y MARCO TEÓRICO 3

2.1 ANTECEDENTES 4
2.1.1 Antecedentes nacionales *4*
2.1.2 Antecedentes internacionales *6*
2.1.3 Discusión de antecedentes *9*
2.2 MARCO TEÓRICO 10
2.2.1 Espectroscopia funcional de infrarrojo cercano (fNIRS) *11*
2.2.2 Ley de Beer-Lambert modificada (MBLL) *11*
2.2.3 Imaginación motora en interfaces cerebro–computadora *12*
2.2.4 Procesamiento digital de señales fNIRS *12*
2.2.5 Extracción de características en señales fNIRS *13*
2.2.6 Aprendizaje automático en BCIs fNIRS *13*
2.2.7 Modelos de clasificación en BCIs basadas en fNIRS *14*
2.2.8 Clasificación selectiva y calibración de probabilidades *15*
2.2.9 Métricas de evaluación en sistemas BCI *15*
2.2.10 Control de avatares en entornos virtuales *16*

3. INSTRUMENTACIÓN ORIGINAL DEL CONJUNTO DE DATOS 16

3.1 INTRODUCCIÓN 16
3.2 SELECCIÓN DE LA BASE DE DATOS 16
3.3 MÓDULO DE ADQUISICIÓN FNIRS DEL CONJUNTO DE DATOS 18
3.3.1 Paradigma experimental de adquisición *18*
3.4 PLATAFORMA DE CÓMPUTO PARA EL PROCESAMIENTO DE DATOS 19
3.5 CONCLUSIONES DEL CAPÍTULO 20

4. METODOLOGÍA DE LA SOLUCIÓN PROPUESTA: SOFTWARE 20

4.1 Introducción 20
4.1.1 Caracterización de la base de datos y estructuras 21
4.1.2 Organización de los archivos y estructura de canales 22
4.2 Etapa de visualización 22
4.2.1 Verificación de la consistencia de canales y etiquetas 23
4.2.2 Visualización de señales crudas por segmentos motores 24
4.2.3 Confirmación de longitudes de onda y frecuencia de muestreo 25
4.3 Conversión MBLL y fusión de sesiones de imaginación motora 26
4.4 Corrección de artefactos en el dominio hemodinámico 29
4.5 Filtrado pasa–banda hemodinámico 31
4.6 Segmentación de épocas y construcción de ventanas de tarea/reposo 33
4.7 Extracción y selección de características 35
4.8 Comité de clasificadores y Stacking con clasificación selectiva (rechazo) y calibración 41
4.8.1 División del conjunto de datos: 15 splits estratificados y evaluación hold-out 42
4.8.2 Validación cruzada anidada: selección conjunta de ventana temporal y canales 43
4.8.3 Entrenamiento de expertos: modelos base y justificación de hiperparámetros 44
4.8.4 Probabilidades out-of-fold (OOF) y construcción del espacio de stacking 50
4.9 Generación de comandos para el avatar y exportación de resultados 54
4.10 Resumen del capítulo 57

5. PRUEBAS Y RESULTADOS 58

5.1 Introducción 58
5.2 Descripción general del conjunto de datos y configuración experimental 58
5.3 Resultados de clasificación sin rechazo 59
5.4 Impacto del rechazo 63
5.4.1 Análisis intra-sujeto 65
5.4.2 Resultados globales del sistema con rechazo (abstención) 66
5.5 Análisis comparativo y general del sistema 70
5.6 Validación funcional con el avatar en Unity 71
5.6.1 Descripción de escenas y máquina de estados 72
5.7 Alcances y limitaciones de los resultados 73
5.7.1 Alcances 73
5.7.2 Limitaciones 74
5.7.3 Limitaciones 75

6. CONCLUSIONES GENERALES 76

7. TRABAJOS FUTUROS 78

7.1 Preprocesamiento y modelado hemodinámico 78
7.2 Paradigma experimental, ampliación de clases y uso de aprendizaje profundo 78
7.3 Implementación en tiempo real y evaluación con usuarios 78
7.4 Optimización computacional y despliegue 78

8. REFERENCIAS BIBLIOGRÁFICAS 79

9. ANEXOS 82

LISTA DE FIGURAS

FIGURA 4-1. DIAGRAMA DE BLOQUES DEL PIPELINE SOFTWARE PARA LA BCI FNIRS PROPUESTA. 21
FIGURA 4-2. DIAGRAMA DE FLUJO PARA LA EXTRACCIÓN DE ETIQUETAS DE LOS CANALES A USAR 23
FIGURA 4-3. DIAGRAMA DE FLUJO PARA LA VISUALIZACIÓN DE SEÑALES FNIRS CRUDAS EN TAREAS DE IMAGINACIÓN MOTORA. 24
FIGURA 4-4. DIAGRAMA DE FLUJO PARA LA VERIFICACIÓN DE WAVELENGTHS. 25
FIGURA 4-5. DIAGRAMA DE FLUJO DEL ALGORITMO DE MBLL. 27
FIGURA 4-6. DIAGRAMA DE FLUJO DEL ALGORITMO DE ELIMINACIÓN DE ARTEFACTOS. 29
FIGURA 4-7. DIAGRAMA DE FLUJO DEL ALGORITMO DE FILTRADO CHEBYSHEV 32
FIGURA 4-8. DIAGRAMA DE FLUJO DEL ALGORITMO DE SEGMENTACIÓN 34
FIGURA 4-9. ANÁLISIS DE EXACTITUD DE VENTANAS GLOBALES PARA TODOS LOS SUJETOS. 37
FIGURA 4-10. DIAGRAMA DE VIOLÍN DE CARACTERÍSTICAS GLOBALES PARA TODO EL DATASET 41
FIGURA 4-11. DIAGRAMA DE FLUJO DEL COMITÉ DE CLASIFICADORES. 42
FIGURA 4-12. DIAGRAMA DE FLUJO DE GENERACIÓN Y LIMPIEZA DE COMANDOS PARA UNITY. 55
FIGURA 4-13. DIAGRAMA DE FLUJO PARA EL CONTROL DEL AVATAR DENTRO DE UNITY 56
FIGURA 4-14. MONTAJE DEL ENTORNO DEL AVATAR DENTRO DE UNITY. 57
FIGURA 5-1. MATRIZ DE CONFUSIÓN SIN RECHAZO GLOBAL. 63
FIGURA 5-2. MÉTRICAS GLOBALES DE LA BCI – ANÁLISIS DE DESEMPEÑO. 67
FIGURA 5-3, MÉTRICAS GLOBALES DE LA BCI – ANÁLISIS DEL META-MODELO 68
FIGURA 5-4. EJECUCIÓN DEL AVATAR EN UNITY. 72
FIGURA 5-5. MÁQUINA DE ESTADOS DEL ANIMATOR (IDLE–RIGHT–LEFT–ERROR). 73

FIGURA A2. 1. SEÑALES CRUDAS PRIMER SEGMENTO – SUJETO 1. 82

FIGURA A3. 1. FUSIÓN DE TRES SESIONES MI Y CONVERSIÓN MBLL (ΔHBO/ΔHBR) – SUJETO 1 83

FIGURA A4. 1. SEÑAL CON ARTEFACTOS ELIMINADOS – SUJETO 1 83
FIGURA A4. 2. RELACIÓN ARTEFACTOS VS. TIEMPO – SUJETO 1 83

FIGURA A5. 1. SEÑAL FILTRADA – SUJETO 1 – FILTRO CHEBYSHEV II 84

FIGURA A6. 1. ÉPOCA SEGMENTADA - LMI - REPOSO - SUJETO 1. 84

FIGURA A7. 1. DIAGRAMA DE VIOLÍN DE CARACTERÍSTICAS – SUJETO 1 (VENTANA C). 85

FIGURA A8. 1. MATRIZ DE CONFUSIÓN SIN RECHAZO – SUJETO 1. 85

FIGURA A9. 1. PANEL CON RECHAZO – SUJETO 1 86

LISTA DE TABLAS

TABLA 2-1.COMPARATIVA DE ANTECEDENTES CONSULTADOS. 9
TABLA 3-1. COMPARACIÓN DE LA BASE DE DATOS FNIRS PARA BCI. 17
TABLA 4-1. ETIQUETAS DE LOS 36 CANALES FNIRS (CLAB) DEL MONTAJE NIRSCOUT UTILIZADO EN EL CONJUNTO DE DATOS. 24
TABLA 4-2. VERIFICACIÓN DE WAVELENGTHS 26
TABLA 4-3. ESQUEMA DE VALIDACIÓN Y PARTICIONAMIENTO (LO DE "15 SPLITS", NESTED-CV Y HOLD-OUT). 44
TABLA 4-4. EXPERTOS DEL COMITÉ, PIPELINES Y CONFIGURACIÓN DE HIPERPARÁMETROS. 49
TABLA 4-5. MÉTRICAS GUARDADAS DEL MODELO. 53
TABLA 5-1, EXACTITUD SIN RECHAZO POR SUETO..... 60
TABLA 5-2. EXACTITUD Y MÉTRICAS CON RECHAZO POR SUJETO. 64
TABLA 5-3. COMPARACIÓN GLOBAL DEL SISTEMA SEGÚN CRITERIO DE SELECCIÓN DEL SPLIT (28 SUJETOS) 70
TABLA 5-4. COMPARACIÓN DE DESEMPEÑOS RESPECTO A LOS ANTECEDENTES. 71

AGRADECIMIENTOS

Agradezco al profesor Humberto Loaiza Correa, director de este trabajo de grado, por su orientación, acompañamiento y rigor académico durante todo el proceso. Su guía y retroalimentación fueron fundamentales para consolidar el enfoque metodológico y fortalecer la calidad del desarrollo realizado.

A mi madre, Ligia Mora, a mi abuela, María V. Mora y a mi hermana Alisson Juliana Mora, les expreso mi gratitud por su apoyo incondicional, sus palabras de aliento y su presencia constante. Su respaldo fue determinante para mantener la disciplina y la motivación necesarias para culminar esta etapa.

Agradezco también a mis compañeros de carrera, por el compañerismo y las jornadas de estudio compartidas, que enriquecieron mi formación. Asimismo, agradezco al director del programa, Andrés David Restrepo, por su receptividad y apoyo institucional.

Finalmente, extiendo mi agradecimiento a todas las personas que contribuyeron directa o indirectamente al desarrollo de este trabajo. Gracias por toda la confianza que se ha depositado en mí para el desarrollo de este trabajo de grado.

Resumen

Este trabajo de grado desarrolla y evalúa una interfaz cerebro–computadora (BCI) unimodal e individual por sujeto, basada en espectroscopia funcional de infrarrojo cercano (fNIRS) para discriminar imaginación motora de mano izquierda y mano derecha (LMI vs. RMI) usando el conjunto de datos abierto de la TU Berlin (28 sujetos). Se implementó en Python un pipeline reproducible por sujeto: verificación de canales y longitudes de onda, conversión de intensidades a concentraciones hemodinámicas mediante la Ley de Beer–Lambert modificada (MBLL), corrección explícita de artefactos de movimiento, filtrado pasa–banda hemodinámico y segmentación de épocas alrededor del inicio de cada ensayo. A partir de ventanas de tarea se extrajeron características temporales de ΔHbO y ΔHbR (incluyendo el contraste $\Delta HbO - \Delta HbR$) y se aplicó selección de canales y variables para controlar la dimensionalidad. La decisión final se realizó con un comité de clasificadores y *stacking* basados en aprendizaje superficial debido a la pequeña dimensionalidad del sistema; además incorporando calibración probabilística y una opción de rechazo basada en confianza para abstenerse en ensayos ambiguos. El desempeño se reporta con métricas de exactitud, balanced accuracy y riesgo–cobertura. De manera global en prueba, se alcanza Accuracy = 70.5%; la exactitud condicionada a aceptados aumenta a 75% y Balanced Accuracy = 75.9%, mientras que el desempeño interno en entrenamiento de manera global (validación cruzada sobre el conjunto de entrenamiento) presenta una Accuracy ≈ 73.7% ± 9.3% de desviación estándar. Estos resultados se encuentran dentro del rango de desempeño reportado en la literatura para sistemas BCI basados en fNIRS en paradigmas de imaginación motora, donde las exactitudes típicas suelen situarse aproximadamente entre 70% y 90% dependiendo del número de canales, características empleadas y clasificador utilizado.

Para la validación funcional, las predicciones se exportaron como comandos discretos y se reprodujeron offline en un entorno Unity con un avatar llevando a cabo movimientos de mano derecha e izquierda según la orden, demostrando la trazabilidad desde la señal hemodinámica hasta una acción observable.

Palabras Clave: BCI; fNIRS; imaginación motora; $\Delta HbO/\Delta HbR$; MBLL; corrección de artefactos; selección de canales (r^2 firmado); stacking ensemble; calibración de probabilidades; clasificación con rechazo; Unity.

Abstract

This thesis develops and evaluates a unimodal functional near-infrared spectroscopy (fNIRS) brain–computer interface (BCI) to discriminate left- vs. right-hand motor imagery (LMI vs. RMI) using the open TU Berlin dataset (28 subjects). A subject-wise Python pipeline was implemented: channel/wavelength

verification, conversion from light intensity to hemodynamic concentration changes via the Modified Beer–Lambert Law (MBLL), explicit motion-artifact correction, hemodynamic band-pass filtering, and epoching around trial onsets. From task windows, temporal features were extracted from ΔHbO and ΔHbR (including the ΔHbO−ΔHbR contrast), followed by channel/feature selection to control dimensionality. Final decisions were produced by an ensemble with stacking, including probability calibration and a confidence-based reject option to abstain on ambiguous trials. Performance is reported with accuracy, balanced accuracy, and risk–coverage metrics. Overall, in testing, Accuracy = 0.705± 0.090 is achieved; conditional accuracy on accepted values increases to 0.750 ± 0.069 and Balanced Accuracy = 0.759 ± 0.071, while internal performance in training (cross-validation on the training set) presents an average Accuracy≈ 0.737 ± 0.053. As a functional validation, predictions were exported as discrete commands and replayed offline in a Unity environment controlling a humanoid avatar, demonstrating an end-to-end link from hemodynamic signals to observable actions.

Keywords: BCI; fNIRS; motor imagery; ΔHbO/ΔHbR; MBLL; artifact correction; channel selection (signed r^2); stacking ensemble; probability calibration; classification with rejection; Unity.

1. Introducción

En este trabajo se desarrolló una interfaz cerebro-computadora (BCI) unimodal basada en espectroscopía funcional de infrarrojo cercano (fNIRS) que permite decodificar, a partir de la hemodinámica cerebral, la instrucción de imaginar la apertura y/o cierre de la mano izquierda o derecha y traducirla en un comando ejecutable de movimiento de mano derecha/izquierda de un avatar dentro de un entorno virtual de manera offline. El eje central del estudio fue el diseño de un flujo completo de procesamiento desde la conversión de los datos crudos hasta la clasificación empleando un conjunto de señales de acceso público y técnicas de aprendizaje supervisado; lo anterior, combina técnicas de procesamiento digital de señales, selección sistemática de canales y estrategias modernas de aprendizaje automático basadas en comités y rechazo adaptativo.

1.1 Contextualización

La parálisis total o parcial de las extremidades superiores producida por enfermedades neurodegenerativas (esclerosis lateral amiotrófica) o lesiones medulares traumáticas afecta gravemente la autonomía de las personas. La Organización Mundial de la Salud estima que el 15 % de la población mundial vive con algún tipo de discapacidad motora [1]; en Colombia la cifra alcanza el 6,3 % según el DANE [2]. Esta pérdida funcional dificulta actividades elementales vestirse, alimentarse, comunicarse y conlleva consecuencias psicológicas y sociales que agravan la exclusión. Las tecnologías de asistencia, y en particular las BCI, se plantean como una vía para restituir parte de la independencia perdida.

Las BCI tradicionales se basan principalmente en electroencefalografía (EEG), la cual presenta limitaciones como baja resolución espacial y alta susceptibilidad al ruido [39], [40]. Adicionalmente, requiere preparación dérmica mediante el uso de electrodos y gel conductor, lo que dificulta su implementación en entornos reales. En contraste, la fNIRS ofrece portabilidad, bajo costo relativo y menor sensibilidad a artefactos electromagnéticos. Midiendo los cambios en oxi- y desoxi-hemoglobina mediante luz de 760 / 850 nm, la fNIRS monitoriza la activación cortical sin procedimientos invasivos y permite al usuario interactuar en entornos más naturales. No obstante, los sistemas basados en fNIRS adolecen todavía de rutas de procesamiento poco estandarizadas, selección subóptima de características y escasa validación inter-sujeto. Este trabajo de grado aborda esas falencias proponiendo un pipeline robusto, cuantificando su rendimiento y sentando bases para una futura implementación en tiempo real.

1.2 Planteamiento del Problema y Objetivos

1.2.1 Planteamiento del problema.

La pregunta que guía el proyecto es: ¿Cómo decodificar instrucciones a partir de señales fNIRS, mediante técnicas de procesamiento de señales y reconocimiento de patrones en una interfaz cerebro-computadora para la realización de acciones ejecutables en un dispositivo virtual? Para responderla resulta imprescindible (i) contar con un conjunto de datos fNIRS confiable, (ii) diseñar métodos de preprocesamiento que mitiguen artefactos fisiológicos, (iii) extraer y seleccionar características discriminativas y (iv) evaluar clasificadores capaces de generalizar entre sujetos.

1.2.2 Objetivo General

Implementar una BCI basada en fNIRS que interprete la imaginación motora de mano izquierda y derecha y ejecute el comando correspondiente en un entorno virtual, utilizando técnicas de procesamiento digital de señales y aprendizaje automático.

1.2.3 Objetivos Específicos

- Seleccionar una base de datos fNIRS de acceso libre obtenida en una institución académica.
- Desarrollar técnicas de preprocesamiento para eliminar artefactos y ruido.
- Identificar y extraer las características que mejor diferencien las instrucciones motoras.
- Implementar un algoritmo de clasificación empleando un modelo superficial o uno profundo.
- Evaluar el desempeño del sistema mediante métricas de clasificación e interpretar los resultados.

1.3 Descripción del trabajo

El desarrollo se estructuró en cinco etapas. Primero, se seleccionó y depuró la base de datos EEG–fNIRS de la TU Berlin: se trabajó con 28 sujetos, excluyendo uno por falta de archivos de artefactos, y se usaron únicamente las tres sesiones de imaginación motora (mano izquierda vs. mano derecha) [3].

En la segunda etapa, las intensidades crudas fNIRS (cnt) se convirtieron a señales hemodinámicas ΔHbO/ΔHbR mediante MBLL. Luego se proyectaron y corrigieron los artefactos de movimiento usando cnt_artifact/mrk_artifact, se aplicó un filtrado pasa–banda hemodinámico Chebyshev II (0,01–0,09 Hz), se fusionaron las tres

sesiones MI y finalmente se realizó la segmentación en épocas (−10 a +25 s) y la normalización por línea base [3].

En la tercera etapa se extrajeron características temporales de HbO y HbR (media, pendiente, varianza, máximos y diferencias HbO–HbR) en ventanas de tarea y reposo, y se aplicó selección de canales basada en r^2 firmado, complementada con métodos embebidos de selección de características (p. ej., RFECV, Boruta, SelectFromModel) para reducir dimensionalidad y sobreajuste [4]–[8].

En la cuarta etapa se entrenó, por sujeto, un ensemble con stacking: expertos (LDA con bagging balanceado, SVM, Random Forest, XGBoost y MLP) combinados con un meta-clasificador logístico calibrado. Sobre sus probabilidades se incorporó clasificación selectiva con rechazo mediante un umbral óptimo por sujeto, asignando estado idle cuando la confianza es insuficiente, con el fin de mejorar robustez en condiciones realistas [9]–[14].

Finalmente, en la quinta etapa se integró la salida del decodificador con un avatar 3D en Unity, generando archivos CSV con comandos discretos (idle, mano derecha, mano izquierda y error) interpretados por un script en C#, permitiendo una validación funcional offline de la interacción BCI–entorno virtual basada en señales fNIRS [15]–[18].

1.4 Organización de los capítulos

El documento se estructura en siete capítulos orientados a la identificación de tareas de imaginación motora (MI) a partir de señales fNIRS (HbO/HbR) mediante procesamiento digital de señales y aprendizaje automático supervisado. Incluye Resumen/Abstract al inicio y Referencias y Anexos al final.

El documento se organiza así: el Capítulo 1 presenta el contexto, la motivación y los objetivos; el Capítulo 2 resume antecedentes y conceptos necesarios (fNIRS, MI y BCI) de forma acotada al alcance del trabajo; el Capítulo 3 describe la instrumentación original del conjunto de datos; el Capítulo 4 detalla el pipeline de software y el esquema de clasificación; el Capítulo 5 reporta las pruebas y resultados; el Capítulo 6 presenta conclusiones; y el Capítulo 7 plantea trabajos futuros. Finalmente, se presentan las referencias bibliográficas y los anexos.

2. Antecedentes y Marco Teórico

Para la identificación de antecedentes se consultaron las bases de datos *IEEE Xplore, Scopus, ACM Digital Library, ScienceDirect* y repositorios institucionales nacionales, considerando principalmente publicaciones entre 2013 y 2025. La búsqueda se realizó empleando combinaciones de palabras clave como *“functional*

near-infrared spectroscopy (fNIRS)”, *“brain–computer interface (BCI)”*, *“motor imagery”*, *“EEG–fNIRS hybrid BCI”* y *“virtual reality / Unity”*, tanto en inglés como en español, priorizando trabajos directamente relacionados con BCIs de imaginación motora y sistemas híbridos EEG–fNIRS.

2.1 Antecedentes

En este capítulo se presentan los antecedentes y conceptos teóricos más relevantes para el desarrollo de una interfaz cerebro–computadora (BCI) basada en señales de espectroscopia funcional de infrarrojo cercano (fNIRS) orientada a la clasificación de imaginación motora de mano izquierda y derecha. La revisión se centra principalmente en trabajos publicados en los últimos años, con énfasis en aquellos que emplean paradigmas de MI, fNIRS o sistemas híbridos EEG–fNIRS, así como en contribuciones nacionales en interfaces cerebro–computadora para rehabilitación y asistencia, con el fin de mantener la vigencia de las metodologías y resultados reportados en la literatura especializada.

2.1.1 Antecedentes nacionales

En Colombia se han reportado diversos desarrollos en interfaces cerebro–computadora no invasivas, principalmente orientados a paradigmas de imaginación motora y potenciales evocados, con aplicaciones en rehabilitación motora y asistencia para personas con discapacidad.

Por ejemplo, en [19] se diseñó uno de los primeros prototipos de interfaz cerebro–computadora reportados en el país. El sistema se basó en señales EEG registradas mediante un sistema multicanal y empleó paradigmas de ritmos sensorimotores (SMR) y potenciales evocados visuales como estímulos para generar comandos de control. El sistema fue evaluado con usuarios sanos en condiciones de laboratorio, utilizando estímulos visuales presentados en pantalla para inducir la actividad cerebral asociada a cada comando. Las señales EEG fueron procesadas mediante extracción de características espectrales y clasificadores lineales, permitiendo el control de un dispositivo virtual en tiempo real. Los resultados demostraron la viabilidad de implementar BCIs no invasivas en entornos académicos colombianos, aun utilizando infraestructura experimental limitada.

Posteriormente, diversos trabajos de grado y artículos han explorado BCIs basadas en EEG para rehabilitación de miembro superior, combinando paradigmas de MI con control de exoesqueletos o brazos robóticos. Posteriormente, diversos trabajos de grado han explorado BCIs basadas en EEG para rehabilitación de miembro superior. En [20] se diseñó una BCI unimodal basada en EEG para la interacción con un sistema de rehabilitación de miembro superior. El sistema utilizó paradigmas de imaginación motora guiados mediante estímulos visuales, en los cuales los

participantes debían imaginar movimientos del brazo durante intervalos definidos de tarea y reposo. Las señales fueron registradas mediante electrodos EEG ubicados sobre regiones sensorimotores, y se extrajeron características de potencia espectral en las bandas μ (8–12 Hz) y β (13–30 Hz). La clasificación de las intenciones de movimiento se realizó mediante modelos lineales, permitiendo discriminar entre diferentes estados de actividad motora. Los experimentos se realizaron con usuarios sanos en condiciones controladas, reportando tasas de clasificación intra-sujeto cercanas al 70 %.

De forma similar, en [21] implementaron una interfaz BCI para la detección de intención de movimiento en rehabilitación de miembro superior, basada en señales EEG unimodales. El sistema empleó estímulos visuales para inducir tareas de imaginación motora, registrando la actividad cerebral mediante un sistema EEG multicanal ubicado sobre regiones sensorimotores. Las señales fueron procesadas mediante técnicas de extracción de características espectrales y clasificadores supervisados, permitiendo detectar la intención de movimiento del usuario. El sistema fue integrado con un robot colaborativo para tareas de rehabilitación, evaluándose con usuarios en condiciones de laboratorio y reportando tasas de acierto intra-sujeto superiores al 70 %.

Otros trabajos nacionales se han centrado en la clasificación de patrones de imaginación motora en BCIs basadas en EEG. En [22] se desarrolló una BCI para la clasificación de patrones de imaginación motora utilizando señales EEG registradas mediante un sistema multicanal sobre regiones sensorimotores. El paradigma experimental consistió en tareas de imaginación motora guiadas por estímulos visuales, en las que los participantes debían imaginar movimientos específicos durante intervalos de tarea definidos. Las señales fueron procesadas mediante transformadas tiempo–frecuencia y descriptores de potencia espectral, y posteriormente clasificadas mediante algoritmos como LDA y SVM. Los resultados mostraron tasas de clasificación intra-sujeto entre el 70 % y el 80 %, evidenciando la viabilidad de este enfoque para BCIs de imaginación motora.

En [23] se estudió el aporte de la covarianza de las señales EEG a BCIs de imaginación motora orientadas a pacientes con lesión medular. El estudio utilizó señales EEG registradas mediante electrodos sobre la corteza sensorimotora, durante tareas de imaginación motora inducidas mediante estímulos visuales. En lugar de utilizar características escalares tradicionales, los autores modelaron las señales mediante matrices de covarianza y técnicas de geometría de la información en espacios *riemannianos*, lo que permitió mejorar la discriminación entre clases de imaginación motora. Los resultados mostraron mejoras significativas en la precisión de clasificación frente a métodos tradicionales basados en características espectrales.

En cuanto a la implementación práctica, también se han propuesto interfaces neuronales de bajo costo orientadas a personas con discapacidades motoras severas. En [24] se desarrolló una BCI basada en EEG utilizando hardware

accesible, integrando plataformas electrónicas de bajo costo y sensores comerciales para registrar actividad cerebral asociada a estímulos visuales o tareas cognitivas simples. El sistema permitió generar comandos de control básicos para dispositivos externos, demostrando la posibilidad de desarrollar BCIs portátiles y económicamente accesibles para aplicaciones asistivas.

De forma similar, en [25] se desarrolló una interfaz cerebro-máquina portátil basada en EEG para el control de un brazo robótico, utilizando paradigmas de actividad cerebral inducidos mediante estímulos visuales. El sistema fue diseñado con hardware portátil y plataformas de desarrollo accesibles, integrando el procesamiento de señales EEG con el control de actuadores robóticos. Los experimentos demostraron la factibilidad de utilizar BCIs de bajo costo para control de dispositivos asistivos, aunque con limitaciones en la velocidad y precisión del control.

En contraste, el uso de fNIRS para BCIs en el contexto colombiano es todavía incipiente. Las revisiones y propuestas multimodales [26] mencionan la fNIRS principalmente como una tecnología emergente complementaria a EEG para procesos de neurorehabilitación, pero sin reportar sistemas unimodales fNIRS completamente implementados para imaginación motora. Adicionalmente, existen reportes de proyectos y notas técnicas que discuten la idea de BCIs híbridas EEG–fNIRS (por ejemplo, propuestas de interfaces optimizadas tipo *“Optef-BCI”),* pero en general se trata de descripciones conceptuales o sistemas en fase experimental sin validaciones exhaustivas en paradigmas de MI [27].

En síntesis, los antecedentes nacionales muestran: (i) una comunidad activa en BCIs con un énfasis claro en EEG, tanto en investigación básica como en aplicaciones de rehabilitación y control de dispositivos [19]–[23]; (ii) desarrollos de bajo costo y portátiles orientados a poblaciones con discapacidad [24],[25]; y (iii) una exploración aún muy limitada del uso de fNIRS, particularmente en paradigmas de imaginación motora y en su integración con entornos gráficos interactivos como videojuegos, avatares o robots [26],[27]. No se identificaron reportes de sistemas unimodales fNIRS para imaginación motora de mano con validación funcional en escenarios tipo videojuego o realidad virtual, lo que resalta la pertinencia del presente trabajo.

En conjunto, estos trabajos evidencian que gran parte de los desarrollos nacionales en BCI no invasivas se han enfocado principalmente en señales EEG y en aplicaciones de rehabilitación y asistencia [19]–[23].

2.1.2 Antecedentes internacionales

A nivel internacional, la investigación en BCIs basadas en fNIRS ha crecido de manera sostenida en los últimos años, tanto en contextos clínicos como en aplicaciones de neuroergonomía, rehabilitación y control de dispositivos. Diversas revisiones han resumido el estado del arte en fNIRS–BCIs, destacando el uso de esta técnica para decodificar tareas cognitivas, carga mental y, en menor medida,

imaginación o ejecución de movimientos [28]–[30]. En estas revisiones se enfatiza que la mayoría de los pipelines incluyen conversión de intensidad a cambios de concentración de hemoglobina mediante la Ley de Beer–Lambert modificada (MBLL), filtrado pasa–banda en la banda hemodinámica (≈0,01–0,1 Hz) y extracción de características simples en el dominio temporal sobre *ΔHbO y ΔHbR* [28]–[30].

En el ámbito específico de imaginación motora, en [31] se propuso uno de los primeros sistemas unimodales basados en fNIRS para discriminar entre imaginación motora de flexión de muñeca izquierda y derecha. El experimento se realizó con 10 sujetos sanos, a quienes se les presentaban indicaciones visuales en una pantalla para imaginar el movimiento durante intervalos de tarea de aproximadamente 10 s. Las señales fueron registradas mediante canales fNIRS ubicados sobre la corteza motora primaria, ubicando optodos sobre la corteza motora contralateral, y se analizaron cambios en las concentraciones de ΔHbO y ΔHbR obtenidas mediante MBLL.

Se extrajeron características temporales como media y pendiente de la señal, utilizando clasificadores lineales como LDA y SVM para distintas ventanas temporales. Los resultados mostraron que la pendiente de la señal hemodinámica mejora el desempeño de clasificación, alcanzando tasas de acierto superiores al 70 % en clasificación intra-sujeto específicamente rangos de 73–83 % en ventanas de10 s y ≈78–87 % para ventanas 2–7 s; lo que demostró la viabilidad de BCIs unimodales basadas en fNIRS para tareas de imaginación motora.

Posteriormente se han publicado bases de datos abiertas de fNIRS para clasificación de movimientos de mano unilateral. Por ejemplo, en [32] se presentó un dataset fNIRS obtenido con 30 participantes, quienes realizaron movimientos de golpeteo de dedos de mano izquierda, mano derecha y pie, con 25 repeticiones por condición. Las señales hemodinámicas fueron registradas mediante múltiples canales fNIRS sobre regiones motoras y se utilizaron cambios de HbO y HbR como variables principales. Utilizando clasificadores SVM, LDA, K-NN con validación cruzada leave-one-out, los autores reportaron exactitudes promedio cercanas al rango 70 % y 90% para clasificación multiclase, con mejoras en clasificación binaria en la mayoría de los participantes.

El conjunto de datos público utilizado en este trabajo corresponde a un experimento híbrido EEG–fNIRS reportado en [33]. El estudio fue realizado con 29 sujetos sanos que ejecutaron dos paradigmas BCI: imaginación motora de mano izquierda/derecha y aritmética mental vs. reposo. Las señales fNIRS se adquirieron mediante 36 canales fisiológicos, distribuidos sobre regiones frontales, motoras y occipitales [28], mientras que el EEG fue registrado simultáneamente mediante un sistema multicanal. En cada sesión los participantes realizaron múltiples repeticiones de cada condición, con períodos de introducción visual (~2 s), tarea (~10 s) y descanso (~15–17 s). Este dataset multimodal fue publicado como banco de pruebas abierto para clasificación de una sola prueba (single-trial classification) y ha sido ampliamente utilizado para comparar algoritmos de clasificación

unimodales y multimodales en BCIs. Estudios que utilizan únicamente la rama fNIRS han reportado exactitudes intra–sujeto típicamente entre 70 % y 90 %, dependiendo de la ventana temporal seleccionada, el número de canales considerados y el clasificador empleado, utilizando desde modelos lineales como *LDA/SVM* hasta bosques aleatorios, *XGBoost* y arquitecturas de redes neuronales convolucionales 1D aplicadas a las series de tiempo hemodinámicas [34]–[36].

En paralelo, se han propuesto esquemas de selección de canales para fNIRS basados en el coeficiente de determinación r^2, información mutua o la importancia de características derivada de modelos de árbol (*Random Forest, Gradient Boosting*). En [37], por ejemplo, estudiaron estrategias de selección automática de canales efectivos en BCIs de imaginación motora basadas en fNIRS. El estudio evaluó diferentes métricas de relevancia de características sobre señales ΔHbO y ΔHbR registradas en canales motores, con el objetivo de identificar subconjuntos de optodos que maximicen la separabilidad entre clases; mostrando que es posible reducir el número de dichos canales mediante criterios de correlación o importancia de características disminuyendo simultáneamente la dimensionalidad del problema, sin degradar el desempeño e incluso mejorando la robustez del clasificador, siempre que el subconjunto seleccionado capture adecuadamente la activación contralateral asociada a la tarea.

Otros trabajos utilizan la información de importancia de características de modelos tipo *Random Forest o XGBoost* para seleccionar subconjuntos de canales relevantes antes del entrenamiento del clasificador final [34],[37].

En cuanto a la interacción con el usuario, varios estudios han explorado la integración de BCIs fNIRS con entornos de realidad virtual o interfaces gráficas interactivas. En [38] se implementó un sistema BCI basado en fNIRS con retroalimentación cerrada (closed-loop) para tareas de imaginación motora. El sistema utilizó canales ubicados sobre la corteza motora para detectar cambios en ΔHbO asociados a la tarea, los cuales eran traducidos en comandos discretos para controlar un entorno interactivo. Los participantes realizaban tareas de imaginación motora guiadas por estímulos visuales, y los resultados mostraron que la señal hemodinámica puede emplearse para generar control en tiempo real en entornos interactivos tipo videojuego, aunque con latencias inherentes a la respuesta hemodinámica. No obstante, la mayoría de estos trabajos se centra en paradigmas sencillos (dos o tres clases) y en clasificadores relativamente simples, con validación funcional realizada generalmente sobre un único algoritmo y sin estrategias avanzadas de rechazo de decisiones inciertas o comités de clasificadores.

Finalmente, en la literatura internacional sobre clasificación en BCIs se ha documentado ampliamente el uso de *ensembles* y estrategias de combinación de clasificadores (bagging, boosting, stacking) para mejorar la robustez frente a la variabilidad inter–sujeto y al ruido en las señales, principalmente en el dominio EEG [39],[40]. Sin embargo, su aplicación sistemática en BCIs basadas exclusivamente

en fNIRS sigue siendo menos frecuente, y son escasos los trabajos que combinan ensembles heterogéneos con mecanismos explícitos de rechazo adaptativo de decisiones en el contexto de imaginación motora.

2.1.3 Discusión de antecedentes

De la revisión de antecedentes se concluye que los pipelines fNIRS–BCI más comunes incluyen: MBLL para obtener ΔHbO/ΔHbR, filtrado pasa–banda ~0,01–0,1 Hz para aislar la respuesta hemodinámica lenta y algún método de corrección de artefactos de movimiento; sin embargo, muchos trabajos no documentan con detalle parámetros clave (DPF, longitudes de onda, ventanas y criterios de corrección), lo que limita la reproducibilidad [28]–[30].

En MI-fNIRS predominan características temporales de baja dimensión (media, pendiente, máximos, varianza y ΔHbO−ΔHbR) por su bajo costo e interpretabilidad [28], [31]. La selección de canales/variables (por ejemplo, con r^2 u otros filtros) es común para reducir dimensionalidad y mejorar estabilidad, aunque requiere salvaguardas para evitar sesgos espaciales [32], [34]. En clasificación, LDA suele ser referencia y se emplean también SVM, árboles y ensambles, siempre con validación cruzada y control de sobreajuste [34]–[36].

Con base en lo anterior, se identifica la necesidad de desarrollar un pipeline de procesamiento para señales fNIRS orientado a la clasificación de imaginación motora que incorpore buenas prácticas reportadas en la literatura, tales como la conversión mediante MBLL, el filtrado en la banda hemodinámica y el uso de características temporales relevantes. Asimismo, se evidencia la oportunidad de explorar estrategias adicionales que mejoren la robustez del sistema, como la selección adecuada de canales, el uso de múltiples clasificadores y la incorporación de mecanismos que permitan manejar la incertidumbre en la toma de decisiones.

En este contexto, el presente trabajo propone un enfoque que integra estas consideraciones dentro de un pipeline completo de procesamiento y clasificación, el cual será descrito en detalle en el capítulo de metodología. En la tabla 2-1, se resumen los antecedentes más relacionados con el presente trabajo, incluyendo sus aspectos más relevantes y que sirvieron de soporte para la metodología más adelante implementada.

Tabla 2-1.Comparativa de antecedentes consultados.

Referencia	Descripción del dataset	Filtrado	Selección de características	Clasificador	Desempeño
[23]	MI EEG para rehabilitación de miembro superior en pacientes con lesión medular (Colombia).	EEG multicanal, filtrado en bandas μ y β, construcción de matrices de covarianza por ensayo.	Covarianza EEG sobre canales sensorimotores; análisis en espacio riemanniano; comparación con potencia espectral clásica.	Clasificación en el espacio de matrices de covarianza con métodos riemannianos y clasificadores lineales; validación intra–sujeto.	Accuracies típicas **≈70–85 %** según sujeto y condición; mejora frente a descriptores escalares clásicos.
[26]	BCI multimodal para rehabilitación motora (EEG + posibles señales adicionales).	EEG con filtrado clásico de ritmos sensorimotores; protocolos de	Potencia espectral en bandas de interés; posibles índices	Clasificadores lineales (LDA/SVM); validación en	Desempeños reportados **≥70 %** en detección de intención de

		rehabilitación estructurados.	combinados por modalidad.	protocolo clínico de rehabilitación.	movimiento; muestra factibilidad de BCIs en rehabilitación en contexto colombiano.
[31]	BCI fNIRS unimodal para imaginación de muñeca derecha vs. izquierda.	NIRS CW sobre corteza motora; MBLL, filtrado hemodinámico (≈0,01–0,1 Hz), segmentación en ventanas 2–10 s.	Medias y pendientes de ΔHbO/ΔHbR por canal y ventana; selección de canales motores contralaterales.	LDA y SVM con validación intra–sujeto; comparación de distintas ventanas de análisis.	Accuracies promedio **≈73–83 %** en 10 s y **≈78–87 %** para ventanas 2–7 s, según clasificador y ventana.
[32]	Dataset abierto fNIRS para movimientos unilaterales de mano.	fNIRS sobre corteza motora; MBLL, filtrado hemodinámico y segmentación en pruebas de movimiento o MI.	Características temporales de ΔHbO/ΔHbR por canal; exploración de selección de canales efectivos.	Clasificadores lineales y no lineales (p. ej. LDA, SVM, k-NN) evaluados en múltiples sujetos.	Accuracies típicas en el rango **70–90 %** según clase, sujeto y número de canales seleccionados.
[34]	BCI híbrido ternario EEG–fNIRS (MI mano + aritmética mental + reposo) usando el mismo dataset que este trabajo.	EEG + fNIRS; MBLL, filtrado hemodinámico, filtrado en bandas EEG, segmentación de ensayos de MI y MA.	Características temporales fNIRS (medias, pendientes) + CSP/PSD EEG; selección de canales basada en discriminabilidad.	Fusión de características EEG–fNIRS y clasificadores lineales (LDA, SVM); validación intra–sujeto y análisis de ternariedad (tres clases).	Se reportan **accuracies intra–sujeto en el orden de 70–90 %** para configuraciones óptimas, mostrando ventaja de la modalidad híbrida sobre unimodal.
[36]	MI fNIRS con clasificación basada en CNN (hasta 4 clases de gestos/MI de mano).	fNIRS CW sobre corteza motora; MBLL, filtrado pasa–banda hemodinámico, segmentación fija de ensayos.	Entrada directa de series de ΔHbO/ΔHbR a una CNN 1D; comparación con características clásicas de dominio temporal.	CNN 1D profunda; comparación con SVM/LDA tradicionales; validación cruzada por sujeto.	Para cuatro clases de MI, accuracies promedio alrededor de **72 % ± 4 %**, superiores a los métodos clásicos en el mismo dataset.
[Trabajo de Grado Realizado]	BCI unimodal fNIRS para imaginación motora de mano izquierda/derecha, con integración funcional en un avatar Unity.	fNIRS CW (dataset [33]); MBLL, corrección de artefactos basada en canales EOG/EMG, filtrado Chebyshev II (0,01–0,09 Hz), fusión de tres sesiones MI, segmentación (–10 a +25 s) y normalización por línea base.	Características temporales de ΔHbO/ΔHbR (media, pendiente, máximo, varianza, ΔHbO–ΔHbR) en ventanas de tarea/reposo; selección de canales motores guiada por $\|r^2\|$, garantizando representación bilateral.	Comité de clasificadores heterogéneo (bagging-LDA, SVM, RF, XGBoost, MLP) con stacking, poda de expertos débiles y umbral de rechazo adaptativo por sujeto; validación intra–sujeto y exportación de comandos discretos para Unity.	Desempeño intra–sujeto en el Split más robusto sin rechazo es **70.5%,** con rechazo **75%** y el BA es **75.9**%, medido sobre 28 sujetos del dataset

2.2 Marco teórico

El marco teórico que soporta el desarrollo de este trabajo se estructura en varios ejes conceptuales relacionados con las interfaces cerebro–computadora basadas en espectroscopia funcional de infrarrojo cercano (fNIRS). En particular, se abordan los fundamentos físicos de la técnica fNIRS, los principios de la Ley de Beer–Lambert modificada, los paradigmas de imaginación motora utilizados en BCIs, las técnicas de procesamiento digital de señales fNIRS, los métodos de aprendizaje automático aplicados a la clasificación de estados cerebrales y las métricas empleadas para evaluar el desempeño de estos sistemas. Estos conceptos constituyen la base teórica necesaria para comprender el diseño y funcionamiento de la interfaz cerebro–computadora propuesta en este trabajo.

2.2.1 Espectroscopia funcional de infrarrojo cercano (fNIRS)

La espectroscopia funcional de infrarrojo cercano (fNIRS) es una técnica óptica no invasiva utilizada para medir cambios en la oxigenación sanguínea cerebral asociados a la actividad neuronal. Esta técnica se basa en la emisión de luz en el rango del infrarrojo cercano (aproximadamente entre 700 y 900 nm), la cual puede penetrar varios centímetros en el tejido biológico antes de ser parcialmente absorbida y dispersada [28], [30].

En un sistema fNIRS típico, una fuente óptica emite luz hacia el cuero cabelludo y un detector ubicado a una distancia aproximada de 2.5 a 3.5 cm registra la fracción de luz que retorna después de atravesar el tejido cerebral. Durante este recorrido, la luz es absorbida principalmente por la oxihemoglobina (HbO) y la desoxihemoglobina (HbR), cuya concentración cambia en respuesta a la actividad neuronal mediante el mecanismo de acoplamiento neurovascular [28], [29].

Cuando una región cortical se activa, aumenta el flujo sanguíneo local, lo que produce un incremento en la concentración de HbO y una disminución relativa de HbR. Este fenómeno genera una señal hemodinámica medible que puede utilizarse para inferir patrones de activación cerebral asociados a tareas cognitivas o motoras [28], [30]. Gracias a su portabilidad, bajo costo relativo y tolerancia al movimiento en comparación con otras técnicas de neuroimagen, la fNIRS ha sido ampliamente utilizada en investigaciones sobre interfaces cerebro–computadora no invasivas [4], [5].

2.2.2 Ley de Beer-Lambert modificada (MBLL)

Los sistemas fNIRS de onda continua registran inicialmente variaciones en la intensidad de la luz detectada, las cuales deben transformarse en cambios de concentración de hemoglobina para poder interpretarse fisiológicamente. Esta conversión se realiza mediante la Ley de Beer–Lambert modificada (MBLL), que extiende la ley clásica de absorción óptica a medios altamente dispersivos como el tejido biológico [28], [30].

La MBLL relaciona los cambios en la absorbancia óptica con las variaciones de concentración de cromóforos presentes en el tejido. En el caso de la fNIRS, los principales cromóforos de interés son la oxihemoglobina (HbO) y la desoxihemoglobina (HbR). Para cada longitud de onda λ, el cambio en la absorbancia óptica puede aproximarse como una combinación lineal de los cambios de concentración de estos dos componentes [30].

Al utilizar al menos dos longitudes de onda diferentes, es posible resolver un sistema de ecuaciones lineales que permite estimar las variaciones relativas de concentración de HbO y HbR en función del tiempo [28], [30]. La MBLL incorpora además el factor de camino diferencial (DPF), el cual corrige el hecho de que la luz sigue trayectorias efectivas más largas que la distancia geométrica entre fuente y detector debido a los fenómenos de dispersión en el tejido.

La aplicación de MBLL constituye un paso fundamental en el procesamiento de señales fNIRS, ya que transforma las mediciones instrumentales de intensidad de luz en señales hemodinámicas interpretables desde el punto de vista fisiológico [68].

2.2.3 Imaginación motora en interfaces cerebro–computadora

Las interfaces cerebro–computadora (Brain–Computer Interfaces, BCI) son sistemas que permiten establecer una comunicación directa entre el cerebro y un dispositivo externo sin utilizar la vía motora periférica. En general, una BCI comprende varias etapas: adquisición de señales cerebrales, preprocesamiento, extracción de características, clasificación y generación de comandos para un sistema externo [39], [40].

Uno de los paradigmas más utilizados en BCIs es la imaginación motora (*Motor Imagery, MI*). En este paradigma, el usuario imagina la ejecución de un movimiento específico sin realizarlo físicamente. Este proceso activa redes neuronales similares a las involucradas en la ejecución real del movimiento, especialmente en la corteza motora primaria, la corteza premotora y el área suplementaria motora [31].

En tareas de imaginación motora de mano izquierda y derecha, la activación cerebral presenta un patrón de lateralización contralateral. En estudios basados en EEG, esta actividad se refleja en cambios en los ritmos sensorimotores μ (8–12 Hz) y β (13–30 Hz). En el caso de la fNIRS, la activación cortical produce un aumento de la concentración de *HbO* y una disminución de *HbR* en las regiones motoras correspondientes [31], [32].

Este tipo de paradigma ha sido ampliamente utilizado para el control de dispositivos externos, sistemas de rehabilitación motora y entornos virtuales interactivos [5], [15]. Además, diversos conjuntos de datos abiertos han sido diseñados específicamente para evaluar sistemas BCI basados en imaginación motora, como el dataset híbrido EEG–fNIRS desarrollado por la Universidad Técnica de Berlín [3], [33].

2.2.4 Procesamiento digital de señales fNIRS

Las señales fNIRS suelen estar afectadas por múltiples fuentes de ruido y artefactos, incluyendo movimientos de la cabeza, contracciones musculares faciales, parpadeos y variaciones fisiológicas sistémicas como cambios respiratorios o cardiovasculares [29], [30]. Por esta razón, el preprocesamiento constituye una etapa fundamental en los sistemas BCI basados en fNIRS.

a) **Corrección de artefactos**
   Los artefactos de movimiento pueden generar cambios abruptos en la intensidad de la señal óptica que no están relacionados con la actividad cerebral. Diversos métodos han sido propuestos para mitigar estos efectos, incluyendo interpolación de segmentos contaminados, filtrado *wavelet* y métodos basados en análisis estadístico de la señal [41]–[45].

La corrección adecuada de estos artefactos es crucial para evitar interpretaciones erróneas de la señal hemodinámica y mejorar la estabilidad de los clasificadores utilizados en sistemas BCI.

b) Filtrado hemodinámico

Las señales fNIRS presentan componentes fisiológicos en diferentes bandas de frecuencia, incluyendo oscilaciones respiratorias, pulsaciones cardíacas y variaciones lentas del flujo sanguíneo. Para aislar la respuesta hemodinámica asociada a la actividad neuronal, es común aplicar filtros pasa–banda que retienen únicamente las frecuencias de interés [30]. En estudios de imaginación motora, la respuesta hemodinámica relevante suele encontrarse en frecuencias inferiores a 0.1 Hz, por lo que los filtros pasa–banda se diseñan típicamente en el rango de 0.01 a 0.1 Hz [4], [30].

c) Segmentación temporal de ensayos

En experimentos de BCI, los registros continuos de señal se dividen en segmentos temporales asociados a eventos experimentales específicos, como el inicio de una tarea o un periodo de reposo. Este proceso se conoce como segmentación en épocas o *epoching*, y permite analizar la evolución temporal de la respuesta hemodinámica asociada a cada ensayo [4]. La segmentación facilita además la extracción de características relevantes y la construcción de conjuntos de datos adecuados para el entrenamiento de clasificadores supervisados.

2.2.5 Extracción de características en señales fNIRS

Una vez preprocesadas las señales, es necesario transformar las series temporales de HbO y HbR en vectores de características que puedan ser utilizados por algoritmos de aprendizaje automático.

En estudios de fNIRS-BCI se utilizan con frecuencia características del dominio temporal, tales como la media, la pendiente, la varianza, el máximo o el contraste entre HbO y HbR dentro de una ventana temporal determinada [4], [5]. Estas características capturan propiedades relevantes de la respuesta hemodinámica asociada a la activación cortical.

Debido al alto número de canales y variables potenciales, también es común aplicar métodos de selección de características para reducir la dimensionalidad del problema y mejorar la generalización de los modelos de clasificación. Entre las técnicas utilizadas se encuentran métodos basados en correlación, eliminación recursiva de características (RFE) y algoritmos basados en importancia de variables como *Random Forest* o *Boruta* [7], [54], [55].

2.2.6 Aprendizaje automático en BCIs fNIRS

Los sistemas BCI utilizan algoritmos de aprendizaje automático para identificar patrones en las señales cerebrales que correspondan a diferentes estados mentales o intenciones del usuario. Entre los algoritmos más utilizados se encuentran los

clasificadores lineales, las máquinas de soporte vectorial (SVM), los árboles de decisión y los métodos de ensamblado [39], [40].

Los métodos de ensemble learning combinan múltiples modelos base con el objetivo de mejorar el desempeño global del sistema. Técnicas como bagging, random forests o boosting han demostrado ser efectivas para manejar conjuntos de datos complejos y con alta variabilidad entre sujetos [50]–[53]. Una estrategia particularmente utilizada es el stacking, en la cual varios clasificadores base generan predicciones que posteriormente son combinadas por un meta-clasificador encargado de producir la decisión final [49].

2.2.7 Modelos de clasificación en BCIs basadas en fNIRS

Para la decodificación de estados mentales a partir de señales fNIRS se han utilizado diversos algoritmos de aprendizaje automático. Entre los más comunes se encuentran los clasificadores lineales, las máquinas de soporte vectorial (SVM), los métodos basados en árboles de decisión y las redes neuronales artificiales [39], [40].

Los clasificadores lineales, como el Análisis Discriminante Lineal (LDA), han sido ampliamente utilizados en sistemas BCI debido a su simplicidad, estabilidad numérica y buen desempeño cuando el número de muestras disponibles es limitado. LDA modela cada clase mediante distribuciones gaussianas con covarianza compartida y calcula funciones discriminantes lineales que permiten separar las clases en el espacio de características [39].

Las máquinas de soporte vectorial (SVM) constituyen otra técnica ampliamente utilizada en BCIs. Estos modelos buscan encontrar un hiperplano que maximice el margen entre las clases, lo que permite obtener clasificadores robustos incluso en espacios de alta dimensionalidad. Mediante el uso de funciones *kernel*, como el *kernel radial basis function (RBF),* es posible capturar relaciones no lineales presentes en las señales cerebrales [39], [40].

Por otra parte, los métodos basados en árboles de decisión y *Random Forest* permiten modelar relaciones no lineales e interacciones entre variables. *Random Forest* consiste en un conjunto de árboles de decisión entrenados sobre subconjuntos aleatorios de los datos y de las variables, cuyos resultados se combinan mediante votación o promedio. Este enfoque ha demostrado ser robusto frente al ruido y adecuado para problemas con gran número de características [51].

Los métodos de *gradient boosting*, como *XGBoost,* representan otra familia de algoritmos ampliamente utilizados en problemas de clasificación complejos. En estos modelos, los árboles se construyen de forma secuencial, de modo que cada

nuevo modelo corrige los errores cometidos por los anteriores. Este enfoque permite construir modelos altamente expresivos manteniendo mecanismos de regularización que controlan el sobreajuste [52], [53].

Finalmente, las redes neuronales artificiales, como el perceptrón multicapa (MLP), permiten aproximar relaciones no lineales complejas entre las variables de entrada y las clases objetivo. Estas redes utilizan capas de neuronas artificiales conectadas mediante funciones de activación no lineales y son entrenadas mediante algoritmos de optimización basados en descenso de gradiente [59].

Debido a que cada uno de estos modelos posee diferentes sesgos inductivos y capacidades de representación, diversos estudios han propuesto combinar múltiples clasificadores para mejorar el desempeño de sistemas BCI. En particular, los enfoques de *ensemble learning* permiten integrar modelos de distinta naturaleza con el fin de obtener decisiones más robustas frente al ruido y la variabilidad inter-sujeto característica de las señales cerebrales [49]–[53].

Entre estos enfoques destaca el *stacking*, en el cual varios clasificadores base generan predicciones que posteriormente son combinadas por un meta-clasificador encargado de producir la decisión final del sistema [49]. Este tipo de estrategias ha sido aplicado con éxito en problemas de clasificación de señales cerebrales, donde la diversidad de modelos puede mejorar la capacidad de generalización del sistema [9]–[14].

2.2.8 Clasificación selectiva y calibración de probabilidades

En aplicaciones prácticas de BCI es importante que el sistema pueda abstenerse de tomar decisiones cuando la confianza del clasificador es insuficiente. Este enfoque se conoce como clasificación selectiva con opción de rechazo y permite mejorar la fiabilidad del sistema al evitar decisiones potencialmente incorrectas [58], [66].

Para implementar este tipo de estrategias, es necesario que los clasificadores produzcan estimaciones probabilísticas bien calibradas. Métodos como el *Platt scaling* o la *calibración isotónica* se utilizan para transformar las puntuaciones del clasificador en probabilidades interpretables [56], [57].

2.2.9 Métricas de evaluación en sistemas BCI

El desempeño de los sistemas BCI se evalúa comúnmente mediante métricas de clasificación como *accuracy, balanced accuracy* y matrices de confusión. La accuracy representa la proporción de predicciones correctas respecto al total de muestras evaluadas.

En problemas con posibles desbalances entre clases, la balanced accuracy resulta más adecuada, ya que promedia la sensibilidad obtenida en cada clase y reduce el sesgo hacia la clase mayoritaria. Además de estas métricas, en sistemas que incorporan opción de rechazo también se analizan relaciones entre cobertura y riesgo, las cuales permiten evaluar el compromiso entre el número de decisiones tomadas por el sistema y la fiabilidad de dichas decisiones [12], [13].

2.2.10 Control de avatares en entornos virtuales

Como parte del marco conceptual, se considera la interacción de la BCI con un entorno virtual en Unity, donde los comandos discretos del clasificador (*idle,* mano izquierda, mano derecha) controlan animaciones de un avatar humanoide. Este tipo de retroalimentación visual se ha utilizado en la literatura para demostrar control BCI en entornos interactivos (p. ej., videojuegos o realidad virtual) [15]–[18], [38]. En este trabajo, la integración se limita a una validación funcional offline mediante archivos de comandos, con el fin de verificar la trazabilidad del pipeline sin introducir requerimientos de operación en línea.

3. Instrumentación original del conjunto de datos

3.1 Introducción

En este capítulo se documenta la selección de la base de datos, con su respectiva instrumentación original de adquisición, el paradigma experimental y la forma en que los registros se entregan en estructuras .mat del conjunto de datos abierto registrado en la Technische Universität Berlin, ya que estos elementos determinan las decisiones de procesamiento posteriores en la BCI diseñada en este trabajo.

En particular, se presenta: (i) el sistema *NIRScout* que empleó la TU Berlin para registrar intensidades a 760/850 nm en canales distribuidos sobre regiones frontal, motora y visual, (ii) el esquema experimental de imaginación motora utilizado y la selección exclusiva de sus sesiones, (iii) la estructura de archivos (*cnt, mrk, mnt* y señales/marcas de artefactos) que sirve como entrada al pipeline, y (iv) la plataforma de cómputo utilizada dentro de este trabajo para ejecutar el procesamiento *offline* y generar comandos que se integran en un entorno de simulación en Unity.

3.2 Selección de la base de datos

La selección de la base de datos constituye un paso fundamental en el desarrollo de sistemas de interfaz cerebro–computadora (BCI), ya que las características del conjunto de datos influyen directamente en el diseño del pipeline de procesamiento, los métodos de extracción de características y los algoritmos de clasificación utilizados.

En el caso de las BCIs basadas en espectroscopia funcional de infrarrojo cercano (fNIRS), existen diversos conjuntos de datos abiertos que han sido utilizados en la literatura para evaluar métodos de clasificación de estados mentales, particularmente en paradigmas de imaginación motora. Entre los datasets más utilizados se encuentran bases de datos unimodales fNIRS y bases de datos multimodales que combinan fNIRS con electroencefalografía (EEG) [4], [5], [33].

Con el fin de seleccionar una base de datos adecuada para el desarrollo del presente trabajo, se realizó una revisión de diferentes datasets disponibles públicamente en la literatura de interfaces cerebro–computadora. La Tabla 3-1 presenta una comparación entre algunos de los conjuntos de datos más relevantes utilizados en investigaciones recientes de BCIs basadas en fNIRS.

Tabla 3-1. Comparación de la base de datos fNIRS para BCI.

Dataset	Modalidad	Tareas	Participantes	Canales	Disponibilidad
TU Berlin EEG-fNIRS dataset [3], [33]	EEG + fNIRS	MI izquierda/derecha, aritmética mental	29	36	Open access
Open fNIRS unilateral hand movement dataset [32]	fNIRS	Movimiento mano izquierda/derecha	18	20	Open access
Naseer & Hong dataset [31]	fNIRS	Imaginación motora muñeca izquierda/derecha	10	16	Dataset experimental
Hybrid EEG-NIRS BCI dataset [34]	EEG + fNIRS	MI y tareas cognitivas	29	múltiples	Open access

A partir de este análisis comparativo se seleccionó el dataset híbrido EEG–fNIRS desarrollado por la Universidad Técnica de Berlín [3], [33], el cual constituye uno de los conjuntos de datos abiertos más utilizados en investigaciones recientes de BCIs.

La elección de este dataset se fundamenta en varias razones. En primer lugar, el conjunto de datos incluye registros de múltiples participantes realizando tareas de imaginación motora de mano izquierda y derecha, lo cual resulta adecuado para el objetivo de este trabajo orientado a la clasificación de estados de imaginación motora. En segundo lugar, el dataset proporciona señales fNIRS registradas con una configuración de 36 canales distribuidos en regiones corticales relevantes para tareas motoras, lo que permite analizar patrones de activación asociados a la corteza motora.

Adicionalmente, el conjunto de datos se encuentra disponible públicamente y ha sido utilizado en diversos estudios previos de clasificación de señales cerebrales, lo

que facilita la reproducibilidad y la comparación de resultados con otros enfoques propuestos en la literatura [3], [33].

Por estas razones, el dataset de TU Berlin fue seleccionado como base experimental para el desarrollo y evaluación del sistema BCI propuesto en este trabajo.

3.3 Módulo de adquisición fNIRS del conjunto de datos

El conjunto de datos utilizado en este trabajo corresponde al experimento híbrido EEG–fNIRS publicado por la TU Berlin, registrado con un sistema *NIRScout (NIRx GmbH*, Berlín, Alemania) [3],[33].

El sistema dispone de 14 fuentes y 16 detectores, configurados para generar 36 canales fisiológicos mediante pares fuente–detector adyacentes distribuidos sobre tres regiones corticales: frontal (alrededor de *Fp1, Fp2 y Fpz*), motora (alrededor de *C3* y *C4*) y visual (alrededor de Oz). La distancia fuente–detector se fijó en 30 mm, y las mediciones se realizaron a dos longitudes de onda, 760 y 850 nm, dentro de la llamada “ventana óptica” del tejido biológico [28]–[30].

Los sensores ópticos del sistema fNIRS (optodas) se montaron en un gorro opaco compartido con los electrodos de EEG, siguiendo el sistema 10–5, garantizando un contacto firme con el cuero cabelludo y reduciendo la influencia de la luz ambiental [30],[33]. La adquisición se llevó a cabo en una sala con iluminación tipo oficina, con los sujetos sentados en un sillón cómodo frente a una pantalla blanca de 50 pulgadas a una distancia de aproximadamente 1,6 m [33].

El sistema *NIRScout* registró las intensidades de luz con una frecuencia de muestreo de 12,5 Hz, posteriormente re-muestreada a 10 Hz en los archivos publicados. No se aplicó ningún procesamiento adicional sobre las señales fNIRS, aparte de la conversión de formato indicada por los autores [33].

Además de los canales fNIRS, el protocolo registró de forma simultánea señales de *EOG, EMG, ECG* y respiración empleando el amplificador EEG; éstas se utilizaron para construir conjuntos de datos específicos de artefactos (dataset C), que resultan especialmente útiles para el diseño de estrategias de corrección de movimiento [33]. En este trabajo, los canales de artefactos asociados a *EOG* y *EMG* se aprovecharon para guiar la interpolación de segmentos contaminados en las señales hemodinámicas.

3.3.1 Paradigma experimental de adquisición

El dataset incluye tres sesiones de imaginación motora (MI) por sujeto (y sesiones de aritmética mental en un bloque separado); sin embargo, en este trabajo se emplean únicamente las sesiones de MI [33]. Cada sesión de MI contempla un reposo inicial (60 s), 20 ensayos y un reposo final (60 s).

En el contexto del dataset utilizado, es importante distinguir entre sesión y ensayo. Una sesión corresponde a un bloque completo del experimento realizado por el sujeto, el cual incluye un periodo de reposo inicial, una serie de repeticiones de la tarea y un periodo de reposo final. En el caso del paradigma de imaginación motora, cada sujeto realizó tres sesiones de MI. Dentro de cada sesión se ejecutan múltiples ensayos (trials), en este caso 20. Cada ensayo corresponde a una repetición individual de la tarea experimental, compuesta por una instrucción visual, un periodo de imaginación motora y un periodo de descanso.

En este trabajo, tras la depuración de los datos, se emplearon 60 ensayos válidos por sujeto (20 ensayos por cada una de las 3 sesiones) (30 LMI y 30 RMI) para el entrenamiento y evaluación de los modelos de clasificación. En cada ensayo la secuencia es:

Introducción visual (2 s) con una flecha que indica MI de mano izquierda o derecha, seguida de un período de tarea (10 s) con cruz de fijación. Durante la tarea, el sujeto imagina el agarre (abrir/cerrar la mano) a ~1 Hz, buscando una imaginación háptica y no sólo visual [31], [33]. Finalmente, se presenta un período de reposo aleatorio (15–17 s) con cruz de fijación, en el que se solicita minimizar movimientos.

La tarea termina con un pitido de 250 ms y la palabra STOP en pantalla durante 1 s [33]. En cada sesión de MI se presentan 10 ensayos por condición (mano izquierda / mano derecha), organizados en bloques pseudoaleatorios. En este trabajo se utilizaron exclusivamente las tres sesiones de MI por sujeto (dataset A), lo que proporciona 60 ensayos de imaginación motora por participante.

En síntesis, el módulo de adquisición entrega señales crudas en formato .mat (cnt, mrk, mnt) y, adicionalmente, registros/marcadores de artefactos de movimiento (cnt_artifact, mrk_artifact) que se emplean posteriormente para corregir segmentos afectados. Estas estructuras constituyen la entrada del pipeline de procesamiento descrito en los capítulos siguientes [33].

3.4 Plataforma de cómputo para el procesamiento de datos

El procesamiento de los 28 sujetos se realizó en un equipo de cómputo de propósito general utilizando Python como lenguaje principal de programación. Se emplearon librerías como *NumPy, SciPy, scikit–learn, imbalanced–learn, xgboost, matplotlib y python–docx,* entre otras. El pipeline completo incluye etapas computacionalmente intensivas, tales como la selección de características mediante métodos *wrapper*, el entrenamiento de múltiples clasificadores por sujeto y la validación cruzada anidada [34]–[36],[39],[40].

Desde el punto de vista funcional, como entrada se tienen los archivos *.mat* crudos (*cnt, mrk, cnt_artifact*, etc.), como procesamiento se tiene el pipeline de

preprocesamiento, extracción / selección de características y entrenamiento / evaluación del ensamble y como salidas se tendrían archivos .mat intermedios (por ejemplo, *Fused_MI_MBLL_cnt.mat, Epochs_MI_classification.mat*) y archivos .*csv* con comandos discretos por ensayo (commands_subjectXX_clean.csv), que se utilizarán en la plataforma de simulación.

En la práctica, el tiempo total de entrenamiento por sujeto puede ser elevado debido a la validación anidada y a la exploración de múltiples configuraciones. Por ello se controló el número de modelos entrenados y se automatizó el pipeline para ejecutar el procesamiento de forma reproducible.

3.5 Conclusiones del capítulo

En este capítulo se caracterizó el marco experimental y computacional que soporta la BCI basada en fNIRS desarrollada en este trabajo, diferenciando claramente las fuentes de información y el entorno de simulación. Se detalló la instrumentación original del conjunto de datos abierto de la TU Berlin (sistema *NIRScout*, configuración de optodas, canales y parámetros de muestreo), así como el paradigma experimental de imaginación motora empleado para el desarrollo del sistema. Asimismo, se estableció la relación entre la estructura de archivos entregada por la base de datos (*cnt, mrk, mnt* y archivos de artefactos) y su papel como entrada del pipeline descrito en capítulos posteriores.

Finalmente, se presentó la plataforma de cómputo utilizada para ejecutar el procesamiento intensivo, resaltando las implicaciones prácticas en tiempos de entrenamiento y la necesidad de estrategias de optimización (reducción de variables, reutilización de resultados intermedios y control del número de modelos). En conjunto, este capítulo define las condiciones de entrada y los recursos técnicos sobre los cuales se fundamenta la metodología de software y la evaluación del desempeño del sistema

4. Metodología de la Solución Propuesta: software

4.1 Introducción

Este capítulo describe la metodología de la solución propuesta desde el punto de vista de software. Se presenta el pipeline implementado en Python para procesar señales fNIRS del dataset TU Berlin [33] (sesiones de imaginación motora) y generar, como salida, comandos discretos que luego se reproducen en Unity para validación funcional. El flujo incluye: conversión MBLL, fusión de sesiones MI, corrección de artefactos, filtrado hemodinámico, segmentación/ventaneo, extracción y selección de características, clasificación con stacking y rechazo, y exportación de resultados.

La Figura 4-1 presenta el diagrama de bloques del pipeline software implementado, mientras que los diagramas de flujo de las funciones principales se incluyen para documentar la secuencia de operaciones realizada por los scripts desarrollados.

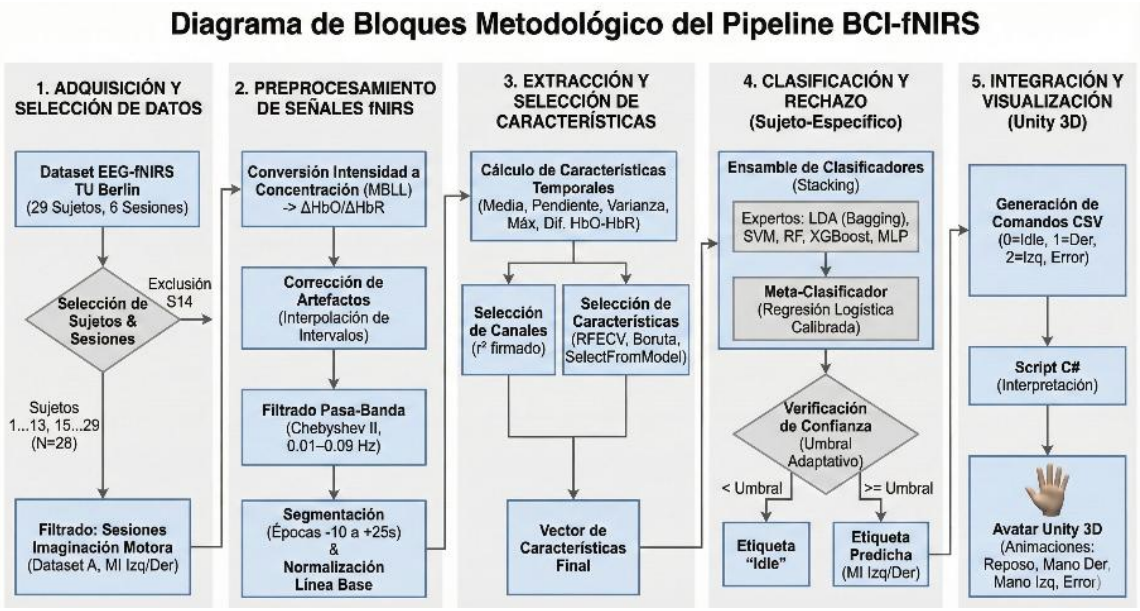


Figura 4-1. Diagrama de bloques del pipeline software para la BCI fNIRS propuesta.

Se denota que previamente a realizar las etapas de procesamiento, se consultó la fuente original de la base de datos [33] que es *BBCI ToolBox,* para poder entender la estructura de cada archivo *.mat* y así poder diseñar los códigos correctamente.

4.1.1 Caracterización de la base de datos y estructuras

Antes de implementar las etapas de preprocesamiento y clasificación, se realizó una fase de caracterización detallada de la base de datos NIRS de la TU Berlin. En primer lugar, se revisó la documentación oficial del conjunto de datos EEG–fNIRS, donde se describe la organización en archivos *cnt.mat, mrk.mat, mnt.mat*, *cnt_artifact.mat, mrk_artifact.mat y mnt_artifact.mat*, así como la asignación de las seis sesiones (tres de imaginación motora y tres de aritmética mental) y los tipos de artefactos de movimiento registrados [33].

Esta documentación remite además a las estructuras básicas *del BBCI Toolbox*, sobre las que se construye el dataset. Posteriormente, se desarrollaron scripts exploratorios en Python utilizando *scipy.io.loadmat* para inspeccionar directamente el contenido de cada archivo .mat. Estos scripts permitieron listar las variables almacenadas, identificar los campos de cada estructura (por ejemplo*, x, fs, clab, wavelengths, title, yUnit, pos, y*), y verificar sus dimensiones. Se comprobó, por ejemplo, que el archivo *cnt.mat* contiene una celda *cnt* de tamaño 1×6, donde cada elemento almacena una matriz x de dimensión *N×72*, junto con la frecuencia de muestreo (fs = 10 Hz), las longitudes de onda (*wavelengths = [760, 850] nm)* y las etiquetas de canal *(clab)* [33].

4.1.2 Organización de los archivos y estructura de canales

Cada sujeto del conjunto de datos cuenta con varios archivos en formato MATLAB (.mat) que almacenan las señales fNIRS crudas, los montajes de canales, los marcadores de eventos y las grabaciones de artefactos.

Cada sujeto dispone de: (i) `cnt.mat`, una celda 1×6 con señales continuas de intensidad; en este trabajo se utilizan únicamente `cnt{1,1}`, `cnt{1,3}` y `cnt{1,5}` (MI). (ii) `mrk.mat`, una celda 1×6 con los tiempos de inicio de ensayo y la etiqueta de condición; para MI, el marcador 1 corresponde a mano izquierda y el marcador 2 a mano derecha. (iii) `mnt.mat`, con el montaje y las etiquetas de canal. (iv) `cnt_artifact.mat` y `mrk_artifact.mat`, con registros y marcas de artefactos inducidos, usados aquí para localizar intervalos a corregir/interpolar. Los datos se trabajan a 10 Hz.

'*mnt.mat*' es la estructura que describe el montaje de canales, es decir, la correspondencia entre pares fuente–detector y posiciones sobre el cuero cabelludo; `*cnt_artifact.mat*` junto con `*mrk_artifact.mat*` son matrices de celdas 1×5 que contienen registros de artefactos de movimiento específicos (*EOG, EMG*, parpadeos, rechinamiento de dientes, apertura de boca) y sus marcas temporales.

Los 36 canales fNIRS fisiológicos (72 canales en total, 36 *HbO* y 36 *HbR* que son 2 longitudes de onda) se distribuyen en tres regiones principales: frontal, motora y visual. Para la BCI propuesta se seleccionan los 24 canales fisiológicos (48) correspondientes a la región motora (alrededor de C3 y C4), ya que son los que presentan mayor sensibilidad a la imaginación motora de mano izquierda y derecha, por tanto, estrechamente ligados a la representación cortical de las manos [31],[32],[36]. Esta selección reduce el tamaño de los datos y se alinea con la localización anatómica de la tarea.

Resumiendo, este apartado, la entrada vienen siendo los 36 canales fisilógicos fNIRS, el procesamiento está dado por el filtrado por etiquetas de canal (*mnt)* para extraer índices motores y reordenar los canales en función de hemisferio y tipo de contacto. Y la salida está dada por las matrices de intensidad de luz reducidas a 24 canales fisiológicos motores, que servirán de base para la conversión hemodinámica (*MBLL*) y etapas posteriores.

4.2 Etapa de visualización

Antes de aplicar la Ley de *Beer–Lambert modificada* y el resto del pipeline de preprocesamiento, se realizó una etapa de inspección visual y verificación de metadatos sobre las señales crudas de intensidad registradas en el archivo *cnt.mat.* El objetivo de esta etapa fue (i) comprobar la consistencia de la estructura del conjunto de datos entre sujetos, (ii) identificar de manera preliminar la presencia de artefactos y (iii) confirmar los parámetros físicos relevantes para la posterior

conversión hemodinámica (número de canales, longitudes de onda y frecuencia de muestreo), de acuerdo con la documentación del dataset [33].

4.2.1 Verificación de la consistencia de canales y etiquetas

Se comprobó la consistencia de la configuración de canales entre sujetos utilizando los archivos mnt.mat. Se verificó que todos los sujetos compartieran exactamente el mismo vector de etiquetas, comenzando por *AF7Fp1, AF3Fp1, AF3AFz, FpzFp1, …,* hasta C6FC6, lo que confirma la homogeneidad del montaje entre participantes. Esta verificación es fundamental para poder definir un conjunto fijo de canales motores (canales 12–35) que se aplicará de manera uniforme a todos los sujetos en las etapas de preprocesamiento y clasificación.

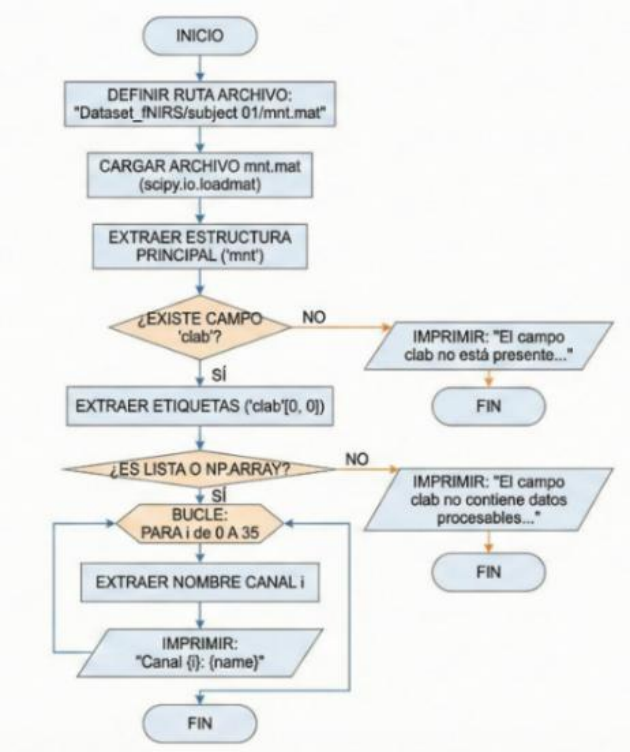


Figura 4-2. Diagrama de flujo para la extracción de etiquetas de los canales a usar

Siguiendo el diagrama de flujo de la Figura 4-2, se define la ruta del archivo *mnt.mat* para cada sujeto, se carga la estructura *mnt* y se verifica la existencia del campo *clab*. Si el campo está presente, se extrae el arreglo clab[0,0], se comprueba que sea una lista o arreglo de *NumPy* y se recorre en un bucle de 0 a 35 para imprimir el nombre de cada canal en la consola. A partir de este procedimiento se obtuvieron las etiquetas resumidas en la Tabla 4-1.

Tabla 4-1. Etiquetas de los 36 canales fNIRS (clab) del montaje NIRScout utilizado en el conjunto de datos.

Canal	Etiqueta
0	AF7Fp1
1	AF3Fp1
2	AF3AFz
3	FpzFp1
4	FpzAFz
5	FpzFp2
6	AF4AFz
7	AF4Fp2
8	AF8Fp2
9	OzPOz
10	OzO1
11	OzO2
12	C5CP5
13	C5FC5
14	C5C3
15	FC3FC5
16	FC3C3
17	FC3FC1

Canal	Etiqueta
18	CP3CP5
19	CP3C3
20	CP3CP1
21	C1C3
22	C1FC1
23	C1CP1
24	C2FC2
25	C2CP2
26	C2C4
27	FC4FC2
28	FC4C4
29	FC4FC6
30	CP4CP6
31	CP4CP2
32	CP4C4
33	C6CP6
34	C6C4
35	C6FC6

A partir de la inspección del montaje (*mnt.mat*), como se puede observar en la Tabla 4-1, se determinó que los 24 canales correspondientes al área motora (alrededor de *C3* y *C4* según las referencias del dataset) son los canales fisiológicos 12 a 35, todos ellos ubicados sobre la corteza sensoriomotora bilateral.

4.2.2 Visualización de señales crudas por segmentos motores

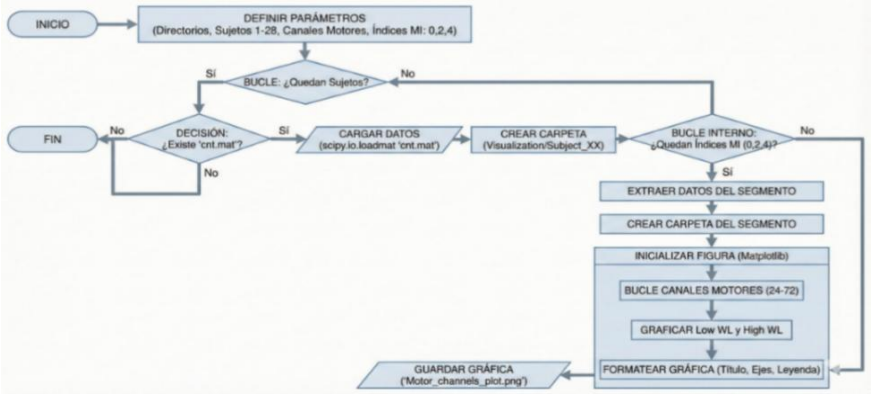


Figura 4-3. Diagrama de flujo para la visualización de señales fNIRS crudas en tareas de imaginación motora.

La Figura 4-3 muestra el diagrama de flujo del proceso implementado para la visualización de las señales crudas de imaginación motora. El procedimiento se ejecuta de forma automática sobre los 28 sujetos considerados en este trabajo.

En la matriz denominada 'x' de *cnt.mat* cada canal fisiológico se representa mediante dos columnas, una por cada longitud de onda (λ baja y λ alta), de manera que 36 canales fisiológicos dan lugar a 72 columnas [33], de los cuales 24 canales fisiológicos son los de interés al ser específicamente para actividad de imaginación motora. Para extraer estos canales en las señales de intensidad, se seleccionan las columnas 24 a 71 de la matriz x, que corresponden a las dos longitudes de onda (760 y 850 nm) de los 24 canales motores.

En el **Anexo 2** Figura A2.1 se presenta la imagen de uno de los segmentos motores para el sujeto 1, como muestra de ejemplo. Observando dicha Figura A2.1, se puede afirmar que los registros contienen combinaciones de deriva lenta, ruido de alta frecuencia y artefactos de movimiento, y motivó el diseño del pipeline de preprocesamiento descrito en las secciones anteriores, en concordancia con recomendaciones recientes y con la documentación del dataset [28]–[30], [33].

4.2.3 Confirmación de longitudes de onda y frecuencia de muestreo

Antes de aplicar la MBLL a las señales fNIRS, se verificó de forma sistemática la consistencia de las longitudes de onda y de los parámetros de adquisición reportados en los archivos *cnt.mat* de cada sujeto. Esta validación es necesaria porque la MBLL requiere conocer con precisión las longitudes de onda utilizadas para seleccionar los coeficientes de extinción de *HbO* y *HbR* adecuados; un error en estos valores introduciría sesgos en las estimaciones de *ΔHbO* y *ΔHbR* y comprometería la comparabilidad entre sujetos y sesiones [28]–[30].

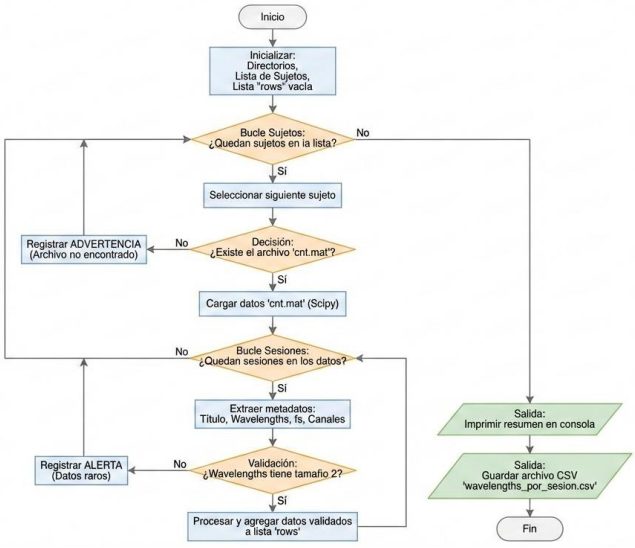


Figura 4-4. Diagrama de flujo para la verificación de wavelengths.

Siguiendo el diagrama de flujo mostrado en la Figura 4-4 (Verificación de *wavelengths*), se implementó un script en Python (check_wavelengths.py) que recorre automáticamente los 28 sujetos, carga cnt.mat y extrae por cada sesión los campos *wavelengths, fs* y *clab* (desempaquetando *object arrays* cuando es

necesario). El programa verifica que *wavelengths* tenga exactamente dos valores; si no, reporta la sesión y la excluye. Con los casos válidos registra *λ_low, λ_high, fs* y número de canales, y exporta una tabla a wavelengths_por_sesion.csv, confirmando la consistencia de los parámetros de adquisición en todo el dataset.

Tabla 4-2. Verificación de wavelengths

Parámetro	Valor encontrado	Comentario
Lambda_low	760 nm	Longitud de onda baja para todos los bloques
Lambda_high	850 nm	Longitud de onda alta para todos los bloques
fs	10 Hz	Frecuencia de muestreo del registro fNIRS
n_channels	72	36 canales fisiológicos × 2 longitudes de onda

En la Tabla 4-2, se evidencia que para las seis sesiones de cada uno de los 28 sujetos se utilizan exactamente las mismas longitudes de onda (760 y 850 nm), la misma frecuencia de muestreo (10 Hz) y el mismo número de canales de intensidad (72 columnas, correspondientes a 36 canales fisiológicos × 2 longitudes de onda), en concordancia con la descripción del sistema NIRScout reportada en el artículo del dataset [33].

4.3 Conversión MBLL y fusión de sesiones de imaginación motora

La MBLL extiende la ley clásica de Beer–Lambert a medios altamente dispersivos como el tejido cerebral, introduciendo un DPF que corrige el hecho de que la luz recorre trayectorias efectivas más largas que la distancia fuente–detector [28]–[30].

La conversión basada en la Ley de Beer–Lambert modificada (MBLL) es necesaria porque el sistema NIRScout registra intensidades de luz en dos longitudes de onda y no directamente concentraciones hemodinámicas. MBLL transforma las intensidades en cambios de concentración de oxihemoglobina y desoxihemoglobina (ΔHbO y ΔHbR), corrigiendo la longitud efectiva del camino óptico mediante el factor de camino diferencial (DPF) y empleando coeficientes de extinción molares para 760/850 nm [28]–[30].

En otras palabras, aplicar la MBLL es el paso que transforma una señal “de instrumento” en una señal hemodinámica cuantificable, directamente relacionada con la respuesta neurovascular. Esto permite, (i) interpretar las curvas como activación/inactivación cortical, (ii) extraer las características que usa la BCI (medias, pendientes, máximos de $\Delta HbO/\Delta HbR$) y (iii) seguir el estándar de la

comunidad fNIRS, que recomienda trabajar siempre en el dominio de concentraciones para análisis y clasificación de tareas [28]–[30]. Sin MBLL, todo el pipeline posterior que contiene características hemodinámicas, selección de canales motores y comparación con otros estudios no sería válido.

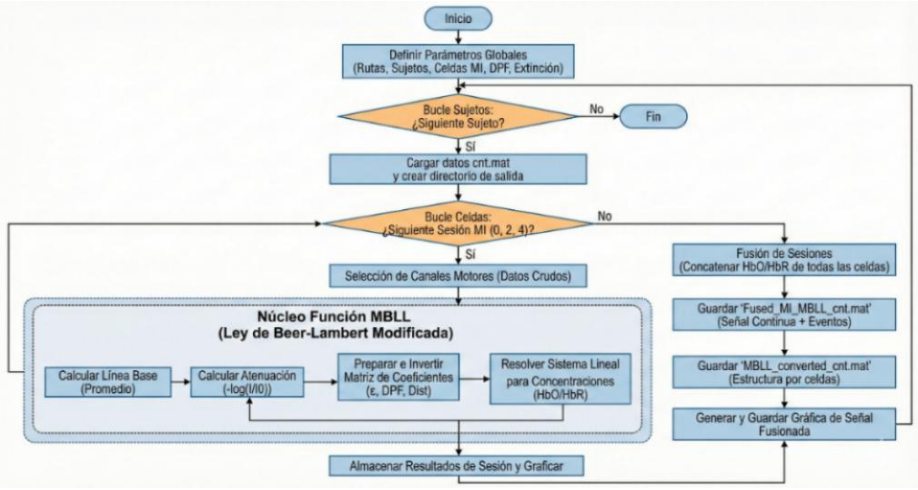


Figura 4-5. Diagrama de flujo del algoritmo de MBLL.

Siguiendo la Figura 4-5, la conversión se implementó en Python mediante una función *mbll(·)* equivalente a *proc_BeerLambert* del BBCI Toolbox [33]. La entrada es una matriz *raw_intensity* de tamaño $T \times (2N)$, donde cada canal fisiológico se representa con dos columnas consecutivas (760 y 850 nm). Para evitar inestabilidades numéricas al aplicar logaritmos, se suma un término pequeño ε (p. ej., 10^{-10}) antes del cálculo de densidad óptica.

La MBLL expresa la variación de absorbancia para cada longitud de onda λ con los cambios de concentración de oxi- y desoxihemoglobina. Primero se calcula la absorbancia relativa:

$$\Delta A(\lambda, t) = -log_{10} \frac{I(\lambda, t)}{I_0(\lambda)} \qquad [E.4.1],$$

que se aproxima como una combinación lineal de los cambios de concentración de oxihemoglobina y desoxihemoglobina,

$$\Delta A(\lambda, t) \approx \varepsilon_{\text{HbO}}(\lambda)\, \Delta[\text{HbO}](t)\, L\, \text{DPF}(\lambda) \; + \; \varepsilon_{\text{HbR}}(\lambda)\, \Delta[\text{HbR}](t)\, L\, \text{DPF}(\lambda) \qquad [E.4.2],$$

Para dos longitudes de onda λ_1y λ_2, se construye la matriz:

$$\text{E} = \begin{bmatrix} \varepsilon_{\text{HbO}}(\lambda_1) L\text{DPF}(\lambda_1) & \varepsilon_{\text{HbR}}(\lambda_1) L\text{DPF}(\lambda_1) \\ \varepsilon_{\text{HbO}}(\lambda_2) L\text{DPF}(\lambda_2) & \varepsilon_{\text{HbR}}(\lambda_2) L\text{DPF}(\lambda_2) \end{bmatrix} \qquad [E.4.3].$$

y las concentraciones relativas se obtienen como:

$$\begin{bmatrix}\Delta[\mathrm{HbO}](t)\\ \Delta[\mathrm{HbR}](t)\end{bmatrix} = \mathrm{E}^{-1}\begin{bmatrix}\Delta A(\lambda_1, t)\\ \Delta A(\lambda_2, t)\end{bmatrix} \qquad [E.4.4].$$

En la implementación, la referencia basal se estimó como el promedio temporal de la señal de intensidad dentro de cada sesión (bloque continuo de adquisición), es decir, considerando todas las muestras del segmento antes de la conversión. Esta aproximación permite obtener una referencia estable frente a fluctuaciones lentas de la señal y es común en escenarios donde no se dispone de un periodo basal explícito separado para cada ensayo.

A partir de esta referencia, se calculó la densidad óptica (cambio relativo) para cada par de columnas del mismo canal. Los coeficientes de extinción se fijaron para 760/850 nm y se escalaron por los factores DPF (5.98, 7.15) reportados para corteza adulta y separaciones de ~3 cm. La matriz de conversión se calcula una sola vez y se aplica muestra a muestra para obtener las series $\Delta HbO \; y \; \Delta HbR$ por canal [28]–[30].

La conversión se aplica únicamente a las tres sesiones de imaginación motora por sujeto (*cnt{1,1}, cnt{1,3} y cnt{1,5}), produciendo matrices* $\Delta\mathrm{HbO}_k, \Delta\mathrm{HbR}_k \in \mathbb{R}^{L_k \times 24}$ y por sesión. Las sesiones 2, 4 y 6 se han eliminado porque corresponden a actividades de aritmética mental, por lo que no son objeto de estudio para este trabajo.

Posteriormente, las tres sesiones se fusionan mediante concatenación temporal. Esta decisión tiene como objetivo aumentar la cantidad de datos disponibles por sujeto, lo cual es especialmente relevante en las señales fNIRS de este proyecto, donde el número de ensayos por sesión es limitado y la señal presenta una considerable relación señal–ruido.

Adicionalmente, la concatenación de sesiones permite capturar la variabilidad intra–sujeto entre diferentes bloques experimentales, incluyendo posibles cambios en la respuesta hemodinámica, condiciones fisiológicas y niveles de atención. De esta manera, el modelo de clasificación se entrena sobre un conjunto de datos más representativo del comportamiento del sujeto, mejorando su capacidad de generalización y reduciendo el riesgo de sobreajuste a una única sesión.

Este enfoque es consistente con esquemas de entrenamiento intra–sujeto comúnmente utilizados en BCIs, donde se busca aprovechar la mayor cantidad de datos disponibles para mejorar la robustez del modelo frente a la variabilidad de las señales fisiológicas [39],[40].

$$\Delta\mathrm{HbO}_{\mathrm{fused}} = \begin{bmatrix}\Delta\mathrm{HbO}_1\\ \Delta\mathrm{HbO}_2\\ \Delta\mathrm{HbO}_3\end{bmatrix} \in \mathbb{R}^{(L_1+L_2+L_3)\times 24} \qquad [E.4.5],$$

y de forma análoga para ΔHbR. Para mantener trazabilidad, se almacenan además los vectores *session_len=[L1,L2,L3]* y *session_onsets*=[0,\,L1,\,L1+L2], que

permiten reconstruir exactamente los límites de cada sesión original dentro de la señal fusionada. Los resultados se guardan en *MBLL_converted_cnt.mat* (por sesión) y en *Fused_MI_MBLL_cnt.mat* (señal fusionada + longitudes e índices de inicio).

En conjunto, esta etapa transforma intensidades ópticas en variables fisiológicas interpretables ($\Delta HbO/\Delta HbR$) y organiza las tres sesiones de MI como una única señal continua por sujeto, lista para las etapas posteriores de limpieza, filtrado y segmentación en épocas.

Como evidencia complementaria del proceso de fusión de sesiones y conversión a concentraciones relativas mediante MBLL, se incluyen en el **Anexo 3** Figura A3.1, la señal fusionada MBLL para el sujeto 1. Las señales MI fusionadas presentan una duración coherente con la concatenación de las tres sesiones de MI del dataset (a 10 Hz) [33]. Las líneas verticales indican los límites entre sesiones y permiten verificar que la concatenación se realizó sin pérdida ni duplicación de muestras. Este control visual también evidencia la variabilidad inter-sujeto típica de fNIRS. (Ver Anexo 1: carpeta MBLL_MI).

4.4 Corrección de artefactos en el dominio hemodinámico

Esta etapa atenúa artefactos de movimiento sobre las señales hemodinámicas fusionadas ΔHbO y ΔHbR (*Fused_MI_MBLL_cnt.mat)* usando los registros del dataset (*cnt_artifact.mat, mrk_artifact.mat*) que contienen cinco tipos inducidos (EOG, EMG, parpadeos, rechinamiento y apertura de boca) con sus *onsets.* En este contexto, el término *onset* se refiere al instante de inicio de cada ensayo de imaginación motora, obtenido a partir de los marcadores de evento en mrk.mat. [33], [41], [42], [44]–[46].

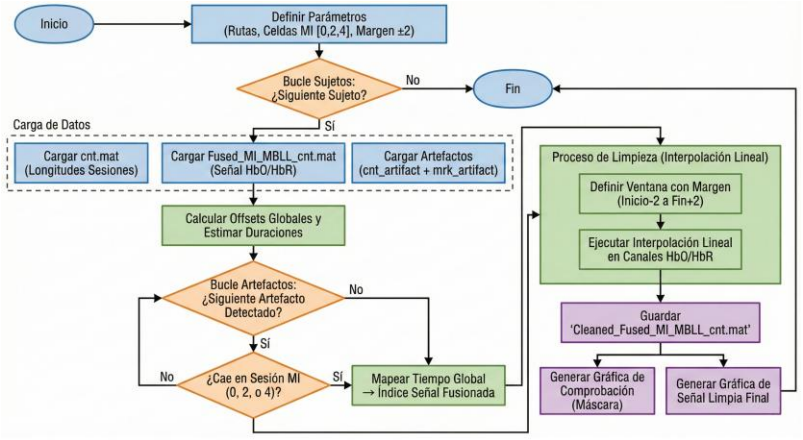


Figura 4-6. Diagrama de flujo del algoritmo de eliminación de artefactos.

Siguiendo la Figura 4-6, para cada sujeto se reconstruyó la línea de tiempo completa a partir de *cnt.mat*. Se calculó la longitud (en muestras) de las seis sesiones originales y su suma acumulada para obtener los desplazamientos de inicio de cada sesión en la señal continua. De forma análoga, se almacenaron las longitudes y desplazamientos de las tres sesiones de MI, necesarios para proyectar las marcas de artefactos sobre la señal MI fusionada y corregir los intervalos correspondientes.

A continuación, se procesan los archivos *cnt_artifact.mat* y *mrk_artifact.mat*. El primero contiene, para cada tipo de artefacto, un bloque hemodinámico con la misma frecuencia de muestreo que el experimento principal, mientras que el segundo almacena los tiempos de inicio (*time*) de cada ocurrencia de artefacto [33].

Dado que las marcas incluyen solo *onsets*, la duración por evento se aproxima de forma uniforme dentro de cada tipo de artefacto como

$$\hat{d} \approx \text{round}\left(\frac{N_t}{N_e}\right) \qquad [E.4.6],$$

donde N_t es la longitud del bloque de artefactos y N_e el número de eventos anotados [33]. Cada artefacto se proyecta a la señal MI fusionada mediante desplazamientos de sesión (con trazabilidad por longitudes e inicios acumulados) y se recorta si excede el final de la señal. La fracción de señal intervenida se cuantifica como

$$\text{coverage}_{\text{MI}} = \frac{\sum_i d_i}{T_{\text{sig}}} \times 100 \qquad [E.4.7],$$

donde T_{sig} es el número total de muestras de la señal MI fusionada. La corrección se implementa por interpolación lineal canal a canal en cada intervalo contaminado. Para el i-ésimo artefacto, el intervalo en muestras se define como,

$$s = \max\left(0,\ t_{\text{fused},i} - \text{margin}\right), e = \min\left(T_{sig},\ t_{fused,i} + d_i + margin\right) \qquad [E.4.8],$$

y *margin* es un margen pequeño. Para cada canal c, el segmento $[s,e)$ se reemplaza por la recta entre los valores de borde:

$$H_{\text{bO,cln}}[k,c] = y_0 + \frac{k-s}{(e-s)-1}\,(y_1 - y_0), k = s, \ldots, e-1 \qquad [E.4.9].$$

aplicado tanto a $H = \Delta HbO$ como $H = \Delta HbR$. Esta operación elimina picos/discontinuidades rápidas típicas de movimiento preservando continuidad y evitando distorsión relevante de la HRF (señal lenta) cuando la proporción corregida es moderada [28]–[30], [41], [44]–[46]. Como indicador global del cambio introducido se calcula, por ejemplo, la norma L_1:

$$\Delta_{L1} = \sum_{t,c} | H_{\text{bO}}(t,c) - H_{\text{bO,cln}}(t,c) | \quad [E.4.10].$$

En síntesis, la etapa de corrección de artefactos implementa un procedimiento trazable y reproducible que: reconstruye la línea de tiempo global del experimento, localiza con precisión los artefactos inducidos por EOG/EMG y otros movimientos dentro de las tres sesiones de imaginación motora, proyecta estas posiciones sobre la señal hemodinámica fusionada y reemplaza los segmentos afectados por interpolaciones lineales canal por canal, con un margen de seguridad controlado. Este enfoque se alinea con las recomendaciones metodológicas de la literatura sobre fNIRS–BCI y sobre corrección de artefactos de movimiento en espectroscopía de infrarrojo cercano [28]–[30],[41], y proporciona señales $\Delta HbO/\Delta HbR$ limpias que sirven como base para las etapas posteriores de filtrado pasa–banda, segmentación de épocas y clasificación.

Los ejemplos visuales y máscaras binarias para el sujeto 1 se reportan en el **Anexo 4**. En cada caso se presentan las tres sesiones de MI fusionadas en una única señal por sujeto, con ΔHbO (rojo) y ΔHbR $(azul)$ en los 24 canales motores, y líneas verticales punteadas que indican los límites entre sesiones. La longitud total de la señal limpia coincide con la señal fusionada antes de la limpieza (≈21k muestras a 10 Hz), lo que confirma que la corrección no elimina ni inserta muestras, sino que únicamente reemplaza por interpolación los intervalos marcados como artefacto.

De forma complementaria, la Figura A4.2 presenta la máscara binaria de artefactos (1 = muestra corregida, 0 = muestra no afectada) para el sujeto 1, permitiendo visualizar la distribución temporal y la fracción del registro intervenida. Se observa que los artefactos aparecen en bloques localizados, se concentran principalmente en la primera mitad; La consistencia de estas máscaras (alineadas con las tres sesiones y con cortes exactos en los límites) confirma que la corrección es controlada y trazable, evitando sobre–corrección y dejando las señales en un estado más estable y fisiológicamente plausible para el filtrado, la segmentación y la extracción de características posteriores. Para ver las gráficas de los demás sujetos ver **Anexo 1** (carpeta *Cleaned_Fused_MI*).

4.5 Filtrado pasa–banda hemodinámico

En la tercera etapa se aplica un filtrado pasa–banda para suprimir derivas lentas y ruido fuera de banda no asociado a la respuesta hemodinámica, sobre las señales ΔHbO y ΔHbR ya convertidas y corregidas por artefactos. Se emplea un filtro *Chebyshev* tipo II de orden 3 con banda de paso 0.01–0.09 Hz y atenuación en banda de rechazo de 30 dB, rango coherente con la dinámica hemodinámica dominante en fNIRS (≈0.01–0.1 Hz) y con recomendaciones metodológicas en fNIRS/fNIRS-BCI; la elección de Chebyshev II se justifica por su banda de paso monótona y alto rechazo fuera de banda con bajo orden, lo que reduce costo computacional y preserva la forma hemodinámica; además, es consistente con

prácticas reportadas para este *dataset* y *toolboxes* tipo BBCI [28]–[30], [33], [41], [48].

En cuanto al orden del filtro, se selecciona un orden 3 como un compromiso entre una transición suficientemente abrupta y la estabilidad numérica del sistema. Órdenes más altos podrían mejorar la selectividad, pero aumentan el riesgo de amplificación de ruido y artefactos, además de incrementar la sensibilidad a efectos de borde en señales cortas como las empleadas en fNIRS. En este contexto, un orden bajo combinado con aplicación en fase cero (*sosfiltfilt*) permite obtener un filtrado efectivo sin distorsionar la dinámica temporal de la respuesta hemodinámica.

La Figura 4-7 presenta el diagrama de flujo del diseño y aplicación del filtro.

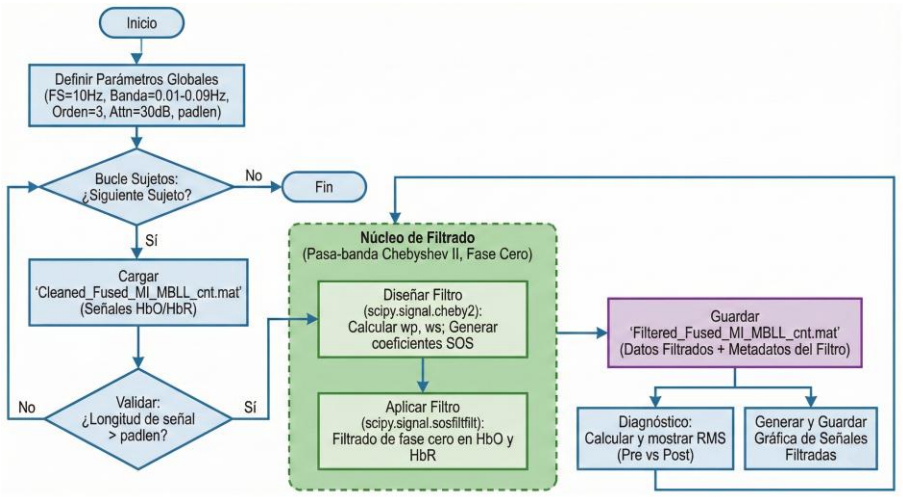


Figura 4-7. Diagrama de flujo del algoritmo de filtrado Chebyshev.

Para $f_s = 10$ Hz, se normaliza la banda respecto a *Nyquist* $f_N = f_s/2$ y se diseña el filtro con cheby2, definiendo además una banda de rechazo más amplia para garantizar el nivel de atenuación. El comportamiento en magnitud del Chebyshev II puede expresarse como

$$| H(j\omega) |^2 = \frac{1}{1 + \frac{1}{\epsilon^2}\, C_n{}^2 \left(\frac{\omega_s}{\omega}\right)} \qquad [E.4.11],$$

donde $C_n(\cdot)$es el polinomio de Chebyshev de orden n, $\omega_s$es la frecuencia rechazo y ϵ se relaciona con la atenuación exigida [41]. El filtro se implementa en secciones de segundo orden (SOS) por estabilidad numérica y se aplica con *sosfiltfilt* (filtrado hacia adelante y atrás) para obtener fase cero, equivalente a:

$$y[n] = (h[n] * h[-n]) * x[n] \qquad [E.4.12],$$

Lo que evita retardos y distorsión temporal en la morfología de $\Delta HbO/\Delta HbR$ [30],[41]. Se usa *padlen* para mitigar transitorios en bordes [47], [48]. El

procedimiento se ejecuta por sujeto y por canal, verificando la longitud mínima requerida y calculando RMS antes/después como diagnóstico de supresión de energía fuera de banda sin colapsar la señal.

Resultado (Sujeto 1). En la Figura A5.1 (**Anexo 5**) se observa que, tras el filtrado, $\Delta HbO \ y \ \Delta HbR$ quedan centradas alrededor de cero y dominadas por oscilaciones lentas, mientras disminuyen las componentes muy lentas (<0.01 Hz) y las rápidas (>0.09 Hz) asociadas a ruido fisiológico o residual de movimiento [28]–[30], [41]. Esto facilita la segmentación en épocas y la extracción de características temporales para clasificación. Las señales filtradas se almacenan en *Filtered_Fused_MI_MBLL_cnt.mat* junto con los parámetros del filtro para reproducibilidad. Para ver más resultados de los demás sujetos, ver **Anexo 1.**

4.6 Segmentación de épocas y construcción de ventanas de tarea/reposo

En esta etapa se toma como entrada las señales hemodinámicas ya filtradas y fusionadas $(\Delta HbO \ y \ \Delta HbR)$ y los marcadores de eventos almacenados en *mrk.mat*, con el fin de construir épocas alineadas por ensayo de imaginación motora (MI). De los seis bloques experimentales que tiene cada sujeto de la base de datos, se usan únicamente los de MI (1, 3 y 5), excluyendo los de aritmética mental (2, 4 y 6) que no son el objeto de estudio para este proyecto, según la estructura del dataset [33]. Los tiempos de evento (ms) se convierten a índices de muestra mediante

$$n = \mathrm{round}\left(t_{\mathrm{ms}} \frac{f_s}{1000}\right), \quad f_s = 10\ Hz \qquad [E.4.13],$$

Como la señal MI está previamente fusionada (tres sesiones concatenadas), los onsets deben trasladarse a la línea temporal global. Para ello se calculan desplazamientos acumulados por sesión y se obtiene el onset fusionado como:

$$t_{\mathrm{fus}} = t_{\mathrm{rel}} + \mathrm{offset}_k^{\mathrm{MI}} \qquad [E.4.14],$$

Este procedimiento (resumido en el diagrama de flujo de la Figura 4-8) incluye: carga de señal filtrada, lectura de *mrk*, conversión a muestras, corrección por *offsets*, extracción de épocas, *baseline*, recorte de *subventanas* y almacenamiento de tensores.

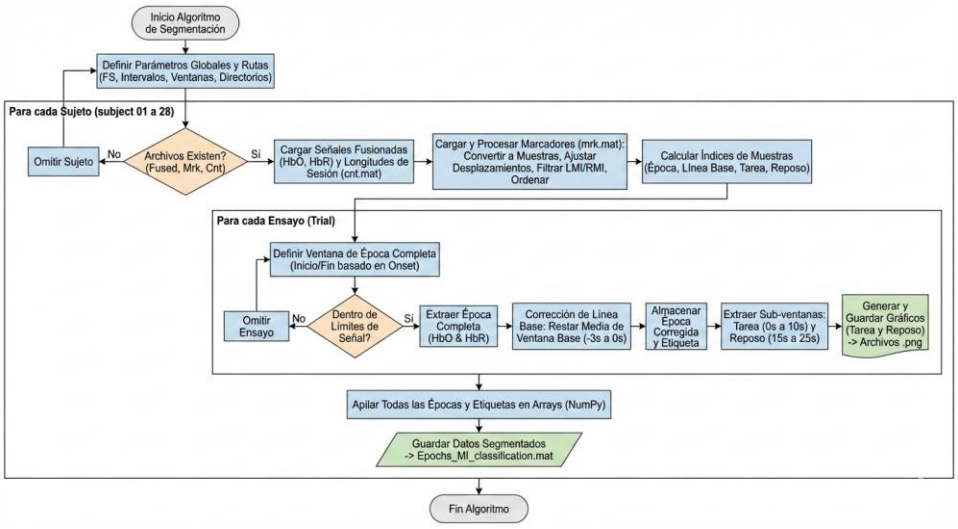


Figura 4-8. Diagrama de flujo del algoritmo de segmentación.

Cada evento genera una época amplia para capturar pre–reposo, respuesta y retorno:

$$I_{\text{epo}} = [-10,\ 25]\ \text{s} \ \rightarrow 350\ \text{muestras a}\ 10\ \text{Hz} \qquad [E.4.15],$$

y se aceptan únicamente épocas completas dentro de la señal (si no, se descartan) [33]. Una época se considera completa si el intervalo definido alrededor del *onset* se encuentra completamente contenido dentro de la señal disponible. Específicamente, para un *onset* en la muestra *n_onset*, la época es válida si se cumple que:

$$n_{onset} - \ 10 \cdot f_s \ \geq \ 0 \ \ y \ \ n_{onset} \ + \ 25 \cdot f_s \ \leq \ N \qquad [E.4.16],$$

donde N es la longitud total de la señal fusionada. En caso contrario, la época se descarta, tal como se implementa en el código mediante la verificación de límites *(s < 0 o e > N)*, evitando así la inclusión de ventanas truncadas que podrían distorsionar la estimación de la respuesta hemodinámica.

A cada época se le aplica corrección de línea base usando [−3,0] s:

$$\tilde{x}^{(j)}_{\text{HbO}}(n,c) = x^{(j)}_{\text{HbO}}(n,c) - \frac{1}{|\ \mathcal{B}\ |}\sum_{m\in\mathcal{B}} \text{x}^{(\text{j})}(m,c)\,, \quad B = [-3{,}0] \qquad [E.4.17],$$

(aplicado a $\Delta HbO\ y\ \Delta HbR$) , práctica estándar para centrar ensayos y reducir desplazamientos lentos residuales [28]–[30].

Finalmente, de cada época se recortan dos subventanas fijas por ensayo: tarea *[0,10] s* y reposo *[15,25] s*, preservando una comparación homogénea dentro del mismo ensayo (tarea vs recuperación tardía), coherente con la cinética lenta de la HRF [30], [34]. El resultado es se guarda en *Epochs_MI_classification.mat* como

tensores (ensayo × tiempo × canal) y etiquetas binarias (LMI/RMI) obtenidas desde mrk.y [33].

Las etiquetas de clase se obtienen directamente de la estructura *mrk.y* del dataset, la cual codifica cada ensayo mediante un esquema *one-hot* ([1,0] = clase 1 y [0,1] = clase 2). En el caso de los bloques de imaginación motora, la clase 1 corresponde a imaginación de mano izquierda (LMI) y la clase 2 a imaginación de mano derecha (RMI), según la convención del dataset [33]. Al reposo no se le asigna una etiqueta, ya que no será tenido en cuenta para la clasificación.

Para su uso en el pipeline de clasificación, esta representación se convierte a etiquetas escalares aplicando la función *argmax* sobre cada vector de clase, de modo que cada época queda asociada de forma unívoca a su condición experimental (LMI o RMI). Esta asignación es consistente con la organización de los ensayos (30 por clase) y se preserva durante todas las etapas posteriores del procesamiento.

En el **Anexo 6** (Figura A6.1) se muestran ejemplos de LMI y reposo del sujeto 1. En tarea *(0–10 s)* se aprecia la dinámica lenta esperada y la anticorrelación HbO/HbR en varios canales; en reposo (15–25 s) se observa, en general, un comportamiento más estable o de retorno hacia basal, respaldando el recorte tardío como referencia dentro del mismo ensayo [28]–[30], [34].

4.7 Extracción y selección de características

Debido a que el número de características generadas por ensayo puede crecer rápidamente al considerar múltiples canales y descriptores temporales, se implementó un proceso de selección de características con el objetivo de reducir la dimensionalidad del problema y mejorar la capacidad de generalización de los clasificadores.

Este proceso se realizó en dos niveles. En primer lugar, se aplicó una preselección de canales basada en el coeficiente de separabilidad r², seleccionando aquellos canales con mayor capacidad discriminativa entre imaginación motora izquierda y derecha. Posteriormente, dentro de los pipelines de clasificación se aplicaron métodos embebidos de selección de características, tales como *RFECV*, *RFE*, *Boruta* y *SelectFromModel*, que permiten identificar automáticamente las variables más relevantes durante el entrenamiento del modelo.

La optimización de estas selecciones se realizó dentro del esquema de validación cruzada anidada, utilizando *balanced accuracy* como métrica de optimización.

A partir de las épocas hemodinámicas previamente segmentadas por ensayo en el intervalo [-10, 25] s (Sección 4.6), y de las subventanas de tarea [0,10] s y reposo [15,25] s definidas dentro de cada época, esta etapa de extracción y selección de características transforma cada ensayo de imaginación motora en un vector de características compacto.

En BCI–fNIRS se privilegian descriptores temporales (p. ej., media, pendiente, varianza y pico) porque capturan directamente la morfología de $\Delta HbO/\Delta HbR$, mantienen interpretabilidad fisiológica y suelen ser competitivos con bajo costo computacional en regímenes de pocas muestras, frente a enfoques espectrales más complejos [4], [5], [31], [33], [36]. Esta decisión es coherente con revisiones que reportan el uso frecuente y efectivo de características temporales en tareas motoras y cognitivas con fNIRS [4], [5].

Cada época puede modelarse como

$$\mathbf{E}^{(k)} \in \mathbb{R}^{T\times C\times 2} \qquad [E.4.18],$$

Donde T = *350 muestras*, $C = 24$ los canales motores y el último eje corresponde a los dos cromóforos (ΔHbO, ΔHbR). La extracción se realiza sobre un catálogo de ventanas $W = [t_{\min}, t_{\max}]$ (p, ej., B: [0,10] s, C: [5,8] s, D: [10,13] s, E: [15,18] s, F: [16,19] s, etc.) definido para cubrir *subfases* de la HRF (ascenso, meseta/pico y retorno), dada la latencia y dinámica lenta del acoplamiento neurovascular en fNIRS [29], [30]. Para cada ventana se define la máscara temporal

$$m_W = \{i \in \{0, \dots, T-1\} : t_i \in W\} \qquad [E.4.19],$$

Y se restringe la época a dichas muestras. Sobre cada canal y ventana se calculan descriptores mínimos pero completos: nivel (media), tendencia (pendiente), extremos (pico), dispersión (varianza) y el contraste ($\Delta HbO - \Delta HbR$), que captura explícitamente la anticorrelación típica entre cromóforos durante activación cortical. Estas magnitudes resumen diferencias entre condiciones (LMI/RMI) en amplitud, dinámica y estabilidad de la HRF, y mantienen una lectura fisiológica directa [29]–[31].

En fNIRS, la respuesta hemodinámica no es instantánea típicamente la variación en HbO/HbR aparece con retardo respecto al inicio del estímulo, alcanza un máximo a los pocos segundos y el retorno a línea base puede extenderse varios segundos más [30]. En particular, se reporta que la señal puede presentar un retardo total del orden de decenas de segundos en volver completamente a basal y que *HbO* y *HbR* pueden cambiar incluso en direcciones opuestas dependiendo del balance local de aporte/consumo, razón por la cual conviene muestrear ventanas que cubran tanto ascenso, meseta/pico como retorno [29], [30].

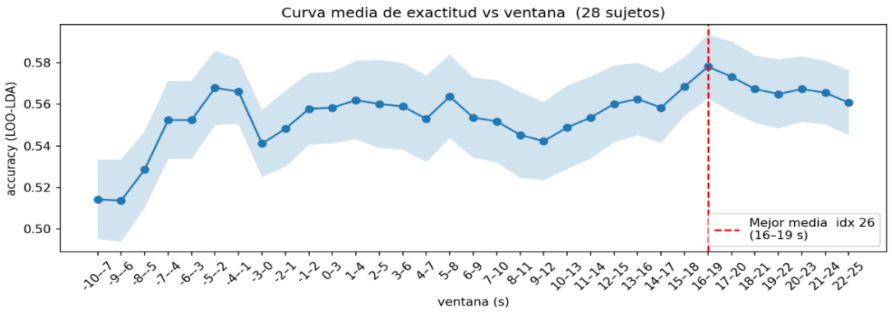


Figura 4-9. Análisis de exactitud de ventanas globales para todos los sujetos.

La Figura 4-9 se obtuvo mediante una evaluación ventana a ventana sobre los ensayos de MI (LMI vs. RMI) de cada sujeto, usando *Leave-One-Out (LOO)* con un clasificador lineal de referencia (LDA). Para cada sujeto se cargaron: (i) una matriz de características por ensayo y por ventana, (ii) las etiquetas, y (iii) los límites temporales de cada ventana. Se emplearon 33 ventanas deslizantes de ancho fijo (3 s) y paso de 1 s que cubren el intervalo completo del ensayo (por ejemplo, desde [−10,−7] hasta [22,25] s), de modo que todas las ventanas se comparan bajo el mismo ancho temporal [31], [36].

Para cada ventana se calculó la exactitud *LOO*: se entrenó el clasificador con 59 ensayos y se probó en el ensayo restante, repitiendo 60 iteraciones y promediando el porcentaje de aciertos.

$$\mathrm{Acc}(w) = \frac{1}{60}\sum_{\mathrm{j}=1}^{60} 1(\hat{\mathrm{y}}_{\mathrm{j}}^{(\mathrm{w})} = \mathrm{y}_{\mathrm{j}}) \qquad [E.4.20],$$

Así, para cada sujeto se obtuvo un vector de exactitudes (una por ventana). Luego se promediaron los 28 vectores para construir la curva grupal (promedio inter-sujeto) usada en la figura.

$$\overline{\mathrm{Acc}}(w) = \frac{1}{28}\sum_{\mathrm{s}=1}^{28} \mathrm{A}_{\mathrm{s,w}} \qquad [E.4.21],$$

$$\mathrm{SEM}(w) = \frac{\mathrm{std}(\mathrm{A}_{\mathrm{s,w}})}{\sqrt{28}} \qquad [E.4.22].$$

En la figura, la línea corresponde al promedio inter-sujeto y el sombreado a ±1 desviación estándar. La línea punteada roja marca la ventana con mayor exactitud media (en este análisis, [16,19] s), utilizada únicamente como diagnóstico para definir un catálogo inicial de ventanas.

Este análisis con LDA se usa solo como diagnóstico para justificar incluir ventanas tempranas y tardías en el catálogo; no fija hiperparámetros del modelo final. La selección definitiva de ventana y canales se realiza después mediante un esquema

anidado y orientado por datos dentro de la validación por sujeto (un esquema *data-driven)*, evitando sesgos por selección previa [39], [59]. En general, mejores ventanas tardías son coherentes con la cinética lenta de la respuesta hemodinámica en fNIRS [30], [31].

La respuesta hemodinámica (HRF) en señales fNIRS presenta una dinámica temporal característica asociada al acoplamiento neurovascular. Tras el inicio del estímulo, la señal no cambia de forma inmediata, sino que existe una fase inicial de retardo seguida de un ascenso progresivo en ΔHbO (y descenso en ΔHbR), una fase de meseta o pico, y finalmente una fase de retorno hacia la línea base. Estas sub-fases pueden extenderse durante varios segundos, dependiendo de la tarea y del sujeto, y constituyen la base para el análisis temporal de la señal [29], [30].

En este contexto, la selección de ventanas temporales no se realiza de forma arbitraria, sino que busca capturar distintas porciones de la HRF, incluyendo la fase de ascenso (respuesta temprana), la región cercana al pico (máxima discriminabilidad) y la fase de retorno (dinámica tardía).

Con base en esta dinámica, el catálogo de ventanas se diseñó para cubrir sub-fases relevantes de la HRF: tarea completa [0,10] s, fase temprana [5,8] s (ascenso), transición inmediata [10,13] s (cercana al pico), y ventanas media-tardías/post-tarea [15,18] s y [16,19] s (retorno hacia la línea base). Adicionalmente, se incluyen ventanas largas tipo integración (p. ej., [2,12]–[5,15]) para estabilizar estimaciones cuando el SNR por ensayo es bajo, aprovechando la naturaleza lenta de la señal hemodinámica [30], [33].

Sobre la ventana seleccionada y cada canal motor se calculan las siete características temporales que alimentan las etapas posteriores de selección y clasificación.

Para cada ensayo k, canal motor c y ventana W (máscara m_W), se extrae un conjunto compacto de descriptores temporales que resume la forma de $\Delta HbO/\Delta HbR$ y conserva interpretabilidad fisiológica [4], [5], [29]–[31]. En particular, se calculan media y pendiente media de *HbO*:

$$\mu_O^{(k,c)} = \frac{1}{\mid m_W \mid} \sum_{i \in m_W} O_{i,c}^{(k)} \qquad [E.4.23],$$

$$s_O^{(k,c)} = \frac{1}{\mid m_W \mid -1} \sum_{i \in m_W} \left(O_{i+1,c}^{(k)} - O_{i,c}^{(k)} \right) \qquad [E.4.24],$$

Se calcularon métricas sobre ΔHbO y, de forma análoga, sobre ΔHbR. Estas características capturan nivel y tendencia de la respuesta: típicamente, activación cortical se asocia a incrementos en ΔHbO, decrementos (o menor incremento) en ΔHbR y pendientes coherentes con el ascenso/retorno hemodinámico, mientras que

valores cercanos a cero o tendencias opuestas sugieren baja respuesta o mayor ruido [30], [31].

Como medidas de amplitud extrema y estabilidad intra-ventana, se calcula el pico de HbO y su varianza

$$p_O^{(k,c)} = \max_{i \in m_W} O_{i,c}^{(k)} \qquad [E.4.25],$$

$$\sigma_O^2\,(k,c) = \frac{1}{\mid m_W \mid -1} \sum_{i \in m_W} \left(O_{i,c}^{(k)} - \mu_O^{(k,c)} \right)^2 \qquad [E.4.26],$$

donde cambios sistemáticos en pico/variabilidad entre LMI y RMI reflejan diferencias en fuerza y consistencia de activación [4], [5], [31]. Se usa σ_O^2 como indicador simple de estabilidad manteniendo baja dimensionalidad; aun así, se incorpora *HbR* y el contraste *HbO–HbR* para robustez fisiológica frente a contribuciones sistémicas [29], [30].

Finalmente, se incluye el contraste neurovascular promedio:

$$d^{(k,c)} = \frac{1}{|m_W|} \sum_{i \in m_W} \left(O_{i,c}^{(k)} - R_{i,c}^{(k)} \right) \qquad [E.4.27],$$

que resume la anticorrelación esperada (incremento de *HbO* con disminución de *HbR*), asociada de forma robusta a activación local [30], [31]. En conjunto, estas características producen un vector de dimensión $7 \times C$ por ensayo; para evitar redundancia y ruido al usar los 24 canales motores, se aplica selección de canales con r^2 firmado, un índice de separabilidad ampliamente usado en BCI (p. ej., BBCI toolbox) y relacionado con medidas de correlación en escenarios binarios [6], [33], [39].

Las siete características temporales resumen los grados de libertad principales de la HRF por canal: nivel promedio (media), dinámica de cambio (pendiente), amplitud máxima (pico), variabilidad intra-ventana (varianza) y el contraste neurovascular ΔHbO−ΔHbR. Con ello se mantiene un balance entre baja dimensionalidad e interpretabilidad fisiológica [29]–[31].

La selección inicial de canales se guía por el r^2 firmado, calculado en cada tiempo t y canal c a partir de las medias y varianzas por clase (1: RMI, 2: LMI):

$$r^2(t,c) = \frac{(\mu_1(t,c) - \mu_2(t,c))^2}{s_1^2(t,c) + s_2^2(t,c) + \varepsilon}\ \mathrm{sign}\big(\mu_1(t,c) - \mu_2(t,c)\big) \qquad [E.4.28],$$

donde ε regulariza el denominador. En la implementación $r^2(t,c)$ se calcula únicamente sobre ΔHbO por su mayor amplitud/SNR a nivel de ensayo; ΔHbR e conserva en el vector final (medias, pendientes y contraste) [29]–[31].

El código promedia | r^2 | sobre la ventana candidata W,

$$| \bar{r^2} | (c; W) = \frac{1}{| m_W |} \sum_{i \in m_W} | r^2(t_i, c) | \qquad [E.4.29],$$

y se seleccionan canales con| $\bar{r^2}$ | $(c; W) \geq \tau$, explorando $\tau \in \{0.05, 0.04, 0.03\}$ (***grid search***). Este rango bajo–moderado es apropiado en fNIRS single-trial porque los tamaños de efecto suelen ser modestos por ruido fisiológico y variabilidad inter-ensayo; barrer τ permite balancear "pocos canales muy fuertes" vs "más canales moderados" sin caer en selecciones vacías o excesivas [7], [37], [29], [30]. El umbral no se fija manualmente: se optimiza dentro de validación cruzada anidada, y si la selección queda por debajo de un mínimo se completa con los canales de mayor $|r^2|$.

Los umbrales τ se restringen a una rejilla corta *{0.03, 0.04, 0.05}* conservadora para datos fNIRS a nivel de ensayo, donde los tamaños de efecto pueden ser porque, en fNIRS a nivel de ensayo único, la separabilidad canal-tiempo medida por |r²| tiende a ser modesta debido a ruido fisiológico y variabilidad inter-ensayo; por tanto, barrer τ en un rango bajo-moderado permite explorar el compromiso "pocos canales muy fuertes" (riesgo de sobreajuste a canales inestables) vs "más canales moderados" (introduciendo redundancia y ruido) sin caer en (i) selecciones vacías (τ demasiado alto) ni (ii) selecciones excesivas (τ demasiado bajo) [29], [30].

Para evitar soluciones degeneradas donde la selección se concentre en un solo hemisferio, se impone una restricción de simetría: se exige un mínimo de $K_\min = 6$ canales totales y al menos $MIN_HEMI = 3$ por hemisferio, garantizando contribución bilateral alrededor de C3/C4. Esto se sustenta (i) neurofisiológicamente, porque la discriminación LMI vs RMI depende de patrones relativos contralaterales/ipsilaterales y puede involucrar activación bilateral [31], [33], y (ii) estadísticamente, porque reduce la dependencia de un único canal potencialmente inestable y controla dimensionalidad (de 168 variables brutas a ≥42 antes de la selección fina) [29]–[31].

Con el subconjunto de canales fijado por split, se aplica selección de características integrada a los pipelines (*RFECV/RFE/Boruta/SelectFromMode*l), optimizando *Balanced Accuracy*:

$$\text{BA} = \frac{1}{2}(\text{TPR}_1 + \text{TPR}_2) \qquad [E.4.30],$$

Priorizando equilibrio entre sensibilidades de ambas clases [54], [55], [59], [60]. El resultado es, por sujeto, una matriz $\mathrm{X} \in \mathbb{R}^{N \times P'}$ con $P' \ll 7 \times 24$ características seleccionadas y trazabilidad de los canales que originan cada variable [54], [55].

El diagrama de violín del sujeto 1 (**Anexo 7**, Fig. A7.1) muestra la distribución de las siete características por clase: el ancho indica densidad (variabilidad inter-ensayo) y el solapamiento entre clases anticipa ambigüedad. En este sujeto se aprecia que *O_mean* y *O_minus_R_mean* tienden a presentar desplazamientos entre clases, aunque con solapamiento moderado, sugiriendo que la separabilidad proviene más de cambios de nivel/contraste *HbO–HbR* que de dinámicas rápidas (pendientes pequeñas). Las colas y dispersión reflejan variabilidad inter-ensayo, lo que justifica que, cuando las distribuciones se solapan, convenga un esquema de decisión selectiva basado en confianza para evitar decisiones forzadas en ensayos ambiguos.

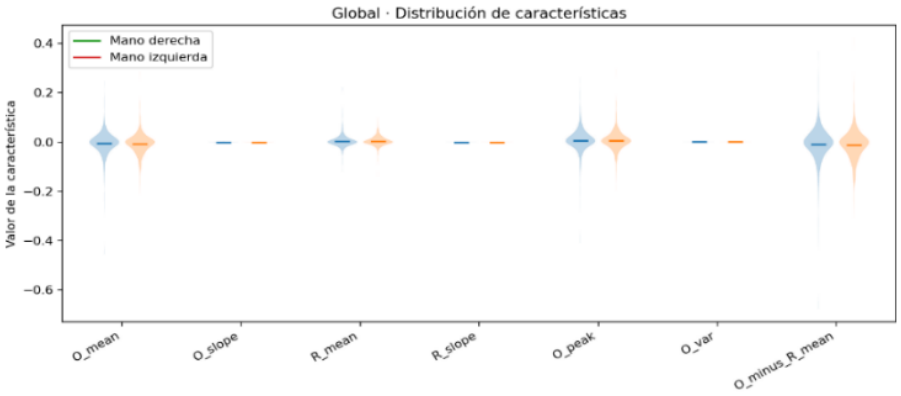


Figura 4-10. Diagrama de violín de características globales para todo el dataset.

En la Figura 4-10 se observa un solapamiento marcado entre clases, esperable en fNIRS a nivel de ensayo único por variabilidad inter-sujeto y diferencias de calidad por canal. En este contexto, las variables asociadas a amplitud/nivel y al contraste *HbO–HbR* (especialmente *O_mean, O_peak* y *O_minus_R_mean*) tienden a aportar mayor discriminación, mientras que *O_slope* y *R_slope* tienden a valores cercanos a cero y aportan información más sutil, fuertemente dependiente de la ventana temporal. Estos resultados respaldan (i) realizar selección anidada de ventana/canales por sujeto y (ii) incorporar rechazo basado en probabilidades calibradas para evitar decisiones forzadas en ensayos ambiguos.

4.8 Comité de clasificadores y Stacking con clasificación selectiva (rechazo) y calibración

La etapa final implementa un esquema de clasificación con ensamble y stacking para discriminar LMI vs. RMI. Se combinan varios expertos pertenecientes a diferentes familias de modelos de aprendizaje automático, incluyendo modelos

lineales (LDA), modelos basados en margen (SVM), métodos de ensamble basados en árboles (Random Forest y XGBoost) y redes neuronales (MLP). Esta diversidad de modelos permite capturar distintos patrones en los datos: relaciones lineales, fronteras no lineales y estructuras complejas en el espacio de características.

La combinación de modelos de distintas familias incrementa la robustez del sistema frente a la variabilidad intra–sujeto y al ruido inherente de las señales fNIRS, ya que cada modelo aporta una perspectiva complementaria del problema. El esquema de stacking permite integrar estas salidas en un meta-clasificador que aprende a ponderar sus contribuciones de forma adaptativa a partir de salidas probabilísticas. Además, se incorpora calibración y una opción de rechazo basada en confianza para abstenerse en ensayos ambiguos, priorizando decisiones confiables sobre decisiones forzadas [56]–[58].

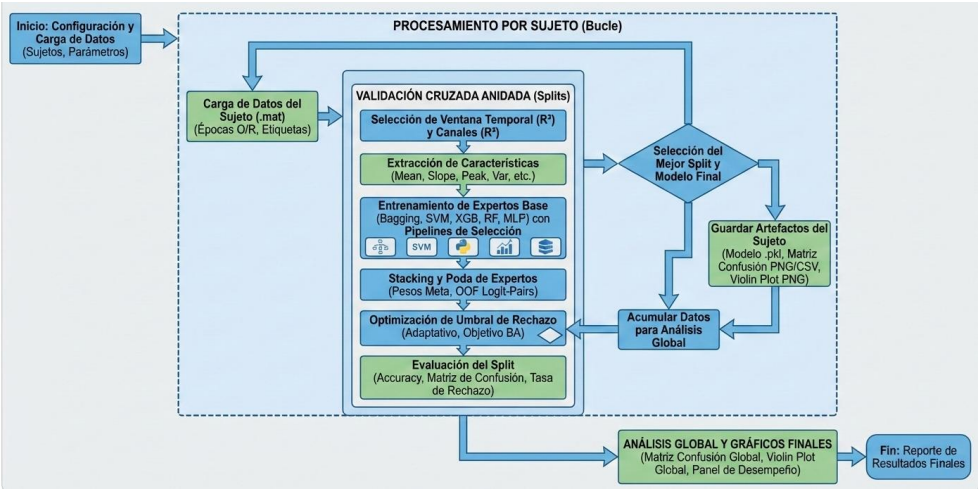


Figura 4-11. Diagrama de flujo del comité de clasificadores.

El diagrama de la Figura 4-11 resume el procesamiento, la validación cruzada anidada (selección de ventana/canales), el entrenamiento de expertos, el *stacking*, la optimización de umbral de rechazo y la exportación de artefactos y métricas para análisis global. dicho algoritmo opera por sujeto (bucle sobre 28 sujetos) y, dentro de cada sujeto, repite el proceso de entrenamiento-evaluación sobre múltiples particiones de datos para obtener estimaciones estables del desempeño y seleccionar un modelo final exportable.

4.8.1 División del conjunto de datos: 15 splits estratificados y evaluación hold-out

Para cada sujeto se dispone de N vectores de características $\mathrm{X} = [\mathrm{x}_1, \ldots, \mathrm{x}_N]^\top$ y etiquetas binarias $y_n \in \{1,2\}$(mano derecha, mano izquierda) [33]. Dado que N es reducido, el desempeño puede variar significativamente dependiendo de qué ensayos caigan en entrenamiento o prueba. Por ello, el procedimiento inicia con una partición repetida mediante *StratifiedShuffleSplit:* se generan $N_{splits} = 15$

particiones "*train–test*" estratificadas, cada una con 75% (45 ensayos) entrenamiento y 25% (15 ensayos) prueba (*hold-out*), manteniendo la proporción de clases en ambos subconjuntos [49]. [59], [60]. Esta estrategia produce una estimación más robusta que un único *hold-out*, al promediar implícitamente sobre diferentes particiones y reducir la varianza asociada a una sola división.

Se fija $N_{splits} = 15$ como un compromiso práctico entre (i) estabilidad estadística y (ii) costo computacional. En conjuntos pequeños, la métrica en un único *hold-out* puede variar sustancialmente según qué ensayos caen en entrenamiento o prueba; por ello, repetir el *hold-out* varias veces permite caracterizar la variabilidad inducida por la partición y reportar media ± desviación estándar, lo que produce conclusiones más robustas que una sola división [59], [60]. Elegir 15 repeticiones proporciona un muestreo suficiente de particiones para estimar esa dispersión sin volver prohibitivo el tiempo de entrenamiento (especialmente al incluir selección anidada e *ensembles*), manteniendo un equilibrio recomendado en prácticas de validación repetida cuando se requiere robustez y trazabilidad de la incertidumbre por partición [49], [59], [60].

En cada Split, el conjunto de entrenamiento $D_{tr}^{(s)}$ tiene 45 ensayos, y el conjunto de prueba $D_{te}^{(s)}$ cuenta con 15 ensayos. El *hold-out* se usa exclusivamente para evaluación externa del *split.* Todas las decisiones de selección de hiperparámetros (ventana temporal, canales y selección de características) se realizan dentro del entrenamiento, mediante validación cruzada anidada, evitando fuga de información (*data leakage*) y sesgos optimistas en el desempeño estimado [59], [39].

4.8.2 Validación cruzada anidada: selección conjunta de ventana temporal y canales

Dentro de cada split, el conjunto de entrenamiento se somete a una validación cruzada anidada de $K = 5$ pliegues (*StratifiedKFold*) para seleccionar la ventana temporal W, el subconjunto de canales y las variables, optimizando *balanced accuracy* (BA) [39], [59], [60]. Para evitar estimaciones degeneradas en pliegues muy pequeños, si falta una clase en algún *fold* se asigna *recall neutral (0.5).* La selección embebida de variables (*RFECV*) se ejecuta exclusivamente dentro del entrenamiento de cada fold mediante una validación interna adicional, evitando fuga de información.

En cada *fold* (*train/validación*) se realiza: (1) estimación de separabilidad firmada r^2por canal y tiempo; (2) promedio de separabilidad por ventana candidata; (3) umbral para preselección de canales; (4) salvaguardas espaciales (simetría hemisférica y mínimo alrededor de C3/C4); (5) estandarización y selección embebida (*RFECV*) dentro de entrenamiento; y (6) puntuación con un modelo de referencia, eligiendo la configuración que maximiza BA en validación. Finalmente, la ventana del *split* se fija por moda y el subconjunto final de canales

por votación (≥ 50% de los folds), manteniendo el conjunto de prueba externo completamente aislado.

El esquema de particionamiento (15 *splits hold-out* estratificados) y la validación cruzada anidada interna se resumen en la Tabla 4-3

Tabla 4-3. Esquema de validación y particionamiento (lo de "15 splits", nested-CV y hold-out).

Componente	Configuración en código	Propósito / justificación
Splits externos (hold-out repetido)	*StratifiedShuffleSplit(n_splits=15, test_size=0.25, random_state=SEED)*	Estimar desempeño con menor varianza que un único hold-out en N pequeño; el test de cada split queda "sellado" (sin decisiones de selección) para evitar leakage. [59]
Nested-CV interno (selección ventana/canales)	*StratifiedKFold(n_splits=5, shuffle=True, random_state=SEED) dentro del train*	Seleccionar hiperparámetros (ventana + canales + features) usando solo train; reduce sobreajuste por *tuning* y da estimación interna (CV_mean ± CV_std). [59], [39]
Selección de ventana	*Catálogo CATALOG + VALID_KEYS y máscaras temporales sobre t_sec*	Recorre sub-fases de HRF (tarea + post-tarea); coherente con latencia de HRF y con el protocolo del dataset. [30], [31], [33]
Selección de canales (r^2 firmado + umbral)	*R2_CANDS=[0.05,0.04,0.03], K_MIN=6 (mejor en rendimiento), MIN_HEMI=3*	Filtra canales discriminativos manteniendo cobertura mínima bilateral para MI izquierda/derecha; evita soluciones degeneradas y dependencia de un solo canal. [6], [31], [33]
Métrica de optimización	*balanced_accuracy*	Robusta en escenarios con particiones pequeñas y/o desempeño asimétrico por clase; recomendada en ML aplicado cuando interesa equilibrio por clase. [59], [60]

La Tabla 4-3 describe el esquema de evaluación empleado. Este diseño separa estrictamente la selección de *hiperparámetros* de la evaluación final del *split*, reduciendo el riesgo de *data leakage* y proporcionando estimaciones más estables en escenarios de tamaño muestral reducido [59]. La optimización se realiza con BA para evitar sesgos por clase en particiones pequeñas y para priorizar desempeños equilibrados, lo cual es consistente con prácticas de evaluación en aprendizaje automático aplicado y BCI [39], [59], [60].

4.8.3 Entrenamiento de expertos: modelos base y justificación de hiperparámetros

Fijada la ventana ganadora W^* y el subconjunto de canales C^* para un *split* dado, se construyen las matrices de características del nivel base

$$X_{tr}^{(s)} \in \mathbb{R}^{N_{tr}\times P} \text{ y } X_{te}^{(s)} \in \mathbb{R}^{N_{te}\times P} \qquad [E.4.31],$$

donde $P = 7 \times |C^{(s)}|$ (7 descriptores por canal; ver Sección 4.7). Posteriormente, para el *stacking* se construye el espacio de decisiones $\mathrm{Z_{tr}} \in \mathbb{R}^{N_{tr}\times 2E}$ (o $Z_{tr}^{(log)} \in \mathbb{R}^{N_{tr}\times E}$ si se usan *logit-pairs*) a partir de probabilidades *out-of-fold* de los E expertos.

A continuación, se entrenan cinco clasificadores base (“expertos”) integrados en pipelines con normalización y selección de características: *LDA, SVM-RBF, Random Forest, XGBoost binario y MLP*. El objetivo del comité es combinar modelos de familias lineales y no lineales con sesgos inductivos distintos (separación lineal, márgenes no lineales, ensambles de árboles y redes neuronales), aumentando la robustez ante la señal fNIRS, que es lenta, ruidosa y altamente variable entre sujetos [39], [50]–[53].

La selección de los cinco expertos se ajusta al escenario del trabajo: clasificación binaria por sujeto con pocos ensayos, señales fNIRS variables y necesidad de salidas probabilísticas para *stacking* y rechazo. Se incluyen modelos lineales y no lineales (*LDA-bagging, SVM-RBF, Random Forest, XGBoost* y *MLP*) para aportar diversidad y robustez.

Los clasificadores base se seleccionan buscando diversidad en los sesgos inductivos y capacidad de modelado, con el fin de capturar distintas estructuras presentes en las señales fNIRS. En particular:

a) LDA se incluye como modelo lineal de referencia ampliamente utilizado en BCIs, debido a su bajo costo computacional, estabilidad con pocas muestras y buen desempeño en problemas de imaginación motora [39].

b) SVM con kernel RBF permite modelar fronteras de decisión no lineales, lo que resulta adecuado cuando las características no son linealmente separables, como suele ocurrir en señales fisiológicas ruidosas [40].

c) Random Forest introduce un enfoque basado en ensambles de árboles, capaz de capturar interacciones no lineales y robusto frente a ruido y *outliers* (desviaciones), además de proporcionar estimaciones de importancia de variables [51].

d) XGBoost se incorpora como método de *boosting* que mejora iterativamente el desempeño mediante la corrección de errores, ofreciendo alta capacidad predictiva en datos tabulares y complejos [52].

e) MLP (red neuronal) permite aproximar funciones altamente no lineales y modelar relaciones complejas entre características, aportando flexibilidad adicional al comité. En este trabajo se utiliza una arquitectura poco profunda y de tamaño reducido, adecuada para conjuntos de datos moderados y espacios de características compactos como los derivados de descriptores temporales en fNIRS. De esta manera, la MLP complementa a los modelos lineales y a los basados en árboles dentro del esquema de ensemble, capturando posibles interacciones no lineales entre variables.

La combinación de estos modelos responde a la necesidad de abordar la variabilidad intra–sujeto y la baja relación señal–ruido típica de fNIRS, donde diferentes clasificadores pueden capturar aspectos complementarios de la señal. Este principio de diversidad es fundamental en esquemas de ensemble y stacking, ya que permite mejorar la generalización al integrar predicciones provenientes de modelos con comportamientos distintos [49].

Adicionalmente, los clasificadores seleccionados son coherentes con la dimensionalidad relativamente baja del espacio de características utilizado en este trabajo, basado en descriptores temporales por canal y ventana. Modelos como LDA, SVM, Random Forest y XGBoost han mostrado buen desempeño en escenarios con número moderado de variables y conjuntos de entrenamiento relativamente pequeños, como suele ocurrir en aplicaciones de BCI basadas en fNIRS.

La diversidad de sesgos inductivos (lineales y no lineales) favorece decisiones más estables cuando se trabaja con pocas muestras y alta dimensionalidad relativa, condición típica en MI-fNIRS. En este trabajo de grado, las salidas de estos expertos se combinan posteriormente mediante *stacking*, por lo que en esta sección se listan únicamente los modelos considerados y su motivación general. A continuación se presenta la configuración específica de cada experto:

a) ***Experto 1: Balanced Bagging con LDA regularizado (shrinkage).*** El primer experto usa LDA por su robustez en BCI cuando el número de muestras (N) es pequeño y el número de variables (P) puede ser moderado/alto [39]. Bajo el supuesto gaussiano con covarianza compartida, la discriminante es

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log\pi_k \qquad [E.4.32],$$

Con predicción $\hat{y} = \arg max_k\, \delta_k(x)$. Para evitar inestabilidad al estimar $\boldsymbol{\Sigma}$ cuando P es comparable con N , se aplica *(solver lsqr,* $\lambda = 0.3$ *),* regularizando

$$\Sigma_\lambda = (1-\lambda)\Sigma + \lambda \mathrm{I}, \qquad \lambda \in [0,1] \qquad [E.4.33],$$

reduciendo varianza y mejorando generalización [39], [59]. Se usa *StandardScaler* (sensibilidad de LDA a escalas), y *RFECV* para controlar sobreajuste eliminando variables redundantes/ruidosas (relevante en fNIRS por confusores y variabilidad por canal) [29], [30], [54], [59]. Finalmente, se ensambla con *BalancedBagging* $(B = 30)$ para reducir varianza, aproximando

$$\hat{p}(y = k \mid x) = \frac{1}{B}\sum_{b=1}^{B}\hat{p}_b(y = k \mid x) \qquad [E.4.34],$$

con *bootstraps* balanceados (estabiliza sensibilidad por clase y coherente con *balanced accuracy*) [60], [59].

b) Experto 2: SVM con kernel RBF + RFE. El segundo experto utiliza una máquina de vectores soporte con *kernel* RBF para capturar fronteras no lineales cuando la separación LMI/RMI no es aproximadamente lineal [39]. La decisión kernelizada es

$$f(\mathrm{x}) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \qquad [E.4.35],$$

donde el *kernel* RBF es

$$K(x, x') = exp(-\gamma \parallel x - x' \parallel^2) \qquad [E.4.36],$$

En el algoritmo, se fija $gamma='scale'$(punto de partida estable ante cambios de escala por sujeto) y $C = 1.0$ (compromiso margen–error; evita extremos de sobre/sub-ajuste en conjuntos pequeños como el presente). Se aplica *StandardScaler* porque el RBF depende directamente de distancias euclidianas. Para reducir dimensionalidad y robustecer el RBF ante ruido, se usa *RFE (LinearSVC)* con $n_features_to_select = 10$ eliminando iterativamente las variables menos relevantes según pesos del modelo lineal [54]. Se activa $probability = True$ (calibración tipo *Platt* sobre el *score*) para obtener probabilidades útiles en *stacking* y rechazo [56], [59], y $class_weight = 'balanced'$ para estabilizar el entrenamiento por *split* [59].

c) ***Experto 3: Random Forest balanceado con selección Boruta.*** El tercer experto se basa en *Random Forest,* un ensamble de árboles entrenados con *bagging* y selección aleatoria de variables, apropiado cuando se sospechan no linealidades e interacciones entre características y además se requiere robustez frente a ruido [51]. Matemáticamente, un bosque predice por votación/promedio de árboles

$$\hat{p}(y = k \mid x) = \frac{1}{T} \sum_{t=1}^{T} \hat{p}_t(y = k \mid x) \qquad [E.4.37],$$

donde cada árbol t se entrena sobre una muestra *bootstrap* y en cada nodo considera un subconjunto aleatorio de variables. Se estabiliza la estimación con $n_estimators = 200$ y se limita complejidad $(max_depth = 6, min_samples_split = 5,\ min_samples_leaf = 3)$ para evitar sobreajuste en N pequeño; $max_features = 'sqrt'$ aumenta diversidad y generalización [51]. La selección con Boruta retiene variables consistentemente superiores a “sombras”, siendo conservadora y adecuada en fNIRS para evitar dependencia de variables ligadas a confusores sistémicos [29], [30], [55]. Se usa $class_weight = 'balanced'$ para estabilidad por clase [59]. Este experto no necesita *wrapper* de etiquetas porque el *RandomForestClassifier* de *scikit-*learn opera de forma nativa con etiquetas {1,2}, entregando $predict_proba$ e importancias útiles en *stacking* [59].

d) ***Experto 4: XGBoost binario con selección por importancia (SelectFromModel).*** El cuarto experto utiliza *XGBoost*, un método de *gradient boosting* que construye un modelo aditivo de árboles donde cada nuevo árbol corrige el error residual del conjunto previo. En su forma general, el predictor es

$$\hat{y}(x) = \sum_{m=1}^{M} f_m(x), \quad f_m \in F \qquad [E.4.38],$$

y su probabilidad logística es

$$p(y = 1 \mid x) = \sigma\big(\mathrm{s(x)}\big) = \frac{1}{1 + e^{-s(x)}} \qquad [E.4.39],$$

incluyendo regularización (p. ej., ***reg_alpha L1***, ***reg_lambda L2***). Para N pequeño se restringe complejidad $(max_depth = 3)$, se controla el paso $(learning_rate = 0.1)$ y se introduce aleatoriedad $(subsample = 0.8, colsample_bytree = 0.8)$; $min_child_weight = 3$ y $gamma = 0.1$ restringen splits de baja ganancia y reducen sobreajuste [52], [53]. La selección se hace con SelectFromModel $(threshold = 'median', min_features = 6)$ para conservar variables con contribución relativa alta. Se usa *wrapper (XGBSafeBinary)* para compatibilizar etiquetas {1,2} con el supuesto binario {0,1} y mantener *predict_proba* estable en pipelines/meta-modelos [52], [59].

e) Experto 5: MLP (perceptrón multicapa) con selección univariada (SelectKBest). El quinto experto usa un perceptrón multicapa (MLP) de una capa oculta, capaz de aproximar relaciones no lineales suaves entre variables. El modelo implementa una función

$$h(x) = g(W_1 x + b_1), \quad \hat{p}(y = k \mid x) = softmax(W_2 h(x) + b_2) \qquad [E.4.40],$$

entrenado con entropía cruzada y regularización L2:

$$\frac{min}{\theta} \sum_{n} \mathcal{L}\big(y_n, \hat{p}(x_n; \theta)\big) + \alpha \parallel \theta \parallel_2^2 \qquad [E.4.41].$$

Dado el tamaño muestral, se usa una capa pequeña (20 neuronas), optimizador ***adam***, $\alpha = 0.001$ para reducir sobreajuste y early_stopping $(validation_fraction = 0.2)$ como regularización efectiva [59]. Para evitar que la red aprenda ruido en alta dimensión, se reduce espacio con SelectKBest (*ANOVA F-score,* $k = 10$), manteniendo variables con mayor separación univariada. En conjunto, el *MLP* aporta una hipótesis no lineal distinta a *SVM* y *árboles*, útil cuando existen combinaciones suaves de descriptores hemodinámicos que discriminan condiciones.

La configuración de cada experto, junto con los pasos de normalización/selección y el meta-clasificador, se resume en la Tabla 4-4.

Tabla 4-4. Expertos del comité, pipelines y configuración de hiperparámetros.

Experto (ID)	Pipeline (orden exacto)	Hiperparámetros clave (tal cual en código)	Rol / justificación (por qué así)	Selección de características - Wrappers / notas
E1: BAG+LDA (bag)	StandardScaler → RFECV(LDA) → BalancedBagging(LDA)	LDA: solver='lsqr', shrinkage=0.3 (en bag y en nested) | RFECV: step=0.1, cv=5 (en expertos), min_features_to_select=6, scoring='balanced_accuracy' | Bagging: n_estimators=30, max_samples=0.6, max_features=0.8, bootstrap=True, bootstrap_features=True, sampling_strategy='auto'	Baseline fuerte en BCI: LDA regularizado es muy usado por su estabilidad/interpretabilidad; bagging reduce varianza y mejora robustez en muestras pequeñas; BA como métrica evita sesgos en splits pequeños.	**No requiere wrapper. Bagging (ensamble por remuestreo) es estándar. [39], [50], [59], [60]**
E2: SVM-RBF (svm)	StandardScaler → RFE(LinearSVC) → SVC(RBF, proba)	RFE: LinearSVC(C=0.1, penalty='l2', dual=False), n_features_to_select=10, step=0.2 | SVC: kernel='rbf', C=1.0, gamma='scale', probability=True, class_weight='balanced'	SVM-RBF captura fronteras no lineales cuando la separabilidad no es estrictamente lineal; probability=True es necesario para stacking; RFE reduce dimensionalidad/overfitting.	**No wrapper. Se usa class_weight='balanced' para robustez por clase en particiones. [39], [54], [59]**
E3: RF+Boruta (rf)	SafeBoruta(RF pequeño) → RandomForestClassifier	Boruta (RF auxiliar): RandomForestClassifier(n_estimators=50), n_estimators='auto', alpha=0.05, max_iter=100 | RF final: n_estimators=200, max_depth=6, min_samples_split=5, min_samples_leaf=3, max_features='sqrt', class_weight='balanced', n_jobs=4	RF es robusto a ruido y variables irrelevantes; Boruta es selección "conservadora" basada en importancia con variables sombra; profundidad y mínimos de hoja/split controlan sobreajuste en N pequeño.	**SafeBoruta evita el caso degenerado "0 features seleccionadas" (fallback top-k por importancia). RF no necesita wrapper de etiquetas: scikit maneja clases {1,2} y predict_proba nativamente. [51], [55], [59]**
E4: XGBoost (xgb)	SafeSelectFromModel(XGB) → XGBSafeBinary	SelectFromModel: threshold='median', min_features=6, max_features=10 (por clase wrapper), importance_getter='feature_importances_' | XGB (selector): n_estimators=60, max_depth=3, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8 | XGB final: n_estimators=100, max_depth=3, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8, min_child_weight=3, gamma=0.1, reg_alpha=0.1, reg_lambda=1	Boosting de árboles suele rendir muy bien en datos tabulares; hiperparámetros (profundidad baja + regularización + subsampling) controlan complejidad y evitan sobreajuste; selección por importancia reduce dimensionalidad y estabiliza.	**XGBSafeBinary garantiza compatibilidad con etiquetas {1,2} (internamente usa {0,1}) y expone classes_/predict_proba/feature_importances_ de forma estable en pipeline. SafeSelectFromModel fuerza mínimo de features si el umbral deja muy pocas. [52], [53], [59]**
E5: MLP (mlp)	StandardScaler → SelectKBest(f_classif) → MLPClassifier	SelectKBest: k=10 | MLP: hidden_layer_sizes=(20,), solver='adam', learning_rate_init=1e-3, alpha=0.001, early_stopping=True, validation_fraction=0.2, n_iter_no_change=15, max_iter=2000	MLP aporta no linealidad flexible; *early stopping* y L2 (alpha) reducen sobreajuste; SelectKBest baja dimensión para que la red sea entrenable con N pequeño.	**No wrapper. Se estandariza siempre por sensibilidad del gradiente. [59]**

Meta: Stacking (logística calibrada)	OOF probas → (opcional) logit-pairs → LogisticRegression → CalibratedClassifierCV(sigmoid)	OOF: cross_val_predict(..., method='predict_proba') con StratifiedKFold(5) | Logit-pairs: (\log(p_2/p_1)) por experto | LR: max_iter=5000, class_weight='balanced', solver='lbfgs' | Calibración: method='sigmoid', cv=5	Stacking aprende la combinación óptima de expertos usando predicciones "honestas" OOF (evita leakage); logit-pairs mejora separabilidad y reduce dimensión; calibración es crítica porque el rechazo usa confianza probabilística.	**La calibración tipo sigmoide está alineada con enfoques clásicos de calibración (Platt) y generalización multiclass. [49], [56], [57], [59]**

4.8.4 Probabilidades out-of-fold (OOF) y construcción del espacio de stacking

Para entrenar el meta-modelo sin sobreajuste, el *stacking* se construye con probabilidades "*out-of-fold*". Cada experto *e* se calculan probabilidades de clase OOF mediante validación cruzada estratificada (*StratifiedKFold*). Es decir, se genera una matriz de probabilidades

$$\mathbf{Z}^{(e)} \in \mathbb{R}^{N\times 2}, Z_{n,k}^{(e)} \approx \Pr(\, y_n = k \mid \boldsymbol{x}_n, \textit{modelo } e\,) \qquad [E.\,4.42],$$

en la que cada probabilidad se ha obtenido evaluando el clasificador entrenado sin usar la muestra n (vía *cross_val_predict* con *predict_proba*, evitando así sobre–ajuste en el meta–nivel. La concatenación de estas matrices para todos los expertos produce

$$\mathbf{Z} = \left[\, \mathbf{Z}^{(1)} \mid \mathbf{Z}^{(2)} \mid \ \ldots \ \mid \mathbf{Z}^{(E)} \,\right] \in \mathbb{R}^{N\times 2E} \qquad [E.\,4.43],$$

que constituye el "espacio de decisiones" del ensemble. Este enfoque implementa la idea de *stacked generalization*: entrenar un meta-modelo con predicciones "honestas" de los modelos base, evitando el *leakage* del entrenamiento directo del meta sobre probabilidades in-*sample* [49], [59].

4.8.4.1 Transformación logit-pairs

Para mejorar la separabilidad y simetría de clases se aplica una transformación a *logit–pairs* (cada par p_1, p_2), de modo que para cada experto se define una característica

$$\ell_n^{(e)} = \log\left(\frac{Z_{n,2}^{(e)}}{Z_{n,1}^{(e)}}\right) \qquad [E.\,4.44],$$

de forma que valores positivos indican evidencia a favor de la clase 2 y negativos a favor de la clase 1. Esta representación reduce dimensionalidad de $2E$ a E y suele facilitar un meta-modelo lineal. La matriz resultante $\mathbf{Z}^{\log} \in \mathbb{R}^{N\times E}$ se alimenta a un meta–clasificador logístico [57], [59]. Posteriormente, el meta-clasificador se implementa como regresión logística balanceada, calibrada para obtener probabilidades bien comportadas, clave para el rechazo [56], [59].

Se fija explícitamente el orden de clases del pipeline como $CLASS_ORDER = [1,2]$ (mano derecha, mano izquierda), de modo que toda probabilidad $p_{n,k}$ sea interpretable sin ambigüedad al usar $\arg max$. Adicionalmente, se guarda el orden exacto de los expertos retenidos *experts_kept*) para reconstruir consistentemente el vector de entrada del meta-modelo en inferencia, evitando desalineaciones de columnas en el espacio de *stacking*.

4.8.4.2 Meta-clasificador, pesos interpretables y poda de expertos

Los pesos interpretables usados para la poda se estiman con una regresión logística auxiliar entrenada sobre el espacio de *stacking* Z^{log} (*OOF*), usando la magnitud de sus coeficientes como medida de contribución relativa por experto. El meta-clasificador final se entrena aparte y se calibra con *CalibratedClassifierCV* para obtener probabilidades confiables. Esta estimación de pesos se usa únicamente para decidir qué expertos mantener.

El meta-modelo es una regresión logística con regularización $L2$ y *class_weight='balanced'.* Se calibra además con *sigmoid calibration (CalibratedClassifierCV*, método sigmoidal) para mejorar calibración probabilística, lo cual es crucial porque el rechazo depende directamente de la confianza (máxima probabilidad) [56], [59]. (En la literatura, la calibración sigmoidal se asocia al enfoque de *Platt* para convertir scores en probabilidades.) [56].

La interpretación del modelo logístico permite estimar un peso relativo interpretable para cada experto. Sea $\beta \in \mathbb{R}^E$ el vector de coeficientes aprendido del meta; el código resume la contribución del experto *e* mediante $w_e = | \beta_e |$ (o, cuando se usan dos coeficientes o columnas por experto, la suma de sus magnitudes/valores absolutos), que se normaliza para obtener pesos relativos

$$\widetilde{w}_e = \frac{w_e}{\sum_j w_j} \qquad [E.4.45].$$

Con estas cantidades se implementa una poda, un experto se considera débil si cumple simultáneamente lo siguiente: expertos con peso $\widetilde{w}_e < 0.08$ y balanced accuracy out-of-fold por debajo de 0.55 se consideran débiles y pueden ser podados del ensemble. Esta poda evita que modelos inconsistentes introduzcan ruido en el espacio meta, en línea con la teoría de ensembles que indica que la ganancia se produce cuando se combinan clasificadores individualmente competentes y diversificados [49], [50].

4.8.4.3 Rechazo adaptativo basado en confianza y optimización por sujeto

Una vez entrenado el *stacking* y calibrado el meta–clasificador, cada muestra n produce un vector de probabilidades:

$$\mathrm{p}_n = (p_{n,1}, p_{n,2}) \qquad [E.4.46],$$

y se calcula la confianza como la probabilidad máxima

$$c_n = \max_k p_{n,k} \qquad [E.4.47].$$

Siguiendo el marco de clasificación con opción de rechazo, si la confianza del modelo (probabilidad máxima del meta-clasificador) es menor que el umbral *θ*, la salida se etiqueta como rechazo/idle (clase 0) [58]. El umbral θ no es global, porque se optimiza por sujeto usando probabilidades *out-of-fold* obtenidas exclusivamente del conjunto de entrenamiento y solo después se aplica al *hold-out* del *split*, evitando optimismo y fuga de información.

En particular, se generan probabilidades *OOF* del meta en $D_{tr}^{(s)}$; con esas probabilidades OOF se optimiza *θ* bajo restricciones de tasa de rechazo; y solo después *θ* se aplica a $D_{te}^{(s)}$, que no participa en la optimización.

Para cada candidato θ (rejilla alrededor de un cuantil asociado a una tasa objetivo $\rho_{obj} \approx 10\%$), se calcula el desempeño en el subconjunto $A_\theta = \{c_n \geq \theta\}$ mediante

$$BA_{acept}(\theta) = \frac{1}{2}\left(\frac{TP_1(\theta)}{TP_1(\theta) + FN_1(\theta)} + \frac{TP_2(\theta)}{TP_2(\theta) + FN_2(\theta)}\right) \quad [E.4.48],$$

y la tasa de rechazo

$$\rho(\theta) = \frac{1}{N}\sum_n \mathbf{1}[\, c_n < \theta] \qquad [E.4.49],$$

permanezca dentro de un rango permitido (5–18 %).

Imponiendo $\rho(\theta) \in [0.05, 0.18]$. Además, se incorpora explícitamente calibración mediante el *ECE* sobre aceptados, usando B *bins* de confianza (error por *bin*, ponderado por cuántas muestras caen en ese *bin*):

$$ECE_{acept}(\theta) = \sum_{b=1}^{B} \frac{|S_b|}{|A_\theta|} |acc(S_b) - conf(S_b)|, \qquad [E.4.50],$$

con

$$acc(S_b) = \frac{1}{|S_b|}\sum_{n \in S_b} 1[\hat{y}_n = y_n], \qquad conf(S_b) = \frac{1}{|S_b|}\sum_{n \in S_b} c_n \qquad [E.4.51].$$

De este modo, el umbral óptimo θ^* maximiza entonces una utilidad selectiva sobre probabilidades *out-of-fold* (OOF) del *meta-model* que equilibra desempeño en aceptados, cobertura (rechazo cercano a ρ_{obj}) y calibración (*ECE* bajo):

$$U(\theta) = BA_{acept}(\theta) - \lambda_{\text{rej}}\left|\rho(\theta) - \rho_{obj}\right| - \lambda_{\text{ece}} ECE_{acept}(\theta) \qquad [E.4.52],$$

sujeta a $\rho_{obj} \in \{0.08, 0.10, 0.12, 0.15\}$ y pesos típicos$\lambda_{\text{rej}} = 1.0$, $\lambda_{\text{ece}} = 0.25$ [58]. Este diseño implementa el *trade-off (compromiso)* clásico error–rechazo y lo adapta a la variabilidad inter-sujeto propia de BCI–fNIRS [29], [30], [58]. Para mantener consistencia entre lo estimado en *OOF* y lo observado en prueba, el umbral se ajusta suavemente por cuantiles de c_n en test para mantener ρ dentro del rango permitido.

Finalmente, por *split* se reportan métricas globales (todos los ensayos) y sobre aceptados (incluyendo ρ, *BA/Accuracy* y desempeño por clase). La selección del split exportado por sujeto se basa en la utilidad selectiva (no en la exactitud puntual sin rechazo), priorizando decisiones confiables y probabilísticamente coherentes para un uso operativo real. Además, se reporta la media y desviación estándar entre *splits* para cuantificar la variabilidad inducida por la partición, y se guarda el paquete reproducible (modelos, canales/características, pesos del meta y θ*), junto con matrices de confusión y gráficas asociadas. A continuación, se presenta la Tabla 4-5 que resume las métricas obtenidas.

Tabla 4-5. Métricas guardadas del modelo.

Métrica	Dónde se calcula	Sobre qué datos	Con rechazo	Para qué se usa
cv_mean_split, cv_std_split	Nested-CV (ventana/canales) dentro de cada split	Solo train (75%)	No	Selección de (W, canales, τ) y estabilidad interna
expert_ba[name]	OOF por experto	Train (OOF)	No	Diagnóstico y criterio de poda
meta_weights	Regresión logística auxiliar sobre (Z_meta)	Train	No	Interpretabilidad y poda
best_thr, best_rej, best_ba, best_ece	Búsqueda de umbral sobre OOF del meta	Train (OOF)	Sí	Elegir (\theta) sin mirar test, controlando BA/rechazo/calibración
acc_holdout	accuracy_score(y_te, y_pred)	Holdout (25%)	No	Reporte del clasificador sin rechazo (2 clases)
acc_accepted	accuracy_score en aceptados	Holdout	Sí	Exactitud condicionada a aceptar
ba_accepted	balanced_accuracy en aceptados	Holdout	Sí	Métrica principal de decisión selectiva
ece_accepted	ECE en aceptados	Holdout	Sí	Calidad de calibración bajo abstención
rej_rate	% (y_pred_rej==0)	Holdout	Sí	Cobertura/confiabilidad
aurc	Área bajo risk–coverage	Holdout	Implícito	Calidad selectiva global (trade-off riesgo–cobertura)
util_holdout	util = ba_acc - lambda_rej * abs(rej - target_rej) - lambda_ece * ece_acc	Holdout	Sí	Criterio para elegir el mejor
Confusión 3×3	confusion_matrix(y_te, y_pred_rej, [1,2,0])	Holdout	Sí	Análisis de errores incluyendo rechazo
Confusión 2×2	confusion_matrix(y_te, y_pred, [1,2])	Holdout	No	Análisis puro sin rechazo
Risk–Coverage + AURC	curva por confianza	Holdout	Sí	Diagnóstico selectivo y comparación entre sujetos
Reliability diagram + ECE	bins de confianza	Aceptados (holdout)	Sí	Diagnóstico de calibración

Aunque para cada sujeto se exporta, el *split* que maximiza una utilidad selectiva en el *hold-out* (integrando desempeño en aceptados, cobertura vía rechazo y calibración), también se reportan la *accuracy* sin rechazo y la media ± desviación estándar de las métricas en *hold-out* sobre 15 particiones, con el fin de cuantificar la variabilidad inducida por el muestreo y evitar conclusiones dependientes de una única división [39], [59].

En conjunto, la combinación de características temporales, selección de canales con r^2 firmado, y stacking con calibración y rechazo adaptativo constituye un pipeline coherente para BCIs hemodinámicas: mantiene interpretabilidad fisiológica, controla el sobreajuste mediante validación cruzada anidada y añade un mecanismo explícito de manejo de incertidumbre para emitir decisiones solo cuando son confiables [29], [30], [58].

4.9 Generación de comandos para el avatar y exportación de resultados

Una vez entrenado el modelo final por sujeto (expertos + meta–clasificador) y fijados los parámetros de personalización $(t_mask, channels, \theta)$, esta etapa no reentrena: únicamente re–ejecuta inferencia sobre el hold-out del split exportado para generar comandos discretos reproducibles para Unity. Con ello se garantiza que la extracción de características y la lógica de decisión coinciden con las usadas en la evaluación externa, preservando el aislamiento entre ajuste y prueba bajo el esquema nested-CV + partición externa [59], [60].

El procedimiento usa: (i) el paquete $*$*_model_ensemble.pkl* (ventana *t_mask,* canales *channels,* umbral *thr_reject,* expertos con sus pipelines y el meta–clasificador *stacking_meta*); y (ii) $*$*_holdout25.mat* (épocas y etiquetas reales). A partir de las épocas del *hold-out* se recalculan características aplicando exactamente *t_mask* y *channels* Cada experto $e \in \{1, \dots, E\}$ produce probabilidades por clase $\hat{p}^{(e)}(\, y = k \mid x_n)$ con $k \in \{1,2\}$, donde por convención 1 = "mano derecha" y 2 = "mano izquierda". y para el meta–nivel se usa el *logit-pair*:

$$\ell_{\mathrm{n}}^{\mathrm{e}} = log\left(\frac{\hat{\mathrm{p}}^{(\mathrm{e})}\,(\mathrm{y}=2 \mid \mathrm{x_n})}{\hat{\mathrm{p}}^{(\mathrm{e})}\,(\mathrm{y}=1 \mid \mathrm{x_n})}\right), \quad con\ \hat{\mathrm{p}}^{(\mathrm{e})} \leftarrow \mathrm{clip}\left(\hat{\mathrm{p}}^{(\mathrm{e})}, \varepsilon, 1-\varepsilon\right) \qquad [E\ 4.53],$$

donde ε evita inestabilidades. Con $Z_n^{log} = \left[\ell_n^{(1)}, \dots, \ell_n^{(E)}\right]$ El meta–clasificador (regresión logística calibrada) produce entonces una distribución final por clases $p_n = (p_{n,1}, p_{n,2})$ y una predicción base sin rechazo

$$\hat{y}_n = \arg \max_{k \in \{1,2\}} p_{n,k} \qquad [E\ 4.54].$$

se define la confianza de decisión como

$$c_n = \max_{k \in \{1,2\}} p_{n,k} \qquad [E\ 4.55],$$

Y se aplica el umbral de rechazo específico del sujeto $\theta = thr_reject$ aprendido/ajustado con probabilidades *out-of-fold* durante el entrenamiento. La salida final del clasificador con rechazo se expresa como una etiqueta ternaria

$$\tilde{y}_n = \begin{cases} \hat{y}_n, & si\ c_n \geq \theta, \\ 0, & si\ c_n < \theta \end{cases} \qquad [E\ 4.56],$$

Donde 0 representa idle/rechazo, coherente con la clasificación con opción de rechazo [58]. Para trazabilidad, se exporta un CSV intermedio por ensayo (*trial, true_label, pred_label, etiqueta nominal*), que registra explícitamente cuándo el sistema decide y cuándo se abstiene. Adicionalmente, para la demostración offline en Unity se genera un CSV limpio con una política conservadora: el avatar solo ejecuta gesto cuando no hay rechazo y, además, la predicción coincide con la etiqueta real (evitando acciones contradictorias durante pruebas controladas). Esto se implementa como:

$$cmd_for_unity_n = \begin{cases} \hat{y}_n, & \tilde{y}_n \neq 0 \ \wedge\ \tilde{y}_n = y_n, \\ 0, & en\ otro\ caso \end{cases} \qquad [E\ 4.57],$$

$$is_error_n = \begin{cases} 1, & \tilde{y}_n \neq 0 \ \wedge\ \tilde{y}_n \neq y_n, \\ 0, & en\ otro\ caso \end{cases} \qquad [E\ 4.58],$$

separando rechazo (abstención deliberada) de error (decisión activa incorrecta), lo cual es clave para analizar confiabilidad bajo rechazo [58]. Esta limpieza depende de $\tilde{y}_n$ y, por tanto, es estrictamente offline; en operación online el comando correspondería a *is_error* no sería observable en tiempo real.

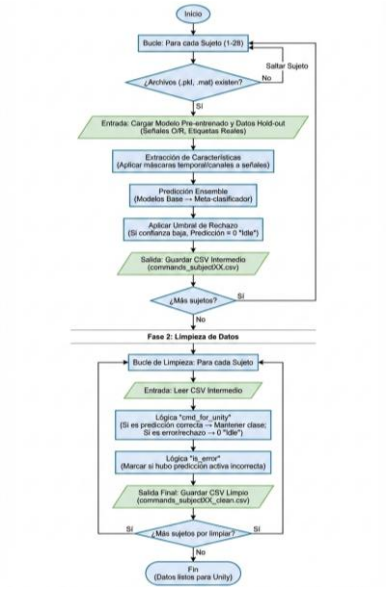


Figura 4-12. Diagrama de flujo de generación y limpieza de comandos para Unity.

La Figura 4-12 sintetiza el flujo de generación y limpieza de comandos. El proceso recorre los sujetos (1–28) y verifica la existencia de los archivos de entrada (modelo entrenado y datos de hold-out). Para cada sujeto válido se recalculan las características usando la ventana y los canales seleccionados, se ejecuta la inferencia de los expertos y el meta-clasificador, y se obtienen probabilidades y etiquetas predichas. Luego se aplica el umbral θ para decidir entre {izquierda, derecha} o rechazo (idle) y se exporta un CSV intermedio. En una segunda fase de limpieza se generan las columnas finales para Unity (comando y bandera de error) y se exporta el CSV limpio listo para consumo por el avatar.

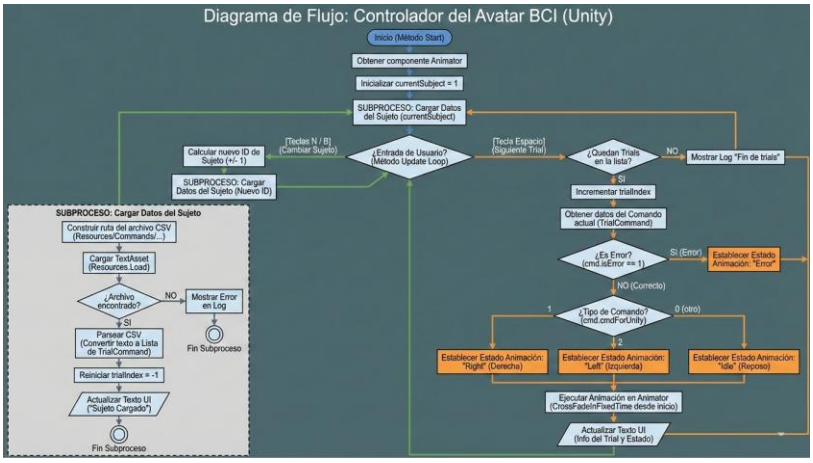


Figura 4-13. Diagrama de flujo para el control del avatar dentro de Unity.

Una vez generados los CSV limpios, el sistema se integra en Unity mediante el controlador *AvatarBCIController*, cuya lógica general se resume en la Figura 4-13. Este script (*MonoBehaviour)* utiliza el ciclo de vida estándar de Unity: en *Start()* valida la referencia al *Animator,* fija el sujeto inicial (*currentSubject=1*) y carga el archivo *commands_subjectXX_clean.csv* desde *Resources/Commands/* como *TextAsset,* parseándolo en una lista de ensayos (*TrialCommand*). En *Update(),* la navegación se realiza por eventos de teclado (barra espaciadora para avanzar ensayo, N/B para cambiar de sujeto), controlando límites para evitar desbordamientos y notificando cuando finalizan los ensayos disponibles.

La reproducción de animaciones implementa una política explícita de prioridad: si *isError==1* se fuerza siempre el estado *Error*; en caso contrario, el estado depende de *cmdForUnity* (*1→Right, 2→Left, 0→Idle*). Esta prioridad permite distinguir visualmente entre error por decisión activa incorrecta y rechazo/neutralidad, incluso cuando el CSV limpio impone *Idle* como medida conservadora. Las transiciones se ejecutan *con Animator.CrossFadeInFixedTime(stateName, 0.1f, 0, 0f)*, asegurando un *cross-fade* breve e inicio consistente de cada gesto al cambiar de ensayo. Finalmente, el controlador actualiza en pantalla un panel de UI *(TextMeshProUGUI)* con sujeto, índice de ensayo, etiqueta real y estado reproducido, aportando trazabilidad visual durante la simulación.



Figura 4-14. Montaje del entorno del avatar dentro de Unity.

El montaje del avatar en Unity sigue la Figura 4-14: el personaje incluye un *Animator* con estados *Idle, Right, Left* y *Error*, y el script *AvatarBCIController* gobierna las transiciones según los comandos exportados. Los archivos `commands_subjectXX_clean.csv` se guardan en *Assets/Resources/Commands/* y se cargan como *TextAsset* para ser parseados en tiempo de ejecución. Así, el avatar reproduce de forma offline la secuencia generada por el pipeline, cerrando el ciclo entre clasificación y simulación.

Con ello, la etapa de simulación virtual queda especificada de extremo a extremo, se define entradas (artefactos del modelo y *hold-out*), transformaciones (reconstrucción de características, inferencia de expertos, *stacking*, rechazo y generación de trazas), salidas (CSV intermedio y CSV limpio) y su consumo en Unity (carga de recursos, control por teclado y ejecución de estados/animaciones), habilitando tanto evaluación cuantitativa (errores vs. rechazos) como evaluación cualitativa del comportamiento del sistema a nivel de usuario final [15]–[18], [58]–[60].

4.10 Resumen del capítulo

En este capítulo se presentó la metodología de la solución propuesta desde la perspectiva de software, describiendo el pipeline en Python para procesamiento y clasificación de señales fNIRS en un escenario BCI. El flujo incluye conversión MBLL, corrección explícita de artefactos, filtrado en banda hemodinámica y segmentación de épocas asociadas a imaginación motora, a partir de las cuales se extraen características temporales y se aplica una selección jerárquica (filtro por $|r^2|$ y métodos embebidos) para controlar la dimensionalidad y el sobreajuste por sujeto. La decisión se implementó mediante *stacking* con calibración y rechazo, permitiendo evaluar desempeño en términos de exactitud, cobertura y confiabilidad. Finalmente, se describió la exportación de resultados a comandos discretos y su integración en Unity para validar la trazabilidad del sistema de manera offline.

5. Pruebas y Resultados

5.1 Introducción

En este capítulo se presentan las pruebas realizadas para evaluar el desempeño de la BCI basada en fNIRS propuesta y se discuten los resultados obtenidos. La evaluación se aborda desde dos perspectivas complementarias.

En primer lugar, se reporta el desempeño cuantitativo del sistema de clasificación por sujeto mediante métricas obtenidas sobre particiones de evaluación externas (hold–out), lo que permite estimar la capacidad del pipeline para discriminar entre imaginación motora de mano izquierda y mano derecha bajo variabilidad inter–sujeto.

En segundo lugar, se realiza una validación funcional mediante la generación de comandos discretos y su ejecución en un entorno de simulación desarrollado en Unity, con el fin de analizar la coherencia operativa de las decisiones del sistema y su comportamiento cuando actúa como controlador de un actuador visual.

El propósito de este capítulo es establecer hasta qué punto el pipeline implementado logra una discriminación útil en términos de desempeño y, además, qué tan confiables son sus salidas cuando se traducen en acciones observables en una demostración offline.de imaginación motora de mano izquierda y derecha y qué tan fiable resulta cuando se integra con un actuador visual.

5.2 Descripción general del conjunto de datos y configuración experimental

El conjunto de datos utilizado está compuesto por 28 sujetos, cada uno con tres sesiones de imaginación motora (MI). Tras la depuración de ensayos, se emplean 60 ensayos válidos por sujeto, balanceados entre clases (30 LMI y 30 RMI). Las señales registradas como intensidad óptica se convierten a concentraciones relativas $\Delta HbO/\Delta HbR$ mediante MBLL y se someten al preprocesamiento definido previamente (corrección explícita de artefactos, filtrado en banda hemodinámica y segmentación por eventos). La segmentación se realiza alrededor del marcador de inicio de tarea, generando épocas fijas de 35 s a 10 Hz (350 muestras), con el fin de capturar tanto la fase de activación como la recuperación hacia la línea base.

La evaluación se plantea por sujeto y con separación estricta entre selección/entrenamiento y evaluación externa. Para cada sujeto se generan múltiples particiones estratificadas entrenamiento–prueba mediante muestreo aleatorio repetido, reservando 25% como *hold-out* y usando el 75% restante para ajuste. Dentro de cada partición, la selección de ventana temporal, subconjunto de canales motores y cualquier reducción adicional de dimensionalidad se realiza

exclusivamente con el conjunto de entrenamiento mediante validación cruzada interna estratificada, mientras que el hold-out se conserva como evaluación externa, evitando fuga de información y reduciendo el riesgo de sobreajuste.

Adicionalmente, el sistema incorpora decisión selectiva: el meta-clasificador en esquema de *stacking* produce probabilidades y se aplica un umbral de confianza para abstenerse en ensayos ambiguos. Dicho umbral se ajusta usando únicamente predicciones *out-of-fold* del entrenamiento, buscando un compromiso entre desempeño en aceptados, cobertura (tasa de rechazo) y calibración, y se evalúa externamente en el *hold-out*. Este protocolo repetido permite estimar el desempeño con menor varianza que un único *hold-out* y analizar simultáneamente discriminación (LMI vs. RMI), estabilidad frente a particiones y fiabilidad bajo abstención en un escenario de datos pequeños típico en BCI.

5.3 Resultados de clasificación sin rechazo

Para cada sujeto se generan 15 particiones estratificadas y, aunque en cada una se calcula la *accuracy* sin rechazo en el *hold-out,* el *split* exportado no es el que obtiene la mayor *accuracy*. En su lugar, se selecciona el *split* que maximiza un criterio de utilidad coherente con el objetivo real del sistema: que cuando el modelo “decide”, lo haga de forma más fiable y con probabilidades bien calibradas.

En concreto, el criterio de selección del *split* pondera en primer lugar, el desempeño discriminativo en el subconjunto aceptado, en segundo lugar, una cobertura controlada mediante una tasa de abstención/rechazo cercana a un objetivo, y en tercer lugar, la calidad probabilística (bajo error de calibración), todo evaluado en el *hold-out* del *split*. En síntesis, el Split escogido es el que cumpla todos los anteriores puntos de la mejor forma en comparación a los demás.

Durante el proceso de entrenamiento y selección de hiperparámetros se utilizó balanced accuracy (BA) como métrica de optimización. Esta métrica se eligió porque permite evaluar el desempeño equilibrado entre clases, evitando que el modelo favorezca una clase sobre otra en escenarios con pocos datos o variabilidad entre particiones. Sin embargo, para facilitar la comparación con trabajos previos reportados en la literatura de BCI–fNIRS, los resultados también se presentan en términos de exactitud (accuracy), ya que esta métrica es la más comúnmente reportada en estudios similares. De esta forma, BA se utiliza como criterio interno de optimización del modelo, mientras que la exactitud se emplea como métrica descriptiva para contextualizar el desempeño obtenido frente a otros estudios.

Para cada sujeto se seleccionó el modelo exportado con una métrica de utilidad selectiva que combina rendimiento en predicciones aceptadas y comportamiento de calibración. Esto evita elegir splits por azar (con pocos ensayos en prueba) y mantiene la selección alineada con el uso posterior de confianza y rechazo.

Tabla 5-1, Exactitud sin rechazo por sueto.

Sujeto	Ventana ganadora	CV interna (media±std)	Acc hold-out (media±std, 15 splits)	Acc hold-out (sin rechazo)	ECE calibración de la probabilidad
1	C	0.645 ± 0.080	0.493 ± 0.121	0.6	0.114
2	D	0.695 ± 0.048	0.502 ± 0.126	0.667	0.150
3	D	0.740 ± 0.110	0.502 ± 0.135	0.733	0.222
4	D	0.865 ± 0.078	0.511 ± 0.147	0.733	0.214
5	D	0.665 ± 0.159	0.462 ± 0.120	0.667	0.136
6	K	0.740 ± 0.086	0.480 ± 0.109	0.733	0.173
7	B	0.665 ± 0.093	0.547 ± 0.109	0.6	0.067
8	E	0.775 ± 0.089	0.507 ± 0.131	0.733	0.159
9	E	0.725 ± 0.076	0.498 ± 0.097	0.667	0.121
10	E	0.775 ± 0.133	0.542 ± 0.076	0.733	0.191
11	B	0.725 ± 0.055	0.569 ± 0.094	0.867	0.350
12	C	0.710 ± 0.077	0.538 ± 0.149	0.867	0.324
13	L	0.680 ± 0.068	0.529 ± 0.130	0.733	0.131
14	B	0.715 ± 0.049	0.520 ± 0.167	0.867	0.323
15	K	0.700 ± 0.150	0.476 ± 0.133	0.667	0.318
16	K	0.725 ± 0.042	0.467 ± 0.126	0.6	0.123
17	B	0.685 ± 0.080	0.476 ± 0.119	0.667	0.281
18	D	0.730 ± 0.135	0.467 ± 0.097	0.6	0.082
19	F	0.795 ± 0.083	0.582 ± 0.137	0.8	0.232
20	E	0.775 ± 0.101	0.458 ± 0.187	0.8	0.241
21	L	0.720 ± 0.048	0.493 ± 0.128	0.733	0.147
22	B	0.780 ± 0.073	0.493 ± 0.094	0.733	0.203
23	C	0.795 ± 0.109	0.542 ± 0.087	0.733	0.199
24	D	0.800 ± 0.130	0.524 ± 0.163	0.6	0.184
25	K	0.770 ± 0.102	0.538 ± 0.120	0.8	0.260
26	B	0.830 ± 0.132	0.480 ± 0.144	0.533	0.128
27	F	0.675 ± 0.115	0.542 ± 0.084	0.6	0.347
28	F	0.735 ± 0.093	0.498 ± 0.111	0.667	0.123

Teniendo en cuenta la Tabla 5-1, primero que todo, se evidencia que para cada sujeto la selección de ventanas temporales es automática y específica, y ocurre dentro del Nested Cross-Validation. El criterio no es fijo: el algoritmo prueba varias ventanas y se queda con la que produce mayor balanced accuracy en validación. En entrenamiento (CV interna con BA) la exactitud inter-sujeto se sitúa en 0.737 ± 0.093, con una desviación estándar aceptable, lo que sugiere un desempeño relativamente consistente entre sujetos durante la validación interna. Al pasar al conjunto de prueba (*hold-out*) del *split* exportado por sujeto, la exactitud se ubica en 0.705 (rango 0.533–0.867), observando así una brecha recurrente entre el desempeño interno y el rendimiento en el hold-out por sujeto.

Este patrón es esperable cuando se optimizan múltiples decisiones (p. ej., ventanas temporales, canales y selección de variables) sobre el conjunto de entrenamiento: aunque el esquema anidado reduce la fuga de información, el ajuste interno puede favorecer configuraciones que capturan particularidades del entrenamiento y, al evaluarse en un hold-out pequeño, el rendimiento externo puede oscilar notablemente [59]. Por tanto, esta diferencia no debe interpretarse como un fallo metodológico, sino como evidencia de un escenario estadísticamente exigente (pocas muestras, alta variabilidad) y de que forzar decisiones en todos los ensayos

expone casos ambiguos donde la confianza no generaliza de forma estable [39], [59].

En términos prácticos, la gran mayoría de los sujetos supera el umbral de azar (0.5): 27/28 sujetos alcanzan al menos 0.60, más de la mitad logran ≥0.733, y un subconjunto alcanza ≥0.80. Este comportamiento sugiere que, para ciertos sujetos, el pipeline logra extraer patrones discriminativos estables entre LMI y RMI incluso cuando se fuerza una decisión en todos los ensayos.

Es importante notar que este desempeño se mide sobre un hold-out pequeño (15 ensayos), por lo que cada error o acierto modifica la exactitud en pasos de 1/15 ≈ 0.0667; esto hace que el indicador sea inherentemente "granular" y sensible a pocos ensayos, una limitación esperable en escenarios de BCI con N reducido [39], [59].

Al promediar sobre las 15 particiones estratificadas, la exactitud media por sujeto (*Acc hold-out mean*) se ubica alrededor de 0.508 ± 0.123, con máximos típicamente por debajo de ~0.60. Esta aparente caída respecto al *split* exportado es consistente con dos fenómenos esperables en este régimen:

- **a)** Se observó alta varianza entre splits por sujeto, coherente con el régimen de pocas muestras (≈45 ensayos para entrenar y ≈15 para probar por split). En este contexto, la composición exacta del conjunto de entrenamiento (qué ensayos entran) puede modificar de forma apreciable los parámetros óptimos (ventana, canales, subconjunto de características) aprendidos dentro del entrenamiento, y por ende desplazar el desempeño en prueba. De hecho, la desviación estándar *intra-sujeto* a través de *splits* es elevada (≈0.12 en promedio, con casos cercanos a 0.19), lo cual indica que el rendimiento depende fuertemente del muestreo. Esta sensibilidad es un resultado clásico cuando se trabaja con pocas muestras y alta dimensionalidad relativa, y justifica precisamente el uso de esquemas repetidos/*nested* para cuantificar incertidumbre y evitar conclusiones basadas en una sola partición [59].
- **b)** No-estacionariedad y variabilidad ensayo-a-ensayo en fNIRS. Aun tras preprocesamiento, las señales hemodinámicas están afectadas por variaciones fisiológicas lentas (derivas), cambios en acoplamiento óptico, respiración/cardiaca y confusores sistémicos que pueden variar entre ensayos y sesiones. Estas fuentes de variabilidad reducen el "tamaño de efecto" estable entre clases a nivel de ensayo único y hacen que algunos *splits* queden accidentalmente con entrenamiento "favorable" y otros con entrenamiento "desfavorable", especialmente cuando el número de ensayos es limitado [29], [30], [31]. En otras palabras: el promedio sobre 15 *splits* refleja el desempeño típico bajo re-muestreo, mientras que un split puntual puede capturar una partición particularmente alineada con el patrón discriminativo del sujeto.

La calibración (ECE) fue heterogénea entre sujetos, con valores moderados a altos (típicamente ~0.19 y con casos >0.30). Esta variabilidad es esperable en BCI-fNIRS y se explica por: (i) el N reducido de ensayos, que incrementa la varianza de la estimación por *bins (intervalos de confianza)* y puede sesgar el ECE según el *binning* [63], [64]; (ii) confusores fisiológicos y variabilidad hemodinámica, que pueden inducir sobre/sub-confianza del clasificador según los patrones dominantes en cada partición [68]–[70]; y (iii) que los modelos discriminativos optimizan principalmente separación/accuracy, no probabilidades bien calibradas, por lo que suele requerirse calibración explícita [62], [65]. En general, estas dificultades se reportan ampliamente por el bajo SNR y la necesidad de técnicas específicas de manejo de ruido/artefactos y variabilidad [67].

Nota: el ECE reportado de forma agrupada (*pooled*) no equivale al promedio de los ECE por sujeto, porque la métrica depende de la ocupación de *bins* y de promedios dentro de cada uno de estos. Por ello, se reportan por separado métricas por sujeto y métricas globales agrupadas.

En este marco, incorporar calibración y criterios basados en fiabilidad busca reducir el desajuste entre confianza y acierto: al alinear probabilidades con exactitud empírica se vuelve legítimo aplicar umbrales de decisión (p. ej., para abstención/rechazo) sin forzar predicciones altamente confiadas pero incorrectas, en línea con la evaluación moderna de clasificación selectiva y los *trade-offs* (compromiso) riesgo–cobertura [58]. Que el promedio sobre *splits* se acerque al azar no implica necesariamente que el sistema “no funcione”; más bien indica que la generalización es frágil en un régimen de pocas muestras y alta variabilidad fisiológica/experimental, por lo que reportar media ± desviación estándar sobre múltiples particiones es una buena práctica para cuantificar estabilidad y evitar sobreinterpretar resultados aislados [59].

Las matrices de confusión resultantes obtenidas para el sujeto 1 del *hold-out*. para el escenario sin rechazo se incluyen en el **Anexo 8** (Figuras A8.1); el banco completo se entrega en el medio digital descrito en el Anexo 1. La matriz de confusión observada de la figura A8.1 produce una exactitud de 9/15, es decir, el 60% de exactitud en precisión y un patrón de error aproximadamente simétrico entre clases. Este resultado es consistente con un sujeto de separabilidad moderada, donde la señal discriminativa es limitada y parte de los ensayos se comporta como ambigua bajo decisión forzada, fenómeno frecuente en fNIRS por confusores fisiológicos y variabilidad inter-ensayo [29], [30].

Por último, se presenta a continuación la matriz de confusión global de todo el dataset.

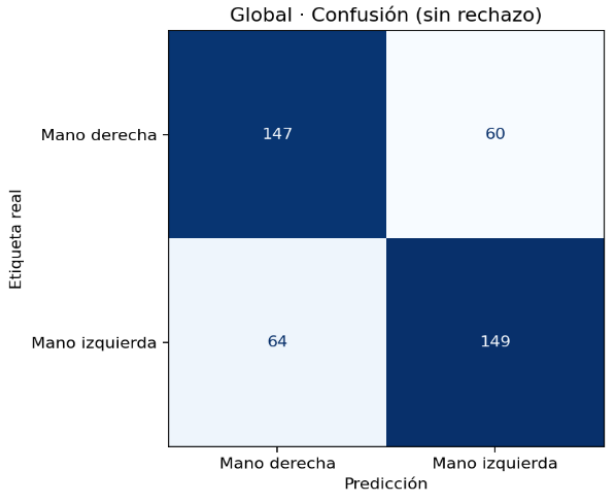


Figura 5-1. Matriz de confusión sin rechazo global.

La matriz global agregada en la Figura 5-1 arroja una exactitud global de 296/420 de aciertos, que se traduce como un desempeño global de 70.5% de efectividad en aciertos, muestra sensibilidad similar en ambas clases (comportamiento relativamente equilibrado).

En conjunto, los resultados sin abstención sugieren que existe señal discriminativa explotable en varios sujetos, pero con alta sensibilidad a la partición; esto justifica analizar posteriormente el sistema bajo decisión selectiva, permitiendo abstenerse en ensayos de baja evidencia en lugar de imponer una predicción potencialmente errónea [58], [59].

5.4 Impacto del rechazo

Esta sección presenta el desempeño global del sistema cuando se habilita el mecanismo de rechazo (abstención) basado en la confianza/probabilidad de salida del clasificador. A diferencia de un esquema forzado (que siempre decide), la evaluación selectiva permite cuantificar simultáneamente la capacidad discriminativa cuando el sistema decide y la cobertura operativa (proporción de ensayos aceptados), aspectos críticos en BCI-fNIRS por el ruido y la alta variabilidad inter-sujeto. Dado que la salida probabilística gobierna decisiones de aceptar/rechazar, la evaluación no se limita a exactitud, sino que incorpora métricas de incertidumbre como curvas riesgo–cobertura y calibración probabilística [56]–[58].

Con rechazo adaptativo, la exactitud condicional sobre ensayos aceptados aumenta, porque el sistema abstiene preferentemente los ensayos con menor

confianza. Este comportamiento se refleja en las curvas riesgo–cobertura y permite ajustar el compromiso entre cobertura y confiabilidad según la aplicación.

Tabla 5-2. Exactitud y métricas con rechazo por sujeto.

Sujeto	θ (umbral)	Rechazo	Acc sin rechazo	Acc aceptados	BA aceptados	ECE aceptados	AURC	Utilidad
1	0.526	0.156	0.6	0.692	0.69	0.035	0.22	0.682
2	0.517	0.089	0.667	0.769	0.812	0.147	0.196	0.753
3	0.502	0.133	0.733	0.714	0.714	0.202	0.394	0.664
4	0.505	0.156	0.733	0.769	0.774	0.262	0.375	0.664
5	0.512	0.156	0.667	0.692	0.69	0.354	0.45	0.58
6	0.508	0.156	0.733	0.786	0.786	0.221	0.298	0.73
7	0.512	0.089	0.6	0.692	0.679	0.155	0.272	0.595
8	0.507	0.089	0.733	0.786	0.786	0.206	0.142	0.734
9	0.511	0.156	0.667	0.714	0.708	0.185	0.185	0.64
10	0.518	0.156	0.733	0.75	0.75	0.258	0.244	0.641
11	0.504	0.133	0.867	0.846	0.845	0.34	0.209	0.738
12	0.505	0.133	0.867	0.846	0.857	0.298	0.048	0.738
13	0.563	0.133	0.733	0.75	0.75	0.113	0.194	0.677
14	0.507	0.111	0.867	0.867	0.866	0.323	0.135	0.719
15	0.51	0.067	0.667	0.769	0.762	0.352	0.344	0.674
16	0.506	0.133	0.6	0.692	0.714	0.12	0.246	0.662
17	0.519	0.133	0.667	0.75	0.786	0.33	0.302	0.681
18	0.508	0.089	0.6	0.667	0.667	0.157	0.432	0.583
19	0.51	0.156	0.8	0.786	0.786	0.214	0.127	0.732
20	0.511	0.156	0.8	0.923	0.938	0.356	0.076	0.826
21	0.55	0.111	0.733	0.75	0.812	0.152	0.277	0.73
22	0.504	0.156	0.733	0.714	0.714	0.182	0.305	0.669
23	0.51	0.067	0.733	0.833	0.8	0.291	0.096	0.683
24	0.516	0.111	0.6	0.692	0.714	0.097	0.362	0.646
25	0.515	0.111	0.8	0.769	0.812	0.268	0.312	0.723
26	0.524	0.133	0.533	0.615	0.607	0.199	0.22	0.535
27	0.568	0.067	0.6	0.667	0.714	0.16	0.423	0.63
28	0.513	0.089	0.667	0.692	0.714	0.121	0.29	0.662

En la partición seleccionada observada en la Tabla 5-2, la tasa de rechazo promedio es de aproximadamente 12.2%, con rango [6.7%, 15.6%], lo que equivale a una cobertura media de 87.8%. Dado que el conjunto de prueba por sujeto es pequeño (≈15 ensayos), la tasa de rechazo tiende a observarse en valores discretos (aprox. 1–2 ensayos rechazados), lo cual es esperable y debe considerarse al interpretar métricas como ECE y AURC, que dependen del número de predicciones aceptadas [56], [59].

Al introducir el rechazo, la exactitud condicionada a aceptados aumenta a 0.750 ± 0.069 (en un rango mín–máx: 0.615–0.923). De forma consistente, la balanced accuracy en aceptados alcanza 0.759 ± 0.071 (mín–máx: 0.607–0.938), lo que es relevante porque BA refleja el equilibrio de sensibilidad entre clases y evita conclusiones dominadas por una clase cuando existe asimetría de error [60]. En términos prácticos, el rechazo logra el efecto esperado: filtra ensayos de baja evidencia y mejora la fiabilidad de las decisiones emitidas, en línea con el marco riesgo–cobertura de la clasificación selectiva [58], [59]. En este conjunto, la exactitud

en aceptados mejora en 21/28 sujetos se mantiene prácticamente igual en 1/28 y disminuye levemente en 6/28 (caídas pequeñas, típicamente cuando el desempeño forzado ya es alto y el rechazo puede retirar ensayos correctos) [58], [59]. Desde la perspectiva de utilidad selectiva, los sujetos mejor posicionados son aquellos en los que un rechazo moderado produce incrementos claros de exactitud en aceptados, alineándose con el objetivo de priorizar decisiones confiables [58].

La calibración en aceptados presenta $ECE_{acc} = 0.218 \pm 0.089$ (rango 0.035–0.356). Esto indica que, aunque el rechazo mejora el desempeño en el subconjunto aceptado, la alineación confianza–precisión varía sustancialmente entre sujetos. Es importante subrayar que una alta exactitud no garantiza buena calibración; por ello, cuando la confianza gobierna el rechazo, se debe evaluar explícitamente ECE y no asumir que el puntaje probabilístico es interpretable [56], [57]. Además, con pocas predicciones aceptadas, ECE puede ser ruidosa (dependencia de binning y tamaño muestral), por lo que su lectura debe acompañarse con AURC y con análisis cualitativos por sujeto [56], [59].

El indicador AURC promedio es 0.256 ± 0.109 (mín–máx: 0.048–0.450). Un AURC menor implica que, al reducir cobertura (rechazar más), el riesgo cae de manera más eficiente, lo cual sugiere que la confianza ordena adecuadamente ejemplos fáciles/difíciles. Por el contrario, AURC altos suelen reflejar presencia de errores "confiados" (mala incertidumbre), limitando la ganancia del rechazo basado únicamente en umbral [58], [59]. En conjunto, estos resultados justifican evaluar el sistema no solo por exactitud, sino por su comportamiento selectivo completo (riesgo, cobertura y calibración), que es el objetivo operativo de un BCI robusto [56]–[58].

El umbral θ se concentra en un intervalo estrecho (0.516 ± 0.017, rango 0.502–0.568), lo cual sugiere que el criterio de selección prioriza consistencia probabilística y control de cobertura, evitando depender de una partición que, por azar, maximice la exactitud de prueba [59]. Esta elección es metodológicamente preferible cuando el sistema se apoya en probabilidades para regular la abstención [56]–[58].

5.4.1 Análisis intra-sujeto

Los paneles para los sujetos 1 que resume la evaluación con rechazo (matriz de confusión con abstención, riesgo–cobertura, *trade-off* (compromiso) exactitud–rechazo y calibración) se incluyen en el Anexo 9 (Figura A9.1); el banco completo se entrega en el medio digital descrito en el Anexo 1

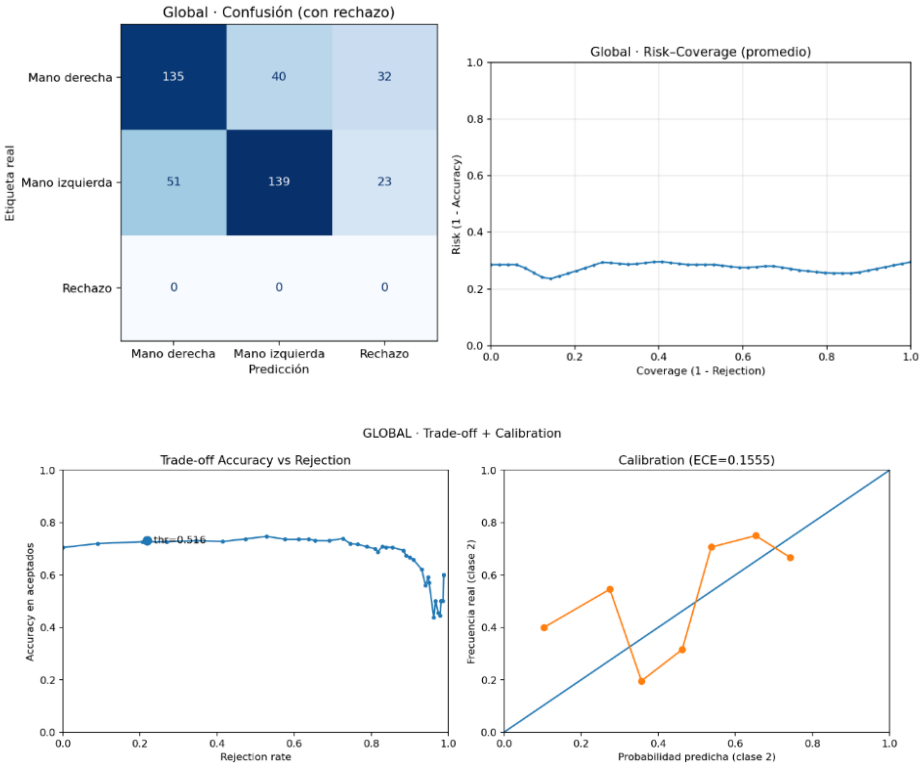
Teniendo en cuenta la Figura A9.1, la matriz de confusión muestra un desempeño moderado con rechazo no nulo: hay errores cruzados en ambas direcciones y aparecen predicciones en la columna Rechazo, confirmando que el umbral θ filtra ensayos ambiguos. En términos funcionales, el rechazo actúa como una "válvula de seguridad" al evitar decisiones forzadas cuando la evidencia es débil, lo cual es pertinente en MI-fNIRS por la alta variabilidad ensayo-a-ensayo [30], [58], [59].

La curva risk–coverage presenta el comportamiento esperado: al disminuir cobertura (mayor rechazo), el riesgo tiende a bajar. Un AURC intermedio sugiere que la confianza separa parcialmente ensayos fáciles y difíciles, aunque no de forma perfecta, por lo que aún persisten algunos errores incluso bajo mayor rechazo (caso típico cuando existen predicciones incorrectas con confianza moderada) [58], [59]. El punto operativo marcado ($\theta \approx 0.526$) refleja un compromiso razonable: rechazo moderado con mejora de exactitud en aceptados, y la curva sugiere rendimientos decrecientes al aumentar excesivamente el rechazo, indicando que a partir de cierto nivel también se descartan ensayos potencialmente correctos [58].

En cuanto a fiabilidad probabilística, la calibración global es moderada (ECE no nulo), lo cual es esperable en conjuntos de prueba pequeños [56], [59]. Sin embargo, al aplicar rechazo, el subconjunto aceptado queda mejor calibrado, que es el efecto deseado: las decisiones emitidas son más interpretables en términos de "confianza ≈ precisión", otorgando significado operativo al umbral θ [56]–[58]. En síntesis, para el sujeto 1 el rechazo mejora la calidad de las decisiones y la calibración de los aceptados, con la limitación de que el ranking por confianza no elimina por completo los errores (AURC intermedio).

5.4.2 Resultados globales del sistema con rechazo (abstención)

A continuación, se presentan las métricas globales del sistema, de las cuales se podrá analizar la capacidad de discriminación de la BCI entre LMI y RMI.



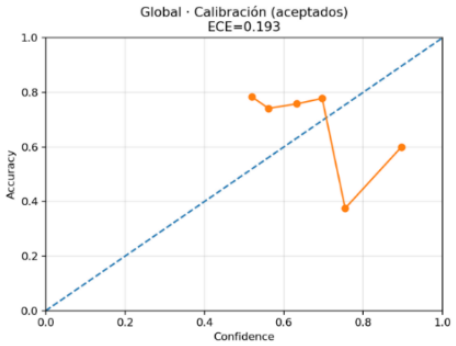


Figura 5-2. Métricas globales de la BCI – análisis de desempeño.

Tomando en cuenta la Figura 5-2, la matriz de confusión global integra ambas clases (mano derecha e izquierda) e incorpora explícitamente la salida Rechazo, permitiendo estimar cobertura y desempeño condicional cuando el sistema decide. Se evalúan 420 ensayos, de los cuales 365 son aceptados (cobertura 0.869) y 55 rechazados (tasa de rechazo 0.131). Sobre los aceptados se observan 274 aciertos, lo que corresponde a Acc(aceptados) ≈ 0.751. Por clase, las sensibilidades sobre aceptados son comparables (TPR(derecha) ≈ 0.771, TPR(izquierda) ≈ 0.732), resultando en BA(aceptados) ≈ 0.752, métrica apropiada para balancear desempeño entre condiciones [60].

El rechazo no es idéntico entre clases: la tasa condicionada a “mano derecha” es mayor que para “mano izquierda”, sugiriendo que la distribución de confianza no es simétrica. En MI-fNIRS esto puede deberse a variabilidad inter-sujeto/ensayo, lateralización y calidad de canales motores, por lo que es pertinente reportar rechazo por clase y no solo global [30], [58], [59]. La curva Risk–Coverage muestra una tendencia relativamente plana en gran parte del rango, indicando que la confianza separa solo parcialmente aciertos y errores (persisten errores con confianza moderada/alta), lo que limita la ganancia marginal del rechazo en un contexto de SNR bajo y muestras pequeñas [58], [59].

Finalmente, el panel Accuracy vs Rejection rate ubica el umbral seleccionado ($\theta \approx 0.516$) en una zona de meseta con rechazo moderado, donde se mejora la confiabilidad sin sacrificar excesivamente la cobertura; para rechazos extremos la estimación se vuelve inestable por el bajo número de aceptados [59]. En consecuencia, la elección de θ se interpreta como un compromiso operativo que controla riesgo manteniendo una cobertura compatible con el uso BCI, en línea con clasificación selectiva [58].La calibración evalúa si la probabilidad reportada por el clasificador es interpretable como frecuencia real de acierto.

A nivel global, la calibración agrupada (ECE pooled) indica una descalibración moderada, ya que se encuentra en torno 0.156, que en algunos rangos de confianza, la exactitud observada no coincide con la probabilidad predicha [62], [63]. Dado que el rechazo se define en función de la confianza, esta propiedad es crucial,

si el modelo es sobreconfiado o subconfiado, un umbral fijo puede resultar subóptimo o inestable entre particiones/sujetos [62], [66].

Al restringir el análisis al subconjunto aceptado, el ECE pooled aumenta a ≈ 0.193. Este comportamiento es compatible con dos efectos conocidos: (i) el conjunto aceptado es una población condicionada (cambia la distribución de confianzas y la ocupación de bins), y (ii) al reducirse el número de muestras por bin, la estimación de confiabilidad se vuelve más ruidosa [56], [57], [59]. En consecuencia, aunque el rechazo eleva la exactitud condicional, la calibración en aceptados sugiere que la confianza sigue siendo “imperfectamente probabilística”, reforzando la necesidad de evaluar conjuntamente exactitud, calibración y riesgo–cobertura [56], [58].

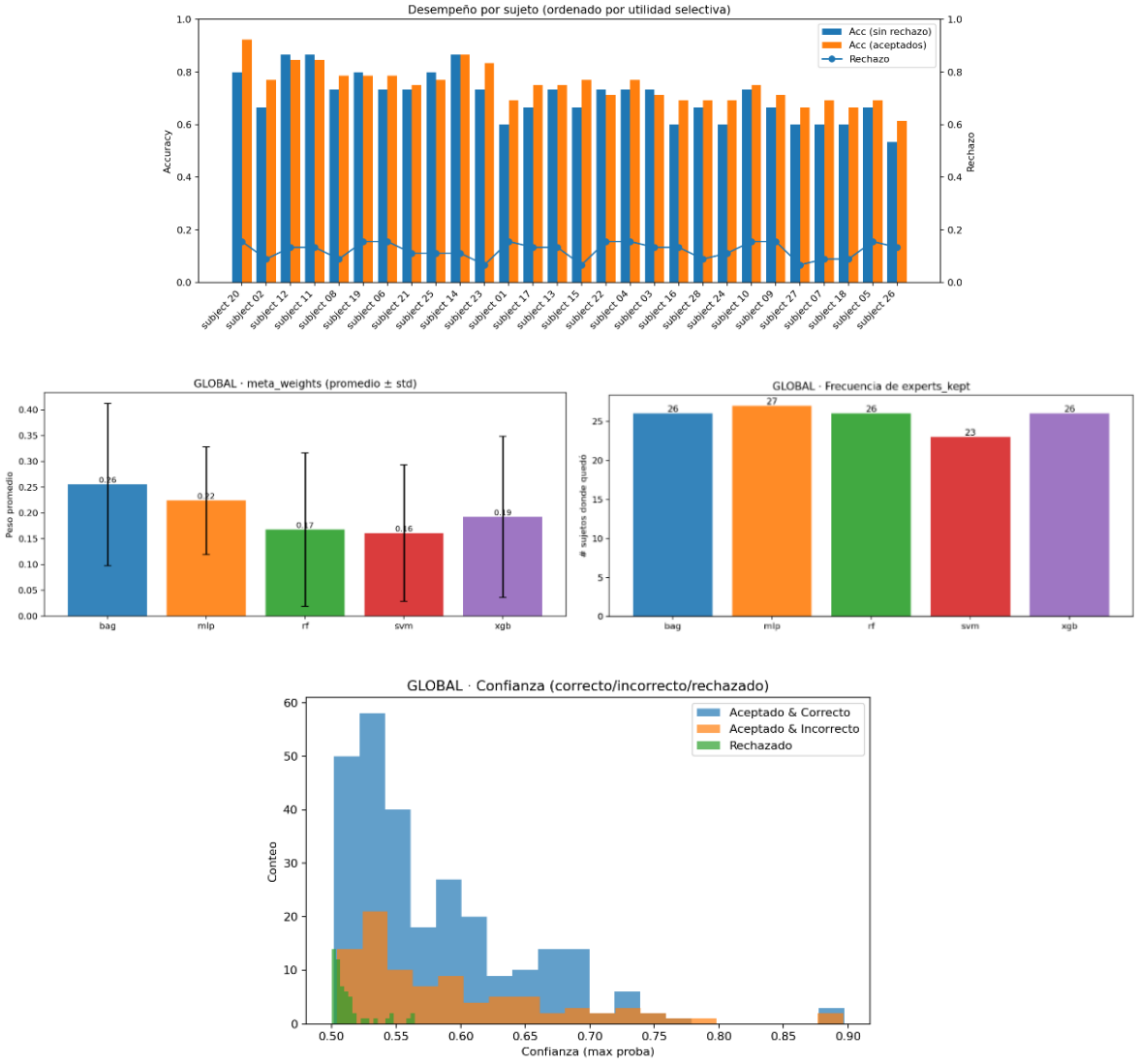


Figura 5-3, Métricas globales de la BCI – análisis del meta-modelo.

La Figura 5-3 muestra, por sujeto, la comparación entre exactitud sin rechazo y exactitud en ensayos aceptados junto con la tasa de rechazo, la cual varía entre sujetos, coherente con la heterogeneidad propia de MI-fNIRS: sujetos con mayor separabilidad requieren menos abstención, mientras que sujetos con menor SNR o lateralización menos marcada requieren rechazos mayores para controlar el riesgo [30], [59]. En la mayoría de sujetos se cumple *Acc(aceptados) > Acc(sin rechazo)*, evidenciando que la abstención mejora la confiabilidad cuando el sistema decide, a costa de reducir cobertura.

Desde la perspectiva de la aplicabilidad usuario de la BCI, este comportamiento puede resultar beneficioso, ya que disminuye la probabilidad de comandos erróneos que podrían generar frustración o pérdida de confianza en el sistema. En su lugar, el sistema opta por no emitir una decisión cuando la evidencia es insuficiente, priorizando decisiones más confiables aun cuando ello implique una menor frecuencia de comandos emitidos. Este compromiso entre confiabilidad y cobertura es especialmente relevante en aplicaciones de BCI para control interactivo, donde los errores de clasificación suelen ser percibidos por el usuario como más perjudiciales que la ausencia temporal de respuesta.

El análisis del meta-modelo muestra que la decisión no depende de un único experto. Los pesos promedio asignan mayor contribución a *bagging (~0.26) y MLP (~0.22),* seguidos por *XGB (~0.19),* con aportes menores de *RF (~0.17) y SVM (~0.16);* las desviaciones estándar reflejan variabilidad inter-sujeto, consistente con que la “mejor inductiva” cambia según características fisiológicas y calidad de señal [59]. Además, el MLP es el experto más frecuente, presente en 27 de 28 sujetos.

El histograma de confianza evidencia el funcionamiento del rechazo: los ensayos rechazados se concentran cerca del umbral (≈0.50–0.55), mientras que los aceptados correctos tienden a presentar confianzas más altas. No obstante, existe solapamiento entre aceptados correctos e incorrectos, incluyendo errores con confianza moderada/alta; esto explica que la ganancia del rechazo sea limitada y que la curva riesgo–cobertura no sea muy pronunciada, pues la confianza aún no ordena perfectamente aciertos y errores [58], [59].

En términos globales, el sistema con rechazo opera con cobertura alta y mejora la exactitud condicional en aceptados. No obstante, la forma de la curva riesgo–cobertura y la calibración muestran que aún pueden ocurrir errores con confianza moderada, por lo que la confianza no separa perfectamente ensayos correctos e incorrectos. Esto refuerza la necesidad de calibración y evaluación selectiva como parte central del pipeline.

5.5 Análisis comparativo y general del sistema

De forma global, el sistema implementa una BCI unimodal fNIRS por sujeto y reproducible, que transforma intensidades ópticas en señales hemodinámicas (ΔHbO/ΔHbR), aplica corrección de artefactos, filtrado hemodinámico, segmentación por eventos, extracción de características temporales e inferencia mediante un comité de clasificadores con *stacking*. Su operación está diseñada para un escenario BCI realista: se decide solo cuando hay evidencia suficiente, usando probabilidades calibradas y opción de rechazo/abstención [56], [58]–[60].

A continuación, en la Tabla 5-3 se compara el desempeño y la lógica de selección del "modelo final" bajo dos estrategias: (i) seleccionar el *split* por mayor *accuracy* y (ii) seleccionar el *split* por mayor utilidad selectiva (desempeño en aceptados + control de rechazo + calibración), que es la estrategia alineada con operación con rechazo [58].

Tabla 5-3. Comparación global del sistema según criterio de selección del split (28 sujetos)

Aspecto	Modelo que selecciona split por mejor Accuracy	Modelo que selecciona split por mejor Utilidad selectiva
Criterio de selección del split exportado	Maximiza *accuracy* puntual en hold-out (decisión forzada).	Maximiza utilidad selectiva: desempeño en aceptados + cercanía a rechazo objetivo + penalización por ECE [56], [58].
Condiciones comunes de operación	- f_s = 10 Hz; épocas −10 a 25 s; baseline −3 a 0 s; ventanas del catálogo HRF; 24 canales motores; 7 descriptores/canal; *stacking* de expertos + meta-logístico calibrado [59], [60].	*Iguales* (mismo pipeline y "máquina de aprendizaje").
Salida operativa	Predicción binaria (L/R) + rechazo por umbral, pero el split fue elegido sin optimizar explícitamente la utilidad bajo rechazo.	Predicción binaria (L/R) + rechazo optimizado por sujeto, coherente con operación selectiva ("decidir solo si es confiable") [58].
Accuracy forzada en prueba (sin rechazo)	0.729 ± 0.060 [0.667–0.867]	0.705 ± 0.090 [0.533–0.867]
BA forzada en prueba (sin rechazo)	0.732 ± 0.060 [0.670–0.875]	0.707 ± 0.090 [0.527–0.875]
Accuracy en aceptados (con rechazo)	0.725 ± 0.075 [0.583–0.867]	0.750 ± 0.069 [0.615–0.923]
BA en aceptados (con rechazo)	0.727 ± 0.083 [0.557–0.866]	0.759 ± 0.071 [0.607–0.938]
Tasa de rechazo ρ	0.148 ± 0.052 [0.000–0.200]	0.131 ± 0.053 [0.000–0.200]
Cobertura (1−ρ)	0.852 ± 0.052 [0.800–1.000]	0.869 ± 0.053 [0.800–1.000]
Calibración (ECE)	0.198 ± 0.073 [0.090–0.339]	0.198 ± 0.082 [0.067–0.350] *(similar; aquí se controla explícitamente en la utilidad)*
BA interna en entrenamiento (nested-CV)	0.734 ± 0.052 [0.585–0.830]	0.737 ± 0.053 [0.645–0.865]
Ventaja principal (para uso con rechazo)	Mejor desempeño forzado, útil si siempre se debe emitir comando.	Mejor desempeño cuando el sistema acepta (↑Acc/↑BA en aceptados) y selección alineada con decisión confiable, clave para control BCI con abstención [58].

En la comparación entre el modelo que selecciona el *split* por mayor *accuracy* y el que selecciona el *split* por mayor utilidad selectiva, se observa un compromiso o *trade-off* coherente con un uso BCI real: el primero puede exhibir mejor desempeño forzado (cuando siempre debe emitir izquierda/derecha), pero su selección es más propensa a ser optimista y frágil porque se apoya en un resultado puntual de prueba.

En contraste, el modelo basado en utilidad selectiva puede sacrificar parte del accuracy sin rechazo, pero se comporta mejor bajo rechazo: cuando el sistema decide (ensayos aceptados), tiende a lograr mayor exactitud y *balanced accuracy*, con una cobertura/rechazo controlados y una lógica de decisión alineada con la confiabilidad probabilística. En términos operativos, esto es preferible para control de un avatar o comandos discretos, porque prioriza aciertos confiables sobre "aciertos totales" y reduce el costo funcional de errores, siguiendo el principio de clasificación con opción de rechazo: es mejor abstenerse ante evidencia insuficiente que forzar una decisión potencialmente incorrecta [56], [58]–[60].

A continuación, en la siguiente tabla se compararán los resultados de este trabajo en contraste a los proyectos consultados en los antecedentes.

Tabla 5-4. Comparación de desempeños respecto a los antecedentes.

Trabajo	Modalidad	Clasificador	Accuracy reportada
[31]	fNIRS MI unimodal	LDA / SVM	73–87 %
[32]	fNIRS dataset para movimientos unilaterales de mano	Varios (LDA, SVM, NN)	70–90 %
[36]	fNIRS MI	CNN 1D	72 ± 4 %
Este trabajo	fNIRS MI unimodal	Stacking + rechazo selectivo	Acc = 75.0 ± 6.9 % / BA = 75.9 ± 7.1 %

La Tabla 5-4 compara el desempeño obtenido en este trabajo con resultados reportados en la literatura para BCIs basadas en fNIRS en paradigmas de imaginación motora. Los estudios existentes reportan exactitudes típicamente en el rango de 70–90 %, dependiendo del número de canales, ventanas temporales y clasificador utilizado [31], [32], [36].

El modelo propuesto en este trabajo alcanza una exactitud promedio de 75.0 ± 6.9 % (BA = 75.9 ± 7.1 %) en los ensayos aceptados, lo cual se encuentra dentro del rango de desempeño reportado para sistemas fNIRS unimodales.

A diferencia de muchos trabajos previos, el sistema propuesto incorpora un esquema de clasificación selectiva con rechazo, que prioriza decisiones confiables frente a la emisión obligatoria de comandos. Por esta razón, el sistema puede presentar una cobertura menor, pero mejora la confiabilidad de las decisiones aceptadas, lo cual es deseable en aplicaciones de control BCI.

5.6 Validación funcional con el avatar en Unity

Finalmente, se realizó una validación funcional del sistema empleando los comandos generados por el clasificador para cada sujeto y reproduciéndolos en un entorno virtual desarrollado en Unity. Para ello, se cargan en la escena los archivos *commands_subjectXX_clean.csv*, que contienen, por ensayo, la etiqueta real, la

predicción y el comando discreto asociado. El avance ensayo por ensayo se realiza con el script *AvatarBCIController*, el cual lee secuencialmente el CSV y activa en el *Animator* del avatar la animación correspondiente. En pantalla se despliega un panel informativo que resume el estado del ensayo actual (sujeto, índice de ensayo, etiqueta real, predicción, comando aplicado y estado del sistema), permitiendo una verificación visual inmediata.

Esta etapa no pretende reemplazar la evaluación cuantitativa (matrices de confusión y métricas), sino complementarla: permite observar cómo las decisiones del clasificador se traducen en acciones visibles, cómo se expresan los errores y cómo opera el estado de "no decisión" *(idle)* asociado al rechazo. Además, evidencia la factibilidad de integrar una BCI-fNIRS en un flujo de control para aplicaciones interactivas, incluso cuando el procesamiento aún se ejecuta en modo offline.

5.6.1 Descripción de escenas y máquina de estados



Figura 5-4. Ejecución del avatar en Unity.

En la Figura 5-4 e muestra la escena inicial tras cargar el sujeto, confirmando el estado "listo" y solicitando al usuario avanzar con la tecla *SPACE***.** Esta pantalla funciona como control de ejecución y verificación de que el CSV del sujeto fue encontrado y leído correctamente antes de iniciar la reproducción de ensayos. El panel en pantalla reporta simultáneamente la etiqueta real y la predicción, junto con el comando enviado al *Animator*. El estado "Correcto" indicaría consistencia entre el comando aplicado y la lógica de control (y, en ensayos aceptados, coincide típicamente con predicción correcta). La lectura del CSV opera de forma homogénea para múltiples sujetos, manteniendo el despliegue de información por ensayo y la activación reproducible de la animación.

Por otro lado, cuando la predicción no coincide con la etiqueta real (o cuando el sistema determina que debe reportarse discrepancia), se activa explícitamente la animación Error (voltear cabeza) y el panel indica el estado "ERROR". Esta representación visual permite identificar de manera inmediata los ensayos problemáticos y relacionarlos posteriormente con los patrones observados en las

matrices de confusión. Por último, en ensayos donde la confianza no supera el umbral definido, el sistema no ejecuta una clase motora y asigna el estado Idle, actuando como una salida funcional equivalente a “no decisión”. En esa escena se podrá evidenciar el comportamiento esperado de la opción de rechazo: evitar forzar un comando motor cuando la evidencia es insuficiente, manteniendo el avatar en reposo y registrando el evento en la interfaz.

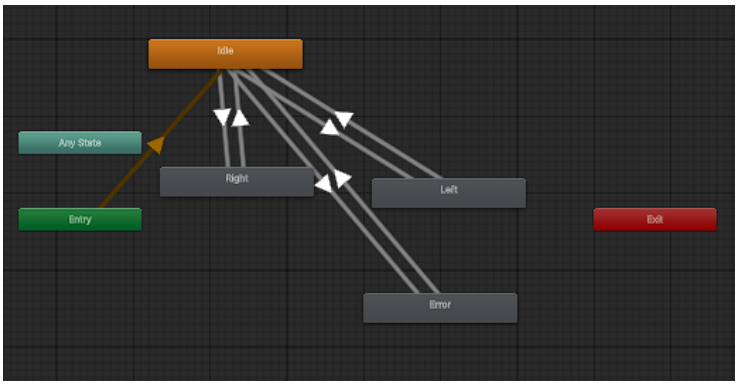


Figura 5-5. Máquina de estados del Animator (Idle–Right–Left–Error).

En la Figura 5-5 se muestra la estructura del *Animator Controller* implementada como una máquina de estados finita, donde Idle actúa como estado base y se habilitan transiciones hacia *Right, Left* y *Error* según el comando leído desde el CSV. Este diseño garantiza una lógica de ejecución simple, trazable y modular: cada ensayo se reduce a una activación de estado, y el retorno a Idle permite preparar el avatar para el siguiente comando.

5.7 Alcances y limitaciones de los resultados

Los resultados obtenidos permiten caracterizar con mayor precisión el desempeño del sistema en un escenario BCI–fNIRS binario (LMI vs. RMI) bajo evaluación por sujeto y con un esquema explícito de rechazo basado en confianza. En esta sección se sintetizan los principales alcances y limitaciones metodológicas y experimentales del estudio, considerando tanto las métricas de clasificación como los análisis de calibración y de riesgo–cobertura previamente discutidos [56]–[60].

5.7.1 Alcances

a) ***Pipeline completo y reproducible a escala inter-sujeto.*** Se implementó un flujo homogéneo de preprocesamiento, segmentación, extracción/selección de características, entrenamiento y evaluación en los 28 sujetos, lo cual es relevante en fNIRS por la alta variabilidad inter-individual. La evaluación por sujeto evita mezclar distribuciones fisiológicas heterogéneas, alineándose con buenas prácticas en BCI [59].

b) ***Evidencia de señal discriminativa en fNIRS motor para LMI vs. RMI.*** Se observaron sujetos con desempeño superior al azar aun sin rechazo, lo que sugiere información lateralizada en señales motoras $\Delta HbO/\Delta HbR$ consistente con la literatura y con la necesidad de considerar ventanas que capturen la dinámica de la HRF [30], [31].

c) ***Selección basada en incertidumbre y no solo en exactitud: salida más operativa.*** El sistema integra calibración/ECE, curvas riesgo–cobertura y trade-off (compromiso) exactitud–rechazo, orientando la decisión a "decidir cuando es confiable", más coherente con una BCI práctica que una optimización exclusiva de exactitud [56], [58].

d) ***Mejora condicional al aceptar: beneficio observable del rechazo.*** Globalmente, la exactitud en aceptados supera al modo sin rechazo, indicando que la abstención filtra parcialmente ensayos ambiguos y mejora la calidad promedio de la salida [58].

e) ***Arquitectura ensamblada robusta y adaptable.*** El uso de ensamblados y un meta-modelo permite combinar inductivos diferentes (lineales/no lineales, sesgo-varianza) y redistribuir pesos por sujeto. La estabilidad observada en la frecuencia de expertos conservados sugiere que el sistema aprovecha complementariedad entre modelos, lo cual es especialmente útil en regímenes de datos pequeños y ruido fisiológico alto, típicos en BCI [59].

f) ***Prototipo demostrativo integrable a un entorno de interacción.*** La integración permite visualizar el comportamiento del sistema y establece una base para migrar a escenarios online, donde el rechazo puede funcionar como modo seguro ante baja confianza.

5.7.2 Limitaciones

a) ***Tamaño muestral por sujeto y alta varianza por partición.*** Aunque el conjunto global incluye 28 sujetos, a nivel individual el número de ensayos es limitado (≈60), y cada *hold-out* contiene muy pocas muestras. Esto induce alta sensibilidad a la partición entrenamiento–prueba y explica que la exactitud media sobre múltiples *splits* pueda acercarse al azar aún cuando existan *splits* con desempeño alto. Este efecto es esperado en "*small-N / high-variance regimes*" y justifica reportar medias y dispersión, no solo el mejor resultado [59].

b) ***Riesgo de optimismo al seleccionar por "mejor test".*** Elegir el *split* por máxima exactitud en *hold-out* puede inflar desempeño (sesgo por selección); aunque la estrategia adoptada prioriza fiabilidad, la amenaza persiste si no se separan estrictamente selección y reporte [59].

c) ***Calibración probabilística aún imperfecta.*** Los ECE por sujeto muestran heterogeneidad y valores moderados en global, lo que indica que la probabilidad predicha no siempre coincide con la frecuencia empírica de acierto. Esto limita la interpretabilidad de la confianza como "probabilidad real" y reduce la eficacia del rechazo, si existen errores con confianza alta, el riesgo no disminuye de forma pronunciada al reducir cobertura (curvas riesgo–cobertura relativamente planas) [62]-[70]. En otras palabras, el

rechazo ayuda, pero su beneficio está acotado por la calidad de la incertidumbre.

d) ***Rechazo asimétrico por clase y potencial impacto en equidad de decisión.*** En el análisis global se observó que la tasa de rechazo puede diferir entre clases, lo que sugiere que el sistema percibe una condición como más ambigua o menos separable. Esto puede ser consecuencia de lateralización variable, canales con distinta calidad, o confusores fisiológicos. Metodológicamente, implica que no basta reportar rechazo global: debe controlarse el rechazo por clase para evitar sesgos operativos [58], [60].
e) ***Dependencia de ventanas/canales y riesgo de inestabilidad fisiológica.*** La selección data-driven de ventanas y canales es poderosa para mejorar desempeño por sujeto, pero también puede volverse sensible a ruido residual y a variaciones sesión a sesión. En fNIRS es frecuente que el patrón hemodinámico cambie por factores no neuronales (estado fisiológico, acoplamiento óptico, movimiento subclínico), lo que puede degradar la transferencia temporal. Por ello, los resultados deben interpretarse como válidos para el dataset bajo las condiciones actuales, y no necesariamente como garantía de estabilidad longitudinal sin adaptación [30], [59].
f) ***Coste computacional y exploración limitada del espacio de hiperparámetros.*** El entrenamiento con validación cruzada anidada, selección de canales/ventanas y ensamblados implica un costo alto. Esto restringe la exploración exhaustiva de hiperparámetros y puede impedir encontrar configuraciones óptimas globales. En consecuencia, el desempeño logrado debe entenderse como una solución robusta y razonable, más que como un límite superior del problema.
g) ***Paradigma restringido y evaluación offline.*** El estudio se limita a dos clases motoras (LMI vs. RMI) y se evalúa offline. En una BCI real, se requiere considerar dinámicas online: deriva temporal, retroalimentación, adaptación del usuario y latencia. El rechazo puede ser beneficioso en línea, pero debe validarse con criterios de utilidad operativa (tiempo hasta comando válido, tasa de comandos erróneos, fatiga) que no se capturan completamente en offline [58].

5.7.3 Limitaciones

En resumen, los resultados muestran que el sistema es capaz de extraer información discriminativa de fNIRS motor y que la incorporación de rechazo aporta una capa operativa orientada a decisiones confiables. Sin embargo, la generalización está condicionada por el régimen de datos pequeños por sujeto, la variabilidad por partición y una calibración todavía imperfecta. En consecuencia, los resultados deben interpretarse como evidencia de factibilidad offline para el paradigma LMI vs. RMI en este dataset y bajo este pipeline. No se pretende afirmar desempeño en escenarios online, multiclase o con transferencia temporal entre sesiones, los cuales requieren validaciones adicionales no incluidas en el alcance de este trabajo.

6. Conclusiones Generales

En este trabajo de grado se diseñó, implementó y evaluó una interfaz cerebro–computadora (BCI) unimodal basada en señales fNIRS para discriminar imaginación motora de mano izquierda vs. mano derecha (LMI vs. RMI). Las conclusiones integran los hallazgos metodológicos y experimentales, confrontan el cumplimiento de los objetivos planteados y resaltan los aportes en procesamiento digital de señales, aprendizaje supervisado y evaluación con incertidumbre bajo un enfoque reproducible.

Se implementó una BCI fNIRS para discriminar LMI vs. RMI y traducir la decisión a comandos discretos reproducibles en Unity. El pipeline se ejecutó de forma homogénea sobre 28 sujetos y obtuvo un desempeño promedio de 70.5% en clasificación forzada. Al incorporar calibración y rechazo, la exactitud sobre predicciones aceptadas aumentó (≈75% con un porcentaje de rechazo del 12.2%), evidenciando utilidad operativa cuando se prioriza confiabilidad sobre cobertura.

Respecto al objetivo específico 1, se seleccionó y empleó una base de datos fNIRS de acceso libre obtenida en un entorno académico y se garantizó un tratamiento homogéneo mediante un pipeline único. El cumplimiento se verifica en la inclusión efectiva de los 28 sujetos procesados bajo las mismas etapas y criterios, permitiendo comparar desempeño sin variaciones metodológicas entre individuos.

En el objetivo específico 2, se desarrollaron técnicas de preprocesamiento para reducir artefactos y ruido: corrección explícita por identificación temporal e interpolación de segmentos contaminados, filtrado pasa–banda hemodinámico y segmentación fija alrededor de marcadores para alinear ensayos. Adicionalmente, la conversión MBLL transformó intensidades ópticas en variables fisiológicamente interpretables ($\Delta HbO/\Delta HbR$). El cumplimiento se evidencia en la generación de señales depuradas y comparables por ensayo/sujeto, sobre las cuales fue posible extraer descriptores y entrenar modelos con resultados reproducibles.

Para el objetivo específico 3, se extrajeron características temporales interpretables y se demostró que la separabilidad depende críticamente de la fase de la respuesta hemodinámica; por ello se empleó un catálogo de ventanas (ascenso, pico/meseta y retorno) con selección data-driven anidada por sujeto. Como resultado verificable, se evidenció que la elección de ventana y canales motores impacta de forma decisiva el rendimiento y que la selección anidada reduce conclusiones dependientes de una sola partición, capturando dinámicas tardías potencialmente más discriminativas.

En el objetivo específico 4, se implementó un algoritmo de clasificación basado en modelos superficiales y no lineales integrados mediante un ensamble con *stacking* (p. ej*., SVM, Random Forest, XGBoost, MLP* y LDA-*Balanced-Bagging*), con poda de expertos y meta-modelo. Su cumplimiento se verifica en un modelo final

operativo con desempeño cuantificado y mejora bajo decisión con incertidumbre, respaldando la robustez práctica del enfoque.

En cuanto al objetivo específico 5, el desempeño se evaluó con un marco riguroso de validación cruzada anidada y métricas clásicas y modernas. La validación anidada separó selección/ajuste de evaluación externa, evitando fuga de información en un régimen de muestras pequeñas. Además de exactitud, se analizaron métricas de fiabilidad (ECE) y curvas riesgo–cobertura para caracterizar el comportamiento del modelo cuando decide “solo si es confiable”, delimitando beneficios y limitaciones del rechazo bajo calibración imperfecta.

Los aportes más relevantes se resumen en: (i) un pipeline de DSP orientado a la HRF con variables interpretables (*ΔHbO, ΔHbR* y *ΔHbO−ΔHbR*); (ii) un esquema de aprendizaje con ensambles y selección anidada para reducir sobreajuste por sujeto; y (iii) evaluación con incertidumbre mediante calibración y opción de rechazo. La reducción de dimensionalidad combina un filtro por separabilidad (r^2 firmado), restricciones de simetría hemisférica y selección embebida, fortaleciendo explicabilidad y reproducibilidad.

En conjunto, el sistema demuestra una trazabilidad completa desde señales hemodinámicas hasta comandos discretos reproducidos en un entorno virtual. Más que un resultado final para uso clínico, el aporte principal es metodológico: un flujo reproducible para MI-fNIRS con control de sesgos, interpretabilidad y mecanismos explícitos de confiabilidad.

Persisten limitaciones propias de MI-fNIRS: artefactos y contribuciones no neuronales, bajo SNR, alta variabilidad inter-sujeto y riesgo de sobreajuste en “pocas muestras–muchas variables”, además de calibración imperfecta. Estas se mitigaron con corrección de artefactos, filtrado hemodinámico, selección anidada de ventana/canales y evaluación riesgo–cobertura. El resultado es un sistema funcional (offline) y modular, con scripts reutilizables que facilitan replicación y futuras extensiones controladas.

7. Trabajos Futuros

Como trabajo futuro, se proponen extensiones puntuales para aumentar robustez y transferibilidad sin cambiar el alcance principal (MI-fNIRS por sujeto).

7.1 Preprocesamiento y modelado hemodinámico

a) Mejorar la corrección de artefactos incorporando métodos automáticos de detección (p. ej., por energía/derivada) y corrección (p. ej., wavelets o spline), comparándolos contra la interpolación basada en marcas del dataset.

b) Incorporar modelado explícito de la respuesta hemodinámica (p. ej., GLM o plantillas HRF) para comparar su efecto sobre las características temporales actuales, manteniendo validación anidada por sujeto.

7.2 Paradigma experimental, ampliación de clases y uso de aprendizaje profundo

a) Evaluar la extensión a más clases dentro de MI (p. ej., incluir estado idle) conservando el mecanismo de rechazo para manejar ambigüedad y cobertura.

b) Adquirir datos locales con el mismo montaje para validar el pipeline fuera de bases de datos abiertas y analizar transferencia entre sesiones y días.

c) Se sugiere evaluar arquitecturas como CNN 1D, LSTM/GRU o transformers sobre series fNIRS (ΔHbO/ΔHbR), comparando rendimiento, estabilidad y costo computacional frente al enfoque actual.

7.3 Implementación en tiempo real y evaluación con usuarios

a) Implementar una versión en línea (*streaming*) como extensión controlada, reutilizando el pipeline y validando latencia, calibración y comportamiento del rechazo bajo restricciones de tiempo real.

b) Realizar pruebas con usuarios para evaluar usabilidad del control por comandos discretos, incluyendo métricas de carga cognitiva y preferencia de umbrales de cobertura.

7.4 Optimización computacional y despliegue

a) Reducir el costo computacional (búsqueda de hiperparámetros y selección) mediante estrategias más eficientes (p. ej., búsqueda bayesiana o early-stopping en ensambles).

b) Empaquetar el código como repositorio reproducible (scripts, configuración y guías) para facilitar replicación y extensión por futuros estudiantes.

8. Referencias Bibliográficas

[1] World Health Organization, "World Report on Disability," 2011. [Online]. Available: https://cdn.who.int/media/docs/default-source/documents/disability/world-report-on-disability-factsheet.pdf?sfvrsn=2a22ed1e_2.

[2] DANE, "Estado actual de la medición de la discapacidad en Colombia," 2022. [Online]. Available: https://www.dane.gov.co/files/investigaciones/notas-estadisticas/abr_2022_nota_estadistica_Estado%20actual_de_la_medici%C3%B3n_de_discapacidad_en%20Colombia.pdf.

[3] J. Shin, A. von Lühmann, B. Blankertz, D.-W. Kim, H.-J. Hwang y K.-R. Müller, "Open Access Dataset for EEG+NIRS Single-Trial Classification," IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2017 (dataset y documentación disponible en la TU Berlin).

[4] N. Naseer y K.-S. Hong, "fNIRS-based brain–computer interfaces: a review," Frontiers in Human Neuroscience, vol. 9, p. 3, 2015.

[5] K. Paulmurugan, V. Vijayaragavan, S. Ghosh, P. Padmanabhan, and B. Gulyás, "Brain–computer interfacing using functional near-infrared spectroscopy (fNIRS)," Biosensors, vol. 11, no. 10, 389, 2021, doi: 10.3390/bios11100389.

[6] S. Brandl y B. Blankertz, "Motor Imagery Under Distraction—An Open Access BCI Dataset," Frontiers in Neuroscience, vol. 14, 566147, 2020 (usa coeficientes r^2 firmados para identificar canales discriminativos en MI).

[7] Y. Li, H. Fang, W. Liu, C. Cheng, H. Chen, "Enhancing Subject-Independent Accuracy in fNIRS-based Brain-Computer Interfaces with Optimized Channel Selection," arXiv preprint arXiv:2502.18719, 2025 (selección de canales basada en correlación para fNIRS-BCI).

[8] A. Real-world fNIRS BCI study, "Real-World fNIRS-Based Brain-Computer Interfaces," 2025 (clasificación de estados tarea/descanso en un entorno de simulación realista con distintas estrategias de extracción de características y selección de canales).

[9] Q. Zhu et al., "Multi-domain Feature Fusion and Stacking Ensemble Classification for Motor Imagery EEG," Brain Sciences, 2025 (framework de ensemble con stacking para MI-EEG).

[10] N. Ahmadi et al., "Enhancing Motor Imagery Electroencephalography Classification with Correlation-Optimized Weighted Stacking Ensemble," Electronics, vol. 13, 2024 (stacking ponderado para mejorar la clasificación MI).

[11] S. Parui et al., "Brain to UAV Communication Model Using Stacked Ensemble CSP Algorithm Based on Motor Imagery EEG Signal," en Proc. IEEE ICC, 2022/2023.

[12] R. Milanés-Hermosilla et al., "Robust Motor Imagery Tasks Classification Using Adaptive Thresholding with a Reject Option," Sensors, 2023 (umbral adaptativo y opción de rechazo en MI-BCI).

[13] D. Chetkin et al., "Bayesian Opportunities for Brain–Computer Interfaces: Improving Reliability via Uncertainty-Aware Decoding," Frontiers in Human Neuroscience, 2023 (uso de incertidumbre para rechazo de decisiones poco fiables).

[14] J. Renardi et al., "Improving P300 Speller Performance with Uncertainty Quantification and Trial Rejection," IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2024.

[15] F. Škola et al., "Progressive Training for Motor Imagery Brain–Computer Interfaces Using Videogames and Virtual Reality," IEEE Trans. Neural Systems and Rehabilitation Engineering, 2019 (entrenamiento de MI en entornos de realidad virtual).

[16] P. Prapas et al., "Mind the Move: Developing a Brain-Computer Interface for Real-Time Interaction with a Virtual Reality Game," 2023 (control de un juego en VR mediante BCI, implementado en Unity).

[17] J. Benitez-Andonegui et al., "Exploring Neurofeedback with Augmented Reality Headsets Using fNIRS," en Proc. CHI, 2020 (uso de fNIRS para adaptar feedback visual en AR mediante Unity).

[18] M. Chen, E. Solovey, G. Smith, "Unity-fNIRS: a Unity project for visualizing fNIRS data in real time," repositorio GitHub (motivación: juego tipo walking simulator informado por BCI)

[19] C. Arboleda et al., "Diseño y construcción de un prototipo de interfaz cerebro-computador," Revista EIA, vol. 6, no. 11, pp. 65–80, 2009.

[20] L. T. L. Loaiza Naranjo, Diseño de una interfaz cerebro computador (BCI) para la interacción con un sistema de rehabilitación de miembro superior, Tesis de pregrado, Universidad Autónoma de Occidente, Colombia, 2016.

[21] E. D. Díaz García and B. R. Vega, Implementación de una interfaz BCI para la detección de intención de movimientos de rehabilitación de miembro superior, Tesis de pregrado, Universidad Autónoma de Bucaramanga, 2023.

[22] J. E. Muñoz Cardona, Clasificación de patrones de imaginación motora en una interfaz cerebro computador basada en EEG, Tesis de pregrado, Universidad Tecnológica de Pereira, 2014.
[23] C. Ferrin-Bolaños, H. Loaiza-Correa, J. Pierre-Díaz, and P. Vélez-Ángel, “Evaluación del aporte de la covarianza de las señales electroencefalográficas a las interfaces cerebro-computador de imaginación motora para pacientes con lesiones de médula espinal,” TecnoLógicas, vol. 22, no. 46, pp. 213–231, 2019.
[24] M. S. S. R. Álvarez et al., “Diseño de una interfaz neuronal para personas con discapacidades motoras severas,” Revista Redes de Ingeniería, vol. 8, pp. 101–112, 2017.
[25] S. Giraldo Pérez, Interfaz cerebro-máquina portátil para accionar un brazo robótico, Tesis de pregrado, Universidad EIA, 2017.
[26] C. D. Ferrin-Bolaños et al., “Interfaz cerebro-computador multimodal para procesos de rehabilitación motora,” Revista Ingenierías USBMed, vol. 9, no. 2, pp. 35–51, 2018.
[27] “Optef-BCI: un interfaz cerebro-computadora híbrido basado en optimización con EEG y fNIRS,” VirtualPro, art. de divulgación, consultado 2025.
[28] M. Ferrari and V. Quaresima, “A brief review on the history of human functional near-infrared spectroscopy (fNIRS) development and fields of application,” NeuroImage, vol. 63, no. 2, pp. 921–935, 2012, doi: 10.1016/j.neuroimage.2012.03.049.
[29] I. Tachtsidis and F. Scholkmann, “False positives and false negatives in functional near-infrared spectroscopy: issues, challenges, and the way forward,” Neurophotonics, vol. 3, no. 3, 031405, 2016, doi: 10.1117/1.NPh.3.3.031405.
[30] F. Scholkmann et al., “A review on continuous wave functional near-infrared spectroscopy and imaging instrumentation and methodology,” NeuroImage, vol. 85, pp. 6–27, 2014.
[31] F. Naseer and K.-S. Hong, “Classification of functional near-infrared spectroscopy signals corresponding to the right- and left-wrist motor imagery for development of a brain–computer interface,” Neuroscience Letters, vol. 553, pp. 84–89, 2013.
[32] S. Bak et al., “Open-access fNIRS dataset for classification of unilateral hand movements,” Electronics, vol. 8, no. 12, p. 1489, 2019.
[33] J. Shin, A. von Lühmann, B. Blankertz, D.-W. Kim, H.-J. Hwang, and K.-R. Müller, “Open Access Dataset for EEG+NIRS Single-Trial Classification,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 10, pp. 1735–1745, 2017, doi: 10.1109/TNSRE.2016.2628057.
[34] J. Shin, J. Kwon, and C.-H. Im, “A ternary hybrid EEG–NIRS brain–computer interface based on motor imagery and mental arithmetic,” Journal of Neural Engineering, vol. 15, no. 3, p. 036009, 2018.
[35] J. Kwon, J. Shin, and C.-H. Im, “Performance evaluation of multi-class hybrid EEG–fNIRS brain–computer interfaces for mental task classification,” PLoS ONE, vol. 15, no. 4, e0231853, 2020.
[36] T. Ma et al., “CNN-based classification of functional near-infrared spectroscopy signals in motor imagery tasks,” Biomedical Signal Processing and Control, vol. 63, p. 102178, 2021.
[37] M. Baghaeifar et al., “Effective channel selection for fNIRS-based motor imagery brain–computer interfaces,” Biomedical Signal Processing and Control, vol. 89, p. 105659, 2025.
[38] V. Ortega-Martínez et al., “Closed-loop motor imagery fNIRS-based brain–computer interface,” in Biophotonics Congress: Biomedical Optics Congress (Translational, Microscopy, OCT, OTS, BRAIN), Optical Society of America, 2022.
[39] F. Lotte et al., “A review of classification algorithms for EEG-based brain–computer interfaces: a 10-year update,” Journal of Neural Engineering, vol. 15, no. 3, p. 031005, 2018.
[40] R. Dhiman et al., “Machine learning techniques for electroencephalogram-based brain–computer interfaces: A review,” Biomedical Signal Processing and Control, vol. 80, p. 104276, 2023.
[41] R. J. Cooper, J. Selb, L. Gagnon, D. Phillip, H. W. Schytz, H. K. Iversen, M. Ashina, and D. A. Boas, “A systematic comparison of motion artifact correction techniques for functional near-infrared spectroscopy,” Frontiers in Neuroscience, vol. 6, art. 147, 2012, doi: 10.3389/fnins.2012.00147.
[42] M. A. Yücel, J. Selb, D. A. Boas, S. S. Cash, and R. J. Cooper, “Reducing motion artifacts for long-term clinical NIRS monitoring using collodion-fixed prism-based optical fibers,” NeuroImage, vol. 85, Pt 1, pp. 192–201, 2014, doi: 10.1016/j.neuroimage.2013.06.054.
[43] The MNE-Python contributors, “Preprocessing functional near-infrared spectroscopy (fNIRS) data,” MNE-Python Documentation (v1.11.0), Disponible en: https://mne.tools/stable/auto_tutorials/preprocessing/70_fnirs_processing.html (consultado: 4-nov-2025)..
[44] F. C. Robertson, T. S. Douglas, and E. M. Meintjes, “Motion artifact removal for functional near infrared spectroscopy: a comparison of methods, ”IEEE Transactions on Biomedical Engineering, vol. 57, no. 6, pp. 1377–1387, 2010, doi: 10.1109/TBME.2009.2038667.
[45] S. Brigadoi et al., “Motion artifacts in functional near-infrared spectroscopy: a comparison of motion correction techniques applied to real cognitive data,” NeuroImage, vol. 85, pp. 181–191, 2014.
[46] F. Scholkmann, S. Spichtig, H. Muehlemann, and M. Wolf, “How to detect and reduce movement artifacts in near-infrared imaging using wavelet filtering,” Journal of Biomedical Optics, vol. 15, no. 6, 066014, 2010.

[47] F. Gustafsson, “Determining the initial states in forward-backward filtering,” IEEE Transactions on Signal Processing, vol. 44, no. 4, pp. 988–992, 1996.
[48] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 3rd ed., Pearson, 2010.
[49] D. H. Wolpert, “Stacked generalization,” Neural Networks, vol. 5, no. 2, pp. 241–259, 1992.
[50] L. Breiman, “Bagging predictors,” Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.
[51] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[52] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. ACM SIGKDD, 2016, pp. 785–794.
[53] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
[54] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” Machine Learning, vol. 46, pp. 389–422, 2002. (RFE clásico)
[55] M. B. Kursa and W. R. Rudnicki, “Feature Selection with the Boruta Package,” Journal of Statistical Software, vol. 36, no. 11, 2010. (Boruta)
[56] J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in Advances in Large Margin Classifiers, MIT Press, 1999. (Platt scaling / sigmoid calibration)
[57] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in Proc. ACM SIGKDD, 2002.
[58] C. K. Chow, “On optimum recognition error and reject tradeoff,” IEEE Transactions on Systems Science and Cybernetics, vol. SSC-6, no. 1, pp. 41–46, 1970.
[59] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[60] G. Lemaître, F. Nogueira, and C. K. Aridas, “Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning,” Journal of Machine Learning Research, vol. 18, no. 17, pp. 1–5, 2017.
[61] A. Gramfort et al., “MEG and EEG data analysis with MNE-Python,” Frontiers in Neuroscience, vol. 7, art. 267, 2013, doi: 10.3389/fnins.2013.00267.
[62] C. Guo, G. Pleiss, Y. Sun y K. Q. Weinberger, “On Calibration of Modern Neural Networks,” en Proc. 34th Int. Conf. Machine Learning (ICML), PMLR, vol. 70, pp. 1321–1330, 2017.
[63] J. Nixon, M. Dusenberry, G. Jerfel, T. Nguyen, J. Liu, L. Zhang y D. Tran, “Measuring Calibration in Deep Learning,” arXiv:1904.01685, 2019 (rev. 2020), doi: 10.48550/arXiv.1904.01685.
[64] R. Roelofs, N. Cain, J. Shlens y M. C. Mozer, “Mitigating Bias in Calibration Error Estimation,” en Proc. 25th Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR, vol. 151, pp. 4036–4054, 2022.
[65] A. Niculescu-Mizil y R. Caruana, “Predicting Good Probabilities with Supervised Learning,” en Proc. 22nd Int. Conf. Machine Learning (ICML ’05), pp. 625–632, 2005, doi: 10.1145/1102351.1102430.
[66] Y. Geifman y R. El-Yaniv, “Selective Classification for Deep Neural Networks,” en Advances in Neural Information Processing Systems 30 (NeurIPS 2017), pp. 4878–4887, 2017.
[68] M. A. Yücel et al., “Best practices for fNIRS publications,” Neurophotonics, vol. 8, no. 1, 012101, 2021, doi: 10.1117/1.NPh.8.1.012101.
[69] M. Caldwell, F. Scholkmann, U. Wolf, M. Wolf, C. Elwell y I. Tachtsidis, “Modelling confounding effects from extracerebral contamination and systemic factors on functional near-infrared spectroscopy,” NeuroImage, vol. 143, pp. 91–105, Dec. 2016, doi: 10.1016/j.neuroimage.2016.08.058.
[70] I. Tachtsidis y F. Scholkmann, “False positives and false negatives in functional near-infrared spectroscopy: issues, challenges, and the way forward,” Neurophotonics, vol. 3, no. 3, 031405, 2016, doi: 10.1117/1.NPh.3.3.031405.

9. Anexos

Anexo 1. Corresponde al repositorio público en GitHub donde se entrega todo el proyecto: scripts en Python para ejecutar el pipeline fNIRS de MI (conversión MBLL, corrección de artefactos, filtrado, segmentación, extracción/selección de características, entrenamiento/evaluación del ensamble con calibración y rechazo) y los artefactos reproducibles generados (señales procesadas, reportes, modelos entrenados y comandos para Unity). La organización del repositorio separa explícitamente datos/etapas intermedias, entrenamiento/modelos y validación funcional (exportación de comandos), de modo que un lector puede rastrear cada resultado del documento hasta el script y la carpeta que lo produce. **https://github.com/CJMORA10/BCI_fNIRS**

Anexo 2. Señales crudas fNIRS de Imaginación Motora (MI). Se presentan las señales crudas correspondientes al primer segmento de imaginación motora (MI). El banco completo de figuras para todos los sujetos (S01–S28) y para cada segmento se entrega en el CD Anexo, organizado por sujeto y segmento, de acuerdo con la estructura descrita en el Anexo 1.

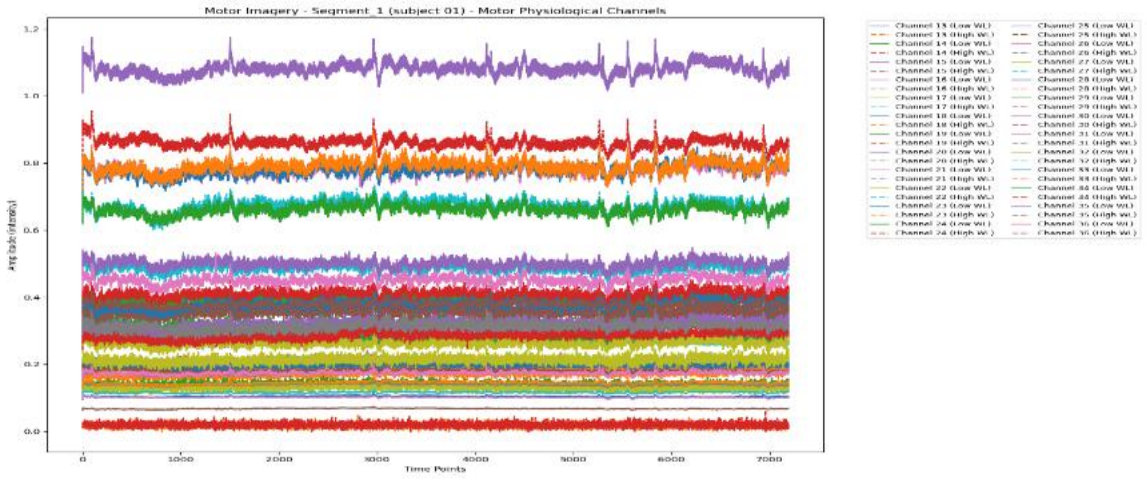


Figura A2. 1. Señales crudas primer segmento – sujeto 1.

Anexo 3. Señales fusionadas de tres sesiones de imaginación motora convertidas mediante MBLL (ejemplos representativos). Se presenta para el sujeto 1, la señal fNIRS correspondiente a imaginación motora tras la fusión de las tres sesiones disponibles y su conversión a concentraciones relativas mediante la MBLL; el banco completo de figuras para todos los sujetos se entrega en el medio digital descrito en el Anexo 1.

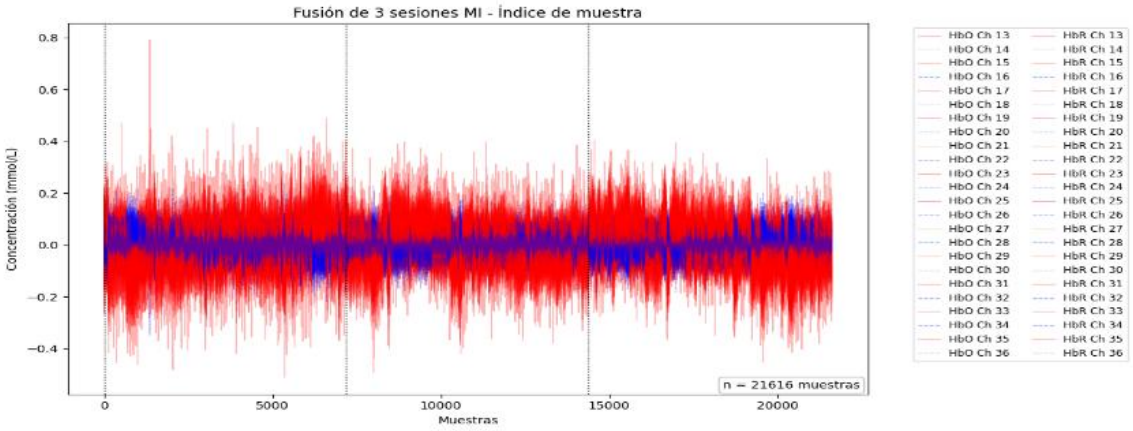


Figura A3. 1. Fusión de tres sesiones MI y conversión MBLL (ΔHbO/ΔHbR) – Sujeto 1.

Anexo 4. Corrección de artefactos en señales fNIRS Se presentan ejemplos representativos del proceso de corrección de artefactos aplicado a las señales fNIRS en el contexto de imaginación motora para el sujeto 1; el banco completo de figuras para todos los sujetos se entrega en el medio digital descrito en el Anexo 1.

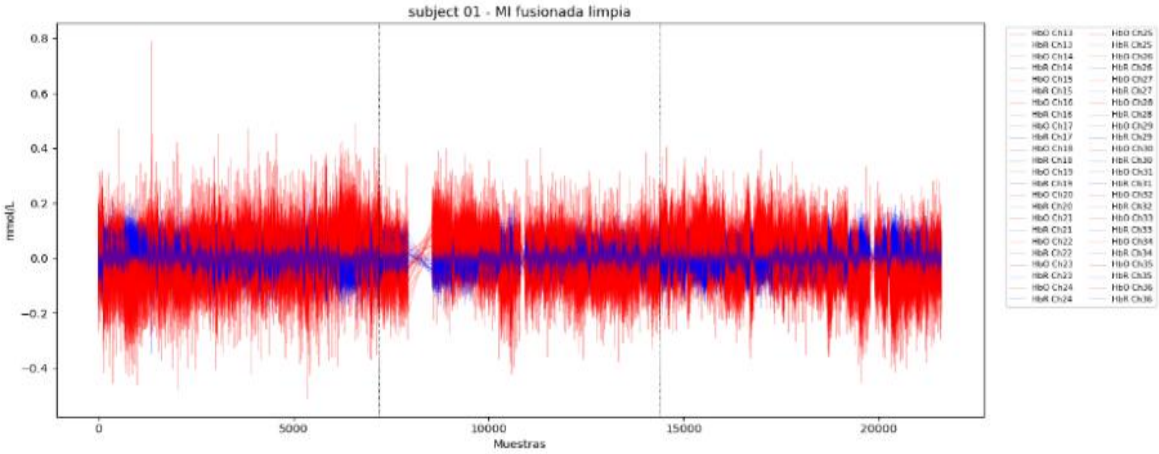


Figura A4. 1. Señal con artefactos eliminados – Sujeto 1.

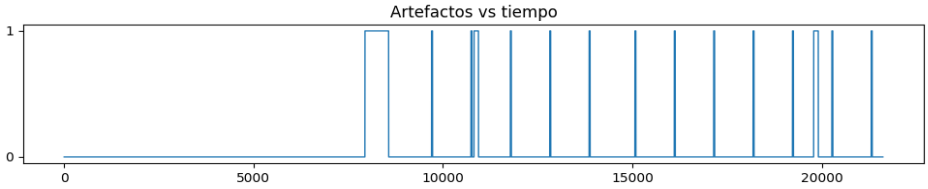


Figura A4. 2. Relación artefactos vs. tiempo – Sujeto 1.

Anexo 5. Señales fNIRS filtradas en banda hemodinámica. Se presenta un ejemplo representativo del efecto del filtrado pasa–banda aplicado a las señales fNIRS del sujeto 1; el banco completo de figuras para todos los sujetos se entrega en el medio digital descrito en el Anexo 1.

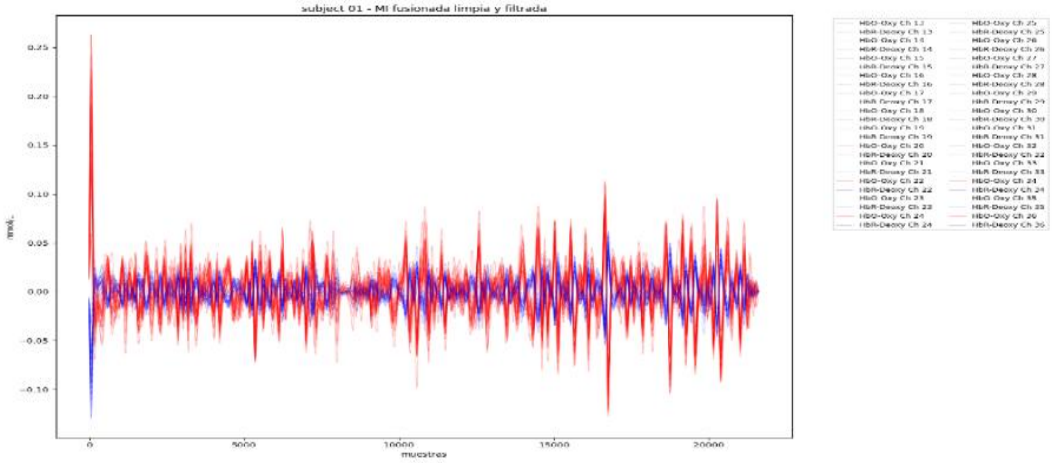


Figura A5. 1. Señal filtrada – Sujeto 1 – Filtro Chebyshev II.

Anexo 6. Ejemplo de segmentación temporal (épocas) para MI y reposo en un sujeto representativo (Sujeto 1) aplicado a las señales fNIRS, El conjunto completo de segmentaciones para todos los sujetos se entrega en el medio digital descrito en el Anexo 1.

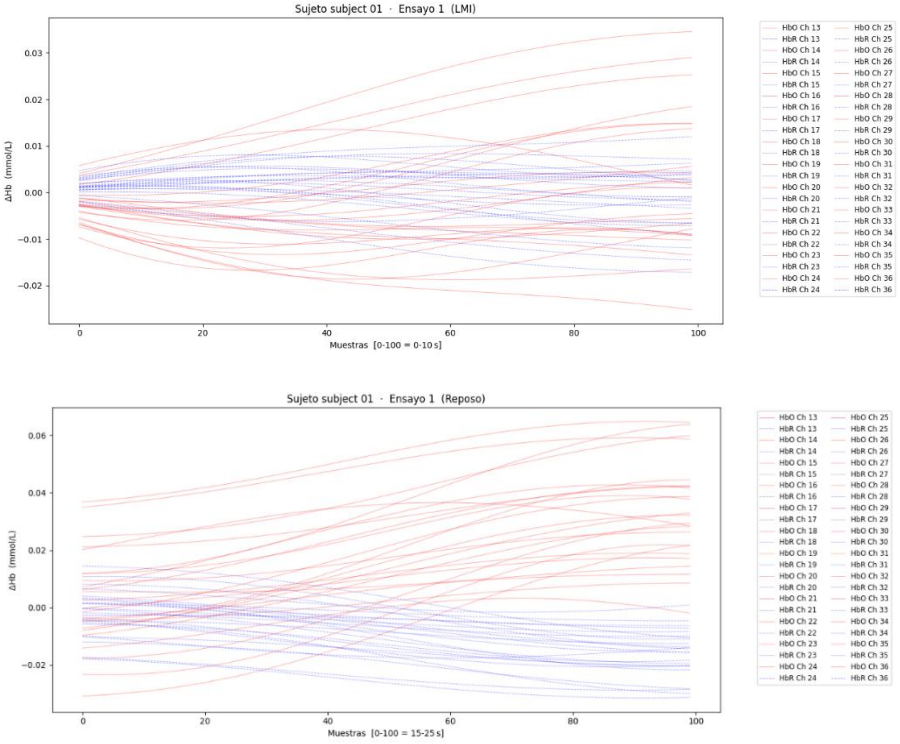


Figura A6. 1. Época segmentada - LMI - Reposo - Sujeto 1.

Anexo 7. Se presenta el ejemplo representativo de la distribución de las características extraídas a partir de las señales fNIRS procesadas del sujeto 1, con el fin de visualizar el grado de separación entre clases (mano izquierda vs. mano derecha). El conjunto completo de figuras para todos los sujetos se entrega en el medio digital descrito en el Anexo 1.

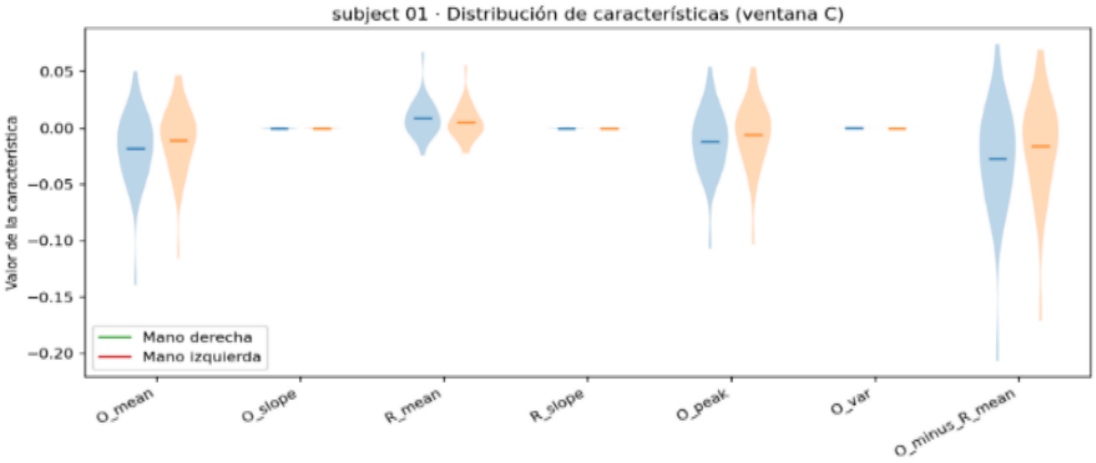


Figura A7. 1. Diagrama de violín de características – Sujeto 1 (ventana C).

Anexo 8. En este anexo se presenta la matriz de confusión correspondiente al escenario sin rechazo (clasificación forzada) del sujeto 1, El conjunto completo de matrices por sujeto (y por partición, cuando aplique) se entrega en el medio digital descrito en el Anexo 1.

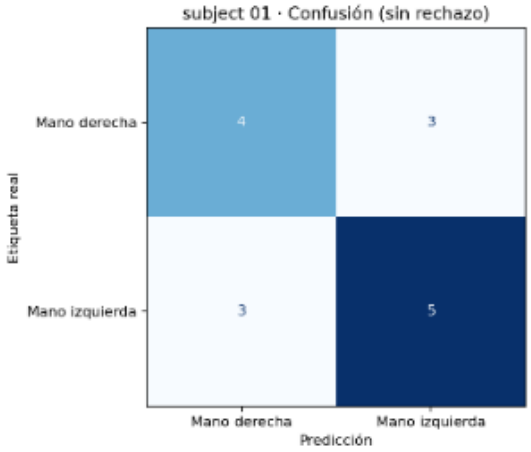


Figura A8. 1. Matriz de confusión sin rechazo – Sujeto 1.

Anexo 9. Paneles de evaluación con rechazo. Este anexo reúne paneles representativos de la evaluación del sistema cuando se habilita el rechazo (abstención) basado en confianza. Para el sujeto1 como demostración, se presenta: matriz de confusión con rechazo, curva riesgo–cobertura (AURC), curva exactitud

vs. tasa de rechazo y gráficos de calibración (ECE), con el fin de complementar la interpretación del desempeño bajo decisión selectiva [56]–[58]. El banco completo por sujeto se entrega en el medio digital descrito en el Anexo 1.

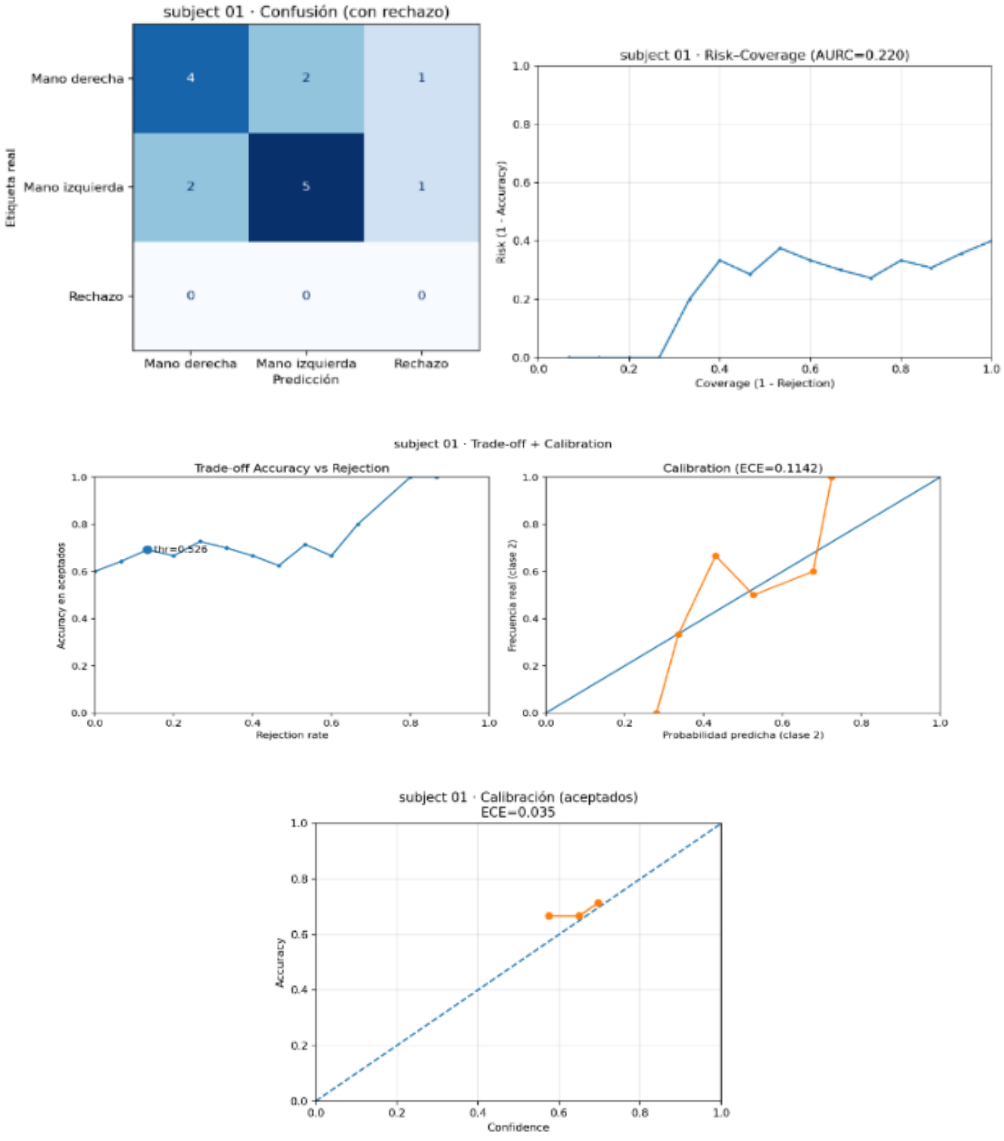


Figura A9. 1. Panel con rechazo – Sujeto 1.