



E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE
TELECOMUNICACIÓN, MENCIÓN EN SISTEMAS DE
TELECOMUNICACIÓN

DISEÑO Y DESARROLLO DE UNA PLATAFORMA DE ENTRENAMIENTO COGNITIVO MEDIANTE UN SISTEMA BRAIN COMPUTER INTERFACE PARA DISPOSITIVOS ANDROID

AUTOR:

Dña. Chiara Franzoni Uriarte

TUTORES:

Dr. D. Roberto Hornero Sánchez
D. Víctor Martínez Cagigal

Valladolid, Junio de 2018

TÍTULO: **Diseño y desarrollo de una plataforma de entrenamiento cognitivo mediante un sistema Brain Computer Interface para dispositivos Android**

AUTOR: **Dña. Chiara Franzoni Uriarte**

TUTORES: **Dr. D. Roberto Hornero Sánchez
D. Víctor Martínez Cagigal**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **Dr. D. Roberto Hornero Sánchez**

SECRETARIO: **Dr. Dña. María García Gadañón**

VOCAL: **Dr. D. Jesús Poza Crespo**

SUPLENTE: **Dr. D. Carlos Gómez Peña**

SUPLENTE: **Dr. D. Miguel López-Coronado**

FECHA:

CALIFICACIÓN:

RESUMEN DEL PROYECTO

Desde que Hans Berger descubrió el electroencefalograma (EEG) en 1929, las ondas cerebrales comenzaron a estudiarse con la finalidad de ayudar al diagnóstico de ciertas enfermedades. En la segunda mitad del siglo XX se describió por primera vez un sistema Brain-Computer Interface (BCI) como una forma de comunicación entre el cerebro y el medio sin la intervención de nervios ni músculos. Estos sistemas permiten el control de dispositivos mediante la clasificación de ciertas características de la actividad cerebral, que reflejen la intención del usuario.

La investigación de sistemas BCI se ha centrado principalmente en desarrollar poderosas técnicas de procesamiento de señales y aprendizaje de máquina, para clasificar con precisión la intención del usuario. No obstante, incluso con los mejores algoritmos, el buen manejo de ciertos sistemas BCI depende significativamente de cómo los usuarios puedan modular voluntariamente su actividad neuronal. Aprender a manejar estos sistemas requiere de un repetido entrenamiento cognitivo con retroalimentación (NFT).

El objetivo de este trabajo se centra en diseñar y desarrollar una plataforma BCI de NFT para Android. Tras realizar una revisión del estado del arte, se concluye que la mejor manera de llevar a cabo el entrenamiento es mediante la imaginación motora (MI), que se basa en los ritmos sensoriomotores (SMR) como señal de control. La finalidad del NFT es mejorar ciertas capacidades cognitivas de los usuarios.

En la arquitectura de la aplicación se distinguen dos partes fundamentales: procesado y feedback. El procesado de señal se realiza en un ordenador mediante BCI2000, que traduce el EEG del usuario en señales de control de la aplicación. Posteriormente, se envían por Bluetooth al dispositivo móvil donde se presenta el feedback al usuario.

La aplicación fue evaluada por cuatro sujetos sanos en tres sesiones de entrenamiento. Los resultados obtenidos fueron bastante prometedores ya que los cuatro sujetos presentaron diferencias significativas en las potencias relativas de las bandas de frecuencia de los SMR. Tras el entrenamiento, los sujetos de estudio completaron un cuestionario de satisfacción que permitió conocer su opinión del sistema implementado y realimentar el proyecto con sus sugerencias.

PALABRAS CLAVE

Brain Computer Interface, imaginación motora, electroencefalografía, ritmos sensoriomotores, entrenamiento con retroalimentación, señal de control, desarrollo cognitivo.

ABSTRACT

Since Hans Berger developed the electroencephalogram (EEG) in 1929, brain signals have been used to help the diagnosis of certain diseases. In the second half of the twentieth century, Brain-Computer Interface (BCI) systems were firstly described as a way of communication between the brain and the environment without the intervention of nerves and muscles. These kinds of systems allow users to control devices by classifying certain brain activity features, which reflect the user's intentions.

BCI research has been heavily focused on developing powerful signal processing and machine learning techniques to improve the identification of user's intentions. However, even with the best algorithms, the control of certain BCI systems still depends on the users' ability to voluntarily modulate their neural activity. Learning to control these systems requires the application of repeated cognitive feedback training (NFT).

The objective of this study is to design and develop an NFT BCI platform for Android devices. After an exhaustive state of the art review, the decision of using motor imagery (MI) for carrying out the training was reached. MI uses sensorimotor rhythms (SMR) as control signals. The purpose of NFT is to improve certain cognitive abilities of the users.

Regarding the architecture of the application, two fundamental parts are distinguished: processing and feedback. Signal processing is performed on a computer using BCI2000, which translates the EEG into control signals. Then, these signals are sent to the mobile device via Bluetooth, where feedback is presented to the user.

The application was tested with four healthy subjects, who performed three training sessions. Results were promising, since the four subjects presented significant differences in the relative powers of the frequency bands of SMR before and after the training. After this NFT training, subjects completed a satisfaction questionnaire. The questions were focused on knowing their personal opinions about the application and registering possible suggestions.

KEYWORDS

Brain Computer Interface, motor imagery, electroencephalography, sensorimotor rhythm, neurofeedback training, control signal, cognitive development

AGRADECIMIENTOS

En primer lugar, quiero expresar mi gratitud a mis tutores Roberto Hornero Sánchez y Víctor Martínez Cagigal por haberme brindado la posibilidad de realizar este Trabajo Fin de Grado y por su paciencia y tiempo dedicado a guiarme en la realización del mismo.

Así mismo, agradecer al Grupo de Ingeniería Biomédica de la Universidad de Valladolid por su cálida acogida y amabilidad con las que te hacen sentir uno más del grupo desde el primer día hasta el último; haciendo especial hincapié en Víctor Martínez y Eduardo Santamaría por su constante ayuda intentando resolver todas mis dudas a lo largo de la elaboración de este trabajo.

Y por último, me gustaría dar las gracias a mi familia y amigos por darme siempre su apoyo incondicional durante todos estos años. A mis padres por la educación que me han dado y el cariño y ayuda sin los cuales no sería quien soy. Y a José por tener tanta paciencia y sacarme siempre una sonrisa. Sin ellos no habría sido capaz de llegar hasta aquí.

ÍNDICE GENERAL

<i>Índice general</i>	xi
<i>Índice de figuras</i>	xv
<i>Índice de tablas</i>	xix
<i>Índice de acrónimos</i>	xxi
1. INTRODUCCIÓN	1
1.1. Ingeniería Biomédica	1
1.2. Señales Biomédicas	2
1.3. Electroencefalograma (EEG)	4
1.4. Brain Computer Interface (BCI)	7
1.5. NeuroFeedback Training	8
1.6. Objetivos del Trabajo Fin de Grado	9
1.7. Estructura del Trabajo Fin de Grado	10
2. BRAIN COMPUTER INTERFACE (BCI)	13
2.1. Introducción	13
2.2. Métodos de registro de la actividad cerebral	15
2.2.1. Métodos invasivos	15
2.2.2. Métodos no invasivos	15
2.3. Tipos de señales de control	17
2.3.1. Ritmos Sensoriomotores (ritmos μ y β)	17
2.3.2. Potenciales corticales lentos (SCP)	20
2.3.3. Potenciales evocados visuales (VEP)	21
2.3.4. Potenciales evocados P300	22
2.4. Procesado del EEG	23
2.4.1. Adquisición y pre-procesado de la señal	23
2.4.2. Extracción de características	25
2.4.3. Clasificación de características	28
2.5. Aplicaciones de los sistemas BCI	29
2.5.1. Comunicación	30
2.5.2. Restauración motora	30

2.5.3.	<i>Control del entorno</i>	31
2.5.4.	<i>Locomoción</i>	32
2.5.5.	<i>Entretenimiento</i>	33
2.5.6.	<i>Otras aplicaciones</i>	33
2.6.	<i>Neurofeedback training basados en BCI</i>	34
2.6.1.	<i>Gomez-Pilar et al - Neurofeedback training with a motor imagery-based BCI: neurocognitive improvements and EEG changes in the elderly [37]</i>	34
2.6.2.	<i>Hwang et al. - Neurofeedback-based motor imagery training for Brain-Computer Interface (BCI) [43].</i>	35
2.6.3.	<i>Shindo et al. - Effects of Neurofeedback training with an electroencephalogram based Brain-Computer Interface for hand paralysis in patients with chronic stroke: a preliminary case series study [17].</i>	38
2.6.4.	<i>Gevensleben et al. - Is neurofeedback an efficacious treatment for ADHD? A randomised controlled clinical trial [18].</i>	39
2.6.5.	<i>Comparativa de los diferentes estudios</i>	40
3.	ANÁLISIS DE LA SEÑAL Y RITMOS SENSORIOMOTORES	43
3.1.	<i>Ritmos sensoriomotores (ritmos μ y β)</i>	43
3.2.	<i>Factores que influyen en el control de los SMR</i>	45
3.3.	<i>Procesado de los ritmos sensoriomotores</i>	46
3.3.1.	<i>Adquisición y registro de la señal</i>	46
3.3.2.	<i>Extracción de características</i>	47
3.3.3.	<i>Clasificación de características</i>	58
4.	DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE NFT	65
4.1.	<i>Objetivo de la aplicación</i>	65
4.2.	<i>Arquitectura de la aplicación</i>	65
4.3.	<i>Adquisición del EEG</i>	66
4.4.	<i>Procesado mediante BCI-2000</i>	67
4.4.1.	<i>Sistema de propósito general BCI2000</i>	68
4.4.2.	<i>Configuración de parámetros</i>	69
4.4.3.	<i>Modificación de código fuente</i>	74
4.5.	<i>Aplicación BCI-NeuroAge</i>	75
4.5.1.	<i>Programación en Unity</i>	75
4.5.2.	<i>Programación en Android Studio</i>	78
4.6.	<i>Manual de usuario</i>	80
4.6.1.	<i>Menú principal</i>	80
4.6.2.	<i>Tareas</i>	83
5.	RESULTADOS	87
5.1.	<i>Sujetos y Tareas</i>	87
5.1.1.	<i>Primera sesión</i>	87
5.1.2.	<i>Segunda sesión</i>	88
5.1.3.	<i>Tercera sesión</i>	88

5.2. <i>Análisis Online</i>	89
5.3. <i>Análisis Offline</i>	90
5.3.1. <i>Usuario 01</i>	92
5.3.2. <i>Usuario 02</i>	95
5.3.3. <i>Usuario 03</i>	98
5.3.4. <i>Usuario 04</i>	101
5.4. <i>Cuestionario de satisfacción</i>	104
6. DISCUSIÓN	107
6.1. <i>Análisis Online</i>	107
6.2. <i>Análisis Offline</i>	108
6.3. <i>Cuestionario de satisfacción</i>	109
6.4. <i>Comparación con otros estudios</i>	110
6.5. <i>Limitaciones</i>	111
7. CONCLUSIONES Y LÍNEAS FUTURAS	113
7.1. <i>Conclusiones</i>	113
7.2. <i>Líneas Futuras</i>	116
Bibliografía	119
A. Código Generado en C++ , Java, C# y MatLab	125
A.1. <i>Modificación del bloque CursorTask</i>	125
A.1.1. <i>BluetoothClient</i>	125
A.1.2. <i>CursorFeedbackTask</i>	130
A.2. <i>Código generado en Unity</i>	132
A.3. <i>Código generado en AndroidStudio</i>	140
A.4. <i>Código generado en MatLab</i>	149
B. Cuestionario de Satisfacción	153
C. Pliego de Condiciones	155
D. Presupuesto	165

ÍNDICE DE FIGURAS

1.1. Representación de las principales señales bioeléctricas: (a) electroencefalograma (EEG), (b) electrocardiograma (ECG) y (c) electromiograma (EMG) [3].	4
1.2. Representación de la localización de los electrodos en el Sistema Internacional 10-20 [5].	5
1.3. Representación de los efectos en el EEG producidos por algunos artefactos [7].	6
2.1. Esquema de los sistemas BCI [12].	14
2.2. Localización de los electrodos en el EEG, el ECoG y los electrodos intracorticales [13].	16
2.3. Representación de la amplitud de los ritmos μ y β al realizar una elección entre dos objetivos (arriba y abajo) [16].	18
2.4. Superposición de la relación entre ERD/ERS % de tres bandas de frecuencias diferentes registradas en el electrodo C3 antes y después del movimiento de un dedo de la mano derecha [16].	19
2.5. SCP para mover un cursor hacia un objetivo (por ejemplo, una letra o icono deseado) en la parte inferior (SCP positivo) o superior (SCP negativo) de una pantalla [16].	20
2.6. Espectro de frecuencia típico de una señal EEG adquirida durante la estimulación visual con una frecuencia de parpadeo de 7 Hz. La respuesta VEP puede verse como los picos a 7 Hz y sus armónicos 14 y 21 Hz [17].	21
2.7. Respuestas del EEG tomadas de la zona centroparietal ante las posibles opciones de una matriz que parpadean. Solo la elección deseada por el usuario presenta un P300 [16].	22
2.8. Arquitectura de un sistema BCI.	23
2.9. Señal de EEG de distintos electrodos afectada por artefactos [20].	25
2.10. A) Ubicaciones de electrodos utilizados en el registro de EEG para cada filtro espacial en rojo. B) Banda de paso de cada uno de los métodos donde se muestra la raíz cuadrada del valor cuadrático medio de la señal. C) Topografía de r^2 y la amplitud espectral para cada método utilizado [16].	27
2.11. Aplicación de un método lineal y uno no lineal al mismo grupo de datos a clasificar [21].	29
2.12. Tres ejemplos de Spellers con diferentes señales de control: a) P300 [24], b) SMR con 4 posibilidades [25], c) SCP con elecciones binarias [26].	31

2.13. En la imagen de la derecha se puede ver una neuroprótesis para un sujeto que ha perdido el brazo [29] y en la imagen de la izquierda se utiliza una combinación de BCI y FES como método de rehabilitación [30]. 32

2.14. Matriz para control domótico mediante P300 [31]. 32

2.15. Control de una silla de ruedas mediante EEG [35]. 33

2.16. Videojuegos controlados mediante EEG. 34

2.17. Representación gráfica de las cinco tareas ha realizar en el experimento [37]. 36

2.18. Representación gráfica la secuencia que sigue el paradigma experimental [43]. 37

2.19. Feedback que se le muestra al sujeto durante el entrenamiento [43]. 37

2.20. a) Ortesis motorizada. La imagen de la izquierda es la posición neutral y la de la derecha es la posición de máxima extensión. Normalmente tarda 3s en pasar de una a otra. b) Posición de los electrodos utilizados para el registro del EEG según el sistema internacional 10-20 y 2,5 cm anterior (A), posterior (P), medio (M) y lateral (L) a C3 y C4. c) Feedback visual del cursor moviéndose hacia abajo (imaginación del movimiento) y hacia arriba (resto) [44]. 39

3.1. (A),(B) Distribución topográfica del estadístico r^2 calculada para (A) la imaginación del movimiento de la mano derecha y (B) movimiento de la mano derecha vs. resto para un ancho de banda de 3Hz centrado en 12 Hz. (C) Espectros del voltaje sobre C3 para comparar el descanso (línea punteada) y MI (línea continua). (D) Espectro r^2 para MI vs. descanso [48]. 44

3.2. Representación visual de las divisiones anatómicas de la corteza motora primaria y la corteza somatosensorial primaria [49]. 45

3.3. Representación de cuatro filtrados espaciales para obtener la señal del electrodo C3. De izquierda a derecha: referencia común , CAR, Laplaciano corto y Laplaciano largo. En todos ellos el electrodo principal C3 se representa de color rojo, y los utilizados para el filtro de color verde [16]. 49

3.4. Representación de los dos primeros componentes principales para la nube de datos representados mediante círculos azules [57]. 50

3.5. Efecto del filtrado espacial de cuatro filtros CSP para discernir entre MI de la mano derecha y MI de la mano izquierda [58]. 51

3.6. Ejemplo de aplicación de un filtro espacial CAR. Las señales son espacialmente más específicas y resaltan más la supresión del ritmo beta durante el período indicado por la barra verde cuando se aplica el filtro (B y D) en comparación a cuando no se aplica (A y C) [48]. 52

3.7. Respuesta de diferentes filtros espaciales para los mismos datos. De izquierda a derecha: filtro CAR, filtro Laplaciano y filtro CSP [9]. 53

3.8. Esquema general de un filtro adaptativo [59]. 54

3.9. División del plano tiempo-frecuencia para: a) La STFT con resolución constante. b) La DWT con resolución variable adaptada para señales biomédicas. c) La WP con resolución variable según el caso. 56

3.10. Calculo de la RP en la banda [10, 20] Hz de la señal comprendida entre [0, 50] Hz. 58

3.11. LDA.	60
3.12. Clasificador SVM, los vectores de soporte se representan con un círculo y los datos de cada clase con símbolos diferentes (+, *). [8].	61
3.13. Clasificador mediante ANN binario [61].	62
3.14. Clasificador mediante K-NN.	63
4.1. Esquema de la arquitectura de la aplicación BCI-NeuroAge.	66
4.2. Localización de los electrodos utilizados en la aplicación (rojo), la referencia (azul) y tierra (amarillo).	67
4.3. Estructura del sistema de propósito general BCI2000 [48].	68
4.4. Interfaz gráfica del módulo Operador [48].	69
4.5. Módulo de procesado de señal [48].	70
4.6. FullMatrix del filtro espacial.	71
4.7. FullMatrix del clasificador lineal.	72
4.8. Ejemplo de los componentes que forman un GameObject con función de cámara.	76
4.9. Diagrama de flujo de una escena.	77
4.10. Icono de la app BCI-NeuroAge.	80
4.11. Ventana emergente solicitando el permiso del usuario para activar el Bluetooth del dispositivo.	81
4.12. Menú principal de la aplicación en orientación horizontal y vertical de izquierda a derecha respectivamente.	82
4.13. Ventana donde se muestra la lista de preferencias que se pueden configurar.	83
4.14. Imágenes que muestran el comportamiento de la Tarea 1.	84
4.15. Iconos de los botones que aparecen al finalizar las tareas. El de la derecha es el de recargar la tarea, y el de la izquierda el de salir de ella al menú principal.	84
4.16. Imágenes que muestran el desarrollo de la Tarea 2.	85
4.17. Imágenes que muestran el desarrollo de la Tarea 3.	85
4.18. Imágenes que muestran el desarrollo de la Tarea 4.	86
4.19. Imágenes que muestran el desarrollo de la Tarea 5	86
5.1. Representación gráfica de los trials que componen cada una de las sesiones de entrenamiento cognitivo.	88
5.2. Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario01, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.	92
5.3. Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.	93
5.4. Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.	94
5.5. Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario02, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.	95

5.6. Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.	96
5.7. Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.	97
5.8. Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario03, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.	98
5.9. Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.	99
5.10. Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.	100
5.11. Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario04, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.	101
5.12. Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.	102
5.13. Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.	103

ÍNDICE DE TABLAS

2.1. Comparación de características de los distintos métodos de registro de la actividad cerebral [15].	17
2.2. Comparación de características de las diferentes señales de control de los sistemas BCI [15].	18
2.3. Clasificación de los diferentes métodos de extracción de características según el dominio en el que trabajen, tiempo, espacio, una combinación de ambos o modelos inversivos. . . .	26
2.4. Clasificación de los diferentes métodos de traducción de características.	28
5.1. Resultados obtenidos por cada usuario en la tres sesiones de entrenamiento cognitivo. . .	89
5.2. Media de los resultados obtenidos por cada usuario en la tres sesiones de entrenamiento cognitivo.	90
5.3. Tests estadísticos según las características de los datos.	91
5.4. Tabla que refleja la magnitud del p-valor para cada usuario obtenido mediante el test de los rangos con signo de Wilcoxon, en cada una de las tres bandas de interés. Las diferencias significativas (p -valor $<0,05$) vienen marcadas con *.	104
5.5. Tabla que refleja la media y SD de los resultados del cuestionario.	105

ÍNDICE DE ACRÓNIMOS

ADHD	<i>Attention Deficit Hiperactivity Disorder</i> Trastorno por Déficit de Atención e Hiperactividad
ALS	<i>Amyotrophic Lateral Sclerosis</i> Esclerosis Lateral Amiotrófica
ANN	<i>Artificial Neural Networks</i> Redes Neuronales Artificiales
AR	<i>Autoregressive Models</i> Modelos Autorregresivos
BCI	<i>Brain Computer Interface</i> Interfaz Cerebro-Computadora
CAR	<i>Common Average Reference</i> Referencia de Media Común
CSP	<i>Common Spatial Patterns</i> Patrones Espaciales Comunes
CVA	<i>Cerebrovascular Accident</i> Accidente Cerebrovascular
CWT	<i>Continous Wavelet Transform</i> Transformada Wavelet Continua
DWT	<i>Discrete Wavelet Transform</i> Transformada Wavelet Discreta
EEG	<i>Electroencefalography</i> Electroencefalografía
ECG	<i>Electrocardiography</i> Electrocardiografía
ECoG	<i>Electrocorticography</i> Electrocorticografía
EMG	<i>Electromyography</i> Electromiografía

EOG	<i>Electrooculography</i> Electrooculografía
ERD	<i>Event-Related Desynchronization</i> Desincronización del evento relacionado
ERS	<i>Event-Related Synchronization</i> Sincronización del evento relacionado
FES	<i>Functional Electrical Stimulation</i> Estimulación Eléctrica Funcional
FFT	<i>Fast Fourier Transform</i> Transformada Rápida de Fourier
FLD	<i>Fisher's Linear Discriminant</i> Discriminante Lineal de Fisher
FIR	<i>Finite Impulse Response</i> Respuesta al Impulso Finita
fMRI	<i>Functional Magnetic Resonance Imaging</i> Imagen por Resonancia Magnética Funcional
GKP	<i>Glossokinetic Potential</i> Potencial de Glossokinetic
ICA	<i>Independent Component Analysis</i> Análisis de Componentes Independientes
IIR	<i>Infinite Impulse Response</i> Respuesta al Impulso Infinita
K-NN	<i>K-Nearest Neighbors</i> K Vecinos más Cercanos
LDA	<i>Linear Discriminant Analysis</i> Análisis Discriminante Lineal
LMS	<i>Least Mean Square</i> Mínimos Cuadrados Promediados
MCG	<i>Magnetocardiography</i> Magnetocardiografía
MEG	<i>Magnetoencephalography</i> Magnetoencefalografía
MI	<i>Motor Imagery</i> Imaginación Motora
MLP	<i>Multilayer Perceptron</i> Perceptrón de Múltiples Capas

MSE	<i>Mean Squared Error</i> Error Cuadrático Medio
NFT	<i>NeuroFeedback Training</i> Entrenamiento Neurológico con Realimentación
NIRS	<i>Near-Infrared Spectroscopy</i> Espectroscopia de Infrarrojo Cercano
PCA	<i>Principal Component Analysis</i> Análisis de Componentes Principales
PET	<i>Positron Emission Tomography</i> Tomografía por Emisión de Positrones
PSD	<i>Power Spectral Density</i> Densidad Espectral de Potencia
RLS	<i>Recursive Least Squares</i> Mínimos Cuadrados Recursivos
RP	<i>Relative Power</i> Potencia Relativa
SCP	<i>Slow Cortical Potentials</i> Potenciales Corticales Lentos
SD	<i>Standard Deviation</i> Desviación Estándar
SMR	<i>Sensory-Motor Rhythms</i> Ritmos Sensoriomotores
SNR	<i>Signal-to-Noise Ratio</i> Relación Señal a Ruido
SQUID	<i>Superconducting Quantum Interference Devices</i> Dispositivos Superconductores de Interferencia Cuántica
STFT	<i>Short-Time Fourier Transform</i> Transformada de Fourier de Tiempo Corto
SVM	<i>Support Vector Machines</i> Máquinas de Soporte Vectorial
UI	<i>User Interface</i> Interfaz de Usuario
UUID	<i>Universally Unique Identifier</i> Identificador Universal Único
VEP	<i>Visual Evoked Potentials</i> Potenciales Evocados Visuales

WP	<i>Wavelet Packet</i>
	Paquetes Wavelet
WT	<i>Wavelet Transform</i>
	Transformada Wavelet

INTRODUCCIÓN

1.1 INGENIERÍA BIOMÉDICA

La Ingeniería Biomédica, como su propio nombre indica, es una ciencia que une la Ingeniería y la medicina. La Ingeniería trata de resolver los problemas, satisfacer las necesidades y mejorar el bienestar de las personas aplicando conocimientos científicos y tecnológicos. La medicina trata de mejorar la salud humana basándose en el estudio de la biología y la bioquímica.

La Ingeniería Biomédica es un campo multidisciplinar que se basa en la utilización de los principios y técnicas de la ingeniería al campo de la medicina para intentar resolver, facilitar y comprender los problemas relacionados con la biología. El comienzo de esta materia difiere de unos historiadores a otros. Algunos afirman que comenzó hace más de 3000 años [1], cuando en una tumba egipcia se encontró, en el cuerpo que ésta guardaba, una prótesis del dedo gordo del pie. Otros no consideran el origen de esta especialidad hasta el desarrollo de la instrumentación eléctrica y electrónica entre los años 1890 y 1910 para poder registrar señales fisiológicas [2].

Se ha producido un avance y desarrollo increíbles desde sus inicios hasta estos días, pasando de dedicarse casi únicamente al desarrollo de instrumentos médicos, a abarcar un gran número de campos en los que se desea mejorar. Con el tiempo, cada vez se va ampliando más la lista de actividades de la Ingeniería Biomédica, debido al rápido avance de la tecnología. A continuación, se nombrarán algunos de los ejemplos más significativos de estos campos de estudio [2]:

- Aplicación de sistemas de análisis de ingeniería (modelado fisiológico, simulación y control) a problemas fisiológicos.
- Detección, medición y monitorización de señales fisiológicas (biosensores e instrumentos biomédicos).
- Interpretación diagnóstica mediante técnicas de procesamiento de señales de datos bioeléctricos.
- Procedimientos y dispositivos terapéuticos y de rehabilitación (ingeniería de rehabilitación).

- Dispositivos para el reemplazo o el aumento de las funciones corporales (órganos artificiales).
- Análisis por ordenador de datos relacionados con el paciente y toma de decisiones clínicas (informática médica e inteligencia artificial).
- Imágenes médicas, es decir, la visualización gráfica de detalles anatómicos o funciones fisiológicas.
- La creación de nuevos productos biológicos (biotecnología e ingeniería de tejidos).
- El desarrollo de nuevos materiales para ser usados dentro del cuerpo (biomateriales).

En conclusión, la Ingeniería Biomédica es una rama multidisciplinar de la ingeniería que abarca desde proyectos teóricos hasta aplicaciones de vanguardia, investigación, desarrollo e implementación. Un beneficio potencial que se deriva del uso de la Ingeniería Biomédica es la identificación de los problemas y las necesidades de la salud que pueden resolverse utilizando la tecnología de la ingeniería y la metodología de sistemas existentes. Por ello, dicha materia ofrece esperanza en la batalla continua para proporcionar atención de alta calidad a un precio razonable, y puede conseguir herramientas y técnicas para hacer más eficaz y eficiente nuestro sistema de atención médica mejorando la calidad de vida [2].

1.2 SEÑALES BIOMÉDICAS

Las señales biomédicas son aquellas que producen los sistemas biológicos que, muy a menudo, es necesario procesar para poder extraer información relevante, que sea capaz de ayudar al médico en el diagnóstico. Para poder diferenciar las distintas señales biomédicas se han realizado distintas clasificaciones, atendiendo al origen de las señales y el medio que se utiliza para su adquisición, según su descripción matemática o por sus características.

Según su descripción matemática se puede distinguir entre:

- **Señales deterministas:** son señales que se pueden describir con exactitud mediante una ecuación matemática, pudiendo de esta manera conocer los valores futuros de la señal si se disponen de los valores pasados. No obstante en el mundo real no hay señales deterministas, ya que siempre vienen acompañadas de algún tipo de ruido desconocido e impredecible, transformándolas en señales no deterministas. En ocasiones es conveniente aproximar la señal por medio de una función determinista. Este tipo de señales, a su vez, se pueden dividir en periódicas, que se repiten a lo largo del tiempo; cuasiperiódicas que se repiten, pero con leves modificaciones; y transitorias.
- **Señales aleatorias:** son aquellas de las que es imposible predecir su comportamiento futuro debido a las grandes fluctuaciones que éstas poseen. Sin embargo, se puede estudiar su estacionariedad y su ergodicidad. Una señal se considera estacionaria si su media y su varianza permanecen constantes a lo largo del tiempo. La ergodicidad de una señal se produce cuando, al poseer una sola muestra de la señal, somos capaces de deducir todos los parámetros de ésta.

Dependiendo del origen de estas señales y del medio utilizado para su adquisición, se pueden establecer los siguientes tipos [2]:

- **Señales eléctricas:** son aquellas que se generan por las células nerviosas y musculares. Se consideran unas de las señales biomédicas más importantes, utilizadas en la mayoría de los campos de la biomedicina.
- **Señales de impedancia** de los tejidos: contienen información sobre la composición, volumen y distribución sanguínea, el sistema nervioso, la actividad endocrina, etc.
- **Señales magnéticas:** como el magnetoencefalograma (MEG, *magnetoencephalography*), que registra la actividad del cerebro; y el magnetocardiograma (MCG, *magnetocardiography*), que recoge la actividad del corazón.
- **Señales mecánicas:** generadas por procesos mecánicos del cuerpo, como la locomoción, el desplazamiento y el flujo de presión.
- **Señales químicas:** generalmente resultantes de mediciones químicas de tejidos vivos, o muestras analizadas en el laboratorio.
- **Señales acústicas:** producidas por funciones del cuerpo como el flujo sanguíneo, el latido del corazón, incluso el movimiento de los músculos.
- **Señales ópticas:** resultantes de las funciones ópticas del sistema biológico, que ocurren de manera natural o son inducidas por la medida.

Como se ha comentado, unas de las señales biomédicas más importantes son las bioeléctricas, pudiendo destacar el electroencefalograma (EEG, *electroencephalography*), el electrocardiograma (ECG, *electrocardiography*) y el electromiograma (EMG, *electromyography*); siendo comúnmente las más utilizadas y que registran la actividad eléctrica del cerebro, del corazón y de los músculos respectivamente. En la Figura 1.1 se puede observar una representación de dichas señales a lo largo del tiempo para poder apreciar sus diferencias.

En este trabajo vamos a centrarnos en el EEG, ya que es la señal utilizada para manejar la aplicación a desarrollar. A continuación, dedicaremos un apartado a explicar las características más significativas de esta señal.

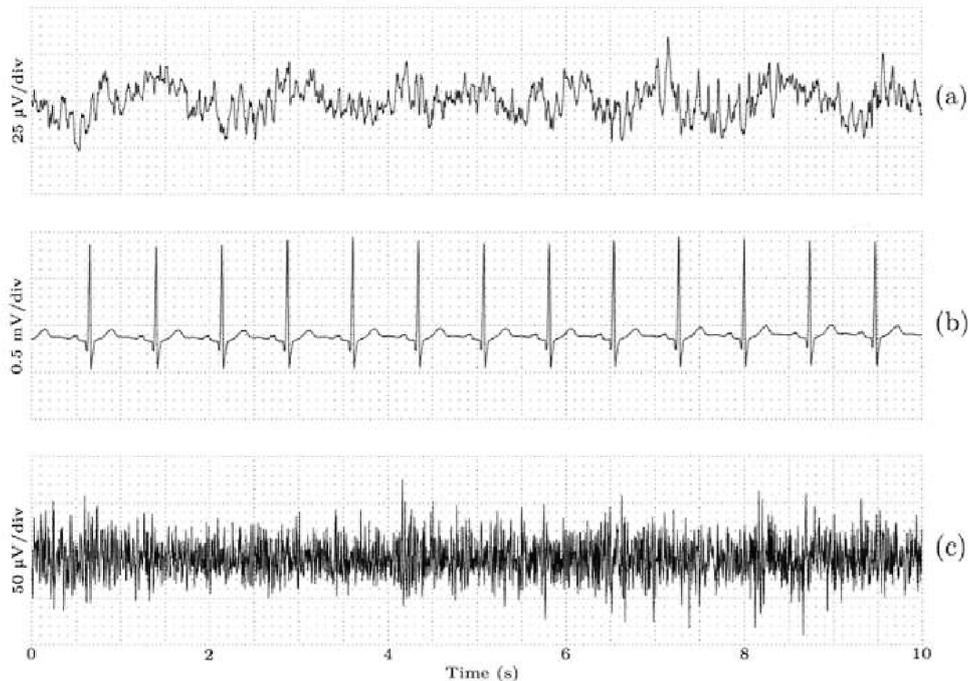


FIGURA 1.1: Representación de las principales señales bioeléctricas: (a) electroencefalograma (EEG), (b) electrocardiograma (ECG) y (c) electromiograma (EMG) [3].

1.3 ELECTROENCEFALOGRAMA (EEG)

El electroencefalograma (EEG) es una señal biomédica que refleja la actividad eléctrica del cerebro, registrada mediante electrodos no invasivos colocados en el cuero cabelludo. El precursor fue Dubois-Reymond, que en 1848 descubrió la presencia de señales eléctricas en el sistema nervioso. Sus sucesores realizaron múltiples estudios sobre ello en animales [4]. Sin embargo, fue en 1929 cuando Hans Berger describió un patrón de oscilación en la actividad eléctrica registrada en el cuero cabelludo humano. Desde entonces el EEG pasó a ser objeto de estudio de innumerables científicos, cobrando mucha importancia en el campo de la psiquiatría y la neurología [4].

El EEG registrado en el cuero cabelludo se compone de la suma de miles de millones de potenciales de acción individuales (a partir de la suma de los potenciales de grandes grupos de neuronas corticales), ya que no es capaz de registrar los potenciales de cada neurona debido a su baja amplitud. Los ciclos rítmicos observados en el EEG generalmente se consideran el resultado de la actividad neuronal entre el tálamo y el córtex cerebral. El tálamo es una estructura subcortical central que transmite señales al nivel cortical, desde las vías ascendentes y descendentes hacia otras regiones del cerebro [4]. La diversidad de esos ritmos es enorme y depende, entre otras cosas, del estado mental del sujeto: si se encuentra relajado, dormido, atento, etc. Normalmente, los ritmos se suelen caracterizar por el rango de frecuencias y la

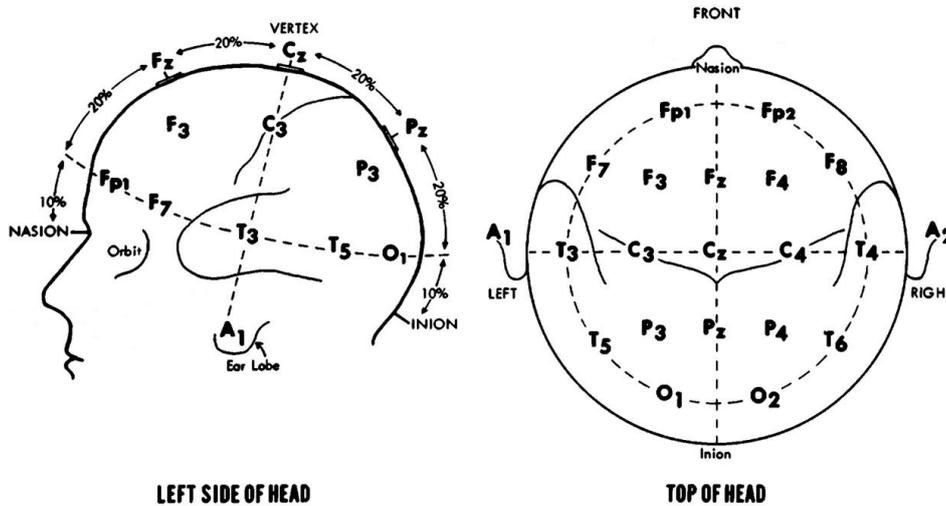


FIGURA 1.2: Representación de la localización de los electrodos en el Sistema Internacional 10-20 [5].

amplitud que posean [3]. La amplitud del EEG generalmente cae dentro del rango de $20\text{-}50\ \mu\text{V}$, aunque puede variar desde menos de $10\ \mu\text{V}$ hasta más de $100\ \mu\text{V}$. La amplitud del EEG tiende a atenuarse en respuesta a una estimulación. Por ejemplo, cuando el sujeto se encuentra con los ojos abiertos se produce una atenuación si lo comparamos con el sujeto con ojos cerrados [4]. Para analizar el comportamiento del EEG en el dominio de la frecuencia, el método más utilizado sea probablemente la Transformada rápida de Fourier (FFT, *fast Fourier Transform*), aunque existen muchos otros. El resultado del análisis espectral se puede separar en cinco bandas de frecuencia: delta (menos de 4 Hz), theta (4-7.5 Hz), alpha (7.5-12.5 Hz), beta (12-30 Hz) y gamma (>30 Hz). La actividad dentro de las bandas delta y theta se denomina *slow-wave activity*, y la que se encuentra en las bandas alpha, beta y gamma, *fast-wave activity* [4].

El registro de esta señal se realiza mediante la colocación de electrodos según indica el sistema estándar, conocido como el Sistema Internacional 10-20. Dicho sistema simplemente mide la distancia entre puntos de referencia fácilmente identificables en la cabeza, y luego ubica a los electrodos al 10% o al 20% de esa distancia [5]. En la Figura 1.2 se puede ver la nomenclatura de los electrodos, que depende de la región del cráneo en la que se encuentren y el orden que sigan. La letra indica la región: **FP**-prefrontal, **F**-frontal, **C**-central, **P**-parietal, **O**-occipital y **T**-temporal; mientras que el subíndice depende del hemisferio en el que se coloque, dejando los números impares para el hemisferio izquierdo, y los pares para el hemisferio derecho. Si el electrodo se encuentra entre ambos hemisferios, el subíndice será la letra **z**. Como se puede comprobar en la Figura 1.2, aparecen unos electrodos que no se corresponden con esta nomenclatura. Estos se corresponden con los electrodos de referencia, que se colocan en los lóbulos de las orejas y no deberían verse afectados por la actividad cerebral denominados **A1** para el lóbulo izquierdo y **A2** para el lóbulo derecho [4].

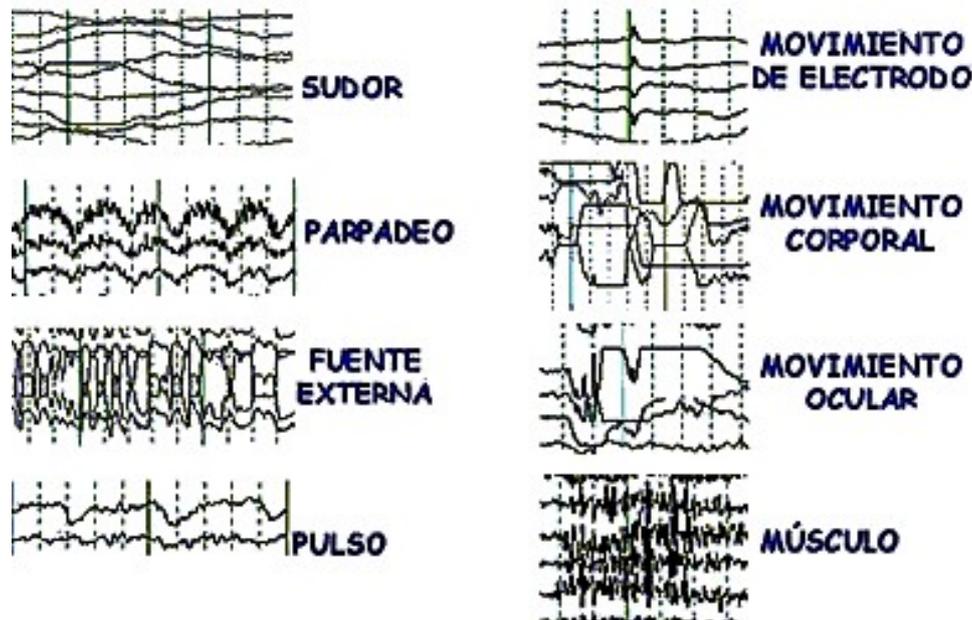


FIGURA 1.3: Representación de los efectos en el EEG producidos por algunos artefactos [7].

Cuando se realiza un registro de EEG hay que tener presente que la señal es muy débil y se encuentra rodeada de otras muchas señales. Estas últimas, al no ser de interés, las consideramos como interferencias, artefactos o ruido. Hay distintas fuentes de interferencia: señales fisiológicas, instrumentos utilizados para el registro, movimientos del sujeto, etc.

Para intentar paliar los efectos de las interferencias y retener la información de interés, normalmente se utilizan amplificadores. También se utilizan los electrodos de referencia para eliminar el ruido en la medida de lo posible, además de varios filtros espaciales como son [6]:

- Método de referencia de media común (CAR, *common average reference*): consiste en restar a la señal registrada por cada electrodo la media del resto, eliminando así las fuentes de ruido externas.
- Filtro Laplaciano: que solo resta a la señal de cada uno de los electrodos el promedio de los electrodos circundantes.
- Filtro bipolar: consiste en restar a cada electrodo la media de los vecinos posicionados en la vertical del principal.

Los artefactos producidos por el propio sujeto deben minimizarse, explicándole detalladamente los movimientos que los provocan para intentar que se produzcan lo menos posible. Los efectos de algunas de estas interferencias se pueden observar en la Figura 1.3.

1.4 BRAIN COMPUTER INTERFACE (BCI)

Las personas, para poder comunicarse o controlar cualquier dispositivo, utilizan sus nervios periféricos y músculos. Todo comienza con la intención que provoca la activación de ciertas partes del cerebro, que envían señales a través del sistema nervioso a los músculos, y estos realizan el movimiento deseado. Un sistema interfaz cerebro-computadora (**BCI**, *brain computer interface*) es un método artificial capaz de manejar aplicaciones mediante señales de control generadas únicamente con la actividad cerebral del sujeto. Estos sistemas ofrecen una alternativa para las personas que no pueden depender de sus músculos y nervios. Por ello, se denominan Interfaz Cerebro-Ordenador [8].

El primero en describir un sistema BCI fue el Dr. Grey Walter en 1964, al conectar electrodos directamente a las áreas motoras de un paciente que debía ser operado. Se le pidió al paciente que presionara un botón para avanzar unas diapositivas que se visualizaban en un proyector mientras se registraba la actividad cerebral, con la idea de después controlarlo sin presionar el botón. A partir de entonces el progreso en este campo ha ido creciendo muy rápidamente, y hoy en día existen un gran número de grupos de investigación dedicados a temas de BCI [8].

Estos sistemas han surgido de la necesidad de establecer métodos alternativos de comunicación para personas que están incapacitadas por alguna enfermedad (distrofia muscular, esclerosis lateral amiotrófica, etc), o accidente (lesiones medulares, apoplejía, amputaciones de miembros, etc), y no pueden utilizar su sistema motor. Hoy en día también hay investigadores dedicados a la creación de estos sistemas para personas sanas, con la finalidad de controlar juegos o como método de rehabilitación [9].

Existen distintos métodos para registrar la actividad cerebral, y se pueden clasificar en métodos **invasivos** o **no invasivos**. Dentro de los no invasivos se encuentran: el **EEG**, el **MEG**, la espectroscopia de infrarrojo cercano (**NIRS**, *near-infrared spectroscopy*), la imagen de resonancia magnética funcional (**fMRI**, *functional magnetic resonance imaging*) y la tomografía de emisión de positrones (**PET**, *positron emission tomography*). Los métodos invasivos son: el electrocorticograma (**ECoG**, *electrocorticography*), donde los electrodos se implantan en la superficie del cerebro; **electrodos epidurales o intracorticales**, donde los electrodos se implantan directamente en el córtex cerebral [9].

Los sistemas BCI pueden clasificarse en **sistemas endógenos** o **sistemas exógenos**. Los sistemas endógenos no necesitan ningún tipo de estimulación externa para la generación de la actividad cerebral deseada. Las señales de control utilizadas en este tipo de sistemas son: los potenciales corticales lentos (**SCP**, *slow cortical potentials*) y los ritmos sensoriomotores (**SMR**, *sensory-motor rhythms*). En cambio, los sistemas exógenos requieren de una estimulación externa para producir la actividad cerebral necesaria. Las señales de control utilizadas en este tipo de sistemas son: los potenciales evocados visuales

(**VEP**, *visual evoked potentials*) y los potenciales evocados **P300** [8].

Las señales cerebrales recogidas por los sistemas BCI deben ser procesadas en tiempo real para poder determinar la intención del usuario en cada momento. Dicho procesado se puede dividir en tres etapas. La primera es el preprocesado, encargada de mejorar la calidad de la señal reduciendo la influencia del ruido. La segunda es la extracción de características, donde se calculan parámetros que permitan caracterizar las intenciones del usuario. Por último, la clasificación de características, donde se determina si el usuario está realizando una selección (intención) en tiempo real, y la transforma en comandos de un dispositivo [8].

1.5 NEUROFEEDBACK TRAINING

Los sistemas BCI detectan cambios en las señales cerebrales que reflejan la intención del usuario. A continuación, las traducen en señales de control mediante sofisticados algoritmos de reconocimiento y clasificación de patrones. La investigación de BCI se ha centrado principalmente en desarrollar poderosas técnicas de procesamiento de señales y aprendizaje automático (*machine learning*), para clasificar con precisión la actividad neuronal. Sin embargo, incluso con los mejores algoritmos, el buen manejo de BCI depende enormemente de cómo los usuarios puedan modular voluntariamente su actividad neuronal. Las personas necesitan producir señales que sean fáciles de detectar. Aprender a manejar dispositivos controlados por BCI requiere de una práctica repetida con retroalimentación (*neurofeedback*) [8].

En este aspecto, el entrenamiento basado en *neurofeedback* (NFT, *neurofeedback training*) se ha demostrado efectivo para que los usuarios consigan modular ciertas características de su actividad cerebral. Las investigaciones que exploran el entrenamiento BCI deben considerar los diferentes protocolos de entrenamiento y las técnicas de retroalimentación que, dependiendo de la función objetivo, serán más o menos eficientes. La comprensión de los principios subyacentes del *neurofeedback* permite adaptar el proceso de entrenamiento de acuerdo con los fundamentos del condicionamiento operante. Aunque no existen datos empíricos sobre un entorno de entrenamiento óptimo, los usuarios pueden beneficiarse de las experiencias con el NFT [8].

Normalmente las definiciones de *neurofeedback* incluyen el proceso de aprendizaje y el condicionamiento operante como elementos básicos del modelo teórico. Las investigaciones han establecido que las personas pueden aprender a controlar diversos parámetros de la actividad eléctrica del cerebro, a través de un proceso de entrenamiento que implica la visualización en tiempo real de los cambios producidos en el EEG. En un paradigma de *neurofeedback* de este tipo, la retroalimentación generalmente se presenta de forma visual, al representar la señal cerebral en un monitor, o mediante modalidades auditivas o táctiles. Ésto plantea la pregunta de cómo representar mejor las diferentes señales cerebrales. A menudo las señales de retroalimentación se presentan en un formato computerizado similar a un juego. Estos entornos pueden

favorecer la motivación y atención del usuario para lograr un objetivo específico (cambios específicos en la actividad cerebral) [8].

Los cambios en la actividad cerebral que reflejan el entrenamiento exitoso de *neurofeedback* se recompensan o refuerzan positivamente. Típicamente, los umbrales se establecen para conseguir una efectividad máxima del entrenamiento, así como los criterios de recompensa se basan en modelos de aprendizaje. Esto significa que la tarea debe ser desafiante aunque el usuario se sienta motivado y recompensado, para evitar su frustración y aburrimiento. Los métodos de NFT se han utilizado ampliamente para obtener beneficios clínicos asociados con la mejora y/o supresión de características particulares del EEG, que se correlacionan con un estado “normal” del funcionamiento cerebral [8].

1.6 OBJETIVOS DEL TRABAJO FIN DE GRADO

El objetivo perseguido en la elaboración de este Trabajo Fin de Grado es diseñar y desarrollar una plataforma de entrenamiento cognitivo mediante un sistema BCI, para dispositivos Android. La aplicación está destinada a mejorar el rendimiento de ciertas funciones cerebrales gracias al entrenamiento cognitivo que conlleva. Para poder alcanzar este objetivo global, se definieron una serie de objetivos intermedios:

1. **Realizar un estudio previo referente a las bases de los sistemas BCI que incluye:**
 - Estudio de los distintos métodos de registro de la actividad cerebral.
 - Estudio de las distintas señales de control de dichos sistemas.
 - Estudio del procesado que se realiza a las señales.
 - Estudio en profundidad de la señal de control elegida: los ritmos sensoriomotores.
2. **Realizar una revisión del estado del arte de otras aplicaciones de entrenamiento cognitivo para:**
 - Determinar que partes debería tener la aplicación.
 - Elegir el mejor diseño para la aplicación.
 - Establecer las funcionalidades que tendría que incluir la aplicación.
3. **Estudiar los lenguajes de programación con los que se debe realizar la aplicación, que son: Java, C++ y C#.**
4. **Desarrollar la aplicación, que consta de tres partes:**
 - Modificar el bloque *CursorTask* de BCI2000.
 - Implementar el menú de la aplicación en *Android Studio*.
 - Implementar las tareas para el entrenamiento cognitivo, con gráficos en 3D, en Unity.

5. **Evaluar la aplicación con 4 sujetos sanos.**
6. **Analizar los resultados obtenidos en *MATLAB*.**
7. **Discutir y comparar los resultados obtenidos.**
8. **Extraer conclusiones del trabajo realizado.**

1.7 ESTRUCTURA DEL TRABAJO FIN DE GRADO

El trabajo Fin de Grado está compuesto por 7 capítulos, que son: introducción, sistemas Brain Computer Interface (BCI), análisis de la señal y ritmos sensoriomotores (μ y β), diseño y desarrollo de una aplicación de neurofeedback training, resultados, discusión y conclusiones.

En el capítulo 1, introducción, se da una visión general sobre la Ingeniería Biomédica, explicando que tipo de señales se utilizan en ese campo y las diferentes maneras que existen de clasificarlas. También se intenta exponer brevemente en qué consisten los sistemas BCI, en qué etapas se dividen, las señales que utilizan, cómo se clasifican y para qué sirven. Para finalizar, se detallan los objetivos y la estructura de este Trabajo Fin de Grado.

En el segundo capítulo, sistemas Brain Computer Interface, se exponen de manera mucho más exhaustiva los puntos tratados en el capítulo de Introducción. En primer lugar, se detallan los diferentes métodos de registro de la actividad cerebral, explicando sus características. A continuación, se enumeran los tipos de señales de control de los sistemas BCI, describiendo su forma y propiedades. Después, se pasa al procesado de la señal comúnmente más utilizada en los sistemas BCI, el EEG, y se determinan las distintas aplicaciones que han ido surgiendo. Por último, se realiza una comparativa entre diferentes estudios, donde se utiliza el neurofeedback training basado en BCI.

En el capítulo de análisis de la señal y ritmos sensoriomotores se especifican las propiedades de los ritmos sensoriomotores, señal de control escogida para manejar la aplicación ha desarrollar. Posteriormente, se detallan los diferentes métodos existentes para procesar dicha señal, desde la adquisición de ésta, pasando por la extracción de características, y por los métodos de clasificación, para poder determinar la intención del usuario.

En el siguiente capítulo, diseño y desarrollo de una aplicación de neurofeedback, se establecen los objetivos que se desean alcanzar al realizar la aplicación. Posteriormente, se detalla la arquitectura de la aplicación, intentando explicar cómo se ha desarrollado desde la adquisición y procesado de la señal escogida, hasta los bloques en los que se divide y las funciones que realiza cada uno. Además, se presenta un manual de usuario para explicar, de manera más visual, cómo se maneja la aplicación.

En el quinto capítulo, resultados, se muestran los resultados obtenidos, tanto en las sesiones de entrenamiento como en el análisis *offline* de los datos recogidos y en el cuestionario de satisfacción realizado por los usuarios. En tales resultados se incluyen: las precisiones de las tareas realizadas, los cambios producidos en el EEG como consecuencia del entrenamiento y las opiniones y sugerencias de mejora recogidas en los cuestionarios de satisfacción.

En el capítulo de discusión se justifican los resultados obtenidos en el apartado anterior. Además, se realiza una comparación con los estudios realizados anteriormente y se establecen las limitaciones de la aplicación que se han observado durante su evaluación.

Finalmente, en el capítulo de conclusiones y líneas futuras, se recogen las principales conclusiones extraídas con la elaboración de este Trabajo Fin de Grado y se plantean las posibles líneas de trabajo futuro surgidas a partir de su realización.

BRAIN COMPUTER INTERFACE (BCI)

2.1 INTRODUCCIÓN

La tecnología Brain Computer Interface (BCI) ha experimentado un gran progreso en los últimos años, convirtiéndose en una poderosa herramienta de comunicación entre humanos y máquinas. Gracias al descubrimiento del EEG en 1929 por Hans Berger, estos sistemas no requieren de ningún tipo de intervención muscular o dispositivo externo para ser capaz de emitir comandos que gobiernan la interacción humano-máquina. En un principio, los grupos orientados a la investigación en este campo tenían como objetivo el desarrollo de dispositivos de asistencia. Estos dispositivos estaban orientados a restaurar la capacidad de movimiento y comunicación de sujetos con incapacidades físicas, reemplazando la funcionalidad motora perdida [10] [11]. Algunos ejemplos son [12]:

- Dispositivos prostéticos: como el implante coclear para personas sordas o las prótesis de brazos o piernas, para amputados y pacientes con lesiones en la médula espinal.
- Dispositivos de estimulación cerebral: para tratar los síntomas de enfermedades como el Parkinson.
- Dispositivos para controlar cursores o deletreadores de palabras: como forma de comunicación para personas que padecen enfermedades como la esclerosis lateral amiotrófica (ALS, *amyotrophic lateral sclerosis*) o apoplejías.
- Dispositivos para el manejo de sillas de ruedas: destinados a pacientes con parálisis cerebral, distrofia muscular, lesiones en la médula espinal, etc.

No obstante ahora, debido al prometedor futuro de BCI, el alcance de la investigación está creciendo. Han surgido grupos cuya finalidad es la generación de aplicaciones de “manos libres” para sujetos sanos, con las implicaciones éticas que conllevan [12].

En todos los sistemas BCI el objetivo es traducir la actividad cerebral en comandos para controlar dispositivos, estimular el cerebro para proporcionar retroalimentación sensorial, o restaurar funciones neuropsicológicas. Para ello, se siguen una o más de las etapas descritas a continuación, representadas en

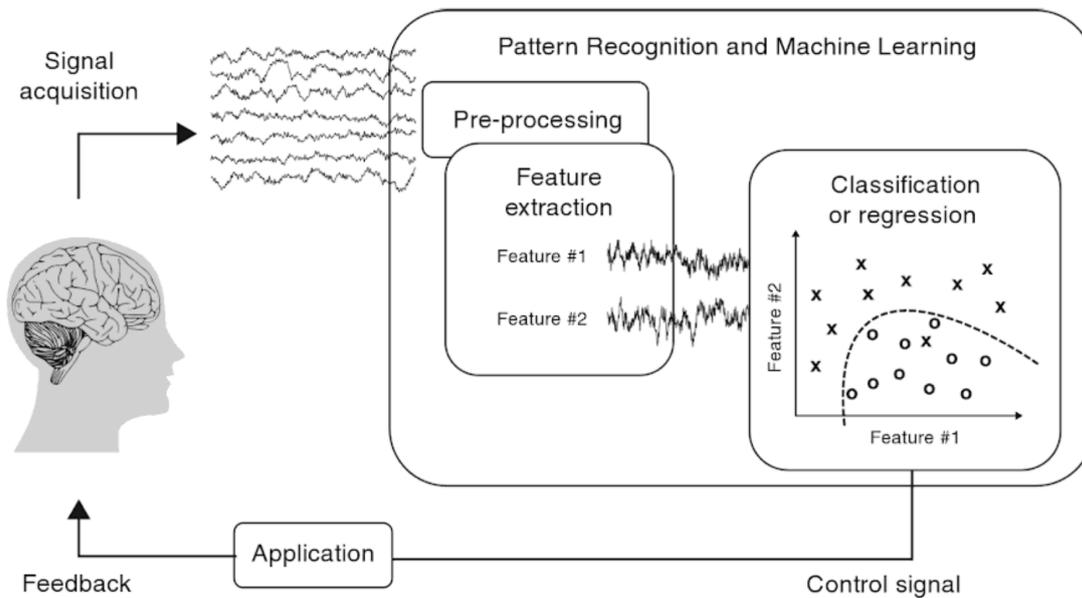


FIGURA 2.1: Esquema de los sistemas BCI [12].

la Figura 2.1:

- **Registro de la actividad cerebral:** existen diferentes métodos para registrar las señales del cerebro y se pueden clasificar en técnicas invasivas o no invasivas.
- **Procesado de la señal:** después de haber obtenido la señal con uno de los métodos de adquisición se realiza el procesado de la señal, el cual se puede dividir en tres etapas:
 - Preprocesado: etapa en la que se intenta mejorar la relación señal a ruido (SNR, *signal-to-noise ratio*) mediante distintas técnicas.
 - Extracción de características: etapa en la que se usan una serie de combinaciones y transformaciones sobre la señal con el objetivo de obtener una información más discriminatoria sobre las intenciones del usuario.
 - Clasificación de características: etapa que se encarga de transformar las características extraídas en señales de control de un dispositivo o de una aplicación.
- **Aplicación:** la señal de control resultante del procesado, debe causar cambios en el entorno (*feedback*), como el movimiento de una silla de ruedas o de una prótesis. Estos cambios pueden percibirse por la vista, el oído o el tacto del usuario.

2.2 MÉTODOS DE REGISTRO DE LA ACTIVIDAD CEREBRAL

El registro efectivo de la actividad cerebral es un primer paso crítico para cualquier sistema BCI. Las señales cerebrales son el reflejo de las acciones neuronales regidas por la actividad que esté realizando el usuario en ese momento.

Estos métodos se pueden clasificar en dos grupos: invasivos y no invasivos. Para determinar qué método es el más adecuado y qué fenómeno se medirá, se debe tener en cuenta tanto la finalidad de la aplicación BCI, como a qué sector de la población está destinada.

2.2.1 MÉTODOS INVASIVOS

Los métodos invasivos se denominan así debido a que requieren de cirugía para poder ser realizados, ya que los electrodos se implantan una vez realizada una craneotomía, cortando las membranas que cubren el cerebro, dejando a éste expuesto. Los distintos métodos de esta categoría son [8]:

- **Electrocorticografía (ECoG)**: es la técnica que utiliza electrodos colocados en la superficie de la corteza cerebral. El ECoG no daña las neuronas, ya que ninguno de los electrodos penetra en el cerebro. Normalmente solo se utilizan en entornos clínicos como, por ejemplo, para monitorizar en el hospital a pacientes con epilepsia.
- **Electrodos epidurales o intracorticales**: son electrodos implantados en la corteza cerebral, por lo que la resolución espacial y temporal aumentan en gran medida, siendo capaces de capturar la actividad de una sola célula cerebral. Sin embargo, también se van deteriorando con el tiempo debido a que las células no neuronales (gliales) rodean a los electrodos formando tejido cicatrizal, que hace de aislante aumentando la impedancia de los electrodos y empeorando el registro de la señal [12].

Ambas técnicas miden la actividad eléctrica del cerebro con una gran resolución espacial y alta calidad de la señal, además de un rango de frecuencias más alto que el resto. Además, los artefactos afectan menos a la señal. No obstante, al ser necesaria neurocirugía, conlleva aspectos éticos y económicos a tener en cuenta. Al poner en peligro la vida del sujeto, no son los métodos generalmente elegidos para los sistemas BCI. En la Figura 2.2 se puede observar la región donde se colocan los electrodos en los distintos métodos de registro [9].

2.2.2 MÉTODOS NO INVASIVOS

Los métodos no invasivos registran la actividad cerebral sin suponer un riesgo para el usuario, a costa de perder calidad en la señal, consiguiendo un rango de frecuencias menor, una resolución espacial más baja y una mayor dependencia a los artefactos. Los distintos métodos de esta categoría son [9]:

- **Electroencefalograma (EEG)**: método que registra la actividad eléctrica del cerebro, mediante la colocación de una serie de electrodos sobre el cuero cabelludo del sujeto. El número de electrodos dependerá de la finalidad del sistema BCI y la señal de control escogida.

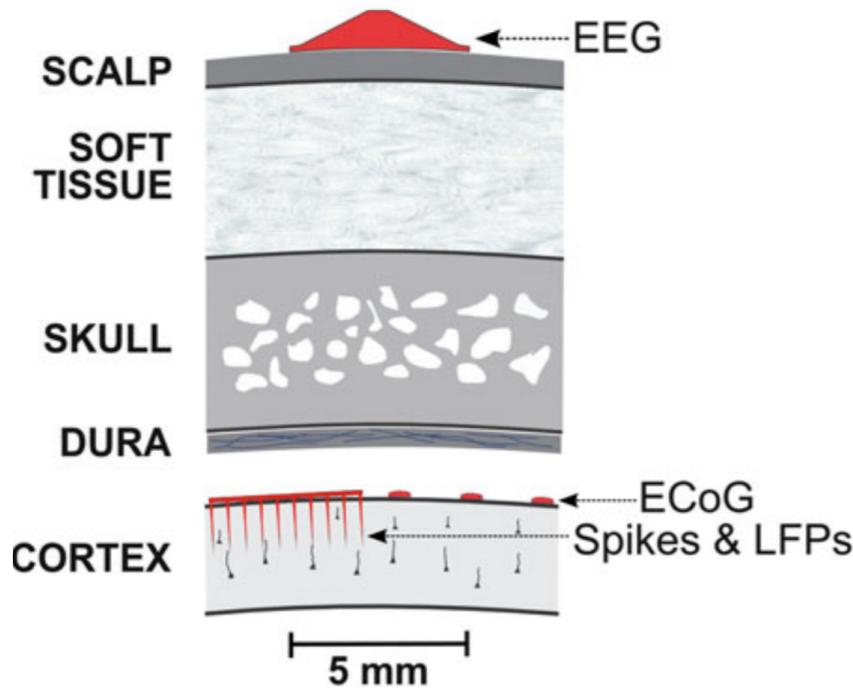


FIGURA 2.2: Localización de los electrodos en el EEG, el ECoG y los electrodos intracorticales [13].

- **Magnetoencefalograma (MEG):** método que registra la actividad magnética del cerebro mediante un magnetómetro. El más utilizado está formado por una matriz de dispositivos de interferencia cuántica superconductor (SQUID, *superconducting quantum interference devices*) [14].
- **Espectroscopia de infrarrojo cercano (NIRS):** importante herramienta de imagen que mide la absorción de la luz NIR (infrarrojo cercano con una longitud de onda de 780-3000 nm) en la hemoglobina de la sangre con o sin oxígeno. Proporciona información sobre la actividad funcional cerebral.
- **Imagen de resonancia magnética funcional (fMRI):** método de representación en imágenes de las regiones cerebrales asociadas a una actividad determinada del usuario. Cuando una zona del cerebro requiere energía para poder realizar una tarea, demanda oxígeno. Por lo que se podrán observar cambios en esa región y así identificarla con la tarea realizada.
- **Tomografía de emisión de positrones (PET):** método de imágenes tridimensionales que muestra la actividad metabólica del cuerpo. Se debe inyectar al paciente una pequeña cantidad de material radiactivo, que funcionará como marcador, y se acumulará en órganos y tejidos.

En la Tabla 2.1 se puede ver una comparativa de las características más importantes de los métodos de registro de la actividad cerebral [15]. En cuanto a resolución espacial, la fMRI, la NIRS y la PET son

TABLA 2.1: Comparación de características de los distintos métodos de registro de la actividad cerebral [15].

Método de adquisición	Actividad registrada	Medida Directa/Indirecta	Resolución temporal	Resolución espacial	Riesgo	Portabilidad
EEG	Eléctrica	Directa	~0.05 s	~10 mm	No invasivo	Portable
MEG	Magnética	Directa	~0.05 s	~5 mm	No invasivo	No portable
ECoG	Eléctrica	Directa	~0.003 s	~1 mm	Invasivo	Portable
Electrodos intracorticales	Eléctrica	Directa	~0.05 s	~0.5 mm(LFP)	Invasivo	Portable
			~0.05 s	~10 mm(MUA)		
			~0.05 s	~10 mm(SUA)		
fMRI	Metabólica	Indirecta	~1 s	~1 mm	No invasivo	No portable
NIRS	Metabólica	Indirecta	~1 s	~5 mm	No invasivo	Portable

mejores con respecto al EEG y al MEG, sin embargo, tienen una resolución temporal mucho peor.

Debido a la necesidad de los sistemas BCI de operar en tiempo real, la resolución temporal es muy importante y, por ello, el EEG y MEG son los más utilizados en sistemas BCI, además de que son no invasivos. De entre los dos es más común el EEG, debido a su bajo coste y portabilidad, ya que el MEG necesita aparatos muy grandes y habitaciones aisladas magnéticamente. Por todo ello, para la aplicación a desarrollar en este trabajo, el método de registro elegido es el EEG.

2.3 TIPOS DE SEÑALES DE CONTROL

Las señales de control son las encargadas de transmitir las intenciones del usuario a la aplicación para manejarla. Como ya se ha comentado, los sistemas BCI pueden clasificarse en endógenos y exógenos. Los sistemas endógenos requieren de una fase de entrenamiento, ya que dependen de la capacidad del usuario para modificar y controlar su actividad electrofisiológica. Las señales de este tipo son: los **ritmos sensoriomotores (SMR)** y los **potenciales corticales lentos (SCP)**. Los sistemas exógenos, en cambio, no requieren de un entrenamiento exhaustivo, debido a que la actividad electrofisiológica es provocada por algún tipo de estímulo externo independiente del usuario. Algunos ejemplos son: los **potenciales evocados visuales (VEP)** y los **potenciales P300**.

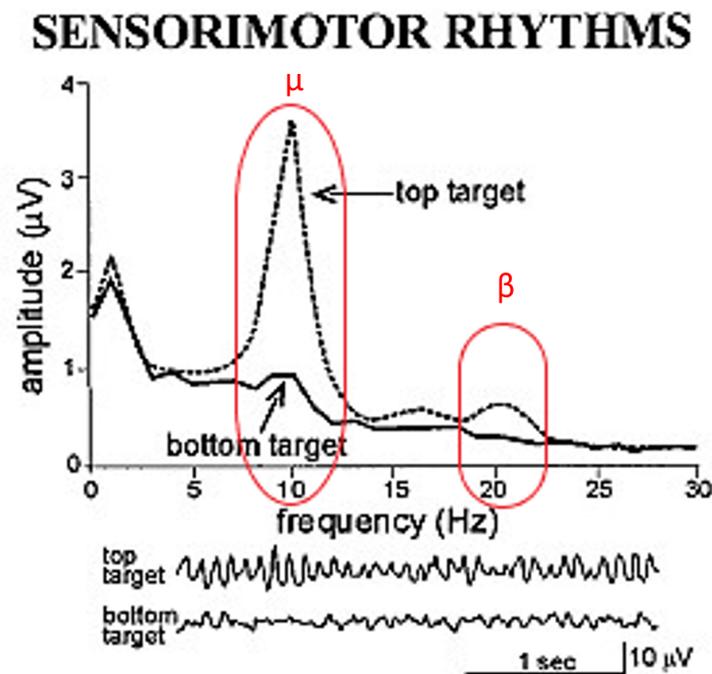
A continuación, se presenta en la Tabla 2.2 una comparación entre las diferentes señales de control anteriormente mencionadas para ver de manera rápida las diferentes características de cada una de ellas [15]. Debido a que la aplicación ha de desarrollarse basándose en NFT, se descartaron las señales exógenas. Dentro de las endógenas se escogieron los **ritmos sensoriomotores** como señal de control, debido a que las precisiones obtenidas y el ratio de transferencia eran superiores a los de los SCP.

2.3.1 RITMOS SENSORIOMOTORES (RITMOS μ Y β)

Los ritmos sensoriomotores normalmente se presentan sobre las zonas corticales motoras primarias. La actividad se encuentra entre 8-12 Hz (ritmos μ) y 18-26 Hz (ritmos β) [8]. Los distintos ritmos que

TABLA 2.2: Comparación de características de las diferentes señales de control de los sistemas BCI [15].

Señal	Fenómeno fisiológico	Nº elecciones	Entrenamiento	Ratio de transferencia
VEP	Modulación de la señal cerebral en la corteza visual	Alto	No	60-100 bits/min
SCP	Cambios en los voltajes lentos de las señales cerebrales	Bajo (2 o 4, muy difícil)	Sí	5-12 bits/min
P300	Picos positivos debidos a estímulos infrecuentes	Alto	No	20-25 bits/min
SMR	Modulaciones en los SMR sincronizadas a actividades motoras	Bajo (2, 3, 4, 5)	Sí	3-35 bits/min

FIGURA 2.3: Representación de la amplitud de los ritmos μ y β al realizar una elección entre dos objetivos (arriba y abajo) [16].

se producen pueden diferenciarse por ubicación, frecuencia y/o relación con algún estímulo sensorial concurrente o movimiento motor. Los ritmos están muy relacionados entre sí, ya que algunos de los ritmos β son armónicos de los ritmos μ , a diferencia de otros que son claramente distinguibles entre sí, proporcionando características del EEG independientes [8]. En la Figura 2.3 se pueden apreciar los cambios de amplitud de los ritmos μ y β al controlar un cursor para moverlo a la parte superior de la pantalla, o a la parte inferior. El control del cursor está enfocado en la banda del ritmo μ , ya que su amplitud aumenta si se quiere mover el cursor hacia arriba y disminuye si se quiere mover hacia abajo.

Al producirse un movimiento, o la preparación de éste, se presenta una disminución de los ritmos sensoriomotores en la región motora contralateral a dicho movimiento. Dicha disminución es debida a la desincronización relacionada con eventos (**ERD**, *event-related desynchronization*). Por el contrario, ante

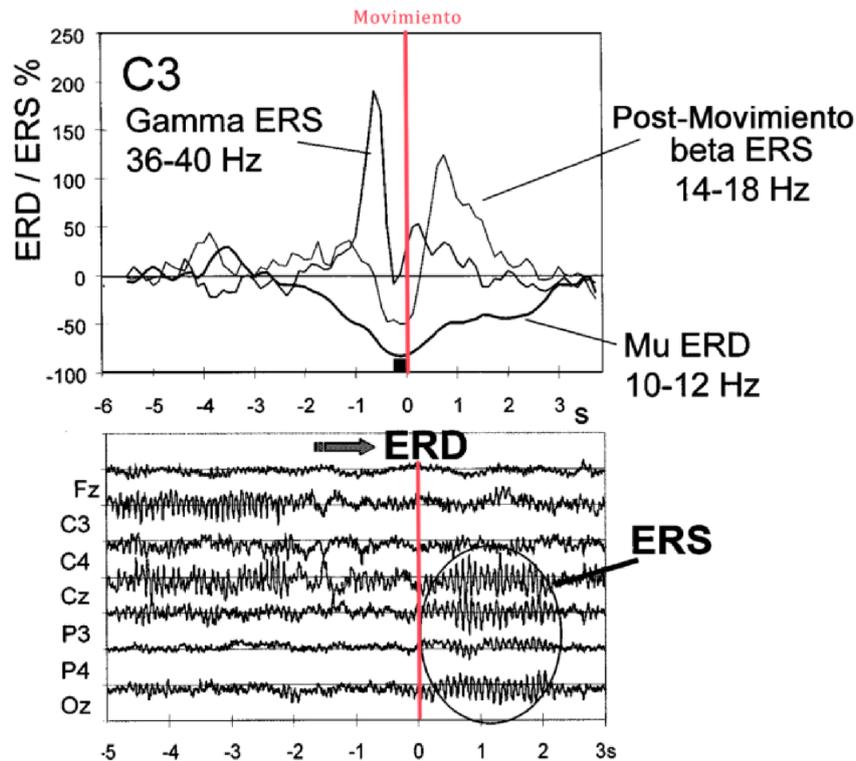


FIGURA 2.4: Superposición de la relación entre ERD/ERS % de tres bandas de frecuencias diferentes registradas en el electrodo C3 antes y después del movimiento de un dedo de la mano derecha [16].

la relajación que ocurre al finalizar el movimiento, se produce un aumento de estos ritmos por la sincronización relacionada con eventos (**ERS**, *event-related synchronization*) [8]. Ambos fenómenos no solo ocurren con la realización de una actividad motora, sino también con la imaginación o la visualización de esa actividad. Por este motivo, los ritmos sensoriomotores son muy utilizados como señal de control en sistemas BCI ya que proporcionan independencia motora a la comunicación persona-máquina.

En las distintas etapas del movimiento, los cambios más significativos se pueden ver en el ERD de los ritmos μ , y en el ERS de los ritmos β posterior al movimiento. También puede resultar interesante registrar las oscilaciones inducidas en frecuencias cercanas a 40 Hz. Estas oscilaciones de 40 Hz son las más fuertes poco antes de que comience un movimiento, mientras que los ritmos β son los más fuertes después de que el movimiento haya finalizado [16]. Todo ello lo podemos observar en la Figura 2.4.

Desde mediados de la década de los 80, se han desarrollado varios BCI basados en los ritmos sensoriomotores. Con ellos queda demostrado que las personas son capaces de controlar las amplitudes de los ritmos μ y β solo con la imaginación del movimiento (**MI**, *motor imagery*). Cabe destacar que el número de dimensiones que se quieran controlar en la aplicación BCI está muy relacionado con la cantidad de

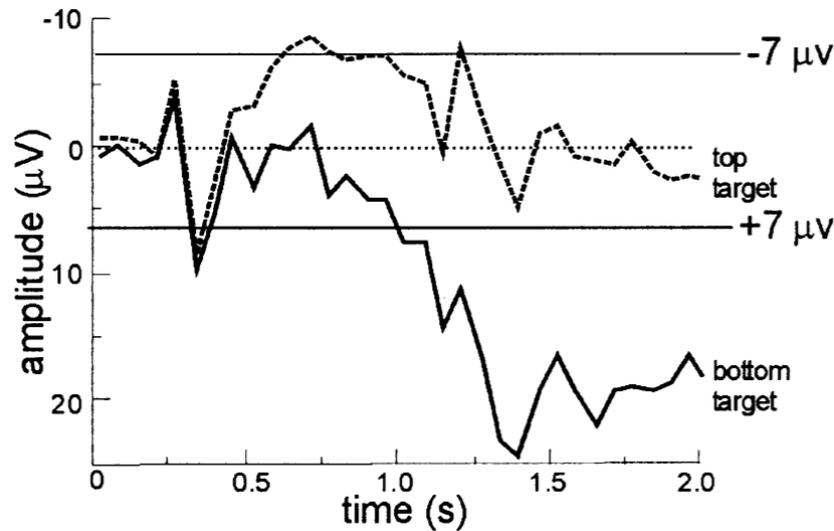


FIGURA 2.5: SCP para mover un cursor hacia un objetivo (por ejemplo, una letra o icono deseado) en la parte inferior (SCP positivo) o superior (SCP negativo) de una pantalla [16].

entrenamiento necesario para que la persona sea capaz de manejarla con una exactitud significativa [8]. En los estudios publicados, se han conseguido precisiones de hasta 95 % y tasas de transferencia de entre 20-25 bits/min [16]. Es una de las señales más utilizadas para BCI, aunque hay personas que no son capaces de controlarlas nunca.

2.3.2 POTENCIALES CORTICALES LENTOS (SCP)

Los potenciales corticales lentos (SCP) son cambios lentos de la señal de EEG, generados en la corteza cerebral, de entre 0.5-10 s. Los SCP negativos están relacionados con el movimiento, además de otras funciones que conllevan activación cortical. Los SCP positivos acompañan a la inhibición mental. Es decir, que reflejan un aumento y una disminución, respectivamente, en la excitación de las neuronas corticales [8] [16]. Por ejemplo, un SCP negativo de desarrollo lento ocurre entre un estímulo de advertencia y un estímulo que requiere una respuesta. El formato estándar para registrar la señal de EEG es con los electrodos sobre los vértices vinculados a los mastoides. Podemos ver un ejemplo de SCP en la Figura 2.5.

En las investigaciones desarrolladas con SCP como señal de control en sistemas BCI, se ha demostrado que las personas son capaces de dominar el movimiento de un objeto en la pantalla de un dispositivo. Es el resultado de un entrenamiento, no es algo que se produzca de manera espontánea ante un estímulo. Normalmente se utiliza para mover un cursor en el eje vertical, mientras se mueve con velocidad constante por el horizontal. El usuario aumenta o decrementa la amplitud de los SCP para alcanzar los objetivos que se presentan, en la parte superior o inferior, respectivamente [8]. Se han conseguido precisiones de hasta un 75 %, y tasas de transmisión de unos 15 bits/min [16].

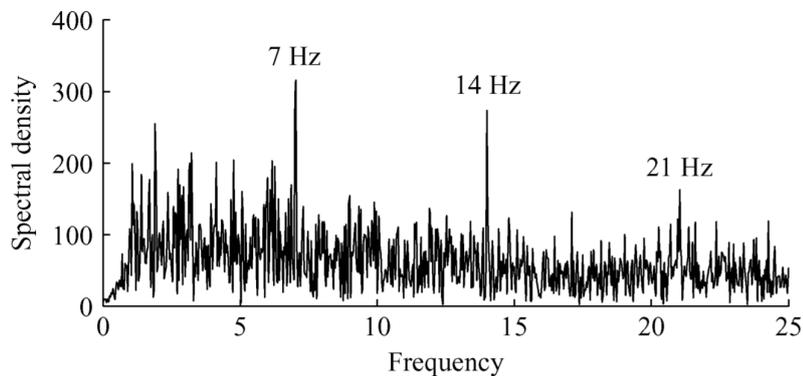


FIGURA 2.6: Espectro de frecuencia típico de una señal EEG adquirida durante la estimulación visual con una frecuencia de parpadeo de 7 Hz. La respuesta VEP puede verse como los picos a 7 Hz y sus armónicos 14 y 21 Hz [17].

2.3.3 POTENCIALES EVOCADOS VISUALES (VEP)

Los potenciales evocados visuales son diferencias de potencial registradas en la zona de la corteza visual, causadas por estímulos visuales breves. Se utilizan para determinar la dirección de la mirada, y así poder interpretar hacia donde desea mover el cursor el usuario [17].

La amplitud de los VEP se calcula unos 100 ms después de la presentación del estímulo visual. A continuación, se compara con VEP registrados sin estimulación para el mismo usuario y así se decide dónde estaba mirando el sujeto. Estos sistemas dependen de la capacidad del usuario para controlar la dirección de la mirada. El método estándar para utilizar los VEP es la presentación de varios objetivos (botones, letras, iconos, etc.), los cuales parpadean a diferentes frecuencias. Posteriormente, se realiza una medición de los VEP para ver con qué frecuencia aparecen, y en el momento en el que el sujeto fija la mirada en uno de los objetivos, aparecerá un pico en la misma frecuencia a la que parpadea el objetivo seleccionado así como en sus armónicos, gracias al fenómeno de resonancia. Al detectar la frecuencia en la que aparece esa elevación espectral se determinará la intención del usuario. Un ejemplo de ello se puede apreciar en la Figura 2.6.

Las respuestas más fuertes de VEP se obtienen para las frecuencias de estimulación en el rango de 5-20 Hz. Utilizando esta señal de control se consiguen precisiones de hasta el 90 %, además de un ratio de transferencia de alrededor de 27 bits/min [17]. Las cifras son muy prometedoras, no obstante las frecuencias más fiables se centran en torno al rango de 13-19 Hz, que casualmente es donde se maximiza el riesgo de ataques epilépticos y fatiga visual [18].

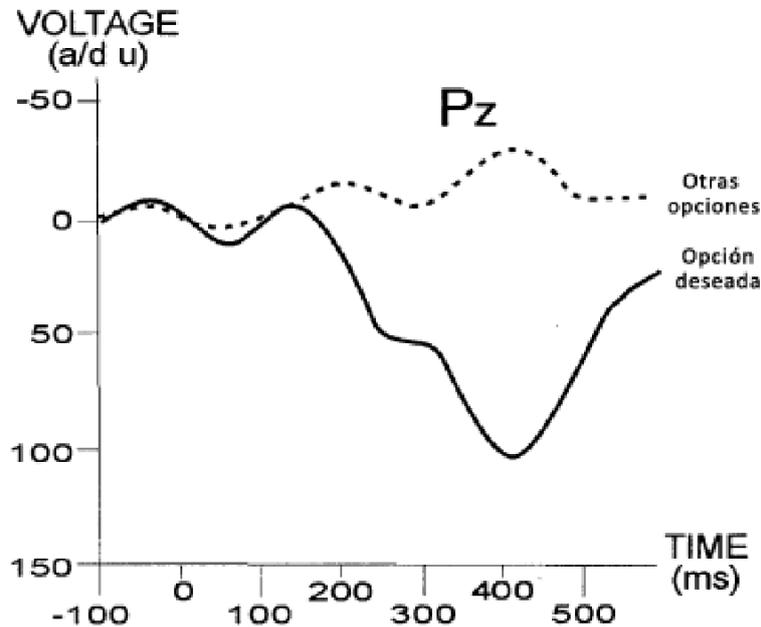


FIGURA 2.7: Respuestas del EEG tomadas de la zona centroparietal ante las posibles opciones de una matriz que parpadean. Solo la elección deseada por el usuario presenta un P300 [16].

2.3.4 POTENCIALES EVOCADOS P300

Los potenciales evocados P300 son picos de voltaje que aparecen sobre la zona parietal del córtex como respuesta a un estímulo, aproximadamente unos 300 ms después de la presentación de éste. Dichos estímulos pueden ser visuales, auditivos o somatosensoriales. Deben aparecer de forma inesperada ante el usuario para provocar en el EEG el pico característico del P300 en la zona parietal del córtex.

Lo más común es utilizar los P300 para poder discernir la elección del usuario en matrices formadas por celdas, ya pueden ser letras para escribir, comandos, etc. Las filas y las columnas de la matriz se van iluminando aleatoriamente, y el sujeto debe fijar su atención en la opción de la matriz que desea escoger. Como el objetivo deseado se presenta de forma infrecuente entre estímulos frecuentes, el P300 aparecerá cuando se ilumina la fila y la columna que contiene la celda deseada [17]. Se calcula la respuesta promedio de cada fila y columna para obtener la amplitud del P300 para cada posible elección. La forma estandarizada del P300 se puede observar en la Figura 2.7.

En la realización de varios experimentos ha quedado demostrado que la frecuencia a la que se iluminan los objetivos influye en la exactitud de los resultados, por lo que es un parámetro a tener muy en cuenta. Además, cuanto menos se espere el usuario el estímulo deseado, más probabilidades habrá de que se presente el P300. Con este método se han llegado a conseguir precisiones entorno al 90% sin apenas entrenamiento, ya que la presentación del P300 es espontánea, con ratios de transferencia de entre 20-30

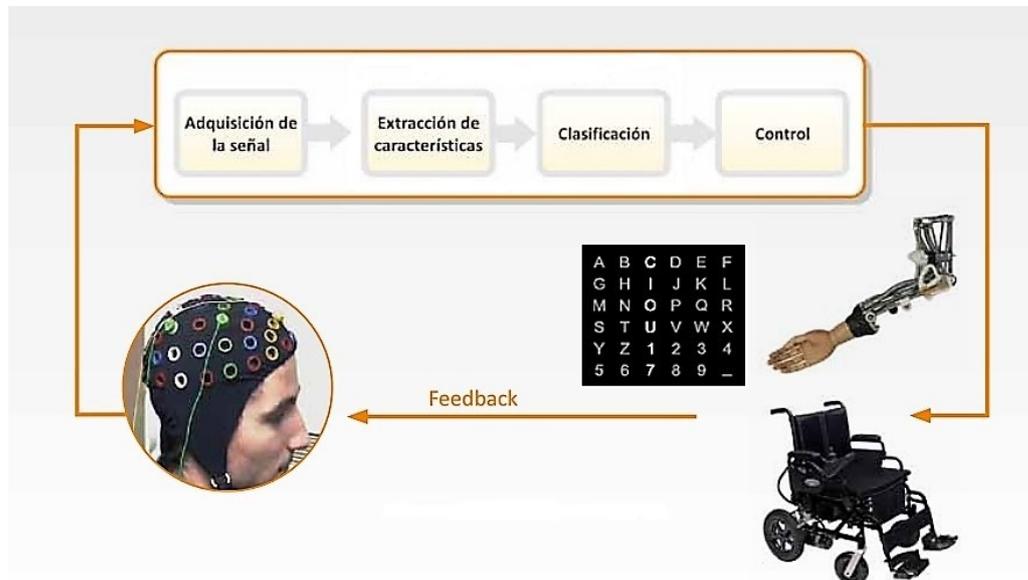


FIGURA 2.8: Arquitectura de un sistema BCI.

bits/min. Con el paso del tiempo, puede que el usuario se acostumbre a los estímulos tomándolos como estímulos frecuentes, por lo que la señal P300 puede dejar de apreciarse de manera clara. Se trata de una de las señales de control más utilizadas en sistemas BCI debido a la elevada precisión que se consigue sin la necesidad de entrenamiento.

2.4 PROCESADO DEL EEG

El procesado del EEG son las etapas que se producen desde la adquisición de las señales de EEG hasta la generación de comandos de control de las aplicaciones BCI. Estas etapas se dividen en **adquisición y pre-procesado de la señal**, **extracción de características**, **selección de características** y **clasificación de características**.

2.4.1 ADQUISICIÓN Y PRE-PROCESADO DE LA SEÑAL

Esta primera etapa consiste en el registro de la señal de EEG, la amplificación de dicha señal para su posterior digitalización y la eliminación de los artefactos que puedan estar afectándola.

La adquisición de la señal comienza con la colocación de los electrodos en el cuero cabelludo, localizados según el *sistema internacional 10-20*. Debido a la débil amplitud de las señales, se debe aplicar un gel conductor entre el electrodo y el cuero cabelludo para poder reducir la impedancia del cráneo. Así se podrán obtener voltajes mayores de señal (del rango de μV) después de amplificarla. El número de electrodos, así como su localización y la señal de control a emplear, dependerán del tipo de aplicación.

Después se digitalizará para realizar un filtrado que elimine, en la medida de lo posible, los artefactos que estén afectando a la señal. Hay distintos tipos de artefactos, y podemos clasificarlos entre **fisiológicos** (surgen de las funciones del cuerpo del sujeto) y **no fisiológicos** (campos electromagnéticos de fuentes externas al cuerpo). Algunos de estos artefactos se puede apreciar en la Figura 2.9. Los artefactos fisiológicos son [19]:

- **Movimiento involuntario en los músculos del cuerpo:** que provocan actividades eléctricas musculares, **EMG**. Los que más afectan al EEG son la contracción de los músculos del cuero cabelludo, cara, mandíbula y cuello. Normalmente tiene lugar cuando el sujeto está ansioso, en tensión o tiene dificultades para relajarse o estarse quieto. Para reducir los efectos de estos artefactos se colocan electrodos de referencia que se puedan correlacionar simultáneamente con el artefacto en el registro.
- **Movimiento del globo ocular y parpadeo:** electrooculograma (**EOG**, *electrooculography*) que genera artefactos producidos por cambios de potencial en el ojo, afectando con mayor intensidad a los electrodos de las zonas frontal y frontoparietal. Se pueden reducir sus efectos colocando un electrodo de referencia sobre la nariz, y usando un factor de rechazo común mediante un amplificador diferencial.
- **Movimiento de la lengua, Potencial de Glossokinetic (GKP, *Glossokinetic potencial*):** producen cambios en el campo eléctrico cercano a la boca y la mandíbula, debido a las características dipolares de la lengua. Se puede hacer un análisis previo para entender las características de estos artefactos, haciendo un registro con el paciente relajado y a continuación pedirle que haga movimientos con la lengua.
- **Actividad eléctrica del corazón:** puede afectar al EEG y confundirse con la actividad epileptiforme. Se puede minimizar la amplitud de estos artefactos si se realiza un registro bipolar, en vez de uno con referencia común. Esta actividad se mide mediante el **ECG**.

Los artefactos no fisiológicos más destacables son [19]:

- **Frecuencia de la corriente eléctrica (50/60 Hz):** producida por la presencia de dispositivos eléctricos, como el propio equipo de registro de EEG. Si la señal se ve afectada por este artefacto y no podemos desconectar el equipo que lo produce, la única alternativa es aplicar un filtro de ranura o filtro *notch* a 50 o 60 Hz para eliminar esa frecuencia.
- **Interrupción en el contacto del electrodo con la piel:** por la falta de gel o el mal funcionamiento de algún electrodo que produce cambios muy bruscos de potencial. Se debe eliminar la frecuencia del artefacto con filtros pasobajo, si se encuentra a bajas frecuencias.
- **Alteraciones en el ambiente alrededor del sujeto:** pueden producirse por cables que actúan como antenas (tomas de corriente, equipos de conmutación, etc.). Teniendo un ambiente controlado

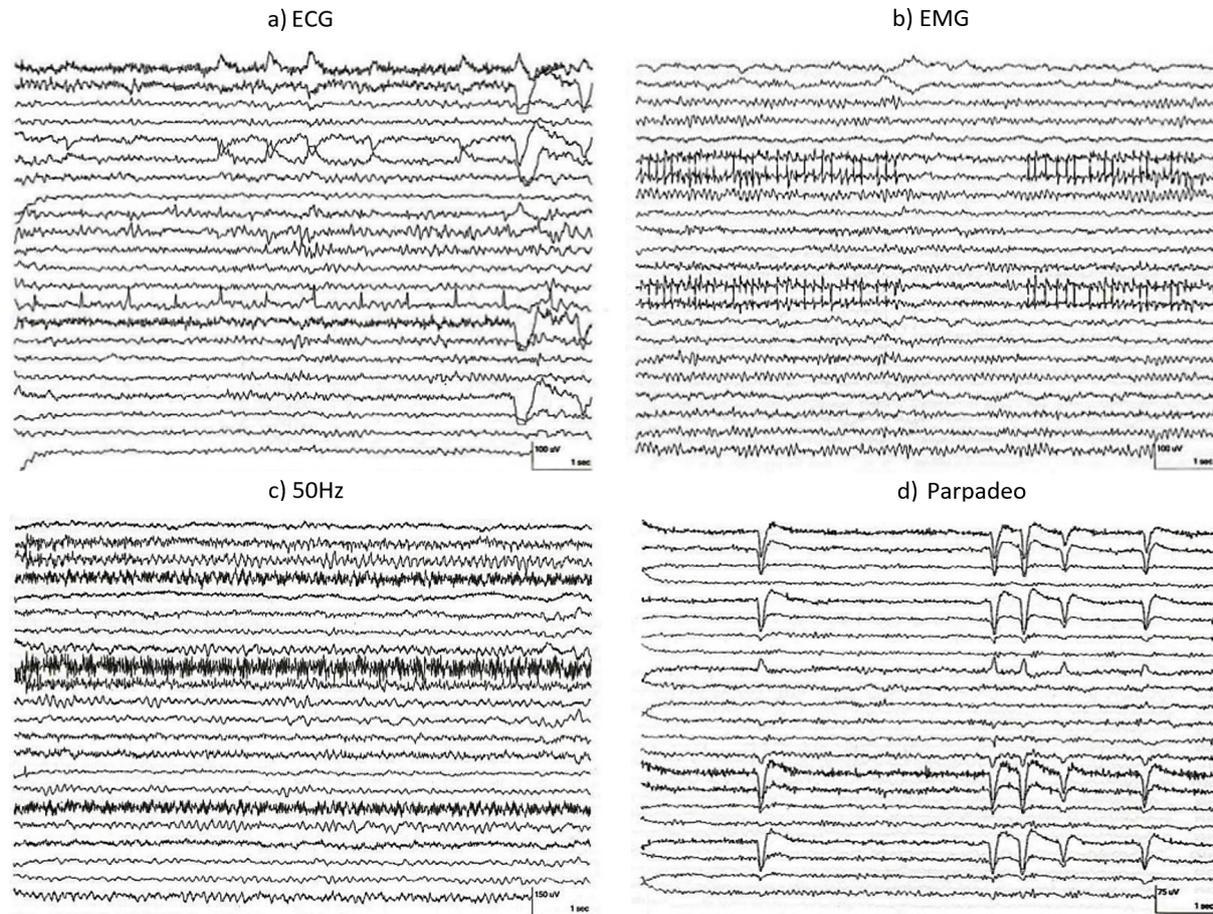


FIGURA 2.9: Señal de EEG de distintos electrodos afectada por artefactos [20].

pueden eliminarse estos artefactos.

Después de haber aplicado los métodos pertinentes, para algunos sistemas es interesante guardar los registros de la señal. Dichos registros se utilizan para realizar un análisis posterior de los datos, si se desea extraer información de ellos. En los sistemas BCI el procesado debe llevarse a cabo en tiempo real, por lo que no es conveniente que sea muy complejo ni que lleve mucho tiempo de conmutación.

2.4.2 EXTRACCIÓN DE CARACTERÍSTICAS

En la etapa de extracción de características se producen combinaciones y transformaciones sobre la señal de EEG resultante de la etapa anterior para obtener una información más discriminante sobre la intención del usuario, aumentando la SNR. Algunos investigadores consideran las técnicas de eliminación de artefactos como métodos de extracción de características, ya que para algunos casos, la salida de esos métodos es directamente la entrada del clasificador.

TABLA 2.3: Clasificación de los diferentes métodos de extracción de características según el dominio en el que trabajan, tiempo, espacio, una combinación de ambos o modelos inversivos.

Tiempo	Espacio	Tiempo-Espacio	Modelos Inversivos
Wavelets	Filtro Laplaciano	Análisis de componentes en Tiempo y Espacio	EEG o ECoG
Transformada de Fourier	Análisis de Componentes principales (PCA)		
Modelos Autorregresivos (AR)			
Filtrado Paso Banda	Análisis de componentes independientes (ICA)	Modelos AR multivariantes	EEG a dipolos fuente
Filtro Adaptado	Patrones espaciales comunes		
Filtro de Kalman	Amplitudes	Coherencia	
Detección de pico	Proporciones y diferencias		

Hay muchos métodos diferentes para extraer características de la señal. Para elegir entre todos ellos se deberá tener en cuenta que tipo de sistema BCI se quiere implementar, así como la señal de control elegida. La información de interés está oculta en un entorno altamente ruidoso, y las señales son de muchas fuentes simultáneas. Si la que nos interesa está superpuesta en tiempo y espacio a múltiples señales correspondientes a distintas tareas cerebrales, no será suficiente con aplicar un método simple como un filtro paso banda para separar la de interés [15].

Normalmente, estos métodos se pueden clasificar en dos tipos: los que trabajan en el dominio del tiempo o de la frecuencia, y los que trabajan con señales en el espacio. Habitualmente se aplica al menos un método de cada tipo, o uno que sea una combinación espacio-temporal. En la Tabla 2.3 se pueden ver qué métodos se emplean en cada tipo.

Las técnicas de filtrado espacial son útiles para mejorar las características con una distribución espacial específica, como en los sistemas BCI que usan ritmos sensoriomotores, ya que se concentran en unas regiones específicas de la corteza cerebral. El filtro Laplaciano calcula la segunda derivada de la distribución del voltaje espacial, que consigue enfatizar las contribuciones de las áreas neuronales más cercanas al electrodo de interés. Por otro lado, el método de referencia de media común (**CAR**) utiliza como referencia un electrodo situado en la oreja. Además, algunas son usadas para reducir la dimensión de los datos, eliminando información irrelevante y redundante, como el análisis de componentes principales

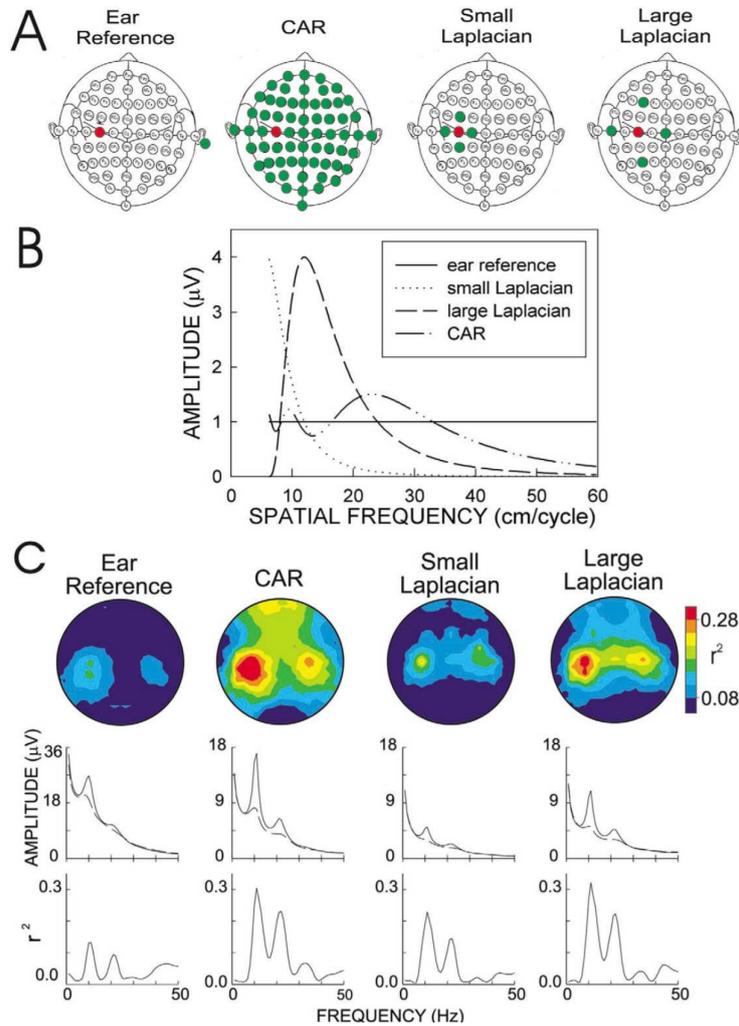


FIGURA 2.10: A) Ubicaciones de electrodos utilizados en el registro de EEG para cada filtro espacial en rojo. B) Banda de paso de cada uno de los métodos donde se muestra la raíz cuadrada del valor cuadrático medio de la señal. C) Topografía de r^2 y la amplitud espectral para cada método utilizado [16].

(**PCA**, *principal component analysis*) y el análisis de componentes independientes (**ICA**, *independent component analysis*), que realizan transformaciones lineales de los diferentes canales. Tanto el filtro Laplaciano como el CAR y el PCA tienen un coste computacional bajo, siendo adecuados para el análisis *online*, es decir, en tiempo real [15] [13]. En la Figura 2.10 se puede ver una comparación entre distintos métodos espaciales.

Las técnicas de procesamiento en el dominio del tiempo y/o la frecuencia trabajan tomando segmentos de la señal a lo largo del tiempo. Algunos métodos, como la **Transformada de Fourier** (dominio frecuencial) requieren segmentos significativamente largos para que funcionen de manera adecuada, lo que implica a su vez más tiempo computacional. Otros métodos, como el Análisis Autorregresivo (**AR**,

TABLA 2.4: Clasificación de los diferentes métodos de traducción de características.

Lineales	No Lineales			
	Estructura Fija	Estructura Modificable		
		Basados en Memoria	Combinación de no linealidades simples	Modelos generativos
Análisis discriminante lineal (LDA)	Análisis cuadrático discriminante (QDA)	Algoritmo de los vecinos más próximos	Redes neuronales (ANN)	Mezcla de Gaussianas
Máquina de soporte vectorial (SVM) lineal		SVM con Kernel	Árbol de decisión	
Perceptrón		Mínimos cuadrados parciales (PLS)	Cuantificación del vector de aprendizaje (LVQ)	Modelo oculto de Markov (HMM)
Regresión, regularización, adaptación				

auto-regressive models) pueden funcionar bien con segmentos de señal más cortos. Por todo ello, estas técnicas funcionan bien en el análisis *offline*, no obstante, para un análisis *online* se requieren respuestas rápidas con un bajo coste computacional. Por ello, los métodos en el dominio de la frecuencia no serán los más adecuados en estos casos [13].

2.4.3 CLASIFICACIÓN DE CARACTERÍSTICAS

La etapa de clasificación o traducción consiste en aplicar algoritmos con el objetivo de convertir las características calculadas en la etapa anterior en comandos de control para una aplicación BCI. Dichos comandos reflejarán de la manera más precisa y confiable posible la intención del usuario. Para que un algoritmo de clasificación sea eficaz deberá ser capaz de adaptarse a los cambios espontáneos en las características, además de alentar y facilitar la adquisición de un mejor control sobre dichas características al usuario. La clasificación determina si las características extraídas pertenecen a una intención del usuario o no; y en tal caso, a qué intención. Las posibles intenciones se entienden como clases. En la aplicación habrá dos opciones, el usuario imagina el movimiento o no, así que existen dos clases. La elección del método estará estrechamente relacionada con el número de objetivos o elecciones posibles para el sujeto [15] [13].

En los métodos de clasificación pueden establecerse dos grupos bien diferenciados: clasificadores **lineales** y **no lineales**. En la Tabla 2.4 se puede ver a qué grupo pertenecen los métodos más utilizados.

Los métodos lineales clasifican basándose en una combinación lineal de las características. Si hubiera

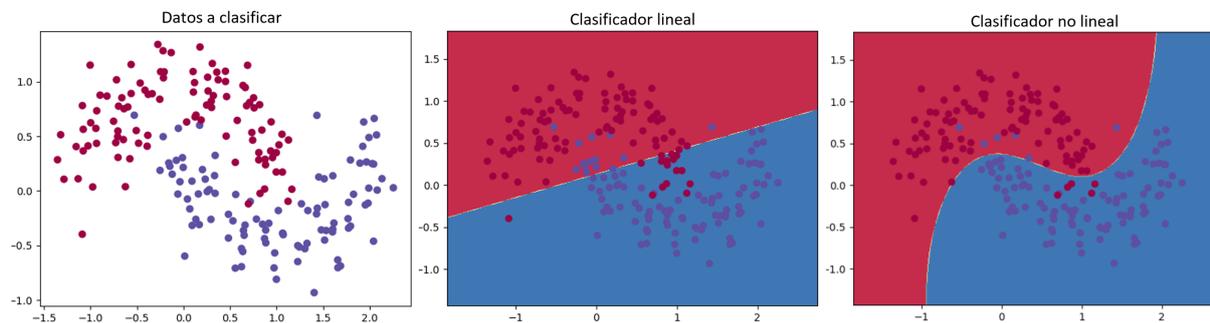


FIGURA 2.11: Aplicación de un método lineal y uno no lineal al mismo grupo de datos a clasificar [21].

n decisiones diferentes, es decir, n clases que clasificar, el método lineal las separaría mediante un hiperplano de dimensión n . Se trata de clasificadores rápidos (con poca carga computacional), y que trabajan bien con altas dimensiones de datos. Los métodos no lineales trabajan igual que los lineales, sin embargo, el hiperplano de separación tiene una dimensión mayor que los datos. En algunos casos, consiguen separar más eficazmente las clases que los lineales, con un mayor coste computacional.

En la Figura 2.11 se puede ver la diferencia entre aplicar un método lineal y uno no lineal al mismo grupo de datos. La clasificación se dificulta si las clases se encuentran muy solapadas, hay mucho ruido o se presentan *outliers* (valores atípicos que pueden aparecer en las señales).

Las señales cerebrales son intrínsecamente no estacionarias. Por ello, los patrones observados durante las sesiones de calibración pueden ser diferentes de los registrados durante la sesión *online*. Además, el entrenamiento mental progresivo de los usuarios o la atención, motivación y concentración, pueden afectar a las señales cerebrales, sin tener en cuenta que las señales de cada usuario son diferentes. Si se quiere un resultado óptimo se debería generar un clasificador diferente para cada sujeto, ya que las señales de control tendrán diferentes características (amplitud del P300, frecuencia de los ritmos sensoriomotores, etc.). Por ello, puede ser necesario aplicar algoritmos adaptativos, para que vaya actualizando el clasificador de manera continua. Normalmente, los algoritmos adaptativos dan como resultado un mejor rendimiento, sin embargo, puede ocurrir que el clasificador aprenda demasiado rápido, lo cual confunde al usuario. Además, podrían ocultar algunas características relevantes de la señal. En consecuencia, debe existir una compensación entre la adaptación y la extracción de características [15].

2.5 APLICACIONES DE LOS SISTEMAS BCI

Hay muchas aplicaciones de los sistemas BCI y dependiendo de la finalidad del sistema deseado, se escogerán los parámetros óptimos para el funcionamiento de la aplicación. Por ejemplo, si se quiere realizar una aplicación de selección de comandos, la señal de control más recomendada es el P300. En cambio, si la

finalidad es el control de un cursor bidimensional, sería más adecuado utilizar los ritmos sensoriomotores. Los sistemas BCI tienen muchos propósitos diferentes: como el control de sillas de ruedas, de prótesis, de robots, procesadores de texto, etc. Todas ellas se pueden clasificar dependiendo de su finalidad en: **comunicación, restauración motora, control del entorno, locomoción y entretenimiento** [15].

2.5.1 COMUNICACIÓN

Las aplicaciones de BCI para la comunicación van destinadas a personas con graves discapacidades en este área, provocadas por enfermedades neurológicas. Normalmente se implementa un teclado virtual que aparece en una pantalla y en el cual el usuario debe seleccionar la letra que desea escribir. Se pueden utilizar diferentes señales de control. En Birbaumer et al. [22] los SCPs se emplean para controlar un cursor hasta dejarlo sobre la letra deseada. Después de un extenso entrenamiento, se consiguió una tasa de aproximadamente 2 caracteres por minuto.

También hay estudios que utilizan los ritmos sensoriomotores. En Obermaier et al. [23] dividen todas las letras en dos subgrupos, uno a la derecha y uno a la izquierda de la pantalla. Con la imaginación del movimiento de la mano o del pie se elige el subgrupo. Una vez elegido se vuelve a dividir en dos, y así sucesivamente hasta que se escoja la letra deseada. La tasa alcanzada se encuentra entre 0,5-0,85 letras por minuto.

La señal de control más popular en este tipo de aplicaciones es el P300, debido a que no requiere de un entrenamiento exhaustivo [15]. En Farwell *et al.* [24] se muestran en una matriz 6x6 las 26 letras del alfabeto junto con otros comandos y símbolos. Las filas y las columnas se iluminan aleatoriamente y el sujeto debe fijar su atención en la letra deseada. Cuando ésta se ilumine aparecerá de forma espontánea un P300 unos 300 ms después de la iluminación de dicha letra. Las tasas de transferencia obtenidas son de aproximadamente 2 letras por minuto. Algunos ejemplos de estas aplicaciones se pueden ver en la Figura 2.12.

Para las personas que han perdido la capacidad de fijar la mirada se utilizan estímulos auditivos en vez de visuales. También se han realizado navegadores web controlados por estas señales, donde destacan los controlados por P300 y SCPs [15].

2.5.2 RESTAURACIÓN MOTORA

Las aplicaciones de BCI para la restauración motora se destinan a personas que sufren una lesión de la médula espinal u otras enfermedades neurológicas que han provocado la pérdida de funciones sensoriales y motoras, reduciendo de manera drástica su calidad de vida.

La restauración del movimiento de agarre, por ejemplo, es factible a través de neuroprótesis guiadas por estimulación eléctrica funcional (FES, *functional electrical stimulation*). FES provoca contracciones

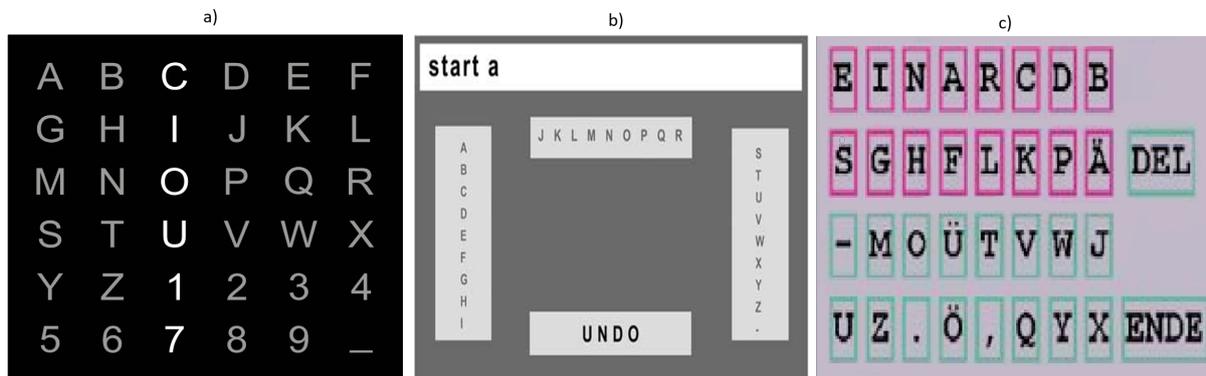


FIGURA 2.12: Tres ejemplos de Spellers con diferentes señales de control: a) P300 [24], b)SMR con 4 posibilidades [25], c) SCP con elecciones binarias [26].

musculares artificiales para compensar la pérdida de funciones voluntarias. Dichas contracciones son provocadas por un potencial de acción artificial que despolariza los nervios motores periféricos intactos [27]. El sistema BCI es el encargado de manejar las FES mediante las señales de control escogidas.

En Pfurtscheller et al. [28] los sujetos consiguen controlar sus manos paralizadas para agarrar objetos mediante la imaginación del movimiento, es decir, mediante ritmos sensoriomotores que controlan el dispositivo FES. Los FES también se han utilizado como método de rehabilitación en pacientes que han sufrido accidentes cerebrovasculares. No obstante los FES no son posibles en todos los pacientes, ya que requieren el uso de movimientos residuales que los pacientes más afectados no son capaces de generar. Por ello, surgieron sistemas BCI para el manejo de neuroprótesis sin el uso de FES, también controlados por los ritmos sensoriomotores [15]. Como ya se ha dicho, cuando se utilizan ritmos sensoriomotores se requiere un largo periodo de entrenamiento.

Asimismo, también se desarrollaron neuroprótesis controladas por un sistema BCI basado en VEP [29]. Se ofrecían distintas opciones, por ejemplo, movimiento de supinación o pronación, abrir y cerrar la mano etc. Alguno de estos ejemplos se puede ver en la Figura 2.13.

2.5.3 CONTROL DEL ENTORNO

Los sistemas BCI para el control del entorno pretenden dar a las personas con discapacidad la mayor independencia posible. Como muchos de los sujetos no pueden salir de casa, estas aplicaciones se enfocan en plataformas que permitan el control de dispositivos domésticos como la televisión, el termostato, abrir la puerta, control de las luces, etc. [31]. En este campo se han desarrollado diferentes aplicaciones basadas en distintas señales de control. Cabría destacar las que utilizan los ritmos sensoriomotores y P300. Ejemplos de estas aplicaciones están representados en la Figura 2.14 [15].

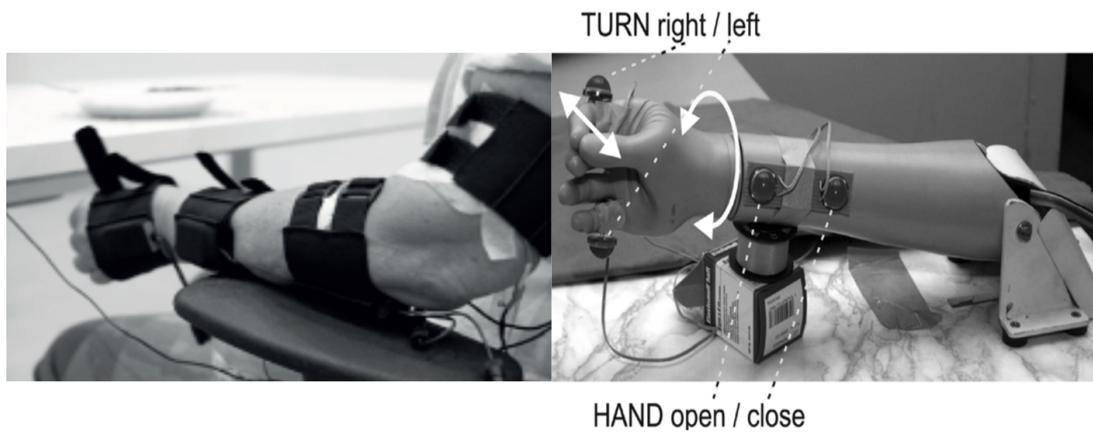


FIGURA 2.13: En la imagen de la derecha se puede ver una neuroprótesis para un sujeto que ha perdido el brazo [29] y en la imagen de la izquierda se utiliza una combinación de BCI y FES como método de rehabilitación [30].



FIGURA 2.14: Matriz para control domótico mediante P300 [31].

2.5.4 LOCOMOCIÓN

Las aplicaciones BCI de locomoción permiten a las personas con discapacidad controlar un medio de transporte para darles más autonomía. Como se puede deducir, la portabilidad es una característica necesaria en estos sistemas, además de tener como objetivo alcanzar el control a tiempo real con una gran precisión. Tanaka *et al.* [32] presentó la primera aplicación para controlar una silla de ruedas mediante EEG. La señal de control escogida fueron los ritmos sensoriomotores. En este estudio se presentaba el suelo dividido en cuadrados. El usuario decidía por qué cuadrados moverse mediante la imaginación del movimiento de las extremidades.

A lo largo de los años se han ido mejorando para intentar aumentar las precisiones debido a los peligros que conlleva una mala elección. Los P300 se han comenzado a emplear como señal de control para aumentar la precisión y superar la tasa de transferencia. Se ha dotado a los sistemas de cierta autonomía para reducir la cantidad de interacciones necesarias [15], por ejemplo, creando rutas mediante guías donde el usuario solo debe escoger qué ruta desea seguir. La silla sigue dicha ruta hasta que el sujeto



FIGURA 2.15: Control de una silla de ruedas mediante EEG [35].

decida parar [33]. Después surgieron sistemas que utilizaban tanto P300 como ritmos sensoriomotores para mejorar la rapidez en el control [34]. Algunos usuarios que no son capaces de manejarlos con gran precisión, requieren de ayuda y supervisión por su seguridad. Además, estos sistemas no son capaces de manejar escenarios desconocidos y poblados [15]. En la Figura 2.15 se puede ver qué se necesita para manejar una silla mediante EEG.

2.5.5 ENTRETENIMIENTO

Las aplicaciones BCI orientadas al entretenimiento suelen ser menos prioritarias en la investigación de estos sistemas ya que normalmente se centran en aplicaciones de asistencia. Sin embargo, en los últimos años ha crecido el interés en sistemas BCI para el entretenimiento debido al gran crecimiento que ha sufrido esta tecnología. Los sistemas BCI pueden conseguir que los videojuegos sean una experiencia más atractiva y desafiante, además de proporcionar información a los creadores de las experiencias del usuario para poder mejorar en el futuro [15]. Se han adaptado juegos muy conocidos, como el Pacman, para poder jugar mediante la imaginación motora [36]. También se pueden utilizar los VEPs y el P300 como método de control de juegos. Muchas empresas importantes, como Microsoft, están investigando sobre sistemas BCI para juegos [15]. En la Figura 2.16 hay algunos ejemplos de juegos controlados mediante BCI.

2.5.6 OTRAS APLICACIONES

La capacidad de los sistemas BCI para producir una retroalimentación (*feedback*) al usuario puede ser la base para multitud de aplicaciones médicas, ya que los sujetos pueden adquirir un control selectivo de ciertas áreas del cerebro, mediante *neurofeedback*. Esto puede conseguir mejorar el rendimiento cognitivo [37], las habilidades del habla [38], el afecto [39] y el control del dolor. Se ha utilizado en el tratamiento de trastornos mentales como la epilepsia [40], el déficit de atención [41], la esquizofrenia, la depresión, el alcoholismo o la pedofilia [15].



FIGURA 2.16: Videojuegos controlados mediante EEG.

También ha surgido el *neuromarketing*, que aplica métodos neurocientíficos a la investigación de mercado, para probar un producto antes de su lanzamiento, o para proporcionar información sobre las preferencias subyacentes de los usuarios [42]. La neuroimagen, puede revelar información oculta sobre las verdaderas preferencias de los usuarios [15].

2.6 NEUROFEEDBACK TRAINING BASADOS EN BCI

Después de haber dado una visión general de las aplicaciones más importantes de los sistemas BCI, en este apartado se desarrolla una revisión del estado del arte sobre sistemas BCI para entrenamiento cognitivo con retroalimentación (**NFT**). Las características de la aplicación desarrollada en este trabajo tienen relación con los resultados obtenidos en los diferentes estudios.

2.6.1 GOMEZ-PILAR ET AL - NEUROFEEDBACK TRAINING WITH A MOTOR IMAGERY-BASED BCI: NEUROCOGNITIVE IMPROVEMENTS AND EEG CHANGES IN THE ELDERLY [37]

En este estudio se ha implementado una aplicación BCI controlada mediante ritmos sensoriomotores que proporciona al usuario una realimentación. La finalidad del estudio consistía en constatar si los sistemas BCI sirven, no solo como herramienta de asistencia para personas con discapacidades, sino también como método de rehabilitación o restauración de funciones motoras o cognitivas en personas mayores [37].

Como utiliza ritmos sensoriomotores, el registro de EEG se ha llevado a cabo con electrodos cercanos a la región motora del cerebro y son: F3, F4, T7, C3, Cz, C4, T8 y Pz. Se ha utilizado como referencia común un electrodo colocado en el lóbulo de la oreja. Las señales pasan por un amplificador *g.USBamp* y se digitalizan a 256 Hz. Durante el NFT las señales se procesan en tiempo real con BCI2000, aplicando

un filtro pasobanda de 0,1-60 Hz. Para extraer la señal de control se aplica un filtro Laplaciano sobre los electrodos C3, C4 y Cz. Las bandas espectrales eran de 3 Hz y estaban centradas en 12, 18 y 21 Hz [37].

La aplicación consta de cinco tareas diferentes. La primera consiste en aprender a imaginar el movimiento de las manos. Cuando aparece la imagen de una puerta cerrada, se debe imaginar el movimiento de la mano derecha, en cambio, si aparece una ventana será el movimiento de la mano izquierda. Si lo realiza correctamente, la puerta o ventana se abrirá (*feedback*). En la segunda aparece un objeto en el centro de la pantalla y el objetivo en uno de los dos extremos. El usuario debe imaginarse el movimiento de la mano correspondiente al lado en el que se encuentre el objetivo. Conforme lo haga, el objeto en el centro se moverá horizontalmente (*feedback*). La tercera es como la segunda, sin embargo, aparece un objetivo a cada lado, siendo el usuario el que tiene que elegir cuál de los dos es el correcto, dependiendo del objeto central. En la cuarta, aparece un personaje en el centro de un camino que avanza hacia adelante. En el camino irán surgiendo obstáculos que el usuario deberá sortear mediante SMR. El personaje se irá desplazando a izquierda o derecha (*feedback*). La última tarea combina ejercicios de memoria con imaginación motora. Primero aparecen dos imágenes de formas, después desaparecen y surge una imagen a cada lado de la pantalla, pero solo una de ellas coincide con una de las iniciales. El usuario debe decidir qué imagen se repite mediante la imaginación motora del lado correspondiente [37]. Estas tareas se pueden ver en la Figura 2.17.

Los experimentos se realizaron en 63 sujetos, divididos en el grupo de control (32 sujetos) y el grupo de NFT (31 sujetos). Primero se les llevo a cabo un test llamado Luria-AND para ver las características neuropsicológicas a ambos grupos. Después, el grupo de NFT realizó 5 sesiones de NFT donde efectuaban las tareas anteriormente descritas. A continuación, volvieron a realizar el test para ver las diferencias en sus funciones neuropsicológicas. Por último, se realizó un análisis *offline* para evaluar la influencia del NFT. Los resultados indicaron que hubo diferencias significativas entre los dos grupos. En el grupo que realizó las tareas se apreció un aumento en la percepción visual, memoria inmediata, habla receptiva y dibujos temáticos, mientras que en el grupo control no se apreciaron cambios significativos. Se concluyó, por tanto, que la aplicación de la herramienta NFT favoreció el aumento de las funciones cognitivas de los usuarios [37].

2.6.2 HWANG ET AL. - NEUROFEEDBACK-BASED MOTOR IMAGERY TRAINING FOR BRAIN-COMPUTER INTERFACE (BCI) [43].

En este estudio se propone un sistema de NFT basado en la imaginación de movimiento con EEG, cuyo objetivo es ayudar a las personas a obtener fácilmente la sensación de la imaginación motora [43].

Los electrodos para el registro del EEG fueron colocados de acuerdo al sistema internacional 10-20. Dependiendo qué etapa del experimento se quisiera registrar, se utilizaron unos u otros. En la etapa de

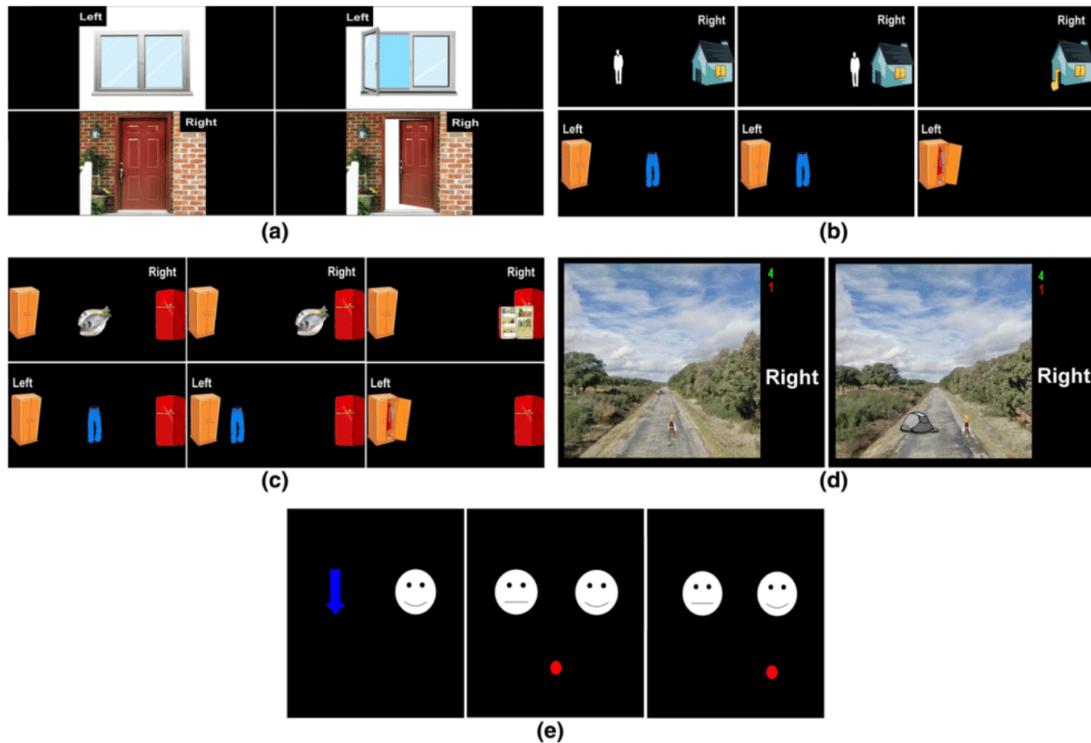


FIGURA 2.17: Representación gráfica de las cinco tareas ha realizar en el experimento [37].

NFT se usaron 16 electrodos (AF3, FC3, C3, CP3, PO3, FCz, Cz, CPz, AF4, FC4, C4, CP4, PO4, T7, T8 y Oz) y en las etapas de registro se utilizaron 15 electrodos (Cz, C1, C2 C3, C4, CPz, CP1, CP2, CP3, CP4, FCz, FC1, FC2, FC3 y FC4) mediante un sistema de adquisición multicanal (WEEG-32). Las señales se registraron a 256 Hz. Además, se adquirieron EMG y EOG para la eliminación de artefactos [43].

Para la realización del experimento contaron con 10 voluntarios sanos, sin antecedentes psiquiátricos, neurológicos u otras enfermedades que pudieran influir en los resultados. Los sujetos se dividieron en dos grupos de 5: el grupo de los controles y el grupo de entrenamiento. En las etapas de registro se colocó delante de los sujetos una pantalla. Los usuarios se encontraban sentados y relajados. Llevaron a cabo el paradigma experimental explicado en la Figura 2.18. Al comienzo, se presentaba la pantalla en gris durante 3s. A continuación aparecía un círculo con un patrón de ajedrez blanco y negro a uno de los dos lados de la pantalla aleatoriamente, durante 0,25s. Su función es indicar al sujeto el movimiento de qué mano (derecha o izquierda) deberá imaginarse. Después, la pantalla se volvía gris de nuevo durante 1s. Seguidamente surgía una X en el centro de la pantalla durante 0,25s, indicándole al sujeto que debía imaginarse el movimiento de la mano correspondiente. Este proceso se repitió 180 veces en cada registro. En la etapa de NFT los usuarios se colocaron sentados y relajados delante de una pantalla. En ella visualizaron a tiempo real la actividad de sus ritmos corticales mientras se imaginan el movimiento de

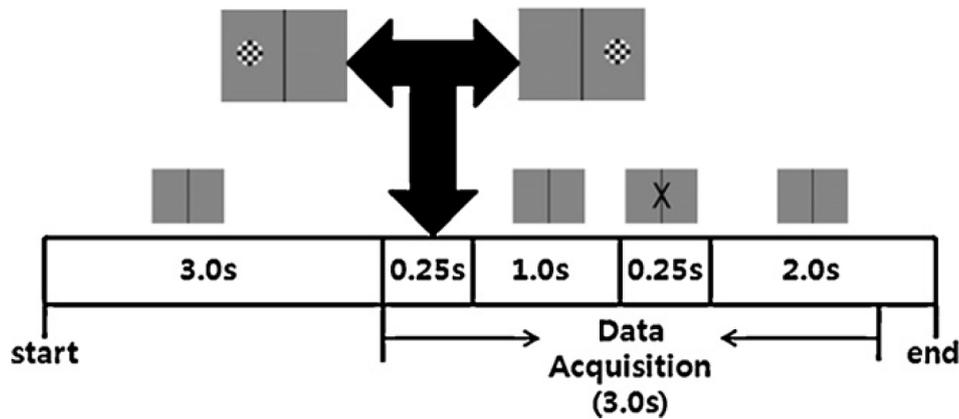


FIGURA 2.18: Representación gráfica la secuencia que sigue el paradigma experimental [43].

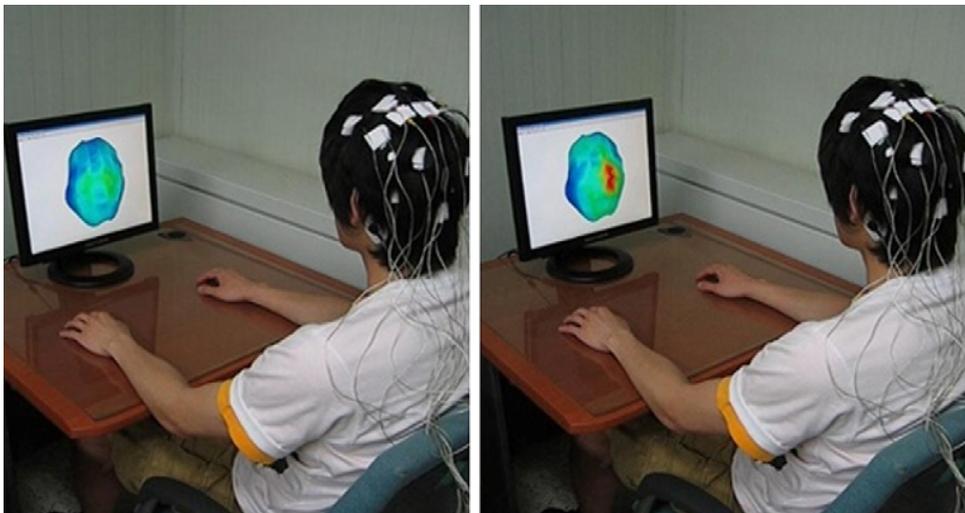


FIGURA 2.19: Feedback que se le muestra al sujeto durante el entrenamiento [43].

su mano izquierda o derecha, como se aprecia en la Figura 2.19. Se les instaba a que fueran aumentando su actividad en la región correspondiente a lo largo del entrenamiento, pudiendo ver en la pantalla el resultado de sus esfuerzos como regiones marcadas con diferentes colores [43].

A los 5 controles se les realizó dos etapas de registro, separadas 30 minutos entre sí sin ningún tipo de NFT. En cambio, al grupo de entrenamiento primero se les realizó un registro. Después llevaron a cabo el entrenamiento de imaginación motora establecido. A continuación, se les volvió a realizar un registro para estudiar los cambios producidos gracias al entrenamiento. Una vez recogidos todos los datos, se desarrollaron mapas en tiempo-frecuencia, seleccionando los electrodos más destacados de las dos etapas de registro. En dichos mapas se pudo ver cómo en el grupo de control no hubo diferencias significativas entre los dos periodos de registro. En cambio, en el grupo de entrenamiento se descubrieron en todos

los integrantes diferencias significativas en las regiones donde se producen los ritmos sensoriomotores. La precisión promedio de clasificación en el grupo de entrenamiento fue de 71,4 %. La precisión no es muy alta, no obstante el objetivo del estudio no era clasificar, sino demostrar la utilidad de estos sistemas para ayudar a personas con discapacidades motoras [43].

2.6.3 SHINDO ET AL. - EFFECTS OF NEUROFEEDBACK TRAINING WITH AN ELECTROENCEPHALOGRAPH BASED BRAIN-COMPUTER INTERFACE FOR HAND PARALYSIS IN PATIENTS WITH CHRONIC STROKE: A PRELIMINARY CASE SERIES STUDY [17].

El objetivo de este estudio es explorar la efectividad de aplicar NFT a personas que sufren parálisis de la mano después de un accidente cerebrovascular (CVA, *cerebrovascular accident*), mediante un sistema basado en BCI [44].

La señal utilizada fue el EEG. Como el sistema estaba basado en ritmos sensoriomotores, los electrodos principales eran C3 y C4. También se usaron los más próximos a ellos, además del Fz como tierra y un electrodo situado en el lóbulo de la oreja como referencia, tal y como se muestra en la 2.20. Esto suma un total de 10 electrodos situados según el *sistema internacional 10-20*. La señal se filtró entre 2-100 Hz antes de su digitalización a 256 Hz. El amplificador que se empleó era de la empresa *G-Tec*. Además, se registró también el EMG para identificar la actividad cuando intentaban extender los dedos de la mano afectada [44].

Para la realización del estudio contaron con la participación de ocho pacientes que habían sufrido un CVA hace más de 180 días, con un promedio de 59 años de edad. Su primer CVA fue unilateral con capacidad para caminar de forma independiente con o sin ayuda, sin déficits cognitivos severos ni convulsiones, ni marcapasos. Ninguno usaba anticonvulsivos, inhibidores selectivos de la recaptación de serotonina y no padecían otras enfermedades graves. Seis de los ocho tenían poca o ninguna actividad EMG de superficie destacable del extensor afectado al intentar extender sus dedos. Cada sesión de 1 h de entrenamiento consistía en tres etapas. Se realizaba una o dos veces por semana durante 4-7 meses, por lo que cada paciente realizó entre 12-20 entrenamientos. La primera etapa consistía en el movimiento del cursor de izquierda a derecha durante 8s, pero la señal de la tarea se presenta 5s después de que aparezca el cursor. Los participantes debían realizar la apertura de la mano afectada, o de la mano no afectada durante 3s. Se desarrollaban 10 ensayos, 5 de cada mano. A continuación, se llevaba a cabo una etapa de calibración para ajustar los parámetros del clasificador. En la pantalla se presentaba una flecha en el centro, apuntando a la izquierda o a la derecha, y el sujeto debía imaginarse el movimiento de la mano correspondiente. En ausencia de flechas se realizaba un descanso durante 1 s. Se llevaban a cabo 40 ensayos. Una vez realizada la calibración, los pacientes entrenaban con retroalimentación, que consistía en imaginarse la apertura de la mano parética (paralizada parcialmente o con debilitamiento de la contractilidad muscular) o dejarla en reposo durante 5s. La altura del cursor reflejaba el valor acumulado de

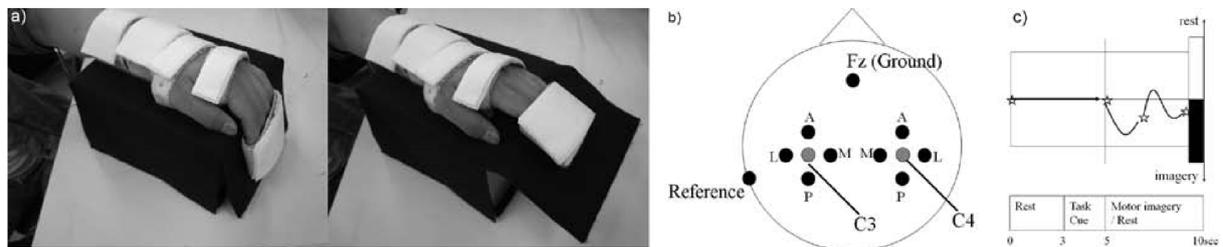


FIGURA 2.20: a) Ortesis motorizada. La imagen de la izquierda es la posición neutral y la de la derecha es la posición de máxima extensión. Normalmente tarda 3s en pasar de una a otra. b) Posición de los electrodos utilizados para el registro del EEG según el sistema internacional 10-20 y 2,5 cm anterior (A), posterior (P), medio (M) y lateral (L) a C3 y C4. c) *Feedback* visual del cursor moviéndose hacia abajo (imaginación del movimiento) y hacia arriba (resto) [44].

la salida del clasificador cada 30 ms, bajando si se conseguían disminuir los SMR claramente. Después de algunas sesiones de entrenamiento, se incluyó una ortesis motorizada para activarla cuando el cursor alcanzara la mitad inferior extendido los dedos paralizados de 90 a 50 grados [44].

Dos de los pacientes mostraron una mejora en la paresia del brazo. Además, dos sujetos presentaron una disminución de la espasticidad (contracción permanente de los músculos) de los dedos paréticos y cuatro mostraron una mejora en la función de los dedos. Gracias a la participación en el estudio, todos los pacientes indicaron que se volvieron más conscientes del uso de su extremidad parética en las actividades diarias y sintieron que podían relajarlos más fácilmente. En cuanto a cambios neurofisiológicos, cuatro sujetos mejoraron su actividad voluntaria de EMG, a pesar de que tenían poca o ninguna antes del entrenamiento. Ésto indicó que todos los participantes habían mejorado la funcionalidad de sus dedos, así como, intensificado la actividad voluntaria de EMG superficial de los extensores de la extremidad afectada. También cabe indicar que los cambios en la función motora parecían estar relacionados con la severidad del deterioro motor, más que con la intensidad y duración del entrenamiento [44].

2.6.4 GEVENSLEBEN ET AL. - IS NEUROFEEDBACK AN EFFICACIOUS TREATMENT FOR ADHD? A RANDOMISED CONTROLLED CLINICAL TRIAL [18].

Este estudio tiene como objetivo la búsqueda de un tratamiento adicional o alternativo para el trastorno de déficit de atención e hiperactividad (ADHD, *attention deficit hiperactivity disorder*) mediante el NFT, utilizando dos protocolos de entrenamiento: potenciales corticales lentos (SCPs) y ritmos μ y β [41].

El registro del EEG durante el entrenamiento se realizó tomando Cz como electrodo principal, con referencia los mastoides. Se empleó un ancho de banda de 1-30Hz para μ y β y 0,1-30 Hz para los SCPs, ambos con una frecuencia de muestreo de 250 Hz. Se colocaron electrodos encima y debajo del ojo para eliminar artefactos [41].

En el experimento participaron 102 niños con ADHD de entre 8 y 12 años. Eran pacientes que no necesitaban medicación de forma inmediata. Se dividieron en dos grupos aleatoriamente: el grupo de NFT (59 niños) y el de AST (35 niños). Se excluyeron a los 8 restantes por diversas razones. Entre ambos grupos no había diferencias concernientes al pre-entrenamiento, variables psicológicas y clínicas. Fueron excluidos los niños con trastornos comórbidos (trastornos asociados al ADHD) que no fueran el trastorno de la conducta, trastornos emocionales, tic nervioso y dislexia. A los participantes no se les dio ningún tipo de medicación ni psicoterapia al menos 6 semanas antes del entrenamiento. Ambos tipos de entrenamiento consistieron en dos bloques de 18 sesiones divididos en 9 sesiones dobles, donde cada sesión duraba aproximadamente 50 minutos. A cada sujeto se le realizaba una evaluación previa, otra intermedia y una posterior al entrenamiento. Los entrenamientos los realizaban en parejas dentro del mismo grupo. En las evaluaciones de las capacidades previas realizaban juegos de ordenador que requerían atención, desarrollo y práctica de estrategias adquiridas, etc. [41].

Durante el entrenamiento, los niños del grupo NFT debían sentarse delante de un monitor. En el entrenamiento mediante SCPs, la tarea consistía en encontrar la estrategia adecuada para mover un balón hacia arriba o hacia abajo. En el entrenamiento de los SMR se presentaba una barra a la izquierda de la pantalla, que representaba la actividad μ , y debían conseguir reducirla. También aparecía una barra a la derecha que representaba la actividad β y debían aumentarla. Cada prueba duraba 8s (2s de referencia y 6s de retroalimentación). El grupo de AST realizaba una terapia cognitiva para trabajar en enfocar la atención, el control de la impulsividad y el procesamiento cuidadoso de tareas para resolver un juego de ordenador [41].

Para finalizar, se llevo a cabo un puntaje de las pruebas para intentar determinar los niveles de atención, hiperactividad e impulsividad. En ambos protocolos de entrenamiento los padres observaron mejoras en las capacidades de los niños, que se corroboraban con el puntaje obtenido. Tanto las clasificaciones de los padres, como de los profesores, revelaron una superioridad en el entrenamiento de NFT para la disminución de la sintomatología del ADHD. Lo cuál demostró que los efectos del NFT son sustanciales y de importancia práctica. Los efectos positivos no solo afectaron a los síntomas sino también a los problemas de adaptación social [41].

2.6.5 COMPARATIVA DE LOS DIFERENTES ESTUDIOS

A lo largo del tiempo se ha podido observar que la señal de control más utilizada en este tipo de sistemas son los ritmos sensoriomotores. En todos los estudios se realizan numerosas sesiones de entrenamiento para poder observar algún cambio, debido a que ese tipo de señales requieren práctica para poder controlarlas. Sin embargo, gracias a la retroalimentación el proceso de aprendizaje se reduce y facilita.

En los estudios que se han tratado en el apartado anterior se puede ver que el *neurofeedback* se utiliza

como una herramienta para intentar mejorar las capacidades de los usuarios. En Gomez-Pilar et al. [37] y en Hwang et al. [43] se pretende mejorar las capacidades cognitivas de sujetos sanos. En cambio, en los estudios Shindo et al. [44] y Gevensleben et al. [41] se utiliza para intentar mejorar la calidad de vida de personas que sufren alguna enfermedad. En todos los estudios, las señales se adquieren mediante el EEG, siguiendo el sistema internacional 10-20 para la colocación de los electrodos, sin embargo no se usan las mismas localizaciones. Además, en todos se ha podido determinar que el *neurofeedback* produce cambios en la actividad cerebral después de varios entrenamientos, consiguiendo demostrar la efectividad del mismo para mejorar las capacidades cognitivas, o como tratamiento o rehabilitación de algunas enfermedades.

Hay muchos otros estudios sobre NFT, enfocados a diferentes objetivos que emplean diversos métodos. En Sterman *et al.* [45] se realiza una revisión de estudios que intentan demostrar la utilidad del *neurofeedback* en sistemas BCI con señales de control como los ritmos sensoriomotores y los SCPs. Su objetivo es el tratamiento de los efectos de la epilepsia, como una alternativa a los tratamientos farmacológicos de dicha enfermedad. En Weiskopf *et al.* [46] se estudia el efecto de un sistema BCI con *neurofeedback* visual, basado en fMRI, tanto en sujetos sanos como en enfermos. Los resultados mostraron que todos los participantes obtuvieron diferencias significativas comparando el control de sus señales antes y después del *neurofeedback*. En Kober *et al.* [47] se implementó un sistema de *neurofeedback* basado en NIRS, para ver su utilidad a la hora de mejorar la activación cortical. A un grupo se le daba el *feedback* real y a otro, el *feedback* de otro usuario. Se determinó la utilidad de este sistema para el entrenamiento de pacientes con lesiones cerebrales focales al aumentar la actividad de áreas cerebrales específicas con fines de rehabilitación.

Dada la utilidad del NFT para rehabilitar o mejorar funciones cognitivas, el presente Trabajo Fin de Grado se ha centrado en desarrollar una herramienta que permita realizar tareas NFT controladas mediante SMR en un dispositivo portable. De esta manera, se contribuye al desarrollo de este tipo de sistemas facilitando a los usuarios un método adecuado para entrenarse cognitivamente.

ANÁLISIS DE LA SEÑAL Y RITMOS SENSORIOMOTORES

3.1 RITMOS SENSORIOMOTORES (RITMOS μ Y β)

Una propiedad fundamental de cualquier red neuronal es la capacidad de las neuronas para trabajar en sincronía y generar actividad oscilatoria. Gran parte de la actividad oscilatoria cerebral, se concentra sobre las áreas corticales sensoriomotoras primarias, y normalmente muestran ritmos en la banda μ (8-12 Hz) y en la banda β (18-26 Hz). Estas actividades se conocen como ritmos sensoriomotores y se distinguen entre sí por ubicación, frecuencia y/o relación con alguna entrada sensorial o salida motora. Algunos de los ritmos β son armónicos de los ritmos μ , mientras que otros son independientes por su topografía y/o temporización. Por lo tanto, se toman como características independientes del EEG. Los SMR son elegidos como señal de control en numerosos sistemas BCI. Están asociados con las áreas corticales más directamente conectadas a las salidas neuromusculares normales del cerebro [8].

El movimiento o la preparación del movimiento suele ir acompañado de una disminución de los SMR, específicamente contra-lateral al mismo. Esta disminución se ha denominado desincronización relacionada con eventos (**ERD**). Su opuesto, es decir, el aumento de los SMR o sincronización relacionada con eventos (**ERS**), ocurre después del movimiento y con la relajación. Son de mucha utilidad en sistemas BCI debido a que ambos, ERD y ERS, aparecen también con la imaginación del movimiento [9]. En la banda μ se pueden distinguir dos patrones. Una desincronización en la banda μ menor (8-10 Hz) producida durante casi cualquier tipo de tarea motora que se extiende por toda la corteza sensoriomotora. Y una desincronización de la banda μ superior (10-12 Hz) restringida topográficamente y relacionada con aspectos más específicos de la tarea [8]. Por ejemplo, en la banda μ inferior se presenta un ERD similar para el movimiento de la mano o del pie. En cambio, en la banda superior aparece un ERD durante el movimiento de la mano y un ERS durante el movimiento del pie. Por ello, para el entrenamiento en sistemas BCI se pueden desarrollar dos patrones: i) una desincronización contralateral al movimiento

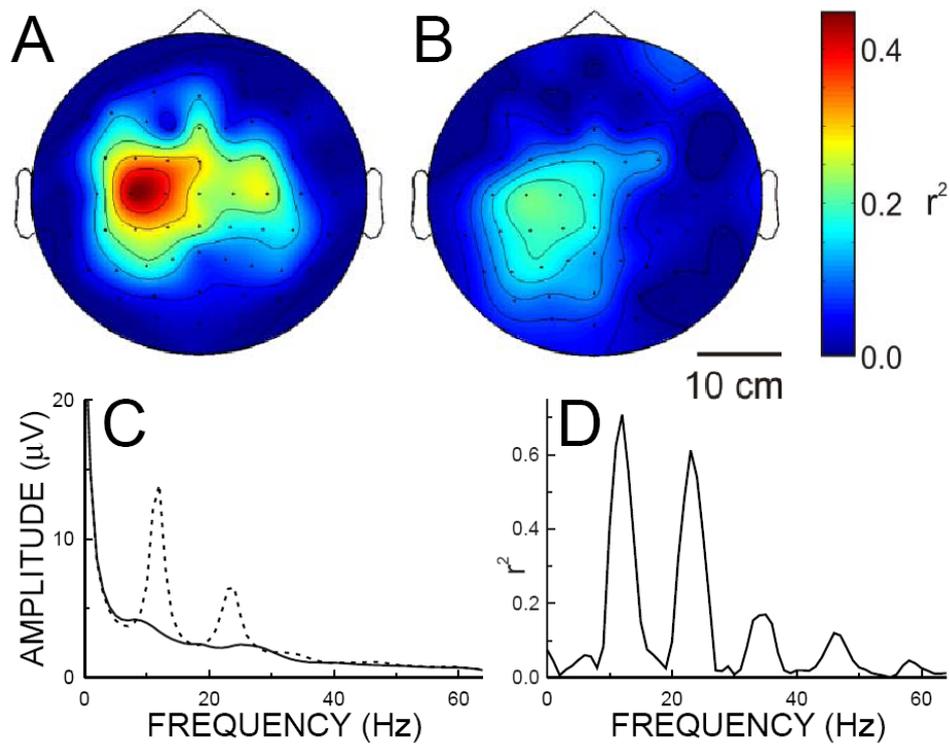


FIGURA 3.1: (A),(B) Distribución topográfica del estadístico r^2 calculada para (A) la imaginación del movimiento de la mano derecha y (B) movimiento de la mano derecha vs. resto para un ancho de banda de 3Hz centrado en 12 Hz. (C) Espectros del voltaje sobre C3 para comparar el descanso (línea punteada) y MI (línea continua). (D) Espectro r^2 para MI vs. descanso [48].

en la banda μ superior, junto con un ERS en el lado ipsilateral; y ii) un ERD de la banda μ inferior bilateralmente simétrico [8].

A lo largo de los años, se ha demostrado en diversos estudios cómo las personas pueden aprender a controlar las amplitudes de los ritmos μ ó β en ausencia del movimiento o sensación del movimiento [8]. En la Figura 3.1 se puede ver la diferencia entre la imaginación del movimiento y el reposo. La actividad inducida mediante MI de la mano derecha es más prominente en la región centrada en el electrodo C3. Mientras que con la MI de la mano izquierda se centra en el C4. Mediante el EEG no se puede distinguir entre pie derecho e izquierdo y la actividad se centra en el electrodo Cz [48].

Normalmente, los sujetos después de un entrenamiento BCI con ritmos sensoriomotores son capaces de cambiar la amplitud de los SMR para mover un cursor en diferentes dimensiones. Lo más simple y más ampliamente utilizado es la imaginación del movimiento de las manos para mover el cursor en una dimensión. Sin embargo, también es posible utilizar la MI de los pies y la lengua, aunque suele resultar más difícil para los usuarios. Después del periodo de aprendizaje a base de entrenamiento, las imágenes

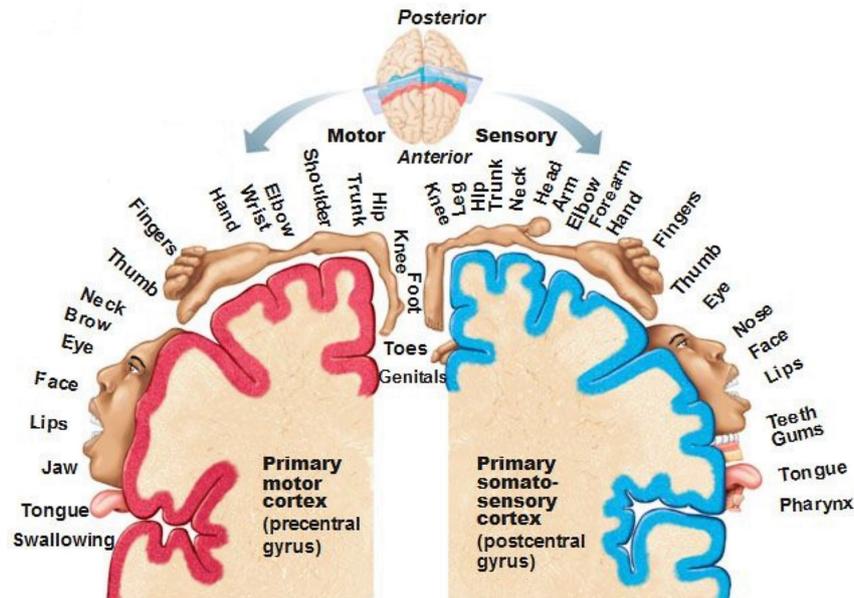


FIGURA 3.2: Representación visual de las divisiones anatómicas de la corteza motora primaria y la corteza somatosensorial primaria [49].

motoras van perdiendo importancia y los sujetos aprenden a mover el cursor automáticamente, realizando un acto motor convencional [8]. Esto ocurre a través de un proceso denominado condicionamiento operante, por el que las personas aprenden a asociar una determinada acción con una respuesta o efecto. En los sistemas BCI se utiliza el *neurofeedback* para indicar al usuario si está realizando bien o mal la tarea deseada y así poder aprender. En la figura 3.2 están señaladas las regiones cerebrales donde se producen los ritmos asociados a los movimientos de las diferentes partes del cuerpo.

Los ritmos sensoriomotores se encuentran mezclados con la actividad EEG de fondo y distintas fuentes de ruido. Por lo tanto, es necesario realizar un procesamiento de la señal de EEG para poder distinguir estos ritmos de la mejor manera posible. A continuación, se detallarán los diferentes métodos de procesamiento de señal de los SMR, destacando los más utilizados en los estudios de sistemas BCI.

3.2 FACTORES QUE INFLUYEN EN EL CONTROL DE LOS SMR

En las diversas investigaciones realizadas se han dado casos de personas incapaces de aprender a controlar los sistemas BCI. En el caso de los SMR se ha visto que hay dos predictores psicológicos que influyen en el rendimiento de SMR-BCI [50]:

- La coordinación viso-motora: mide la precisión del sujeto en las habilidades motrices finas (coordinación en los movimientos de los músculos pequeños del cuerpo, por ejemplo, de los dedos de las

manos y los ojos), y en el procesamiento de la información. En Hammer *et al.* [50] se demostró que la coordinación visio-motora representó el 11 % de la variabilidad en el rendimiento de BCI. La influencia de las habilidades motrices finas en el rendimiento de BCIs basados en MI apoya la idea de que el aprendizaje mediante *neurofeedback* es similar al aprendizaje motor, debido a que estas habilidades comparten propiedades similares. En primer lugar, la modulación de SMR está muy relacionada con la ejecución, la imaginación y la observación del movimiento preciso. En segundo lugar, ambos están influenciados de manera similar por *feedback*, práctica o entrenamiento y el sueño [50].

- La capacidad de concentración en las tareas: se ha demostrado que representa casi el 19 % de la varianza en el rendimiento de SMR-BCI. Poder concentrarse en la tarea es vital para el buen manejo de estos sistemas, además de ayudar a ignorar los estímulos que producen distracción. Se cree que realizar una tarea de BCI requiere de capacidades de autorregulación para concentrarse, y así cumplir con el objetivo, a pesar de las distracciones que se presenten y los pensamientos del propio sujeto [50].

También se ha podido demostrar que el aumento de la fatiga debido a un largo entrenamiento influye en los resultados. Además, cabe destacar un factor fisiológico: el “pico μ ”, que se produce durante la relajación, un componente necesario pero no suficiente para el control de dichos sistemas. La motivación y el estado de ánimo de los pacientes, además de presentar las tareas de manera atractiva para que supongan un desafío, también aumenta positivamente el rendimiento en el control de los SMR-BCI [50].

Normalmente los usuarios se imaginan el movimiento a través de imágenes visuales de ellos mismos, como si se tratara de una película. Sin embargo, se ha comprobado que se obtienen mejores resultados con la imaginación cinestésica del movimiento, que se basa en la recreación de la experiencia [51].

3.3 PROCESADO DE LOS RITMOS SENSORIOMOTORES

3.3.1 ADQUISICIÓN Y REGISTRO DE LA SEÑAL

En la mayoría de estudios sobre sistemas BCI se utilizan electrodos húmedos para registrar la señal. Para la colocación de esos electrodos se debe preparar el cuero cabelludo del sujeto con el uso de geles electrolíticos, que permitan reducir la impedancia del cráneo para conseguir un mejor registro de las señales. Por cuestiones prácticas, es ventajoso minimizar el número de electrodos necesarios, así se reducirá el tiempo de colocación de los sensores y la carga computacional al trabajar con el EEG.

Los electrodos mas utilizados son los de la región sensoriomotora, dando especial importancia a los electrodos C3 y C4, donde los SRM se registran con mayor intensidad. Si estamos ante la MI de los pies,

el electrodo predominante debe ser el Cz. En muchos estudios también se colocan electrodos cerca de los ojos para medir el EOG y poder eliminar los artefactos del pestañeo y movimiento ocular. Para el registro del EEG es necesario determinar que electrodos harán de fuente, y cuáles de referencia. Por ello, existen diferentes métodos de registro:

- **Registros bipolares:** se registran diferencias de potencial entre pares de electrodos activos). Debido al elevado número de combinaciones posibles, la *Federación Internacional de Electroencefalografía* recomienda utilizar 16 pares, diferenciando si el montaje es longitudinal o transversal.
- **Registro con referencia común:** se produce entre un electrodo activo fijado a la cabeza y un electrodo de referencia relativamente inactivo. Frecuentemente, se toma el lóbulo de la oreja, la nariz o el mentón como punto de referencia.
- **Registro con referencia promedio:** se conectan todos los electrodos con iguales resistencias a un punto, que sirve como referencia común. El potencial en ese punto es el potencial promedio de todos los electrodos de la cabeza.

3.3.2 EXTRACCIÓN DE CARACTERÍSTICAS

Las señales de EEG registradas se pueden procesar de varias maneras para mejorar la calidad de la señal. Los pasos típicos que se suelen llevar a cabo son: i) un filtrado en el dominio de la frecuencia (reduciendo o eliminando artefactos) y ii) un filtrado espacial (reduciendo el difuminado espacial); sin importar el orden en el que se apliquen. Todo ello con el objetivo de maximizar la información discriminativa y, por lo tanto, preparar de manera óptima los datos para el siguiente paso de clasificación. A continuación, se detallarán los diferentes tipos de filtrado que se realizan para extraer características. En este caso, para sistemas que utilizan los SMR como señal de control [52].

1. **Filtrado espacial:** el filtrado espacial tiene como objetivo reducir el efecto de difuminado espacial. El desequilibrio espacial ocurre como un efecto de la distancia entre el sensor y las fuentes de señal en el cerebro, debido a la no homogeneidad de los tejidos que se encuentran entre ellos. Hay una gran cantidad de métodos que intentan reducir ese desequilibrio, sin embargo los más utilizados son los filtros Laplacianos, el análisis de componentes independientes (ICA), el análisis de componentes principales (PCA), los patrones espaciales comunes (CSP) y el método de referencia de media común (CAR).
 - a) **Filtro Laplaciano:** el filtro Laplaciano calcula, para cada electrodo ubicado en el cuero cabelludo, la segunda derivada de la distribución instantánea de la tensión espacial, con el fin de enfatizar la actividad que se produce en un radio concreto alrededor de cada electrodo. Por lo tanto, se trata de un filtro espacial paso alto, que acentúa la actividad localizada y reduce la más difusa. Se puede lograr una resolución espacial alta utilizando una gran cantidad de electrodos repartidos por el cuero cabelludo [53].

El valor del Laplaciano de cada electrodo se calcula combinando el valor en esa ubicación con los valores de un conjunto de electrodos circundantes. La distancia al conjunto de electrodos circundante determina las características del filtrado, siendo el Laplaciano más sensible a los potenciales con frecuencias espaciales más altas, y menos sensible a aquellos con frecuencias espaciales más bajas a medida que la distancia disminuye [54]. Si la señal de control está muy localizada, y esa localización permanece estable a lo largo del tiempo, el Laplaciano con una distancia corta entre el electrodo principal y sus adyacentes obtendrá una mejor relación señal a ruido. No obstante, si está más dispersa o su localización varía con el tiempo, un Laplaciano con distancia mayor obtendría mejores resultados [54].

Para obtener el valor del Laplaciano se realizan aproximaciones discretizadas de la segunda derivada espacial de la distribución Gaussiana bidimensional. Las aproximaciones se simplifican aún más, de tal manera que, en cada punto de tiempo t , a la suma ponderada del potencial v_h se le resta la actividad media de los electrodos adyacentes al canal de interés, como se puede observar en la ecuación 3.1 [48]:

$$v_h^{LAP}(t) = v_h(t) - \sum_{i \in S_i} w_{h,i} \cdot v_i(t), \quad (3.1)$$

donde $w_{h,i}$ es el peso en función de la distancia $d_{h,i}$ entre el electrodo de interés, representado por el subíndice h , y los adyacentes, representados con el subíndice i .

$$w_{h,i} = \frac{\frac{1}{d_{h,i}}}{\sum_{i \in S_i} \frac{1}{d_{h,i}}} \quad (3.2)$$

Cuando el Laplaciano se calcula con los electrodos adyacentes al electrodo en el que se aplica el filtrado, se denomina Laplaciano corto. En cambio, si se calcula con electrodos más lejanos al electrodo principal, estamos hablando de Laplaciano largo. En la figura 3.3 se puede apreciar la diferencia entre ambos filtros, utilizando en los dos casos cuatro electrodos vecinos para el cálculo. Como el conjunto de vecinos está formado por cuatro, el peso de cada uno será $w_{h,i} = -1/4 = -0,25$.

- b) **Análisis de componentes independientes:** El ICA es una solución estadística particular al problema de la separación ciega de fuentes, consistente en obtener las fuentes originales de un proceso a partir de una mezcla de dichas señales. Por tanto, ICA trata de subdividir la señal EEG en diversas componentes independientes suponiendo que la señal de origen posee una independencia estadística y una distribución no gaussiana [55]. El objetivo de esta técnica es el de revelar características del EEG ocultas por el ruido basal.

Una escena particular del problema de la separación ciega de fuentes sería la siguiente: dos personas (S_1 y S_2) están hablando en un extremo de una habitación simultáneamente y

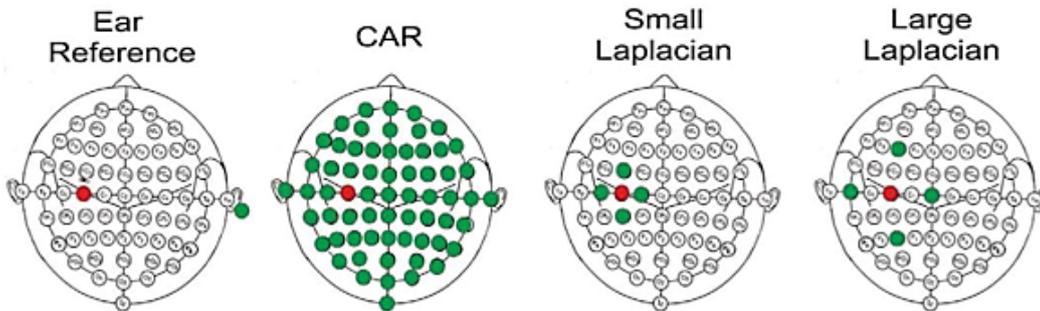


FIGURA 3.3: Representación de cuatro filtrados espaciales para obtener la señal del electrodo C3. De izquierda a derecha: referencia común, CAR, Laplaciano corto y Laplaciano largo. En todos ellos el electrodo principal C3 se representa de color rojo, y los utilizados para el filtro de color verde [16].

en el otro extremo se encuentran dos micrófonos (R_1 y R_2) en distintas localizaciones que registran la conversación. Las señales registradas podrían expresarse de la siguiente manera:

$$\begin{cases} R_1(t) = a_{11}S_1(t) + a_{12}S_2(t) \\ R_2(t) = a_{21}S_1(t) + a_{22}S_2(t) \end{cases} \quad (3.3)$$

Si conociésemos los valores de a_{11} , a_{12} , a_{21} y a_{22} podríamos resolver esas ecuaciones para obtener los valores de S_1 y S_2 , y así distinguir ambas conversaciones sin solapamiento alguno. Desgraciadamente, esos pesos son desconocidos y únicamente podemos obtenerlos si consideramos que las fuentes (S_1 y S_2) son señales independientes y de naturaleza no gaussiana. Este ejemplo se conoce como el problema *cocktail party*, y es el mismo que intenta resolver ICA. La señal EEG también se considera un problema *cocktail party*, puesto que los electrodos registran la señal EEG en distintas localizaciones sobre el cuero cabelludo. Con ICA se intentan distinguir aquellas fuentes que producen esa mezcla y, concretamente, las que producen los SMR para facilitar su posterior detección [55].

Por tanto, de forma análoga a (3.4) se puede expresar el problema generalizándolo de la siguiente forma:

$$\begin{cases} x_1(t) = a_{11}S_1(t) + a_{12}S_2(t) + \dots \\ x_2(t) = a_{21}S_1(t) + a_{22}S_2(t) + \dots \\ \dots \end{cases} \quad \rightarrow \mathbf{x} = \mathbf{A} \cdot \mathbf{s} \quad (3.4)$$

siendo \mathbf{x} la matriz vertical de las señales mezcladas y conocidas, \mathbf{A} la matriz que contiene los pesos a_{ij} y \mathbf{s} la matriz de las fuentes que se desean discriminar. Se asume que los coeficientes de pesos a_{ij} son desconocidos y que permiten que la matriz \mathbf{A} sea invertible, además de asumir

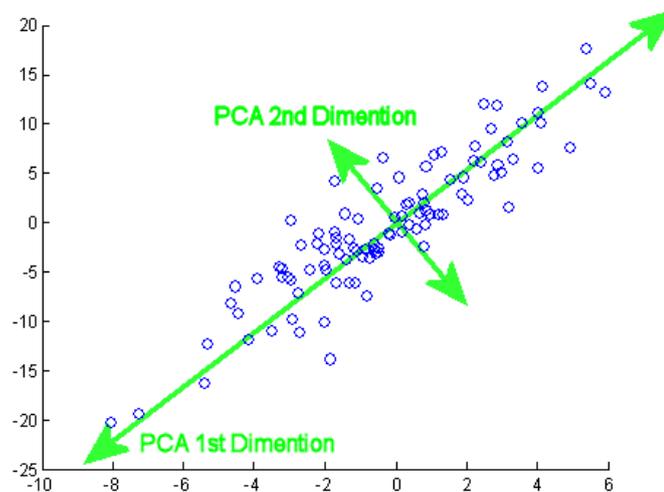


FIGURA 3.4: Representación de los dos primeros componentes principales para la nube de datos representados mediante círculos azules [57].

(como ya se ha indicado), que las componentes de \mathbf{s} son no gaussianas e independientes, que la mezcla es lineal y espontánea y que los datos son estacionarios. Bajo estas suposiciones, se concluye que existe una matriz \mathbf{W} con coeficientes w_{ij} que permite obtener las fuentes estimadas $\hat{\mathbf{s}}$ al multiplicarse con las señales mezcladas, de la siguiente manera:

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} = \mathbf{W} \cdot [\mathbf{A} \cdot \mathbf{s}] = \hat{\mathbf{s}} \rightarrow \mathbf{x} = \mathbf{A} \cdot \mathbf{s} \quad (3.5)$$

Por tanto, la matriz \mathbf{W} es la matriz pseudoinversa de \mathbf{A} , expresado: $\mathbf{W} = \hat{\mathbf{A}}^+$. Existen una gran variedad de algoritmos para hallar la matriz \mathbf{W} , entre ellos *Infomax*, *JADE* o *FastICA*, siendo *Infomax* el que mejores resultados presenta [55].

- c) **Análisis de componentes principales:** el PCA es un procedimiento matemático que, mediante transformaciones ortogonales, pretende reducir la dimensionalidad de las señales. Dado que muchas de esas señales están correlacionadas, el método busca un conjunto de valores linealmente no correlacionados, llamados componentes principales [56].

Las componentes principales son direcciones del espacio en las cuales la varianza de los datos proyectados es máxima. En la Figura 3.4 se puede ver cómo, para los datos representados, las dos primeras componentes principales apuntan a la dirección de máxima varianza.

Para el cálculo de los componentes principales primero se deben preparar los datos centrándolos y escalándolos. Para centrarlos se le resta la media a cada variable, consiguiendo que la media del conjunto de datos sea cero. Para escalarlos, se divide cada uno por su desviación

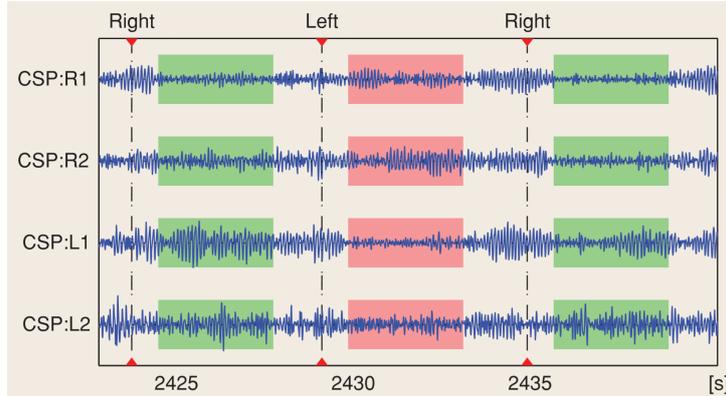


FIGURA 3.5: Efecto del filtrado espacial de cuatro filtros CSP para discernir entre MI de la mano derecha y MI de la mano izquierda [58].

estándar, logrando un conjunto con varianza la unidad. A continuación, se debe calcular la matriz de covarianza/correlación (definida en 3.6), así como los autovalores y autovectores de dicha matriz. Una vez se tengan todos los autovectores se ordenan por autovalores de mayor a menor. El autovector con autovalores más altos será el que más varianza refleje en los datos. Las componentes principales serán los autovectores calculados, y el número de autovectores escogido determinará la nueva dimensionalidad de los datos [56].

$$\text{covar}(x, y) = \sigma_{xy}^2 = \frac{\sum_i (x_i - m_x)(y_i - m_y)}{n - 1}, \quad (3.6)$$

siendo m_x y m_y la media de x e y respectivamente.

- d) **Patrones espaciales comunes:** los patrones espaciales comunes (CSP, *common spatial patterns*) son procedimientos matemáticos utilizados para maximizar la varianza para un conjunto de los datos y la minimizan para el otro, suponiendo que los datos multicanal estén basados en dos clases diferentes. Trabaja como un filtro paso-banda, que pretende obtener una discriminación efectiva de los estados mentales que se caracterizan por los efectos ERD/ERS. En la Figura 3.5 se puede ver el efecto del filtrado espacial CSP. El análisis se realizó para obtener cuatro filtros que discriminaran las MI de la mano izquierda y la derecha. Los filtros se aplican a los datos de un EEG filtrados con un paso-banda continuo. Los intervalos de MI de la mano derecha se somborean en verde, y los de la izquierda en rojo. Se puede ver cómo los filtros CSP L1 y L2 muestran mayor varianza durante MI de la mano derecha, mientras que durante MI de la mano izquierda la varianza es mayor en los filtros CSP R1 y R2 [58].

El CSP ha sido uno de los métodos más populares para la extracción de características de los SMR. La efectividad del CSP, generalmente, depende en gran medida de la selección de la banda de filtrado. Dado que la banda más adecuada puede ser específica para cada sujeto, se

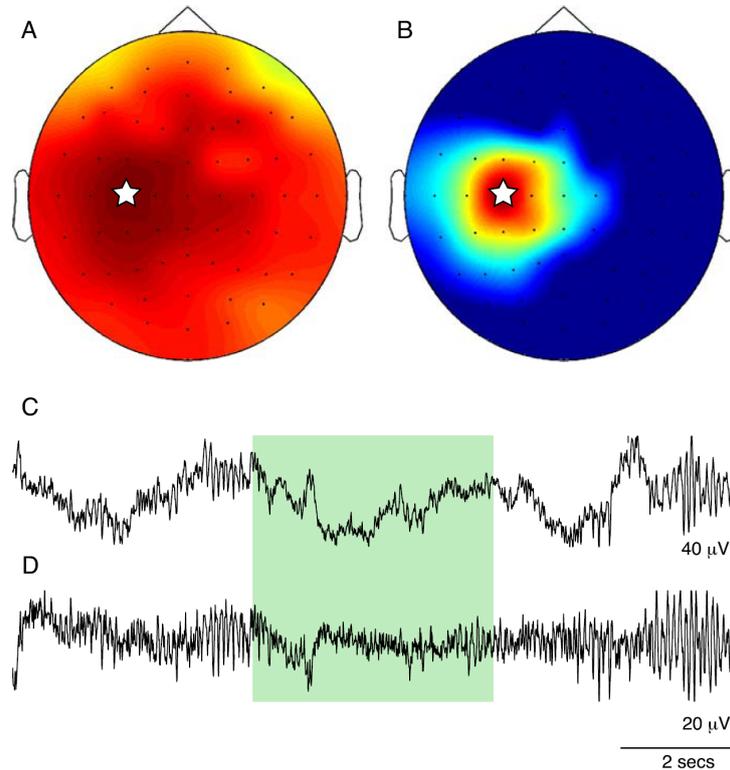


FIGURA 3.6: Ejemplo de aplicación de un filtro espacial CAR. Las señales son espacialmente más específicas y resaltan más la supresión del ritmo beta durante el período indicado por la barra verde cuando se aplica el filtro (B y D) en comparación a cuando no se aplica (A y C) [48].

han sugerido métodos que utilizan sub-bandas superpuestas para optimizar los resultados [56].

- e) **Método de referencia de media común:** el filtro CAR, es un filtro paso alto parecido al filtro Laplaciano, en el cual se acentúan las componentes que se encuentran más localizadas. Se implementa restando al canal de interés la media común del voltaje de todos los demás electrodos, como se indica en la ecuación 3.2. Si toda la cabeza estuviera cubierta por electrodos equiespaciados entre sí, y el potencial generado en el córtex estuviera muy localizado y generado por fuentes puntuales, el filtro CAR daría como resultado una distribución de voltaje espacial con una media cero [54].

$$v_i^{CAR}(t) = v_i(t) - \frac{1}{H} \cdot \sum_{i=1}^H v_i(t), \quad (3.7)$$

siendo $v_i(t)$ el voltaje del electrodo de interés en cada instante de tiempo t , y H el número total de electrodos.

En la Figura 3.6 se ilustran las topografías del efecto de un filtro CAR. En A, la señal no filtrada es espacialmente amplia; mientras que en B, la señal a la salida del CAR enfatiza

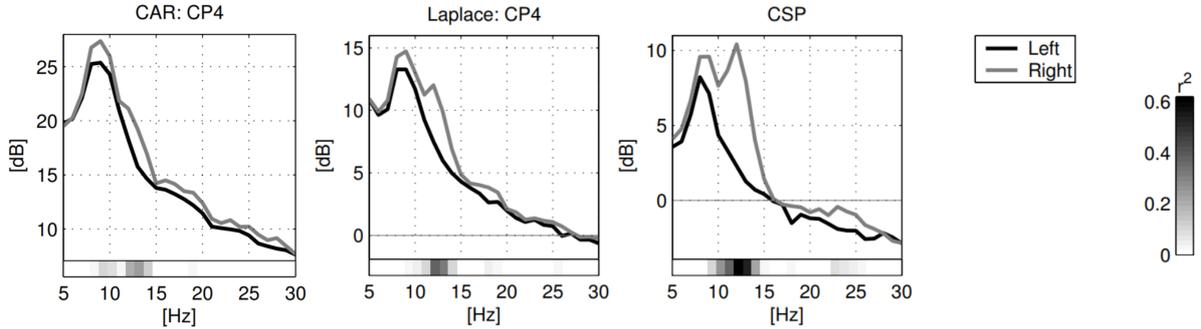


FIGURA 3.7: Respuesta de diferentes filtros espaciales para los mismos datos. De izquierda a derecha: filtro CAR, filtro Laplaciano y filtro CSP [9].

características espaciales locales. En ambas imágenes se toma como electrodo de interés el C3, marcado con una estrella. Además, se incluye una representación de los ritmos β a lo largo del tiempo, indicando en color verde cuando el sujeto abre y cierra su mano derecha (las oscilaciones beta se suprimen en ese periodo), donde C es la señal sin filtrar y la D la señal después del filtro CAR. Además de enfatizar la supresión de las oscilaciones, el filtro también elimina las fluctuaciones lentas, así excluye la varianza de la señal que no está relacionada con la tarea de movimiento [48].

En la Figura 3.7 se puede observar la respuesta a diferentes filtros espaciales para una misma señal. El objetivo de los filtros es discernir entre la MI de la mano derecha y de la mano izquierda. Se miden mediante el cálculo del r^2 . Así, se comprueba que los filtros que mejores resultados obtienen son el CSP y el Laplaciano [9].

2. **Filtrado y análisis temporal:** el filtrado en el dominio del tiempo consiste en emplear alguna técnica de promediado sobre las señales, que permita anular algunas componentes aleatorias o ruido. Una de las técnicas más habituales es el promediado sincronizado.

a) **Promediado sincronizado:** el promediado sincronizado sirve para reducir el ruido de la señal cuando se tienen varias épocas con referencia temporal. Por tanto, las señales son parecidas: $x_i(n) \approx x_j(n)$; sin embargo el ruido no: $w_i(n) \neq w_j(n)$. Para llevar a cabo este método, se realiza un promediado de todas las épocas de la señal, estableciendo como referencia ($n = 0$) el punto en el que se ha generado el estímulo, que indica la tarea que se debe realizar. Al sumarse, como se muestra en 3.8, la componentes de x se añaden pero las de w se cancelan, eliminando el ruido.

$$\sum_{k=1}^M y_k(n) = \sum_{k=1}^M x_k(n) + \sum_{k=1}^M w_k(n), \quad (3.8)$$

siendo $y_k(n) = x_k(n) + w_k(n)$ una descomposición de la señal, donde $x_k(n)$ es la señal original sin ruido, $w_k(n)$ es el ruido en la k -ésima época, k el índice de promediado, n el índice de

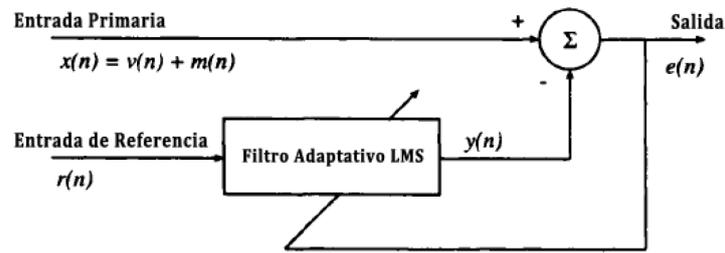


FIGURA 3.8: Esquema general de un filtro adaptativo [59].

tiempo discreto, M el número de épocas y N el número de muestras de cada época con $n = 1, 2, \dots, N$. El promediado sincronizado se utiliza normalmente con los P300.

- b) **Filtrado adaptativo:** los filtros adaptativos no suelen emplearse para la extracción de características de señales biomédicas, no obstante, pueden resultar útiles para la eliminación de artefactos conocidos, como puede ser el ECG. Este tipo de filtros no suponen que las señales sean estacionarias, sin embargo necesitan una señal fuertemente correlada con la señal ruidosa a eliminar [59].

Se trata de filtros que son capaces de adaptarse continuamente a las características de la señal, cambiando el vector de pesos a lo largo del tiempo. Existen diferentes algoritmos que implementan este tipo de filtros: i) el algoritmo de mínimos cuadrados promediados (LMS, *least mean square*), que trata de ajustar el vector de pesos para minimizar el error cuadrático medio (MSE, *mean squared error*), es fácil de implementar y no se requiere derivar, elevar al cuadrado o promediar, aunque se base en minimizar el MSE; y ii) el algoritmo de mínimos cuadrados recursivos (RLS, *recursive least squares*), que se utiliza para señales con oscilaciones rápidas en tiempo [59]. En la Figura 3.8 se puede ver un esquema general de un filtro adaptativo [59].

3. **Filtrado y análisis frecuencial:** como se ha descrito en otros apartados, los SMR están concentrados en unas determinadas bandas de frecuencia (μ y β), por lo tanto, para el procesado de estas señales se utiliza comúnmente el dominio frecuencial. Algunos de los métodos para realizar la transformación del dominio del tiempo al dominio de la frecuencia son: la transformada rápida de Fourier (FFT), la transformada Wavelet y procedimientos basados en parámetros autoregresivos.

- a) **STFT:** la transformada de Fourier de tiempo corto (**STFT**, *short-time Fourier transform*) sirve para pasar del dominio temporal al dominio frecuencial. La STFT se realiza dividiendo la señal en segmentos que se suponen estacionarios, mediante una ventana temporal, para después calcular la transformada de dichos segmentos, como se puede observar en la ecuación 3.9. La longitud de la ventana en tiempo determina la resolución frecuencial, por lo que se crea un compromiso entre resolución frecuencial y temporal. Para filtrar las señales más largas, se

usa la técnica de superposición y adición.

$$X_w(n, k) = \sum_{m=0}^{N-1} x[m] \cdot w[m-n] e^{\frac{-j2\pi km}{N}}, \quad (3.9)$$

donde $w[m]$ representa la función de ventana, $x[n]$ el segmento de la señal al que se aplica la STFT, $k \in [0, L]$, $n \in [0, N_t - 1]$ y $N_t = N/L$ el número de segmentos no solapados en los que se ha dividido la señal. Este método supone que los segmentos de señal son estacionarios. Una vez establecida la longitud de ventana mantiene una resolución tiempo-frecuencia fija, como se observa en la Figura 3.9. Por ello, no es uno de los filtros más utilizados para el procesamiento de señales biomédicas, debido a que normalmente se presentan simultáneamente oscilaciones rápidas en intervalos cortos y variaciones lentas en intervalos largos, necesitando una resolución variable.

- b) **WT**: la transformada wavelet (**WT**, *wavelet transform*) utiliza un método parecido al de la STFT, sin embargo ofrece a la vez una buena resolución temporal para oscilaciones rápidas, y una buena resolución frecuencial para oscilaciones lentas, además de desplazar la ventana para ir adquiriendo los segmentos de la señal, también la escala. Una familia de wavelets ($\Psi_{\tau,s}(t)$) se define mediante dilataciones/compresiones y desplazamientos en el tiempo de una función denominada wavelet madre ($\Psi(t)$), como se define en 3.10.

$$\Psi_{\tau,s}(t) = \frac{1}{\sqrt{s}} \cdot \Psi\left(\frac{t-\tau}{s}\right), \quad (3.10)$$

donde τ es el parámetro de transformación y s el de escalado. Existe una gran variedad de funciones wavelet madre (Morlet, Meyer, Daubechies, Symlets, etc.), que deberán ser escogidas cuidadosamente para cada caso, y así optimizar el rendimiento de la aplicación.

La transformada wavelet continua (**CWT**, *continous wavelet transform*) se aplica a señales continuas y se define como la correlación entre la señal y la función wavelet madre escalada y desplazada (ecuación 3.11), con una resolución temporal creciente y una resolución frecuencial decreciente conforme aumenta la frecuencia, como se puede observar en la Figura 3.9.

$$w(s, \tau) = \int_{-\infty}^{\infty} x(t) \cdot \frac{1}{\sqrt{s}} \cdot \Psi\left(\frac{t-\tau}{s}\right) dt \quad (3.11)$$

Debido a su alta carga computacional al trabajar con señales continuas, y su redundancia, su uso debe estar muy justificado. Puede ser muestreada sin producir una pérdida efectiva de información, lo que dio lugar a la transformada wavelet discreta (**DWT**, *discrete wavelet transform*). La DWT se puede entender como una CWT en la que se realiza un muestreo diádico de la escala y el desplazamiento (ecuación 3.12), consiguiendo una resolución en

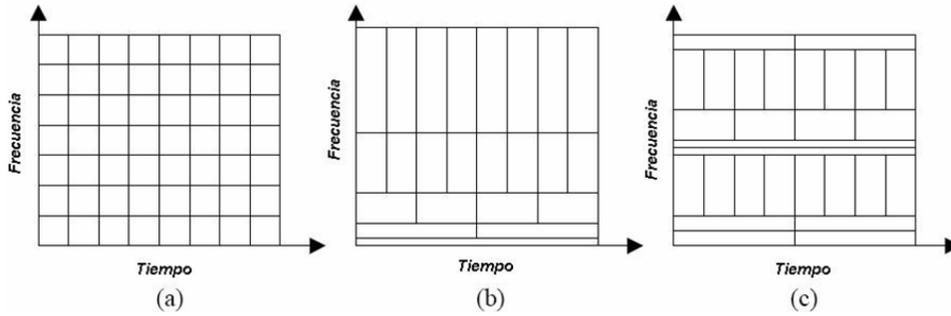


FIGURA 3.9: División del plano tiempo-frecuencia para: a) La STFT con resolución constante. b) La DWT con resolución variable adaptada para señales biomédicas. c) La WP con resolución variable según el caso.

tiempo-frecuencia variable con un coste computacional mucho menor.

$$s = 2^{-j}; \tau = k \cdot 2^{-j} \Rightarrow \Psi_{j,k}(t) = 2^{j/2} \cdot \Psi(2^j \cdot t - k), \quad (3.12)$$

donde el factor $2^{j/2}$ se denomina factor de normalización, con el que se garantiza la condición de ortogonalidad.

A partir de la DWT, se creó una generalización del concepto wavelet, llamada paquetes wavelet (**WP**, *wavelet packet*). Para trabajar con WP se parte de un algoritmo en forma de árbol, totalmente simétrico, donde cada rama representa un filtro. Al final del árbol la resolución frecuencial es muy buena, no obstante la temporal mala. Después se realiza un criterio de búsqueda y se podan las ramas del árbol que no interesen para conseguir la mejor base tiempo-frecuencia posible en cada caso. Se puede ver un ejemplo en la Figura 3.9.

- c) **Filtros FIR y filtros IIR:** los filtros de respuesta al impulso finita (**FIR**, *finite impulse response*) y los filtros de respuesta al impulso infinita (**IIR**, *infinite impulse response*) son filtros digitales de frecuencia. Se suelen utilizar cuando se quiere eliminar alguna banda de frecuencia, ya sea por proporcionar información redundante, ser innecesaria o demasiado ruidosa. Para poder filtrar la banda de frecuencias deseada se requieren dos secuencias a y b de longitudes n_a y n_b , respectivamente. Para calcularlas se pueden utilizar varios métodos, como el butterworth o el elíptico. En 3.13 se puede ver la expresión en la que la señal fuente $x(t)$, se filtra a $y(t)$ para todo t [9].

$$\begin{aligned} y(t)^{FIR} &= b(1)x(t) + b(2)x(t-1) + \dots + b(n_b)x(t-n_b-1) \\ a(1)y(t)^{IIR} &= b(1)x(t) + b(2)x(t-1) + \dots + b(n_b)x(t-n_b-1) \\ &\quad - a(2)y(t-1) - \dots - a(n_a)y(t-n_a-1) \end{aligned} \quad (3.13)$$

Los filtros FIR dependen tanto de la entrada actual como de entradas anteriores. Los

IIR, además dependen de salidas anteriores del filtro, como se observa en 3.13. Los filtros IIR pueden producir pendientes más pronunciadas entre la banda de paso y la banda eliminada. También requieren menos memoria y son más rápidos, sin embargo su diseño es más complicado porque pueden volverse inestables, mientras que los filtros FIR siempre se mantienen estables.

- d) **Estimación espectral autorregresiva:** la AR es un enfoque paramétrico cuyo objetivo es extraer ciertas estadísticas, tales como la varianza de los datos, para modelar las series en tiempo [60]. Por lo tanto, los coeficientes de un modelo AR pueden usarse como características de las señales cerebrales. Dados k canales de señales cerebrales $s_1(t), \dots, s_k(t)$, estas señales pueden ser modeladas como se indica a continuación:

$$s_i(t) = a_{i1}s_i(t-1) + \dots + a_{ip}s_i(t-p) + \epsilon_i(t), \quad (3.14)$$

donde $i = 1, \dots, k$; a_{i1}, \dots, a_{ip} son los coeficientes del modelo, ϵ_i es el proceso de ruido blanco y media cero y p es el orden del modelo. Este es un método simple de extracción de características, no obstante existen otros más avanzados como el AR multivariante o el Anadaptive AR [8].

El modelo AR es muy utilizado en señales de EEG por su no estacionariedad. Para que el modelo sea eficaz en un caso concreto es muy importante determinar un orden apropiado. Si es excesivamente bajo, el espectro se suavizará demasiado, y si es excesivamente alto, pueden surgir picos espurios. Además, al no estar limitado por la duración del proceso de entrada, es capaz de proporcionar una resolución de segmentos cortos superior a otros métodos, como la STFT [8].

- e) **Potencias relativas:** las potencias relativas (**RP**, *relative power*) estiman la aportación a la actividad oscilatoria global de las componentes espectrales en los rangos de frecuencia establecidos. Para poder calcular la RP de una banda, primero es necesario estimar la densidad espectral de potencia de la señal (**PSD**, *power spectral density*). Una vez se disponga de la PSD la RP se calcula como se indica a continuación:

$$RP_{banda} = \frac{\sum_{i=f_1}^{f_2} PSD(i)}{\sum_{j \in f} PSD(j)}, \quad (3.15)$$

donde la *banda* $\in [f_1, f_2]$ y $f \in rango_{total}$ de frecuencias de la señal. En la representación gráfica de la Figura 3.10 se observa que la RP es la relación entre el área de la PSD encerrada en la banda de frecuencia de interés, y el área total.

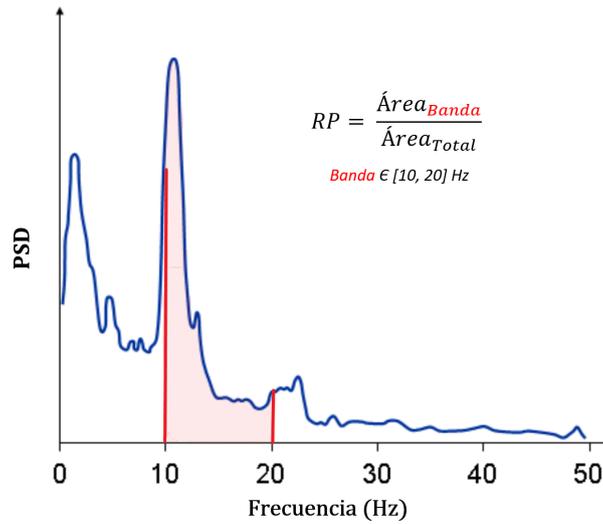


FIGURA 3.10: Cálculo de la RP en la banda [10, 20] Hz de la señal comprendida entre [0, 50] Hz.

3.3.3 CLASIFICACIÓN DE CARACTERÍSTICAS

Después de la etapa de extracción de características, pasamos a la de clasificación (también llamada traducción de características). Un algoritmo de clasificación pretende construir un modelo a partir de los datos de entrenamiento, y utilizarlo para clasificar datos nuevos (test). Las señales de control resultantes de la extracción de características pasan por el clasificador escogido y se determina a cuál de las clases establecidas pertenecen.

Existe una gran variedad de métodos de clasificación y se pueden dividir en clasificadores **lineales** y **no lineales**. A continuación, se explicarán los más destacados de cada uno de los grupos [54].

1. **Clasificadores lineales:** los clasificadores lineales utilizan funciones lineales para clasificar las señales de control en clases. Estos métodos proyectan datos de las características con alta dimensión en un espacio dimensional inferior, para poder separarlos más fácilmente en dos clases.

- a) **Discriminante lineal de Fisher:** el discriminante lineal de Fisher (**FLD**, Fisher's linear discriminant) mide la separabilidad de los datos con dos factores: la distancia entre la media de los datos de la clase proyectados (debe ser grande) y el tamaño de la varianza de los datos en esa dirección (debe ser pequeña) [8]. Esto se logra maximizando la relación entre la dispersión entre clases y la dispersión dentro de cada clase:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}, \quad (3.16)$$

$$S_B = (m_1 - m_2)(m_1 - m_2)^T, \quad (3.17)$$

$$S_W = S_1 + S_2, \quad (3.18)$$

donde S_B es la matriz de dispersión entre clases para el caso de que haya dos clases, S_W es la matriz de dispersión dentro de cada clase, w es un vector de pesos y $J(w)$ es la función que se busca maximizar. La matriz de dispersión S_w y la media de la clase m_w se definen como sigue:

$$S_W = \sum_{j \in I_w} (x_j - m_w)(x_j - m_w)^T, \quad (3.19)$$

$$m_w = \frac{1}{n_w} \sum_{j \in I_w} x_j, \quad (3.20)$$

donde n_w es el número de muestras de datos que pertenecen a la clase w e I_w es el conjunto de índices de las muestras de datos. El vector de peso w es uno de los autovectores generalizados que diagonalizan conjuntamente S_B y S_W , y se puede calcular como:

$$w = S_W^{-1}(m_1 - m_2). \quad (3.21)$$

Por tanto, la norma para clasificar mediante FLD es la siguiente:

$$y = \begin{cases} 1 & \text{if } w^T x \geq b \\ -1 & \text{if } w^T x < b \end{cases}, \quad (3.22)$$

donde b es:

$$b = \frac{1}{2} w^T (m_1 + m_2) \quad (3.23)$$

Para casos en los que haya más de dos clases, se puede construir un clasificador para cada una de ellas [8].

- b) **Análisis discriminante lineal:** el análisis discriminante lineal (**LDA**, *linear discriminant analysis*) asume la normalidad y homocedasticidad de los datos, a diferencia del FLD. Es simple de usar y tiene una baja carga computacional. Este clasificador falla si la función discriminatoria se encuentra en la varianza de los datos y no en su media [54]. Aunque se suele utilizar en los casos en los que hay que discernir entre dos clases, es posible ampliar el enfoque a múltiples clases [54]. En la Figura 3.11 se puede ver cómo clasifica el LDA entre dos clases.
- c) **Máquina de vectores de soporte:** la máquina de vectores de soporte (**SVM**, *support vector machine*) se basa en un conjunto de algoritmos que tratan de construir un hiperplano o conjunto de hiperplanos en un grupo de dimensionalidad muy alta. El método maximiza la distancia entre el hiperplano y los puntos más cercanos de cada clase, que se denominan

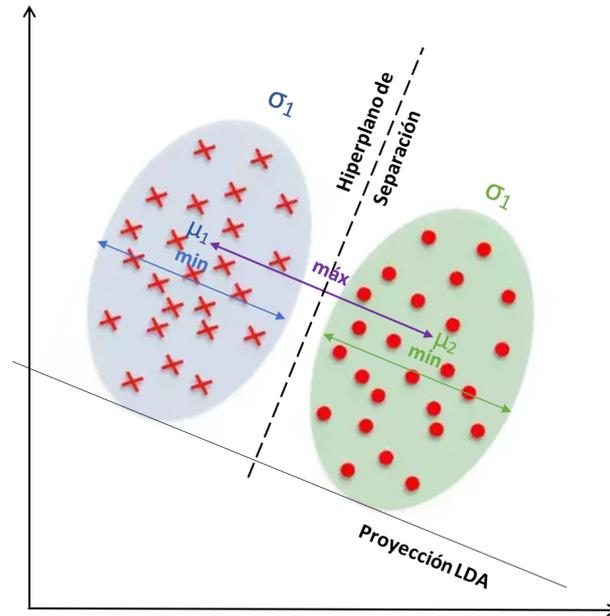


FIGURA 3.11: LDA.

vectores de soporte. El objetivo es proporcionar una buena generalización, maximizando el rendimiento de la máquina mientras se minimiza la complejidad del modelo aprendido.

La SVM trata de minimizar la función de coste de los datos que se define como:

$$J(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i, \quad (3.24)$$

donde w es el hiperplano óptimo de separación, $\|w\|^2$ la norma euclídea y $J(w, \xi)$ la función a minimizar, bajo las restricciones:

$$y_i(w^t X_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i = 1, \dots, n, \quad (3.25)$$

donde x son los datos de entrenamiento, $y \in [-1, 1]$ son las etiquetas de cada clase, C es una constante de regulación, ξ un parámetro de tolerancia y b es el sesgo. Las ξ miden la desviación de los datos de la condición de separabilidad. El SVM emplea la norma de clasificación lineal definida en 3.22 [8] [52].

Aunque normalmente se usa como un clasificador binario, como el de la Figura 3.12, también puede utilizar los métodos anteriormente descritos para expandirse a multiclase. Además, puede generalizarse empleando una función objetivo cuadrática (Kernel), lo que lo convertiría en un método de clasificación no lineal.

2. **Clasificadores no lineales:** en la práctica, cuando un clasificador lineal no es lo suficientemente

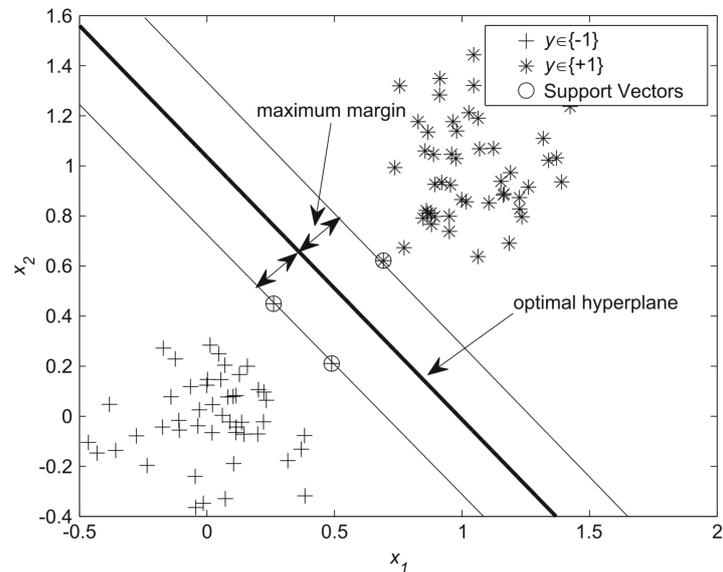


FIGURA 3.12: Clasificador SVM, los vectores de soporte se representan con un círculo y los datos de cada clase con símbolos diferentes (+,*). [8]

bueno como para discernir entre las clases, se utilizan los clasificadores no lineales, que utilizan regiones de clasificación no lineales.

- a) **Red neuronal artificial:** la red neuronal artificial (**ANN**, *artificial neural networks*) es una construcción de clasificadores no lineales a partir de una gran cantidad de elementos interconectados, llamados neuronas. Cada neurona imita a una neurona biológica y es capaz de realizar tareas computacionales simples. La ANN más utilizada es el perceptrón de múltiples capas (**MLP**, *multilayer perceptron*), en el que la red se divide en tres capas, la de entrada, una oculta y la de salida. La ventaja de MLP es que su operación rápida facilita su implementación y requiere conjuntos pequeños de entrenamiento [61].

La complejidad de una ANN se estima por el número de neuronas en la capa oculta, a mayor número de neuronas, más complejidad, pero menos errores en la clasificación. Normalmente el número de neuronas se decide después de utilizar el método de ensayo y error. El número de neuronas de entrada determina la cantidad de características, y el de salida el número de clases establecidas. En la Figura 3.13 se puede ver el esquema de una ANN para clasificar dos clases [61].

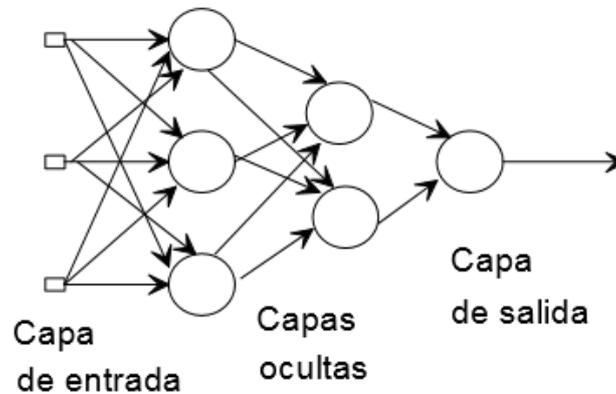


FIGURA 3.13: Clasificador mediante ANN binario [61].

- b) **K-vecinos más cercanos:** el clasificador K-vecinos más cercanos (**K-NN**, *k-nearest neighbors*) es un modelo que se basa en distancias. Primero, se colocan los datos de entrenamiento que están etiquetados (se sabe a qué clase pertenecen). Cuando llega una nueva muestra (test), se coloca en el plano junto con los datos de entrenamiento y se determinan los K vecinos más cercanos mediante la distancia clásica euclidiana. Para simplificar, se supondrá que solo hay dos características (dimensiones) y ,por tanto, los datos estarán colocados en un plano 2D:

$$d(p1, p2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.26)$$

Una vez se sabe cuáles son los más cercanos, se establece la clase de la nueva muestra consultando cuál es la clase mayoritaria entre sus vecinos. En K-NN, la letra K indica el número entero de vecinos que se consideran para tomar la decisión. En la Figura 3.14 se muestran 26 datos de entrenamiento en dos de las clases posibles, la clase 1 representada por círculos rojos rellenos y la clase 2 por círculos rojos vacíos. Las variables predictoras son X_1 y X_2 . La muestra a clasificar se representa con un triángulo verde. Si $K=3$ (3-NN), el triángulo pertenecerá a la clase 2, ya que hay solo un vecino de la clase 1 y dos de la clase 2 dentro del círculo que los contiene. En cambio, si se utilizara un 1-NN, el triángulo sería de la clase 1, ya que el vecino más cercano a éste pertenece a esa clase, y lo mismo ocurriría para el 5-NN.

También hay posibles variantes del algoritmo básico, como utilizar los vecinos más cercanos con distancia ponderada. Se puede ponderar la contribución de cada vecino de acuerdo a la distancia entre él y el ejemplar a ser clasificado, dando mayor peso (3.27) a los más cercanos.

$$w_i = \frac{1}{d(p_1, p_2)^2}, \quad (3.27)$$

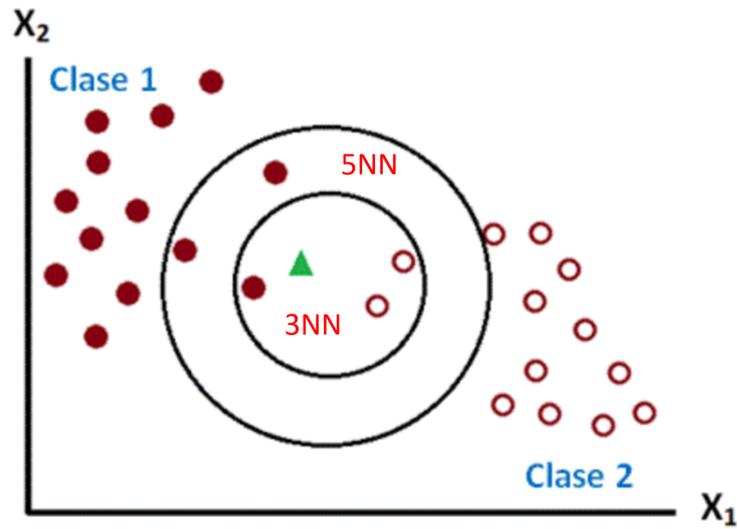


FIGURA 3.14: Clasificador mediante K-NN.

de esta manera se evita el riesgo de permitir que todos los datos de entrenamiento contribuyan a la clasificación de p , ya que al ser muy distantes, su peso prácticamente los anularía. Esto requiere mayor carga computacional, sin embargo, proporciona al algoritmo más robustez ante *outliers*.

DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE NFT

4.1 OBJETIVO DE LA APLICACIÓN

El objetivo perseguido en la elaboración de este Trabajo Fin de Grado es diseñar y desarrollar una plataforma de entrenamiento cognitivo mediante un sistema BCI, para dispositivos Android. La aplicación está destinada a mejorar el rendimiento de ciertas funciones cerebrales gracias al entrenamiento cognitivo que se realiza mediante SMR .

La portabilidad es un aspecto esencial a tener en cuenta. Por esa razón, se emplea el EEG y se busca integrar el sistema en un dispositivo portable basado en Android. Además, el EEG es un método de bajo coste, no invasivo y de fácil uso, que se utiliza mayoritariamente en sistemas BCI. La señal de control escogida han sido los SMR debido a la necesidad de aprendizaje para poder modular la señal, ya que se desea que haya un entrenamiento cognitivo. Por ello, se descartaron las señales que se producen de manera espontánea. Dentro de las endógenas, los SMR fueron los escogidos por su alta tasa de transferencia de bits/s con respecto a las otras señales de control.

Las tareas de entrenamiento de la aplicación deben resultar visualmente atractivas para los usuarios, así se intentará fomentar la concentración y la motivación de éstos, además de procurar que resulten desafiantes. Estos aspectos se han demostrado claves en el rendimiento de los sistemas basados en SMR [50]. Asimismo, las tareas a realizar se has diseñado para que sean intuitivas y provean al usuario de una realimentación constante.

4.2 ARQUITECTURA DE LA APLICACIÓN

La aplicación puede dividirse en tres partes principales, cuyas características y funcionamiento se detallarán en los siguientes apartados: la adquisición de la señal, el procesado mediante BCI2000 y la

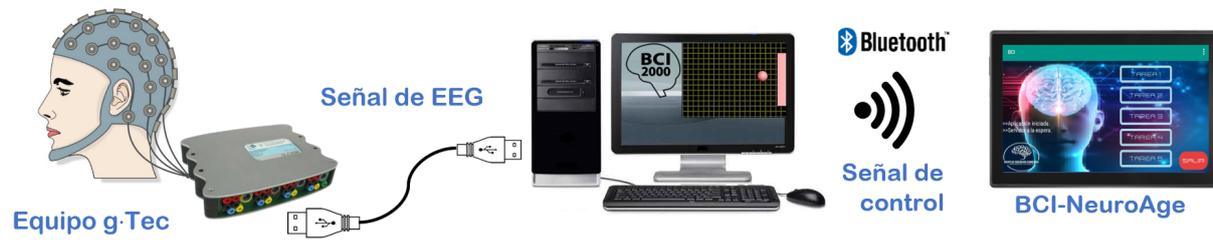


FIGURA 4.1: Esquema de la arquitectura de la aplicación BCI-NeuroAge.

aplicación para Android *BCI-NeuroAge*.

La primera es la adquisición de la señal de EEG, mediante un casco de electrodos y un amplificador, ambos de la empresa *G-Tec*. El gorro tiene capacidad para 64 canales, mientras que el amplificador solo para 16. Para el desarrollo del NFT se utilizan 8 canales.

Cuando se comienza el registro de la señal, ésta se envía a través de un cable USB a un ordenador, en el cual se procesa el EEG mediante el sistema de propósito general BCI2000. El módulo de *cursorTask* que incluye BCI2000 ha sido modificado. En esta etapa la señal se filtra, se extraen las características de los SMR generados y se clasifican para proporcionar las señales de control en tiempo real. El procesamiento de la señal se ha configurado para que se diferencie entre la MI de la mano derecha y la MI de la mano izquierda. Se emplearán dos clasificadores que discriminan clases binarias: i) un clasificador diferencia entre MI de la mano derecha y ninguna MI (basal); y ii) el otro diferencia entre MI de la mano izquierda y ninguna MI. Es decir, se obtendrán dos señales de control en todo momento, y se supondrá que la salida con valor más alto será la que nos indique la intención del usuario.

Una vez extraídas las señales de control, se envían por Bluetooth al dispositivo Android donde se esté ejecutando la aplicación *BCI-NeuroAge*. La aplicación *BCI-NeuroAge*, cuando se ejecuta, inicia el cliente Bluetooth y espera para recibir las señales de control. Una vez recibidas, se encarga de procesarlas y convertirlas en un *feedback* adecuado. *BCI-NeuroAge* consta de 5 tareas distintas. Las dos primeras solo proporcionan *feedback* positivo, es decir, los objetos solo se moverán hacia el lado correcto cuando el sujeto lo haga adecuadamente, pero no hacia el lado incorrecto cuando lo haga erróneamente. Las otras tres tareas presentan tanto *feedback* positivo como negativo, aumentando así su complejidad. Para facilitar la comprensión de la arquitectura, en la Figura 4.1 se representa la misma de manera esquemática.

4.3 ADQUISICIÓN DEL EEG

El registro de la actividad cerebral se realiza mediante la señal EEG, que se adquiere con un gorro de electrodos cuya posición viene regulada mediante el *Sistema Internacional 10-20*. El gorro se coloca sobre

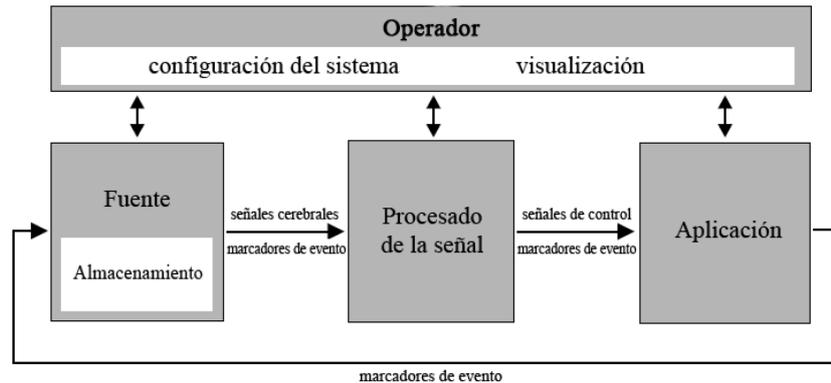


FIGURA 4.3: Estructura del sistema de propósito general BCI2000 [48].

entrenamiento cognitivo.

4.4.1 SISTEMA DE PROPÓSITO GENERAL BCI2000

Hasta la fecha, BCI2000 se ha utilizado para implementar multitud de sistemas BCI con diferentes métodos de adquisición, señales de control y procesado, debido a que posibilita la personalización de los parámetros para el procesado y adquisición de la señal. Además, provee facilidad de integración en otros entornos.

BCI2000 está compuesto de cuatro módulos que se comunican entre sí: **Fuente** (Adquisición y almacenamiento de datos), **Procesado de la señal**, **Aplicación de usuario** e **Interfaz de operador**, como se ve en la Figura 4.3. Estos módulos son programas independientes que se comunican entre sí mediante un protocolo basado en TCP/IP [48].

Las señales cerebrales se procesan de forma síncrona en bloques que contienen un número fijo de muestras, que son adquiridas en el módulo Fuente. Durante el funcionamiento del sistema, cada vez que se adquiere un nuevo bloque de datos, el módulo Fuente lo envía al de Procesado de señal, donde se extraen las características de la señal, se traducen dichas características en señales de control y se envían al módulo de Aplicación. Finalmente, el módulo de Aplicación envía los marcadores de eventos resultantes al módulo Fuente, donde se almacenan en el disco junto con las señales sin procesar, con el fin de poder realizar un análisis *offline*.

El módulo Operador provee al usuario de una interfaz gráfica que permite configurar los distintos parámetros del sistema y visualizar en tiempo real el análisis de la señal. Se encarga del manejo de los sistemas. Por ello, se comunica con el resto de módulos y permite al usuario iniciar, pausar, continuar o suspender las operaciones del sistema.

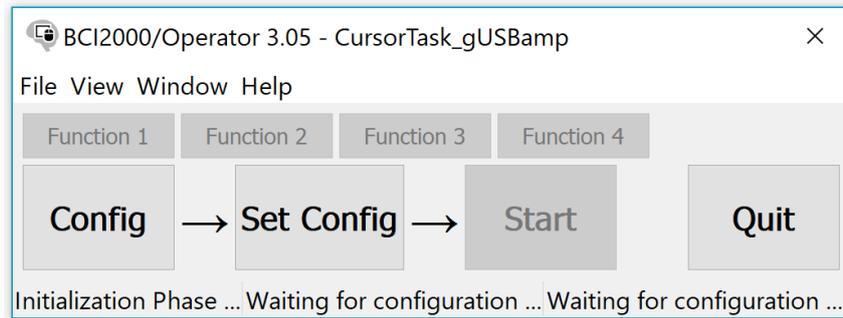


FIGURA 4.4: Interfaz gráfica del módulo Operador [48].

El sistema BCI2000 propone una solución para implementar un entrenamiento con *neurofeedback* basado en MI con la finalidad de mover un cursor llamado *CursorTask*. Al iniciar cualquiera de las aplicaciones de BCI2000, aparece la interfaz gráfica del Operador, como se puede ver en la Figura 4.4. Como el *CursorTask* es el sistema que se ha modificado para utilizarlo en la aplicación, nos centraremos en sus parámetros de configuración de ahora en adelante.

4.4.2 CONFIGURACIÓN DE PARÁMETROS

Una vez ejecutado el archivo *CursorTask_gUSBamp.bat*, aparece la interfaz gráfica mostrada en la Figura 4.4. Si se pulsa el botón de *Config* se abre una nueva ventana donde se muestran la gran cantidad de parámetros que el usuario puede configurar. Esta ventana se divide en diferentes pestañas, clasificando así los parámetros en: Visualización, Sistema, Almacenamiento, Fuente, Filtrado, Conector y Aplicación. Las diferentes configuraciones pueden guardarse en unos archivos de parámetros para cargarlas en cualquier momento.

En la pestaña de Visualización se establece qué variables se desea que aparezcan cuando se comience a ejecutar la aplicación, el rango de tiempo de la señal, si se quiere mostrar la ventana de Aplicación, etc. Los parámetros del Sistema establecen las direcciones IP de los módulos, los protocolos y las versiones utilizadas de cada uno de ellos, las variables de aleatorización, etc. En Almacenamiento se establece la ruta del directorio donde se quiere guardar la señal, el nombre, el formato y los datos de configuración que se desea que aparezcan en el archivo donde se guarda la señal. Las pestañas detalladas hasta ahora, junto con la de Conector, no han sido modificadas y han tomado el valor del archivo de parámetros *parms/mu_tutorial/MuFeedback.prm*.

Algunos de los parámetros de la Fuente, en cambio, sí se han modificado. El número de muestras por bloque se establece en 8, junto con los 8 canales digitalizados (cada canal es un electrodo). La ganancia de cada canal se pone a 1 y el *offset* a 0. A los canales se les da el nombre del electrodo correspondiente, según el sistema internacional 10-20. Los nombres son: F3, F4, T7, C3, Cz, C4, T8 y Pz. Se determina

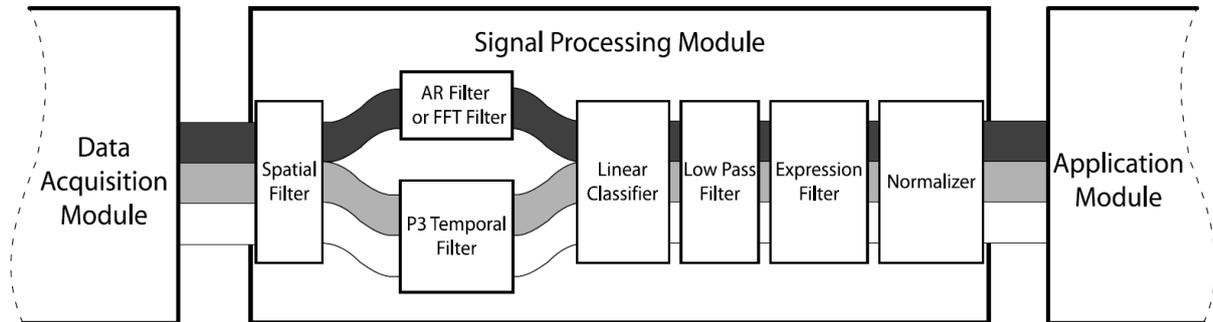


FIGURA 4.5: Módulo de procesamiento de señal [48].

que el sistema utilizará tanto tierra como referencia común. Después, se establece un filtro paso banda de Butterworth, con frecuencias de corte 0,1 y 60 Hz y de orden 8. Además, se usa un filtro Notch para eliminar el ruido producido por la corriente a 50 Hz, eliminando la banda de 48-52 Hz. Esta vez el filtro es de tipo Chebychev y de orden 4. Por último, al *buffer* se le asigna un tamaño de 2s y se añaden los 8 canales a la lista de canales transmitidos el mismo orden.

A continuación, viene la pestaña de Filtrado, donde se determinan las características de los filtros que componen el módulo del procesamiento de señal. Como puede verse en la Figura 4.5, en el caso de SMR ese módulo consta de: un filtro espacial, un filtro AR o FFT, un clasificador lineal, un filtro paso bajo, un filtro de expresión y un normalizador.

El **Filtro Espacial** se define mediante una matriz que determina la transformación lineal aplicada a la señal de entrada del filtro. En esta matriz, las columnas representan los canales de entrada, y las filas representan los canales de salida. Cada elemento de la matriz define un peso para cada canal de entrada (columna) que afecta al canal de salida correspondiente (fila). En la Figura 4.6 se puede observar qué valores se han asignado a cada señal de entrada para los dos canales de salida establecidos, y dichos pesos se corresponden con los que se obtienen al utilizar un **Filtro Laplaciano**. El primer canal de salida se obtiene al restar a la señal del canal C3 los cuatro de su alrededor, con un peso de $1/4$ cada uno (Laplaciano), que son: F3, T7, Cz y Pz. Para el segundo canal de salida, se resta a la señal del canal C4 los canales: F4, T8, Cz y Pz, con un peso de $1/4$ también. El canal C3_OUT servirá para que el clasificador determine si el usuario está realizando MI de la mano derecha. El canal C4_OUT servirá para que el clasificador determine si el usuario está realizando MI de la mano izquierda.

A continuación, se debe escoger la estimación espectral, teniendo como posibilidades un filtro AR o un filtro FFT. Para esta aplicación se utiliza un **Filtro AR** de orden 16, que calcula un modelo autorregresivo de su entrada, es decir, de la señal filtrada espacialmente, utilizando el método de máxima entropía. La estimación se realiza por separado de cada uno de los canales de entrada del filtro, y su

	F3	F4	T7	C3	Cz	C4	T8	Pz
C3_OUT	-0.25	0.00	-0.25	1.00	-0.25	0.00	0.00	-0.25
C4_OUT	0.00	-0.25	0.00	0.00	-0.25	1.00	-0.25	-0.25

FIGURA 4.6: FullMatrix del filtro espacial.

salida proporciona un espectro estimado de la amplitud recogido en “contenedores”. Dichos contenedores tendrán un ancho de 3 Hz. El primero estará centrado en 0 Hz y el último en 30 Hz. Cada uno estará compuesto por 15 puntos de evaluación. Los elementos o puntos de evaluación de la señal de salida corresponden a intervalos de frecuencia uniformemente espaciados. La longitud de la ventana de datos de entrada, sobre la cual se calcula el modelo del espectro, se ha establecido en 5s y es de tipo rectangular [48].

Una vez realizado el filtrado espectral se pasa al **Clasificador Lineal**, que traduce las características extraídas por el filtro espectral en señales de control, utilizando una ecuación lineal:

$$salida_k = \sum_{i=1}^N \sum_{j=1}^M entrada_{ij} Clasificador_{ijk}. \quad (4.1)$$

Por lo tanto, cada señal de control es una combinación lineal de las características de la señal. Los datos de entrada tienen dos índices (N canales x M elementos), y los datos de salida tienen un único índice (C canales x 1 elemento). Por ello, el clasificador actúa como una matriz $N \times M \times C$, que determina la salida después del sumatorio de canales y elementos. La matriz del clasificador establecida se puede ver en la Figura 4.7.

El procedimiento más adecuado sería realizar una sesión inicial con cada usuario para determinar las frecuencias óptimas con las que discernir entre MI de la mano derecha y de la mano izquierda, ya que varían de un sujeto a otro. En este caso se han escogido las tres frecuencias más comunes (12, 18 y 21 Hz) para cada canal de entrada ($C3_OUT$ y $C4_OUT$), otorgando a cada una el mismo peso.

El **Filtro de Expresión** utiliza expresiones aritméticas para determinar su resultado. Como las expresiones aritméticas pueden contener variables de estado además de la señal de entrada del filtro, esto proporciona una forma de modificar el procesamiento de los datos, según el estado del sistema. Sin embargo, en la aplicación este filtro no se utiliza.

	input channel	input element (bin)	output channel	weight
1	C4_OUT	12Hz	1	1
2	C3_OUT	12Hz	2	1
3	C4_OUT	18Hz	1	1
4	C3_OUT	18Hz	2	1
5	C4_OUT	21Hz	1	1
6	C3_OUT	21Hz	2	1

FIGURA 4.7: FullMatrix del clasificador lineal.

La última etapa del módulo de procesado es el **Normalizador**, donde destacan tres de los parámetros: el *offset*, la ganancia y la adaptación. En las primeras tareas de las sesiones de entrenamiento, se establecerá que haya adaptación tanto del *offset* como de la ganancia, es decir, esos valores se irán modificando cada vez que se produzca una salida del clasificador, para discernir de la mejor manera posible la intención del usuario. Los valores afectan a la salida del normalizador como se indica a continuación:

$$salida_i = (entrada_i - offset_i) \cdot ganancia_i. \quad (4.2)$$

Conforme se vea que el sujeto va aprendiendo y que la adaptación de dichos parámetros va siendo cada vez más innecesaria, se modificará el valor de la adaptación para que ésta no se produzca y los parámetros permanezcan constantes. La salida del Normalizador es la señal de control utilizada en la aplicación. Debido a la manera en que se ha configurado el clasificador, este producirá dos señales de salida, provocando lo mismo en el Normalizador. Cada vez que haya que clasificar y normalizar, habrá dos señales de control, una para indicar si se considera que la intención del usuario era elegir la izquierda y otra señal para expresar si se cree que el usuario deseaba elegir la derecha.

Por último, los parámetros de la pestaña de aplicación sí han sido modificados para el desarrollo correcto de la aplicación. Los primeros que aparecen sirven para determinar qué fracción de la pantalla se quiere destinar a la ventana donde se presenta el cursor, el objetivo del *trial* y el *feedback*. Además, se pueden elegir los colores del fondo, del cursor y el objetivo, así como las dimensiones de éstos. Todos estos parámetros no interesan porque el sujeto visualiza el *feedback*, el cursor y el objetivo en el dispositivo

Android, donde se ejecuta la aplicación, y no verá la pantalla del ordenador donde se esté ejecutando BCI2000. Los parámetros que sí son de interés son los referentes a la secuencia de la tarea, donde se definen:

- ***PreRunDuration***: duración en segundos de la pausa antes de que comience el primer *trial* de la secuencia.
- ***PreFeedbackDuration***: duración en segundos de la visualización del objetivo antes de que comience el *feedback*.
- ***FeedbackDuration***: duración en segundos del *feedback*, que determina a su vez la velocidad del cursor. Como el *feedback* que nos interesa no se realiza en BCI2000, este parámetro nos es indiferente.
- ***PostFeedbackDuration***: duración en segundos de la visualización del resultado, una vez finalizado el *trial*.
- ***ITIDuration***: duración en segundos de la pausa entre *trials*.
- ***MinRunLength***: duración de una ejecución. Corresponde al tiempo en el que se están grabando continuamente datos en un archivo. La ejecución no se parará en medio de un *trial*, por lo que la duración puede ser mayor que el valor del parámetro.
- ***NumberOfTrials***: número de *trials* en una ejecución. Si se define este parámetro, ***MinRunLength*** debe estar vacío, y viceversa. En este caso, se utiliza ***NumberOfTrials***.
- ***MaxFeedbackDuration***: tiempo máximo en segundos que dura un *trial*. Es decir, si no se consigue el objetivo en este tiempo, el *trial* se termina.

Todos estos parámetros deben configurarse de igual manera en la aplicación Android, para que ambos sistemas estén sincronizados, es decir, hasta que no se inicie el sistema en BCI2000 y comience el primer *trial*, tampoco empezará en la aplicación Android *BCI-NeuroAge*. Para que concuerde todo en ambos sistemas, en los parámetros se ha determinado que solo haya dos objetivos, derecha o izquierda, y que en cada *trial* se establezca de manera aleatoria.

Una vez se tenga la configuración deseada cargada, se debe pulsar el botón de ***Set Config*** de la interfaz del Operador para iniciar la ejecución. Cuando la señal de EEG se ha estabilizado lo suficiente y se tenga la aplicación Android a la espera, se pulsa el botón ***Start*** que inicia la secuencia de *trials*. El botón ***Quit*** se utiliza si se desea cerrar la aplicación. Una vez pulsado el botón ***Start***, aparece en su lugar el botón de ***Suspend***, que sirve para poner fin a la ejecución. Una vez finalizada la ejecución, en vez de aparecer el botón ***Start***, aparece el botón ***Resume***, que se utiliza para volver a realizar la ejecución cuando no se ha producido ningún cambio en la configuración. Sin embargo, si se desea modificar la configuración, se deberá pulsar ***Set Config*** de nuevo y reaparecerá el botón ***Start***.

4.4.3 MODIFICACIÓN DE CÓDIGO FUENTE

Se han llevado a cabo modificaciones en el código fuente del bloque *CursorTask*, que está escrito en C++ y han sido realizadas mediante *VisualStudio-2012*, debido a que versiones más recientes no son compatibles con BCI2000. Se ha añadido un archivo nuevo, "*BluetoothClient.cpp*", que se encarga de crear un cliente Bluetooth y conectarlo al dispositivo Android, cuyo nombre debe ser *BCIMobilePhone*, e incluye una función para enviar datos mediante el cliente Bluetooth. Todas estos cambios pueden encontrarse en el Apéndice A.

Los archivos ya existentes, "*CursorFeedbackTask.cpp*" y "*CursorFeedbackTask.h*", han sufrido cambios para poder integrar la nueva funcionalidad Bluetooth. Concretamente, en el archivo de cabecera (*CursorFeedbackTask.h*) se han añadido nuevos parámetros para permitir la conexión y los distintos métodos definidos en el nuevo archivo *BluetoothClient.cpp*.

En el archivo *CursorFeedbackTask.cpp* se realizan las llamadas a los métodos del nuevo archivo. En la función *OnInitialize*, que se lleva a cabo cuando se pulsa el botón *Set Config* para establecer la configuración, se inicializan los valores de los parámetros, incluyendo el nombre del dispositivo y su dirección Bluetooth.

En la función *OnTrialBegin*, que se ejecuta cada vez que comienza un *trial*, se envía mediante la conexión Bluetooth llamando al método correspondiente del archivo *BluetoothClient.cpp* (*send_through_bluetooth*), el objetivo del *trial*. Es decir, se enviará un 1 si el objetivo está a la izquierda y un 2 si el objetivo está a la derecha. Esto es necesario debido a que los parámetros del Normalizador deben adaptarse continuamente en las primeras tareas y la adaptación depende del objetivo. Por ello, el objetivo debe ser el mismo tanto en BCI2000 como en la aplicación *BCI-NeuroAge*. En caso contrario, el *offset* y la *ganancia* se adaptarían erróneamente. Por la misma razón, los *trials* deben estar sincronizados entre los dos sistemas. Para evitar que se produzca un acierto en el sistema BCI2000 (lo que provoca el fin del *trial* y el comienzo del siguiente), sin que se produzca uno en *BCI-NeuroAge*, se ha eliminado el movimiento del cursor en BCI2000, ya que no es necesario al recibir el *feedback* de la aplicación *BCI-NeuroAge*.

Para evitar el movimiento del cursor simplemente se ha eliminado la línea que lo producía, que se encontraba en el método *DoFeedback* del archivo *CursorFeedbackTask.cpp*. Dicho método se ejecuta cada vez que se produce una salida del Normalizador. En esta función se ha añadido un fragmento de código, en el cual se van guardando los valores de la señal de control normalizada (salida del Normalizador) en un *buffer*. Una vez que el *buffer* contenga 80 valores de las señales de control, éste se envía mediante la conexión Bluetooth llamando al método del archivo *BluetoothClient.cpp*, *send_through_bluetooth*. La utilización de un *buffer* se justifica debido a que cada vez que se utiliza el método de envío, se debe crear una conexión entre cliente y servidor para asegurar que los datos se envían de forma correcta, lo que

ralentiza mucho el programa. Por ello, al crear un *buffer* se reduce en gran cantidad el número de veces que se utiliza el método, aumentando la velocidad. Un aspecto importante, dado que el sistema necesita recibir los datos en tiempo real.

4.5 APLICACIÓN BCI-NEUROAGE

La aplicación *BCI-NeuroAge* es la que se debe instalar en el dispositivo Android que se vaya a utilizar en las sesiones de entrenamiento cognitivo. Esta aplicación se ha implementado utilizando la herramienta para desarrolladores proporcionada por *Google*, *Android Studio 3.0.1*, y la plataforma para el desarrollo de juegos en 3D y 2D, *Unity 2017.2.0f3*. La programación realizada en dichas plataformas se puede encontrar en el Apéndice A.

Android Studio es una potente herramienta para crear aplicaciones Android mediante el lenguaje de programación Java. Esta herramienta proporciona de manera gratuita un elevado número de funcionalidades, como la depuración de la aplicación en tiempo de ejecución en un dispositivo real mediante conexión USB, además de facilitar al programador información sobre errores lógicos o sintácticos de manera constante durante la generación del código.

Unity es un software para el desarrollo de videojuegos multiplataforma, creado por *Unity Technologies*, que posee una versión gratuita (*Unity Personal*) destinada a principiantes. Esta tiene menos funcionalidades que su versiones profesionales, sin embargo, igualmente cuenta con una gran cantidad de herramientas, que hacen el desarrollo de contenidos interactivos 2D y 3D lo más accesible posible a todo el mundo. El motor gráfico utiliza OpenGL ES en Android y iOS.

4.5.1 PROGRAMACIÓN EN UNITY

La plataforma *Unity* se ha utilizado para realizar las diferentes tareas, utilizando gráficos en 3D para intentar hacerla lo más visualmente atractiva posible, con el fin de incentivar la concentración y motivación del usuario.

En un proyecto de *Unity* se trabaja con escenas. Las escenas contienen los entornos del juego. Cada una es como un nivel único donde se colocan los obstáculos, el entorno y las decoraciones, construyendo el juego en pedazos. Cada escena mostrará al usuario una de las 5 tareas que se proponen en el proyecto. Adicionalmente, se ha creado otra escena para poder transferir los datos de configuración a todas las demás de manera eficiente. Esta última será la escena que jugará el papel de administrador, y es la que se ocupa de cargar la tarea que se haya seleccionado en el menú de la aplicación y guardar la configuración seleccionada por el usuario (número de *trials*, duración de éstos, etc.).

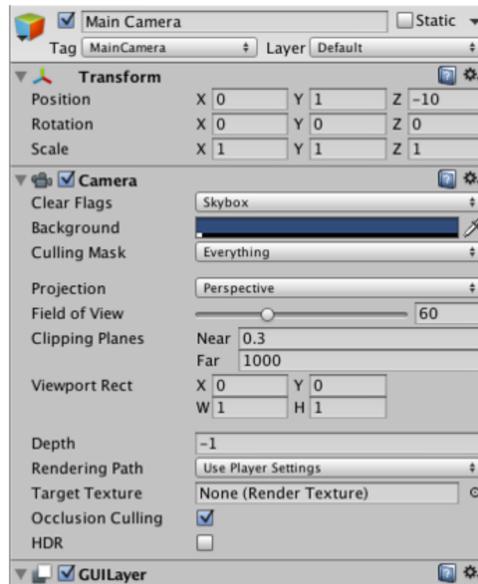


FIGURA 4.8: Ejemplo de los componentes que forman un *GameObject* con función de cámara.

Cada escena está formada por diferentes objetos 3D o 2D, instanciados como un *GameObject*, desde personajes hasta luces, cámaras y efectos de animación. Sin embargo, un *GameObject* no puede hacer nada por sí mismo. Se necesita darle propiedades antes de que pueda convertirse en un personaje, un entorno o un obstáculo. Para darle las propiedades necesarias para que tome el comportamiento deseado, se le deben agregar componentes. Según el tipo de objeto que se desea crear, se agregarán diferentes combinaciones de componentes al *GameObject*. Se puede pensar en un *GameObject* como en una olla vacía, y en los componentes como ingredientes que componen la receta del juego.

Por poner un ejemplo, el componente ***Transform*** de un *GameObject* permite determinar la posición, la rotación y la escala de dicho objeto en la escena. En la Figura 4.8 se pueden ver los componentes de un objeto que toda escena tiene, una cámara principal.

Para poder añadir funcionalidades a un *GameObject* se utilizan los *scripts*. Cuando se asocia un *script* a un *GameObject*, el *script* se convierte en un componente del objeto, y se compila como tal ejecutando cualquier funcionalidad que se haya programado en él. El *script* tiene predefinidas unas funciones, las que se han utilizado en este caso son:

- ***Awake()***: esta función siempre se llama antes de la función *Start*. Si un objeto está inactivo durante el inicio, *Awake()* no se ejecutará hasta que no se active. Esta función se utiliza para inicializar variables.
- ***Start()***: esta función se llama antes de la primera actualización del esqueleto del sistema, solo si la instancia *script* está activa. Se utiliza para inicializar punteros a *GameObjects*. Además, en

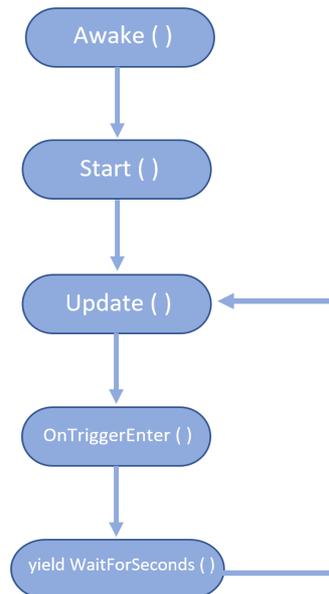


FIGURA 4.9: Diagrama de flujo de una escena.

los objetos encargados del control global de la escena, se encarga de crear un archivo donde se guardarán los valores de las señales de control para su estudio *offline*.

- ***Update()***: la función *Update* se ejecuta una vez por marco (es el sistema estructural del juego). En esta función se recogen los valores de todas las variables que cambian con el tiempo y se desarrolla el comportamiento de los objetos que interactúan en la escena. Se les proporciona movimiento, cambios de color, etc.
- ***yield WaitForSeconds(seg)***: es un tipo de corutina que normalmente se ejecuta después de la función *Update*. Suspende o pausa la ejecución hasta que termine la función *yield*, que lo hará después del retardo especificado.
- ***OnTriggerEnter(Collider)***: esta función detecta cuándo los contenedores de dos *GameObject* entran en contacto, y se utiliza en la Tarea 5 para detectar la colisión entre el personaje y cualquiera de los obstáculos.

En la Figura 4.9 se puede observar un diagrama que representa el flujo que sigue una escena. Entre cada una de las funciones pueden añadirse muchas más, no obstante solo se representan las utilizadas en la aplicación.

En las 5 escenas de las tareas hay objetos comunes en todas, como la cámara principal, las luces y un *GlobalController*. La cámara nos muestra lo que se verá en la pantalla, y se puede mover durante la ejecución si se quiere seguir al personaje, por ejemplo, como ocurre en la Tarea 5. Nos muestra la escena como si estuviera grabándola desde la posición que se le de en el componente *Transform*, es decir, si la

cámara está colocada en frente de un objeto, es dicho objeto lo que se observará en la pantalla. Las luces producen brillos y sombras según se configuren, y son necesarias para poder ver la escena, sin ellas todo quedaría en la oscuridad y no se apreciarían los objetos. El *GlobalController* hace las veces de administrador de la escena. Además, se ocupa de las tareas que no son propias de ningún objeto en particular, pero sí necesarias para que se obtenga el resultado deseado. Algunas de las tareas del *GlobalController* son: i) salir al menú principal si se pulsa la tecla “atrás” del dispositivo, o si se selecciona el botón “salir” de la escena; ii) hacer aparecer mensajes para indicar el final del *trial*, tanto si se ha producido un acierto o si ha expirado el tiempo establecido para el *trial*; y iii) al final de la Tarea, da la posibilidad de recargarla o salir al menú principal.

Además de los objetos comunes, cada escena posee diferentes objetos con distintas formas y funcionalidades, según se haya diseñado la tarea. Eso sí, todas las tareas tienen un objetivo. Si se cumple dentro del tiempo establecido se considera un acierto, y si no, se considera un fallo, mostrando al final de la tarea el recuento. Además, las escenas de las tareas poseen al menos un *GameObject* que hace el papel de personaje o jugador. Mediante el movimiento y/o el cambio de color de esos objetos se proporciona el *feedback* al usuario. Una vez finalizado el desarrollo de las tareas y la escena administradora, se exportó el proyecto de *Unity* en forma de proyecto **Gradle** para dispositivos Android, para así poder insertarlo en el proyecto de *Android Studio* en forma de librería.

La escena que se encarga de la administración no muestra nada en pantalla, simplemente accede a los datos que se han configurado en el menú y carga la escena de la tarea seleccionada.

4.5.2 PROGRAMACIÓN EN ANDROID STUDIO

La aplicación *BCI-NeuroAge*, como toda aplicación Android, se compone de una o varias actividades. Una actividad es una instancia de clase que presenta una interfaz gráfica al usuario y que contiene los elementos necesarios para que pueda interactuar con ella. Cada una de las actividades tiene asociado un fichero xml, que define la interfaz gráfica, y uno o varios archivos .java con el código de ejecución de la actividad.

Las partes de la aplicación que se han desarrollado con la herramienta de Google, *Android Studio*, son: el menú principal, el menú de preferencias, la gestión del servidor Bluetooth y la integración de las Tareas diseñadas en *Unity*. La actividad principal de la aplicación, que es la que se ejecuta cuando se lanza la aplicación, está contenida en el archivo “*MainActivity.java*” y su interfaz gráfica está diseñada en varios archivos “*activity_main.xml*” diferentes, que se intercalan según las propiedades de la pantalla (vertical, horizontal, grande, mediana o pequeña). Para poder entender esta actividad conviene explicar lo que es el hilo de interfaz de usuario (**UI**, *user interface*). El hilo UI es el proceso padre de toda la aplicación, y en él se ejecuta la actividad principal. Cualquier otro proceso que se inicie en paralelo

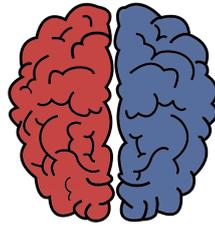
heredará sus atributos, pasando a ser un proceso hijo.

Al lanzar la aplicación, el programa comienza con la creación del servidor Bluetooth mediante el método “*startServer*”. Este método crea un proceso hijo en la clase “*AcceptThread*”, que hereda de “*Thread*”, poniendo en marcha otro hilo que ejecuta el código contenido en la función “*run()*” de la clase. Una vez realizadas las operaciones necesarias, la clase “*AcceptThread*” está lista para recibir conexiones Bluetooth mediante el método bloqueante “*accept()*”, de la clase “*BluetoothServerSocket*”. Esta operación crea un punto de acceso al programa mediante un *socket* Bluetooth, que está determinado por un identificador universal único (UUID, *universally unique identifier*). El proceso quedará bloqueado a la espera de que un dispositivo se conecte a este *socket*.

El proceso principal, mientras tanto, ha creado un escuchador para cada botón del menú principal. Hay un botón para cada una de las 5 tareas, un botón para salir de la aplicación y un menú de preferencias para poder configurar las variables de las tareas, para así sincronizarse con BCI2000. Los escuchadores de los botones lanzarán la actividad correspondiente cuando se pulse uno de ellos. Si se pulsa el botón de Preferencias se lanzará la actividad “*PreferenciasActivity.java*”, cuya interfaz gráfica está definida en el archivo “*preferencias.xml*”. Si se pulsa uno de los botones de las 5 tareas, las variables establecidas en el menú de preferencias y la tarea escogida se almacenan en una clase llamada “*VariablesGlobales.java*”. A continuación, se lanza la actividad “*UnityPlayerActivity.java*”, que cargará la escena que se ocupa de la administración global y accederá a los valores de las variables globales, utilizando los métodos definidos en “*InterfaceUnity.java*”.

Hasta que no se pulse uno de los botones de las tareas, no se iniciará el programa *CursorTask* modificado en BCI2000. Cuando se inicie el cliente Bluetooth programado, intentará conectarse al *socket* con el UUID establecido para enviarle la información que hemos explicado anteriormente. Al llegar la petición de conexión al servidor, se acepta y se instancia la clase “*ConnectedThread*”, que también hereda de “*Thread*” y que crea un nuevo proceso hijo. En este punto, el proceso que estaba bloqueado en “*AcceptThread*” continúa su curso y queda inmediatamente disponible para aceptar otra conexión, mientras que el nuevo hilo creado por “*ConnectedThread*” se encarga de recibir y procesar los paquetes de datos que llegan al punto de acceso. Una vez procesados y convertidos en cadenas de texto, si se trata del objetivo del *trial*, se guarda en una de las variables globales y se activa el *flag* que indica el comienzo de la tarea. Si se trata del *buffer* con los valores de las señales de control, se envía al hilo principal. Hay que añadir que la comunicación entre procesos no es trivial y es necesario un objeto de una clase que herede “*Handler*”, que es una herramienta proporcionada por *Google* para tal cometido.

Una vez enviado el *buffer* de datos al hilo principal, se crea otro proceso hijo para ir leyendo dichos datos con la misma velocidad con la que se van generando, añadiendo un retardo de unos pocos milise-

FIGURA 4.10: Icono de la app *BCI-NeuroAge*.

gundos entre lectura y lectura. Si no se añade dicho retardo, los datos se leerían tan rápido que solo se apreciaría la última lectura y se tomaría el valor de dicha lectura todo el tiempo, hasta que llegue un nuevo *buffer*.

4.6 MANUAL DE USUARIO

A continuación, se realiza un análisis detallado del funcionamiento de la aplicación, con la intención de servir de guía para que el usuario sepa cómo navegar por ella y comprenda su funcionamiento. La aplicación se ha diseñado con el objetivo de que sea intuitiva y fácil de manejar. De esta manera, se espera que los usuarios se sientan cómodos y no se produzca fatiga a la hora de trabajar con ella.

4.6.1 MENÚ PRINCIPAL

Cuando en el dispositivo Android se lanza la aplicación pulsando en el icono que se muestra en la Figura 4.10, lo primero que aparece es el menú principal. Si el sistema Bluetooth del dispositivo está desactivado, saltará una pequeña ventana encima del menú, solicitando permiso para activarlo, ya que es necesario para el funcionamiento de la aplicación. Dicha ventana aparece en la Figura 4.11. Si el Bluetooth ya estaba activo, o si se cierra la ventana de permiso aceptando la activación del Bluetooth, solo queda el menú principal.

En la Figura 4.12 se muestra la interfaz gráfica del menú principal. En la parte superior o inferior izquierda de la pantalla, según se encuentre en posición vertical u horizontal, aparece una lista de comandos que nos indican cómo se encuentra el servidor Bluetooth. Dependiendo de lo que ocurra, hay varias posibilidades, que se explican a continuación:

- **"Error: el adaptador Bluetooth por defecto no se encuentra"**: este mensaje aparece si, al comprobar si el dispositivo Android tiene un adaptador Bluetooth por defecto, no se ha encontrado ninguno y convendría comprobar si tiene la funcionalidad Bluetooth. En caso contrario, se recomienda cambiar a otro dispositivo que sí la posea.
- **"Aplicación iniciada"**: este mensaje aparece cuando el Bluetooth del dispositivo se encuentra activo y en funcionamiento.

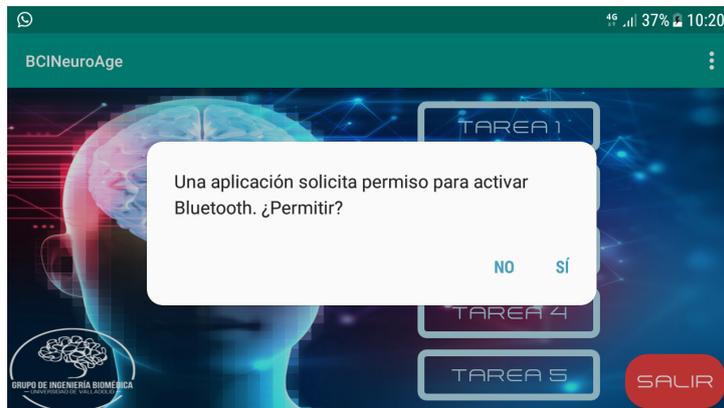


FIGURA 4.11: Ventana emergente solicitando el permiso del usuario para activar el Bluetooth del dispositivo.

- **"Error: no se puede lanzar el puerto RFCOM"**: este mensaje aparece si se ha intentado abrir un puerto RFCOM para establecer una comunicación con los identificadores: "*BCIMobilePhone*", como nombre del dispositivo; y el UUID establecido.
- **"Servidor a la espera"**: este mensaje aparece si el subproceso para recibir los comandos vía Bluetooth se ha creado correctamente y está esperando a la escucha para cuando reciba alguno.
- **"Error: el servidor no puede lanzarse"**: este mensaje aparece si no se ha podido crear el subproceso que se encarga de recibir los datos enviados por Bluetooth.

En la pantalla se pueden apreciar 7 botones. Los botones para seleccionar la tarea que se quiere realizar están ubicados en el centro de la pantalla, si se encuentra en posición vertical, y más hacia la derecha si está en horizontal. Cada botón indica el número de la tarea que se abrirá si se pulsa. No obstante, antes de comenzar cualquier tarea, es conveniente abrir el menú de preferencias para configurar los parámetros oportunos.

El botón que permite acceder a las preferencias se encuentra en la parte superior derecha de la pantalla. Está representado por una fila de tres puntos en vertical. Cuando se pulsa el botón, aparece un menú desplegable donde solo se visualiza la palabra Preferencias. Si se presiona en ella se abrirá el menú de preferencias que se muestra en la Figura 4.13.

En las preferencias se pueden cambiar distintos parámetros, que sirven para poder configurar las variables de las tareas y así poder sincronizarlas con BCI2000. Los parámetros configurables se dividen en tres categorías: los comunes a las 4 primeras, los personalizados para la quinta y los relativos al Bluetooth.

Los comunes a las 4 primeras tareas son:

- **Modo *Debug***: si se activa esta variable significa que estamos en modo depuración, orientado a

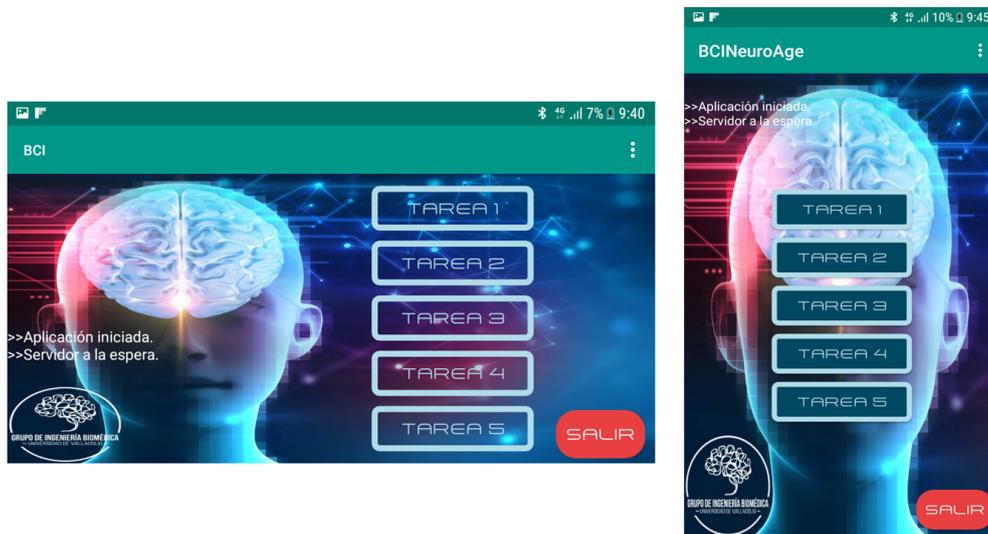


FIGURA 4.12: Menú principal de la aplicación en orientación horizontal y vertical de izquierda a derecha respectivamente.

realizar pruebas con la aplicación. Esto provoca que en las tareas aparezca, en la parte inferior izquierda de la pantalla, el valor que tienen algunas de las variables en tiempo real, tales como, las salidas del clasificador de BCI2000 que se envían a la aplicación.

- **Número de *trials* en cada tarea:** indica el número entero de *trials* o intentos de cada tarea que se van a realizar.
- **Segundos que dura un *trial*:** indica el tiempo máximo en segundos que dura cada uno de los *trials*. Si este tiempo vence antes de que se acierte el *trial*, se pausará la tarea. Cuando comience el siguiente *trial* en BCI2000 se iniciará también en la aplicación.

Los de la tarea 5 son:

- **Velocidad inicial:** indica el valor de la velocidad inicial con la que el personaje de la tarea 5 va andando hacia delante.
- **Disminución de velocidad:** si se activa este parámetro, cada vez que ocurra un fallo de un *trial*, la velocidad del personaje de la tarea 5 disminuirá la cantidad que se establezca en el incremento de velocidad.
- **Incremento de velocidad:** cuando se supera uno de los *trials* de la tarea (esquivar un obstáculo) se aumentará la velocidad del jugador una cantidad igual a la especificada aquí.
- **Duración de la tarea 5:** indica la duración en minutos de la tarea 5.

Los relativos al Bluetooth son:

- **Nombre del cliente Bluetooth:** indica el nombre del dispositivo Android que se debe buscar para conectarse vía Bluetooth.

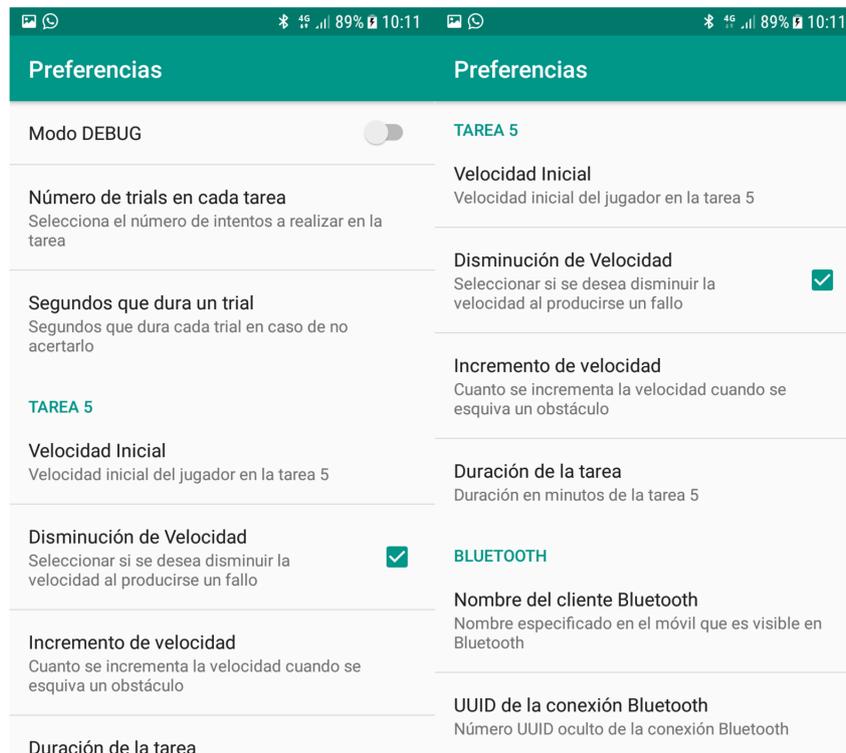


FIGURA 4.13: Ventana donde se muestra la lista de preferencias que se pueden configurar.

- **UUID de la conexión Bluetooth:** indica el identificador único que determina el socket que proporciona el punto de acceso de la conexión Bluetooth.

4.6.2 TAREAS

Si se pulsa uno de los botones del menú principal para acceder a una de las tareas, se lanzará la tarea seleccionada. Cada una de las tareas tienen un comportamiento e interfaz diferentes, que se explican a continuación.

Cuando se inicia la Tarea 1 aparece en la pantalla una mesa, encima de la cual hay un cubo a la derecha y otro a la izquierda, como se muestra en la imagen de la izquierda en la Figura 4.14. El cubo de la derecha es de color azul y el de la izquierda de color rojo. Cuando se inicia el *trial*, aparece en la parte superior el objetivo de dicho *trial*. Solo hay dos objetivos posibles: derecha o izquierda. Además del objetivo, aparece una barra en la parte superior del lado correspondiente al mismo. Si el objetivo es derecha, el usuario deberá imaginar el movimiento de la mano derecha; y si es izquierda, el movimiento de la mano izquierda. Si lo realiza correctamente, el cubo del lado objetivo comenzará a ascender y girar cada vez más rápido. Si no lo realiza correctamente no ocurre nada, los cubos no se mueven. El *trial* finaliza si el cubo consigue superar la barra superior, o si antes de que eso ocurra se vence el tiempo máximo del



FIGURA 4.14: Imágenes que muestran el comportamiento de la Tarea 1.



FIGURA 4.15: Iconos de los botones que aparecen al finalizar las tareas. El de la derecha es el de recargar la tarea, y el de la izquierda el de salir de ella al menú principal.

trial. La tarea acabará cuando se completen el número de *trials* especificados y aparecerá el recuento final de fallos y aciertos. Además, en la parte superior central, se colocan dos botones que permiten recargar la tarea o salir de ella, cuyos iconos se muestran en la Figura 4.15. Al finalizar las tareas aparecen dichos botones y el contador de aciertos y fallos.

Al comenzar la Tarea 2 aparece un cubo blanco que gira en el centro de la pantalla y una barra con un marcador señalado con una flecha en la parte inferior. El relleno de la barra va en degradado de blanco a azul, desde el marcador hacia la derecha; y de blanco a rojo, hacia la izquierda. Además, en cada extremo de la barra aparece una mano, y dependiendo del objetivo se marcará la mano derecha o la izquierda. Una vez comience el *trial*, en la parte superior aparecerá el objetivo (derecha o izquierda), además de un marcador en forma de círculo blanco que rodeará a la imagen de la mano del lado del objetivo. La mitad de la barra del lado contrario al objetivo se vuelve blanca. Cuando el usuario realice la MI de la mano correspondiente al objetivo de manera adecuada, conseguirá que tanto el cubo como el marcador de la barra y la flecha se muevan hacia ese lado. El cubo, conforme se va moviendo hacia un lado o hacia otro, va tomando el color que señala el marcador de la barra, como se puede ver en la Figura 4.16. Si el sujeto no lo realiza correctamente no ocurre nada, es decir, ni el marcador ni el cubo ni la flecha se moverán. El *trial* finaliza cuando se consigue que el marcador de la barra llegue al extremo de ésta consiguiendo un acierto; o también si expira el tiempo máximo del *trial* sin conseguirlo, lo que se considera un fallo. La tarea terminará al completar todos los *trials* que se hayan configurado.

La tarea 3, al comenzar, presenta una barra en blanco colocada verticalmente a cada lado de la pantalla. Encima de las dos barras aparece una imagen de una mano derecha en la de la derecha y una



FIGURA 4.16: Imágenes que muestran el desarrollo de la Tarea 2.



FIGURA 4.17: Imágenes que muestran el desarrollo de la Tarea 3.

imagen de una mano izquierda en la de la izquierda. En el centro de la pantalla hay una bola gris que hace el papel del jugador. Cuando comienza el *trial*, se indica en la parte superior de la pantalla el objetivo de dicho *trial* (derecha o izquierda). Además, se marca tanto la barra como la mano que está encima de color amarillo. Cuando el usuario realice la MI de la mano correspondiente de forma correcta, la bola se moverá hacia ese lado, rellenando la barra del lado del objetivo. La barra de la derecha se irá llenando de azul y la de la izquierda de rojo, como aparece en la Figura 4.17. Si por el contrario lo realiza de manera incorrecta, la bola se moverá hacia el lado contrario al objetivo, llenando la barra de ese lado de color gris. El *trial* finaliza cuando la barra correcta se llena del todo, tomándolo como un acierto; o cuando expira el tiempo del *trial* sin conseguirlo, lo que se considera un fallo. La tarea terminará cuando se hayan completado todos los *trials*.

Cuando se inicia la tarea 4, en la pantalla aparece: i) un campo de fútbol con una portería en el extremo derecho; y ii) un campo de baloncesto con una canasta en el extremo izquierdo. Cuando comienza el *trial*, en la parte superior de la pantalla aparecerá el objetivo: mete un gol o encesta. Si el objetivo es derecha, es decir, “mete un gol”, en el centro de la pantalla aparecerá un balón de fútbol. Si el objetivo es izquierda, es decir, “encesta”, aparecerá un balón de baloncesto en el centro de la pantalla, como se observa en la Figura 4.18. Los balones se mueven tanto hacia la derecha como hacia la izquierda, según los valores de salida de BCI2000. El *trial* finaliza cuando el balón de fútbol llega al extremo derecho del campo de fútbol, produciéndose el efecto de meter un gol; o cuando el balón de baloncesto llega al extre-

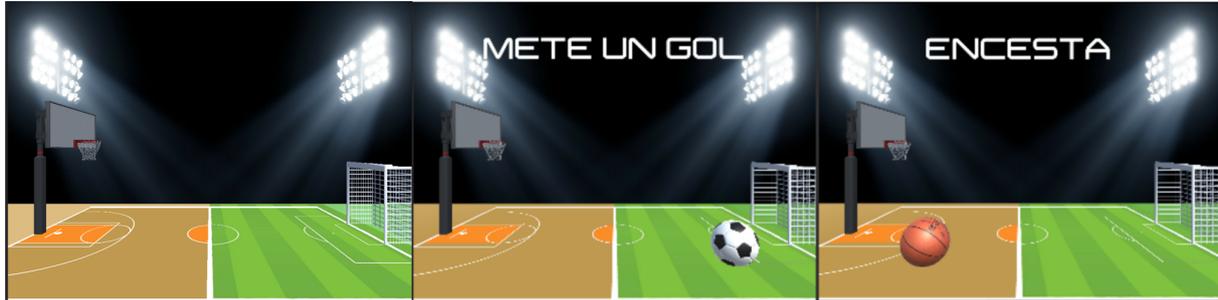


FIGURA 4.18: Imágenes que muestran el desarrollo de la Tarea 4.



FIGURA 4.19: Imágenes que muestran el desarrollo de la Tarea 5

mo izquierdo del campo de baloncesto, produciéndose el efecto de meter una canasta. En ambos casos se toma como un acierto. También se da por terminado si lo anterior no se consigue y el tiempo máximo de *trial* expira, considerándolo como un fallo. La tarea terminará cuando se hayan completado todos los *trials*.

La tarea 5 comienza con la aparición en la pantalla de un camino, por el cual va andando el personaje, que comienza situado en el centro del mismo. El personaje está representado por una mujer que se irá moviendo hacia la derecha o hacia la izquierda, según sean las señales de control del usuario. Esta vez no aparecerá en la parte superior de la pantalla el objetivo del *trial*. En vez de eso, se verá en la parte superior derecha un contador de aciertos y fallos. En el camino irán surgiendo una serie de obstáculos, considerando cada obstáculo un *trial*. Dichos obstáculos podrán presentarse en el centro, en el extremo derecho o en el extremo izquierdo del camino, como se observa en la Figura 4.19. El *trial* se considera acertado si se ha conseguido esquivar el obstáculo. En cambio, se considerará un fallo si el personaje se choca con el obstáculo, lo que producirá una pausa. En dicha pausa la figura del personaje parpadeará junto al obstáculo. Después de la pausa producida por el fallo, el personaje se sitúa automáticamente en el centro del camino, como cuando se comienza la tarea. Cada vez que se produce un acierto, la velocidad del personaje aumenta una cantidad igual a la configurada en las Preferencias. En cambio, si se produce un fallo, puede o no disminuir esa cantidad, dependiendo de si está activa la variable correspondiente en el menú de Preferencias. La tarea termina cuando expira el tiempo establecido en el menú de Preferencias.

RESULTADOS

5.1 SUJETOS Y TAREAS

La aplicación ha sido evaluada por parte de cuatro sujetos varones sanos, cuyas edades están comprendidas entre los 23 y los 25 años (24 ± 1). Cada usuario ha realizado tres sesiones de entrenamiento cognitivo, de aproximadamente 90 minutos. Durante todas las sesiones, se han utilizado los mismos electrodos para registrar el EEG que se indican en la Figura 4.2, así como 256 Hz de frecuencia de muestreo. A continuación, se describen el objetivo y las especificaciones de cada sesión, que se realizan en días distintos para no fatigar a los usuarios y darles tiempo para asimilar los posibles cambios en sus funciones cognitivas.

5.1.1 PRIMERA SESIÓN

En la primera sesión, antes de comenzar el entrenamiento cognitivo, se realiza un registro basal del EEG del usuario con el sujeto relajado y los ojos cerrados. Ese registro tiene una duración de 2 minutos. A continuación, se le explica el funcionamiento de la aplicación, diciéndole que durante la realización de las tareas esté lo más quieto posible. Para la realización de las tareas, deberá imaginarse el movimiento de la mano derecha o de la mano izquierda, según se le indique. Se comienza con la realización de la tarea 1 en modo adaptación, lo que quiere decir que, según complete cada *trial*, el *offset* y la ganancia del normalizador se modificarán para adaptarse a la actividad cerebral del sujeto. Una vez realizada la adaptación se volverá a realizar la tarea 1 con los valores del *offset* y de la ganancia resultantes, manteniéndolos constantes. Si el supervisor de las pruebas observa que la velocidad del *feedback* no es la suficiente, o que el *offset* de una de las señales es demasiado alto o demasiado bajo, los modificará para intentar mejorar la realimentación visual.

En esta sesión deberán realizarse 70 *trials* de la tarea 1, 50 *trials* de la tarea 2 y 20 *trials* de las tareas 3 y 4. Los *trials* duran 10 segundos cada uno. El tiempo entre *trials* es de 5 segundos. Se realiza varias veces cada tarea para permitir descansar a los usuarios. Dichas tareas están compuestas por 20 *trials*. En

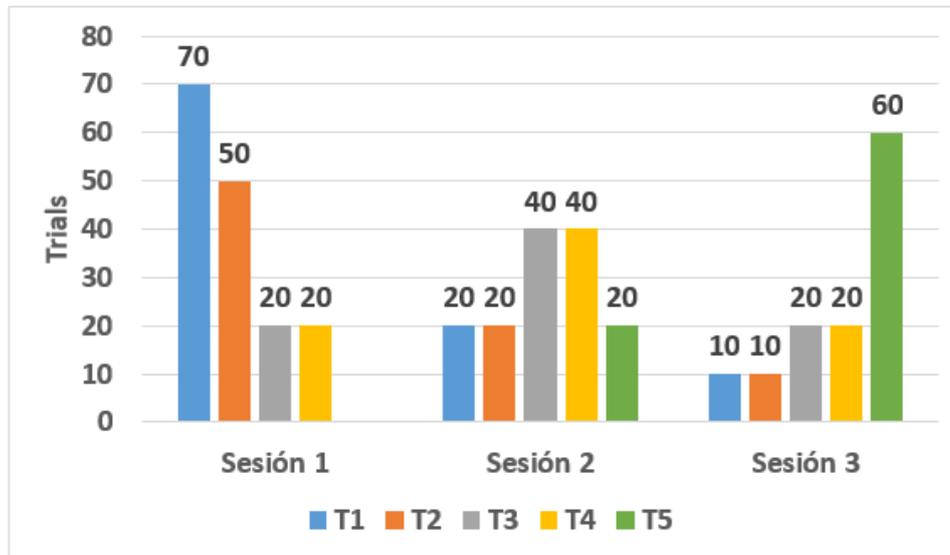


FIGURA 5.1: Representación gráfica de los *trials* que componen cada una de las sesiones de entrenamiento cognitivo.

la Figura 5.1 aparece un diagrama con los *trials* de cada tarea en las tres sesiones. Si se observa que el sujeto consigue los objetivos en un tiempo bastante inferior a los 10 segundos, se reduce la duración de los *trials* y se aumenta el número de éstos.

5.1.2 SEGUNDA SESIÓN

En la segunda sesión, al igual que en la primera, se realiza al comienzo una etapa de adaptación. Esta vez en la tarea 3, que posee *feedback* positivo y negativo. Una vez adaptados los valores, se irán modificando por el supervisor bajo su propio criterio. Cuando se ha realizado un par de veces la tarea 3, ahora compuesta por seis *trials* de 10 segundos, se pasa a realizar una vez la tarea 1 y otra la tarea 2, compuestas de 20 *trials*, como en la primera sesión. A continuación, se realizan tres veces las tareas 3 y 4. Para finalizar, deberán completarse 10 minutos de la tarea 5.

5.1.3 TERCERA SESIÓN

La tercera sesión es la última que compone el entrenamiento y está centrada en la tarea 5, suponiendo que el usuario ya posee cierto control de los SMR después de las dos sesiones de entrenamiento anteriores. Para comenzar, se adaptan los valores del normalizador con la tarea 3 y 4. Una vez adaptados, se realizan las cuatro primeras tareas con los valores constantes y, a continuación, se pasa a la tarea 5. Para evitar el cansancio del usuario, se realizan varias tandas de 5 minutos de dicha tarea. Una vez completadas todas las tareas, se realiza un registro basal del EEG del usuario, con el sujeto relajado y con los ojos cerrados, con una duración de 2 minutos.

TABLA 5.1: Resultados obtenidos por cada usuario en la tres sesiones de entrenamiento cognitivo.

U	SESIÓN 1					Media	SD
	T1	T2	T3	T4	T5		
U01	81,66 %	83,75 %	60,00 %	60,00 %	-	71,35 %	11,38 %
U02	76,66 %	88,75 %	70,00 %	55,00 %	-	72,60 %	12,18 %
U03	75,00 %	76,00 %	70,00 %	60,00 %	-	70,25 %	6,34 %
U04	86,66 %	97,50 %	50,00 %	30,00 %	-	66,04 %	27,26 %
U	SESIÓN 2					Media	SD
	T1	T2	T3	T4	T5		
U01	90,00 %	80,00 %	52,50 %	50,00 %	70,00 %	68,50 %	15,46 %
U02	100,00 %	80,00 %	41,66 %	55,00 %	40,00 %	63,33 %	23,26 %
U03	75,00 %	80,00 %	30,00 %	32,50 %	55,00 %	54,50 %	20,76 %
U04	55,00 %	100,00 %	60,00 %	40,00 %	55,00 %	62,00 %	20,15 %
U	SESIÓN 3					Media	SD
	T1	T2	T3	T4	T5		
U01	80,00 %	80,00 %	50,00 %	45,00 %	55,00 %	62,00 %	15,03 %
U02	90,00 %	80,00 %	50,00 %	40,00 %	55,00 %	63,00 %	18,87 %
U03	80,00 %	70,00 %	30,00 %	35,00 %	65,00 %	56,00 %	19,85 %
U04	90,00 %	90,00 %	40,00 %	50,00 %	55,00 %	65,00 %	20,98 %

5.2 ANÁLISIS ONLINE

Antes de comenzar la primera sesión, se dieron una serie de instrucciones y consejos para realizar el entrenamiento de la mejor manera posible. Es importante que los sujetos estén relajados y no realicen movimientos bruscos con la cabeza. Además, se indica la imaginación del movimiento que se utiliza para generar correctamente los SMR, incidiendo en el hecho de que cada sujeto es libre de imaginar el movimiento con el que se encuentre más cómodo, con el fin de recrear la sensación que les produce cuando lo realizan de verdad.

En la Tabla 5.1 están desglosados los resultados de cada usuario en las sesiones de entrenamiento cognitivo. En la primera sesión se ha observado que, después de la etapa de adaptación, en las dos primeras tareas de *feedback* positivo, todos los usuarios han alcanzado precisiones superiores al 75 %. En cambio, en las tareas 3 y 4 casi todos los usuarios acertaban prácticamente todos los *trials* de uno de los lados, sin embargo los del otro no. El rango de aciertos de estas tareas disminuye al 30-70 % de precisión. Además, en las primeras tareas los valores de adaptación han sido modificados manualmente para intentar mejorar el rendimiento de los usuarios. En todos los casos, el valor del *offset* de la señal de control que indica la MI de la mano derecha era superior al de la mano izquierda.

En la segunda sesión se incluyen tareas con *feedback* positivo y negativo. Se ha comprobado que, aún con la etapa de adaptación, los usuarios no consiguen el objetivo de los *trials* de la manera deseada. Se

TABLA 5.2: Media de los resultados obtenidos por cada usuario en la tres sesiones de entrenamiento cognitivo.

U	Media					Media Total	SD
	Feedback Positivo		Feedback Positivo y Negativo				
	T1	T2	T3	T4	T5		
U01	83,88 %	81,25 %	52,50 %	50,00 %	55,00 %	65,21 %	15,66 %
U02	88,88 %	82,92 %	53,88 %	50,00 %	47,50 %	65,86 %	19,52 %
U03	76,66 %	75,33 %	36,66 %	35,83 %	57,50 %	56,32 %	19,13 %
U04	77,22 %	95,83 %	50,00 %	40,00 %	52,50 %	63,87 %	22,93 %

intentan modificar los valores de adaptación, sin embargo los resultados son muy irregulares. Si se aumenta mucho el *offset* de uno de los canales, se provoca que el objeto siempre se vaya hacia un lado. Si se consigue que no se vaya siempre hacia el mismo lado, el comportamiento es muy irregular. En una tarea puede que se acierten más de la mitad de los *trials*, y en la siguiente el usuario no consiga controlarlo. Incluso en algún *trial*, el objeto se mueve hacia el lado contrario del deseado.

Ante los resultados de la segunda sesión, las tareas propuestas para la tercera se acortan. Al finalizar cada tarea, se observa el mismo comportamiento de la segunda sesión. El rendimiento es errático y los sujetos no consiguen controlar las tareas 3, 4 y 5, es decir, las que poseen *feedback* positivo y negativo. No obstante, se alcanzan precisiones superiores al 55 % en las tareas 1 y 2 con *feedback* positivo. La media y la desviación típica (SD, *standard deviation*) de los resultados de cada usuario está recogida en la Tabla 5.2. Se puede observar que el sujeto U02 es el que ha alcanzado una precisión global más alta teniendo en cuenta todas las tareas. En cambio, el sujeto U03 es el que ha obtenido una precisión global más baja, con prácticamente un 10 % menos de aciertos que el U02.

5.3 ANÁLISIS OFFLINE

Una vez finalizadas las sesiones, se ha realizado un estudio *offline* para determinar si el entrenamiento con *neurofeedback* produce cambios significativos en las bandas de frecuencia en las cuales se encuentran los SMR. Para dicho estudio se ha implementado un *script* en MatLab que se puede encontrar en el Anexo A. En dicho *script* se calcula, para cada usuario, la potencia relativa en tres bandas de frecuencia, todas de 3Hz de ancho de banda y centradas en 12, 18 y 21 Hz. Dicha potencia se calcula para cada uno de los 8 canales de entrada. Es decir, de cada usuario se calculan dos matrices de 3x8, que contienen la RP de cada canal en las tres bandas de frecuencia mencionadas, antes y después del entrenamiento cognitivo.

Para el cálculo de la RP, se toma primero la señal registrada y se filtra dejando solo lo que encierra la banda 0,5-60 Hz, mediante un filtro paso banda FIR con ventana Hamming de orden 1000. A continuación, se realiza el cálculo de la PSD mediante el método de Welch, con una ventana Hamming de 2^{13}

TABLA 5.3: Tests estadísticos según las características de los datos.

Nivel de medida	Características de las muestras					Correlación
	1 grupo	2 Grupos		Más de 2 Grupos		
		Independientes	Dependientes	Independientes	Dependientes	
Catagórica o Nominal	χ^2 o binomial	χ^2	Macnarmar's χ^2	χ^2	Cochran's Q	—
Rango o Ordinal	—	Mann Whitney U	Test de los rangos con signo de Wilcoxon	Krusal Wallis H	Friedman's ANOVA	Spearman's rho
Paramétrica (intervalo y radio)	z test o t test	t test entre grupos	t test sin grupos	1 way ANOVA entre grupos	1 way ANOVA	Pearson's r
		Factorial (2 ways) ANOVA				—

muestras, una DFT de 2^{13} muestras y un solapamiento del 90%. Una vez calculada la PSD, se normaliza y se extrae la RP de cada canal en las tres bandas de interés, dividiendo el área encerrada bajo la curva en cada banda por el área total.

Una vez se han calculado todas las potencias antes y después del entrenamiento cognitivo (dos matrices de 3×8), se aplica un test estadístico. En este caso, el test elegido es el Test de los rangos con signo de Wilcoxon, debido a la naturaleza de los datos: univariados, dependientes, continuos y no normales. En la Tabla 5.3 se pueden observar diferentes tests estadísticos dependiendo de las características de las muestras. El test de los rangos con signo de Wilcoxon normalmente se utiliza para comparar las diferencias entre dos muestras de datos de la misma persona, tomados antes y después de un acontecimiento (en este caso el NFT). Este test compara si las diferencias entre los datos siguen una distribución simétrica entorno a un valor, basándose en las diferencias positivas y negativas que surgen entre ellos. Para este análisis se utiliza el test en cada una de las 3 bandas de interés. Es decir, compara las dos matrices de 3×8 fila a fila para obtener 3 **p-valores** de cada sujeto: uno para la banda de 12 Hz, otro para la de 18 Hz y otro para la de 21 Hz. El *p*-valor indica que existen diferencias significativas si es inferior o igual que 0,05. En caso contrario determina que no se han producido cambios importantes. Dado que se han comparado los valores de RP antes y después del entrenamiento cognitivo, los *p*-valores indican si existen diferencias en el EEG de los usuarios causadas por dicho entrenamiento.

Después de haber realizado el registro del EEG basal al completar la última sesión, se llevo a cabo el análisis *offline* anteriormente descrito. En todos los usuarios se produjo un aumento de la potencia relativa en las bandas de interés, menos en uno. A continuación, se detallan los datos obtenidos en dicho análisis para cada usuario.

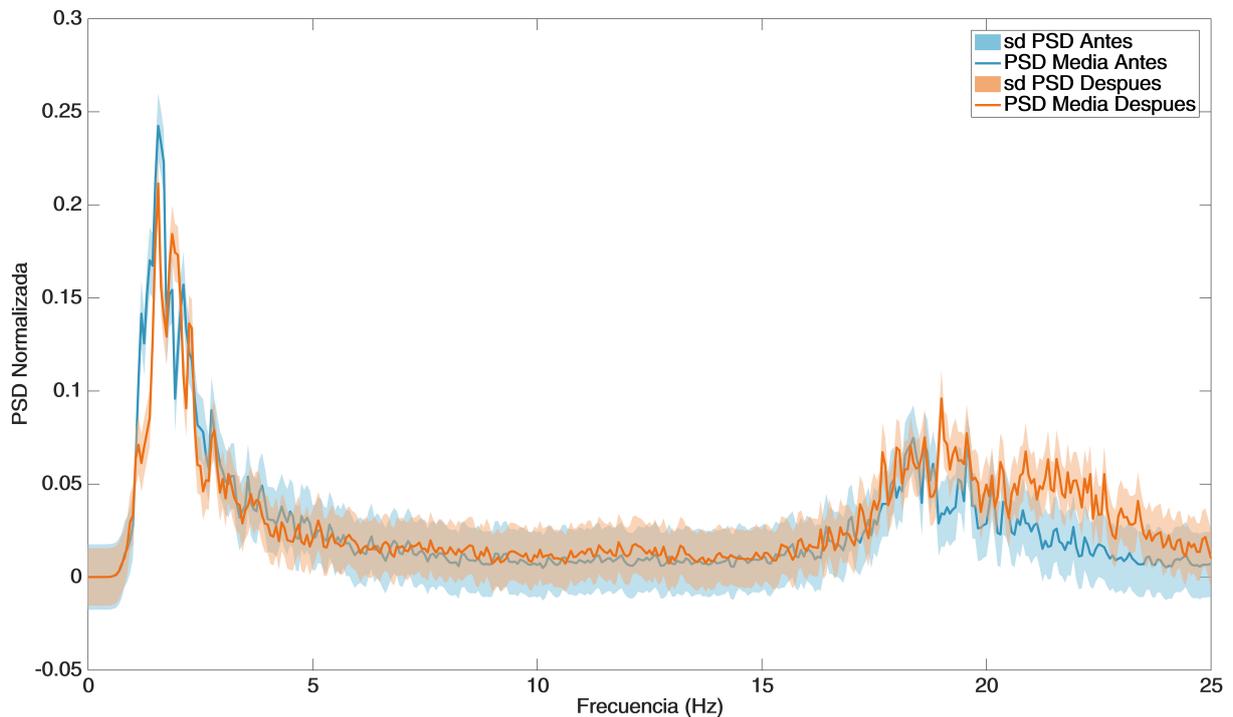


FIGURA 5.2: Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario01, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.

5.3.1 USUARIO 01

El usuario 01 presentó diferencias significativas en la banda centrada en los 21 Hz. Esas diferencias muestran un aumento en la potencia relativa de dicha banda después del entrenamiento. El p -valor obtenido es de 0,0234, y se localiza en la zona central y frontal. A continuación, se detalla dónde se ha producido el aumento y en qué canales se aprecian diferencias.

En la Figura 5.2 se puede observar la media de la PSD y SD, tanto antes del entrenamiento (azul) como después (naranja). Al analizar con más detenimiento la banda centrada en 21 Hz, se puede observar un claro aumento de la potencia después del entrenamiento. En la banda de 18 Hz también aumenta la potencia, pero más levemente. Con respecto a la banda de 12 Hz, si tenemos en cuenta la desviación, el aumento no se aprecia claramente.

A continuación, en la Figura 5.3 se puede observar una representación espacial de la diferencia entre las potencias relativas de cada banda, antes y después del entrenamiento. Para poder discernir si el entrenamiento ha producido un aumento o una disminución en la RP de las bandas, la diferencia se calcula restando a las RP antes del NFT, las de después del mismo. Si los valores de la diferencia son positivos, se ha producido un aumento de la potencia; si son negativos, una disminución; y si son próximos a cero,

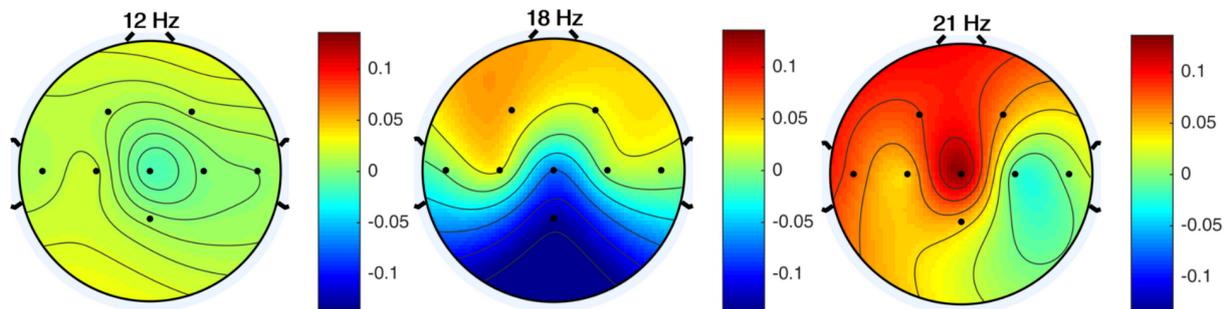


FIGURA 5.3: Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.

la potencia se ha mantenido prácticamente sin cambios. Al analizar la Figura 5.3, se observa cómo en la banda de los 12 Hz no hay prácticamente diferencias, dado que los valores están todos cercanos al cero. En la de 18 Hz hay un ligero aumento en la parte frontal, más evidente en el lado izquierdo que en el derecho. Finalmente, en la banda de 21 Hz se produce un aumento elevado en la parte central y frontal, también más pronunciado en el lado izquierdo.

En la Figura 5.4 se representa la PSD antes y después del entrenamiento en cada uno de los canales por separado. Las bandas de frecuencia de interés están sombreadas para poder apreciar mejor las diferencias. Los canales donde se aprecia un mayor aumento de la potencia en la banda de 21 Hz son F3, F4, Cz, T7 y C3. En la banda centrada en los 18 Hz se observa un mayor aumento en los canales F3, F4 y C4, y para la banda de los 12 Hz se intuye que hay aumento, sin embargo no se va claramente en ninguno de los canales.

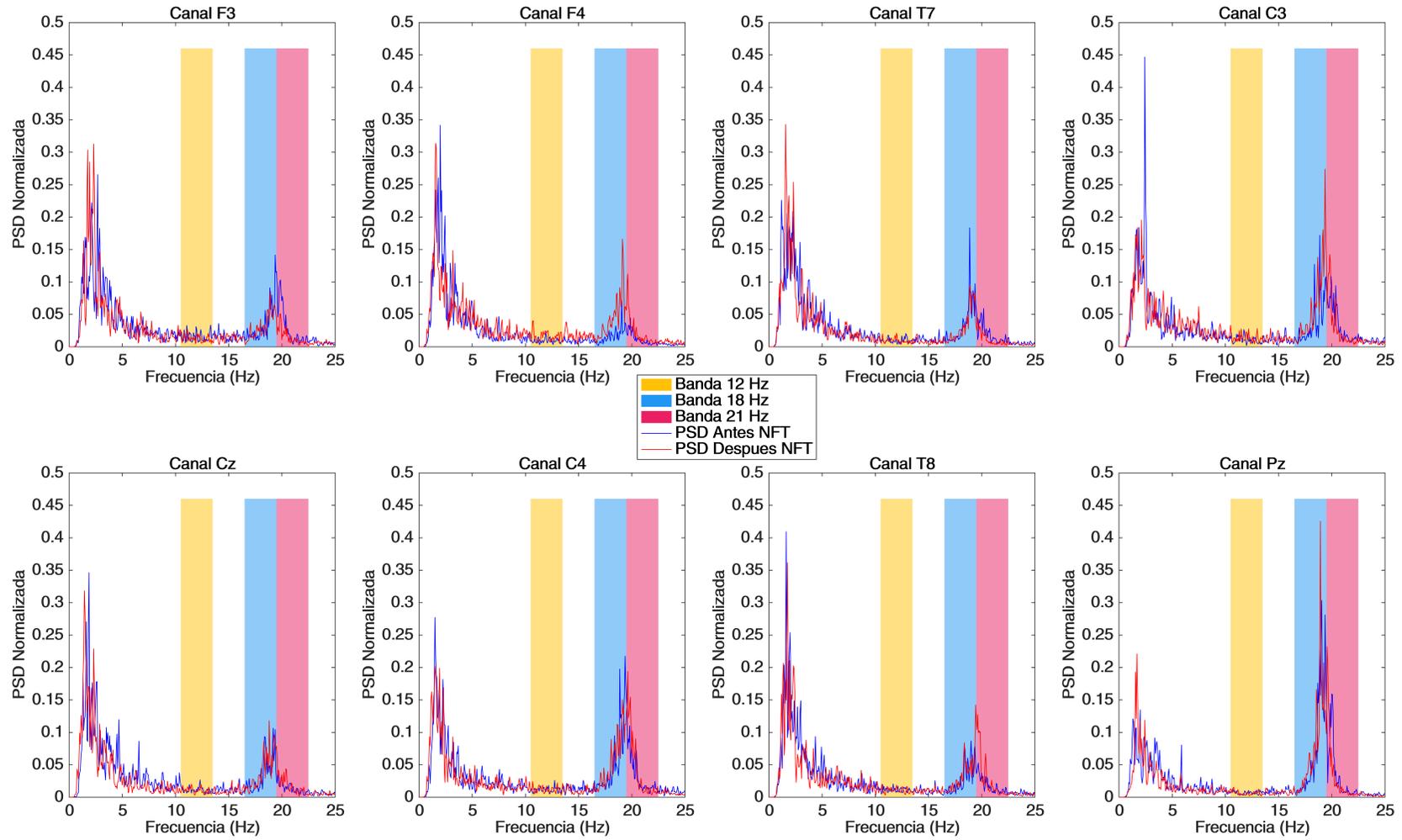


FIGURA 5.4: Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.

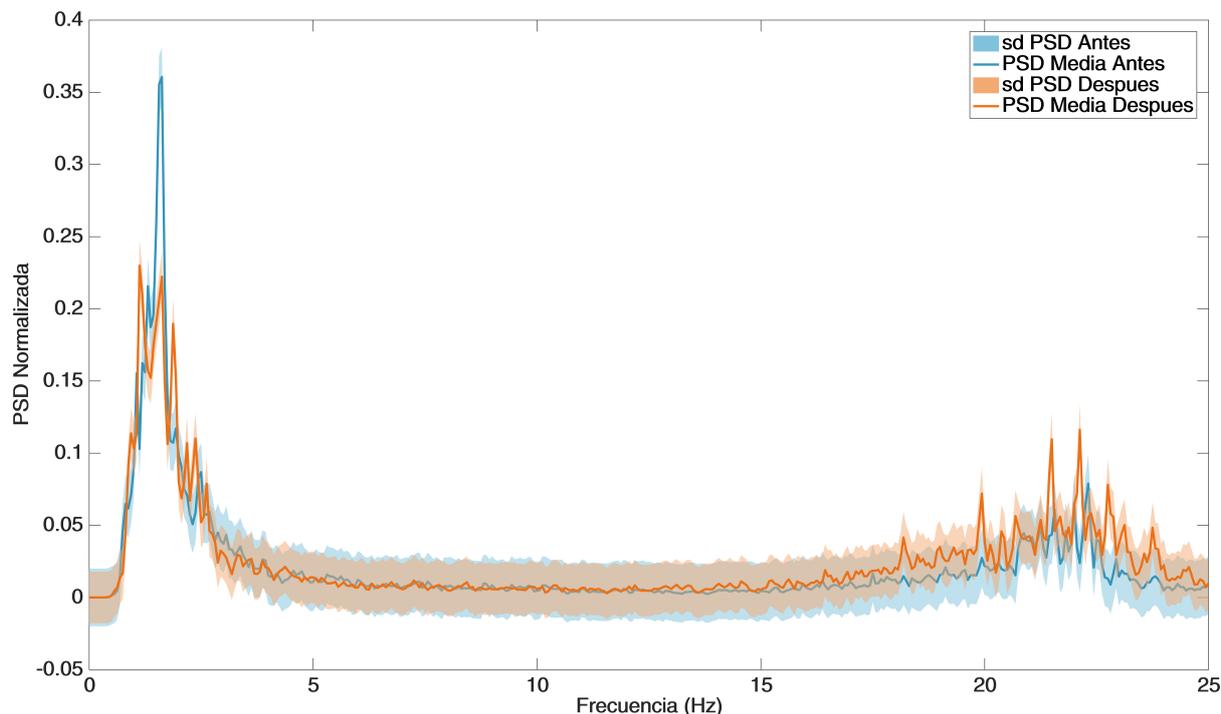


FIGURA 5.5: Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario02, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.

5.3.2 USUARIO 02

El usuario 02 presentó diferencias significativas en la banda centrada en los 18 Hz. Esas diferencias muestran un aumento en la potencia relativa de dicha banda después del entrenamiento. El p -valor obtenido es de 0,0391, y se localiza en la zona central y lateral derecha. A continuación, se detalla dónde se ha producido el aumento y en qué canales se aprecian diferencias.

En la Figura 5.5 se puede observar la media de la PSD y su SD, tanto antes del entrenamiento (azul) como después (naranja). Al analizar con más detenimiento la banda centrada en 18 Hz, se puede apreciar un claro aumento de la potencia después del entrenamiento, donde prácticamente no hay solapamiento entre las PSDs. En la banda de 21 Hz también aumenta la potencia, sin embargo no se aprecia tan claramente debido al solapamiento. Con respecto a la banda de 12 Hz no se consiguen observar diferencias significativas.

A continuación, en la Figura 5.6 se puede observar una representación espacial de la diferencia entre las potencias relativas de cada banda, antes y después del entrenamiento. Para poder discernir si el entrenamiento ha producido un aumento o una disminución en la RP de las bandas, la diferencia se calcula restando a las RP antes del NFT, las de después del mismo. Si los valores de la diferencia son positivos,

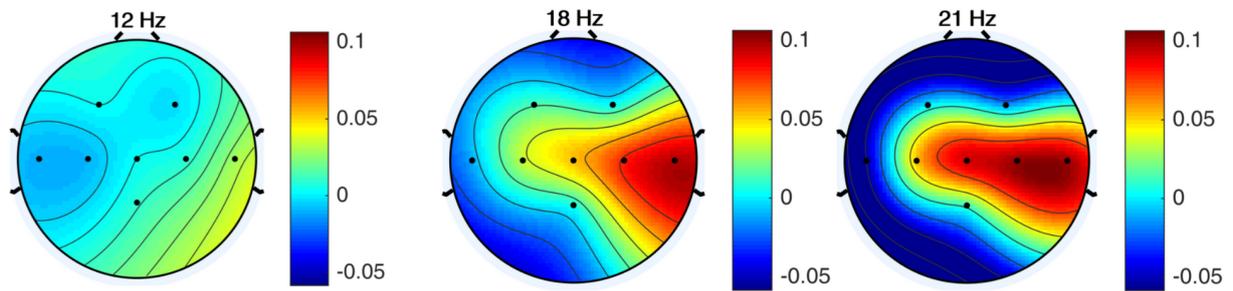


FIGURA 5.6: Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.

se ha producido un aumento de la potencia; si son negativos, una disminución; y si son próximos a cero, la potencia se ha mantenido prácticamente sin cambios. Al analizar la Figura 5.6, se observa como en la banda de los 12 Hz no hay prácticamente diferencias, dado que los valores están todos cercanos al cero. En la de 18 Hz, el aumento se produce sobre todo en la región lateral derecha y central. En la banda de 21 Hz, aunque no han surgido diferencias significativas, si podemos apreciar que la potencia relativa de la banda, ha aumentado en la región central y lateral derecha.

En la Figura 5.7 se representa la PSD antes y después del entrenamiento en cada uno de los canales por separado. Las bandas de frecuencia de interés están sombreadas para poder apreciar mejor las diferencias. Los canales donde se aprecia un mayor aumento de la potencia en la banda de 18 Hz son Cz, C4, T8 y C3. En la banda centrada en los 21 Hz se observa un mayor aumento en los canales C3, T8, Cz y C4. Para la banda de los 12 Hz se intuye que hay aumento en la mayoría de los canales, sin embargo no se va claramente debido al solapamiento.

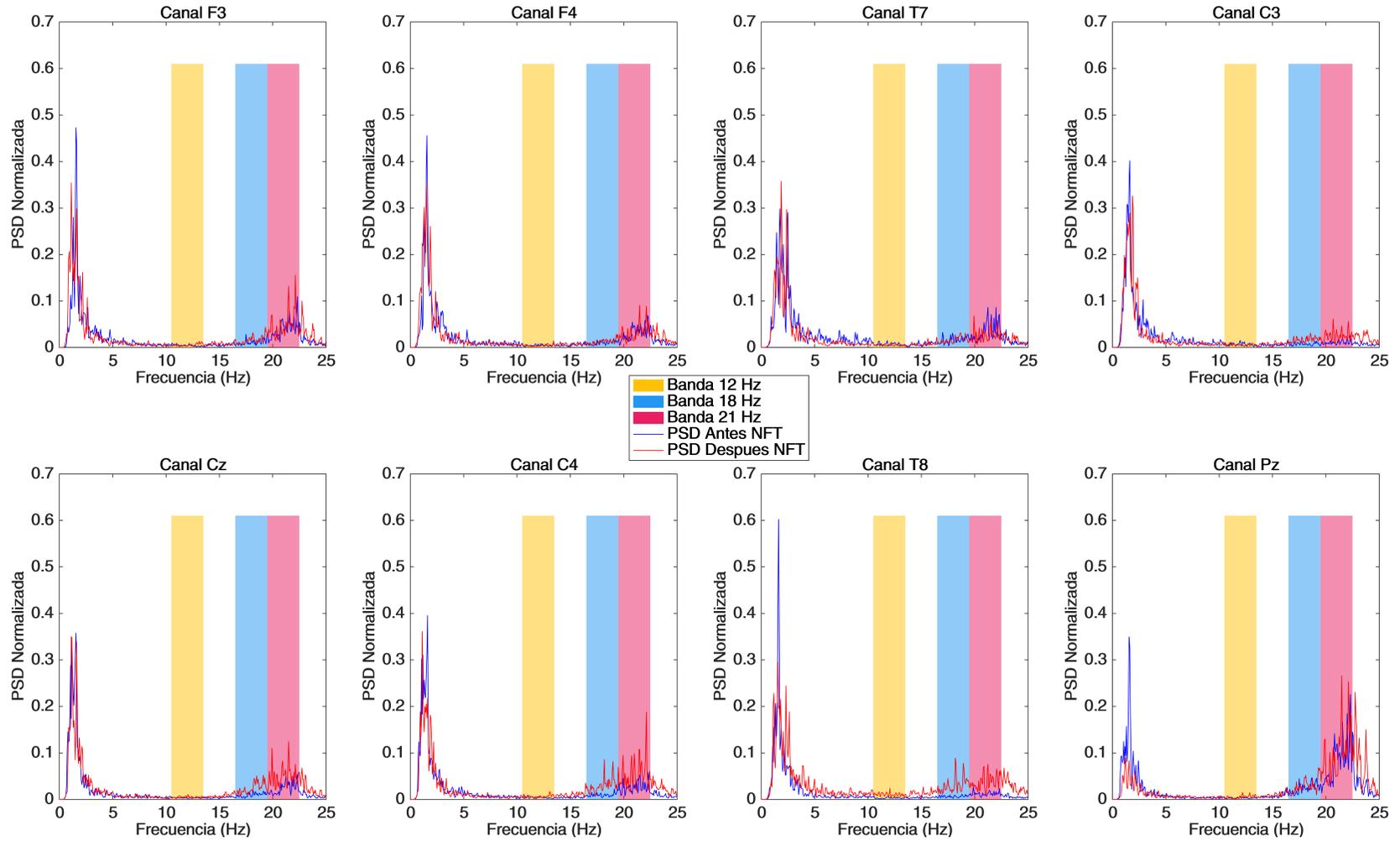


FIGURA 5.7: Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.

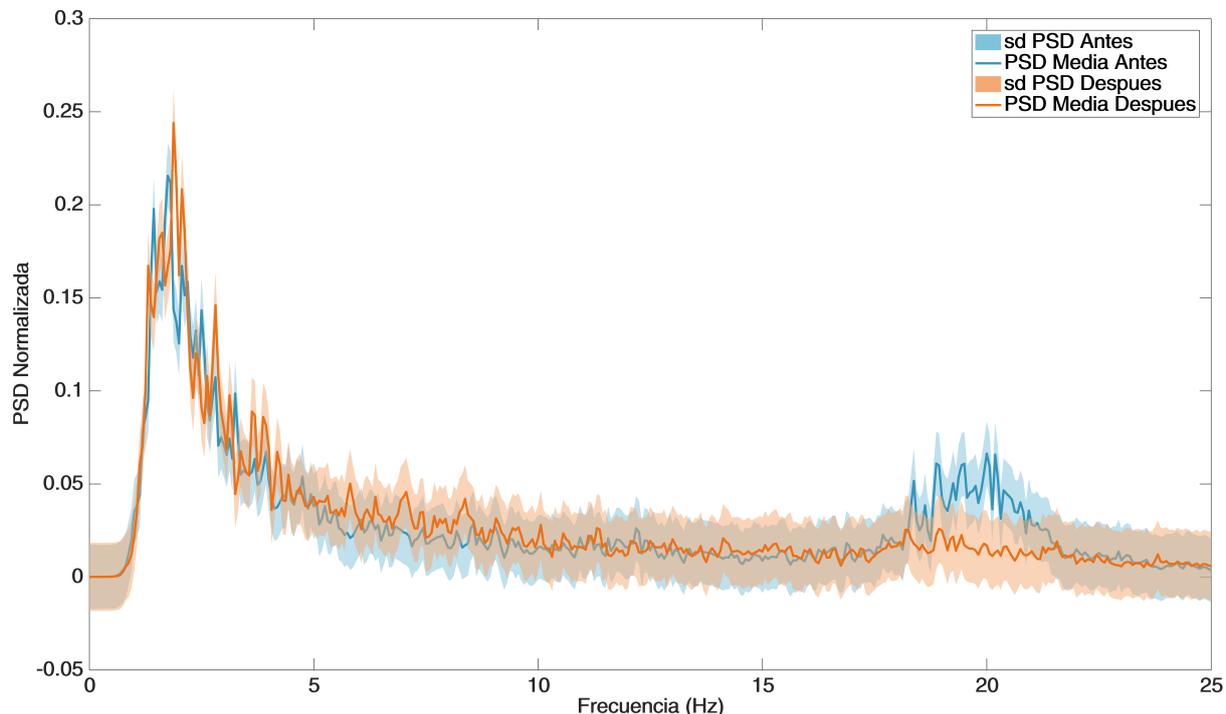


FIGURA 5.8: Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario03, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.

5.3.3 USUARIO 03

El usuario 03 presentó diferencias significativas en las bandas centradas en los 18 y en los 21 Hz. Esas diferencias muestran una disminución en la potencia relativa de dichas bandas después del entrenamiento, en contraposición a lo que ocurre con los demás usuarios. El p -valor obtenido en ambas es 0,0078 y se localiza en la zona central y lateral derecha. A continuación, se detalla dónde se ha producido el aumento y en qué canales se aprecian diferencias.

En la Figura 5.8 se puede observar la media de la PSD y su SD, tanto antes del entrenamiento (azul) como después (naranja). Al analizar con más detenimiento las bandas centrada en 18 y 21 Hz se puede apreciar una clara disminución de la potencia después del entrenamiento, donde prácticamente no hay solapamiento entre las PSDs. En la banda de 12 Hz en cambio, no se puede distinguir a simple vista si la potencia aumenta o disminuye.

A continuación, en la Figura 5.9 se puede observar una representación espacial de la diferencia entre las potencias relativas de cada banda, antes y después del entrenamiento. Para poder discernir si el entrenamiento ha producido un aumento o una disminución en la RP de las bandas, la diferencia se calcula restando a las RP antes del NFT, las de después del mismo. Si los valores de la diferencia son positivos,

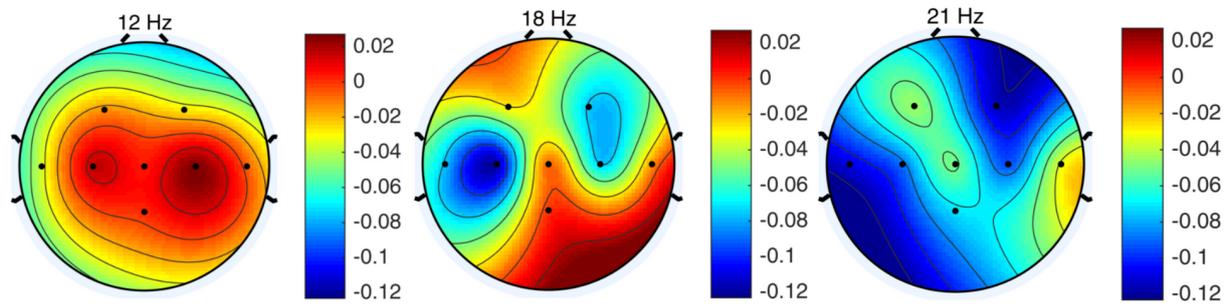


FIGURA 5.9: Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.

se ha producido un aumento de la potencia; si son negativos, una disminución; y si son próximos a cero, la potencia se ha mantenido prácticamente sin cambios. Al analizar la Figura 5.9, se observa como en la banda de los 12 Hz se ha producido un aumento muy leve de la potencia relativa en la zona centro-lateral. En la de 18 Hz, al igual que en la de 21 Hz, se produce una disminución de la potencia relativa. En la banda centrada en los 18 Hz esta disminución está localizada, sobre todo, en la región lateral izquierda.

En la Figura 5.10 se representa la PSD antes y después del entrenamiento en cada uno de los canales por separado. Las bandas de frecuencia de interés están sombreadas para poder apreciar mejor las diferencias. Los canales donde se aprecia una mayor disminución de la potencia en las bandas de 18 y 21 Hz son F3, F4, T7, C3 y C4. En la banda centrada en los 12 Hz se observa un mayor aumento en los canales C3 y C4.

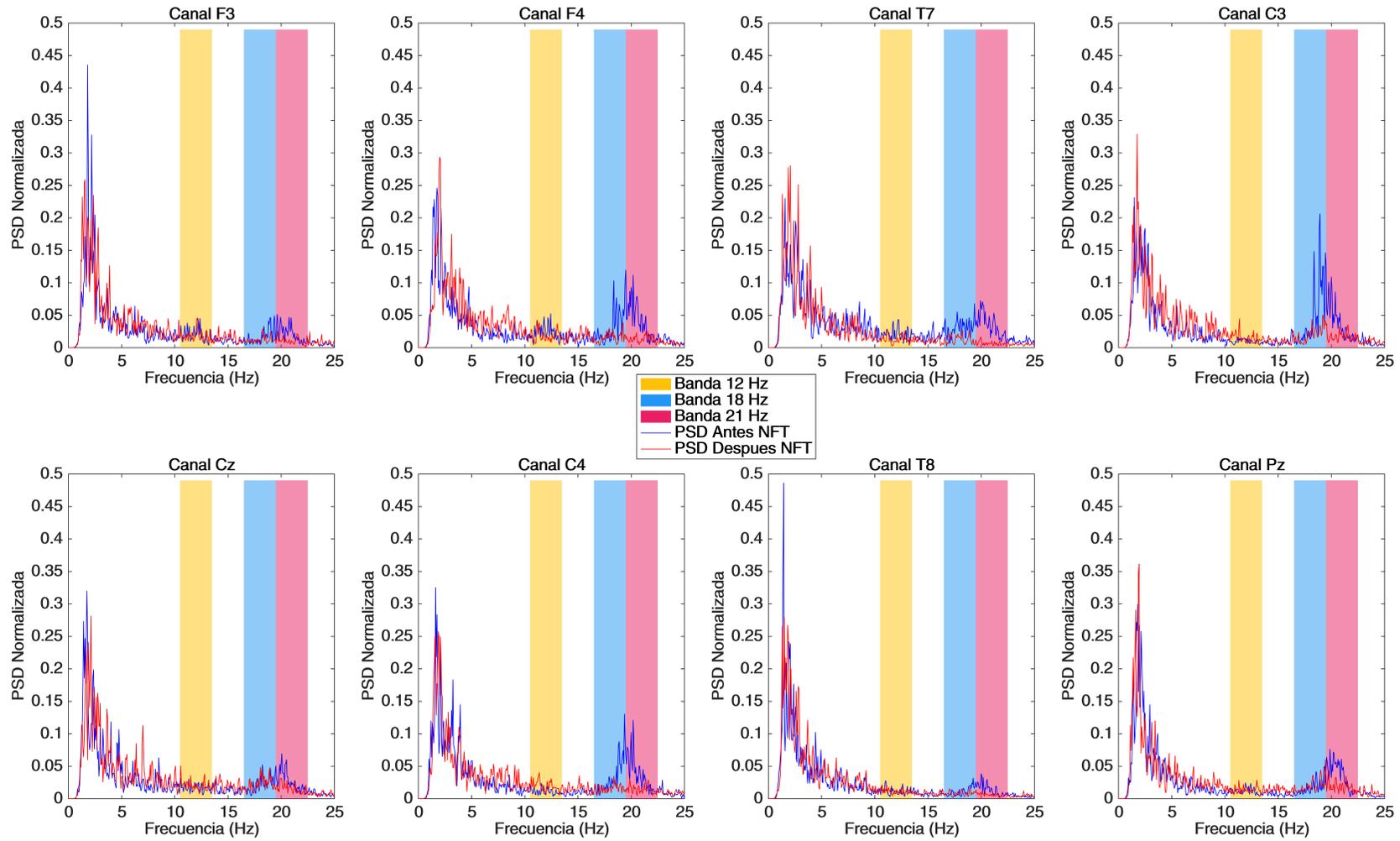


FIGURA 5.10: Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.

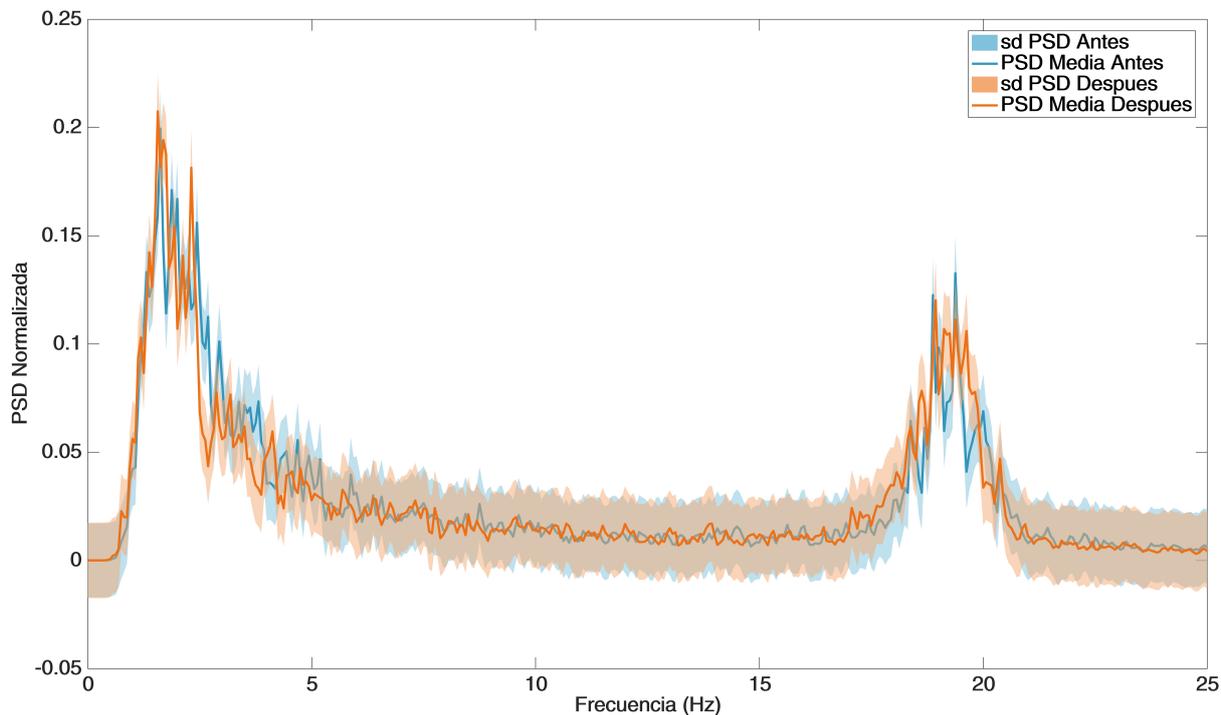


FIGURA 5.11: Representación gráfica de la media de la PSD de todos los canales con su SD del Usuario04, antes (azul) y después (naranja) del entrenamiento cognitivo. Se muestra la banda de 0-25 Hz para poder observar las bandas de interés con más detalle.

5.3.4 USUARIO 04

El usuario 04 presentó diferencias significativas en las bandas centradas en los 12 y en los 18 Hz. Esas diferencias muestran un aumento en la potencia relativa de dichas bandas después del entrenamiento. El p -valor obtenido en las bandas de 12 y 18 Hz son 0,0391 y 0,0078, respectivamente. La mayor parte localizadas en la región frontal derecha y lateral izquierda. A continuación, se detalla dónde se ha producido el aumento y en qué canales se aprecian diferencias.

En la Figura 5.11 se puede observar la media de la PSD y su SD, tanto antes del entrenamiento (azul) como después (naranja). En este caso, a simple vista es complicado apreciar si hay un aumento significativo en las bandas. Donde más se aprecian diferencias es en la banda centrada en 18 Hz, ya que las PSDs están menos solapadas.

A continuación, en la Figura 5.12 se puede observar una representación espacial de la diferencia entre las potencias relativas de cada banda, antes y después del entrenamiento. Para poder discernir si el entrenamiento ha producido un aumento o una disminución en la RP de las bandas, la diferencia se calcula restando a las RP antes del NFT, las de después del mismo. Si los valores de la diferencia son positivos, se ha producido un aumento de la potencia; si son negativos, una disminución; y si son próximos a cero,

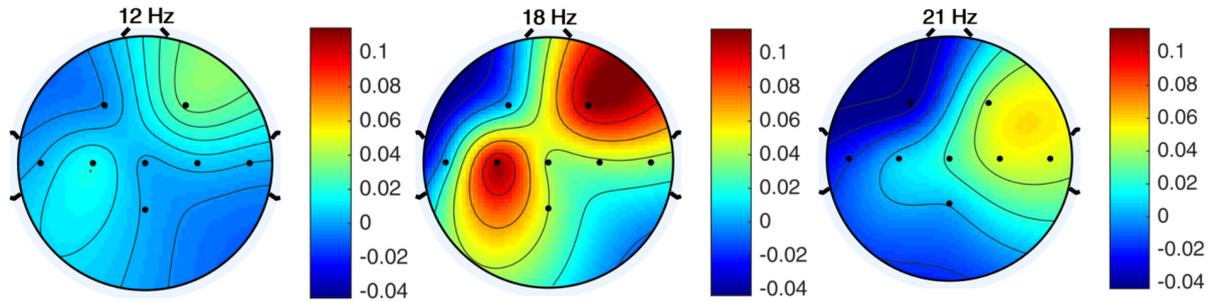


FIGURA 5.12: Representación espacial de la diferencia entre la potencia relativa de cada banda después del entrenamiento con la potencia relativa antes del entrenamiento.

la potencia se ha mantenido prácticamente sin cambios. Al analizar la Figura 5.12, se observa como en la banda de los 12 Hz se ha producido un aumento muy leve de la potencia relativa en la zona centro-lateral izquierda y en la frontal derecha. En la de 18 Hz, se produce un aumento de la potencia relativa, sobre todo en las mismas zonas que en la banda anterior. Por último, en la banda centrada en los 18 Hz se produce un aumento en la región lateral derecha.

En la Figura 5.13 se representa la PSD antes y después del entrenamiento en cada uno de los canales por separado. Las bandas de frecuencia de interés están sombreadas para poder apreciar mejor las diferencias. Los canales donde se aprecia un mayor aumento de la potencia en la banda de 12 Hz son F4 y C3. En la banda centrada en los 18 Hz se observa un mayor aumento en los canales F4, C3 y Pz. En la banda de 21 Hz el aumento se aprecia en los canales Cz, C4 y F4.

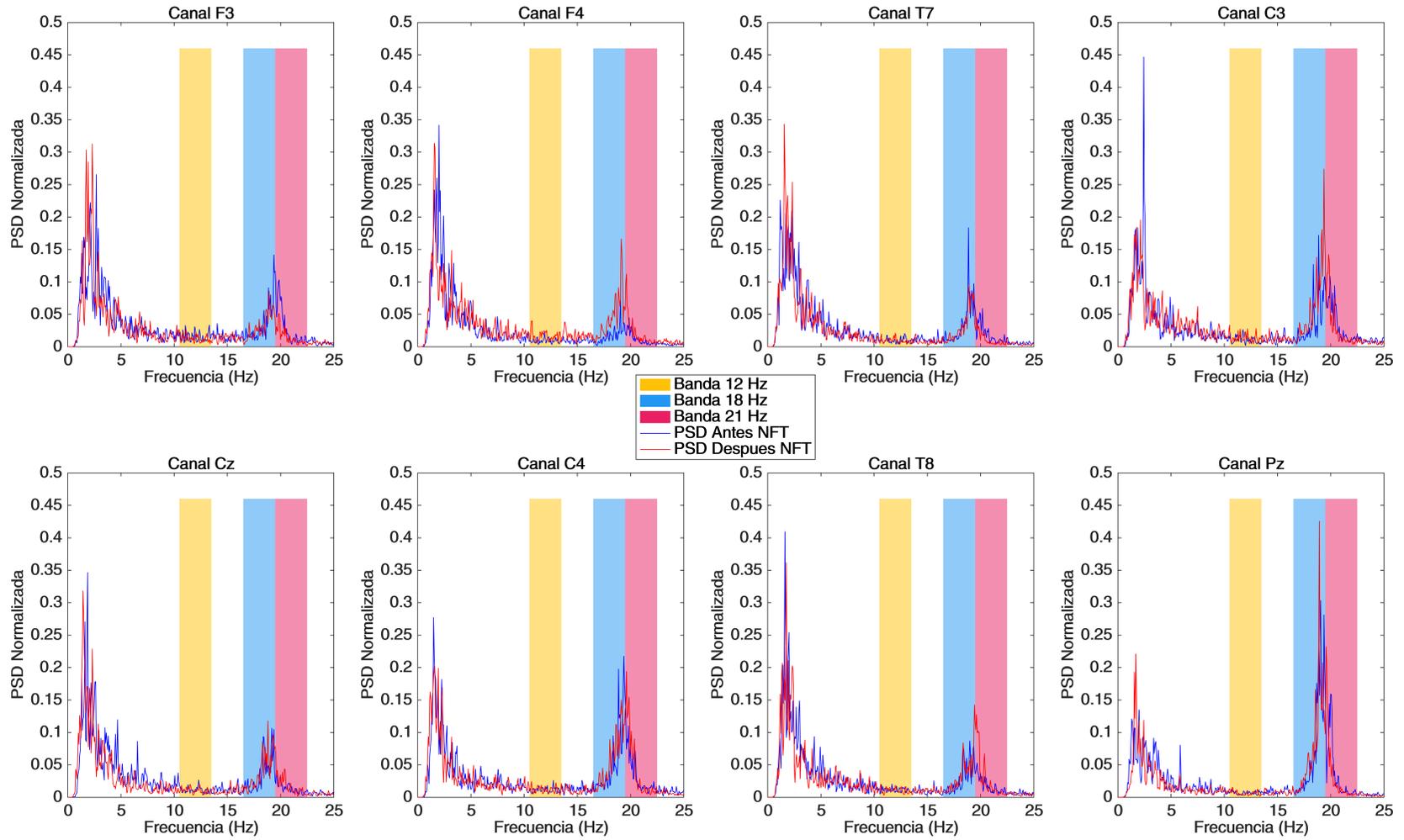


FIGURA 5.13: Representación gráfica de la PSD antes (azul) y después (rojo) del entrenamiento cognitivo en cada uno de los 8 canales.

TABLA 5.4: Tabla que refleja la magnitud del p -valor para cada usuario obtenido mediante el test de los rangos con signo de Wilcoxon, en cada una de las tres bandas de interés. Las diferencias significativas (p -valor $<0,05$) vienen marcadas con *.

Usuario	p-valor		
	12 Hz	18 Hz	21 Hz
U01	0,1953	0,8438	0,0234*
U02	0,5469	0,0391*	0,2500
U03	0,6400	0,0078*	0,0078*
U04	0,0391	0,0078*	0,1953

En la Tabla 5.4 se indican los p -valores de cada usuario, en las tres bandas de frecuencia de interés. Como ya se ha explicado, el p -valor se obtiene al comparar las RP de los 8 canales antes y después del NFT, mediante el test de los rangos con signo de Wilcoxon. Se puede observar que donde más diferencias significativas se encuentran es en la banda de los 18 Hz. Concretamente, tres de los cuatro usuarios han obtenido diferencias significativas en sus SMR después de completar el entrenamiento.

5.4 CUESTIONARIO DE SATISFACCIÓN

Una vez finalizadas las sesiones de entrenamiento, se entregó a los usuarios un cuestionario de satisfacción con el fin de recoger su experiencia personal y las sugerencias para mejorar la aplicación.

El cuestionario está formado por 9 afirmaciones tanto positivas como negativas, para conseguir que razonen las respuestas. Todas se deben valorar desde “totalmente en desacuerdo” hasta “totalmente de acuerdo”. Después se incluyen dos preguntas para determinar si el sujeto estaría dispuesto a volver a participar en las pruebas de una aplicación similar, y para que indique qué aspectos mejoraría de la aplicación actual. Dicho cuestionario se muestra en el Anexo B.

Las respuestas obtenidas con respecto a las 9 afirmaciones que forman el cuestionario se detallan en la Tabla 5.5. Para cuantificar las respuestas se ha calculado la media y la SD de cada afirmación. Esta estructura de cuestionario se conoce como escala Likert de 5 puntos. Los posibles valores de las respuestas son los siguientes:

1. Totalmente en desacuerdo.
2. En desacuerdo.
3. Neutral.
4. De acuerdo.

TABLA 5.5: Tabla que refleja le media y SD de los resultados del cuestionario.

	Cuestión	Puntuación	
		Media	SD
1	Me ha resultado interesante conocer y utilizar esta aplicación.	4,50	0,50
2	Las sesiones de entrenamiento me han resultado aburridas.	2,25	0,43
3	La interfaz del sistema es intuitiva y fácil de entender.	5,00	0,00
4	Las sesiones requieren demasiada concentración, lo que termina cansándome.	2,50	0,50
5	El feedback de las distintas tareas me gusta y me motiva para conseguir el objetivo.	4,50	0,50
6	La duración de las sesiones de entrenamiento me ha parecido excesiva.	2,00	0,71
7	He conseguido mejorar en el manejo de la aplicación después de las sesiones de entrenamiento.	3,50	1,12
8	La aplicación tiene poca utilidad y no me ha parecido interesante.	1,25	0,43
9	La adquisición de la actividad cerebral mediante el EEG con el gorro y electrodos activos es poco práctica.	2,50	0,50

5. Totalmente de acuerdo.

A la pregunta de si volvería a participar en un proyecto de características similares, todos los sujetos contestaron que sí. La pregunta restante tenía como intención recoger las sugerencias de los usuarios para mejorar la aplicación. El U01 subrayó que el implantar todo el sistema en la *Tablet*, sin la necesidad de disponer de un ordenador, mejoraría la portabilidad y simplicidad del sistema. El U03 indicó la necesidad de mejorar la adaptación del sistema, ya que no consiguió en ningún momento controlar las tareas con *feedback* positivo y negativo. Por último, el U04 recaló que, en su opinión, los objetivos de los *trials* deberían ser más aleatorios, ya que cuando el trial es impar, era capaz de predecir el objetivo del siguiente *trial*.

DISCUSIÓN

El objetivo de este capítulo es discutir los resultados presentados en el capítulo anterior. En primer lugar, se analizan los resultados obtenidos por los distintos sujetos en las sesiones de entrenamiento cognitivo. Posteriormente, se discuten los resultados obtenidos en los cuestionarios de satisfacción y las sugerencias de mejora propuestas por los usuarios. A continuación, se analizan los resultados obtenidos en el análisis *offline* de los datos. Finalmente, se realiza una pequeña comparación con los resultados de otros estudios y se determinan las limitaciones de la aplicación desarrollada.

6.1 ANÁLISIS ONLINE

En las tres sesiones de entrenamiento realizadas por los usuarios se han obtenido resultados similares. En las primeras dos tareas, todos los sujetos han obtenido muy buenos resultados al estar en el rango de 75-100 % de aciertos, como se puede observar en la Figura 5.1. Sin embargo, se ha puesto de manifiesto la limitada capacidad de adaptación del normalizador. En todas las sesiones, los parámetros de *offset* y ganancia del normalizador han tenido que ser modificados por el supervisor de los entrenamientos.

Debido a la mala adaptación, los resultados obtenidos en las tareas 3, 4 y 5 han sido inferiores a los deseados. Ni siquiera modificando los parámetros entre tareas manualmente se consiguió mejorar el rendimiento. Los usuarios obtuvieron resultados muy erráticos y no obtuvieron el control deseado en las 3 tareas mencionadas.

Los resultados obtenidos y la interacción que se ha producido en las sesiones con los parámetros de adaptación, se ha puesto de manifiesto que dichos parámetros influyen demasiado en la señal de control. Por lo tanto, los valores que se establezcan en el *offset* y la ganancia del normalizador serán determinantes en el control de los objetos de las tareas. En algunos casos el movimiento de dichos objetos no era consecuencia de la intención del usuario, sino de los valores establecidos en el normalizador.

El normalizador tiene como objetivo determinar los valores óptimos de *offset* y ganancia mediante

una etapa de adaptación personalizada para cada usuario. Esto es necesario debido a que, normalmente, las personas tienen la actividad cerebral de uno de los hemisferios más desarrollada que otra. Si no se tuviera esto en cuenta, siempre se determinaría que, por su mayor actividad, el hemisferio dominante refleja la intención del usuario. Es decir, siempre se decantaría por el mismo lado.

Específicamente, del clasificador se obtienen dos señales de control: una para el movimiento hacia la derecha y otra para el movimiento hacia la izquierda. En esas señales de control se encuentra la intención del usuario. A continuación, esas señales pasan por el normalizador, en el cuál hay un valor de *offset* y de ganancia para cada señal de control. Primero se le resta el *offset* para después multiplicarle la ganancia. De las señales de salida del normalizador, se escoge la mayor de las dos como la decisión del usuario. Con esto queda claro que influyen directamente en el valor de la señal ya clasificada, cuando lo más correcto sería adaptar el propio clasificador, para que pueda discernir mejor la intención del usuario. Si la etapa de adaptación se realiza después, como ahora, y los valores que intervienen se adaptan de manera incorrecta, el manejo de las tareas con feedback positivo y negativo es inviable.

6.2 ANÁLISIS OFFLINE

El análisis de los datos se ha centrado en el rango de frecuencias de 0-25 Hz, ya que contiene las bandas donde se producen los SMR. Principalmente, se ha intentado averiguar si, a causa del entrenamiento con *neurofeedback*, se ha producido un aumento de la potencia en las bandas de los SMR. Como ya se ha especificado, dichas bandas están centradas en 12, 18 y 21 Hz, y tienen un ancho de banda de 3 Hz.

Para determinar si realmente se ha producido un aumento, se han realizado distintos análisis de los datos: estadístico y descriptivo. A raíz de los mismos, se han encontrado diferencias significativas en tres de los sujetos, que indican un aumento de la potencia. Este aumento no siempre se produce en las mismas bandas de frecuencia. Esto es debido a que cada persona tiene unas bandas óptimas donde localizar los SMR. Lo ideal sería calcular esas bandas antes del entrenamiento *online* y utilizarlas en la extracción y clasificación de características para obtener mejores resultados.

La magnitud del *p*-valor ha sido el indicador para establecer diferencias significativas. Dicho parámetro se ha calculado mediante el análisis estadístico de los rangos con signo de Wilcoxon. Se ha escogido este método debido a la naturaleza de los datos: univariados, dependientes, continuos y no normales.

El usuario U03 ha sido el único que ha presentado diferencias significativas causadas por la disminución de la potencia, en vez de por el aumento. El sujeto ha indicado que, en otro estudio de características similares, el supervisor le advirtió de su mala capacidad para modificar sus SMR. Esto podría ser la causa de que no se haya producido un aumento de la potencia. En cambio, sí se ha producido una disminución

que ha podido ser causada por un aprendizaje erróneo mediante el *neurofeedback*. Es decir, es posible que el sujeto haya aprendido a disminuir sus SMR en vez de aumentarlos para intentar manejar la aplicación. Esto provocaría el efecto contrario, causando un comportamiento indeseado en los *trials* de las sesiones de entrenamiento. La patente disminución de la potencia que se ha producido en la señal del U03, queda reflejada en la Figura 5.8.

Los resultados de los otros 3 sujetos, U01, U02 y U03, apoyan el buen funcionamiento de la aplicación, en lo que al control y aumento de los SMR se refiere. Los mejores resultados han sido los del U03, al experimentar diferencias significativas en dos de las tres bandas de frecuencia. El aumento de la potencia en las bandas de los SMR ya se ha demostrado anteriormente en la literatura científica, reafirmando la utilidad de la aplicación en el entrenamiento cognitivo. En el estudio Gómez-Pilar *et al* [37] se relaciona el aumento de la potencia conseguido mediante NFT con la mejora de ciertas funciones cognitivas.

Además de la media de la PSD de todos los canales, también se ha representado la PSD antes y después del entrenamiento de cada canal por separado. Con esta representación se puede determinar si realmente es en los electrodos más cercanos a la región motora donde se experimenta el aumento de la potencia. En la Figura 5.4, la Figura 5.7, la Figura 5.10 y la Figura 5.13 se puede observar cómo los sujetos U01, U02 y U04 experimentan el aumento en los canales de la región motora y sus adyacentes. El caso más claro es el del sujeto U02, ya que los aumentos más pronunciados se encuentran en los canales C3, C4 y Cz. No obstante, en algunos casos, se percibe también un gran aumento en las regiones temporales, que se encuentran más alejadas de las de interés. Cabe destacar que el U03, a pesar de experimentar una disminución de la potencia, dicha disminución se encuentra mayormente localizada en los canales C3, C4, F3 y F4, que se corresponden con la región del cortex motor y sus adyacentes. Estos resultados apoyan la capacidad del NFT para conseguir un aumento de la potencia en las regiones del cortex motor que, precisamente, era uno de los principales objetivos de la aplicación. Además, podría estar relacionado con un aumento en las capacidades cognitivas de los usuarios. No obstante, no se puede asegurar ya que no se han evaluado tales capacidades mediante tests neuropsicológicos.

6.3 CUESTIONARIO DE SATISFACCIÓN

A continuación, se discuten los resultados de los cuestionarios de satisfacción realizados a los sujetos, además de las sugerencias aportadas para mejorar la aplicación.

Los cuatro sujetos estuvieron de acuerdo en que la interfaz del sistema es intuitiva y fácil de comprender. Además, indicaron que les había resultado interesante conocer y utilizar esta aplicación, así como que el *feedback* de las distintas tareas les gustó y motivó para conseguir los objetivos. Esto indica que tanto la interfaz como el *feedback* han cumplido con los objetivos propuestos, además de motivar al

usuario para obtener resultados. Ante la afirmación de si las sesiones les habían resultado aburridas, el sujeto U02 se posicionó de manera neutral, mientras que los otros 3 indicaron que estaban en desacuerdo. Por ello, podemos considerar que la variación de las tareas disminuye el desinterés y la pasividad de los sujetos, y el *feedback* les motiva positivamente. Los sujetos U02 el U04 se posicionaron de manera neutral en cuanto a si las sesiones requieren demasiada concentración y terminan provocando el cansancio de los sujetos. En cambio, el U01 y U03 estaban en desacuerdo ante dicha afirmación. Todos los usuarios menos el U02 opinan que la duración de las sesiones de entrenamiento no era excesiva. El U02 ha contestado de manera neutral. En la cuestión de si han conseguido mejorar en el manejo de la aplicación después de las sesiones de entrenamiento, hay disparidad de opiniones. El U01 contestó de manera neutral, el U02 ha indicado que estaba de acuerdo con la afirmación, el U03 en cambio estaba en desacuerdo y el U04 totalmente de acuerdo. Con respecto a la afirmación de que la aplicación tiene poca utilidad y no parece interesante, en la cuál todos estaban en desacuerdo. En la última de las afirmaciones, los sujetos U02 y U04 han indicado que la adquisición de la actividad cerebral mediante el EEG con el gorro y electrodos activos les parece práctica, mientras que los usuarios U01 y U03 se han posicionado de manera neutral. Todos los sujetos respondieron positivamente a la pregunta de si volverían a participar en un proyecto de características similares.

Atendiendo a las sugerencias de mejora, el U01 indicó la mejora que supondría implementar todo el sistema en la *tablet*. Para ello, sería necesario poder conectar el amplificador directamente a la *tablet* o disponer de un gorro de electrodos inalámbrico. Si el gorro inalámbrico y su amplificador consiguen una señal de EEG adecuada, mejoraría la simplicidad de la arquitectura de la aplicación, así como su portabilidad y comodidad. El U03 ha sugerido que debería mejorarse la etapa de adaptación, debido a que él obtuvo un control muy limitado de la aplicación. Seguramente, no logró un mejor control por la mala adaptación de los parámetros *offset* y ganancia del normalizador. También puede deberse a que dicho usuario no sea capaz de modificar sus SMR, ya que presentó un control inferior al resto de los sujetos. Para mejorar la aplicación y la normalización debería realizarse en el clasificador y no a su salida, modificando el código del clasificador lineal encargado de determinar la intención del usuario. Por último, el U04 afirmó que cuando se encontraba realizando un *trial* impar era capaz de predecir el objetivo del siguiente *trial*. Según su opinión, debido al carácter estacionario de los objetivos. Sugería, por ello, hacer que los objetivos se generaran de manera aleatoria. Se debe hacer constar que la elección de los objetivos de los *trials* ya es aleatoria, pero al haber solo dos objetivos posibles es normal que haya patrones que se repitan. Ante este aspecto, no es necesario modificar nada de la aplicación.

6.4 COMPARACIÓN CON OTROS ESTUDIOS

Atendiendo a las funcionalidades de la aplicación, la mayor similitud se encuentra con respecto al estudio de Gomez-Pilar *et al* [37], puesto que, aunque en muchos otros estudios también se hayan utilizado

los SMR como señales de control (Hwang *et al* [43] y Shindo *et al* [44]), es el único que tiene como objetivo el desarrollo cognitivo de los sujetos, restándole importancia a la precisión al completar las tareas propuestas. Sin embargo, también posee diferencias con respecto a este estudio. Entre ellas, destaca el hecho de que en esta aplicación se calculan dos señales de control en cada instante de tiempo, una nos indica la salida del clasificador para el hemisferio derecho y la otra para el izquierdo. Por el contrario, en el estudio Gomez-Pilar *et al* [37] solo se obtiene una señal de control, que se calcula con la salida del clasificador, cuya entrada es el Laplaciano centrado en el electrodo C4. Por ello, en el estudio mencionado se decide que el usuario ha elegido izquierda cuando el sujeto realice la MI de la mano izquierda, provocando el aumento de la señal de control. Por el contrario, se decide derecha tanto si el usuario realiza la MI de la mano derecha como si no realiza ninguna MI, provocando la disminución de la señal de control. En este Trabajo Fin de Grado, al tener dos señales se proporciona información más precisa de la intención del usuario, ya que debería ser posible distinguir entre MI de la mano derecha, de la izquierda y ninguna MI. También se debe mencionar que, al tener dos señales de control, los parámetros de adaptación tienen demasiada influencia, por lo que no se ha conseguido manejar correctamente las tres últimas tareas. Además, en el estudio de Gomez-Pilar *et al* [37], antes de la primera sesión y después de la última, se aplicó un test neuropsicológico a todos los usuarios, para poder determinar si el entrenamiento había producido mejoras en las capacidades cognitivas. Desafortunadamente, este análisis no ha podido llevarse a cabo en el presente estudio.

Algunos de los estudios no solo utilizan los SMR para realizar el entrenamiento, sino que también aplican los SCP para intentar mejorar los efectos (Gevensleben *et al* [41]). Otros estudios, en vez de centrarse en el desarrollo cognitivo, buscan la mejora en algunas funciones motoras, o el control de dispositivos que las mejoren. En el estudio de Hwang *et al* [43], que utiliza únicamente SMR, las precisiones promedio obtenidas han sido superiores a las de este trabajo, alcanzando una media de 71,40 %.

6.5 LIMITACIONES

Dentro de las limitaciones de la aplicación destaca el hecho de que la etapa de adaptación no se realiza de manera adecuada. Se debería adaptar el clasificador para que tenga en cuenta la diferencia entre los hemisferios. De esta forma, se mejorarían las precisiones de las tareas, lo que podría favorecer el entrenamiento.

Adicionalmente, al procesar los datos en el ordenador (mediante BCI2000) para luego enviarlos al dispositivo Android (a través de una conexión Bluetooth), se provoca un desfase debido al retardo producido por el tiempo que tarda en llenar el *buffer* y enviarlo por la conexión. Dicho proceso ralentiza la ejecución de BCI2000. En una futura versión sería interesante, de algún modo, enviar la señal del casco directamente al dispositivo Android, para realizar todo el procesado y el *feedback* en dicho dispositivo.

Así sería innecesaria la sincronización y se mejoraría el rendimiento de manera notable.

Asimismo, sería conveniente evaluar la aplicación con un mayor grupo de usuarios, para valorar su funcionamiento y asegurar la generalización de los resultados. Además, resultaría interesante realizar algún tipo de test o análisis neuropsicológico para estudiar si se ha producido un desarrollo cognitivo.

CONCLUSIONES Y LÍNEAS FUTURAS

7.1 CONCLUSIONES

A lo largo de este trabajo se ha llevado a cabo un estudio sobre los sistemas BCI, analizando los principales métodos para registrar la actividad cerebral y los tipos de señales de control que gobiernan dichos sistemas. Se ha considerado el método de EEG como la mejor solución para desarrollar la aplicación de NFT, debido a que se trata de un método no invasivo, portátil y de bajo coste. También se han estudiado las etapas principales del procesado de la señal EEG, empezando por la etapa de adquisición de la señal, seguida por la extracción de características y terminando por clasificación de características. Adicionalmente, se ha realizado una búsqueda exhaustiva del estado del arte sobre los estudios con NFT realizados anteriormente y se han comparado entre ellos para extraer las ventajas y los inconvenientes de cada uno. En base a esa comparativa se han propuesto los objetivos de la aplicación y se han elegido los ritmos sensoriomotores como la señal de control más apropiada.

Posteriormente, se han detallado en profundidad las características de los ritmos sensoriomotores y los factores que influyen en ellos, así como las diferencias individuales entre los sujetos. Una vez estudiada la señal de control a utilizar, se han detallado los diversos métodos de extracción de características para los ritmos sensoriomotores considerando un análisis espacial, temporal y espectral. Dentro del análisis espacial se han descrito: el filtrado Laplaciano, los métodos de referencia de media común, los patrones espaciales comunes, el análisis de componentes independientes y el análisis de componentes principales. Con respecto a las técnicas de análisis temporal se han tratado: el promediado sincronizado y el filtrado adaptativo. En último lugar, dentro del análisis espectral se han introducido los conceptos de: transformada de Fourier de tiempo corto, transformadas wavelet, filtros FIR e IIR, estimación espectral autorregresiva y potencia relativa. Después de detallar la extracción de características, se han recogido los métodos principales de traducción de las mismas, entre ellos el discriminante lineal de Fisher, las máquinas de vectores de soporte y los clasificadores no lineales. Se ha puesto especial énfasis en el análisis discriminante lineal, que es el método utilizado en la aplicación desarrollada.

En cuanto al diseño de la aplicación de entrenamiento con *neurofeedback*, basado en la señal EEG y los ritmos sensoriomotores, el primer paso ha consistido en recopilar las necesidades de la aplicación y formular los objetivos de diseño. Una vez establecidos, se ha estructurado la misma en tres partes principales: etapa de adquisición de la actividad cerebral, etapa de procesado del EEG y *BCI-NeuroAge*. La primera se ha llevado a cabo con un equipo de *G.Tec*. El procesado de la señal se ha desarrollado con la ayuda del sistema de propósito general BCI2000, cuya finalidad es la de facilitar el desarrollo, la investigación y la monitorización de aplicaciones BCI en tiempo real.

La etapa de procesado del EEG se ha implementado en lenguaje C++ y se encarga de adquirir y procesar la señal EEG a través de BCI2000. Para ello, se ha utilizado un filtrado espacial Laplaciano, seguido de un clasificador lineal y, por último, un normalizador. Una vez iniciado el programa, este proporciona las señales de control necesarias para controlar la aplicación *BCI-NeuroAge*. Dichas señales, junto con las variables necesarias para la sincronización, se envían a través de Bluetooth al dispositivo final.

El bloque *BCI-NeuroAge* se ha implementado en Java y C#, para dispositivos Android. La finalidad de esta etapa es proporcionar al usuario la realimentación necesaria para el entrenamiento cognitivo. El *feedback* es consecuencia directa de los valores de las señales de control que recibe de BCI2000. *BCI-NeuroAge* consta de 5 tareas diferentes, dos de ellas con *feedback* únicamente positivo, y las otras tres con *feedback* tanto positivo como negativo. Cada tarea posee una interfaz gráfica diferente con un *feedback* característico para intentar estimular al usuario en el entrenamiento cognitivo.

Una vez desarrollada la aplicación, ésta ha sido evaluada por parte de 4 sujetos sanos durante tres sesiones de entrenamiento. La disposición de los electrodos ha seguido la norma internacional 10/20, ubicando los mismos en las localizaciones F3, F4, T7, C3, Cz, C4, T8 y Pz. Antes de comenzar las tareas de la primera sesión, se ha realizado a los sujetos un registro EEG basal de 2 minutos. A continuación, se han llevado a cabo las tareas propuestas para la sesión, con una duración aproximada de 90 minutos. En las primeras tareas de cada sesión, se realiza la etapa de adaptación. En la segunda y tercera sesión, se llevan a cabo las tareas propuestas, adaptando manualmente los parámetros de la aplicación cuando sea necesario. Al finalizar la última sesión se ha realizado un registro basal de 2 minutos a cada sujeto, además de un cuestionario de satisfacción en el que han reflejado sus experiencias y sugerencias de mejora.

Para terminar, se han expuesto los resultados obtenidos y se han discutido. Estos resultados reflejan el buen funcionamiento del sistema en algunos aspectos, y la necesidad de mejora de otros. En cuanto a la interfaz gráfica de la aplicación, a todos los usuarios les ha parecido intuitiva y el *feedback* diseñado les ha gustado y motivado. Además, se han apreciado cambios en las señales EEG de los usuarios a raíz del entrenamiento cognitivo. Esto indica que el *neurofeedback* funciona correctamente. No obstante, aunque

se puede esperar que el sistema sea más preciso que en otros estudios al emplear dos señales de control, una mala adaptación provoca que los usuarios no obtengan precisiones mayores en las tres últimas tareas.

Se concluye que el rendimiento del sistema podría mejorar considerablemente si la etapa de adaptación se realizara en el clasificador y no a la salida de éste. Además, si el procesado se llevara a cabo en el mismo dispositivo Android, donde se proporciona el *feedback*, el sistema sería más eficiente. Esto produciría mucho menos retardo y los *buffer* y la sincronización tampoco serían necesarios. A pesar de las posibles mejoras mencionadas, el entrenamiento realizado ha modulado la actividad cerebral de todos los sujetos. En tres de ellos se ha producido un aumento de la potencia relativa en las bandas de los SMR. Sin embargo, en el sujeto restante se ha producido una disminución de dicha potencia.

Respecto a los objetivos del Trabajo Fin de Grado desarrollados en el capítulo uno, se han conseguido alcanzar todos, como vemos en el siguiente resumen:

- Se han estudiado las bases de los sistemas BCI.
- Se ha realizado una revisión del estado del arte de otras aplicaciones de NFT.
- Se han estudiado los lenguajes de programación con los que se ha realizado la aplicación: Java, C++ y C#.
- Se ha desarrollado la aplicación propuesta.
- Se ha evaluado la aplicación con 4 sujetos sanos.
- Se han analizado los resultados mediante un *script* en MATLAB.
- Se han discutido y comparado los resultados obtenidos.
- Se han extraído conclusiones del trabajo realizado.

A modo de resumen, se incluyen a continuación las conclusiones extraídas del cumplimiento de los anteriores objetivos:

1. La aplicación desarrollada adquiere la señal de control mediante EEG por ser un método no invasivo, portátil y de bajo coste.
2. La señal de control utilizada son los SMR, puesto que son endógenos y requieren entrenamiento.
3. La extracción de características se realiza mediante un filtro espacial Laplaciano, tomando como electrodos principales C3 y C4 por ser los más cercanos a la región motora del cortex. A continuación, se aplica un filtro AR que proporciona una estimación de la amplitud del espectro.
4. La clasificación se lleva a cabo mediante dos LDA, uno para el hemisferio derecho y otro para el izquierdo. Los LDA toman la RP de tres bandas de frecuencia centradas en 12, 18 y 21 Hz, por ser las más utilizadas en SMR.

5. La arquitectura de la aplicación está distribuida entre un ordenador, que realiza el procesado de señal, y un dispositivo móvil, que proporciona en *neurofeedback*. La elección de esta estructura se ha debido a la imposibilidad de conectar el equipo disponible de *G.Tec* directamente al dispositivo Android.
6. El objetivo de la aplicación *BCI-NeuroAge* es el desarrollo cognitivo mediante el NFT. El *feedback* se ha diseñado de la manera más atractiva posible para incentivar la motivación del usuario.
7. La evaluación de la aplicación se ha realizado con 4 sujetos sanos mediante tres sesiones de entrenamiento. Al inicio de las mismas, es necesaria una etapa de adaptación del normalizador. Adicionalmente, se hizo un registro basal del EEG antes y después del entrenamiento para un estudio *offline*. Al terminar se les entregó un cuestionario de satisfacción, con el fin de recoger las experiencias y sugerencias de cada uno de los usuarios.
8. Los resultados de la evaluación han mostrado diferencias significativas en la potencia relativa del espectro del EEG en las bandas correspondientes a los SMR de todos los sujetos. Las precisiones de las tareas 1 y 2 han sido satisfactorias, estando entorno al 82,75 %. Sin embargo, las precisiones obtenidas en las tareas con *feedback* positivo y negativo han inferiores a las deseadas, alcanzando un promedio de 42,82 %.
9. La comparación con otros estudios sugiere que el aumento de la potencia relativa en las bandas de los SMR está relacionado con una mejora de ciertas capacidades cognitivas. Sin embargo, no se ha podido realizar ningún test neuropsicológico para demostrarlo.

Después de la evaluación de la aplicación se han descubierto ciertas limitaciones, entre las que destacan: i) el mal comportamiento de la adaptación en las tareas con *feedback* positivo y negativo; ii) la necesidad de sincronización entre BCI2000 y la aplicación debido al desfase producido por el retardo del envío de la señal de control; y iii) la conveniencia de evaluar la aplicación con un mayor grupo de usuarios, para valorar su funcionamiento y asegurar la generalización de los resultados.

7.2 LÍNEAS FUTURAS

Las líneas futuras que debe seguir este Trabajo Fin de Grado se extraen de las limitaciones que se han encontrado a la hora de evaluarlo. La más importante es intentar mejorar la etapa de adaptación actual, ya que se ha visto un funcionamiento pobre en las tareas con *feedback* positivo y negativo durante los entrenamientos. Una solución podría ser integrar la adaptación al clasificador, convirtiendo las salidas del mismo en las señales de control de la aplicación, eliminando la necesidad del normalizador.

En cuanto a las tareas, se puede añadir alguna en la que intervenga la memoria del sujeto, si se deseara desarrollarla con el entrenamiento.

Desde el punto de vista de la comodidad y portabilidad, se podría implementar un sustituto a BCI2000 que se encargase del procesado de la señal para dispositivos Android. Así, todo el sistema podría unificarse en un único dispositivo portátil, facilitando el uso de sistemas de adquisición de señal inalámbricos, que son más cómodos para los usuarios.

En cuanto a la evaluación de la aplicación, debería realizarse con un número de sujetos más elevado y pertenecientes a la población objetivo de la aplicación, que son las personas mayores. Además, sería conveniente realizar algún tipo de test neuropsicológico para poder estimar si se ha producido un desarrollo cognitivo, y en qué capacidades ha tenido lugar.

BIBLIOGRAFÍA

- [1] A. J. Thurston, “Paré and prosthetics: the early history of artificial limbs.” *ANZ Journal of Surgery*, vol. 77, no. 12, pp. 1114–1119, 2007.
- [2] J. D. Bronzino, *The Biomedical Engineering Handbook. Medical Devices and Systems*, 3rd ed., R. C. Dorf, Ed. Connecticut, U.S.A: CRC press, Taylor & Francis Group, 2006.
- [3] L. Sörmmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*, E. A. Press, Ed. Elseiver, 2005.
- [4] J. R. Evans and A. Abarbanel, *Introduction to Quantitative EEG and Neurofeedback*, 1st ed. Georgia, USA: Academic Press, May 1999.
- [5] N. Boutros and F. Struve, “Kaplan and sadock’s comprehensive textbook of psychiatry,” http://www.parsehclinic.com/files/filebox/kaplan/kaplan_ctp_8th/1.14.htm, accessed: 2018-05-14.
- [6] S. Thong and N. V. Thakor, *Quantitative EEG Analysis Methods and Clinical Applications*. Massachusetts, USA: Artech House, 2009.
- [7] J. J. Guzmán, “La actividad cerebral,” <http://neurofisiologiagranada.com/eeg/eeg-actividadcerebral.htm>, accessed: 2018-05-14.
- [8] B. Graimann, B. Allison, and G. Pfurtscheller, *Brain-Computer Interface, Revolutionizing Human-Computer Interaction*. Springer, 2010.
- [9] Z. P. Liang and P. C. Lauterbur, *Toward Brain-Computer Interfacing*. Massachusetts, USA: IEEE Press, 2000.
- [10] S. N. Abdulkader, A. Atia, and M.-S. M. Mostafa, “Brain computer interfacing: Applications and challenges,” *Egyptian Informatics Journal*, vol. 16, no. 2, pp. 213–230, July 2015.
- [11] B. Wei, X. Wu, and D. Zhang, “Design and implementation of an improved bci system based on alpha rhythm,” *3rd International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–4, 2009.
- [12] R. P. N. Rao, *Brain-Computer Interfacing, An Introduction*, 1st ed. Seattle: Cambridge University Press, 2013.

- [13] B. He, S. Gao, H. Yuan, and J. R. Wolpaw, *Neural Engineering*, 2nd ed. Springer, 2013.
- [14] A. Guillot and C. Collet, *The Neurophysiological Foundations of Mental and Motor Imagery*. New York: Oxford University Press, 2010.
- [15] L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain computer interfaces, a review,” *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.
- [16] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Brain–computer interfaces for communication and control,” *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.
- [17] O. Friman, I. Volosyak, and A. Gräser, “Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces,” *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 4, pp. 742–750, 2007.
- [18] M. A. Pastor, J. Artieda, J. Arbizu, M. Valencia, and J. C. Masdeu, “Human cerebral activation during steady-state visual-evoked responses,” *The Journal of Neuroscience*, vol. 23, no. 37, p. 11621–11627, 2003.
- [19] A. Quintero-Rincón, M. Risk, and S. Liberczuk, “Preprocesamiento de eeg con filtros hampel,” *IEEE Latin America Transactions*, vol. 2012, no. 89, pp. 1–6, 2012.
- [20] “Image gallery: eeg artefact,” <http://keywordsuggest.org/gallery/393195.html>, accessed: 2018-05-14.
- [21] “Clasificación lineal contra no lineal con python,” <http://eenube.com/index.php/mas-articulos/28-aprendizaje-maquina/124-casificadoresnolineales>, accessed: 2018-05-14.
- [22] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kubler, J. Perelmouter, E. Taub, and H. Flor, “A spelling device for the paralysed,” *Nature*, vol. 398, no. 6725, p. 297–298, 1999.
- [23] B. Obermaier, G. R. Muller, and G. Pfurtscheller, ““virtual keyboard” controlled by spontaneous eeg activity,” *IEEE Transactions on Neural System and Rehabilitation Engineering*, vol. 11, no. 4, p. 422–426, 2003.
- [24] L. A. Farwell and E. Donchin, “Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials,” *Electroencephalography and clinical Neurophysiology*, vol. 70, p. 510–523, 1988.
- [25] Y. R. Tabar and U. Halici, “Brain computer interfaces for silent speech,” *European Review*, vol. 25, no. 2, pp. 208–230, 2017.
- [26] M. Bensch, A. A. Karim, J. Mellinger, T. Hinterberger, M. Tangermann, and M. Bogdan, “Nessi: An eeg-controlled web browser for severely paralyzed,” *omputational intelligence and neuroscience*, vol. 2007, 2007.

- [27] G. P. Braz, M. Russold, and G. M. Davis, “Functional electrical stimulation control of standing and stepping after spinal cord injury: A review of technical characteristics,” *Neuromodulation: technology at the neural interface*, vol. 12, no. 3, p. 180–190, 2009.
- [28] G. Pfurtscheller, G. R. Müller, J. Pfurtscheller, H. J. Gerner, and R. Rupp, “‘thought’-control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia,” *Neuroscience Letters*, vol. 351, p. 33–36, 2003.
- [29] G. R. Müller-Putz and G. Pfurtscheller, “Control of an electrical prosthesis with an ssvp-based bci,” *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 1, p. 361–364, 2008.
- [30] “Robots de rehabilitación reducen los temblores del párkinson,” <http://www.agenciasinc.es/Noticias/Robots-de-rehabilitacion-reducen-los-temblores-del-parkinson>, accessed: 2018-05-14.
- [31] R. Corralejo, R. Hornero, and D. Álvarez, “A p300-based bci aimed to manage electronic devices for people with severe disabilities,” *Medical and Biological Engineering and Computing*, vol. 52, no. 10, pp. 861–872, 2014.
- [32] K. Tanaka, K. Matsunaga, and H. O. Wang, “Electroencephalogram-based control of an electric wheelchair,” *IEEE Transactions on Robotics*, vol. 21, no. 4, p. 762–766, 2005.
- [33] B. Rebsamen, E. Burdet, G. Cuntai, T. Chee Leong, Z. Qiang, M. Ang, and C. Laugier, “Controlling a wheelchair using a bci with low information transfer rate,” in *IEEE 10th International Conference on Rehabilitation Robotics*, ser. ICORR’07, Noordwijk, The Netherlands, June 2007, pp. 1003–1008.
- [34] B. Rebsamen, G. Cuntai, Z. Haihong, W. Chuanchu, T. Cheeleong, M. H. Ang, and E. Burdet, “A brain controlled wheelchair to navigate in familiar environments,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, p. 590–598, 2010.
- [35] A. Orbe, “Universidad de zaragoza. interfaz cerebro máquina,” <http://sinapsis-aom.blogspot.com.es/2010/05/universidad-de-zaragoza-interfaz.html>, accessed: 2018-05-14.
- [36] K. Roman, B. Benjamin, C. Gabriel, and M. Klaus-Robert, “The berlin brain-computer interface (bbci)—towards a new communication channel for online control in gaming applications,” *Multimed Tools Appl*, vol. 33, p. 73–90, 2007.
- [37] J. GomezPilar, R. Corralejo, L. F. Nicolas-Alonso, D. Álvarez, and R. Hornero, “Neurofeedback training with a motor imagery-based bci: neurocognitive improvements and eeg changes in the elderly,” *Medical and Biological Engineering and Computing*, vol. 54, no. 11, pp. 1655–1666, 2016.
- [38] G. Rota, R. Sitaram, R. Veit, M. Erb, N. Weiskopf, G. Dogil, and N. Birbaumer, “Self-regulation of regional cortical activity using real-time fmri: The right inferior frontal gyrus and linguistic processing,” *Human Brain Mapping*, vol. 30, no. 5, p. 1605–1614, 2009.

- [39] R. Sitaram, S. Lee, S. Ruiz, M. Rana, R. Veit, and N. Birbaumer, “Real-time support vector classification and feedback of multiple emotional brain states,” *Neuroimage*, vol. 56, pp. 753–765, 2011.
- [40] M. B. Sterman and T. Egner, “Foundation and practice of neurofeedback for the treatment of epilepsy,” *Applied Psychophysiology and Biofeedback*, vol. 31, no. 1, pp. 21–35, 2006.
- [41] H. Gevensleben, B. Holl, B. Albrecht, C. Vogel, D. Schlamp, O. Kratz, P. Studer, G. H. Rothenberger, A. Moll, and H. Heinrich, “Is neurofeedback an efficacious treatment for adhd? a randomised controlled clinical trial,” *Child Psychology and Psychiatry*, vol. 50, no. 7, 2009.
- [42] D. Ariely and G. S. Berns, “Neuromarketing: The hope and hype of neuroimaging in business,” *Nature Reviews Neuroscience*, vol. 11, p. 284–292, 2010.
- [43] H. Hwang, K. Kwon, and C. Im, “Neurofeedback-based motor imagery training for brain–computer interface (bci),” *Journal of Neuroscience Methods*, vol. 179, no. 1, pp. 150–156, 2009.
- [44] K. Shindo, K. Kawashima, J. Ushiba, and M. Liu, “Effects of neurofeedback training with an electroencephalogram-based brain-computer interface for hand paralysis in patients with chronic stroke: a preliminary case series study,” *Journal of rehabilitation medicine*, vol. 43, no. 10, pp. 951–957, 2011.
- [45] H. Gevensleben, B. Holl, B. Albrecht, C. Vogel, D. Schlamp, O. Kratz, P. Studer, G. H. Rothenberger, A. Moll, and H. Heinrich, “Distinct eeg effects related to neurofeedback training in children with adhd: a randomized controlled trial.” *Child Psychology and Psychiatry*, vol. 50, no. 7, 2009.
- [46] N. Weiskopf, K. Mathiak, S. W. Bock, F. Scharnowski, R. Veit, W. Grodd, R. Goebel, and N. Birbaumer, “Principles of a brain-computer interface (bci) based on real-time functional magnetic resonance imaging (fmri),” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 966–970, 2004.
- [47] S. E. Kober, G. Wood, J. Kurzmann, E. V. C. Friedrich, M. Stangl, T. Wippel, A. Väljamäe, and C. Neuper, “Near-infrared spectroscopy based neurofeedback training increases specific motor imagery related cortical activation compared to sham feedback,” *Biological Psychology*, vol. 95, pp. 21–30, 2014.
- [48] G. Schalk and J. Mellinger, *A Practical Guide to Brain–Computer Interfacing with BCI2000*. London: Springer, 2010.
- [49] “Brain and cranial nerves,” <https://www.slideshare.net/meleebirdsong/activity-7-62436792>, accessed: 2018-05-14.
- [50] E. M. Hammer, S. Hakder, B. Blankertz, C. Sannelli, T. Dickhaus, S. Kleih, K.-R. Müller, and A. Kübler, “Psychological predictors of smr-bci performance,” *Biological Psychology*, vol. 89, no. 1, pp. 80–86, 2012.

- [51] C. Neuper, R. Scherer, M. Reiner, and G. Pfurtscheller, "Imagery of motor actions: Differential effects of kinesthetic and visual-motor mode of imagery in single-trial eeg," *Cognitive Brain Research*, vol. 25, pp. 668–677, 2005.
- [52] M. Z. Ilyas, P. Saad, and M. I. Ahmad, "A survey of analysis and classification of eeg signals for brain-computer interfaces," *2nd International Conference on Biomedical Engineering*, 2015.
- [53] P. Herman, G. Prasad, T. M. McGinnity, and D. Coyle, "Comparative analysis of spectral approaches to feature extraction for eeg-based motor imagery classification," *IEEE Transactions on Neural System and Rehabilitation Engineering*, vol. 16, no. 4, pp. 317–326, 2008.
- [54] M. R. Lakshmi, T. V. Prasad, and V. C. Prakash, "Survey on eeg signal processing methods," *Internaional Journal of Advanced Reserch in Computer Science and Software Engineering*, vol. 4, no. 1, pp. 84–91, 2014.
- [55] J. Qin, Y. Li, and A. Cichocki, "Ica and committee machine-based algorithm for cursor control in a bci system," in *International Symposium on Neural Networks*, Berlin Heidelberg, June 2005, p. 973–978.
- [56] X. Yu, P. Chum, and K.-B. Sim, "Analysis the effect of pca for feature reduction in non-stationary eeg based motor imagery of bci system," *Optik International Journal for Light and Electron Optics*, vol. 125, no. 3, pp. 1498–1502, 2014.
- [57] G. Li, "How do i interpret the results of a pca analysis?" <https://www.quora.com/How-do-I-interpret-the-results-of-a-PCA-analysis>, accessed: 2018-05-16.
- [58] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller, "Optimizing spatial filters for robust eeg single-trial analysis," *IEEE Signal Processing*, pp. 41–56, 2008.
- [59] R. M. Rangayyan, "Filtering for removal artifacts," in *Biomedical Signal Analysis: A Case-Study Approach*. Canada: IEEE Press & Wiley, 2002., p. 73–176.
- [60] D. J. Krusienski, D. J. McFarland, and J. R. Wolpaw, "An evaluation of autoregressive spectral estimation model order for brain-computer interface application," *Proceedings of the 28th IEEE*, 2006.
- [61] S. R. J., *Artificial Neural Networks*. McGraw-Hill, June 1997.

Apéndice A

CÓDIGO GENERADO EN C++ , JAVA, C# Y MATLAB

A continuación se recoge el código correspondiente a la aplicación desarrollada durante el presente Trabajo Fin de Grado.

A.1 MODIFICACIÓN DEL BLOQUE CURSORTASK

En este apartado se muestra el código añadido al bloque CursorTask de BCI2000 para modificar su comportamiento. Se ha añadido un archivo nuevo, BluetoothClient.cpp, y modificado el archivo de cabecera CursorFeedbackTask.h y el archivo fuente CursorFeedbackTask.cpp, ya existentes.

A.1.1 BLUETOOTHCLIENT

```
/**
 * Bluetooth client for BCI-NeuroAge
 * @version: 1.0
 */
using namespace std;
// Libraries and headers
#pragma comment(lib, "ws2_32.lib")
#include "P3SpellerTask.h"
#include <winsock2.h>
#include <ws2bth.h>
#include <strsafe.h>
#include <string>
#include <iostream>

BluetoothClient::BluetoothClient(){
}
ULONG BluetoothClient::runClient (_In_ SOCKADDR_BTH RemoteAddr, _In_ char*
    CXN_TEST_DATA_STRING, _In_ SIZE_T CXN_TRANSFER_DATA_LENGTH){
```

```

ULONG          ulRetCode = CXN_SUCCESS;
int           iCxnCount = 0;
char          *pszData = NULL;
SOCKET         LocalSocket = INVALID_SOCKET;
SOCKADDR_BTH  SockAddrBthServer = RemoteAddr;
HRESULT        res;

pszData = (char *) HeapAlloc(GetProcessHeap(), HEAP_ZERO_MEMORY, CXN_TRANSFER_DATA_LENGTH
);
if ( NULL == pszData ) {
ulRetCode = STATUS_NO_MEMORY;
bciout << "CRITICAL=|_HeapAlloc_failed_|_out_of_memory,_gle_|" << GetLastError();
}

if ( CXN_SUCCESS == ulRetCode ) {
SockAddrBthServer.addressFamily = AF_BTH;
SockAddrBthServer.serviceClassId = g_guidServiceClass;
SockAddrBthServer.port = 0;

res = StringCbCopyN(pszData, CXN_TRANSFER_DATA_LENGTH, CXN_TEST_DATA_STRING,
CXN_TRANSFER_DATA_LENGTH);
if ( FAILED(res) ) {
bciout << "CRITICAL=|_Creating_a_static_data_string_failed";
ulRetCode = CXN_ERROR;
}
}

if ( CXN_SUCCESS == ulRetCode ) {

pszData[(CXN_TRANSFER_DATA_LENGTH/sizeof(wchar_t)) - 1] = 0;

LocalSocket = socket(AF_BTH, SOCK_STREAM, BIHPROTO_RFCOMM);
if ( INVALID_SOCKET == LocalSocket ) {
bciout << "CRITICAL=|_socket()_call_failed._WSAGetLastError=|" << WSAGetLastError();
ulRetCode = CXN_ERROR;
}

if ( SOCKET_ERROR == connect(LocalSocket, (struct sockaddr *) &SockAddrBthServer, sizeof(
SOCKADDR_BTH)) ) {
bciout << "CRITICAL=|_connect()_call_failed._Target_Device_is_not_available._
WSAGetLastError=|" << WSAGetLastError();
ulRetCode = CXN_ERROR;
}

if ( SOCKET_ERROR == send(LocalSocket, pszData, (int)CXN_TRANSFER_DATA_LENGTH, 0) ) {

```

```

bciout << "CRITICAL= send() call failed w/socket=[0x%I64X], szData=[%p], dataLen=[%I64u]. WSAGetLastError=[%d]\n", (ULONG64)LocalSocket, pszData, (ULONG64)
    CXN_TRANSFER_DATA_LENGTH, WSAGetLastError();
ulRetCode = CXN_ERROR;
}

if ( SOCKET_ERROR == closesocket(LocalSocket) ) {
bciout << "CRITICAL= closesocket() call failed w/socket=[0x%I64X]. WSAGetLastError=[%d]\n", (ULONG64)LocalSocket, WSAGetLastError();
ulRetCode = CXN_ERROR;
}
LocalSocket = INVALID_SOCKET;
}

if ( INVALID_SOCKET != LocalSocket ) {
    closesocket(LocalSocket);
    LocalSocket = INVALID_SOCKET;
}

if ( NULL != pszData ) {
    HeapFree(GetProcessHeap(), 0, pszData);
    pszData = NULL;
}
return(ulRetCode);
}

ULONG BluetoothClient::nameToBthAddr(_In_ const char *pszRemoteName, _Out_ Psockaddr_bth
    pRemoteBtAddr)
{
    INT          iResult = CXN_SUCCESS;
    BOOL         bContinueLookup = FALSE, bRemoteDeviceFound = FALSE;
    ULONG        ulFlags = 0, ulPQSSize = sizeof(WSAQUERYSET);
    HANDLE       hLookup = NULL;
    PWSAQUERYSET pWSAQuerySet = NULL;

    ZeroMemory(pRemoteBtAddr, sizeof(*pRemoteBtAddr));

    pWSAQuerySet = (PWSAQUERYSET) HeapAlloc(GetProcessHeap(), HEAP_ZERO_MEMORY, ulPQSSize);
    if ( NULL == pWSAQuerySet ) {
        iResult = STATUS_NO_MEMORY;
        bciout << "ERROR! Unable to allocate memory for WSAQUERYSET\n";
    }

    if ( CXN_SUCCESS == iResult ) {

```

```

for ( INT iRetryCount = 0; !bRemoteDeviceFound && ( iRetryCount < CXN_MAX_INQUIRY_RETRY);
      iRetryCount++ ) {

ulFlags = LUP_CONTAINERS;
ulFlags |= LUP_RETURN_NAME;
ulFlags |= LUP_RETURN_ADDR;

if ( 0 == iRetryCount ) {

} else {
ulFlags |= LUP_FLUSHCACHE;

bciout << "*INFO*|_Unable to find device. Waiting for %d seconds before re-inquiry ... \n"
        ", CXN_DELAY_NEXT_INQUIRY;
Sleep(CXN_DELAY_NEXT_INQUIRY * 1000);

bciout << "*INFO*|_Inquiring device ... \n";
}

iResult = CXN_SUCCESS;
hLookup = 0;
bContinueLookup = FALSE;
ZeroMemory(pWSAQuerySet, ulPQSSize);
pWSAQuerySet->dwNameSpace = NS_BTH;
pWSAQuerySet->dwSize = sizeof(WSAQUERYSET);
iResult = WSALookupServiceBegin(pWSAQuerySet, ulFlags, &hLookup);

if ( (NO_ERROR == iResult) && (NULL != hLookup) ) {
bContinueLookup = TRUE;
} else if ( 0 < iRetryCount ) {
bciout << "=CRITICAL=_|_WSALookupServiceBegin() failed with error code %d, _
        WSAGetLastError=_|_%d\n", iResult, WSAGetLastError();
break;
}

while ( bContinueLookup ) {

// ZeroMemory(pWSAQuerySet, ulPQSSize);
// pWSAQuerySet->dwNameSpace = NS_BTH;
// pWSAQuerySet->dwSize = sizeof(WSAQUERYSET);
if ( NO_ERROR == WSALookupServiceNext(hLookup, ulFlags, &ulPQSSize, pWSAQuerySet) ) {
if ( ( pWSAQuerySet->lpszServiceInstanceName != NULL ) &&
( CXN_SUCCESS == strcmp(pWSAQuerySet->lpszServiceInstanceName, pszRemoteName) ) ) {

CopyMemory(pRemoteBtAddr, (PSOCKADDR_BTH) pWSAQuerySet->lpcsaBuffer->RemoteAddr.
        lpSockaddr, sizeof(*pRemoteBtAddr));

```

```
bRemoteDeviceFound = TRUE;
bContinueLookup = FALSE;
}
} else {
iResult = WSAGetLastError();
if ( WSA_E_NO_MORE == iResult ) { //No more data
bContinueLookup = FALSE;
} else if ( WSAEFAULT == iResult ) {

HeapFree(GetProcessHeap(), 0, pWSAQuerySet);
pWSAQuerySet = (PWSAQUERYSET) HeapAlloc(GetProcessHeap(), HEAP_ZERO_MEMORY, ulPQSSize);
if ( NULL == pWSAQuerySet ) {
bciout << " !ERROR! | | Unable to allocate memory for WSAQUERYSET\n";
iResult = STATUS_NO_MEMORY;
bContinueLookup = FALSE;
}
} else {
bciout << "=CRITICAL= | | WSALookupServiceNext() failed with error code %d\n", iResult;
bContinueLookup = FALSE;
}
}
}

WSALookupServiceEnd(hLookup);

if ( STATUS_NO_MEMORY == iResult ) {
break;
}
}
}

if ( NULL != pWSAQuerySet ) {
HeapFree(GetProcessHeap(), 0, pWSAQuerySet);
pWSAQuerySet = NULL;
}

if ( bRemoteDeviceFound ) {
iResult = CXN_SUCCESS;
} else {
iResult = CXN_ERROR;
}

return iResult;
}
```

```

int BluetoothClient::send_through_bluetooth (char *command, SOCKADDR_BTH RemoteBthAddr,
int ulRetCode) {

SIZE_T CXN_TRANSFER_DATA_LENGTH = sizeof(command) * strlen(command);

if ( CXN_SUCCESS == ulRetCode) {
ulRetCode = runClient(RemoteBthAddr, command, CXN_TRANSFER_DATA_LENGTH);
}
if (CXN_SUCCESS == ulRetCode){
bciout << "Comando transmitido via Bluetooth a AppRunner_server";
}
return ulRetCode;
}

```

A.1.2 CURSORFEEDBACKTASK

```

// Bluetooth Client
BluetoothClient bluetoothClient = BluetoothClient();
ULONG ulRetCode = CXN_SUCCESS;
SOCKADDR_BTH RemoteBthAddr = {0};
const char *DEVICE_NAME;
std::stringstream s;
std::stringstream t;
int i = 0;
std::chrono::system_clock::time_point inic;

/* ----- BCI-NeuroAge / BCIMobilePhone ----- */
"Application:BCI-NeuroAge string Bluetooth_Device=%BCIMobilePhone %%"
"Bluetooth name of the device that is running the app",
"Application:BCISocialApp string Bluetooth_UUID=%B62C4E8D-62CC-404b-BBBF-
BF3E3BBB1375 %%"
"Bluetooth UUID that identifies the connection",

CursorFeedbackTask::~CursorFeedbackTask()
{
if(WSACleanup() != 0){
bciout << "WSACleanup() failed with error code" << WSAGetLastError();
}
/* ----- */
delete mpFeedbackScene;
}

CursorFeedbackTask::OnInitialize( const SignalProperties& /*Input*/ )
{
/* ----- BCI-NeuroAge / BCIMobilePhone ----- */
string sDevice = Parameter("Bluetooth_Device");

```

```

DEVICE_NAME = sDevice.c_str();

WSADATA wsd;
if(WSAStartup(MAKEWORD(2,2), &wsd) != 0){
    bciout << "WSAStartup() failed with error code" << WSAGetLastError();
    AppLog << "WSAStartup() failed with error code" << WSAGetLastError() << endl;
}
ulRetCode = bluetoothClient.nameToBthAddr(DEVICE_NAME, &RemoteBthAddr);
if ( CXN_SUCCESS != ulRetCode ) {
    bciout << "FATAL_ERROR: Unable to get address of the remote radio having name\n"
<< DEVICE_NAME;
    AppLog << "FATAL_ERROR: Unable to get address of the remote radio having name\n"
<< endl;
} else {
    bciout << "INFO: Bluetooth client is ready";
    AppLog << "INFO: Bluetooth client is ready" << endl;
}
...
}
void
CursorFeedbackTask::DoFeedback( const GenericSignal& ControlSignal, bool&
doProgress )
{
    typedef std::chrono::high_resolution_clock Time;
    typedef std::chrono::duration<double, milli> fsec;
    typedef std::chrono::milliseconds ms;

    char *cmd = "";
    if(i==0){
        inic = Time::now();
    }

    if(i<40)
    {
        s << ControlSignal(0,0) << "_" << ControlSignal(1,0)<< "_";
        i++;
    }
    if(i==40)
    {
        auto fin = Time::now();
        fsec fs = fin-inic;
        s << fs.count();
        int leng = s.str().length()+1;
        cmd = new char [s.str().length()+1];
        std::strcpy (cmd, s.str().c_str());
        // Send the command through Bluetooth connection

```

```

        bluetoothClient.send_through_bluetooth(cmd, RemoteBthAddr, ulRetCode);
        s.str("");
        i=0;
    }
    ...
}
/* _____ */

```

A.2 CÓDIGO GENERADO EN UNITY

A continuación, se muestra parte del código creado para la realización de las tareas, en Unity, mediante el lenguaje C#:

```

\\Scene Controller Code

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class ScenesController : MonoBehaviour
{

    public bool run;
    public int target;
    public int trial;
    //Preferencias
    public bool debug; //Variable indica modo debug
    public int seg_trial; //Segundos que debe durar un trial si el jugador no lo acierta
        antes
    public int numtrials;
    public int tarea;
    //Preferencias para la Tarea 5
    public float seg_tarea5; //Minutos que dura la tarea 5
    public float vel_inic; //velocidad inicial del player
    public bool dismin; //Si se quiere que disminuya la velocidad cuando se produce un fallo
    public float incremvel;
    public float right;
    public float left;
    //Clases de java
    AndroidJavaObject unityPlayer;
    AndroidJavaObject jo;
    AndroidJavaObject interfaceUnity;
    AndroidJavaObject context;

    void Awake()
    {

```

```

DontDestroyOnLoad(gameObject);
}
private void Start()
{
unityPlayer = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
jo = unityPlayer.GetStatic<AndroidJavaObject>("currentActivity");

context = jo.Call<AndroidJavaObject>("getApplicationContext");
interfaceUnity = new AndroidJavaObject("com.gib.bci_nft.InterfaceUnity");
debug = interfaceUnity.Call<bool>("darDebug", context);
seg_trial = interfaceUnity.Call<int>("darSeg_trial", context);
num_trials = interfaceUnity.Call<int>("darNum_trials", context);
tarea = interfaceUnity.Call<int>("darTarea", context);
seg_tarea5 = interfaceUnity.Call<float>("darDur_t5", context);
vel_inic = interfaceUnity.Call<float>("darVel_inicial", context);
dismin = interfaceUnity.Call<bool>("darDism_vel", context);
increm_vel = interfaceUnity.Call<float>("darIncre_vel", context);
SceneManager.LoadScene(tarea, LoadSceneMode.Single);
}

private void Update()
{
run = interfaceUnity.Call<bool>("darRun", context);
target = interfaceUnity.Call<int>("darTarget", context);
trial = interfaceUnity.Call<int>("darTrial", context);
right = interfaceUnity.Call<float>("darRight", context);
left = interfaceUnity.Call<float>("darLeft", context);
}

public void FinReload()
{
interfaceUnity.Call("reloadVariables", context);
}
}

using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;
using System.IO;

public class GlobalControllerT1 : MonoBehaviour {
private GameObject playerR, playerL, limit; //Cubo de la derecha, Cubo de la izquierda,
    Barra que marca el limite para acertar el trial
private ScenesController scenesController; //Objeto para acceder a los datos de las
    preferencias
public TextMesh textTrial, textFinal;

```

```

public Quaternion cubeR, cubeL;
public bool final, debug;
public TextMesh pause_text;
public int posi;
public bool goal, fail;
public int ceros, unos, valor;
//public int[] posMarker;
public float segFinish;
public float segs;
public int n_trials, seg_trail, pause, seg_pause, num_max_right;
public TextWriter writer;
//Valor de decision
public float right;
public float left;
//Variables de sincronizacion
public int target; //2 derecha 1 izquierda
public int i;
public bool run, new_trial;
private int count;
private bool b_run;
public int trial_bci, trial_bci_ant;

private void Awake()
{
Time.timeScale = 1;
aciertos = fallos = posi = ceros = unos = valor = seg_pause = count = trial_bci =
    trial_bci_ant = target = 0;
trial = trialAnt = 1;
segs = 0;
final = run = goal = fail = new_trial = false;
i = 0;
scenesController = GameObject.Find("Controller").GetComponent<ScenesController>();
n_trials = scenesController.n_trials;
seg_trail = scenesController.seg_trial;
debug = scenesController.debug;
num_max_right = (n_trials + 1) / 2;
}

void Start()
{
scenesController.FinReload();
pause_text = GameObject.Find("Pause").GetComponent<TextMesh>();
playerL = GameObject.Find("LeftCube");
playerR = GameObject.Find("RightCube");
limit = GameObject.Find("Limit");
cubeR = playerR.transform.rotation;

```

```

cubeL = playerL.transform.rotation;
textTrial = GameObject.Find("TextTrial").GetComponent<TextMesh>();
textFinal = GameObject.Find("TextFinal").GetComponent<TextMesh>();

if (!File.Exists(Application.persistentDataPath.ToString() + "/" + System.DateTime.Now.
    ToString("HHmmssddMM") + ".csv"))
{
writer = new StreamWriter(Application.persistentDataPath.ToString() + "/" + System.
    DateTime.Now.ToString("HHmmssddMM") + ".csv", false);
}
}

void Update()
{

run = scenesController.run;
trial_bci = scenesController.trial;
if (run && count == 0){
count = 1;
i = 0;
Time.timeScale = 1;
new_trial = true;
target = scenesController.target;
StartCoroutine(NuevoTrial());
playerR.GetComponent<RightCubeController>().MoveRight();
playerL.GetComponent<LeftCubeController>().MoveLeft();
segs = 0;
segFinish = 0;
}
right = scenesController.right;
left = scenesController.left;
if (debug)
{
GameObject.Find("TextRL").GetComponent<TextMesh>().text = "Right:␣" + right + "\nLeft:␣"
    + left;
}
if (trial <= n_trials)
{
segs += Time.deltaTime;
segFinish += Time.deltaTime;
if (trialAnt != trial )
{
target = 0;
if(trial_bci != trial_bci_ant)
{
new_trial = true;

```

```

target = scenesController.target;
trial_bci_ant = trial_bci;
}
StartCoroutine(NuevoTrial());
}
}
else
{
final = true;
StartCoroutine(NuevoTrial());
}
if (Input.GetKey(KeyCode.Escape))
{
Exit();
}
}

IEnumerator NuevoTrial()
{
segs = 0;
playerR.transform.position = new Vector3(4, -1.6f, 0);
playerR.transform.rotation = cubeR;
playerR.GetComponent<RightCubeController>().Stop();
playerL.transform.position = new Vector3(-4, -1.6f, 0);
playerL.transform.rotation = cubeL;
playerL.GetComponent<LeftCubeController>().Stop();
limit.transform.position = new Vector3(0, 0, 10);
if(goal)
{
Time.timeScale = 0;
textTrial.text = "";
if (segFinish < seg_trail)
{
if (segFinish < seg_trail - 2)
{
if (segFinish < seg_trail - 3)
{
textFinal.text = "EXCELENTE!";
}
}
else
{
textFinal.text = "MUY BIEN!";
}
}
}
else
{

```

```
textFinal.text = "BIEN_HECHO!";
}
}
segFinish = 0;
yield return new WaitForSecondsRealtime(1);
goal = false;
textFinal.text = "";
Time.timeScale = 1;
}
if (fail)
{
Time.timeScale = 0;
textTrial.text = "";
textFinal.text = "SIGUE_PROBANDO";
yield return new WaitForSecondsRealtime(1);
fail = false;
textFinal.text = "";
Time.timeScale = 1;
}
segFinish = 0;
if (final && goal == false && fail == false)
{
Time.timeScale = 0;
pause_text.text = "";
textFinal.text = "FINAL\n\nACIERTOS: " + aciertos + "\nFALLOS: " + fallos;
GameObject.Find("Exit").transform.position = new Vector3(-1.4f, 4f, 0);
GameObject.Find("Reload").transform.position = new Vector3(1.4f, 4f, 0);
writer.Close();
}
else
{
if (new_trial)
{
if (target == 2)
{
limit.transform.position = new Vector3(5.5f, 3, 4);
textTrial.text = "DERECHA";
}
if (target == 1)
{
limit.transform.position = new Vector3(-5.5f, 3, 4);
textTrial.text = "IZQUIERDA";
}
trialAnt = trial;
new_trial = false;
}
```

```

playerR.GetComponent<RightCubeController>().MoveRight();
playerL.GetComponent<LeftCubeController>().MoveLeft();
posi++;

}
}
}

public void Exit()
{
writer.Close();
AndroidJavaClass unityPlayer = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
AndroidJavaObject jo = unityPlayer.GetStatic<AndroidJavaObject>("currentActivity");
jo.Call("quitUnityActivity");
}

public void Reload()
{
writer.Close();
SceneManager.LoadScene(1, LoadSceneMode.Single);
}
}

\\RightPlayer Tarea 1 Code

using UnityEngine;
using System.IO;
using System.Collections;

public class RightCubeController : MonoBehaviour {
public float Velocidad = 0.0001f; //Velocidad
public float rotacion;
private bool right;
public GlobalControllerT1 GlobalControl; GlobalControllerT1
public int marker;
public Vector3 posicion;
public float segs = 0;
public int seg_trial = 0;
public TextWriter writer;
public bool run;
public int target;
public float v_right;
public float v_left;

void Start()
{

```

```

GlobalControl = GameObject.Find("GlobalControl").GetComponent<GlobalControllerT1>();
seg_trial = GlobalControl.seg_trial;
writer = GlobalControl.writer;
run = false;
}

void Update()
{
run = GlobalControl.run;
if (run)
{
target = GlobalControl.target;
posicion = transform.position;
//posMarker = GlobalControl.posMarker;
segs = GlobalControl.segs;
marker = GlobalControl.posi - 1;
v_right = GlobalControl.right;
v_left = GlobalControl.left;

rotacion = posicion.y + 1.6f;
this.transform.Rotate(0, rotacion, rotacion);
if (segs <= seg_trial)
{
if (right)
{
writer.WriteLine("1," + GlobalControl.trial + "," + (target-1) + "," + v_right + "," +
v_left);
if (target == 2)
{
if (v_right > 0 && posicion.y <= 4 && right)
{
this.transform.Translate(Vector3.up * Time.deltaTime * (v_right / 2.5f), Space.World);
}
}
if (posicion.y >= 4)
{
GlobalControl.goal = true;
GlobalControl.aciertos++;
GlobalControl.trial++;
marker = marker + 1;
}
}
}
else
{

```

```

if (target == 2)
{
//TimeOut
GlobalControl.trial++;
GlobalControl.fallos++;
marker = marker + 1;
GlobalControl.fail = true;
}

}
}
if (Input.GetKeyUp(KeyCode.Escape))
{
AndroidJavaClass jc = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
AndroidJavaObject jo = jc.GetStatic<AndroidJavaObject>("currentActivity");
jo.Call("onBackPressed");
}
}
public void MoveRight()
{
right = true;
}
public void Stop()
{
right = false;
}
}
}

```

A.3 CÓDIGO GENERADO EN ANDROIDSTUDIO

A continuación, se muestra parte del código creado en AndroidStudio, mediante JavaScript, para generar el menú principal, el menú de preferencias, el servidor Bluetooth...

```

//MainActivity.java

package com.gib.bci_nft;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.graphics.Typeface;

```

```
import android.os.Handler;
import android.os.Message;
import android.preference.PreferenceManager;
import android.support.v4.content.LocalBroadcastManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.ToggleButton;

import com.gib.UnityBCI.UnityPlayerActivity;
import com.unity3d.player.UnityPlayer;

import java.io.IOException;
import java.io.InputStream;
import java.util.Timer;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.util.UUID;

import static android.provider.MediaStore.Audio.PlaylistsColumns.NAME;
import static java.lang.Double.parseDouble;
import static java.lang.Integer.parseInt;
import static java.lang.Long.parseLong;

public class MainActivity extends AppCompatActivity {
    private Button tarea1;
    private Button tarea2;
    private Button tarea3;
    private Button tarea4;
    private Button tarea5;
    private Button salir;
    public int n_trial, seg_trial, seg_pause, tarea, target;
    public float vel_inicial, incre_vel, duracion_t5, right, left;
    public boolean dismin_vel, debug, run;
    public Context context;
    protected static final int REQUEST_CODE = 10;
    protected TextView cmd;
    protected ToggleButton toggleButton;
    public int trial = -1;
    private static BluetoothAdapter mBluetoothAdapter;
```

```

private static BluetoothServerSocket mmServerSocket;
private final static int BLUETOOTH_START = 1;
private static String command = "";
private static String command_now = "";
public int i = 0;

public static final String INTENT_FILTER = "handle_debug_mode";
public static final String INTENT_BUNDLE_VALUE = "DEBUG_MODE";
private static boolean DEBUG_MODE = false;
public static String NAME;
private static UUID MY_UUID;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);
SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(this);

Constants.IS_MENU = pref.getBoolean("isMenu", true);
NAME = pref.getString("pref_bluetooth_name", getResources().getString(R.string.bluetooth))
;
Constants.NAME = NAME;
Constants.UUID = pref.getString("pref_bluetooth_UUID", getResources().getString(R.string.
    UUID));
MY_UUID = UUID.fromString(Constants.UUID);
cmd = (TextView) findViewById(R.id.initial_cmd_list);
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

if (mBluetoothAdapter == null) { cmd.append(">>" + getString(R.string.
    cmd_error_defaultbluetooth)); }
else{
if (mBluetoothAdapter.getScanMode() != BluetoothAdapter.SCAN_MODE_CONNECTABLE) {
Intent discoverableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(discoverableIntent, BLUETOOTH_START);
}
else{
cmd.append(">>" + getString(R.string.cmd_ready)+"\n"); // Command verbose
startServer(null);
}
}
LocalBroadcastManager.getInstance(this).registerReceiver(
mChangeListener, new IntentFilter(INTENT_FILTER));

tarea1 = (Button) findViewById(R.id.button01);
tarea2 = (Button) findViewById(R.id.button02);

```

```
tarea3 = (Button) findViewById(R.id.button03);
tarea4 = (Button) findViewById(R.id.button04);
tarea5 = (Button) findViewById(R.id.button05);
salir = (Button) findViewById(R.id.button06);+
String font_path = "fonts/Arkitech_Light.otf";
Typeface TF = Typeface.createFromAsset(getAssets(), font_path);
tarea1.setTypeface(TF);
tarea2.setTypeface(TF);
tarea3.setTypeface(TF);
tarea4.setTypeface(TF);
tarea5.setTypeface(TF);
salir.setTypeface(TF);
tarea1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v){
establecerPreferencias();
tarea = 1;
seleccionarTarea(tarea);
Intent i = new Intent(MainActivity.this, UnityPlayerActivity.class);
startActivity(i);
}
});

tarea2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v){
establecerPreferencias();
tarea = 2;
seleccionarTarea(tarea);
Intent i = new Intent(MainActivity.this, UnityPlayerActivity.class);
startActivity(i);
}
});

tarea3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v){
establecerPreferencias();
tarea = 3;
seleccionarTarea(tarea);
Intent i = new Intent(MainActivity.this, UnityPlayerActivity.class);
startActivity(i);
}
});

tarea4.setOnClickListener(new View.OnClickListener() {
```

```

@Override
public void onClick(View v){
    establecerPreferencias();
    tarea = 4;
    seleccionarTarea(tarea);
    Intent i = new Intent(MainActivity.this, UnityPlayerActivity.class);
    startActivity(i);
}
});

tarea5.setOnClickListener(new View.OnClickListener(){
@Override
public void onClick(View v){
    establecerPreferencias();
    tarea = 5;
    seleccionarTarea(tarea);
    Intent i = new Intent(MainActivity.this, UnityPlayerActivity.class);
    startActivity(i);
}
});

salir.setOnClickListener(new View.OnClickListener(){
@Override
public void onClick(View v){
    stopServer();
    System.exit(0);
}
});
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    //Para crear el menu primero le pasamos el fichero XML (menu_main) y el objeto donde lo
        queremos crear (menu)
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        lanzarPreferencias(null);
    }
    return super.onOptionsItemSelected(item);
}
}

```

```

public void lanzarPreferencias(View view){
Intent i = new Intent(this, PreferenciasActivity.class);
startActivityForResult(i, REQUEST_CODE);
}

public void seleccionarTarea(int tarea){
context = getApplicationContext();
VariablesGlobales var = ((VariablesGlobales)context);
var.setTarea(tarea);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data){
super.onActivityResult(requestCode, resultCode, data);
if(requestCode == REQUEST_CODE){
establecerPreferencias();
}
if(requestCode == BLUETOOTH_START){
cmd.append(">>" + getString(R.string.cmd_ready)+"\n"); // Command verbose
startServer(null);
}
}

public void establecerPreferencias(){
SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(this);
debug = pref.getBoolean("debug_mode", false);
n_trial = parseInt(pref.getString("trials", "5"));
seg_trial = parseInt(pref.getString("segundos_trial", "5"));
vel_inicial = Float.parseFloat(pref.getString("velInicial", "5"));
incre_vel = Float.parseFloat(pref.getString("increVel", "0.5"));
duracion_t5 = Float.parseFloat(pref.getString("duracion", "3"));
dismin_vel = pref.getBoolean("disminVel", true);
context = getApplicationContext();
VariablesGlobales var = ((VariablesGlobales)context);
var.setDebug(debug);
var.setNum_trials(n_trial);
var.setSeg_trial(seg_trial);
var.setVel_inicial(vel_inicial);
var.setIncre_vel(incre_vel);
var.setDuracion_t5(duracion_t5);
var.setDism_vel(dismin_vel);
}
private BroadcastReceiver mChangeListener = new BroadcastReceiver() {
@Override
public void onReceive(Context context, Intent intentData) {
// you can do anything here
if (intentData != null && intentData.hasExtra(INTENT_BUNDLE_VALUE)) {

```

```

String value = intentData.getStringExtra(INTENT_BUNDLE_VALUE);
startActivity(value);
}
}
};

public void startActivity(String debug_command){
command = debug_command;
DEBUG_MODE = true;
startActivity();
}
public void startActivity() {

// If the app is on debug mode, disable the static activity
if(DEBUG_MODE){ Constants.STATIC_ACTIVITY = 0; }

// Check is command is "sleep" (pause)
if (command.equals("sleep")) {
cmd.append(">>" + getString(R.string.cmd_command) + command + "\n");
if (Constants.SLEEP == 0){
Constants.SLEEP = 1; // When SLEEP_ACTIVITY is 1, the app ignores the following
commands
}else {
Constants.SLEEP = 0;
}
}
}

void startServer(View view) {
if (mBluetoothAdapter.getScanMode() != BluetoothAdapter.SCAN_MODE_CONNECTABLE) {
Intent discoverableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(discoverableIntent , BLUETOOTH_START);
}
try {
BluetoothServerSocket tmp = null;
try {
mmServerSocket = mBluetoothAdapter.listenUsingRfcommWithServiceRecord(NAME, MY_UUID);
} catch (IOException e) { cmd.append(">>" + getString(R.string.cmd_error_rfcom)); }

AcceptThread acceptThread = new AcceptThread();
acceptThread.start();
cmd.append(">>" + getString(R.string.cmd_wait_server) + "\n");

} catch (Exception ex) {
cmd.append(">>" + getString(R.string.cmd_error_launchserver));
}
}
}

```

```
public class AcceptThread extends Thread {
BluetoothSocket mmSocket;
public void run() {
mmSocket = null;
while (true) {
try {
mmSocket = mmServerSocket.accept();
if (mmSocket != null) {
ConnectedThread connected = new ConnectedThread(mmSocket);
connected.start();
}
} catch (IOException e) { break; }
}
}
}

public class ConnectedThread extends Thread {
BluetoothSocket mmSocket;
InputStream mmInStream;

public ConnectedThread(BluetoothSocket socket){
mmSocket = socket;
}

public void run() {
InputStream tmpIn = null;
byte[] buffer = new byte[2048]; // Creating a buffer of 2048B, or 1024 commands,
// since each command has 2B length
int bytes; // Bytes returned from read()
String msg;

try { tmpIn = mmSocket.getInputStream(); }
catch (IOException e) { }

mmInStream = tmpIn;
while (true) {
try {
bytes = mmInStream.read(buffer); // Read from the InputStream
msg = new String(buffer);
break;
} catch (IOException e) {
msg = "ERROR";
break;
}
}
}
```

```

Pattern p = Pattern.compile("\u0000");
Matcher m = p.matcher(msg);
if (m.find()) { msg = m.replaceAll(""); }

if( msg.equals("1") || msg.equals("2")){
VariablesGlobales var = ((VariablesGlobales) context);
run = true;
target = Integer.parseInt(msg);
var.setRun(run);
var.setTarget(target);
var.sumTrial();

}
else{
Message handlerMessage = new Message();
handlerMessage.obj = msg;
handler.sendMessage(handlerMessage);
}

}
}

@Override
protected void onDestroy () {
super.onDestroy ();
try {
mmServerSocket.close ();
} catch (Exception ex) {
cmd.append(">>" + getString(R.string.cmd_error_closeserver));
}
LocalBroadcastManager.getInstance(this).unregisterReceiver(mChangeListener);

SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(this);
pref.edit().putBoolean("isMenu", Constants.IS_MENU).commit();
}

private Handler handler = new Handler() {;
@Override
public void handleMessage(Message msg) {

context = getApplicationContext();
final VariablesGlobales var = ((VariablesGlobales) context);
command = (String) msg.obj;
command_now = command;
i=0;

```

```

final String [] parts = command_now.split("_");
double duration = parseDouble(parts[80]);
final long dur_ms = (long) duration;
Thread thread = (new Thread(new Runnable(){
@Override
public void run()
{
for(i=0; i<79; i++) {
left = Float.parseFloat(parts[i]);
var.setLeft(left);
i++;
right = Float.parseFloat(parts[i]);
var.setRigth(right);
try {
Thread.sleep(1200/100);
} catch (InterruptedException e) {
e.printStackTrace();
}
}
var.setRigth(0);
var.setLeft(0);
i = 0;
}
}));
thread.start();
DEBUG_MODE = false;
}
};

public void stopServer(){
try {
mmServerSocket.close();
cmd.append(">>" + getString(R.string.cmd_server_closed) + "\n");
} catch (Exception ex) {
cmd.append(">>" + getString(R.string.cmd_error_closeserver));
}
}
}
}

```

A.4 CÓDIGO GENERADO EN MATLAB

A continuación, se muestra el código creado para el análisis offline del entrenamiento cognitivo, mediante MatLab:

```

[signal, states, parameters] = load_bcidat('SignalPre');
signal = signal-repmat(mean(signal,2),1,8);

```

```

[dimx,dimy]=size(signal);
fs = 256;
signal = signal - repmat(mean(signal, 1), dimx, 1);
coef = load('FPBand2');
coef = coef.FPBand2;
signalFilt = filtfilt(coef, 1, double(signal));
PSD = pwelch(signalFilt, window(@hamming, 2^13), 7373, 2^13);
PSDNorm = normc(PSD);
f = linspace(0, fs, length(PSD));

PSDUtil = zeros(401,8);
for j = 1:8
for i = 1:401
PSDUtil(i,j) = PSDNorm(i,j);
end
end
band = [0.1 60];
frequency_band = [10.5 13.5; 16.5 19.5; 19.5 22.5];
ind_band = logical((f >= band(1)) & (f <= band(2)));
num_bands = size(frequency_band, 1);

RelativePower = NaN(num_bands, dimy);
for j = 1:dimy
for i = 1:num_bands
ind_frequency_band = min(find(f >= frequency_band(i,1))):max(find(f <= frequency_band(i
,2)));
RelativePower(i,j) = sum(PSDNorm(ind_frequency_band, j));
end
end
RelativePower = RelativePower/sum(PSDNorm(ind_band));

[signal2, states2, parameters2] = load_bcidat('SignalPost');
signal2 = signal2-repmat(mean(signal2,2),1,8);
[dimx,dimy]=size(signal2);
signal2 = signal2 - repmat(mean(signal2, 1), dimx, 1);
signalFilt2 = filtfilt(coef, 1, double(signal2));

PSD2 = pwelch(signalFilt2, window(@hamming, 2^13), 7373, 2^13);
PSDNorm2 = normc(PSD2);
f2 = linspace(0, fs, length(PSD2));
for j = 1:8
for i = 1:401
PSDUtil2(i,j) = PSDNorm2(i,j);
end
end
f3 = linspace(0, 25, length(PSDUtil2));

```

```

PSDMean = mean(PSDUtil, 2);
PSDMean2 = mean(PSDUtil2, 2);
PSDMean = PSDMean';
PSDMean2 = PSDMean2';

options.handle = figure(1);
options.alpha = 0.5;
options.line_width = 2;
options.x_axis = f3;
options.error = 'std';
options.color_area = [128 193 219]./255;
options.color_line = [ 52 148 186]./255;

plot_areaerrorbar(PSDMean, options);
hold on;
xlabel('Frecuencia (Hz)');
ylabel('PSD Normalizada');
options.color_area = [243 169 114]./255;
options.color_line = [236 112 22]./255;
plot_areaerrorbar(PSDMean2, options);hold on;
hold off;

options.handle = figure(2);
options.color_area = [128 193 219]./255;
options.color_line = [ 52 148 186]./255;
subplot(1,2,1);
plot_areaerrorbar(PSDMean, options);
hold on;
options.color_area = [243 169 114]./255;
options.color_line = [236 112 22]./255;
subplot(1,2,2);
plot_areaerrorbar(PSDMean2, options);
hold on;
hold off;
RelativePower2 = NaN(num_bands,dimy);
ch = ['F3';'F4';'T7';'C3';'Cz';'C4';'T8';'Pz'];
figure(3);
for j = 1:8
subplot(2,4,j);
fill([10.5 10.5 13.5 13.5], [0 0.44 0.44 0], [75 0 130]./255, 'facealpha', 0.5, '
    edgcolor', 'none'); hold on;
fill([16.5 16.5 19.5 19.5], [0 0.44 0.44 0], [106 90 205]./255, 'facealpha', 0.5, '
    edgcolor', 'none'); hold on;
fill([19.5 19.5 22.5 22.5], [0 0.44 0.44 0], [128 0 128]./255, 'facealpha', 0.5, '
    edgcolor', 'none'); hold on;
title(['Canal ' ch(j,:)]);

```

```

plot(f3, PSDUtil(:,j), 'b');
hold on
plot(f3, PSDUtil2(:,j), 'r'); hold on;

for i = 1:num_bands
ind_frequency_band = min(find(f >= frequency_band(i,1))):max(find(f <= frequency_band(i
,2)));
RelativePower2(i,j) = sum(PSDNorm2(ind_frequency_band,j));
end
hold off;
end
legend('Banda 12 Hz', 'Banda 18 Hz', 'Banda 21 Hz', 'PSD Antes NFT', 'PSD Despues NFT');
RelativePower2 = RelativePower2/sum(PSDNorm2(ind_band));
legp = [12,18,21];
for i = 1:1:3
p = signrank(RelativePower(i,:), RelativePower2(i,:));
fprintf('- Banda %a Hz: %.4f \n', legp(i), p);
end

pr = RelativePower2-RelativePower;
figure(4);
for i = 1:1:3
subplot(1,3,i);
title([ num2str(legp(i)) ' Hz']);
colorbar;
topoplot(pr(i,:), '8locTFG.loc', 'maplimits', [ min(pr(:)), max(pr(:))]);
hold on;

```

Apéndice B

CUESTIONARIO DE SATISFACCIÓN

CUESTIONARIO DE SATISFACCIÓN DE LA APLICACIÓN BCI-NeuroAge

Valore las siguientes afirmaciones sobre la aplicación BCI-NeuroAge y el entrenamiento realizado.

1. Me ha resultado interesante conocer y utilizar esta aplicación BCI.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

2. Las sesiones de entrenamiento me han resultado aburridas.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

3. La interfaz del sistema es intuitiva y fácil de usar.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

4. Las sesiones requieren demasiada concentración, lo que termina cansándome.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

5. El feedback de las distintas tareas me gusta y me motiva para conseguir el objetivo.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

6. La duración de las sesiones me ha parecido excesiva.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

7. He conseguido mejorar en el manejo de la aplicación después de las sesiones de entrenamiento.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

8. La aplicación tiene poca utilidad y no me ha parecido interesante.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

9. La adquisición de la actividad cerebral mediante el EEG con el gorro y electrodos es poco práctica.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

10. ¿Volvería a participar en un proyecto de características similares?

- a) Sí.
- b) Sí, aunque me supone bastante esfuerzo.
- c) No. ¿Por qué razón?

11. ¿Qué aspectos de la aplicación se podrían mejorar?

Apéndice C

PLIEGO DE CONDICIONES

Este anexo contiene las condiciones legales que guiarán a la realización del Trabajo Fin de Grado titulado como “Diseño y desarrollo de una plataforma de entrenamiento cognitivo mediante un sistema Brain Computer Interface para dispositivos Android”.

En lo que sigue se supondrá que el trabajo ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de utilizar el sistema BCI. Dicha empresa ha debido desarrollar una línea de investigación con el objeto de llevar a cabo el trabajo. Esta línea de investigación, junto con el posterior desarrollo de la aplicación, está amparada por las condiciones particulares del siguiente pliego. Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes condiciones:

Condición 1

La modalidad de contratación será el concurso. La adjudicación se hará por tanto a la proposición más favorable, sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho de declararlo desierto.

Condición 2

El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

Condición 3

En la oferta se hará constar el precio total por el que se compromete a realizar la obra y el

tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

Condición 4

La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos que sea preciso para el desarrollo de la misma.

Condición 5

Aparte del Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no está obligado a aceptarla.

Condición 6

El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

Condición 7

Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le haya comunicado por escrito el Ingeniero Director de obras siempre que dicha obra se haya ejecutado a los preceptos de los pliegos de condiciones, de acuerdo a los cuales se harán modificaciones y la valoración de las diversas unidades, sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto no podrán servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

Condición 8

Tanto en las certificaciones de la obra como en la liquidación final, se abonarán los tra-

bajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

Condición 9

Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata, pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero considere justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

Condición 10

Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere, y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento se sujetarán siempre al establecido en el punto anterior.

Condición 11

Cuando el contratista, con la autorización del Ingeniero Director de obras, emplee material de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio, ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sino a lo que le correspondería si se hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

Condición 12

Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ella se formen, o en su

defecto, por lo que resulte de su medición final.

Condición 13

El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras, así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

Condición 14

Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

Condición 15

La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

Condición 16

La forma de pago será por certificaciones mensuales de la obra ejecutada de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

Condición 17

La fecha de comienzo de las obras será a partir de los quince días naturales del replanteo oficial de las mismas, y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna a la reclamación de la fianza.

Condición 18

Si el contratista, al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

Condición 19

El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras o con el delegado que este designe, para todo lo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

Condición 20

Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista la conservación de la obra ejecutada hasta la recepción de la misma, por lo que el deterioro total o parcial de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o construido por su cuenta.

Condición 21

El contratista deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa por retraso en la ejecución siempre que este no sea debido a causas de fuerza mayor. A la terminación de la obra se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

Condición 22

Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas establecidas.

Condición 23

Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad

“Presupuesto de Ejecución por contrata”, y anteriormente llamado “Presupuesto de Ejecución Material” que hoy designa otro concepto.

La empresa constructora, que ha desarrollado este proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares.

Condición particular 1

La propiedad intelectual de los procesos descritos y analizados en el presente trabajo pertenece por entero a la empresa constructora representada por el Ingeniero Director del Proyecto.

Condición particular 2

La empresa constructora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos posteriores, para la misma empresa cliente o para otra.

Condición particular 3

Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con la autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

Condición particular 4

En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

Condición particular 5

En todas las reproducciones se indicará su procedencia, explicando el nombre del proyecto,

nombre del Ingeniero Director y empresa consultora.

Condición particular 6

Si el proyecto pasa a la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de este, la empresa consultora decidirá aceptar o no la modificación propuesta.

Condición particular 7

Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta al añadirla.

Condición particular 8

Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

Condición particular 9

Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

Condición particular 10

La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

Condición particular 11

La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia de este hecho. En este caso deberá autorizar expresamente los

proyectos presentados por otros.

Condición particular 12

El Ingeniero Director del proyecto será responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona asignada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.

Los requisitos, clasificados como documentación, hardware y software, que han sido necesarios para la elaboración del presente trabajo fin de grado son los siguientes.

DOCUMENTACIÓN

- Servicio de la Biblioteca de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid.

HARDWARE

- Equipo portátil Intel® Core™ i7 quad-core 6700HQ, con 2.6 GHz y 8 GB de RAM.
- Tablet con CPU Quad-Core 2.15 GHz, 4 GB de RAM.
- Equipo BCI:
 - Amplificador g.USBamp de 16 canales.
 - Gorro EEG de 64 canales.
 - Electrodo EEG de oro.
 - Electrodo de referencia.
 - Gel conductor.

SOFTWARE

- Microsoft® Windows 10 Home.
Copyright© 2013, Microsoft Corporation.

- TeXstudio 2.12.8.
- MATLAB® Student R2012a (7.14.0.739).
Copyright© 2012, The MathWorks Inc.
- Adobe Photoshop CS5 Extended (Versión 12.0).
Copyright© 1990–2010, Adobe Systems Incorporated.
- Adobe® Reader® XI (Versión 11.0.07).
Copyright© 1984–2012, Adobe Systems Incorporated.
- Google Chrome (Versión 65.0.3325.181).
Copyright© 2018, Google Inc.
- Microsoft® Visual Studio Professional (Versión 11.0.50727.1).
Copyright© 2012, Microsoft Corporation.
- IntelliJ® Platform Android Studio 3.0.1.
Copyright© 2017, IntelliJ Platform.
- Unity 2017.2.0f3 Personal.
Copyright© 2017, Unity Technologies ApS
- g.USBamp Simulink High-Speed Online Processing.
- Plataforma de propósito general BCI2000.

Apéndice D

PRESUPUESTO

El presupuesto estimado de acuerdo a las condiciones expuestas en el anexo de pliego de condiciones es el siguiente.

EJECUCIÓN MATERIAL

Equipo portátil Intel® Core™ i7 quad-core 6700HQ, con 2.6 GHz y 8 GB de RAM	926€
Tablet con CPU Quad-Core 2.15 GHz, 4 GB de RAM	689€
Equipo BCI	13.300€
Sistema Operativo: Microsoft® Windows 10 Home	145€
Software: Microsoft® Office Home and Student 2016	149€
Software: MATLAB® Student R2012a	427€
Software: Adobe Photoshop CS5 Extended	80€
<hr/>	
TOTAL	15.716€

GASTOS GENERALES

16 % sobre la ejecución material	2.514,56€
----------------------------------	-----------

BENEFICIO INDUSTRIAL

6 % sobre la ejecución material	942,96€
---------------------------------	---------

MATERIAL FUNGIBLE

Gastos de Impresión (tinta y papel)	200€
Encuadernación	120€
DVDs	2€
<hr/>	
TOTAL	322€

HONORARIOS DEL PROYECTO

800 horas a 30€/hora	24.000€
----------------------	---------

SUBTOTAL DEL PRESUPUESTO

Subtotal del presupuesto	43.495,52€
--------------------------	------------

I.V.A. APLICABLE

21 % sobre el subtotal de presupuesto	9.134,06€
---------------------------------------	-----------

TOTAL DEL PRESUPUESTO

Total del presupuesto	52.629,58€
------------------------------	-------------------

El total del presupuesto, en euros, asciende a:

<p>CINCUENTA Y DOS MIL SEISCIENTOS VEINTINUEVE EUROS CON CINCUENTA Y OCHO CÉNTIMOS DE EURO</p>

En Valladolid, Mayo de 2018.

Fdo. Chiara Franzoni Uriarte.

