



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

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

**Diseño y desarrollo de una aplicación para
navegar por Internet mediante sistemas Brain
Computer Interface orientada a personas con
grave discapacidad**

Autor:

D. Víctor Martínez Cagigal

Tutor:

Dr. D. Roberto Hornero Sánchez

Valladolid, 26 de Junio de 2014

TÍTULO: **Diseño y desarrollo de una aplicación para navegar por Internet mediante sistemas Brain Computer Interface (BCI) orientada a personas con grave discapacidad**

AUTOR: **D. Víctor Martínez Cagigal**

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

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

TRIBUNAL

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

VOCAL: **Dr. D. Jesús Poza Crespo**

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

SUPLENTE **Dr. D. Carlos Gómez Peña**

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

FECHA: **8 de Julio de 2014**

CALIFICACIÓN:

Resumen del TFG

El electroencefalograma (EEG) fue desarrollado por Hans Berger en 1929 y las ondas cerebrales, hasta entonces desconocidas, fueron investigadas durante el siglo XX como ayuda para el diagnóstico de la epilepsia o diversos trastornos del sueño. Durante este periodo de tiempo se especuló sobre utilizar las señales EEG para desarrollar un sistema de comunicación entre el cerebro y el medio sin la intervención de los mecanismos normales de los nervios y los músculos periféricos. Ese sistema surgió a cargo de Dr. Jacques Vidal en 1977 y se bautizó como Brain Computer Interface (BCI). Estos tipos de sistemas se basan en monitorizar la actividad cerebral y traducir determinadas características, correspondientes a las intenciones del usuario, en comandos de un dispositivo.

En primer lugar, en el presente trabajo se lleva a cabo una revisión del estado del arte sobre los sistemas BCI, los métodos empleados para registrar la actividad cerebral, las señales de control que gobiernan dichos sistemas, el procesado de la señal y sobre los navegadores web desarrollados con anterioridad. A continuación, se estudia en profundidad el análisis de la señal y los potenciales evocados P300, señal de control utilizada en la aplicación a desarrollar. En este análisis se detallan los artefactos comunes que degradan la señal y los distintos métodos de extracción y traducción de características. Entre los métodos de extracción se encuentran el análisis espacial, temporal o frecuencial; y con respecto a los métodos de traducción se tratan tanto los lineales como los no lineales. Adicionalmente, se establecen las pautas para diseñar un umbral basado en las curvas ROC que discrimine correctamente dos poblaciones.

Posteriormente, se presenta el diseño y desarrollo de una aplicación BCI para navegar por Internet mediante las ondas cerebrales, destinada a las personas que sufren una grave discapacidad. Esta aplicación utiliza los potenciales evocados P300 como señal de control gracias al paradigma *odd-ball*, basado en exponer una serie de estímulos objetivo infrecuentes camuflados entre estímulos no-objetivo frecuentes. El usuario involuntariamente generará un potencial P300 cada vez que detecte los estímulos infrecuentes, seleccionando comandos que gobernarán la aplicación. Dicha aplicación fue evaluada satisfactoriamente por parte de cinco sujetos sanos en tres sesiones distintas. Finalmente, se exponen los resultados de la evaluación, se discuten y se extraen las conclusiones oportunas.

Palabras clave

Brain Computer Interface, electroencefalograma, potenciales evocados P300, extracción y traducción de características, artefactos, umbral, curvas ROC, navegador web, paradigma *odd-ball*.

Abstract

Electroencephalogram (EEG) was developed by Hans Berger in 1929 and brain signals, previously unknown, were researched over the 20th century in order to help the epilepsy or sleeping disorders diagnosis. During this period of time, people speculated that those signals might be used for developing a communication and control system between brain and environment without the normal intermediaries of peripheral nerves and muscles. In 1977, this system was invented by Dr. Jacques Vidal and it was called Brain Computer Interface (BCI). These kinds of systems are based on monitoring the brain activity and translating certain features, which reflect user intentions, into device commands.

Firstly, this project performs a review of state-of-the-art about BCI systems, methods applied in order to register brain activity, controls signals which lead these systems, signal processing and about previously developed BCI web browsers. Signal analysis and P300 evoked potentials, the selected control signal, are deeply studied next. In this analysis, common artifacts which degrade the EEG signal and various feature extraction and translation methods are detailed. Among feature extraction methods are included spatial, temporal and spectral analysis. Regarding the feature translation, lineal and non-lineal methods are treated. In addition, guidelines in order to designing a threshold based on ROC curves which correctly discriminates two populations are established too.

Later, a BCI application is developed. This application aims to allow disabled people surfing the Internet using only their own brain signals. The signal control which leads the application using the *odd-ball* paradigm is the P300 evoked potential. This paradigm is based on presenting several infrequent target stimuli interspersed with frequent routine stimuli. Typically and unwittingly, user will evoke in the EEG over parietal cortex a positive peak at about 300ms when the user detects the infrequent stimuli. The application was successfully evaluated by five healthy subjects in three different sessions. Finally, evaluation results are exposed and discussed and appropriate conclusions are drawn.

Keywords

Brain Computer Interface, electroencephalogram, P300 evoked potentials, feature extraction and translation, artifacts, threshold, ROC curves, web browser, *odd-ball* paradigm.

AGRADECIMIENTOS

En primer lugar quisiera agradecer a Roberto Hornero Sánchez la oportunidad que me ha brindado para realizar este trabajo y la paciencia que ha demostrado al resolver mis constantes dudas a lo largo del mismo.

Al Grupo de Ingeniería Biomédica de la Universidad de Valladolid, en especial a Rebeca Corralejo Palacios y a Javier Gómez Pilar, por proveerme una ayuda constante a la hora de introducirme en el mundo de la ingeniería biomédica y los sistemas BCI.

A todos mis compañeros y amigos, porque vuestra ayuda, vuestros consejos y vuestra presencia a lo largo de mi vida han hecho posible que forje mi personalidad y concluya este ciclo felizmente.

A Andrea, por aguantarme como nadie sabe hacerlo.

Y finalmente, el mayor de los agradecimientos es para mi familia, especialmente para mis padres y mi hermana, por el esfuerzo que han demostrado para encarrilarme e inculcarme los principios de los que me enorgullezco, y por el apoyo que he recibido todos estos años.

ÍNDICE GENERAL

CAPÍTULO 1	
INTRODUCCIÓN	1
1.1. Señales biomédicas	1
1.2. Electroencefalograma (EEG)	3
1.3. Brain Computer Interface (BCI)	5
1.4. Objetivos del Trabajo de Fin de Grado	6
1.5. Estructura del Trabajo de Fin de Grado	7
CAPÍTULO 2	
SISTEMAS BRAIN COMPUTER INTERFACE	9
2.1. Introducción	9
2.2. Métodos para registrar la actividad cerebral	11
2.3. Tipos de señales de control en BCI	13
2.3.1. Potenciales Evocados Visuales	14
2.3.2. Potenciales Corticales Lentos	15
2.3.3. Ritmos sensoriomotores (ritmos μ y β)	16
2.3.4. Potenciales de neuronas corticales	18
2.3.5. Potenciales evocados P300	19
2.4. Procesado de la señal EEG	20
2.4.1. Adquisición de la señal	20
2.4.2. Procesado de la señal	22
2.4.3. Aplicación	27
2.5. Aplicaciones de los sistemas BCI	27
2.5.1. Selección de letras	27
2.5.2. Movimiento de un cursor	28
2.5.3. Desplazamiento sobre un mapa	28
2.6. Navegadores Web BCI	30
2.6.1. Bensch et al — “Nessi: An EEG-Controlled Web Browser for Severely Paralyzed Patients” [19]	30
2.6.2. Mugler <i>et al</i> — “Design and Implementation of a P300-Based Brain-Computer Interface for Controlling an Internet Browser” [22]	31
2.6.3. Sirvent Blaco <i>et al</i> — “Visual Evoked Potential-Based Brain-Machine Interface Applications to Assist Disabled People” [23]	32
2.6.4. Yu <i>et al</i> — “Surfing the Internet with a BCI Mouse” [24]	34
2.6.5. Liu <i>et al</i> — “An Online Brain-Computer Interface Using Non-Flashing Visual Evoked Potentials” [25]	36
2.6.6. Comparativa de los navegadores web	38

CAPÍTULO 3	
ANÁLISIS DE LA SEÑAL Y POTENCIALES EVOCADOS P300	43
3.1. Potenciales evocados P300	43
3.1.1. Aspectos psicológicos	45
3.1.2. Diferencias individuales entre sujetos	46
3.1.3. Factores farmacológicos	48
3.1.4. Enfermedades neuronales	48
3.2. Procesado de los potenciales evocados P300	49
3.2.1. Extracción de características	49
3.2.2. Traducción de características	58
3.3. Umbralización	63
3.3.1. Curva ROC	64
CAPÍTULO 4	
DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE NAVEGACIÓN WEB	69
4.1. Objetivo de la aplicación	69
4.2. Partes de la aplicación	69
4.3. Funcionamiento general de la aplicación	70
4.4. Adquisición de la señal EEG	71
4.5. BCI Browser: P3Speller	72
4.5.1. Sistema de propósito general BCI2000	72
4.5.2. Configuración de los parámetros	73
4.5.3. Matrices <i>odd-ball</i>	75
4.6. BCI Browser: Navigator	77
4.7. Sujetos y tareas	79
4.7.1. Primera sesión	79
4.7.2. Segunda sesión	80
4.7.3. Tercera sesión	81
CAPÍTULO 5	
RESULTADOS	83
5.1. Determinación del umbral	83
5.2. Tareas propuestas	89
5.3. Cuestionario de satisfacción	94
CAPÍTULO 6	
DISCUSIÓN	97
6.1. Determinación del umbral	97
6.2. Tareas propuestas	99
6.3. Cuestionario de satisfacción	101

6.4. Comparación con otros estudios	102
6.5. Limitaciones de la aplicación	103
CAPÍTULO 7	
CONCLUSIONES Y LÍNEAS FUTURAS	105
7.1. Conclusiones	105
7.2. Líneas Futuras	107
REFERENCIAS	109
ANEXO A - ÍNDICE DE ACRÓNIMOS	113
ANEXO B - CÓDIGO GENERADO EN C++ Y JAVASCRIPT	117
B.1. Aplicación BCI Browser: Navigator	117
B.2. Aplicación BCI Browser: P3Speller	126
ANEXO C - TAREAS Y CUESTIONARIO	137
C.1. Tareas propuestas	137
C.2. Cuestionario de satisfacción	139
ANEXO D - PLIEGO DE CONDICIONES	141
ANEXO E - PRESUPUESTO	149

ÍNDICE DE FIGURAS

Capítulo 1: Introducción

- Figura 1-1.** Ejemplos de las tres señales bioeléctricas principales recogidas sobre la superficie corporal: (a) un electroencefalograma (EEG), (b) un electrocardiograma (ECG), y (c) un electromiograma (EMG). 2
- Figura 1-2.** Imágenes PET que detectan la disminución de glucosa asociada con la enfermedad del Alzheimer. A la izquierda un sujeto sano, en el centro un sujeto con DCL (Deterioro Cognitivo Leve) y a la derecha un enfermo de Alzheimer. 3
- Figura 1-3.** Ritmos electroencefalográficos observados durante el paso de vigilia a sueño profundo: (a) agitado, (b) relajado, (c) adormecido, (d) dormido, (e) sueño profundo. Se aprecia una disminución frecuencial a medida que el sujeto se duerme. 4
- Figura 1-4.** Distribución de los electrodos según el sistema internacional 10/20. 5
- Figura 1-5.** Icono diseñado para la aplicación web. 7

Capítulo 2: Sistemas Brain Computer Interface

- Figura 2-1.** Esquema del primer sistema BCI controlado por VEP y desarrollado por Dr. Jacques Vidal en 1977. En (a) aparecen los botones que se iluminaban intermitentemente para controlar el cursor. En (b) aparece el laberinto del cual debían salir los usuarios moviendo el cursor [7] 9
- Figura 2-2.** Estructura habitual que siguen los sistemas BCI desde la adquisición de la señal cruda hasta la conversión de ésta en comandos de control de un dispositivo 11
- Figura 2-3.** Localización de los electrodos de distintas técnicas que emplean señales eléctricas 13
- Figura 2-4.** Regiones en las que se divide la corteza cerebral 14
- Figura 2-5.** Espectro frecuencial de una señal EEG recogida durante una estimulación visual a 7Hz. El fenómeno de resonancia de los potenciales VEP produce picos en 7Hz y sus armónicos 15
- Figura 2-6.** SCP recogidos durante una selección binaria [5] 15
- Figura 2-7.** Ritmos μ y β . En la selección inferior se percibe una disminución de los mismos [5] 16
- Figura 2-8.** Señal EEG registrada antes y después de realizar un movimiento con el dedo índice de la mano donde se aprecian los ERD y ERS. El panel superior se corresponde al electrodo C3 [5] 17
- Figura 2-9.** Esquema que muestra la implantación de electrodos epidurales o intracorticales [5] 18
- Figura 2-10.** Potencial evocado P300 recogido sobre el área centro-parietal del córtex mediante la técnica *odd-ball* con una matriz luminosa [5] 19

Figura 2-11. Etapas en las que se divide el tratamiento de la señal EEG: adquisición de la señal, procesamiento de la señal (extracción y traducción de características) y aplicación 20

Figura 2-12. Artefactos en la señal EEG causados por (a) el movimiento ocular y (b) pestañeos repetitivos y voluntarios..... 21

Figura 2-13. Registro de 5 segundos de duración de la señal EEG contaminada por episodios intermitentes de artefactos electromiográficos (EMG) 22

Figura 2-14. Comparación entre cuatro métodos espaciales de extracción de características. (A) Localización de los electrodos utilizados con la señal objetivo en rojo, concretamente, se quiere medir la señal del electrodo C3. (B) Banda de paso para cada método, raíz cuadrada de los valores cuadráticos medios de la señal recogida en C3. (C) Topografía de r_2 medida y amplitud espectral para cada método estudiado [5] 24

Figura 2-15. Clasificador lineal que maximiza el margen mínimo: en (a) se muestra el comportamiento óptimo y en (b) su comportamiento en presencia de un *outlier* 25

Figura 2-16. Comparación de los hiperplanos de separación para los tres métodos de traducción de características: K1 es un método lineal de una dimensión, K2 es un método lineal de dos dimensiones y K3 es un método no lineal..... 26

Figura 2-17. Selección de letras utilizando potenciales corticales lentos (SCP) con selecciones binarias: mitad de las mismas en magenta y la otra mitad en azul [19] 28

Figura 2-18. Control del movimiento de un cursor verticalmente mediante potenciales corticales lentos para seleccionar entre dos opciones..... 29

Figura 2-19. Control del movimiento en videojuegos mediante ritmos sensoriomotores μ y β [20] 29

Figura 2-20. Control del desplazamiento sobre un mapa con potenciales evocados visuales (VEP) [21]..... 29

Figura 2-21. Interfaz gráfica de *Nessi*, el primer navegador basado en potenciales SCP. En (a) se muestra la interfaz de selección binaria y la realimentación al usuario, mientras que en (b) aparecen las distintas codificaciones del teclado virtual [19] 31

Figura 2-22. Matriz utilizada para provocar los potenciales evocados P300 en el navegador web desarrollado por Mugler *et al* [22] 32

Figura 2-23. Potenciales recogidos en el estudio de Sirvent Blasco *et al* [23]. En (a) se muestra el potencial evocado N2PC, producido unos 200ms después del estímulo y recogido sobre el electrodo PO8 (zona parietal-occipital). En (b) se muestra el potencial evocado P300, producido unos 300ms después del estímulo y recogido sobre el electrodo Cz..... 33

Figura 2-24. Máquina de estados que gobierna las transiciones entre las matrices del navegador web de Sirvent Blasco *et al* [23] 34

Figura 2-25. Interfaz gráfica del navegador web desarrollado por Yu *et al* [24]. Se puede observar cómo se muestra un rectángulo a medida que el cursor se acerca al hipervínculo 35

Figura 2-26. Método de selección de letras, números y comandos implementado con potenciales evocados P300 y usado en el navegador de Yu *et al* [24] 36

Figura 2-27. Interfaz de usuario para el análisis *offline* en el estudio de Liu *et al* [25] donde se prueba la eficacia de los mVEP..... 37

Figura 2-28. Interfaz de usuario para el análisis *online* en el estudio de Liu *et al* [25] donde se estudia el rendimiento de la aplicación web..... 38

Capítulo 3: Análisis de la Señal y Potenciales Evocados P300

Figura 3-1. Potencial evocado P300 recogido en el electrodo Pz y generado con un *odd-ball* auditivo donde el usuario debía detectar tonos objetivos con probabilidad de ocurrencia de 0,2. La curva punteada (a) se corresponde con la primera época registrada y la curva (b) se corresponde con un promediado sincronizado de 90 épocas, permitiendo distinguir el potencial. La línea roja rayada indica el momento en el que se presentó el estímulo 44

Figura 3-2. Componentes del potencial evocado P300 en dos localizaciones distintas: Fz y Pz. En la gráfica superior aparece la respuesta cuando el sujeto no atiende al estímulo, donde se observa que la onda P3a sigue apareciendo. En la gráfica inferior, por el contrario, aparece la respuesta cuando el sujeto atiende al estímulo, generando las tres componentes principales: P3a, P3b y *slow wave* [29] 45

Figura 3-3. Variación de la forma del P300 para el electrodo Pz en función de la probabilidad de estímulo (a) y la probabilidad temporal (b) al utilizar un paradigma *odd-ball* auditivo [29] 46

Figura 3-4. Potenciales evocados P300 recogidos en el vértex (Cz) y en un electrodo nasofaríngeo (Pg2) según la dificultad de discriminación de los estímulos objetivos [29] 46

Figura 3-5. Variaciones de la forma del potencial evocado P300 según la edad del sujeto. En (a) se observa la variación de la latencia con la edad, pudiendo distinguir tres tramos distintos: de 5 a 12, de 12 a 20, y de 20 años en adelante. En (b) se observa el potencial P300 y su localización en Fz y Pz para niños de 7-8años y para adultos, pudiendo observar la onda positiva de los niños en la zona frontal y la latencia del P300 en la zona parietal [29] 47

Figura 3-6. Potencial evocado P300 recogido en el electrodo Pz y generado con un *odd-ball* auditivo para un sujeto esquizofrénico (en rojo) y un sujeto sano de control (en azul) con un promediado de 90 épocas 49

Figura 3-7. Tres filtrados espaciales diferentes para obtener la señal del canal C3: (a) Laplaciano corto, (b) Laplaciano largo y (c) método de referencia de media común (CAR). El electrodo objetivo C3 se corresponde con el círculo vacío y sus vecinos con círculos rellenos 52

Figura 3-8. Búsqueda de la dirección que consigue que la varianza de los datos proyectados sea máxima (D_1) en un problema de análisis de componentes principales 54

Figura 3-9. Esquema del filtrado adaptativo para eliminar una fuente de ruido..... 56

Figura 3-10. Plano tiempo-frecuencia para STFT en (a), con resoluciones fijas; DWT en (b), con resoluciones variables adaptadas a las señales biomédicas; y WPT en (c), con resoluciones variables según la intención del usuario 58

Figura 3-11. En la figura superior, representación del promediado sincronizado de 90 épocas de P300 para un *odd-ball* auditivo. En el centro, representación del espectrograma del mismo obtenido a través de la STFT. En la figura inferior, representación del escalograma del mismo obtenido a través de la CWT 59

Figura 3-12. Representación gráfica del hiperplano óptimo de separación para el algoritmo SVM en una clasificación binaria	62
Figura 3-13. Posibles errores en la clasificación SVM. En (a) el dato considerado se encuentra dentro de la región de separación y se clasifica correctamente. En (b) el dato considerado se encuentra dentro de la región de separación y se clasifica incorrectamente.....	63
Figura 3-14. Matriz de confusión que recoge los cuatro posibles resultados tras aplicar un clasificador binario: las columnas establecen el valor real de la clase y las filas el valor predicho por el clasificador	65
Figura 3-15. Construcción de una curva ROC (b) para las clases positivo (P) y negativo (N) con distribuciones gaussianas (a).....	65
Figura 3-16. Detección del umbral óptimo: punto que minimiza la distancia entre éste y el punto óptimo de la curva ROC. Este umbral óptimo maximiza el par <i>Sensibilidad-Especificidad</i>	66
Figura 3-17. Área bajo la curva ROC (AUC) en tres casos distintos. En (a) se obtiene un valor diagnóstico perfecto, en (b) un buen valor diagnóstico y en (c) la peor curva posible, sin valor diagnóstico alguno	67

Capítulo 4: Diseño de una Aplicación de Navegación Web

Figura 4-1. Estructura principal de la aplicación de navegación web desarrollada	70
Figura 4-2. Interfaz gráfica de la aplicación desarrollada.....	71
Figura 4-3. Disposición de los electrodos utilizados en la aplicación según el sistema internacional 10/20: Fz, Cz, Pz, P3, P4, PO7, PO8 y Oz.....	71
Figura 4-4. Estructura del sistema de propósito general BCI2000, compuesta por cuatro módulos independientes.....	72
Figura 4-5. Interfaz gráfica del “Operador”, donde se configuran los parámetros relativos a la aplicación, además de permitir el control de ésta a tiempo real	73
Figura 4-6. Optimización de la aplicación para pantallas panorámicas con relación de aspecto 16:9.....	74
Figura 4-7. Línea de tiempo de cada intento de <i>BCI Browser: P3Speller</i>	75
Figura 4-8. Matrices utilizadas en la aplicación <i>BCI Browser: P3Speller</i> . A la izquierda se muestra la matriz de navegación, a la derecha, la matriz de teclado	76
Figura 4-9. Funciones del comando “←” de la matriz de navegación en función del estado de la aplicación en el momento de su selección.....	77
Figura 4-10. Etiquetado de hipervínculos con <i>BCI Browser: Navigator</i> . En rojo se indica la posición del <i>buffer</i> que contiene el carácter introducido por el usuario, en verde los nodos posibles que concuerdan con la codificación introducida (o posibles selecciones en potencia) y en gris el resto de nodos que no pueden seleccionarse con esa codificación	78

Figura 4-11. Interfaz gráfica de la herramienta *P300Classifier*, utilizada para hallar el vector de pesos adaptado para cada usuario a partir de una serie de muestras de entrenamiento 80

Figura 4-12. Representación gráfica de los datos de la segunda sesión para el sujeto U1. A la izquierda aparecen las selecciones atendiendo e ignorando los estímulos para las 24 selecciones de la segunda sesión. A la derecha aparece la curva ROC para esa sesión concreta y el umbral óptimo calculado 81

Figura 4-13. El sujeto U5 evaluando la aplicación en la tercera sesión. La pantalla de la izquierda contiene la aplicación mientras que la de la derecha muestra el voltaje de cada electrodo y la información sobre las celdas seleccionadas y el umbral 82

Capítulo 5: Resultados

Figura 5-1. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U1. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión 84

Figura 5-2. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U2. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión 85

Figura 5-3. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U3. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión 87

Figura 5-4. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U4. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión 88

Figura 5-5. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U5. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión 90

Capítulo 6: Discusión

Figura 6-1. Comparación del valor de AUC para las dos primeras sesiones destinadas a la determinación del umbral óptimo de cada sujeto 98

Figura 6-2. Comparación de la precisión obtenida para las dos primeras sesiones destinadas a la determinación del umbral óptimo de cada sujeto 98

Figura 6-3. Tipo de selecciones realizadas para cada sujeto durante las tres primeras tareas propuestas 99

Figura 6-4. Comparación de la precisión obtenida para las tres primeras tareas destinadas a evaluar el funcionamiento general de la aplicación 100

Figura 6-5. Comparación de la precisión obtenida para todas las sesiones, tanto la determinación del umbral (sesión 1: UmbralS1 y sesión 2: UmbralS2) como las tareas propuestas 101

ÍNDICE DE TABLAS

Capítulo 2: Sistemas Brain Computer Interface

TABLA 2-1. CLASIFICACIÓN DE LOS MÉTODOS MÁS EMPLEADOS EN LA EXTRACCIÓN DE CARACTERÍSTICAS.....	23
TABLA 2-2. CLASIFICACIÓN DE LOS MÉTODOS MÁS EMPLEADOS EN LA TRADUCCIÓN DE CARACTERÍSTICAS.....	25
TABLA 2-3. COMPARATIVA DE LOS DISTINTOS ARTÍCULOS SOBRE NAVEGADORES WEB ATENDIENDO A SU METODOLOGÍA Y FUNCIONALIDADES	41

Capítulo 3: Análisis de la Señal y Potenciales Evocados P300

TABLA 3-1. CARACTERÍSTICAS DE LOS USUARIOS INDIVIDUALES QUE MODIFICAN LA FORMA DEL P300.....	50
---	----

Capítulo 4: Diseño de una Aplicación de Navegación Web

TABLA 4-1. ESPECIFICACIONES DE LA PRIMERA SESIÓN, UTILIZADA PARA ENTRENAR EL CLASIFICADOR	79
TABLA 4-2. ESPECIFICACIONES DE LA SEGUNDA SESIÓN, UTILIZADA PARA HALLAR EL UMBRAL ÓPTIMO	81
TABLA 4-3. EDAD DE LOS CINCO SUJETOS MASCULINOS	82

Capítulo 5: Resultados

TABLA 5-1. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U1.....	83
TABLA 5-2. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U2.....	85
TABLA 5-3. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U3.....	86
TABLA 5-4. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U4.....	88
TABLA 5-5. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U5.....	89
TABLA 5-6. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U1.....	91
TABLA 5-7. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U2	92
TABLA 5-8. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U3.....	92
TABLA 5-8. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U4	93
TABLA 5-9. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U5	93
TABLA 5-10. RESULTADOS OBTENIDOS PARA EL CUESTIONARIO DE SATISFACCIÓN DE LOS USUARIOS .	94

1.1. Señales biomédicas

La ingeniería biomédica es un campo multidisciplinar que aplica las técnicas y métodos de la ingeniería a la comprensión, definición y resolución de problemas de las especialidades relacionadas con la medicina y la biología. Entre sus objetivos se encuentran el reducir la subjetividad de las medidas diagnósticas manuales, el mejorar la calidad de las señales registradas y el mejorar la precisión de la medida, entre otros. En consecuencia, se encarga de procesar las señales producidas por los sistemas biológicos para obtener información relevante sobre los mismos con fines diagnósticos. Esas señales se conocen con el nombre de señales biomédicas.

Las señales biomédicas pueden clasificarse atendiendo a distintos criterios: según su descripción matemática, su origen o sus características. Según su descripción matemática se clasifican como señales deterministas o señales aleatorias.

Las señales deterministas son aquellas que pueden representarse matemáticamente de forma explícita mediante una expresión cerrada, de manera que sus posibles valores futuros son predecibles conociendo poca información sobre su pasado. A su vez, las señales deterministas pueden subdividirse en señales periódicas, cuasiperiódicas y transitorias.

Las señales aleatorias no permiten conocer su valor futuro con exactitud ni siquiera conociendo todos sus valores pasados. Estas señales son muy abundantes en los procesos físicos, y en consecuencia el análisis de su estacionariedad y su ergodicidad cobra importancia. La estacionariedad y la ergodicidad son dos propiedades que pueden cumplir las señales aleatorias. La primera de ellas exige que el comportamiento estadístico de la señal no cambie a lo largo del tiempo, pudiendo distinguir entre distintos grados de estacionariedad: en sentido estricto y en sentido amplio. Con respecto a la segunda, se dice que una señal aleatoria es ergódica si toda su aleatoriedad está presente en cualquiera de sus realizaciones, y por tanto, consideramos que una realización es representativa de todas las demás [1].

Atendiendo al origen de las señales biomédicas, se pueden clasificar como señales eléctricas, impedancias, señales acústicas, señales mecánicas, señales biomagnéticas, señales bioquímicas o imágenes. Las señales más conocidas se encuentran dentro de la clasificación de señales eléctricas, las cuales recogen la actividad eléctrica de distintas partes del cuerpo humano, como son el electroencefalograma (EEG) y el electrocorticograma (ECoG), que recogen la actividad cerebral; el electrooculograma (EOG), que recoge la actividad ocular; el electromiograma (EMG), que recoge la actividad muscular; y el electrocardiograma (ECG), que recoge la actividad del corazón. En la Figura 1-1 se ilustran tres de estas señales: EEG, ECG, EMG [2].

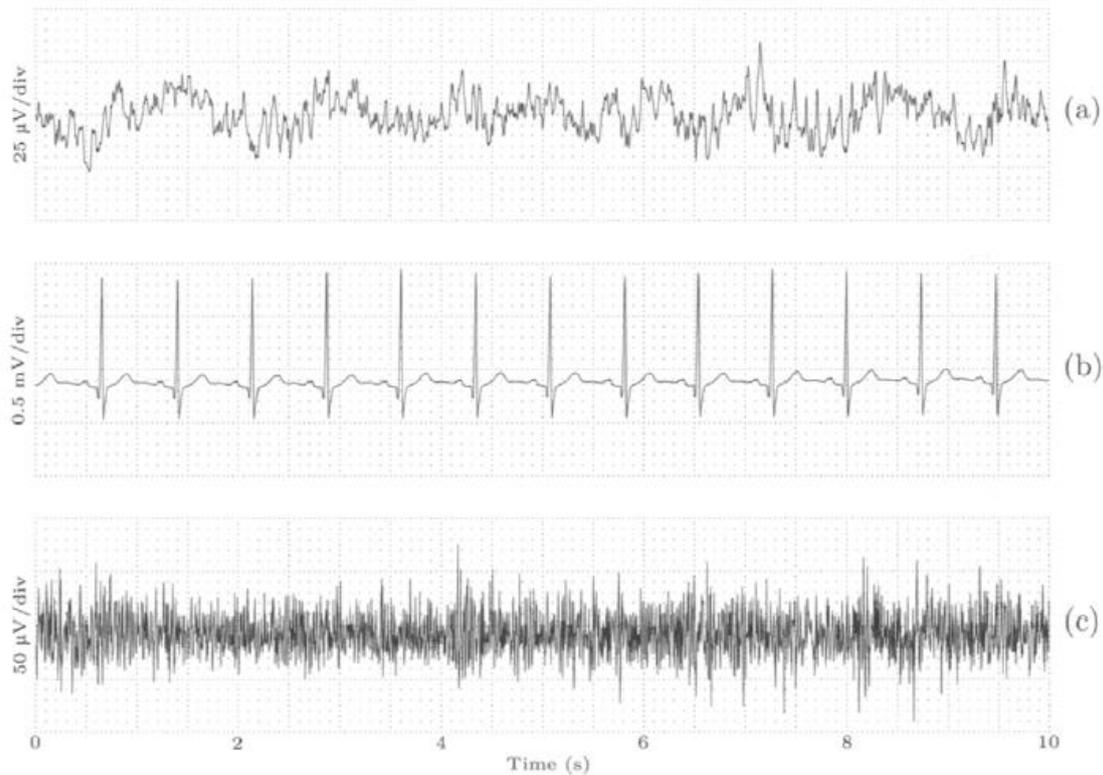


Figura 1-1. Ejemplos de las tres señales bioeléctricas principales recogidas sobre la superficie corporal: (a) un electroencefalograma (EEG), (b) un electrocardiograma (ECG), y (c) un electromiograma (EMG).

Dentro de las señales magnéticas destaca el magnetoencefalograma (MEG), que registra la actividad funcional del cerebro mediante la captación de campos magnéticos, permitiendo estudiar las relaciones entre las estructuras cerebrales y sus funciones. Otro tipo de señales son las señales bioquímicas, las cuales son el resultado de mediciones químicas de los tejidos vivos o de muestras analizadas en el laboratorio clínico.

Para finalizar, también destacan las imágenes biomédicas obtenidas mediante técnicas que emplean diversos principios físicos, tales como la imagen por resonancia magnética funcional (fMRI), que permite localizar regiones cerebrales específicas; la tomografía por emisión de positrones (PET), que mide la actividad metabólica del cuerpo humano; o la retinografía, que permite visualizar el estado de la retina para diagnosticar enfermedades como la retinopatía diabética. En la Figura 1-2 se muestra una imagen PET en el estudio de la enfermedad de Alzheimer [3].

Sin embargo, para cumplir con los objetivos principales de la ingeniería biomédica la señal tiene que sufrir una serie de transformaciones antes de poder ser analizada. Estas transformaciones se aplicarán, generalmente, en tres etapas: obtención y registro de las señales (muestreo, cuantificación y digitalización de la señal), eliminación de artefactos y, finalmente, procesado de la señal (segmentación y filtrado de la misma).

En el siguiente apartado nos centraremos en el EEG dado que es la señal utilizada en el presente estudio, introduciendo sus características principales, la forma correcta para registrar la señal y los artefactos que degradan la misma.

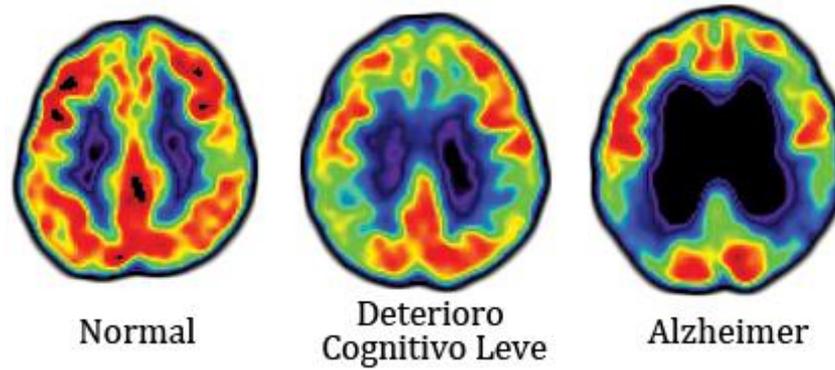


Figura 1-2. Imágenes PET que detectan la disminución de glucosa asociada con la enfermedad del Alzheimer. A la izquierda un sujeto sano, en el centro un sujeto con DCL (Deterioro Cognitivo Leve) y a la derecha un enfermo de Alzheimer.

1.2. Electroencefalograma (EEG)

El electroencefalograma (*electroencephalogram*, EEG) es una señal biomédica que registra la actividad bioeléctrica del cerebro de forma no invasiva mediante la implantación de una serie de electrodos sobre el cuero cabelludo. Debido a la atenuación que sufre la señal por las capas que separan el córtex y el electrodo (flúidos, huesos, y piel), es imposible medir la actividad de una única neurona. Consecuentemente, el EEG es la agrupación de la actividad de millones de neuronas al mismo tiempo. Esta señal se comporta de manera oscilatoria y repetitiva, y por lo tanto, nos solemos referir a esta actividad conjunta como *ritmo*. Los ritmos suelen clasificarse según su rango frecuencial y su amplitud relativa.

La amplitud de la señal EEG está relacionada con el grado de sincronía con el cual interactúan las neuronas. Una excitación sincronizada de un grupo de neuronas producirá una amplitud alta en la superficie del cráneo debido a la interferencia constructiva que se originará gracias a esa sincronización. Sin embargo, una excitación asíncrona de las neuronas producirá una señal EEG irregular con bajas amplitudes, debido a la interferencia destructiva de las actividades individuales. Este rango de amplitudes abarca desde unos pocos μV hasta $100\mu\text{V}$ [2].

El rango frecuencial de la señal EEG depende de la actividad que se esté realizando y suele estar relacionado con la amplitud de la misma. Un estado de somnolencia producirá bajas frecuencias y una sincronía entre las neuronas, causando una amplitud grande. El caso contrario se producirá cuando el cerebro está concentrado en alguna tarea determinada, desplazando el espectro a frecuencias más altas y generando una actividad asíncrona de las neuronas, forzando una amplitud más baja. En la Figura 1-3 se observan varios ritmos según el estado del sujeto [2]. El rango frecuencial comprende desde 0.5Hz (*banda delta*) hasta más de 30Hz (*banda gamma*).

La colocación de los electrodos debe atenderse a un estándar conocido como el *sistema internacional 10/20*, aprobado por la Federación Internacional de Sociedades de Electroencefalografía y Neurofisiología Clínica [4]. Este estándar distribuye los electrodos diferenciándolos según las distintas zonas del córtex y el hemisfero al que pertenecen. El cráneo se divide en seis zonas: frontopolar (Fp), frontal (F), central (C), parietal (P), occipital (O) y temporal (T). Tomando como medida de referencia la distancia recorrida entre el entrecejo (nación) y la nuca (inión) del sujeto, la sección central se encontrará a mitad de distancia y se separará de las secciones contiguas (frontal y parietal) un 20% de la distancia total. A su vez, estas dos secciones estarán

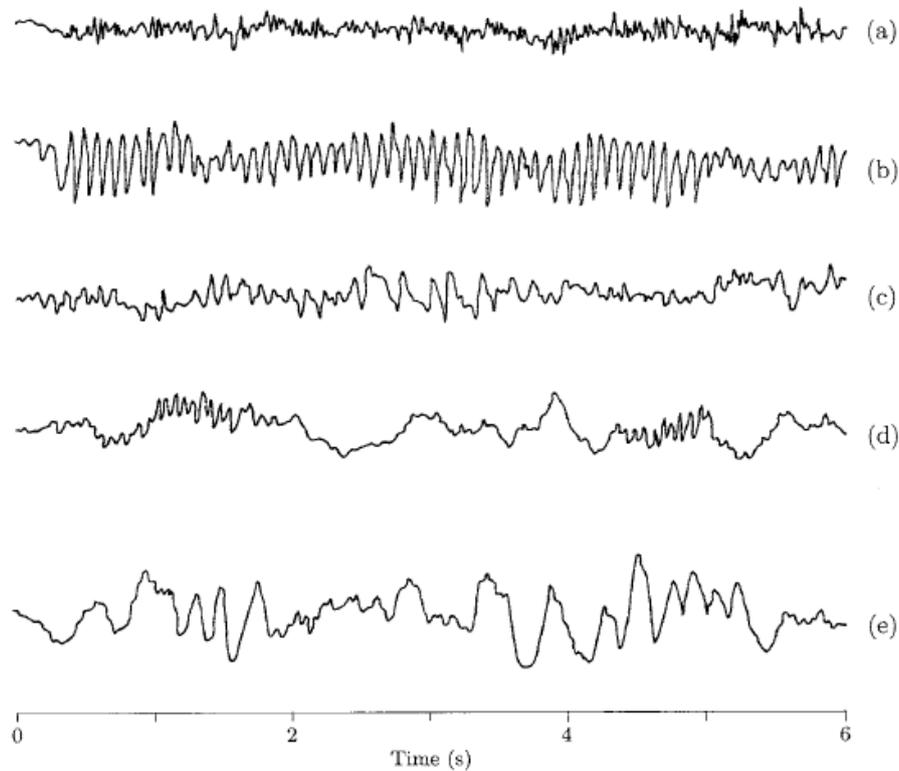


Figura 1-3. Ritmos electroencefalográficos observados durante el paso de vigilia a sueño profundo: (a) agitado, (b) relajado, (c) adormecido, (d) dormido, (e) sueño profundo. Se aprecia una disminución frecuencial a medida que el sujeto se duerme.

separadas de las secciones frontopolar y occipital, respectivamente, por un 10% de la longitud total.

La nomenclatura de los electrodos está compuesta de la letra que identifica la sección junto a un subíndice que depende del hemisferio en el que se encuentren: para el hemisferio izquierdo se asignan números impares y para el hemisferio derecho se asignan números pares. En el caso de encontrarse en el centro de ambos hemisferios, se identifican mediante una letra “z” (*zero*) minúscula en lugar de un número. Con respecto a los electrodos de referencia, suelen ubicarse en los lóbulos de las orejas y se denominan A1 (izquierdo) y A2 (derecho). En la Figura 1-4 aparecen las distribuciones de los electrodos recomendadas según el sistema internacional 10/20.

Dado que muchas componentes de ruido, incluso las que son externas al cuerpo humano, aparecen de manera común en todas las señales EEG, hay que aplicar un filtro espacial antes de procesar la señal. La referencia en la oreja elimina muchas de estas componentes, no obstante, adicionalmente se utilizan canales bipolares, métodos de media común o filtros Laplacianos. Los canales bipolares toman la diferencia de potencial entre dos electrodos no muy alejados. El método de referencia de media común (*Common Average Reference, CAR*) resta a la señal recogida por un electrodo la media de todo el conjunto de electrodos. Finalmente, el filtro Laplaciano resta a la señal recogida por un electrodo las señales adyacentes.

Además de las componentes de ruido externas al cuerpo humano, la señal puede verse afectada por una serie de artefactos o señales diferentes a la de interés. En el EEG los artefactos más conflictivos son el movimiento ocular y el movimiento muscular [5].

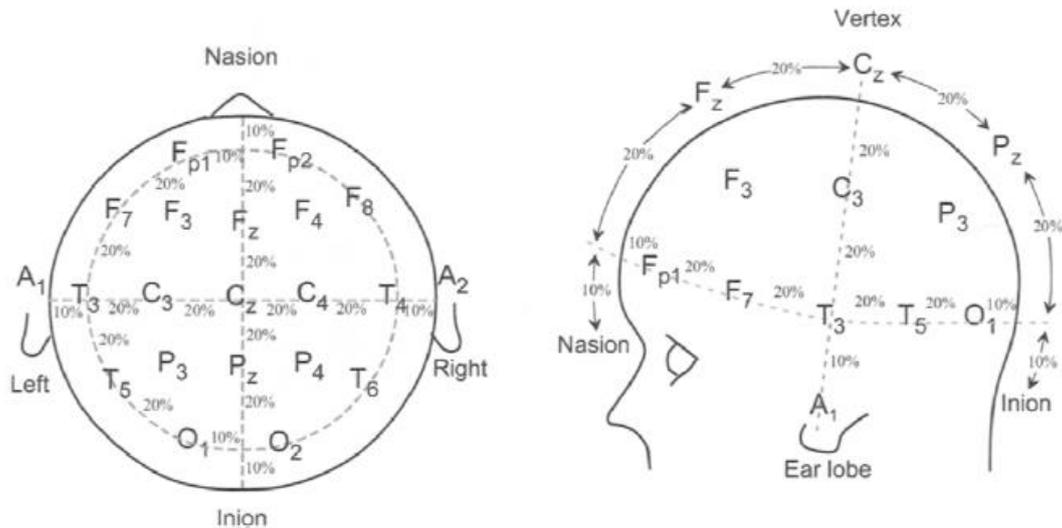


Figura 1-4. Distribución de los electrodos según el sistema internacional 10/20.

1.3. Brain Computer Interface (BCI)

Los sistemas Brain Computer Interface (*Interfaz Cerebro-Computadora*, BCI) establecen un sistema de comunicación entre el cerebro y el medio sin la intervención de los mecanismos normales de los nervios y músculos periféricos. Generalmente, los sistemas BCI procesan el EEG del usuario de manera no invasiva y traducen las intenciones del mismo en comandos de un dispositivo [6]. No obstante, también existen sistemas BCI que utilizan métodos invasivos como el ECoG o los electrodos epidurales o intracorticales como métodos de registro de la actividad cerebral. Estos sistemas consiguen mayor resolución a cambio de comprometer su estabilidad e implicar un riesgo.

Para identificar y traducir las intenciones del usuario en comandos de un determinado dispositivo, BCI utiliza señales de control tales como los potenciales evocados visuales (VEP), los potenciales evocados P300, los potenciales corticales lentos (SCP) o los ritmos sensoriomotores μ y β . Estas señales de control pueden pertenecer a dos grupos: sistemas endógenos y sistemas exógenos.

Los sistemas BCI endógenos no necesitan de ningún tipo de estimulación externa para general la actividad cerebral necesaria para clasificar las intenciones del usuario. Estos sistemas requieren entrenamiento para poder controlar satisfactoriamente la aplicación BCI. Habitualmente utilizan potenciales SCP y ritmos sensoriomotores μ y β .

En contrapartida, los sistemas BCI exógenos requieren una estimulación externa para producir la actividad cerebral necesaria para clasificar las intenciones del usuario. Estos sistemas no requieren entrenamiento alguno puesto que utilizan la respuesta natural del cerebro a diversos estímulos externos. Habitualmente utilizan potenciales VEP y potenciales evocados P300.

La señal debe ser tratada antes de poder traducir las intenciones del usuario. Este tratamiento está compuesto por dos etapas principales: adquisición de la señal y procesamiento de la señal. La primera etapa, adquisición de la señal, trata de amplificar y digitalizar la señal obtenida, además de eliminar aquellos artefactos que se han acoplado a la misma.

El procesado de la señal se compone de dos sub-etapas: la extracción y la clasificación de características. La extracción de características utiliza una serie de combinaciones y transformaciones sobre la señal EEG con el objetivo de obtener una información más discriminadora sobre las intenciones del usuario. La última parte, clasificación de características, transforma las características extraídas de la señal EEG en señales de control de un dispositivo o de una aplicación. El procesado de la señal se tratará con más detalle en el segundo y tercer capítulo.

El desarrollo de los sistemas BCI tiene como objetivo el aumentar la independencia y la capacidad de comunicación en usuarios con enfermedades neurodegenerativas o graves discapacidades. Los sistemas BCI pueden servir de gran ayuda para personas afectadas con afasia, apraxia, autismo, parálisis, lesión medular, amputaciones, distrofias neuromusculares o esclerosis lateral amiotrófica, entre otras enfermedades. En consecuencia, las aplicaciones BCI abarcan desde el control de prótesis hasta el control de un navegador web, pasando por ayudas para la comunicación, control domótico o entretenimiento en videojuegos.

1.4. Objetivos del Trabajo de Fin de Grado

El objetivo general del presente trabajo de fin de grado es el desarrollo de una aplicación mediante sistemas BCI que permita navegar por Internet utilizando únicamente las ondas cerebrales. Esta aplicación se dirige a personas con un elevado grado de discapacidad y utiliza el EEG, un método no invasivo y de bajo coste, para traducir las intenciones del usuario en comandos de la aplicación del navegador web. Para alcanzar este objetivo general, se deben cumplir los siguientes objetivos específicos:

1. Estudiar las distintas técnicas para registrar la actividad cerebral, los sistemas BCI y las señales de control que se utilizan normalmente para traducir las intenciones del usuario en comandos de un dispositivo.
2. Realizar una revisión del estado del arte de los navegadores web mediante BCI desarrollados con anterioridad junto con una comparativa que refleje las funcionalidades y las precisiones de cada uno de ellos.
3. Establecer las funcionalidades que debe implementar la aplicación en base a la comparativa realizada.
4. Diseñar las matrices que generan los potenciales evocados P300 y desarrollar la parte de la aplicación que adquiere la señal EEG, la procesa y envía los comandos necesarios a la segunda parte de la misma; mediante el lenguaje C++.
5. Desarrollar la última parte de la aplicación, la cual interpreta los comandos y refleja las intenciones del usuario como una extensión del navegador *Google Chrome*; mediante el lenguaje JavaScript. En la Figura 1-5 aparece el icono de la misma.
6. Diseñar un umbral para cada usuario que permita discriminar entre ausencia o no de atención de los estímulos utilizados para generar los potenciales evocados P300.
7. Evaluar el clasificador, el umbral óptimo y las funcionalidades de la aplicación por parte de cinco sujetos sanos en tres sesiones distintas.
8. Discutir los resultados obtenidos y compararlos con otros estudios.
9. Extraer conclusiones a partir de los resultados obtenidos.



Figura 1-5. Icono diseñado para la aplicación web.

1.5. Estructura del Trabajo de Fin de Grado

El trabajo de fin de grado está compuesto por siete capítulos: introducción, sistemas Brain Computer Interface, análisis de la señal y potenciales evocados P300, diseño de una aplicación de navegador web, resultados, discusión y conclusiones y líneas futuras.

En el primer capítulo, introducción, se describen los distintos tipos de señales biomédicas y su clasificación atendiendo a distintas características de la señal. Posteriormente, se detallan las características de la señal EEG y la adquisición de la señal mediante el estándar internacional 10/20. A continuación, se introduce el concepto de sistema BCI, se enumeran las distintas señales de control ampliamente utilizadas y se establece, brevemente, la estructura del procesamiento de la señal. Finalmente, se detallan los objetivos y la estructura del presente trabajo.

En el segundo capítulo, sistemas Brain Computer Interface, se tratan, de manera más detallada, los aspectos sobre BCI previamente introducidos en el primer capítulo. En primer lugar, se estudian los distintos métodos para registrar la actividad cerebral, seguido por la enumeración y explicación de las distintas señales de control. Posteriormente se detalla el procesamiento de la señal EEG y se enumeran las distintas aplicaciones que han surgido gracias a los sistemas BCI, junto a una comparativa entre los distintos navegadores web desarrollados con anterioridad. En último lugar, se comentan las limitaciones actuales de dichos sistemas.

En el tercer capítulo, análisis de la señal y potenciales evocados P300, se especifican las propiedades de los potenciales evocados P300 (señal de control utilizada en la aplicación web). Posteriormente, se introducen los distintos métodos utilizados para procesar la señal en la aplicación desarrollada, tanto para la extracción de características como para la clasificación de las mismas. Finalmente, se incluye un apartado de umbralización, donde se estudia cómo diferenciar la actividad basal del usuario con respecto al deseo de seleccionar un comando en la aplicación del navegador web.

En el cuarto capítulo, diseño de una aplicación de navegador web, se describen los objetivos, la estructura y el desarrollo de la aplicación web desarrollada. Además de incluir un apartado sobre los sujetos que han evaluado la aplicación y las tareas que debieron realizar.

En los siguientes capítulos se muestran, analizan y discuten los resultados obtenidos, además de compararlos con los estudios presentados anteriormente, y se evalúa el funcionamiento de la aplicación en general. Para terminar, se incluye un capítulo sobre conclusiones y líneas futuras de investigación a partir del presente trabajo.

SISTEMAS BRAIN COMPUTER INTERFACE

2.1. Introducción

La historia de los sistemas BCI está íntimamente ligada con el desarrollo de los electroencefalogramas. El EEG fue desarrollado por Hans Berger en 1929 y las ondas cerebrales, hasta entonces desconocidas, fueron investigadas durante el siglo XX, aplicando este sistema como diagnóstico de la epilepsia o diversos trastornos del sueño. Durante este periodo de tiempo se especuló sobre utilizar las señales EEG para desarrollar un sistema de comunicación entre el cerebro y el medio sin la intervención de los mecanismos normales de los nervios y músculos periféricos [6].

El primer sistema BCI surgió en 1977 a cargo del Dr. Jacques Vidal, permitiendo controlar el movimiento de un cursor en dos dimensiones mediante los potenciales evocados visuales (VEP) que se detectaban en el EEG del usuario. El sistema se componía de cuatro botones que se iluminaban con luces intermitentes de xenón a distintas frecuencias, los cuales representaban las cuatro direcciones básicas: arriba, abajo, derecha e izquierda. Cuando la mirada del usuario se centraba en uno de estos botones quedaban reflejados en el EEG los potenciales VEP que aparecen como respuesta a un estímulo visual de manera natural. De esta manera, el usuario era capaz de manejar, rígidamente, un cursor con el objetivo de escapar de un laberinto, por ejemplo [7]. El esquema de éste primitivo e innovador sistema BCI aparece en la Figura 2-1.

No obstante, aunque éste sistema fue pionero en el ámbito de la interfaz cerebro-computadora, los sistemas BCI no se han desarrollado completamente hasta la década de los 90 debido al coste computacional que requieren las aplicaciones prácticas.

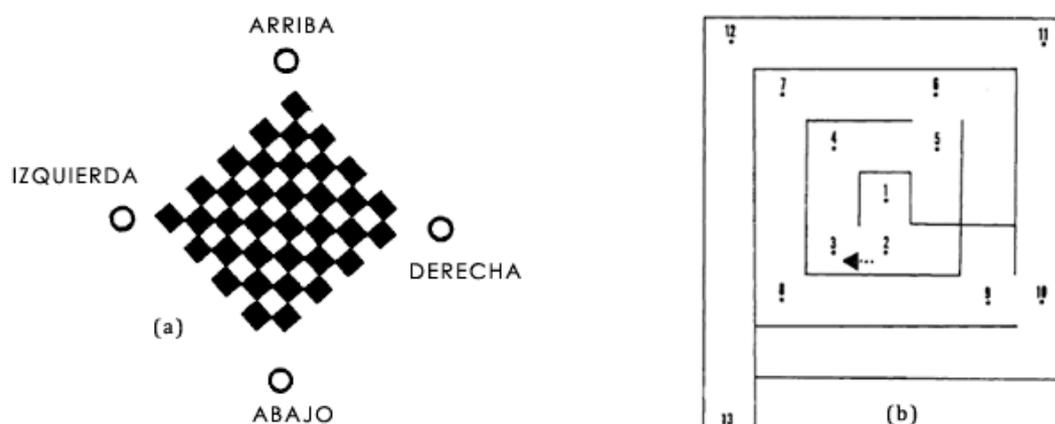


Figura 2-1. Esquema del primer sistema BCI controlado por VEP y desarrollado por Dr. Jacques Vidal en 1977. En (a) aparecen los botones que se iluminaban intermitentemente para controlar el cursor. En (b) aparece el laberinto del cual debían salir los usuarios moviendo el cursor [7].

Tal y como se ha explicado anteriormente, un sistema BCI puede clasificarse atendiendo a varios aspectos. Podemos hablar de sistemas BCI endógenos y exógenos. Los sistemas BCI endógenos no necesitan de ningún tipo de estimulación externa para generar la actividad cerebral necesaria para clasificar las intenciones del usuario y, por tanto, requieren entrenamiento. En contrapartida, los sistemas BCI exógenos sí que requieren de una estimulación externa para producir esa actividad cerebral. Dado que la respuesta del cerebro es natural, no necesitan entrenamiento.

Atendiendo a los métodos que se utilizan para registrar la actividad cerebral, un sistema BCI se puede clasificar en dos grupos: sistemas invasivos y sistemas no invasivos. Los sistemas invasivos, menos utilizados en la práctica con seres humanos, necesitan una intervención previa para implantar los electrodos que recogen la actividad bioeléctrica del cerebro. Consecuentemente, estos sistemas implican un riesgo para la salud a cambio de obtener mayor calidad en las señales registradas. Los sistemas BCI invasivos suelen utilizar el ECoG como método de adquisición de la señal. En cambio, los sistemas no invasivos no requieren de ningún tipo de implantación previa para recoger la actividad cerebral, convirtiéndose así en los sistemas más utilizados en la práctica. Estos sistemas suelen utilizar el EEG como método de adquisición de la señal.

La finalidad principal de los sistemas BCI es la de mejorar la calidad de vida de personas con discapacidades severas, ya sea por sufrir una enfermedad neurodegenerativa (como la esclerosis lateral amiotrófica o distrofia neuromuscular), o por sufrir algún tipo de parálisis o amputación (como las lesiones cerebrales, medulares o la apoplejía), ayudando en la comunicación e incluso en el control de dispositivos domóticos.

La estructura de los sistemas BCI comprende dos etapas bien diferenciadas: la adquisición de la señal y el procesamiento de la misma. En este capítulo se sigue la misma estructura, indicando cuáles son los principales métodos de adquisición de la señal y las distintas señales de control que se utilizan, detallando sus características y la zona del córtex donde se producen.

A continuación se detallan los artefactos más comunes que afectan a la señal, entendiendo como artefacto cualquier otra señal recogida diferente a la de interés, siendo éstos el movimiento ocular y pestaños, la actividad eléctrica de músculos cercanos y la actividad eléctrica del corazón.

Una vez digitalizada la señal y libre de artefactos, se trata el procesado de la misma, distinguiendo dos sub-etapas: la extracción de características y el algoritmo de traducción. En la extracción de características 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. A continuación, el algoritmo de traducción (también conocido con el nombre de traducción de características) se encarga de transformar las características extraídas en señales de control de un dispositivo o de una aplicación [5]. La estructura habitual de los sistemas BCI se puede observar detenidamente en la Figura 2-2.

Posteriormente se enumeran las principales aplicaciones de los sistemas BCI tales como ayudas en la comunicación, selección de letras, movimiento de un cursor, desplazamiento sobre un mapa, etcétera; haciendo hincapié en las aplicaciones que se encargan de controlar un navegador web. Seguidamente se realiza una presentación sobre el estado del arte de los navegadores basados en BCI desarrollados con anterioridad, además de incluir una comparación entre cinco soluciones distintas.

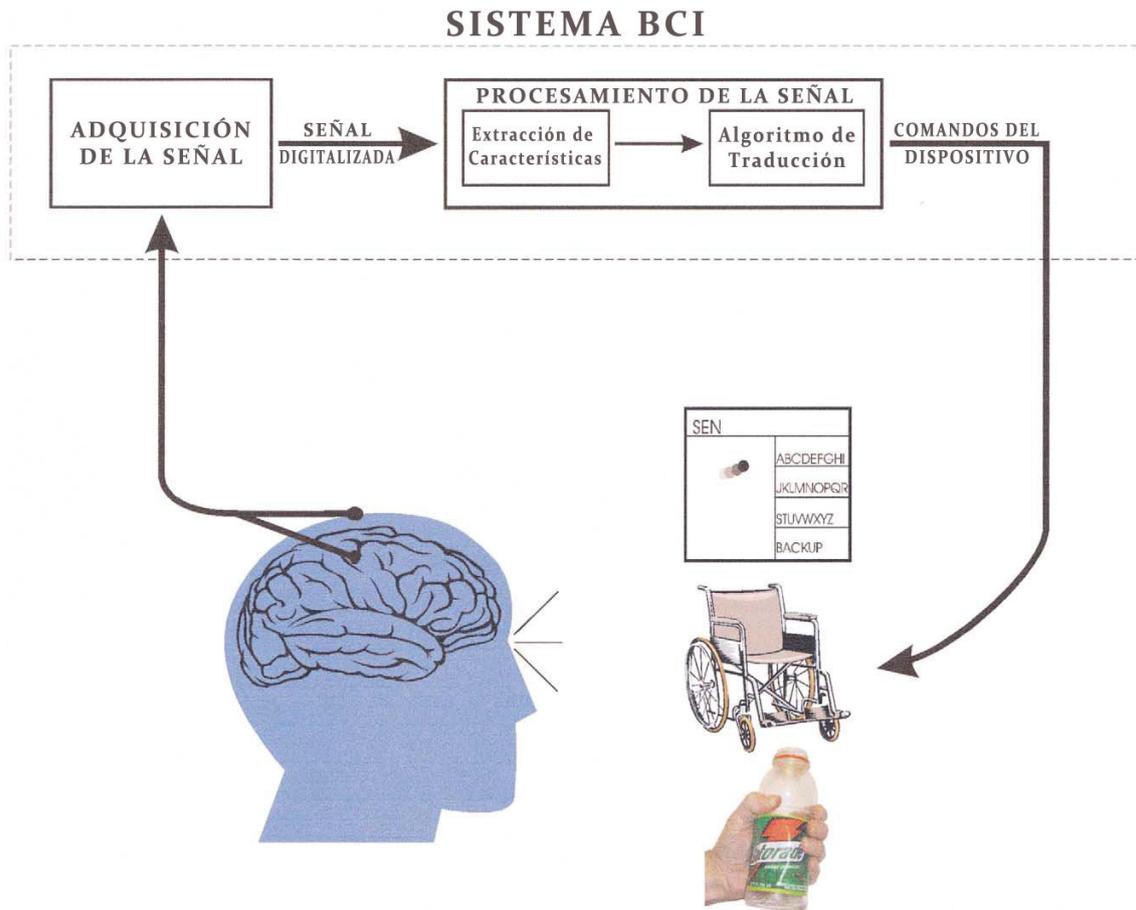


Figura 2-2. Estructura habitual que siguen los sistemas BCI desde la adquisición de la señal cruda hasta la conversión de ésta en comandos de control de un dispositivo.

2.2. Métodos para registrar la actividad cerebral

Existen una gran variedad de métodos de adquisición de las señales que se utilizan para controlar los sistemas BCI. Estos métodos se pueden clasificar en métodos que extraen señales eléctricas, ampliamente utilizados, y métodos que no utilizan señales eléctricas, detallados a continuación.

- **Imágenes de resonancia magnética funcional (*functional Magnetic Resonance Imaging, fMRI*).** La fMRI hace uso de los fenómenos de resonancia magnética para medir la respuesta hemodinámica del cerebro con fines clínicos y de investigación. La principal utilidad de esta técnica es la de proveer información sobre la localización de las funciones cerebrales mientras se realiza alguna acción. Normalmente se utiliza para pacientes que requieren cirugía cerebral.
- **Magnetoencefalografía (*Magnetoencephalography, MEG*).** La técnica MEG se define como un método no invasivo que registra la actividad funcional del cerebro mediante la captación de campos magnéticos, permitiendo investigar las relaciones entre las estructuras cerebrales y sus funciones.
- **Espectroscopía de infrarrojo cercano (*Near Infrared Spectroscopy, NIRS*).** La espectroscopía estudia la interacción de la radiación electromagnética con la materia. En este caso se emiten fotones de luz cercanos al infrarrojo (NIR, longitudes de onda entre 780 nm y 3.000 nm)

sobre la piel de la frente del paciente. Después de dispersarse por el interior del cuero cabelludo, cráneo y cerebro, parte de esos fotones vuelven a la piel gracias a los fenómenos de reflectancia espectral. Al medir la cantidad de fotones recibidos se extraen conclusiones sobre la respuesta hemodinámica del cerebro y los niveles de saturación de oxígeno y azúcar en sangre.

- **Tomografía por emisión de positrones (*Positron Emission Tomography, PET*)**. La técnica PET se basa en detectar y analizar la distribución tridimensional que adopta en el interior del cuerpo un radiofármaco administrado a través de una inyección intravenosa. Esta técnica se clasifica como una técnica no invasiva de diagnóstico *in vivo* por imagen capaz de medir la actividad metabólica del cuerpo humano.

Aunque todas las técnicas mencionadas son métodos no invasivos, no están exentas de inconvenientes. La resolución espacial que proveen las técnicas fMRI y NIRS es excelente, sin embargo, la resolución temporal es muy pobre, causando que no se puedan utilizar en aplicaciones que requieren funcionar en tiempo real. No ocurre lo mismo para las técnicas MEG, las cuales consiguen muy buenas resoluciones tanto temporales como espaciales, sin embargo, dado que ésta técnica y fMRI requieren un superconductor, el equipamiento necesario para llevarlas a cabo tiene un coste demasiado elevado. Adicionalmente, también necesitan aislamiento magnético, provocando el uso de habitaciones aisladas y equipamientos demasiado voluminosos [8].

Teniendo en cuenta que la aplicación a desarrollar, y en general todas las aplicaciones que utilizan BCI, requiere un sistema portátil, eficiente, de coste razonable y a tiempo real, las técnicas explicadas previamente no se adecúan a los requerimientos buscados. Es por esta razón por la cual los métodos que extraen señales eléctricas son los más utilizados actualmente, detallados a continuación y representados esquemáticamente en la Figura 2-3.

- **Electroencefalografía (*Electroencephalography, EEG*)**. La técnica EEG es un método no invasivo que registra la actividad bioeléctrica del cerebro mediante la implantación de una serie de electrodos sobre el cuero cabelludo. Normalmente, en los sistemas BCI se utilizan 64 electrodos, no obstante, para aplicaciones concretas se suelen aplicar menos, dependiendo de la señal de control que se desee analizar.
- **Electrocorticografía (*Electrocorticography, ECoG*)**. La técnica ECoG persigue el mismo objetivo que la técnica EEG, sin embargo, ésta sí es una técnica invasiva. Se basa en implantar electrodos directamente sobre la superficie del córtex, proveyendo mayor resolución temporal y espectral a cambio de implicar un riesgo para el paciente.
- **Electrodos epidurales o intracorticales**. Los electrodos epidurales se basan en la colocación de electrodos de cristal con forma de cono hueco en neuronas aisladas. Estos electrodos ofrecen mayor resolución espacial y temporal que las técnicas anteriores, sin embargo, el tiempo que pueden permanecer implantados es limitado, además de que la neurona en cuestión podría morir u otra neurona podría pasar a realizar sus funciones, haciendo inservible la obtención de la señal a través de éste método.

Cada alternativa posee ventajas y desventajas. El EEG es una técnica frecuentemente utilizada debido a su naturaleza no invasiva. No obstante, ofrece una resolución temporal y espacial limitada, además de estar contaminado por artefactos

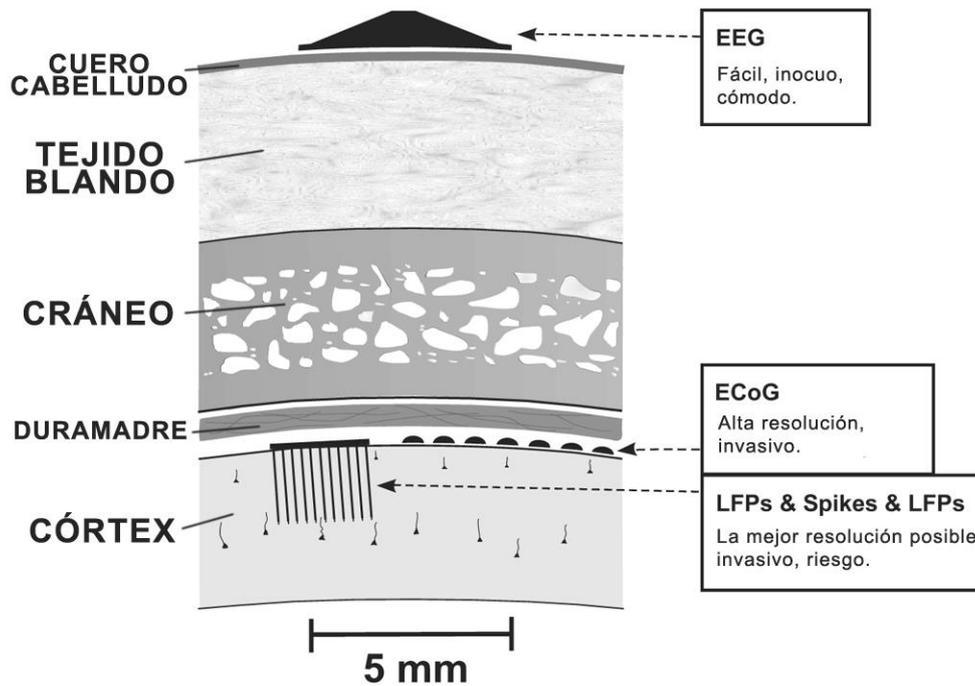


Figura 2-3. Localización de los electrodos de distintas técnicas que emplean señales eléctricas.

tales como el movimiento de los ojos o de los músculos cercanos. También requiere un mantenimiento continuo para asegurar una baja impedancia en los electrodos y así obtener una señal con calidad.

A diferencia del EEG, el ECoG ofrece mayor resolución espacial y está libre de artefactos. En contrapartida, este método requiere una intervención quirúrgica y en la práctica solamente lo usan las personas que ya poseen estos electrodos porque sufren de epilepsia. Para finalizar, los electrodos epidurales o intracorticales ofrecen las mayores resoluciones posibles pero, debido a los inconvenientes ya mencionados, solamente se usan en experimentos con animales [8].

Parece razonable considerar el EEG como el método más indicado para la aplicación a desarrollar. El EEG es un método no invasivo, portátil, fácil de usar y con un precio razonable, lo cual lo convierte en el mejor candidato posible. Después de obtener la señal a través de los electrodos del cuero cabelludo, se procede a digitalizarla y eliminar los artefactos que pueden deteriorarla.

2.3. Tipos de señales de control en BCI

Una vez seleccionado el EEG como método para registrar la actividad cerebral en la aplicación BCI a desarrollar, es necesario elegir una señal de control específica que permita gobernar la aplicación. Estas señales se pueden generar de manera natural como respuesta a un estímulo externo (señales de control exógenas), o se pueden generar de manera voluntaria con entrenamiento (señales de control endógenas).

Para registrar un EEG en los sistemas BCI habitualmente se utilizan 64 electrodos, sin embargo, para aplicaciones que usan señales de control concretas se puede reducir este número conociendo la localización específica donde se generan. Para comprender la localización de los electrodos a la hora de utilizar una aplicación BCI se incluye a continuación un gráfico donde se indican las distintas regiones en las que se divide la corteza cerebral (Figura 2-4).

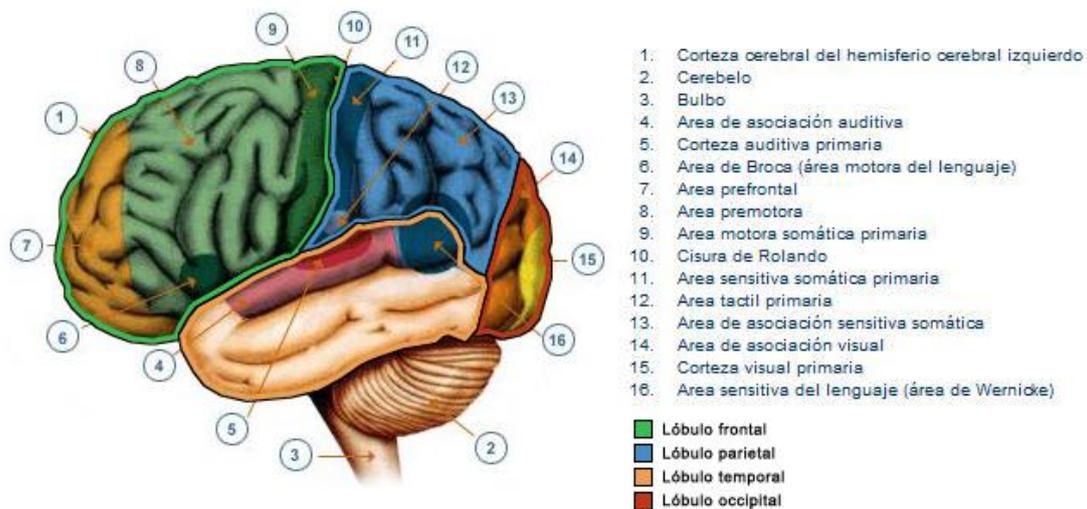


Figura 2-4. Regiones en las que se divide la corteza cerebral.

Las señales de control que trataremos a continuación son las siguientes: potenciales VEP, potenciales corticales lentos SCP, ritmos sensoriomotores μ y β , potenciales de neuronas corticales y potenciales evocados P300.

2.3.1. Potenciales Evocados Visuales

Los potenciales evocados visuales (*Visual Evoked Potentials*, VEP) son cambios de potencial que se producen en la corteza visual tras una estimulación luminosa [5]. Estos potenciales VEP producen un fenómeno de resonancia cuando el sujeto se fija en una fuente luminosa que parpadea a una determinada frecuencia superior a 4Hz. Este fenómeno de resonancia se utiliza para determinar la dirección del ojo [9]. Se recogen en la parte visual del córtex, ubicada en torno al lóbulo parietal y occipital.

Un método muy utilizado para determinar la dirección de la mirada consiste en la presentación al usuario de una serie de fuentes de luz que parpadean a distintas frecuencias. Cuando el usuario se fija en una fuente concreta, el sistema mide la frecuencia con la que aparecen los potenciales VEP (en torno a 100ms después de la iluminación [5]) y se genera el fenómeno de resonancia. Este fenómeno provoca que en el EEG del usuario aparezca un pico en la frecuencia que corresponde a la fuente de luz en la que se fija el usuario, además de sus armónicos. Distinguiendo este pico frecuencial se determina la dirección del ojo y, por tanto, las intenciones del usuario [9]. En la Figura 2-5 aparece el espectro frecuencial de un usuario cuando experimenta este método.

Esta señal de control es exógena, provocada por una respuesta natural del cerebro, y por tanto no requiere entrenamiento. Gracias a ello, se consiguen precisiones de hasta el 90% en pocas sesiones [10]. Además, Friman *et al* [9] demostró que el número de electrodos afecta en la precisión del sistema, concluyendo que los mejores resultados se obtienen cuando se utilizan pocos electrodos colocados correctamente en sobre la zona donde se generan los potenciales VEP. Las tasas binarias obtenidas se encuentran en torno a los 27 bits por minuto [11].

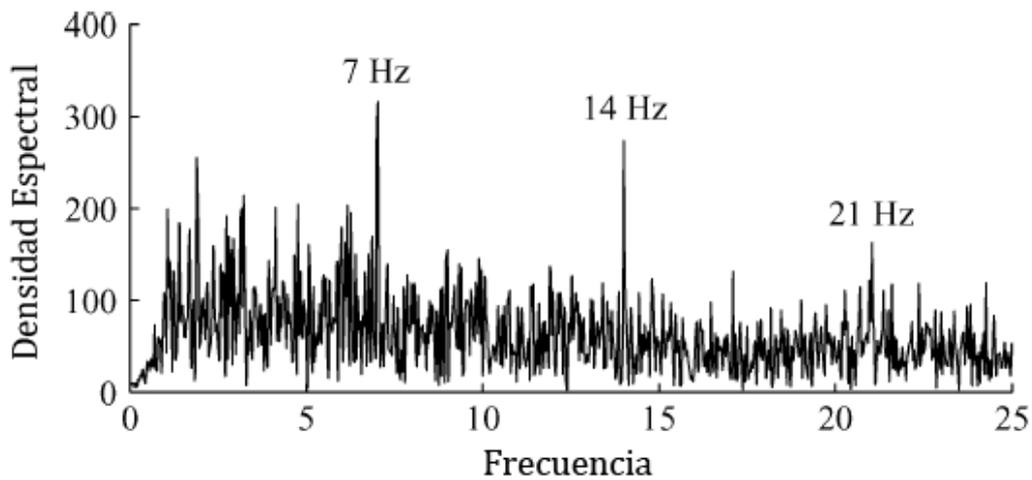


Figura 2-5. Espectro frecuencial de una señal EEG recogida durante una estimulación visual a 7Hz. El fenómeno de resonancia de los potenciales VEP produce picos en 7Hz y sus armónicos.

Los sistemas BCI que se basan en los potenciales VEP se clasifican dentro de los sistemas BCI dependientes, puesto que para poder realizar con éxito el control mediante estas señales se requiere que el usuario sea capaz de mantener la mirada fija en un punto concreto. El rango de aplicaciones es inmenso, pasando desde la selección de diversos comandos al control de un cursor en dos dimensiones.

2.3.2. Potenciales Corticales Lentos

Los potenciales corticales lentos (*Slow Cortical Potentials*, SCP) son cambios de voltaje lentos, entre 0.5 y 10 segundos, generados en el córtex a bajas frecuencias. Los valores SCP negativos están asociados al movimiento y otras funciones que involucran un aumento de la activación cortical, mientras que valores SCP positivos están asociados a la ausencia de movimiento o reducción de la activación cortical [5]. En la Figura 2-6 se pueden observar las diferencias entre el aumento y reducción de la activación cortical. Estos potenciales se generan en el vértex, la zona superior de la cabeza, en torno a la unión del lóbulo frontal y parietal, asociada con el movimiento.

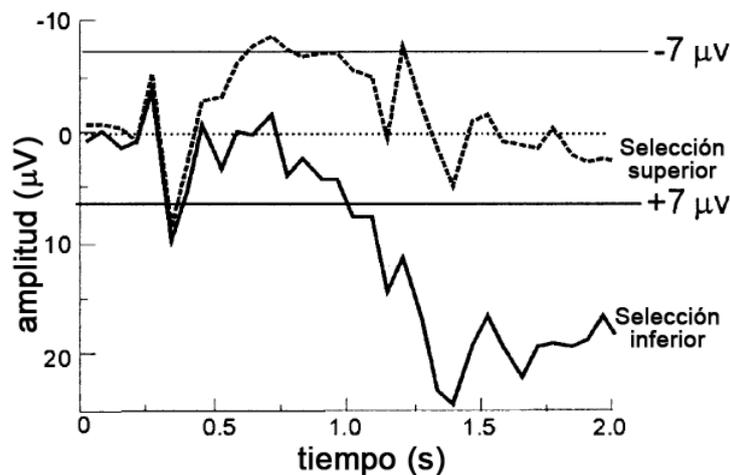


Figura 2-6. SCP recogidos durante una selección binaria [5].

Esta señal es endógena y requiere de entrenamiento para poder controlarse correctamente. En el estudio de Birbaumer *et al* [12] se desarrolla la técnica conocida como “dispositivo de interpretación de pensamiento” (*Thought Translation Device*, TTD). Para empezar se mide durante 2 segundos el voltaje inicial del EEG del usuario y durante los 2 segundos siguientes se determina si se ha producido o no activación cortical. La técnica consiste en un cursor que se mueve horizontalmente a velocidad constante, mientras que el movimiento vertical se controla a través de los SCP. Por encima del movimiento del cursor aparecen una serie de selecciones, y cuando el usuario desea seleccionar alguna de ellas basta con generar un SCP que produzca que el cursor se desplace verticalmente sobre ella. Es, por tanto, un sistema de selección binaria.

El entrenamiento para controlar adecuadamente estas señales de control varía entre 1 y 5 meses (obteniendo precisiones en torno al 75%), constituyendo la principal desventaja de esta señal. Las tasas binarias conseguidas se sitúan en torno a los 15 bits por minuto, inferiores a las obtenidas con los potenciales VEP [13].

Los artefactos que más afectan a los potenciales SCP son la respiración del sujeto y, sobre todo, el movimiento ocular. En consecuencia, deberán ser eliminados en la etapa de adquisición de la señal para poder analizarse adecuadamente.

2.3.3. Ritmos sensoriomotores (ritmos μ y β)

La señal EEG es el resultado de la agrupación de la actividad conjunta de millones de neuronas, lo cual produce un comportamiento oscilatorio y repetitivo, y por consiguiente, nos solemos referir a esa actividad conjunta como *ritmo*. En estado de vigila la mayoría de personas muestran una actividad en la banda de 8-12Hz cuando no están realizando ninguna acción motora o procesando algún estímulo externo (este fenómeno se puede apreciar en la Figura 2-7).

Esta actividad se denomina *ritmo μ* cuando se produce sobre la zona somatosensorial o motora del córtex y *ritmo α* cuando se produce sobre la zona visual del córtex. Los electrodos, por tanto, que son capaces de obtener la señal se deben colocar en estas zonas, ocupando las posiciones Cz, C3 y C4; y en ocasiones también F3, F4, O1 y O2.

Los *ritmos μ* (de 8 a 12 Hz) están asociados con otros ritmos de mayor frecuencia llamados *ritmos β* (de 18 a 26 Hz) y, aunque alguno de los *ritmos β* se correspondan con los armónicos de los *ritmos μ* , otros son independientes, pudiendo usar ambos como dos características distintas o dos señales de control autónomas [5].

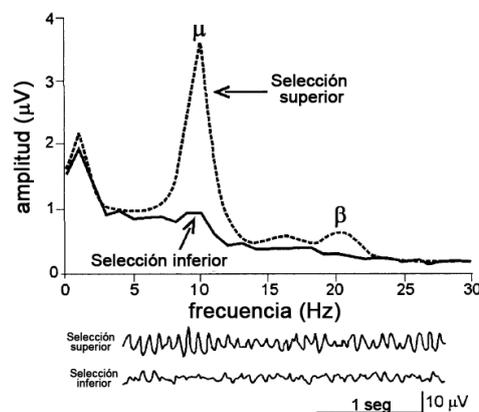


Figura 2-7. Ritmos μ y β . En la selección inferior se percibe una disminución de los mismos [5].

La preparación del movimiento o la intención de realizarlo conllevan una disminución de los ritmos μ y β , particularmente contralateral al movimiento. Esta disminución se denomina ERD (*Event-Related Desynchronization*, Desincronización del evento relacionado) y ocurre pocos segundos antes de realizar el movimiento. Se conoce como “desincronización” porque la señal pierde su carácter oscilatorio, provocando una disminución de los picos frecuenciales que la caracterizaban.

En contraposición, una vez se realiza o se finaliza la intención de realizar un movimiento, se produce un aumento de los ritmos μ y β . Este aumento se denomina ERS (*Event-Related Synchronization*, Sincronización del evento relacionado) y determina el punto en el cual la señal volverá a obtener su carácter oscilatorio, provocando la recuperación de sus picos frecuenciales característicos. Adicionalmente, tanto los ERD como los ERS se localizan en el lado del cerebro contrario al movimiento que se desea realizar [5].

En la Figura 2-8 se recogen varios registros que muestran estos fenómenos [14]. En la gráfica superior se muestran tres tipos de oscilaciones a distintas frecuencias recogidas por el mismo electrodo (C3) sobre la zona sensoriomotora cuando el usuario mueve el dedo índice de la mano. Se puede apreciar un claro ERD en el ritmo μ que aparece, aproximadamente, 2.5s antes de la realización del movimiento (marcado en línea roja, correspondiente a $t=0$ s) y recupera el nivel inicial pasados unos segundos. La actividad del ritmo β presenta un ERD de corta duración cerca del inicio del movimiento, seguido por un ERS con un máximo al segundo de ejecutar el movimiento. Con respecto a las oscilaciones de la banda γ (36-40Hz), presentan un ERS antes de realizar el movimiento, pudiendo proveer información y ser útiles en ocasiones, sin embargo, estas oscilaciones de alta frecuencia no están presentes en todos los seres humanos.

En el gráfico inferior se muestra el EEG en cada electrodo para el mismo movimiento. Se pueden observar claramente los ERD en las posiciones centrales de los electrodos (C3, C4 y Cz) antes de realizar el movimiento y los ERS inmediatamente después del mismo sobre los electrodos de la zona visual del córtex (ritmo α , P3, P4, Oz).

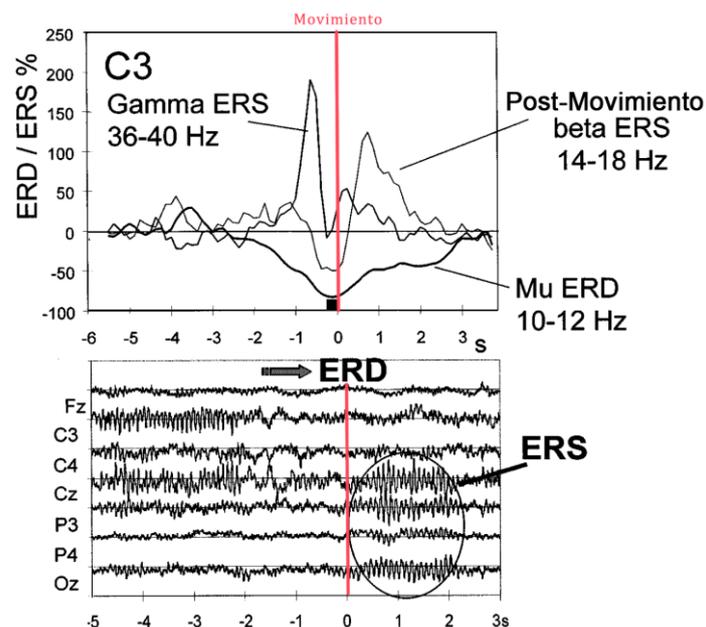


Figura 2-8. Señal EEG registrada antes y después de realizar un movimiento con el dedo índice de la mano donde se aprecian los ERD y ERS. El panel superior se corresponde al electrodo C3 [5].

Se ha demostrado que no solamente la ejecución del movimiento genera los ERD y los ERS, también la imaginación del movimiento o la visualización del mismo provocan los mismos cambios en los ritmos μ y β . Esta es la razón por la cual un sistema BCI basado en ritmos sensoriomotores se trata como un sistema independiente, el cual no depende de la capacidad del usuario para fijar la vista en un punto concreto o mover una articulación, simplemente es necesario imaginar el movimiento para generar la señal de control [15].

Al igual que los SCP, la señal de control generada gracias a los ritmos sensoriomotores es endógena y, por tanto, requiere entrenamiento para poder ser controlada de manera adecuada. Con un entrenamiento de unas 6 horas se pueden llegar a alcanzar precisiones por encima del 80% y tasas binarias de 20-25 bits por minuto [5].

2.3.4. Potenciales de neuronas corticales

Los potenciales de neuronas corticales se corresponden a la activación de neuronas aisladas. Estos potenciales se recogen mediante electrodos epidurales o intracorticales, considerándose un método invasivo. En la Figura 2-9 aparece un esquema sobre la implantación de un electrodo intracortical.

Diversos estudios con monos han demostrado que éstos son capaces de aprender a controlar la activación de neuronas aisladas, así que se espera que los humanos sean capaces también. Sin embargo, la evaluación de esta especulación se retrasó debido a la ausencia de los electrodos adecuados y a que no era posible registrar la actividad neuronal de forma estable a largo plazo (normalmente no superan el año). De hecho, la neurona que está siendo monitorizada por el electrodo en cuestión puede morir o relevar su función a otra neurona distinta, causando que el electrodo se vuelva inútil [5].

Utilizando esta señal de control se podría controlar un cursor en 1D para seleccionar comandos o letras en una pantalla. También podría llegar a controlarse un cursor en 2D si se logra el control de la actividad EMG residual (se obtuvieron tasas de 15bits por minuto y precisiones del 67% en animales). Sin embargo, no se puede generalizar con total seguridad para los humanos y debido a la escasez de experimentos con los mismos no se puede determinar que su uso sea satisfactorio.

Dado que los experimentos con animales se realizaron cuando éstos realizaron los movimientos oportunos, tampoco se conoce si se producirán los mismos patrones en la actividad neuronal cuando no se realice el movimiento debido a la incapacidad del usuario para realizarlo.

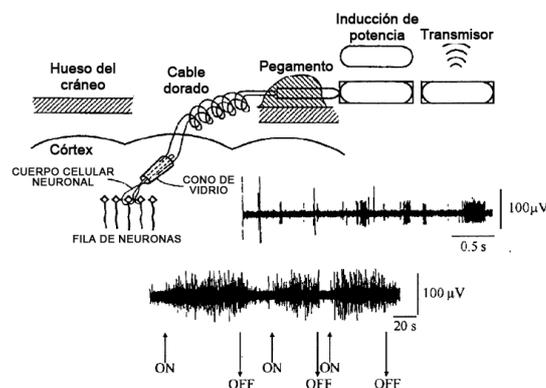


Figura 2-9. Esquema que muestra la implantación de electrodos epidurales o intracorticales [5].

2.3.5. Potenciales evocados P300

Los potenciales evocados P300 son deflexiones positivas de voltaje localizadas en la zona parietal del córtex como respuesta a un estímulo que aparecen en torno a 300ms después de que éste se produzca [5]. Por tanto, el potencial P300 es una señal exógena (no requiere entrenamiento) provocada por una respuesta natural del cerebro ante un estímulo, ya sea visual, auditivo o somato-sensorial [2]. Los electrodos que se emplean para recoger este potencial normalmente se sitúan en la región centro-parietal del córtex, típicamente en Cz, Fz y Pz.

La aparición de los potenciales evocados P300 se provoca de acuerdo al paradigma *odd-ball*, el cual trata de presentar un estímulo deseado de forma infrecuente camuflado entre estímulos frecuentes. Es decir, el potencial evocado P300 se produce cuando la opción que pretende elegir el usuario se encuentra entre muchas otras opciones que no interesan. En consecuencia, se concluye que cuanto más improbable sea el estímulo deseado, más probable será que aparezca el potencial evocado P300 y que su amplitud sea mayor.

La aplicación más frecuente de este paradigma se basa en presentar una matriz cuyas celdas sean las opciones a seleccionar: habitualmente letras, números o comandos de una aplicación. En cada intento se iluminan aleatoriamente las filas y las columnas de forma que, tras un número determinado de iluminaciones, cada fila y columna se haya iluminado una vez. El usuario, fijándose en la celda que desea seleccionar, provocará dos potenciales evocados P300 cuando se iluminen la fila y la columna que contienen dicha celda. Promediando la respuesta para cada elemento de la matriz y detectando el potencial P300 con mayor amplitud se determina cuál es la celda seleccionada por el usuario [5]. El resultado de este tipo de aplicaciones se puede observar en la Figura 2-10.

Dado que se trata de una señal exógena y no requiere entrenamiento, se pueden alcanzar precisiones muy altas en intervalos muy cortos de tiempo (90% de precisión con 4 selecciones por minuto [5]). Sin embargo, su rendimiento puede verse afectado a lo largo del tiempo cuando el usuario se acostumbra a los estímulos que generan los P300, pudiendo dejar de ser notables.

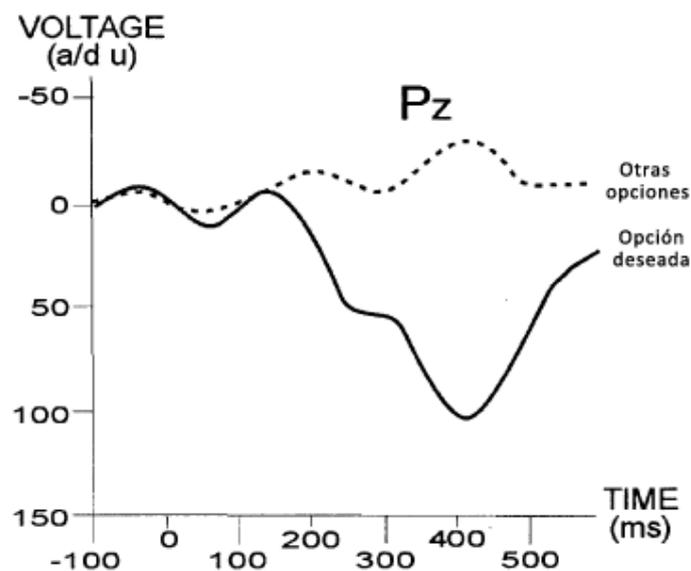


Figura 2-10. Potencial evocado P300 recogido sobre el área centro-parietal del córtex mediante la técnica *odd-ball* con una matriz luminosa [5].

2.4. Procesado de la señal EEG

El tratamiento de la señal EEG puede dividirse en tres etapas principales: adquisición de la señal, procesado de la señal y aplicación. A su vez, el procesado de la señal se puede subdividir en otras dos etapas, siendo éstas la extracción de características y la traducción de las mismas. En la Figura 2-11 se representan de manera esquemática.

2.4.1. Adquisición de la señal

La primera etapa en el tratamiento de la señal EEG es la de la adquisición de la señal. Esta etapa se encarga de obtener la señal, amplificarla, digitalizarla, eliminar los artefactos que la degraden y almacenarla para un posterior análisis *offline*.

Tal y como se comentó en el capítulo de introducción, la obtención de la señal EEG se realiza mediante la colocación de los electrodos superficiales atendiendo al *sistema internacional 10/20* con la ayuda de un gel conductor que reduzca la impedancia del cráneo. Posteriormente, se amplificará y se digitalizará con una frecuencia de muestreo adecuada. De acuerdo al teorema de Nyquist debe ser mayor o igual que el doble de la frecuencia máxima de la banda que deseemos analizar.

A continuación, la señal debe ser filtrada para eliminar los artefactos que pueda contener. En la señal EEG se distinguen tres artefactos principales que son capaces de degradar la señal, siendo éstos el movimiento ocular (EOG) y pestañeos, la actividad de músculos cercanos y la actividad cardiaca (ECG) [2].

- **Movimiento ocular y pestañeos.** La actividad eléctrica producida por el movimiento ocular, o EOG, es claramente visible en la señal EEG debido a su gran intensidad. El EOG refleja la diferencia de potencial que varía durante el movimiento ocular entre la córnea y la retina, cuya amplitud se comporta de manera proporcional al ángulo de visión. La intensidad de esta señal depende de la proximidad del electrodo y, por esta razón, este artefacto molesta más en los electrodos situados en la zona frontopolar del córtex, efecto que puede observarse en la Figura 2-12. Este artefacto no solo está presente en el estado de vigilia, sino que aparece también cuando el sujeto duerme, concretamente cuando entra en fase REM. El pestañeo también produce una actividad molesta, de mayor frecuencia que el movimiento ocular y con cambios más abruptos.



Figura 2-11. Etapas en las que se divide el tratamiento de la señal EEG: adquisición de la señal, procesamiento de la señal (extracción y traducción de características) y aplicación.

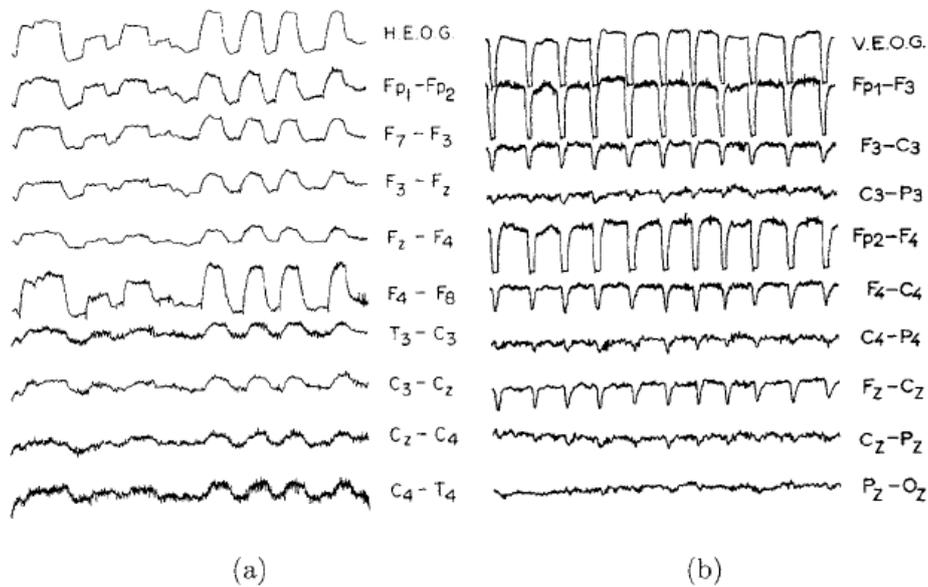


Figura 2-12. Artefactos en la señal EEG causados por (a) el movimiento ocular y (b) pestañeos repetitivos y voluntarios.

En términos de procesamiento, el artefacto causado por el movimiento de los ojos se puede reducir obteniendo una señal de referencia en la que solamente aparezca la molestia, colocando electrodos cerca de los ojos. De esta manera se podría eliminar el artefacto, por ejemplo, con un filtrado adaptativo.

- **Actividad muscular cercana.** Otro artefacto común lo causa la actividad eléctrica muscular, sobre todo, de músculos cercanos. Esta actividad se encuentra normalmente cuando el sujeto está en estado de vigilia y se encuentra tragando, haciendo muecas, ciñendo el cejo, masticando, hablando, o en resumen, realizando cualquier actividad que implique mover los músculos faciales o la mandíbula. Este artefacto se reduce drásticamente cuando el sujeto se encuentra relajado o dormido. El efecto que produce depende de la fuerza de la contracción muscular, pudiendo generar irregularidades de baja amplitud si la contracción es débil o altas irregularidades que se asemejan a ruido blanco cuando la contracción es fuerte. Este artefacto puede observarse en la Figura 2-13.

En términos de procesamiento, la actividad muscular es un artefacto mucho más molesto que el movimiento ocular, puesto que solapa la *banda β* dentro del rango entre 15 y 30Hz. Esta desventaja está agravada por el hecho de que es imposible adquirir una señal de referencia que solamente contenga la actividad muscular molesta, a diferencia del EOG.

- **Actividad cardiaca.** El ECG puede llegar a interferir con la señal EEG en algunas personas. Aunque la amplitud del ECG normalmente es baja en el cráneo en comparación con el EEG, puede aparecer en determinados electrodos y en determinados cuerpos, por ejemplo, en personas bajas con cuellos cortos. Debido a la naturaleza repetitiva y regular que caracteriza al ECG, la eliminación del mismo resulta sencilla.

Para finalizar esta etapa, aunque la aplicación a desarrollar se realiza en tiempo real, se puede almacenar el registro para poder realizar un posible análisis *offline*.

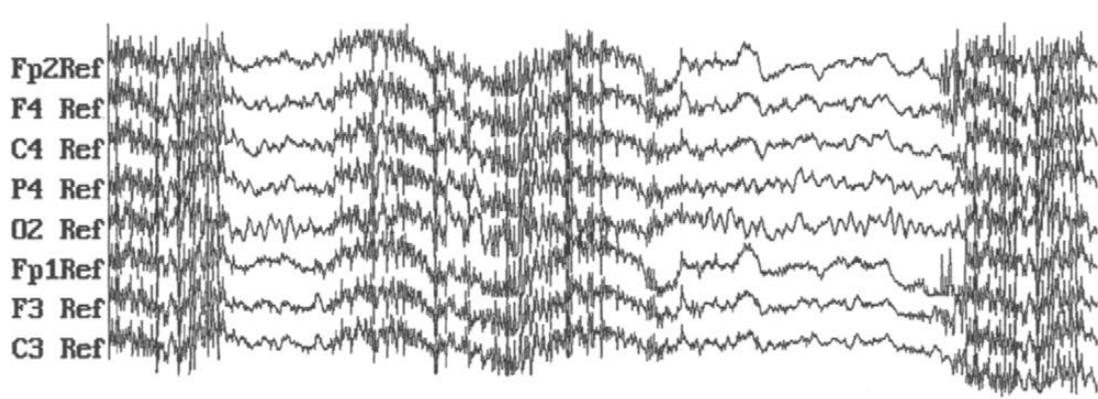


Figura 2-13. Registro de 5 segundos de duración de la señal EEG contaminada por episodios intermitentes de artefactos electromiográficos (EMG).

2.4.2. Procesado de la señal

La segunda etapa en el tratamiento de la señal EEG es la de la del procesado de la señal, a su vez subdividida en la extracción de características y la traducción de las mismas.

a) Extracción de características

La etapa de extracción de características se basa en obtener información que sea capaz de discriminar o identificar la señal EEG. Esta información se denomina característica y será lo que permita diferenciar entre distintas patologías o intenciones del usuario una vez sea analizada.

Aparte de los artefactos ya mencionados, existen otros factores que pueden afectar a las características de la señal del usuario. Entre ellos se encuentran la motivación, la intención, la frustración, la fatiga, etcétera. El buen funcionamiento del sistema BCI depende de la adaptabilidad del sistema al usuario así como de la selección de las técnicas de procesado de señal adecuadas [5].

Aunque en otros tipos de sistemas que tratan también las señales biomédicas la cancelación de artefactos y la extracción de características están totalmente diferenciadas, en los sistemas BCI muchos autores clasifican técnicas que se pueden emplear para cancelación de artefactos como técnicas de extracción de características debido a que su salida usualmente se convierte en la entrada del clasificador. En la Tabla 2-1 se muestran una serie de técnicas de extracción de características empleadas con anterioridad en los sistemas BCI [16].

Los principales métodos se dividen en dos grandes grupos: los que trabajan con señales en tiempo o frecuencia y los que trabajan con señales en el espacio. Lo más común es aplicar un método de cada tipo indistintamente del orden en el que se apliquen, tratando las dimensiones de manera independiente, o emplear un solo método espacio-temporal.

Dentro de los métodos que trabajan en el dominio temporal y frecuencial destacan los métodos temporales a causa de su rapidez y del poco coste computacional que requieren. Estos métodos suelen utilizarse en aplicaciones BCI a tiempo real y la elección de los mismos depende en gran parte de la señal de control utilizada y de los artefactos que la degraden. Por ejemplo, un simple promediado sincronizado aumenta de manera significativa la capacidad de detección de los

TABLA 2-1. CLASIFICACIÓN DE LOS MÉTODOS MÁS EMPLEADOS EN LA EXTRACCIÓN DE CARACTERÍSTICAS.

Tiempo-Frecuencia	Espacio	Tiempo-Espacio	Modelos Inversos
Transformada de Fourier	Filtro Laplaciano	Análisis en Componentes de Tiempo y Espacio	EEG a ECoG
Transformada Wavelet			
Modelos Autorregresivos (AR)	Análisis de Componentes Principales (PCA)	Modelos Autorregresivos Multivariable	
Filtrado Paso-Banda	Análisis de Componentes Independientes (ICA)		
Filtro Adaptado	Patrones Espaciales Comunes (CSP)	Coherencia	EEG a Dipolos Fuente
Filtro de Kalman	Amplitudes		
Detección de Pico	Proporciones y Diferencias		

potenciales evocados P300. Sin embargo, los métodos que operan en el dominio frecuencial requieren una transformación de la señal, convirtiéndose así en métodos que requieren más coste computacional que los temporales. De esta manera, para aplicaciones a tiempo real interesan aquellos métodos espectrales que operan con segmentos de la señal relativamente cortos, descartando la transformada de Fourier.

Los métodos espaciales también dependen de la señal de control que utiliza el sistema y, concretamente, de lo localizada que esté su fuente en la corteza cerebral. Por consiguiente, el método más adecuado para detectar los ritmos μ y β , que se encuentran bastante localizados en el córtex, no será el mismo que para detectar los potenciales SCP o los P300, que se encuentran más distribuidos a lo largo del mismo.

Una medida útil para realizar un análisis *offline* de los métodos espaciales es el coeficiente de determinación o r^2 , que básicamente consiste en la correlación de los datos elevada al cuadrado. En la Figura 2-14 se muestra el resultado de este coeficiente para intentar detectar los ritmos μ y β sobre el electrodo C3 empleando cuatro métodos espaciales diferentes, siendo éstos el filtro Laplaciano con dos distancias distintas de separación de los electrodos, el método de referencia de media común (CAR) y el uso del electrodo C3 con una única referencia en la oreja. Tal y como se puede observar, el filtro Laplaciano con una distancia entre electrodos de 6cm o el método de referencia de media común (CAR) proporcionan mejores resultados que los otros dos métodos analizados [5].

Además de los cuatro métodos espaciales comparados, es frecuente en los sistemas BCI utilizar PCA (*Principal Component Analysis*, Análisis de Componentes Principales) o ICA (*Independent Component Analysis*, Análisis de Componentes Independientes), útiles para realizar una combinación lineal de varios canales y seleccionar la información más relevante. Por esta razón también son conocidos como métodos de reducción de dimensionalidad.

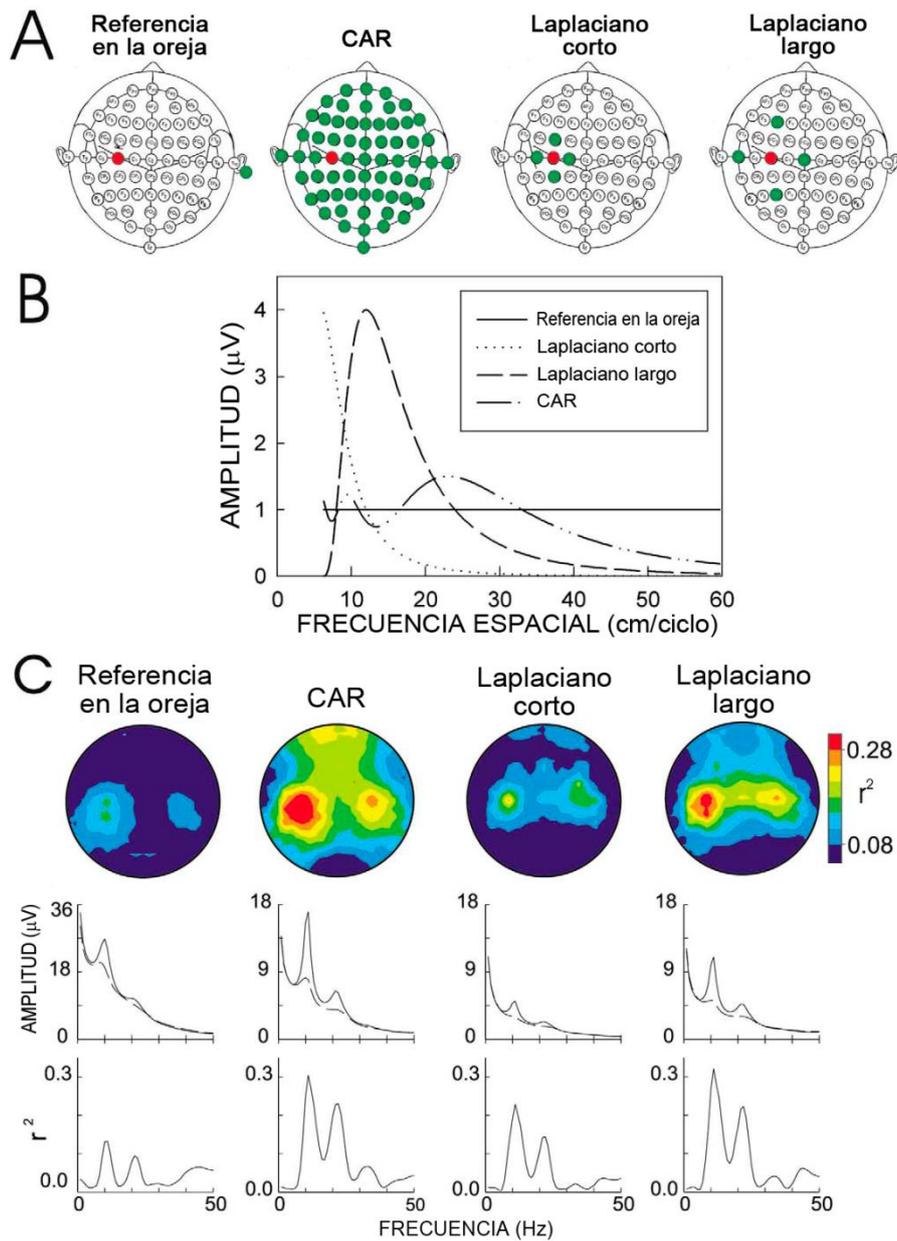


Figura 2-14. Comparación entre cuatro métodos espaciales de extracción de características. (A) Localización de los electrodos utilizados con la señal objetivo en rojo, concretamente, se quiere medir la señal del electrodo C3. (B) Banda de paso para cada método, raíz cuadrada de los valores cuadráticos medios de la señal recogida en C3. (C) Topografía de r^2 medida y amplitud espectral para cada método estudiado [5].

b) Traducción de características

La etapa de traducción de características, o algoritmo de clasificación, se encarga de convertir las características extraídas anteriormente en comandos de un dispositivo. Para ello, clasificará las características obtenidas en función de diversos parámetros y las asociará con un comando determinado [5]. El objetivo es asegurar que la selección atribuida se corresponda con la selección que el usuario pretendía realizar.

Los clasificadores se pueden dividir en dos grupos: clasificadores lineales y no lineales. En la Tabla 2-2 aparecen los clasificadores más usados en los sistemas BCI para ambos grupos.

TABLA 2-2. CLASIFICACIÓN DE LOS MÉTODOS MÁS EMPLEADOS EN LA TRADUCCIÓN DE CARACTERÍSTICAS.

Lineales	No Lineales			
	Estructura Fija	Estructura Modificable		
		Basados en Memoria	Combinaciones de No Linealidades Simples	Modelos Generativos
Análisis Discriminante Lineal (LDA)	Análisis Cuadrático Discriminante (QDA)	Algoritmo del vecino más próximo	Redes Neuronales Artificiales (ANN)	Modelo de Mezclas Gaussianas
Perceptrón		Máquinas de Vectores de Soporte (SVM)	Árbol de Decisión	Modelo Oculto de Markov (HMM)
Regresión, regularización y adaptación		Regresión de Mínimos Cuadrados Parciales (PLS Regression)	Aprendizaje por Cuantificación Vectorial (LVQ)	

Los métodos lineales asumen que los datos pueden separarse linealmente a través de un hiperplano de separación, el cual divide el espacio de datos en varias regiones, cada una correspondiente a una selección distinta. Este hiperplano de separación se coloca de tal manera que minimice el margen mínimo (distancia mínima entre cada punto y el hiperplano) o el margen medio [17]. En la Figura 2-15 se puede observar un clasificador lineal que maximiza el margen mínimo.

Pese a que los métodos lineales suelen ser más robustos que los no lineales y suelen requerir menor carga computacional, también pueden fallar en presencia de ruido u *outliers*, valores atípicos que provocan datos numéricamente distantes del resto, frecuentemente engañosos. En la Figura 2-15 (b) se observa cómo la presencia de *outliers* puede cambiar la localización del hiperplano de separación si la influencia de valores atípicos no está limitada. Se recomienda aplicar una regulación de los datos para limitar su influencia antes de aplicar el clasificador para obtener un hiperplano más fiable.

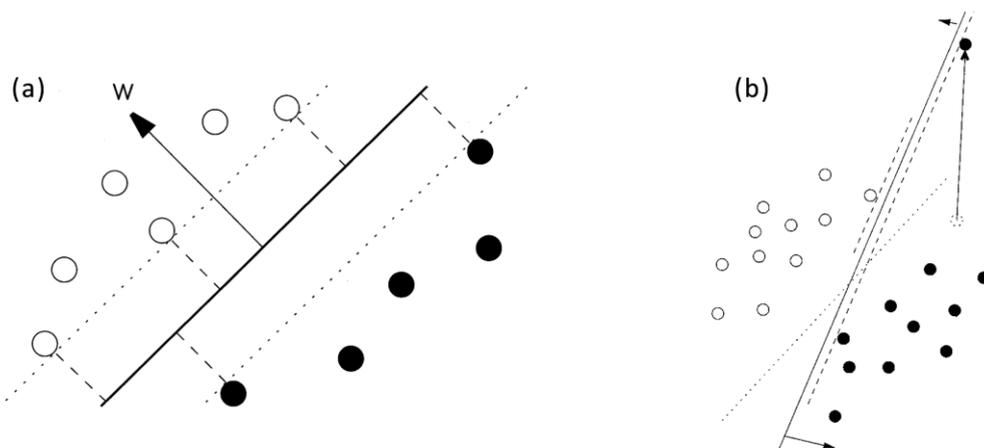


Figura 2-15. Clasificador lineal que maximiza el margen mínimo: en (a) se muestra el comportamiento óptimo y en (b) su comportamiento en presencia de un *outlier*.

Los métodos no lineales son capaces de obtener precisiones mayores al tener en cuenta un mayor número de parámetros que los métodos lineales a cambio de requerir un mayor coste computacional [17].

En el estudio de Fabiani *et al* [18] se comparan tres métodos de traducción de características para diferenciar entre tres clases de datos: opción superior, inferior y central. El resultado puede observarse gráficamente en la Figura 2-16. Los dos primeros clasificadores son lineales: K1 y K2. Sin embargo, K1 solamente emplea una dimensión, mientras que K2 emplea dos dimensiones. El clasificador K3 se corresponde con una función no lineal de K1. Tal y como se puede observar se aprecia una mejora significativa en los resultados de los clasificadores K2 y K3. Esto se debe principalmente a que K1 consiste en dos hiperplanos de decisión paralelos, por lo que impone una fuerte restricción en la clasificación de los datos.

Generalmente se prefiere la simplicidad de los métodos lineales. Los métodos de clasificación no lineales son preferibles en el caso de que el número de muestras sea pequeño y la información sobre los datos sea limitada [17].

Adicionalmente, la etapa de traducción de características también debe considerar la adaptabilidad al usuario [5]. Los clasificadores óptimos varían entre usuarios, es decir, el clasificador óptimo para un usuario normalmente no será el mismo que el clasificador óptimo para otro usuario distinto y, debido a esto, el sistema BCI debe cumplir una serie de requisitos repartidos en tres niveles.

El primer nivel requiere adaptar las características de las señales a cada usuario, puesto que habrá diferencias significativas entre un usuario y otro, por ejemplo, la amplitud de los potenciales SCP o P300. El segundo nivel requiere adaptar el clasificador a las fluctuaciones que pueden sufrir las características obtenidas debido a variaciones entre la motivación, la fatiga o la frustración, entre otros, del usuario dependiendo de la sesión. Finalmente, el tercer nivel requiere mantener la precisión del clasificador a largo plazo, siendo capaz de detectar disminuciones en los niveles de atención del usuario, responder ante ellos y ofrecer una realimentación que lo motive para los próximos intentos.

El tercer nivel está fuertemente ligado al fenómeno de aprendizaje de los clasificadores. Existen una serie de clasificadores, habitualmente no lineales, que son capaces de “aprender” las características de la señal de cada usuario y adaptarse a ellas gracias a una serie de datos de entrenamiento. Entre ellos se encuentran las máquinas de vectores de soporte (SVM, *Support Vector Machines*) o las redes neuronales artificiales (ANN, *Artificial Neural Networks*), caracterizados por ser algoritmos muy rápidos una vez están entrenados, y adecuados para aplicaciones en tiempo-real.

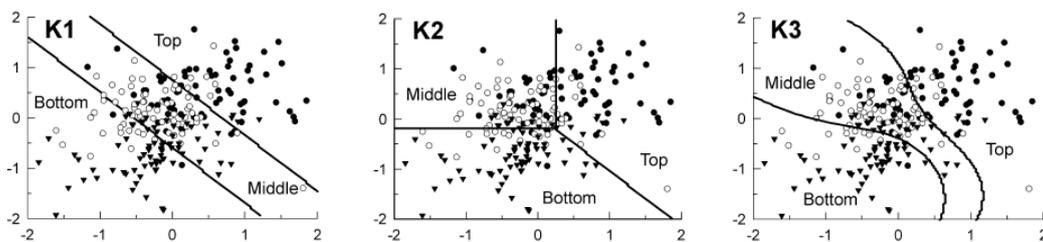


Figura 2-16. Comparación de los hiperplanos de separación para los tres métodos de traducción de características: K1 es un método lineal de una dimensión, K2 es un método lineal de dos dimensiones y K3 es un método no lineal.

2.4.3. Aplicación

La tercera y última etapa del tratamiento de la señal es la etapa de aplicación, la cual se encargará de recibir la señal de control producida por la traducción de características y de implementar un *software* que traduzca ésta señal en comandos que realicen las acciones correspondientes. En la siguiente sección se detallarán las aplicaciones más comunes de los sistemas BCI.

2.5. Aplicaciones de los sistemas BCI

Existen multitud de aplicaciones prácticas desarrolladas con sistemas BCI. A continuación se exponen las aplicaciones más básicas que utilizan las señales de control mencionadas anteriormente, como son la selección de letras, el movimiento de un cursor o el desplazamiento sobre un mapa.

2.5.1. Selección de letras

La aplicación de selección de letras es, probablemente, la más utilizada en los sistemas BCI. Un sinnúmero de aplicaciones más complejas utilizan técnicas de selección de letras, números o comandos para implementar alguna funcionalidad, ya sea con potenciales evocados P300, potenciales VEP o potenciales SCP.

a) Potenciales evocados P300

Las aplicaciones de selección de letras, números o comandos con potenciales evocados P300 se basan en la técnica *odd-ball* descrita anteriormente. Se presenta al usuario una matriz de, por ejemplo, 6x6 letras y números. Cada 125ms se ilumina una fila o columna de forma que tras una ronda de 12 iluminaciones cada fila y columna se han iluminado una vez y, por lo tanto, cada letra se ha iluminado dos veces.

Cada vez que se ilumina la letra que el usuario quiere seleccionar, éste provoca la aparición de dos potenciales evocados P300. El número de intentos que serán necesarios para obtener buenas precisiones lo determinará el clasificador que se haya generado para cada usuario y la experiencia del mismo con estos sistemas. Normalmente 10 intentos son suficientes para obtener una buena precisión [43].

b) Potenciales Evocados Visuales

Una manera de seleccionar letras para mejorar la comunicación de personas con graves discapacidades utilizando como señal de control los potenciales VEP se muestra a continuación.

La interfaz de la aplicación presenta una matriz de letras, cuyas celdas se iluminan continuamente a distintas frecuencias. El usuario debe mirar fijamente a la celda que contiene la letra que desea escribir, generando un fenómeno de resonancia a la misma frecuencia a la que se ilumina la letra a seleccionar. El sistema identifica los picos frecuenciales de ésta y sus armónicos y determina hacia dónde está mirando el usuario. Si la matriz es demasiado grande como para utilizar todas las frecuencias indistintamente, se pueden iluminar aleatoriamente subgrupos de letras hasta que cada una de ellas se haya iluminado a una frecuencia diferente.

El utilizar una señal de control exógena tiene la ventaja de no requerir entrenamiento alguno. Sin embargo, sí es necesaria la capacidad de mantener la mirada fija en un punto.

c) Potenciales Corticales Lentos

Dado que utilizando únicamente los potenciales SCP solamente pueden realizarse selecciones binarias, la selección de letras se lleva a cabo dividiendo sucesivamente el alfabeto en dos partes: primero se elige una mitad, luego un cuarto, etcétera.

Teniendo en cuenta que los potenciales corticales lentos se clasifican dentro de las señales de control endógenas, el usuario necesitará entrenamiento para poder seleccionar las letras adecuadamente. En la Figura 2-17 se puede observar la solución de Bensch *et al* [19].

2.5.2. Movimiento de un cursor

El objetivo de esta aplicación será el de aprovechar el control de movimiento de un cursor para seleccionar botones, navegar por menús o moverse dentro de algún videojuego.

a) Potenciales Corticales Lentos

Una aplicación utilizando potenciales corticales lentos se basa en mostrar por pantalla un cursor que se desplace horizontalmente a velocidad constante. El usuario debe controlar el movimiento vertical con los potenciales corticales lentos y así seleccionar una de las dos opciones posibles. Este procedimiento puede observarse en la Figura 2-18.

b) Ritmos Sensoriomotores μ y β

A diferencia de los cursores controlados mediante potenciales corticales lentos, los ritmos sensoriomotores μ y β pueden llegar a controlar un cursor en dos dimensiones con el suficiente entrenamiento. De esta manera, se pueden controlar menús complejos, aplicarse al control de prótesis, al movimiento de sillas de ruedas o el movimiento en videojuegos, tal y como se muestra en la Figura 2-19 [20].

2.5.3. Desplazamiento sobre un mapa

Esta aplicación se puede ver como una extensión de la anterior. El usuario se coloca frente a una pantalla en la que se muestra un mapa por satélite junto a cuatro botones que controlan el movimiento vertical y horizontal.

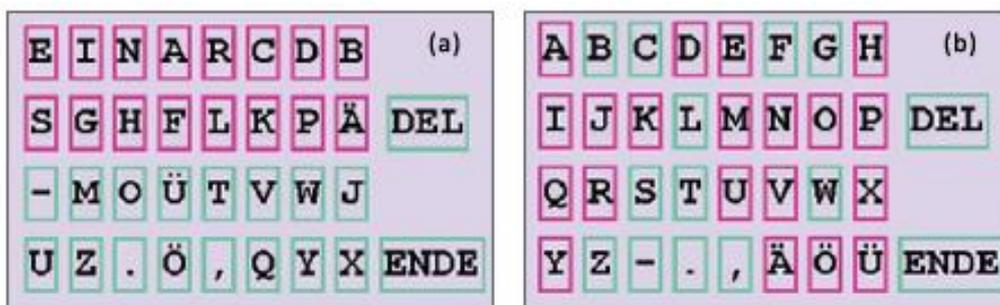


Figura 2-17. Selección de letras utilizando potenciales corticales lentos (SCP) con selecciones binarias: mitad de las mismas en magenta y la otra mitad en azul [19].

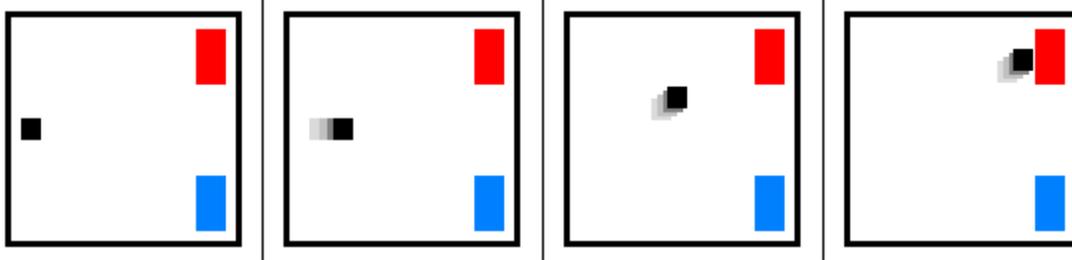


Figura 2-18. Control del movimiento de un cursor verticalmente mediante potenciales corticales lentos para seleccionar entre dos opciones.

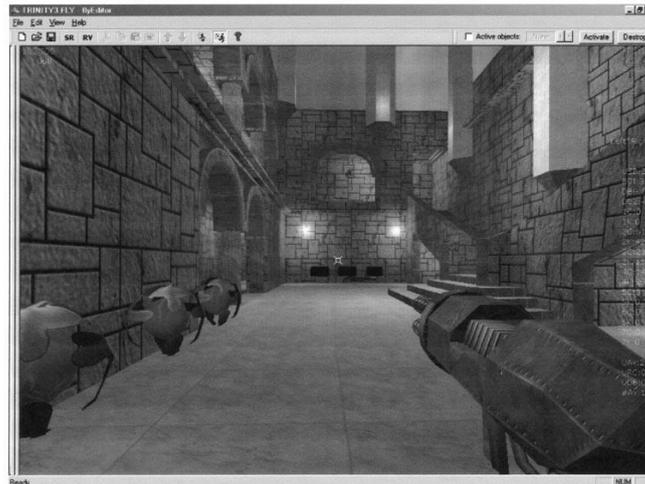


Figura 2-19. Control del movimiento en videojuegos mediante ritmos sensoriomotores μ y β [20].

En el trabajo de Trejo *et al* [21] se desarrolla este sistema, bautizado como *Think Pointer BCI System* (o sistema de puntero dirigido por el pensamiento) empleando potenciales VEP. Cada botón se ilumina a una frecuencia distinta y el usuario mira fijamente al que quiere seleccionar, controlando así el desplazamiento sobre el mapa. Midiendo la frecuencia en la que aparecen los VEP del usuario, se determina la dirección seleccionada. En la Figura 2-20 aparece la pantalla que se muestra al usuario.

Una aplicación similar podría controlarse mediante los ritmos μ y β , controlando el desplazamiento en dos dimensiones, tal y como se ha utilizado anteriormente en el control del movimiento de un cursor.

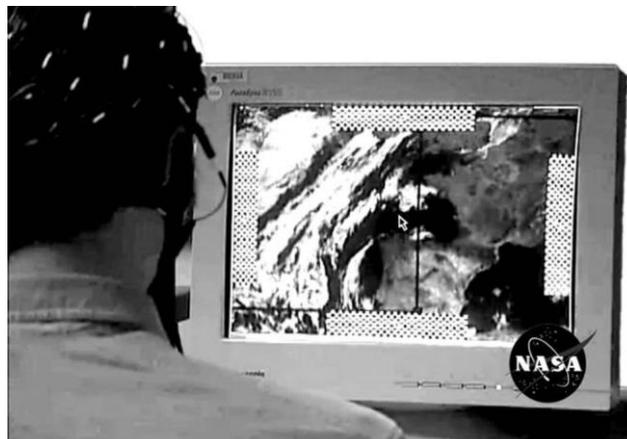


Figura 2-20. Control del desplazamiento sobre un mapa con potenciales evocados visuales [21].

2.6. Navegadores Web BCI

Una vez vistas las aplicaciones más sencillas en las que se basan los sistemas BCI es hora de introducir una revisión del estado del arte sobre los navegadores web desarrollados con anterioridad. La decisión de cómo implementar la aplicación a desarrollar en el presente trabajo de fin de grado se basa en la comparativa que se muestra a continuación. En esta comparativa se estudian cinco artículos distintos, atendiendo a las señales de control que utilizan y a las funcionalidades que implementan.

2.6.1. Bensch *et al* — “Nessi: An EEG-Controlled Web Browser for Severely Paralyzed Patients” [19]

El primer artículo a tratar se centra en mejorar el primer navegador web creado mediante BCI, bautizado como Descartes, utilizando potenciales SCP como señal de control.

Nessi (*Neuronal Signal Surfing Interface*), el navegador web implementado como mejora de Descartes, funciona a través de selecciones binarias controladas por potenciales SCP. Los potenciales corticales lentos se corresponden a cambios de voltaje, entre 0.5 y 10s de duración, generados en el córtex a bajas frecuencias tras seleccionar una opción binaria (aumento o reducción de la activación cortical) [5].

La interfaz se compone del navegador en la parte derecha de la pantalla y la selección binaria en la parte izquierda. Ésta distingue entre dos objetivos: el magenta y el azul; y se ofrece una realimentación al usuario en forma de círculo amarillo para que el control sea más intuitivo. El usuario seleccionará el objetivo magenta desplazando el círculo amarillo hacia la parte superior y seleccionará el objetivo azul desplazando el círculo amarillo hacia la parte inferior, tal y como se observa en la Figura 2-21 (a).

Todos los hipervínculos de la página web que se esté visualizando estarán rodeados por rectángulos magentas o azules. El usuario deberá elegir el color que corresponda al hipervínculo que desea seleccionar y, tras sucesivas selecciones, se habrán descartado todos los hipervínculos no deseados. Para facilitar la selección de los hipervínculos todos aquellos que apunten a la misma URL se agruparán.

Una vez se ha seleccionado una página, el usuario tiene un tiempo de lectura predeterminado por el supervisor. Durante este tiempo se desactiva el control de selección y la realimentación. Si el usuario se confunde de selección, podrá corregir su error. No obstante, si una web carece de hipervínculos, el navegador volverá a la página anterior una vez haya terminado el tiempo de lectura. Con respecto a la estructura de la página inicial, se ofrece una lista de las URLs más usadas por el usuario, es decir, una lista de marcadores.

A la hora de introducir caracteres en algún formulario web, aparecerá un teclado virtual controlado también por potenciales SCP. El color del marco rectangular de cada carácter no es aleatorio, sino que puede estar basado en distintas codificaciones, entre ellas *Huffman-Coded* o *LSP* (se ha demostrado que los mejores resultados los proporciona la codificación *LSP*). En la Figura 2-21 (b) se observan los posibles teclados virtuales que implementa la aplicación.

Adicionalmente, se ha diseñado una interfaz para leer y escribir e-mails. La pantalla se divide en varias partes, cada una de ellas con una función distinta: mostrar los e-mails, leer un e-mail y escribir un e-mail. Las direcciones estarán contenidas en un *Address Book* que tendrá que crear el supervisor.

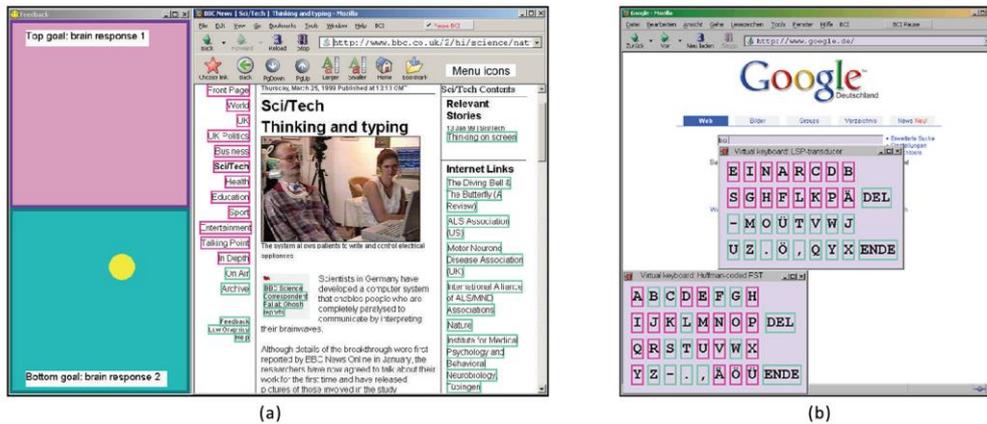


Figura 2-21. Interfaz gráfica de *Nessi*, el primer navegador basado en potenciales SCP. En (a) se muestra la interfaz de selección binaria y la realimentación al usuario, mientras que en (b) aparecen las distintas codificaciones del teclado virtual [19].

Nessi fue evaluado por cuatro pacientes con ALS (*Amyotrophic Lateral Sclerosis*, o Esclerosis Lateral Amiotrófica) y dos personas sanas, sin embargo, no hay resultados publicados.

2.6.2. Mugler *et al* — “Design and Implementation of a P300-Based Brain-Computer Interface for Controlling an Internet Browser” [22]

A diferencia de las publicaciones anteriores, en este artículo se desarrolla un nuevo navegador web controlado por potenciales evocados P300. El objetivo es asegurar un verdadero acceso a Internet para los pacientes, los cuales podrán visitar cualquier página accesible para los navegadores habituales.

Gracias al código abierto de Mozilla Firefox y sus *add-ons*, el navegador desarrollado es capaz de navegar por cualquier web, introducir URLs y manejar la barra de desplazamiento (*scroll up & down*).

Los hipervínculos se etiquetan con uno o dos caracteres y se seleccionan introduciéndolos con la técnica de selección de letras explicada anteriormente. De esta manera, la búsqueda del hipervínculo deseado es mucho más rápida que con un sistema de selección binaria. Para introducir los caracteres se usa la técnica *odd-ball*, mostrando una matriz de caracteres cuyas filas y columnas se iluminan aleatoriamente. El usuario debe prestar atención al carácter que le gustaría seleccionar y, en consecuencia, se generan dos potenciales P300, uno cuando se ilumina la columna que lo contiene y otro cuando se ilumina la fila [5]. La matriz utilizada se puede observar en la Figura 2-22.

Se introducen pausas de 5s de manera predeterminada para evitar los distintos tiempos de carga de las páginas web. En consecuencia, no se podrá realizar ninguna acción hasta que hayan finalizado. También se incluye un modo lectura que deshabilita todos los comandos exceptuando el movimiento de la barra de desplazamiento. Si el usuario introduce un comando incorrectamente puede corregirlo utilizando la opción de *back* (si seleccionó un link no deseado), *reload* (si se introdujo una letra incorrecta en el filtro) o *delete* (para errores tipográficos).

Este sistema fue probado con 10 personas sanas y 3 pacientes con ALS. En cada sesión se ordenó a los pacientes realizar una serie de tareas, tales como buscar términos en la *Wikipedia*, buscar información sobre la universidad local o comprar un libro específico por Internet.



Figura 2-22. Matriz utilizada para provocar los potenciales evocados P300 en el navegador web desarrollado por Mugler *et al* [22].

Las sesiones dedicadas a los participantes con ALS fueron interrumpidas en varios intervalos de tiempo por problemas respiratorios, excesiva generación de saliva, etcétera. Las personas sanas manejaron el navegador web desarrollado con una precisión del 90%, mientras que los pacientes con ALS lo hicieron con una precisión del 72%. Consecuentemente, es evidente que la neurodegeneración en el área motora puede provocar una gran reducción de atención y un movimiento ocular mucho más lento que el de una persona sana.

2.6.3. Sirvent Blaco *et al* — “Visual Evoked Potential-Based Brain-Machine Interface Applications to Assist Disabled People” [23]

En este artículo se desarrollan tres aplicaciones orientadas a personas discapacitadas basadas en los potenciales evocados P300 y N2PC. La primera aplicación es un navegador web, la segunda un control de una mano robótica y la tercera es una herramienta de comunicación básica. Este informe se centrará en la aplicación de interés, es decir, el navegador web.

Este sistema pretende aumentar el rendimiento del navegador tratando de precisar más la detección de los potenciales evocados teniendo en cuenta no solo los potenciales P300, sino también los N2PC. El estímulo que produjo los potenciales P300 y N2PC se generó con la tradicional técnica *odd-ball*. Se ilumina una matriz de selección aleatoriamente y los usuarios generan involuntariamente los potenciales al ver iluminada la celda que quieren seleccionar, además, cuanto menor sea la probabilidad del evento mayor serán los potenciales evocados [5].

La señal P300 se corresponde con una deflexión positiva de la señal EEG unos 300ms después del estímulo. Para detectar los potenciales P300 se realizó un filtrado espacial y como consecuencia se utilizaron los electrodos Cz, Fz, Pz y cercanos (e.g C3 y C4). El segundo tipo de potencial evocado que interesa es el N2PC, el cual se corresponde a una deflexión negativa de la señal EEG en torno a unos 200ms después del estímulo. La “N” denota una polaridad negativa, el “2” su la latencia a los 200ms, “P” viene de posterior o parietal y “C” de contralateral al movimiento. Para detectar correctamente este potencial se ubicaron los electrodos en Po4, Po3 y cercanos (e.g P3, P4, PO1 y PO3).

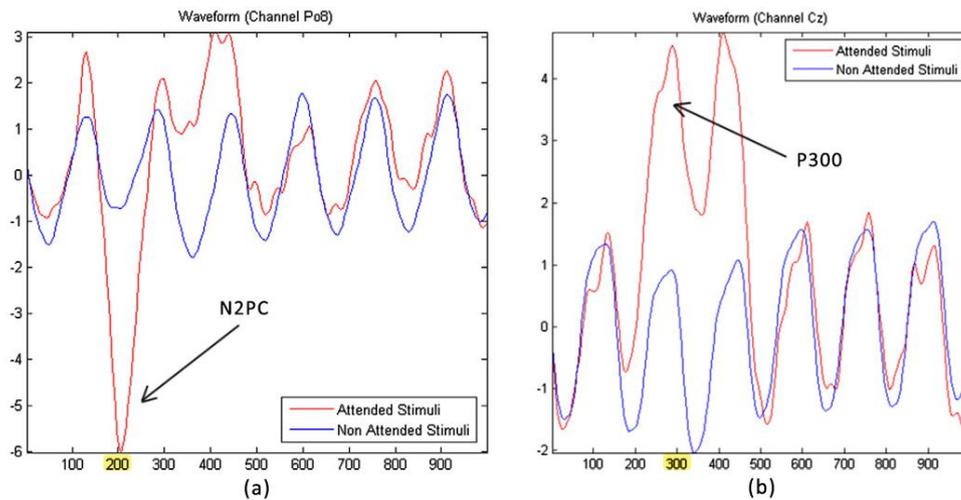


Figura 2-23. Potenciales recogidos en el estudio de Sirvent Blasco *et al* [23]. En (a) se muestra el potencial evocado N2PC, producido unos 200ms después del estímulo y recogido sobre el electrodo PO8 (zona parietal-occipital). En (b) se muestra el potencial evocado P300, producido unos 300ms después del estímulo y recogido sobre el electrodo Cz.

Con el fin de detectar los potenciales evocados con la mejor resolución posible, se compararon la señal obtenida en cada electrodo cuando no se atendía al estímulo y la señal obtenida cuando sí se atendía al mismo, ilustrado en la Figura 2-23.

Para la detección y el tratamiento de las señales EEG se usaron 16 electrodos conectados a un amplificador de 16 canales. Como resultado, la mayor diferencia entre las señales se produjo en la zona parietal/occipital del córtex y se concluyó que la contribución de los potenciales N2PC es más fuerte que la de los potenciales P300.

Las señales se digitalizaron con una frecuencia de muestreo de 256Hz y 8 bits por muestra. Se aplicó un filtro paso banda y un filtro *notch* (filtro elimina-banda muy estrecho en 50 o 60 Hz) para evitar interferencias electromagnéticas con la red eléctrica. Adicionalmente, se aplicó un filtro temporal que consideró solamente los primeros 600ms posteriores a cada estímulo con el fin de evitar patrones repetitivos y artefactos.

Posteriormente, se diezmaron las señales a 20Hz y se obtuvieron 192 muestras, de las cuales se seleccionaron las 60 más significativas, consiguiendo una media de 4 muestras por canal. Para el proceso de traducción de características se utilizó el clasificador SWLDA (*StepWise Linear Discriminant Analysis*, o Análisis Discriminante Lineal Paso-a-paso).

La interfaz del navegador se compone de dos partes: la navegación y el control. La ventana de navegación ocupa 2/3 de la parte vertical de la pantalla y el control el 1/3 restante. En la ventana de control pueden aparecer tres matrices implementadas con la técnica *odd-ball* según las intenciones del usuario: el teclado, el ratón y el control de distancia.

El teclado virtual posee una matriz que contiene los caracteres básicos: letras, números y símbolos necesarios para navegar satisfactoriamente. La matriz del ratón contiene las distintas direcciones a tomar, las acciones del mismo (*clicks* derecho e izquierdo, *scrolls*) y el botón del modo lectura. Para terminar, el control de distancia ofrece nueve opciones para establecer la distancia en píxeles que debería recorrer el cursor del ratón hacia la dirección seleccionada. La máquina de estados que gobierna las transiciones entre las tres matrices se muestra en la Figura 2-24.

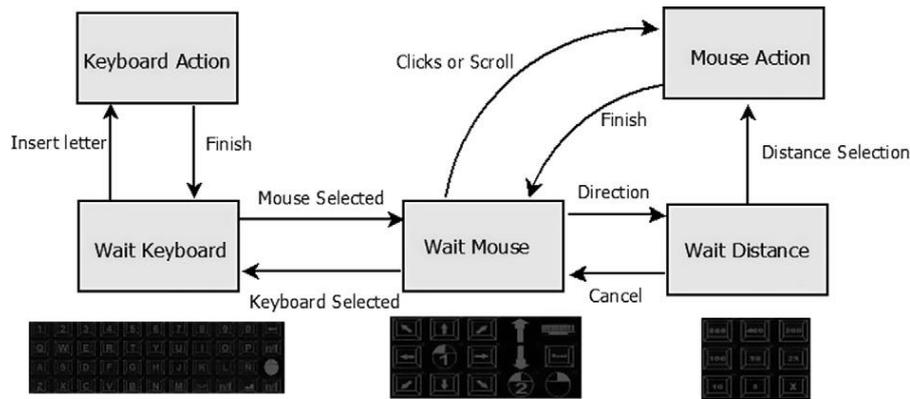


Figura 2-24. Máquina de estados que gobierna las transiciones entre las matrices del navegador web

Inicialmente, el navegador muestra el teclado virtual. A través del mismo se puede acceder a la matriz del ratón. Si la opción seleccionada es la de moverse a través de una dirección determinada, aparecerá a continuación la matriz del control de distancias.

La aplicación se evaluó por parte de cuatro usuarios sanos. Antes de la realización de los experimentos se tuvo que llevar a cabo una sesión de entrenamiento para establecer los coeficientes del clasificador SWLDA de cada usuario. En esta sesión se les ordenó escribir una serie de palabras con el teclado virtual.

En el experimento se ordenó a los usuarios realizar una búsqueda en *Google* para poner a prueba su capacidad de control del movimiento del ratón e introducción de caracteres con el teclado. El tiempo medio en finalizar el experimento fue de 4.20min y la precisión al introducir caracteres con el teclado virtual fue del 93% [23]. También se aprecia que los resultados en el segundo intento fueron significativamente mejores, lo cual indica que los usuarios se adaptan rápidamente al sistema.

Se concluye que los resultados obtenidos teniendo en cuenta ambos potenciales (P300 y N2PC) son mejores que utilizando solamente uno de ellos. No obstante, se ha demostrado que es posible implementar una aplicación utilizando únicamente potenciales N2PC, normalmente ignorados.

2.6.4. Yu *et al* — “Surfing the Internet with a BCI Mouse” [24]

Este artículo se centra en aproximar la navegación web a través de BCI a la navegación web con un ordenador común mediante la implementación de un cursor controlado simultáneamente con potenciales evocados P300 y ritmos sensoriomotores μ y β . También se ha diseñado un filtro capaz de ignorar los hipervínculos no deseados. Gracias a este sistema, los usuarios se desenvuelven rápidamente en un entorno con multitud de hipervínculos filtrando los no deseados y seleccionando el objetivo con el cursor.

La posición del ratón se actualiza cada 200ms y los movimientos verticales y horizontales son independientes entre sí. El movimiento vertical se controla a través de potenciales evocados P300 y el horizontal a través de los ritmos μ y β .

En la interfaz del usuario aparecen ocho botones que se iluminan aleatoriamente, destinados a evocar los potenciales P300. Tres de ellos se encuentran en la parte superior de la pantalla (UP), otros tres de ellos en la parte inferior (DOWN) y dos de ellos en los lados (STOP), los cuales provocan que el cursor suba, baje o pare, respectivamente. En cada actualización del ratón se tiene en cuenta la dirección tomada en la muestra anterior para no interrumpir el movimiento del cursor. El

movimiento vertical solamente podrá interrumpirse cuando el usuario se centre en el botón de *STOP*. La interfaz del usuario se muestra en la Figura 2-25.

Una vez que el usuario ha seleccionado el movimiento vertical del ratón con los potenciales P300 deberá controlar el movimiento horizontal del mismo con los ritmos sensoriomotores, es decir, imaginando el movimiento de su mano derecha o izquierda, según sea la dirección que quiera tomar.

Dado que los botones continúan iluminándose mientras el usuario lee la página web o mientras controla el movimiento horizontal del cursor, se introduce un umbral para la detección de los potenciales evocados P300. Si se supera este umbral indicará que el usuario está prestando atención expresamente a uno de los botones, y en consecuencia se activará la función del botón seleccionado. Sin embargo, si no se supera este umbral se considera que el usuario no está prestando atención al mismo, y no se ejecutará ninguna acción.

A la hora de encontrar un hipervínculo, éste aparecerá en el centro de una caja cuando el cursor pase sobre el mismo. Si el usuario desea seleccionarlo, deberá prestar atención al botón izquierdo de *STOP*. No obstante, si el usuario desea rechazarlo, bastará con seguir controlando los ritmos μ y β imaginando el movimiento de una de las manos, causando que el cursor abandone el hipervínculo.

En numerosas páginas web el número de hipervínculos es extremadamente alto. Para facilitar la navegación a través de todos ellos, se introduce un filtro en la parte superior derecha de la pantalla. Introduciendo unas palabras clave, se filtrarán todos los hipervínculos y solamente podrán seleccionarse aquellos que las contengan. Para introducir estas palabras clave aparecerá una nueva ventana con una matriz de letras y símbolos (*P300-Based Speller*) que se iluminarán aleatoriamente para poder ser controlados a través de potenciales evocados P300, mostrada en la Figura 2-26.

La página inicial se compone de los ocho hipervínculos más usados por el usuario, facilitando así la tarea de introducir URLs comunes a través de la matriz de letras y números y reduciendo el tiempo utilizado en la navegación. Estos hipervínculos aparecen acompañados de una imagen previa de la página web y están lo suficientemente separados entre sí como para que el usuario pueda seleccionarlos rápidamente y tener que rechazar el menor número posible.

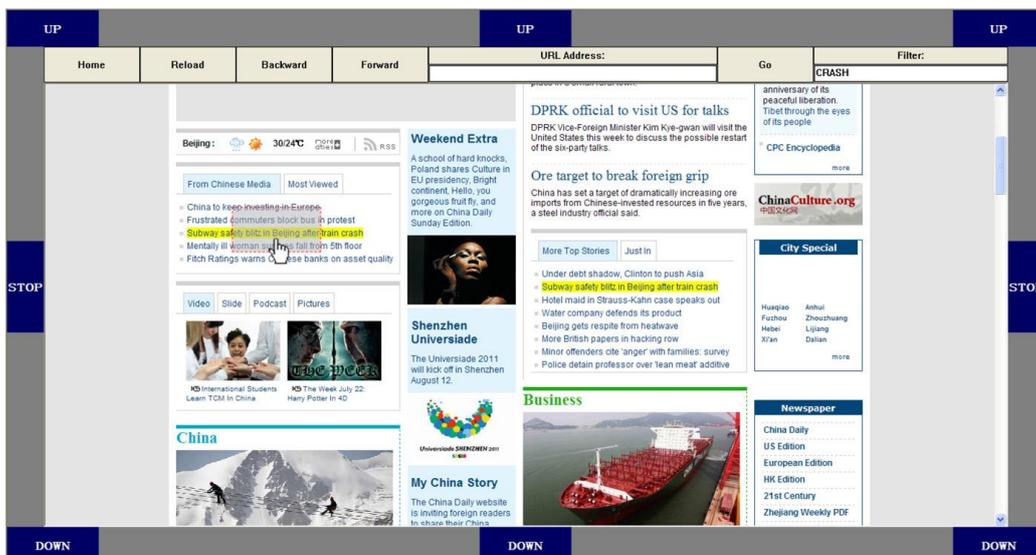


Figura 2-25. Interfaz gráfica del navegador web desarrollado por Yu *et al* [24]. Se puede observar cómo se muestra un rectángulo a medida que el cursor se acerca al hipervínculo.



Figura 2-26. Método de selección de letras, números y comandos implementado con potenciales evocados P300 y usado en el navegador de Yu *et al* [24].

Antes de la realización de los experimentos se llevaron a cabo tres sesiones de entrenamiento. La primera de ellas estuvo dedicada a practicar con el *P300-Based Speller*, la segunda estuvo centrada en practicar la detección de los ritmos μ y β imaginando el movimiento de las manos y, como consecuencia, a establecer los parámetros necesarios para una correcta detección y, finalmente, la tercera sesión practicó la selección, o bien el rechazo, de los hipervínculos.

En el primer experimento se mostró al usuario 10 esferas colocadas aleatoriamente en la pantalla. Nueve de ellas eran verdes y solamente una era de color rojo. El usuario tenía que conseguir seleccionar la esfera roja y rechazar todas las esferas verdes que encontrase en el camino. Los resultados, para siete sujetos sanos, muestran que la media de precisión para este experimento de entre todos los usuarios fue un 93.21%, el tiempo medio en cumplir el objetivo fue de 32.71 segundos y el tiempo medio en decidir si se selecciona o rechaza un objetivo fue de 1.25 segundos.

En el segundo experimento se ordenó a los usuarios que navegasen a través de una página web y utilizarasen el filtro de hipervínculos para seleccionar un vínculo de interés. Para una media de 6.8 caracteres introducidos en el filtro, se tardó una media de 92.26 segundos en utilizar el *P300-Based Speller*. La media de selecciones fue de 3.2 y el tiempo total en terminar el experimento fue de 247.08 segundos (~4min).

Estos resultados dan a entender que el navegador BCI es aceptable para todos los usuarios. Sin embargo, los usuarios demandaron una mejora en la velocidad del sistema.

2.6.5. Liu *et al* — “An Online Brain-Computer Interface Using Non-Flashing Visual Evoked Potentials” [25]

En este estudio se presenta un navegador web controlado por BCI basado en potenciales mVEP (*Motion-onset Visual Evoked Potentials*, o Potenciales Evocados Visuales de Movimiento). Este procedimiento difiere un poco de la técnica tradicional *odd-ball*, donde ciertas matrices de selección se iluminaban aleatoriamente.

Con el fin de reducir la fatiga visual del usuario y aumentar su comodidad, se utiliza una técnica en la que las selecciones no se iluminan constantemente. También destaca como objetivo el utilizar el menor número de canales posibles para detectar la respuesta a los estímulos.

Para empezar, se realiza un análisis *offline* que pretende investigar el patrón espacial para ubicar la señal óptima de los mVEPs, y en consecuencia, minimizar el número de canales necesarios. La interfaz gráfica presentada consta de seis botones virtuales que contienen un cursor vertical, tal y como se muestra en la Figura 2-27.

Este cursor aparece en el borde derecho de cada botón y se desplaza hacia la izquierda para generar el estímulo. El cursor se mueve dentro de cada botón durante 100ms y el intervalo de tiempo entre dos intentos es de 500ms. Teniendo en cuenta que un intento concluye cuando el cursor aparece y desaparece en todos los botones, la duración del mismo será de unos 1100ms. Los usuarios deben fijarse en un botón objetivo durante 15 intentos para obtener un bloque de datos. Con el fin de aumentar la concentración de los usuarios, se pide que cuenten el número de veces que el cursor aparece en el botón deseado.

A continuación se realiza un análisis *online* para evaluar el sistema mVEP y la aplicación web. En la evaluación del sistema se utiliza la misma interfaz gráfica que en el análisis *offline*, sin embargo, esta vez se establece una tarea. Se ordena a los usuarios introducir una secuencia con letras proporcionadas. El usuario recibe una realimentación cuando se haya seleccionado una opción, iluminándose durante 3s.

Con respecto a la aplicación web, el teclado virtual mostrado en la Figura 2-28 aparece cada vez que sea necesario rellenar un formulario. Para reducir la complejidad, la página de *Google* es la versión para móviles. Gracias a esta interfaz del teclado virtual, cada selección se realiza en dos pasos: el primero selecciona el conjunto de 6 caracteres y el segundo se encarga de seleccionar uno de los caracteres del grupo anterior. Adicionalmente, el carácter “<” sirve para retroceder o eliminar un carácter y el “>” permite buscar y seleccionar el hipervínculo deseado.

Para determinar la detección óptima de la señal se utilizaron 30 electrodos y se estableció una ventana en las señales obtenidas entre 100ms y 500ms para seguir el potencial mVEP. También se utilizó un clasificador SVM como algoritmo de traducción. Como resultado, el electrodo P3 consiguió el mayor valor de r^2 en torno a 195ms después del estímulo. Utilizando solamente este canal se obtuvo una precisión del sistema del 84%, sin embargo, utilizando seis canales se mejoró considerablemente el resultado, obteniendo una precisión del 93%.

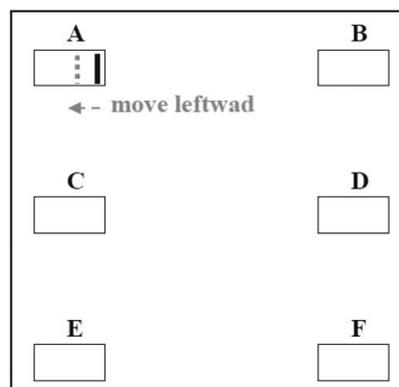


Figura 2-27. Interfaz de usuario para el análisis *offline* en el estudio de Liu *et al* [25] donde se prueba la eficacia de los mVEP.

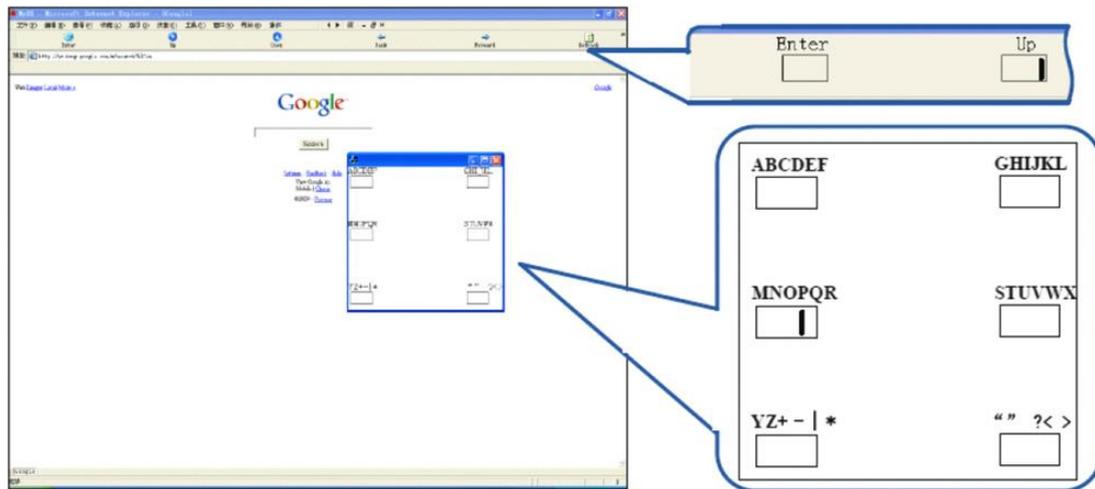


Figura 2-28. Interfaz de usuario para el análisis *online* en el estudio de Liu *et al* [25] donde se estudia el rendimiento de la aplicación web.

Se ha demostrado que el tiempo de entrenamiento necesario para configurar el clasificador utilizando mVEP no sobrepasa los 5 minutos. También se ha detectado que las amplitudes de los potenciales son mayores en la evaluación *online* con respecto a la evaluación *offline*, debido a que en la primera evaluación los usuarios estaban más motivados a cumplir una meta.

La realimentación instantánea ayudó a los usuarios a mejorar su control sobre el sistema, convirtiéndolo en un sistema fácil de manejar. Los doce sujetos sanos fueron capaces de navegar libremente. Dado que este artículo más que presentar un navegador web presenta una técnica innovadora para poder controlar una aplicación libre de fatigas visuales, no se incluirá en la comparativa por falta de información.

2.6.6. Comparativa de los navegadores web

Con el transcurso de los años se han empezado a sustituir los potenciales SCP por los potenciales evocados P300 como señales de control. En los últimos artículos, además de los potenciales evocados P300 también se utilizan los potenciales evocados N2PC, los ritmos sensoriomotores y los potenciales mVEP, lo cual produce grandes mejoras de rendimiento.

Con respecto al entrenamiento, todos los navegadores web desarrollados en estos artículos necesitan entrenamiento por parte del usuario, ya sea para controlar correctamente la aplicación (potenciales SCP o ritmos μ y β) o para establecer diversos coeficientes que puedan mejorar su experiencia con la misma (potenciales P300, N2PC y mVEP).

Sin embargo, atendiendo a las funcionalidades, se encuentran grandes diferencias.

- i. La selección de hipervínculos en los dos primeros artículos estaba centrada en buscar el deseado entre una lista. En Bensch *et al* [19] se utiliza una selección binaria para encontrarlo a base de iteraciones y en Mugler *et al* [22] es necesario seleccionarlo introduciendo los caracteres de su etiquetado. En cambio, en los últimos artículos, Sirvent Blasco *et al* [23] y Yu *et al* [24], la selección de hipervínculos se lleva a cabo desplazando un cursor sobre los mismos.

- ii. El filtrado de hipervínculos solamente está disponible en Yu *et al* [24]. Este filtro y su método de selección de hipervínculos confieren al navegador una rapidez que no poseen sus predecesores.
- iii. Solamente Bensch *et al* [19] requiere supervisión. Varias de sus funcionalidades, como el tiempo de lectura o la lista de direcciones de correo, son gestionadas por el supervisor.
- iv. El historial de navegación está disponible en Sirvent Blasco *et al* [23]. No obstante, no está orientado a personas discapacitadas debido a la dificultad para su uso. El resto de navegadores no incluyen esta funcionalidad.
- v. Todos los navegadores implementan la funcionalidad de los marcadores excepto Mugler *et al* [22]. Sin embargo, Sirvent Blasco *et al* [23] no lo orienta a personas discapacitadas y Yu *et al* [24] simplemente almacena los ocho más populares en su página de inicio, no permitiendo su modificación.
- vi. A diferencia de los primeros navegadores web implementados con BCI, como *Descartes*, todos los navegadores de esta comparativa poseen una barra para introducir URLs.
- vii. Todos los navegadores exceptuando Yu *et al* [24] implementan un modo de lectura, aunque su metodología hace que no lo necesite. En Bensch *et al* [19] el supervisor debe introducir el tiempo que el usuario puede estar leyendo una página, sin embargo, en Mugler *et al* [22] y Sirvent Blasco *et al* [23] este modo es controlado plenamente por el usuario.
- viii. Finalmente, tanto Sirvent Blasco *et al* [23] como Yu *et al* [24] implementan un ratón que es capaz de mover un cursor. No obstante, en Sirvent Blasco *et al* [23] el desplazamiento es más rígido y es necesario introducir la distancia en píxeles a recorrer, mientras que en Yu *et al* [24] no es necesario.

En términos de precisiones, Yu *et al* [24] consigue la mayor precisión con un 93.21%, seguido por Sirvent Blasco *et al* [23] y Liu *et al* [25] con 90% y, finalmente, por Mugler *et al* [22] con 90% para los sujetos sanos y 72% para los pacientes con ALS. No obstante, cada uno de estos estudios utilizó un distinto número de sujetos. Por tanto, las precisiones obtenidas podrían variar si todos los estudios utilizaran el mismo número de sujetos sanos y de pacientes.

Esta comparativa aparece en la Tabla 2-3 a modo de resumen. A continuación se muestran una serie de artículos que se limitan a desarrollar otras técnicas que pueden implementarse como mejora en cada uno de los navegadores web.

En Liu *et al* [25] se desarrolla un sistema basado en potenciales evocados mVEP. Mediante el desplazamiento de un cursor se evocan los potenciales y, en consecuencia, no es necesario utilizar botones que se iluminan constantemente. Con respecto a los artículos comparados anteriormente, reduce drásticamente la fatiga visual producida por los cambios de brillo de la pantalla, convirtiéndose en una alternativa innovadora.

Yin *et al* [26] determinó el brillo óptimo para no dañar a los ojos cuando se utilizan los potenciales evocados visuales P300 mediante la técnica *odd-ball*, algo a tener en cuenta. En el experimento se concluyó que la luminancia RGB debe encontrarse entre 160 y 180. La fórmula de la luminancia es la siguiente [27].

$$Y = 0,21246 \cdot R + 0,7152 \cdot G + 0,0722 \cdot B, \quad \text{con } R, G, B, Y \in [0,255] \quad (2.1)$$

En Kübler et al [28] se desarrolla un sistema de deletreo basado en potenciales evocados P300 auditivos. Los artículos basados anteriormente se basaban en una capacidad visual ordinaria, sin embargo, muchos de los pacientes con ALS han perdido el control de los músculos oculares y, por tanto, no podrían utilizar dichos sistemas. Este sistema posee la ventaja de poder ser usado por personas con una capacidad visual reducida, a diferencia del resto. Sin embargo, la precisión obtenida (23.53%) en este sistema es muy inferior al resto. También es notable que las personas con capacidad visual reducida obtengan mejores precisiones que las sanas, debido a un desarrollo superior de su audición.

TABLA 2-3. COMPARATIVA DE LOS DISTINTOS ARTÍCULOS SOBRE NAVEGADORES WEB ATENDIENDO A SU METODOLOGÍA Y FUNCIONALIDADES.

Artículo	Basado en	Funcionalidades										Sujetos	Precisión
		Requiere entrenamiento	Requiere supervisión	Selección de hipervínculos	Filtrado de hipervínculos	Marcadores	Historial	Barra de URL	Modo de lectura	Implementa ratón	Distancias mediante píxeles		
<i>Bensch et al</i> (2007) [19]	Potenciales Corticales Lentos (SCP)	Sí	Sí	Mediante selección binaria	No	Sí	No	Sí	Tiempo impuesto por el supervisor	No	No	4 ALS	No especificado
												2 sanos	
<i>Mugler et al</i> (2010) [22]	Potenciales Evocados P300	Sí	No	Etiquetado con uno o dos caracteres	No	No	No	Sí	Sí	No	No	3 ALS	72%
												10 sanos	90%
<i>Sirvent Blasco et al</i> (2012) [23]	Potenciales Evocados P300 y N2PC	Sí	No	Mediante desplazamiento del cursor	No	No orientado a personas discapacitadas	No orientado a personas discapacitadas	Sí	Sí	Sí	Sí	4 sanos	93%
<i>Yu et al</i> (2012) [24]	Potenciales Evocados P300 y ritmos sensorio-motores	Sí	No	Mediante desplazamiento del cursor	Sí	Ocho marcadores populares en la página inicial	No	Sí	No	Sí	No	7 sanos	93.21%
<i>Liu et al</i> (2010) [25]	Potenciales Evocados mVEP	Sí	No	No especificado	No	No especificado	No especificado	Sí	No especificado	No	No	12 sanos	93%

ANÁLISIS DE LA SEÑAL Y POTENCIALES EVOCADOS P300

3.1. Potenciales evocados P300

Tal y como se ha explicado de manera breve en el anterior capítulo, un potencial evocado P300 se corresponde con una deflexión positiva del EEG del usuario provocada como respuesta natural del cerebro ante un estímulo [5]. Se localiza en torno a la zona centro-parietal del córtex (Fz, Cz y Pz), su amplitud se encuentra en torno a los $10\mu\text{V}$ y su nombre deriva del hecho de que su latencia con respecto al inicio del estímulo se encuentra en torno a 300ms. También se conoce como potencial P3 puesto que es el pico con la tercera amplitud más elevada entre los potenciales evocados sensoriales [29].

La manera más habitual de provocar un potencial evocado P300 se fundamenta en aplicar el paradigma *odd-ball*, consistente en presentar una serie de estímulos objetivo de forma infrecuente camuflados entre otros estímulos no deseados mucho más frecuentes. Está demostrado que a menor probabilidad de evento mayor será el potencial P300 generado.

Numerosos estudios han tratado de caracterizar este potencial y de establecer las pautas óptimas para su obtención. Se ha declarado que la utilización de filtros paso-alto para eliminar las variaciones lentas de la señal influye en la forma del potencial, siendo conveniente aplicar un filtro a partir de 0.1Hz (frecuencias de corte mayores distorsionarían la forma de la señal).

La respuesta de los potenciales P300 es pequeña y aparece superpuesta con la actividad EEG de fondo, así como con ruido, y debido a ello, una respuesta individual no suele ser reconocible. No obstante, resulta fundamental poder distinguir la actividad evocada por el estímulo de la actividad EEG de fondo. En este sentido conviene mencionar que la señal P300 está solapada en frecuencia con el contenido espectral del EEG de fondo, por lo que no es posible aplicar un filtrado en frecuencia para aislar la respuesta evocada. Como alternativa, comúnmente se realiza un filtrado en el dominio del tiempo aprovechando la relación temporal, repetitiva a lo largo de las diferentes épocas de la señal obtenida, entre la aparición del estímulo y la actividad evocada, típicamente a 300ms desde la presentación del estímulo. Este filtrado temporal se conoce como promediado sincronizado, y consiste en promediar todas las épocas de la señal estableciendo como referencia el punto en el que se ha generado el estímulo que produce el potencial P300 [30]. En la Figura 3-1 se observa el resultado de promediar la señal a la hora de detectar los potenciales P300.

Dentro de la parte de adquisición de la señal, la eliminación de artefactos, y en concreto la eliminación movimiento ocular, es una etapa muy importante a la hora de analizar la señal. Para eliminar el movimiento ocular y los pestañeos producidos por el usuario que degradan los potenciales P300 se pueden utilizar dos enfoques distintos, explicados a continuación [29].

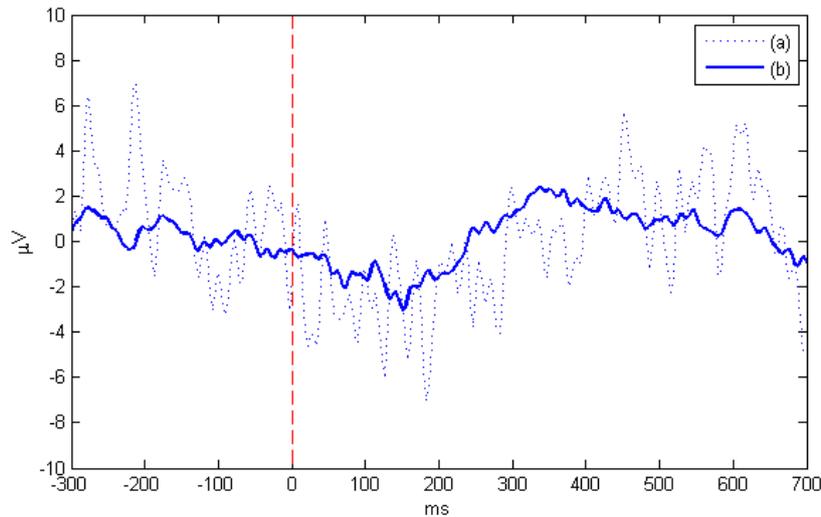


Figura 3-1. Potencial evocado P300 recogido en el electrodo Pz y generado con un *odd-ball* auditivo donde el usuario debía detectar tonos objetivos con probabilidad de ocurrencia de 0,2. La curva punteada (a) se corresponde con la primera época registrada y la curva (b) se corresponde con un promediado sincronizado de 90 épocas, permitiendo distinguir el potencial. La línea roja rayada indica el momento en el que se presentó el estímulo.

El primero se basa en rechazar las épocas donde se hayan producido movimientos oculares o pestañeos. Para ello se monitoriza el EOG posicionando electrodos cerca de los ojos (típicamente encima y debajo de ellos) y se rechaza cada época si la señal EOG supera cierto umbral (e.g., $\pm 100 \mu\text{V}$). Sin embargo, este enfoque posee dos inconvenientes principales: el primero reside en que rechazar épocas de la señal disminuye la eficiencia del registro (en el peor caso posible todas las épocas serían rechazadas) y, por tanto, se recomienda a los sujetos no mover los ojos o pestañear mientras se realizan las pruebas; esto conduce al segundo inconveniente, puesto que recomendar a los sujetos no producir movimientos oculares cambia la naturaleza de la prueba, causando que los sujetos dividan su atención entre sus ojos y el estímulo, produciendo un cambio en la forma del potencial P300.

El segundo enfoque se basa en sustraer la señal EOG del EEG. Para ello se suelen emplear filtros adaptativos, los cuales requieren una señal de referencia fuertemente correlada con el ruido o artefacto que se desea eliminar (en este caso sería la señal EOG extraída con los electrodos situados en torno al ojo). Desafortunadamente esta extracción no es perfecta y puede eliminar ciertas porciones de la señal EEG recogida en los electrodos frontales, más cercanos a los ojos.

La detección de los potenciales evocados P300 comúnmente se lleva a cabo detectando un pico de amplitud que se eleva de la actividad basal, el cual se encuentra retrasado 200 o más milisegundos después de la presentación del estímulo. Esta detección se realiza en un solo electrodo, típicamente Cz o Pz, y la latencia del P300 varía de un electrodo a otro, alcanzando el valor mínimo en los electrodos situados en la zona frontal del córtex [29].

Sin embargo, esta manera de tratar a los potenciales P300 no tiene en cuenta la posibilidad de que éstos estén conformados por múltiples procesos generados en distintas localizaciones. Un estudio más detallado del potencial P300 revela que éste está compuesto de tres ondas positivas superpuestas e independientes: P3a con un máximo en torno a 250ms, P3b con un máximo en torno a 350ms y una onda lenta (SW, *Slow Wave*) [31]. Estas componentes se observan en la Figura 3-2.

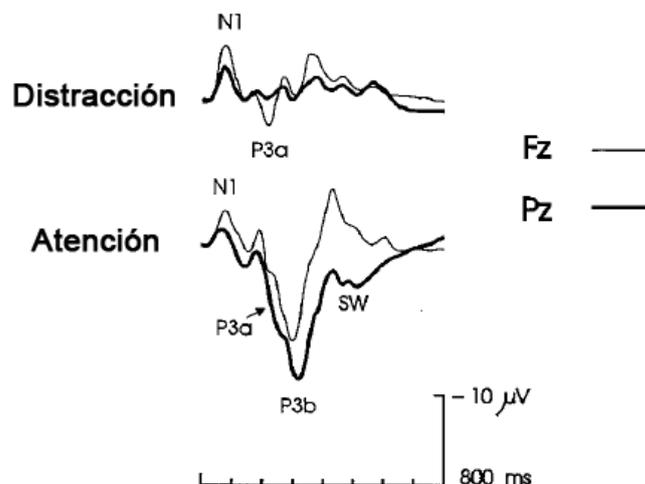


Figura 3-2. Componentes del potencial evocado P300 en dos localizaciones distintas: Fz y Pz. En la gráfica superior aparece la respuesta cuando el sujeto no atiende al estímulo, donde se observa que la onda P3a sigue apareciendo. En la gráfica inferior, por el contrario, aparece la respuesta cuando el sujeto atiende al estímulo, generando las tres componentes principales: P3a, P3b y *slow wave* [29].

Estas componentes se consideran independientes porque poseen distintas relaciones con diversas variables experimentales. La onda P3a se distribuye en la zona frontal del córtex, mientras que la onda P3b se detecta mejor en la zona parietal. Aunque las tres componentes varían de la misma manera con la probabilidad de estímulo (aumentando su amplitud a medida que el objetivo objetivo se hace más improbable), son sensibles a la información obtenida sobre el estímulo en cuestión: la onda P3a no se afecta si el sujeto está atendiendo o no al estímulo, mientras que la onda P3b y la onda lenta aumentan su amplitud si el sujeto pone atención, tal y como se puede observar, de nuevo, en la Figura 3-2. Si fuese necesario identificar las tres componentes por separado se podría realizar un PCA (*Principal Component Analysis*, o Análisis de Componentes Principales).

A continuación se muestra cómo varía la forma del potencial evocado P300 según las características del sujeto, atendiendo a sus aspectos psicológicos, diferencias individuales entre sujetos, los factores farmacológicos y las enfermedades neuronales.

3.1.1. Aspectos psicológicos

Dentro de los aspectos psicológicos del sujeto cobran importancia la atención del sujeto y la probabilidad de que ocurra el estímulo que generará el P300. Numerosos estudios han demostrado que la amplitud del potencial P300 aumenta de manera significativa cuando el usuario aumenta su nivel de atención y cuanto menor sea la probabilidad de aparición del estímulo objetivo. En la Figura 3-3 (a) se muestra el efecto de la probabilidad del estímulo en el P300.

Además, los efectos de la probabilidad del estímulo ocurren independientemente del conocimiento sobre ellos del sujeto. Si se modifica la probabilidad del estímulo durante un experimento y el usuario no es consciente de ello, la amplitud del potencial P300 cambia igualmente [29].

Aunque esté fuertemente correlado con la probabilidad de estímulo, otro aspecto importante es la probabilidad temporal, es decir, el tiempo que pasa entre la aparición de un estímulo objetivo y el siguiente. Se ha demostrado que cuanto mayor sea este tiempo mayor será la amplitud obtenida en el potencial evocado P300, tal y como muestra la Figura 3-3 (b).

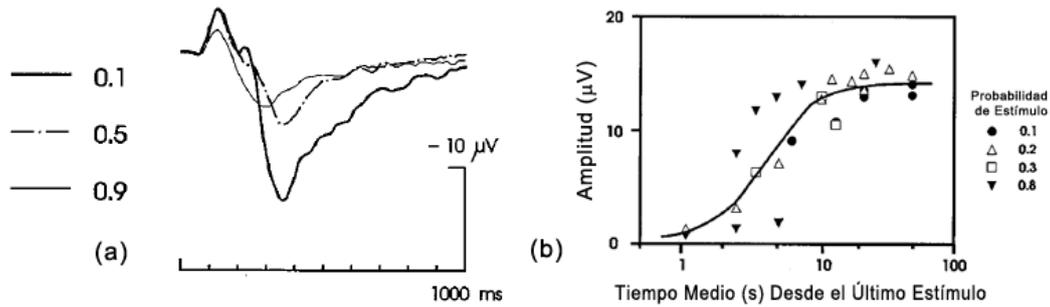


Figura 3-3. Variación de la forma del P300 para el electrodo Pz en función de la probabilidad de estímulo (a) y la probabilidad temporal (b) al utilizar un paradigma *odd-ball* auditivo [29].

Para terminar, también influye la dificultad de la tarea en la forma del P300 obtenido. Cuando discriminar el estímulo objetivo del resto de estímulos se convierte en una tarea ardua, la amplitud del potencial P300 disminuye y su latencia aumenta, tal y como se puede observar en la Figura 3-4.

La disminución de amplitud es producida por una reducción de la confianza del usuario. La amplitud es mayor cuando éste se encuentra confiado porque la tarea es sencilla, sin embargo, al incrementarse al dificultad de la tarea la confianza en sí mismo disminuye, causando que la amplitud del potencial P300 caiga. No obstante, si la tarea se vuelve demasiado sencilla, la amplitud puede disminuir también, puesto que el usuario no dedicará toda su atención [29].

3.1.2. Diferencias individuales entre sujetos

También afectan a la forma del potencial P300 las diferencias individuales de cada sujeto, siendo éstas la edad, la inteligencia, la personalidad o el oído absoluto, entre otras. La edad de los sujetos posee dos comportamientos distintos según el rango que se analice.

Para sujetos adultos con más de 20 años, aproximadamente, la latencia del potencial P300 muestra un comportamiento lineal con la edad provocando un aumento de unos 1.3ms por año (con un error estándar de 31ms), tal y como muestra la Figura 3-4 (a) [29]. Analizando el tiempo de reacción (RT, *Reaction Time*) del sujeto, se ha demostrado que éste se encuentra en torno a 50ms después de la generación del P300. Sin embargo, el RT puede cambiar con la edad dependiendo de la dificultad de la tarea: si la tarea es sencilla, la edad aumenta más la latencia del P300 que el RT, mientras que si la tarea es complicada, ocurre el fenómeno contrario.

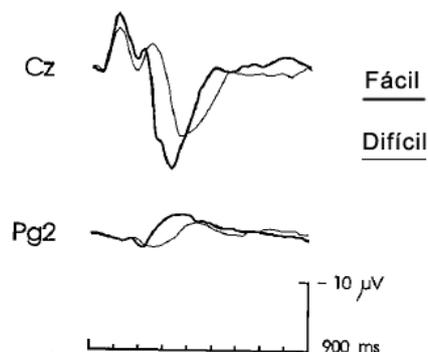


Figura 3-4. Potenciales evocados P300 recogidos en el vértex (Cz) y en un electrodo nasofaríngeo (Pg2) según la dificultad de discriminación de los estímulos objetivos [29].

En general también puede afirmarse que la amplitud del P300 decrece con la edad en sujetos adultos, además de modificarse la localización del potencial, ubicándose más en la zona frontal del córtex a medida que se envejece [29].

No obstante, el comportamiento del potencial P300 varía de forma distinta para los niños (entre 5 y 20 años) que para los adultos. Determinar el comportamiento del potencial P300 en niños es complicado debido al gran número de artefactos que presentan y a que normalmente no son capaces de mantener constante su atención.

El potencial evocado P300 para los niños se solapa con otra onda negativa producida en la zona frontocentral del córtex, más intensa cuanto más pequeño sea el niño. Sin embargo, en la zona parietal del córtex sí se muestra una onda parecida al P300 de los adultos, caracterizada por tener mayor latencia. Esta latencia disminuye a medida que aumenta la edad del niño, pudiendo distinguir dos tramos principales: entre los 5 y 12 años de edad la latencia disminuye con una tasa de 25ms por año, mientras que entre los 12 y 20 años de edad disminuye más lentamente, con una tasa de 1-5ms por año [29]. Estos efectos pueden observarse en la Figura 3-5.

Adicionalmente, la distinción entre la onda P3b y la SW en niños se convierte en una tarea ardua y confusa. Con respecto a la amplitud, ésta aumenta a medida que aumenta la edad del niño hasta los 13 años, aproximadamente. Después de ese punto la amplitud decrece levemente como en los adultos. Los efectos de la probabilidad del estímulo en los adultos son similares a los encontrados en niños.

Para terminar, cabe destacar que la localización del potencial P300 en el córtex para los niños difiere de la localización de los adultos. A medida que los niños crecen, la localización del mismo se desplaza desde la zona parietal del córtex a la zona frontocentral. Además de ello, la ubicación de los potenciales P300 visuales y auditivos en los adultos es similar, sin embargo, en los niños se obtienen mejor los potenciales P300 visuales en las zonas frontocentrales que los potenciales P300 auditivos [29].

Dejando a un lado la edad del sujeto, también se ha estudiado la variación de la forma del potencial P300 según la inteligencia o personalidad del sujeto. Considerando la inteligencia como el coeficiente intelectual (CI) clásico, se ha demostrado que a medida que aumenta el CI, disminuyen tanto la latencia como la amplitud del P300.

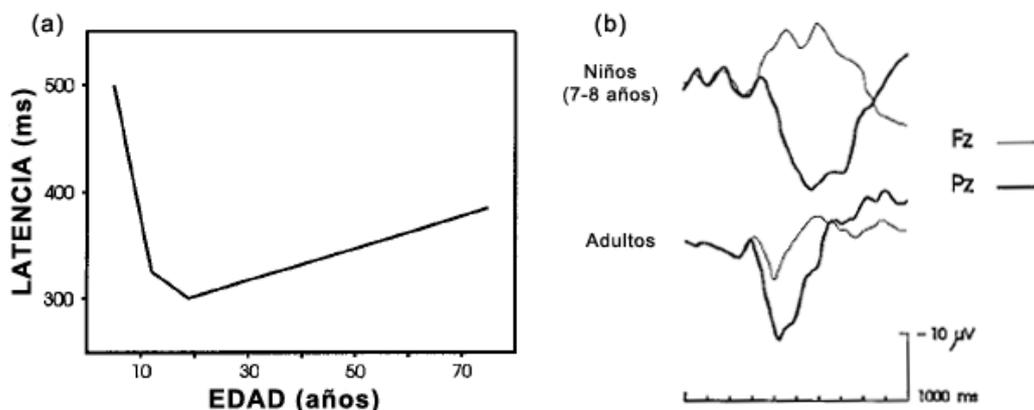


Figura 3-5. Variaciones de la forma del potencial evocado P300 según la edad del sujeto. En (a) se observa la variación de la latencia con la edad, pudiendo distinguir tres tramos distintos: de 5 a 12, de 12 a 20, y de 20 años en adelante. En (b) se observa el potencial P300 y su localización en Fz y Pz para niños de 7-8 años y para adultos, pudiendo observar la onda positiva de los niños en la zona frontal y la latencia del P300 en la zona parietal [29].

También se ha demostrado que la personalidad y, concretamente, la extroversión, el neuroticismo y el psicoticismo (rasgos elegidos por Hans Eysenck para describir la misma), modifican la forma del P300. Los resultados del experimento demuestran que cuanto más extrovertido sea el usuario, menor será el RT medido, posiblemente a causa de la impulsividad característica de éstos sujetos, además de mostrar una disminución de amplitud, posiblemente a causa de la imposibilidad de mantener la atención en largos intervalos de tiempo. También demuestran que el neuroticismo, o inestabilidad emocional, está negativamente correlado con la latencia, es decir, cuanto más inestable sea emocionalmente un sujeto menor será la latencia del P300 recogido. Finalmente, se concluye también que a medida que aumenta el psicoticismo (caracterizado por su agresividad y antipatía) menor será la amplitud del P300 [32].

Finalmente se expone el efecto de una característica concreta e infrecuente en la mayor parte de los sujetos, conocida como oído absoluto. El oído absoluto permite, al sujeto que lo posee, identificar o producir una nota musical concreta sin la ayuda de una nota de referencia, capacidad fuertemente relacionada con la memoria auditiva. Se ha demostrado que los potenciales P300 auditivos generados atendiendo al paradigma *odd-ball* para los sujetos que poseen oído absoluto muestran una amplitud minúscula, posiblemente debido a que los sujetos con oído absoluto no necesitan estar atentos para distinguir un estímulo auditivo infrecuente. Tal y como era de esperar, los resultados para P300 visuales son los mismos que para los sujetos normales [33].

3.1.3. Factores farmacológicos

Cuando el sujeto se encuentra bajo los efectos de ciertos medicamentos, del alcohol o de las drogas, la forma de sus potenciales evocados P300 varía.

Concretamente, los efectos del alcohol producen una reducción de amplitud del potencial P300 y un aumento de su latencia [29]. Sin embargo, este fenómeno no aparece cuando la dificultad de la tarea de discriminación entre los estímulos es complicada, probablemente debido a que el sujeto intenta superar los efectos del alcohol cuando la tarea lo requiere.

Los efectos producidos por las drogas y medicamentos son muy diversos, entre ellos destacan el efecto de la escopolamina (también conocida como *burundanga*), que disminuye drásticamente la amplitud del P300 hasta hacerlo desaparecer en *odd-ball* auditivos debido a la pérdida de memoria a corto plazo; el metilfenidato (psicoestimulante aprobado para el tratamiento de la hiperactividad), que aumenta el RT del sujeto; la clonidina (utilizado como antihipertensivo), las benzodiacepinas (medicamentos psicotrópicos) y los antihistamínicos, que atenúa la amplitud; y la dopamina, que disminuye la latencia; entre otros [29].

3.1.4. Enfermedades neuronales

Las enfermedades neuronales también afectan a la forma del potencial evocado P300, a continuación se tratan los efectos de enfermedades como la esquizofrenia, el autismo, la depresión, el trastorno obsesivo-compulsivo, la psicopatía o la demencia.

Los sujetos que sufren de esquizofrenia generan potenciales P300 con muy baja amplitud, siendo prácticamente irreconocibles, tal y como se muestra en la Figura 3-6. Este fenómeno no parece estar relacionado con la falta de atención o la medicación sino con la severidad de los síntomas del paciente y, adicionalmente, los

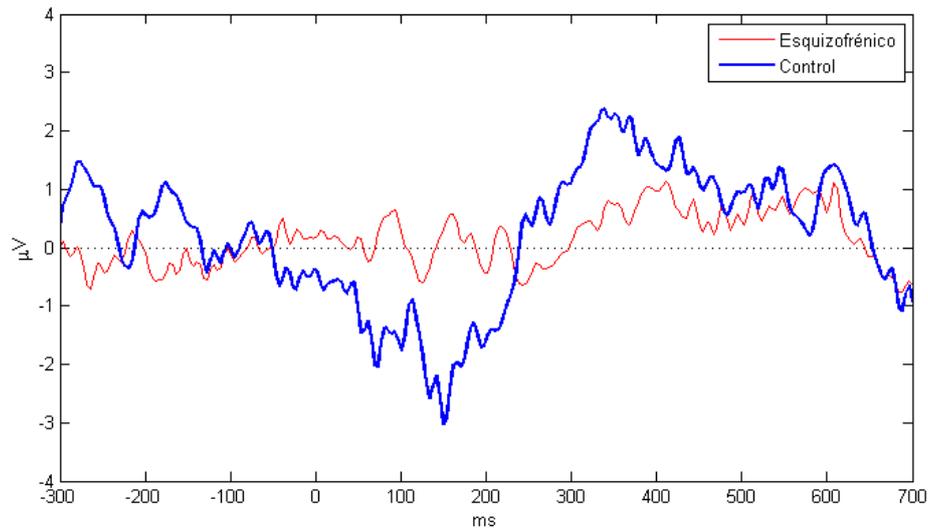


Figura 3-6. Potencial evocado P300 recogido en el electrodo Pz y generado con un *odd-ball* auditivo para un sujeto esquizofrénico (en rojo) y un sujeto sano de control (en azul) con un promediado de 90 épocas.

sujetos esquizofrénicos muestran un potencial P300 menos concentrado en la zona parietal del córtex que los sujetos de control [29].

El autismo y la depresión severa también afectan a la forma del potencial P300, causando una reducción de su amplitud. Sin embargo, en algunos tipos de pacientes psiquiátricos la latencia del P300 puede disminuir, como es el caso de los pacientes que sufren un trastorno obsesivo-compulsivo. Otro fenómeno característico consiste en obtener un potencial P300 mucho más prolongado, típico de los pacientes que sufren psicopatía o ALS. Para terminar, también se ha demostrado que la dislexia y la demencia producen una disminución de amplitud acompañada de un aumento en la latencia del potencial [29].

En la Tabla 3-1 se recogen todas las posibles variaciones del potencial P300 según las distintas características de cada sujeto, explicadas anteriormente.

3.2. Procesado de los potenciales evocados P300

Una vez adquiridos los potenciales P300 y libres de los artefactos comunes tales como el movimiento ocular, el movimiento de los músculos cercanos o la actividad eléctrica del corazón; es necesario procesar la señal para identificar la presencia de los potenciales P300.

3.2.1. Extracción de características

Normalmente, en los sistemas BCI que emplean los potenciales evocados P300 como señal de control, la extracción de características se convierte en una etapa muy sencilla. Tal y como se vio en el capítulo 2, se pueden clasificar las técnicas de extracción de características en dos grandes grupos: filtrado en tiempo o frecuencia y filtrado espacial, como norma general.

Es habitual el utilizar un filtrado de cada tipo en el sistema BCI, sin importar el orden en el que se apliquen. Adicionalmente, la eliminación de artefactos y la extracción de características no están completamente separadas en este tipo de sistemas, pudiendo aplicar un mismo método para ambas. En estas ocasiones los

TABLA 3-1. CARACTERÍSTICAS DE LOS USUARIOS INDIVIDUALES QUE MODIFICAN LA FORMA DEL P300.

Aspectos Psicológicos						
Aumento de →	Atención	Probabilidad de Estímulo	Tiempo entre Estímulos	Sencillez de la Tarea		
Amplitud de P300	Aumenta	Disminuye	Aumenta	Aumenta (*)		
Latencia de P300				Disminuye		
(*) Si las tareas son demasiado sencillas acaba por disminuir						
Diferencias Individuales entre Sujetos						
Aumento de →	Edad		Coeficiente Intelectual	Personalidad		
	Adultos (>20 años)	Niños (5-20 años)		Extroversión	Neuroticismo	Psicoticismo
Amplitud de P300	Disminuye	Aumenta (hasta los 13)	Disminuye	Disminuye		Disminuye
Latencia de P300	Aumenta	Disminuye	Disminuye		Disminuye	
Otros efectos	Se desplaza a la zona frontal	Onda parietal disminuye	(**)	RT disminuye		
(**) El oído absoluto no muestra potenciales P300 auditivos, solamente visuales						
Factores Farmacológicos						
En presencia de →	Alcohol	Escopolamina (burundanga)	Metilfenidato	Clonidina Benzodiacepina Antihistamínicos	Dopamina	
Amplitud de P300	Disminuye	Disminuye (casi desaparece)		Disminuye		
Latencia de P300	Aumenta				Disminuye	
Otros efectos	No afecta en tareas complicadas		RT aumenta			
Enfermedades Neuronales						
En presencia de →	Esquizofrenia	Autismo Depresión severa	Trastorno Obsesivo-Compulsivo	Psicopatía ALS	Dislexia Demencia	
Amplitud de P300	Disminuye (casi desaparece)	Disminuye		Disminuye	Disminuye	
Latencia de P300			Disminuye		Aumenta	
Otros efectos	Menor concentración en la zona parietal			Potencial más prolongado		

métodos de eliminación de artefactos pueden considerarse como técnicas de extracción de características debido a que su salida se introduce seguidamente como entrada del algoritmo de traducción.

A continuación se estudian una serie de métodos de extracción de características de cada grupo que se han empleado con anterioridad en los sistemas BCI y utilizan los potenciales evocados P300 como señal de control. Cabe destacar que la aplicación a desarrollar utilizará dos métodos: como filtrado temporal realizará un promediado sincronizado y como filtrado espacial empleará indistintamente filtros Laplacianos y métodos de referencia de media común (CAR).

a) Filtrado espacial

El objetivo de realizar un filtrado espacial es reducir el difuminado espacial. El difuminado espacial es un efecto provocado por la distancia entre los electrodos y las fuentes que generan la señal dentro el cerebro debido a la no homogeneidad de los tejidos que se encuentran entre ellos.

Se han propuesto diversos métodos para reducir este difuminado espacial, siendo los más populares los filtros Laplacianos, los métodos de referencia de media común (CAR), el análisis de componentes independientes (ICA) y el análisis de componentes principales (PCA) [34].

a.1) Filtro Laplaciano

Para empezar, el filtro Laplaciano calcula, para cada electrodo, la segunda derivada de la distribución de voltaje espacial instantánea y, consecuentemente, enfatiza la actividad originada en un radio concreto alrededor de cada electrodo. De esta manera, en la práctica se comporta como un filtro espacial paso-alto que acentúa la actividad que está más localizada y reduce la actividad más difusa.

A continuación, se halla el valor del Laplaciano para cada electrodo que se va a utilizar, combinando la señal recogida en esas posiciones con las señales de un conjunto de electrodos adyacentes. La distancia desde el electrodo hasta el conjunto de electrodos adyacentes determina las características del filtro: cuanto menor sea la distancia el Laplaciano será más sensible a potenciales con altas frecuencias espaciales y menos sensible a potenciales con bajas frecuencias espaciales, y viceversa. Es decir, funcionará mejor con señales de control muy localizadas en el córtex [35].

Para obtener el Laplaciano, se emplea un método de diferencia finita que aproxima la segunda derivada de la distribución espacial Gaussiana en dos dimensiones del cuero cabelludo en un electrodo, restando la actividad media de los electrodos adyacentes al canal de interés, de acuerdo con la siguiente fórmula [34].

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

Donde $v_n(t)$ es el voltaje recogido por el canal n -ésimo, concretamente h para el canal en el que se aplica el filtrado e i para los canales adyacentes. Adicionalmente, S_i se corresponde con el conjunto de índices de los canales adyacentes y $w_{h,i}$ es el peso en función de la distancia $d_{h,i}$ entre el electrodo de interés y sus vecinos.

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

Por tanto, se habla de Laplaciano corto cuando los electrodos vecinos se corresponden con los electrodos adyacentes al electrodo en el que se aplica el filtrado y de Laplaciano largo cuando los electrodos vecinos se encuentran a una distancia mayor. En la Figura 3-7 (a) y (b) se emplean estos dos tipos de Laplacianos sustrayendo la media de cuatro canales vecinos, indicados con un círculo relleno de la localización central, indicada con un círculo abierto. Por tanto, el peso de cada vecino será de $w_{h,i} = -0.25$.

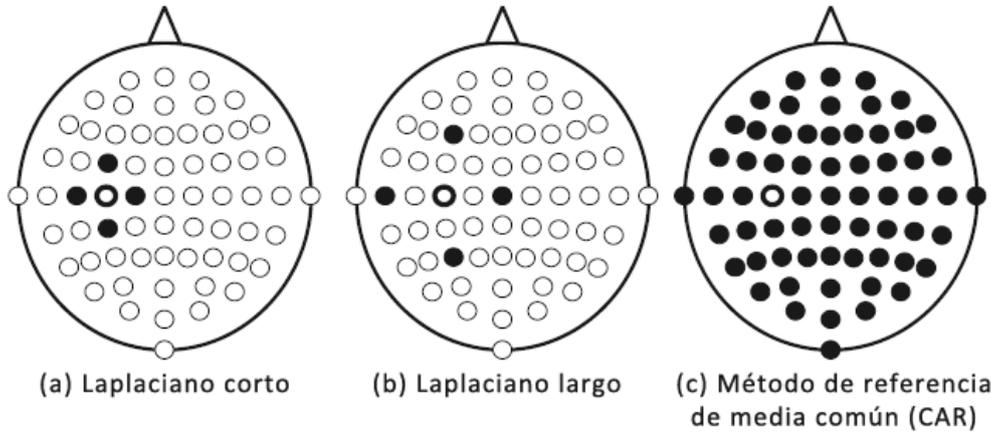


Figura 3-7. Tres filtrados espaciales diferentes para obtener la señal del canal C3: (a) Laplaciano corto, (b) Laplaciano largo y (c) método de referencia de media común (CAR). El electrodo objetivo C3 se corresponde con el círculo vacío y sus vecinos con círculos rellenos.

a.2) Método de referencia de media común

El método de referencia de media común (CAR, *Common Average Reference*) se basa en el mismo procedimiento que el filtro Laplaciano, sin embargo, difiere de éste en que esta vez se resta del canal obtenido la media común de todos los electrodos. Si la cabeza está completamente cubierta de electrodos equiespaciados entre ellos y el potencial generado en el córtex está muy localizado en fuentes puntuales, el método CAR provee un voltaje espacial de media cero, además de actuar como un filtro espacial paso-alto, acentuando las componentes que se encuentran más localizadas [35]. En la Figura 3-7 (c) se pueden observar los electrodos involucrados.

El voltaje del canal deseado se obtiene restando a éste la media común del voltaje obtenido en el resto de electrodos ($\frac{1}{H} \cdot \sum_{i=1}^H v_i(t)$), de acuerdo a la expresión [34]:

$$v_h^{CAR}(t) = v_h(t) - \frac{1}{H} \cdot \sum_{i=1}^H v_i(t) \quad (3.3)$$

Donde $v_n(t)$ es el voltaje recogido por el canal n -ésimo referenciado a la oreja, etiquetando como h el canal en el que se aplica el filtrado y siendo H el número total de electrodos empleados, incluyendo el canal filtrado.

a.3) Análisis de componentes independientes

El análisis de componentes independientes (ICA, *Independent Component Analysis*) 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 [36]. El objetivo de esta técnica es el de revelar características del EEG ocultas por el ruido basal para detectar los potenciales P300 con un solo intento (en Li *et al* [36] se han conseguido precisiones del 76.67%).

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 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:

$$\begin{cases} R_1(t) = a_{11}S_1(t) + a_{12}S_2(t) \\ R_2(t) = a_{21}S_1(t) + a_{22}S_2(t) \end{cases} \quad (3.4)$$

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 potenciales evocados P300 para facilitar su posterior detección [36].

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

$$\begin{cases} x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) + \dots \\ x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) + \dots \\ \dots \end{cases} \rightarrow \mathbf{x} = \mathbf{A} \cdot \mathbf{s} \quad (3.5)$$

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 (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:

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} = \mathbf{W} \cdot [\mathbf{A} \cdot \mathbf{s}] = \hat{\mathbf{s}} \rightarrow \hat{\mathbf{s}} = \mathbf{W} \cdot \mathbf{x} \quad (3.6)$$

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 [36].

a.4) Análisis de componentes principales

El análisis de componentes principales (PCA, *Principal Component Analysis*) es un método de reducción de dimensionalidad que permite seleccionar las componentes de la señal EEG más relevantes y descartar los canales que proveen información redundante. Este método cobra importancia cuando se trabaja con sistemas multicanal para discriminar qué canales aportan información útil para la discriminación de tareas y cuáles no.

El análisis PCA trata de encontrar, a partir del espacio geométrico resultante de proyectar los datos, una dirección del espacio tal que la varianza de los datos proyectados sea máxima. Este problema se describe en la Figura 3-8 y, tal y como se puede observar, la dirección que consigue la varianza máxima de los datos es D_1 , mientras que D_2 es una dirección cualquiera [37].

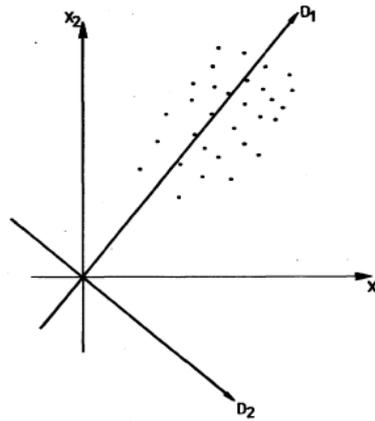


Figura 3-8. Búsqueda de la dirección que consigue que la varianza de los datos proyectados sea máxima (D_1) en un problema de análisis de componentes principales.

Para encontrar las direcciones buscadas se parte de un conjunto de datos de dimensión n , y éstas se obtienen restando la media de cada dimensión, calculando la matriz de covarianzas y obteniendo sus autovalores y autovectores. Estos autovectores se corresponden con las direcciones buscadas y se cumple que, además de ser ortonormales, al proyectar los datos sobre ellos la varianza que se obtiene en cada dirección coincide con su autovalor asociado. Por ende, el autovector asociado al autovalor más alto se corresponde con la dirección de máxima varianza [37].

De esta manera se pueden seleccionar las m direcciones principales y sobre ellas proyectar los datos, obteniendo un espacio resultante de dimensión m . Por tanto, si la dimensión m es menor que n , se habrá producido una compresión de los datos y en consecuencia, una pérdida de información. Se pretende que la pérdida de información sea principalmente de información redundante y, a causa de ello, se descartan los autovectores con los autovalores más pequeños.

b) Filtrado y análisis temporal

Dentro del filtrado y del análisis temporal para la extracción de los potenciales P300 cobran una gran importancia el promediado sincronizado, la detección de picos (*peak picking*), el cálculo del área de la señal y el filtrado adaptativo. Estas técnicas son muy rápidas y muy convenientes para el análisis a tiempo-real de la señal, sin embargo, para un análisis *offline* más exhaustivo, pueden emplearse técnicas frecuenciales, como la STFT (*Short Time Fourier Transform*, Transformada de Fourier de Tiempo Corto) o las transformadas *wavelet*.

Además del análisis temporal para determinar los potenciales P300, algunas de las técnicas temporales descritas a continuación también se aplican para eliminar artefactos comunes, tal y como se ha explicado anteriormente.

b.1) Promediado sincronizado

El promediado sincronizado permite distinguir los potenciales evocados P300 del ruido del EEG basal. Dado que el contenido espectral no deseado solapa la banda frecuencial donde se detectan los potenciales, los filtros lineales (como el filtrado paso-alto, paso-bajo o paso-banda) no son capaces de separar ambas señales [30].

El promediado sincronizado requiere poseer varias épocas de realización de la señal tras aplicar de manera repetitiva un estímulo y poseer una referencia temporal

para alinearlas adecuadamente. Estos requerimientos hacen que este análisis temporal sea idóneo para distinguir los potenciales P300 generados gracias al paradigma *odd-ball*.

Tal y como muestra la ecuación (3.7), consiste en promediar todas las épocas de la señal estableciendo como referencia ($n = 0$) el punto en el que se ha generado el estímulo que produce el potencial P300 [30].

$$Prom(n) = \sum_{k=1}^M y_k(n) = \sum_{k=1}^M x_k(n) + \sum_{k=1}^M w_k(n), \quad n = 1, 2, \dots, N \quad (3.7)$$

Siendo $y_k(n) = x_k(n) + w_k(n)$ una realización de la señal (con $x_k(n)$ la señal original sin ruido y $w_k(n)$ el ruido en la k -ésima época), k el índice de promediado, n el índice de tiempo discreto, M el número de épocas y N el número de muestras en cada época.

En el caso de la aplicación de la matriz *odd-ball* que genera P300 para la selección de letras, conocida como *P3Speller*, la manera de detectar qué celda se ha seleccionado es realizando un promediado sincronizado. Tanto la fila como la columna que contienen la celda deseada deben haber generado un potencial evocado P300 y, puesto que es imposible visualizar la presencia de un potencial P300 en una sola época de la señal, es necesario aplicar un promediado sincronizado de todas las combinaciones de filas y columnas (si la matriz es de 6x6 se promediarán 36 épocas). Posteriormente se aplicará la detección de picos o el cálculo del área para detectarlos automáticamente y poder averiguar cuál es la celda seleccionada [38].

b.2) Detección de picos y cálculo del área

La detección de picos o *peak picking* es una técnica muy simple que trata de detectar los potenciales evocados P300 monitorizando la amplitud de un promediado sincronizado de la señal. Para ello, se determina la diferencia entre el punto más negativo previo a la ventana establecida y el punto más alto dentro de dicha ventana [39] y se compara con la obtenida para un segmento basal. Los tamaños de la ventana son muy diversos, normalmente comprendiendo desde cientos de milisegundos antes de cada estímulo hasta más de 500ms después de éste, para comprender el potencial evocado, típicamente 300ms después del estímulo.

El cálculo del área comprendida bajo la señal promediada es un método muy sencillo y útil para detectar la presencia de este tipo de potenciales. En la práctica se suman todos los puntos de la ventana establecida y se determina la presencia de un P300 si el área comprendida es drásticamente mayor que en un segmento basal [39].

b.3) Filtrado adaptativo

El filtrado adaptativo no suele utilizarse frecuentemente en las aplicaciones que utilizan los potenciales evocados P300 debido a su estacionariedad, sin embargo, es útil para eliminar artefactos conocidos (e.g., el ECG residual). El filtrado adaptativo elimina un artefacto de la señal EEG sin asumir estacionariedad en la señal, no obstante, requiere una señal fuertemente correlada con el ruido a eliminar [30].

Tal y como propone su nombre, este filtro se adapta continuamente a las características de la señal, cambiando el vector de pesos a lo largo del tiempo, como se observa en la Figura 3-9. Para ello partimos de una señal de entrada $x(n)$, mezcla de la señal de interés $v(n)$ y del ruido primario $m(n)$, asumiendo que ambas componentes están incorreladas, como expresa la ecuación (3.8).

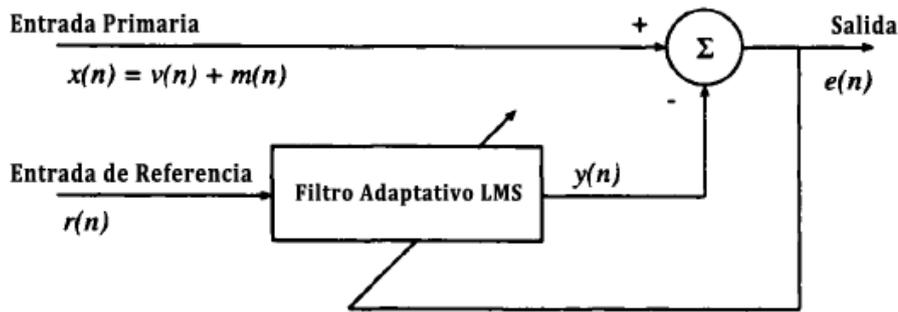


Figura 3-9. Esquema del filtrado adaptativo para eliminar una fuente de ruido.

El filtro adaptativo LMS filtra la referencia $r(n)$, fuertemente correlada con $m(n)$, para obtener una señal $y(n)$ lo más parecida al ruido primario que sea posible. Posteriormente, $y(n)$ es sustraída de la señal, obteniendo la estimación de la señal de interés $e(n)$, expresado en la ecuación (3.9).

$$x(n) = v(n) + m(n) \quad (3.8)$$

$$e(n) = \hat{v}(n) = x(n) - y(n) \quad (3.9)$$

El algoritmo LMS (*Least Mean Squares*, o Mínimos Cuadrados Promediados), que rige el funcionamiento del filtro adaptativo, trata de ajustar el vector de pesos \mathbf{w} para minimizar el error cuadrático medio (MSE, *Mean Square Error*) calculando un nuevo vector de pesos $\mathbf{w}(n+1)$ en base al vector de pesos actual $\mathbf{w}(n)$ más una correlación proporcional al gradiente negativo MSE. La ecuación (3.10) es una particularización de la regla de Widrow-Hoff y se encarga de actualizar los pesos en cada momento.

$$e^2(n) = x^2(n) - 2x(n)\mathbf{r}^T(n)\mathbf{w}(n) + \mathbf{w}^T(n)\mathbf{r}(n)\mathbf{r}^T(n)\mathbf{w}(n) \quad (3.10)$$

Este algoritmo es fácil de implementar y no requiere derivar, elevar al cuadrado o promediar aunque se base en minimizar el MSE. También puede utilizarse el algoritmo RLS (*Recursive Least Squares*, o Mínimos Cuadrados Recursivos), óptimo cuando las señales varían mucho en oscilaciones rápidas de tiempo [30].

c) Análisis tiempo-frecuencia

El uso de las técnicas de análisis espectral para extraer las características en aplicaciones que usan potenciales evocados P300 también está ampliamente extendido, en las que destacan la STFT (*Short-Time Fourier Transform*, o Transformada de Fourier de Tiempo Corto); las técnicas *wavelet*, tanto CWT (*Continuous Wavelet Transform*, o Transformada Wavelet Continua), DWT (*Discrete Wavelet Transform*, o Transformada Wavelet Discreta) y WPT (*Wavelet Packet Transform*, o Transformada de Paquetes Wavelet); y los modelos autorregresivos (AR, *Autoregressive Models*).

c.1) Transformada de Fourier de Tiempo Corto

La STFT (*Short-Time Fourier Transform*, o Transformada de Fourier de Tiempo Corto) básicamente divide la señal a analizar en una serie de segmentos que se suponen estacionarios y calcula la transformada de Fourier en ellos. Dado que la STFT no calcula la transformada de Fourier continua sino una versión enventanada, aparece distorsión debida a la ventana empleada creando un compromiso entre resolución temporal y frecuencial. Su expresión matemática se resume en la ecuación (3.11).

$$X_w[n, k] = \sum_{m=0}^{N-1} x[m] \cdot w^*[m-n] \cdot e^{-\frac{j2\pi km}{N}}, \quad \text{con} \quad \begin{array}{l} k \in [0, L-1] \\ n \in [0, N_T-1] \end{array} \quad (3.11)$$

Donde $w[n]$ representa la función ventana, $x[m]$ el segmento de la señal en el que se aplica la STFT y $N_T = N/L$ el número de segmentos no solapados en los que se ha dividido la señal [2].

Sin embargo, la suposición de que la señal segmentada es estacionaria y la resolución fija en el plano tiempo-frecuencia (Figura 3-10 (a)), que presenta su espectrograma una vez fijada la ventana, hacen que otras técnicas como las transformadas *wavelet* sean mucho más adecuadas en el análisis de señales biomédicas.

c.2) Transformadas Wavelet

Las desventajas de la STFT provocaron el desarrollo de las transformadas *wavelet* aplicadas a las señales EEG como sustitutas de este método. Todas las técnicas *wavelet* que se exponen a continuación se basan en la traslación y el escalado de una función *wavelet* madre (ecuación 3.12), generando así una resolución variable en el plano-tiempo frecuencia con buena resolución temporal para oscilaciones rápidas y buena resolución frecuencial para oscilaciones lentas, atendiendo al comportamiento característico de las señales biomédicas en general [2].

$$\varphi_{\tau,s} = \frac{1}{\sqrt{s}} \cdot \varphi\left(\frac{t-\tau}{s}\right) \quad (3.12)$$

Donde $\varphi(\tau, s)$ es la función *wavelet* madre, τ es el parámetro de traslación y s el parámetro de escalado. Existe un gran conjunto de *wavelets* madre y, habitualmente, el rendimiento de la aplicación se basa principalmente en elegir correctamente la más indicada; entre ellas se encuentran la *wavelet* de Haar, de Morlet, de Meyer, de Daubechies (en [40] se han conseguido precisiones de hasta 97.50% utilizando una *wavelet* madre de Daubechies4), etcétera.

La transformada *wavelet* continua (CWT, *Continous Wavelet Transform*) se define como la correlación entre la señal EEG y la función *wavelet* madre escalada y desplazada, expresada en la ecuación (3.13). De esta manera obtenemos una representación de la señal en el plano tiempo-escala, donde las bajas frecuencias de la señal se analizarían con altas escalas (produciendo una buena resolución frecuencial), y las altas frecuencias se analizarían con las escalas más bajas (produciendo una buena resolución temporal); tal y como muestra la Figura 3-10 (b).

$$w(\tau, s) = \int_{-\infty}^{+\infty} x(t) \cdot \frac{1}{\sqrt{s}} \cdot \varphi\left(\frac{t-\tau}{s}\right) dt \quad (3.13)$$

Sin embargo, esta transformación de la señal es demasiado redundante, dando paso a la transformada *wavelet* discreta (DWT, *Discrete Fourier Transform*). La DWT muestrea los valores de los parámetros de traslación (τ) y de escalado (s) de forma diádica (ecuación (3.14)) y, puesto que la CWT es demasiado redundante, la DWT no pierde excesiva información discriminativa sobre la señal. De esta manera, se requiere menor coste computacional, además de permitir un análisis multiresolución.

$$s = 2^{-j}; \tau = k \cdot 2^{-j} \rightarrow \varphi_{j,k}(t) = 2^{j/2} \cdot \varphi(2^j \cdot t - k) \quad (3.14)$$

Donde el factor $2^{j/2}$ se denomina factor de normalización, el cual garantiza la condición de ortonormalidad [2].

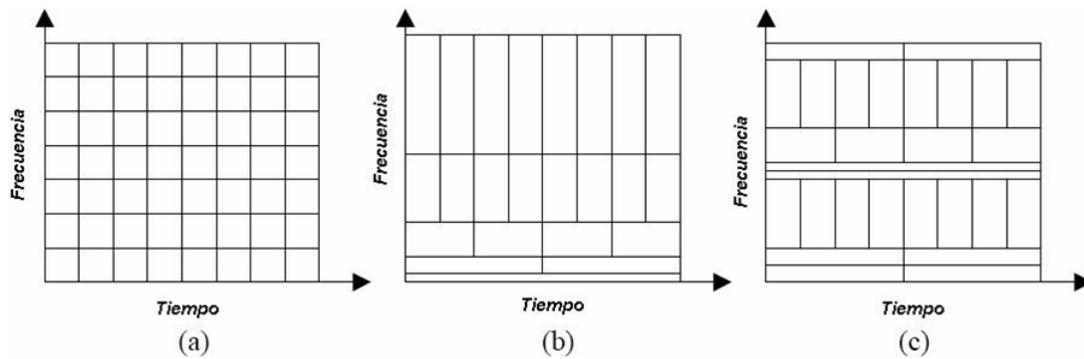


Figura 3-10. Plano tiempo-frecuencia para STFT en (a), con resoluciones fijas; DWT en (b), con resoluciones variables adaptadas a las señales biomédicas; y WPT en (c), con resoluciones variables según la intención del usuario.

El análisis multiresolución se basa en descomponer la señal en una parte de aproximación (coeficientes $c_{j,k}$) y una de detalle (coeficientes $d_{j,k}$). La aproximación refleja las principales características de la señal y el detalle contiene las fluctuaciones más rápidas, pudiendo considerarse como un filtro paso-bajo y un filtro paso-alto, respectivamente. A su vez, la aproximación se subdivide en otra señal de aproximación y otra de detalle, y así sucesivamente hasta alcanzar el número de escalas de descomposición deseado.

En la Figura 3-11 se muestran el espectrograma obtenido de la STFT (distribución de la densidad de potencia en el dominio tiempo-frecuencia) y el escalograma obtenido de la CWT (distribución de la densidad de potencia en el dominio tiempo-escala) para un promediado sincronizado de 90 épocas P300 obtenidas a través de un paradigma *odd-ball* auditivo. Tal y como se puede observar, la resolución del espectrograma es fija y no representa adecuadamente la distribución de potencia debido a una mala resolución temporal. Sin embargo, ese problema no se encuentra en el escalograma, donde sí existe una buena resolución temporal para las escalas más bajas [2].

No obstante, los resultados más precisos los provee la transformada de paquetes *wavelet* (WPT, *Wavelet Packet Transform*). La diferencia respecto a la DWT radica en que ésta nueva transformada, aparte de la señal de aproximación, subdivide también la señal de detalle, creando un árbol de descomposición mucho más extenso. Este análisis multiresolución extendido permite cambiar la resolución del plano tiempo-frecuencia tal y como desee el usuario a cambio de requerir un mayor coste computacional.

3.2.2. Traducción de características

Una vez extraídos los potenciales P300 de la señal EEG conviene clasificar si se ha producido un potencial o no en un momento determinado. Esta tarea la realizan los métodos de traducción de características, también conocidos con el nombre de clasificadores. Los clasificadores pueden dividirse en dos grandes grupos, tal y como se ha explicado anteriormente: los clasificadores lineales y los no lineales. Adicionalmente, en esta etapa es necesario considerar la adaptabilidad del usuario y asegurar que el clasificador vaya a funcionar en todas las situaciones posibles [5].

A continuación, se detallan los algoritmos de traducción más usados para clasificar las dos clases existentes en las señales de control P300: presencia o ausencia de potencial. Entre ellos se encuentran discriminante lineal de Fisher (FLD, *Fisher's*

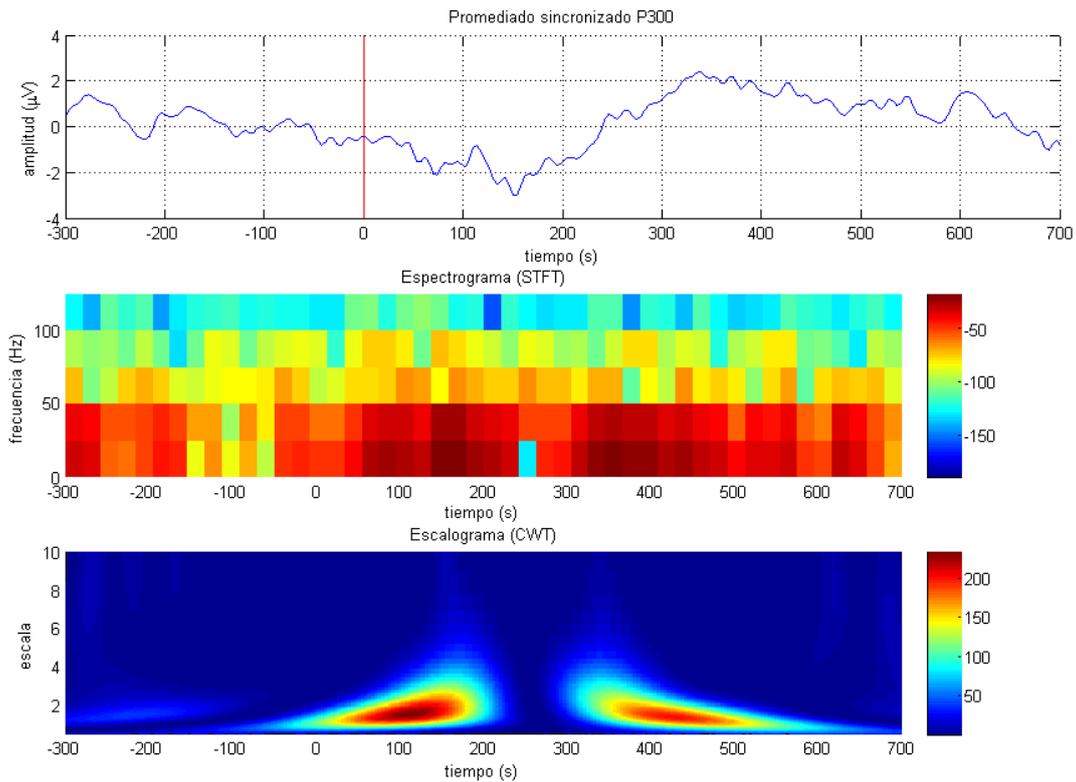


Figura 3-11. En la figura superior, representación del promediado sincronizado de 90 épocas de P300 para un *odd-ball* auditivo. En el centro, representación del espectrograma del mismo obtenido a través de la STFT. En la figura inferior, representación del escalograma del mismo obtenido a través de la CWT.

Linear Discriminant), el análisis discriminante lineal paso a paso (SWLDA, *StepWise Linear Discriminant Analysis*) y las máquinas de vectores de soporte (SVM, *Support Vector Machines*).

Determinar la presencia o la ausencia de un potencial evocado P300 a través de las características extraídas puede considerarse un problema de clasificación binario con una función discriminante cuyo hiperplano de decisión se define como:

$$\mathbf{w} \cdot \mathbf{f}(\mathbf{x}) + b = 0 \tag{3.15}$$

Donde \mathbf{x} es el vector de características, $f(\cdot)$ es la función de transformación, \mathbf{w} es un vector de pesos y b es el sesgo (*bias*). Para los métodos no lineales, la función $f(\cdot)$ puede representar una transformación que asigna las características a un espacio de dimensión mayor con el propósito de crear un conjunto de puntos que puedan separarse linealmente. Para los métodos lineales, la función $f(\cdot)$ simplemente es $f(\mathbf{x}) = \mathbf{x}$, [41].

En consecuencia, el vector de pesos \mathbf{w} asignará una etiqueta de clase a cada época o estímulo computado (comprendida entre -1 y +1), concretamente, +1 y -1 para el estímulo atendido y no atendido, respectivamente. Este diseño permite distinguir la respuesta con la mayor distancia positiva de todas las filas y columnas del *P300Speller* y determinar cuál es la celda seleccionada. Por tanto, los métodos explicados a continuación son distintos enfoques sobre cómo calcular ese vector de pesos (para los métodos lineales) y, adicionalmente, la función de transformación (para no lineales).

a) Discriminante lineal de Fisher

El FLD (*Fisher's Linear Discriminant*, o Discriminante Lineal de Fisher) es el punto de referencia para determinar el hiperplano de separación óptimo entre dos clases. Para clasificaciones binarias como la presentada aquí, la solución FLD y LMS son equivalentes, tratando de encontrar, a partir del espacio geométrico resultante de proyectar los datos, una dirección del espacio tal que el MSE de los datos proyectados sea mínimo. Este clasificador solamente se puede emplear cuando las dos clases poseen una distribución gaussiana y tienen la misma covarianza [41].

El vector de pesos para el caso propuesto se calcula de la siguiente manera:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{Y} \quad (3.16)$$

Siendo \mathbf{X} la matriz formada por los vectores de características e \mathbf{Y} las etiquetas de cada clase, tal y como se ha explicado anteriormente.

Se ha comprobado que los resultados obtenidos para el FLD reflejan con mayor fidelidad la calidad del estimador que PCM (*Pearson's Correlation Method*, o Método de Correlación de Pearson), debido a que este último solo indica si existe una relación lineal entre las dos variables, a diferencia de FLD, el cual emplea la covarianza entre características [41].

b) Análisis discriminante lineal paso-a-paso

SWLDA (*StepWise Linear Discriminant Analysis*, o Análisis Discriminante Lineal Paso-a-paso) es el método empleado como clasificador en la aplicación desarrollada en el presente trabajo fin de grado.

En general, los métodos de análisis discriminante lineal (LDA, *Linear Discriminant Analysis*) tratan de encontrar una combinación lineal de características capaz de separar dos o más clases de eventos mediante el uso de estadísticas de segundo orden para obtener el vector de pesos \mathbf{w} . En consecuencia, la calidad del clasificador resultante depende de la calidad de la covarianza estimada y, en la práctica, por cuestiones de tiempo y coste computacional, es difícil o imposible obtener la cantidad de datos necesarios para establecer un buen clasificador LDA.

Por esta razón surge el algoritmo SWLDA, el cual trata de mejorar la covarianza estimada reduciendo el número de pesos mayores que cero al mínimo. SWDLA empieza con un subconjunto de dimensiones de datos y determina para cada dimensión cuánto afectará a la calidad de la estimación el incluirla o eliminarla en el cálculo, y modifica el conjunto de datos consecuentemente.

Realmente, este análisis es una particularización del método de regresión paso-a-paso, el cual combina dos técnicas básicas de selección de características: el análisis paso-a-paso hacia delante y el análisis paso-a-paso hacia atrás [44].

El análisis paso-a-paso hacia delante (*forward stepwise regression*) parte de un conjunto vacío de características y evalúa la significación de cada una, añadiéndola al conjunto si supera un cierto umbral o criterio de entrada y descartándola en caso contrario.

El análisis paso-a-paso hacia atrás (*backward stepwise regression*) realiza la función contraria: parte de un conjunto con todas las características posibles y evalúa cada una de ellas, preguntándose si deberían estar incluidas en el conjunto o no. Para ello, establece un criterio de salida y recorre el conjunto de características comprobando si éstas cumplen el criterio o no. Si una característica cumple el criterio, se descarta, reduciendo así el tamaño del conjunto.

El algoritmo SWLDA combina estos métodos de la siguiente manera: parte de un conjunto vacío y realiza un análisis paso-a-paso hacia delante para añadir las características más significativas, es decir, aquellas que cumplan con el criterio de entrada. Cada vez que se añade una característica al conjunto de datos, se hallan los p -valores de nuevo y se establece si alguna de las características incluidas en el conjunto cumple un determinado criterio de salida, utilizando un análisis paso-a-paso hacia atrás. Si la respuesta es afirmativa, se descartan las características que lo cumplan y se vuelve a realizar un análisis paso-a-paso hacia delante. Este procedimiento se repite constantemente hasta que el conjunto de datos alcance el número máximo de características o hasta que no existan más características que cumplan los criterios descritos [44].

Numerosos estudios han demostrado la efectividad del uso de SWLDA como algoritmo de entrenamiento para el clasificador empleado en aplicaciones que usan potenciales evocados P300, como el *P3Speller* [41]. Normalmente se emplean los siguientes criterios de entrada y de salida, expresados en la ecuación (3.17).

$$\begin{cases} \text{Criterio de entrada} \rightarrow p\text{-valor} < 0.1 \\ \text{Criterio de salida} \rightarrow p\text{-valor} > 0.15 \end{cases} \quad (3.17)$$

Por tanto, se añaden a la función discriminante, o conjunto de datos, las características que obtienen un p -valor inferior a 0.1. Después de cada introducción a la función discriminante se calculan los nuevos p -valores y se realiza un análisis paso-a-paso hacia atrás para eliminar las características menos significantes, en concreto, las que obtienen un p -valor superior a 0.15.

c) Máquinas de vectores de soporte

Las SVM (*Support Vector Machines*, o Máquinas de Vectores de Soporte) son un conjunto de algoritmos de aprendizaje supervisado que tratan de construir un hiperplano o conjunto de hiperplanos en un conjunto de dimensionalidad muy alta que pueden ser utilizados en problemas de clasificación o regresión no lineal.

El hiperplano buscado debe de guardar la máxima distancia (margen ρ) con los puntos que estén más cerca del mismo, y por esta razón, es conocido como un clasificador de margen máximo, ilustrado en la Figura 3-12. La manera más simple de separar las clases es mediante una función lineal, sin embargo, los universos a estudiar no suelen presentar esos casos idílicos y suele ser necesario proyectar la información a un espacio de características de mayor dimensión a través de funciones Kernel [42].

Dado el vector de características \mathbf{x} y el vector de etiquetas de clases \mathbf{d} (ambos vectores con longitud N), cuyos valores cumplen que $d_i \in [\pm 1]$, la ecuación (3.15) puede reformularse como sigue [41].

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{para } i \in [1, N] \quad (3.18)$$

El hiperplano óptimo de separación deberá maximizar la ecuación (3.18), cuyo margen de separación para la clasificación binaria es equivalente a minimizar la norma euclídea del vector de pesos óptimo \mathbf{w}_0 , como expresa la ecuación (3.19) [42].

$$\rho = \frac{2}{\|\mathbf{w}_0\|} \quad (3.19)$$

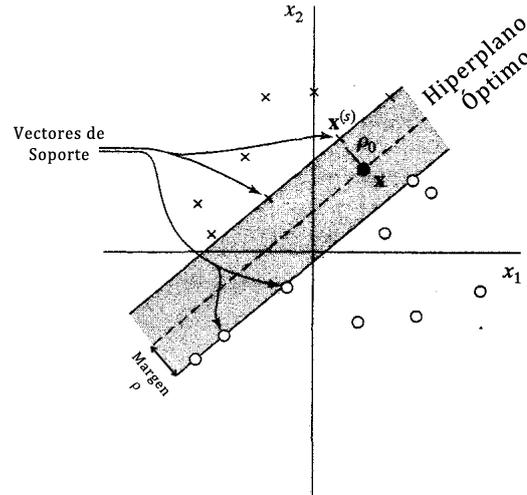


Figura 3-12. Representación gráfica del hiperplano óptimo de separación para el algoritmo SVM en una clasificación binaria.

Los puntos que se encuentran más próximos al hiperplano de separación y cuyo margen de separación ρ_0 es máximo se denominan vectores de soporte y su clasificación es complicada. Una vez conocida la expresión del margen de separación es necesario maximizar la ecuación (3.18) mediante el método de los multiplicadores de Lagrange.

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j x_i^T x_j \quad (3.20)$$

A continuación se hallan los multiplicadores de Lagrange óptimos $\alpha[\alpha_0, \dots, \alpha_i]$ tratando de maximizar la ecuación (3.20) anterior, sujeta a las siguientes restricciones.

$$\sum_{i=1}^N \alpha_i d_i = 0, \quad \text{con } \alpha_i \geq 0 \quad \text{para } i = 1, \dots, N \quad (3.20)$$

Finalmente, el hiperplano óptimo, el vector de pesos óptimo w_0 y el sesgo óptimo b_0 se detallan en las ecuaciones (3.22) y (3.23) para una separación lineal [42].

$$w_0 = \sum_{i=1}^N \alpha_{0,i} d_i x_i \quad (3.22)$$

$$b_0 = 1 - w_0^T x^{(s)}, \quad \text{con } d^{(s)} = 1 \quad (3.23)$$

Desgraciadamente, cuando las clases no pueden separarse linealmente no se puede construir el hiperplano con las ecuaciones (3.22) y (3.23) sin que se produzcan errores en la clasificación. El parámetro ξ_i mide la diferencia desde el patrón considerado hasta el hiperplano óptimo, cuantizando este error y pudiendo darse dos casos, representados en la Figura 3-13 [42].

1. Que uno de los datos caiga en la zona de decisión y en el lado correcto del hiperplano de separación, Figura 3-13 (a). Entonces $0 \leq \xi_i \leq 1$.
2. Que uno de los datos caiga en el lado incorrecto de la frontera de decisión establecida por el hiperplano, Figura 3-13 (b). Entonces $\xi_i > 1$.

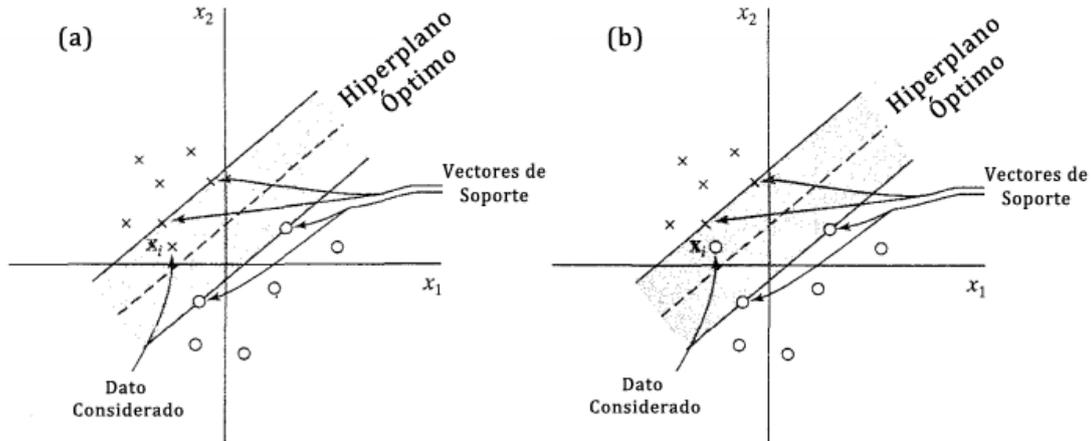


Figura 3-13. Posibles errores en la clasificación SVM. En (a) el dato considerado se encuentra dentro de la región de separación y se clasifica correctamente. En (b) el dato considerado se encuentra dentro de la región de separación y se clasifica incorrectamente.

El hiperplano óptimo cuando las variables no son linealmente separables se encuentra aumentando la dimensión del conjunto de datos y minimizando la ecuación siguiente (3.24), donde C es un parámetro escalar determinado experimentalmente.

$$\phi(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \quad (3.24)$$

Para realizar esta transformación no lineal se utilizan funciones Kernel gaussianas debido a sus propiedades de aproximación universal, como se indica a continuación [41].

$$score_{SVM} = \sum_{i=1}^N \alpha_{0,i} d_i \cdot \mathbf{K}(x_i, x_j), \quad \text{con } \mathbf{K}(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (3.25)$$

Donde los parámetros C y σ^2 se obtienen experimentalmente. En [41] se demuestra que los mejores parámetros para el *P3Speller* son $C = 10$ y $\sigma^2 = 10^3$ para ese experimento, sin embargo, pueden variar para cada sujeto.

3.3. Umbralización

Al clasificar cada estímulo y asignar pesos a cada uno de ellos para determinar cuál es, en el caso del *P3Speller*, la fila y columna que ha querido seleccionar el usuario, se establece que los estímulos P300 con mayor peso son los seleccionados.

Sin embargo, puede darse el caso de que el usuario no haya querido seleccionar nada puesto que no estaba atendiendo a la presentación de los estímulos. En ese caso el clasificador establecerá una selección igualmente escogiendo el peso mayor, aunque los pesos de cada estímulo sean más pequeños de lo normal. Como solución a este problema se propone la implementación de un umbral que decida cuándo tener en cuenta la decisión del clasificador y cuándo no, con el fin de evitar selecciones indeseadas por el usuario.

El umbral óptimo utilizado en el presente trabajo se basa en la curva ROC resultante de representar la sensibilidad y especificidad obtenidas en las sesiones de entrenamiento con los usuarios, procedimiento explicado a continuación.

3.3.1. Curva ROC

Una curva ROC (acrónimo de *Receive Operating Characteristic*, o Característica Operativa del Receptor) es una representación gráfica de la *sensibilidad* frente a *1-especificidad* para un sistema de clasificación binario.

Considerando un problema de predicción de clases binario en el que las clases se etiquetan como positivas **P** o negativas **N**, existen cuatro tipos de resultados tras la clasificación. Por ejemplo, si se intentase clasificar la presencia de una enfermedad en ciertos sujetos y se etiquetase como positivos a los enfermos y como negativos a los sujetos sanos, ésta clasificación podría causar una de las cuatro situaciones siguientes:

1. **TP** (*True Positives*, o Verdaderos Positivos): Cuando las clases positivas **P** se clasifican como positivas **P**, es decir, los sujetos enfermos se diagnosticarían correctamente como enfermos.
2. **FP** (*False Positives*, o Falsos Positivos): Cuando las clases negativas **N** se clasifican como positivas **P**, es decir, los sujetos sanos se diagnosticarían erróneamente como enfermos.
3. **TN** (*True Negatives*, o Verdaderos Negativos): Cuando las clases negativas **N** se clasifican como negativas **N**, es decir, los sujetos sanos se diagnosticarían correctamente como sanos.
4. **FN** (*False Negatives*, o Falsos Negativos): Cuando las clases positivas **P** se clasifican como negativas **N**, es decir, los sujetos enfermos se diagnosticarían erróneamente como sanos.

En la Figura 3-14 se muestran los cuatro posibles resultados en una matriz de confusión según el valor de la clase y la predicción del clasificador. En base a estos cuatro posibles resultados se establecen los siguientes parámetros:

1. **Sensibilidad**: Probabilidad de que las clases positivas **P** se clasifiquen correctamente, es decir, probabilidad de que el clasificador determine enfermedad cuando es aplicado sobre una población de sujetos enfermos.

$$\text{Sensibilidad} = \frac{TP}{TP + FN} \quad (3.26)$$

2. **Especificidad**: Probabilidad de que las clases negativas **N** se clasifiquen correctamente, es decir, probabilidad de que el clasificador determine ausencia de enfermedad cuando es aplicado sobre una población de sujetos sanos.

$$\text{Especificidad} = \frac{TN}{TN + FP} \quad (3.27)$$

3. **Precisión**: Probabilidad de que ambas clases se clasifiquen correctamente, es decir, probabilidad de que un sujeto, esté sano o enfermo, se clasifique correctamente.

$$\text{Precisión} = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.28)$$

4. **PPV** (*Positive Predictive Value*, o Valor Predictivo Positivo): Probabilidad de que una clase clasificada como positiva **P** sea realmente positiva **P**, es decir, probabilidad de que un sujeto clasificado como enfermo esté realmente enfermo.

$$\text{PPV} = \frac{TP}{TP + FP} \quad (3.29)$$

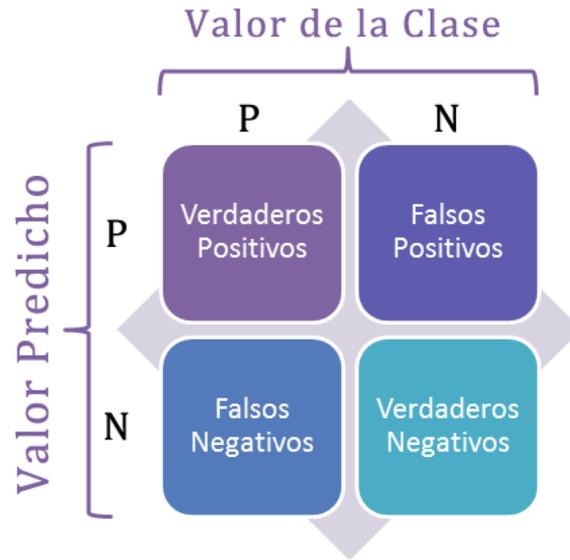


Figura 3-14. Matriz de confusión que recoge los cuatro posibles resultados tras aplicar un clasificador binario: las columnas establecen el valor real de la clase y las filas el valor predicho por el clasificador.

- 5. **NPV** (*Negative Predictive Value*, o Valor Predictivo Negativo): Probabilidad de que una clase clasificada como negativa *N* sea realmente negativa *N*, es decir, probabilidad de que un sujeto clasificado como sano esté realmente sano.

$$NPV = \frac{TN}{TN + FN} \tag{3.30}$$

La curva ROC está íntimamente ligada con la sensibilidad y la especificidad de las dos clases de datos, concretamente, representa en el eje de abscisas *1-especificidad* y en el eje de ordenadas la *sensibilidad*. La finalidad de las curvas ROC es la determinación de un umbral de decisión que permita dividir las dos clases de datos de manera óptima.

El primer paso para la construcción de una curva ROC es establecer una serie de umbrales μ_k que se recorrerá sucesivamente. Posteriormente, se calcula la sensibilidad y la especificidad de ambas clases para cada umbral y finalmente se representa gráficamente con los ejes descritos. En conclusión, se deben obtener tantos pares sensibilidad-especificidad como umbrales se hayan definido. En la Figura 3-15 se aprecian los valores de la curva ROC para tres umbrales distintos.

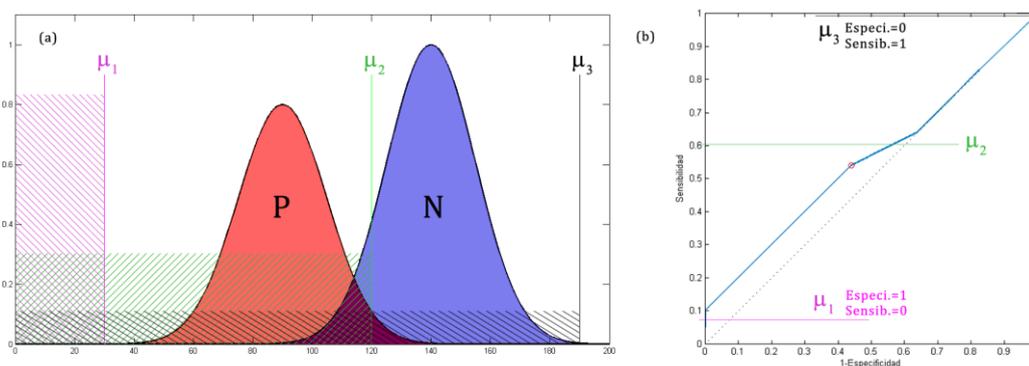


Figura 3-15. Construcción de una curva ROC (b) para las clases positivo (P) y negativo (N) con distribuciones gaussianas (a).

Los datos que se encuentren por debajo del umbral elegido se considerarán negativos N y los que se encuentren por encima se considerarán positivos P . Tal y como se puede observar, el umbral indicado como μ_1 clasifica correctamente a todos los negativos N pero incorrectamente a todos los positivos P y, por tanto, su especificidad será de 1 y su sensibilidad será nula. El umbral indicado como μ_2 clasifica correctamente a casi todos los positivos P , aunque posee errores de clasificación para ambas clases y, por tanto, su especificidad y su sensibilidad serán mayores que 0 pero menores que 1. Finalmente, el umbral indicado como μ_3 clasifica correctamente a todos los positivos P pero incorrectamente a todos los negativos N y, por tanto, su especificidad será nula y su sensibilidad valdrá 1.

Una vez construida la curva es necesario establecer el umbral óptimo para la clasificación. El punto óptimo se encontraría en (0,1) puesto que la especificidad y la sensibilidad serían máximas, clasificando correctamente a todas las clases. Sin embargo, una curva ROC solamente contendrá este punto si ambas distribuciones están completamente separadas. Cuando las distribuciones están solapadas, se establece el umbral óptimo en el punto que minimice la distancia hasta el punto óptimo, tal y como muestra la Figura 3-16.

El área bajo la curva ROC indica la robustez del método. Este área posee un valor comprendido entre 0.5 y 1, donde 1 representa un valor diagnóstico perfecto (puesto que clasifica correctamente ambas clases) y 0.5 es una prueba sin capacidad discriminatoria diagnóstica (ver Figura 3-17). Si se obtienen valores inferiores a 0.5 se recomienda cambiar el orden de las clases, puesto que la curva es simétrica.

El umbral óptimo típico se establece tal y como se ha explicado anteriormente, sin embargo, la elección del umbral depende completamente de cada caso particular y siempre se verá afectado por el compromiso entre especificidad y sensibilidad. Considerando el caso de sujetos sanos y enfermos nuevamente; en unos casos será preferible descartar el menor número posible de enfermos aunque se diagnostiquen erróneamente sujetos sanos, por ejemplo cuando las consecuencias de la enfermedad sean graves, tendiendo entonces a seleccionar umbrales que proporcionen un mayor valor de sensibilidad; mientras que en otros casos se puede preferir reducir costes y

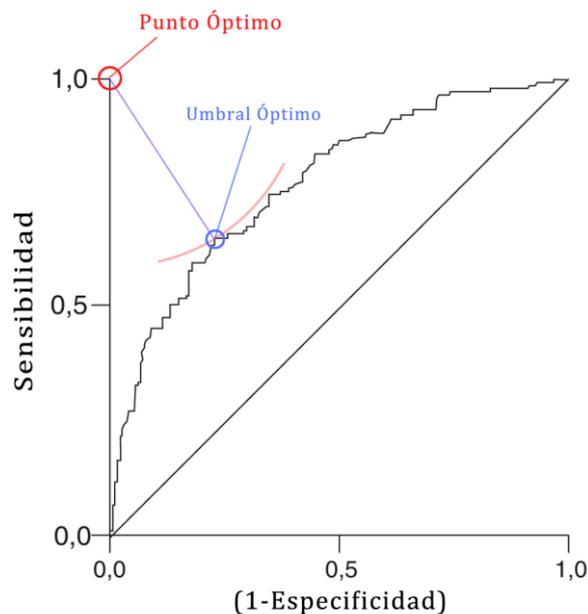


Figura 3-16. Detección del umbral óptimo: punto que minimiza la distancia entre éste y el punto óptimo de la curva ROC. Este umbral óptimo maximiza el par *Sensibilidad-Especificidad*.

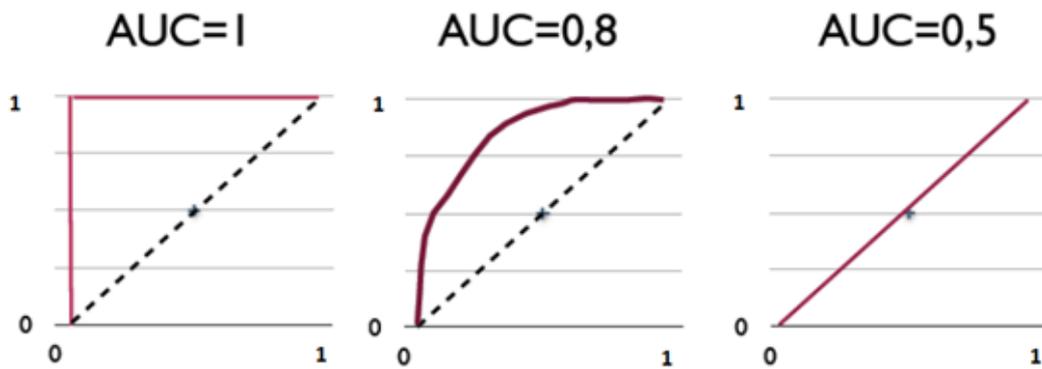


Figura 3-17. Área bajo la curva ROC (AUC) en tres casos distintos. En (a) se obtiene un valor diagnóstico perfecto, en (b) un buen valor diagnóstico y en (c) la peor curva posible, sin valor diagnóstico alguno.

no sobrediagnosticar la enfermedad, tendiendo en estos casos hacia umbrales que supongan un mayor valor de especificidad.

En la aplicación desarrollada en el presente trabajo de fin de grado es necesario establecer un umbral para detectar si el usuario atiende a los estímulos en el *P3Speller* o, por el contrario, los ignora. En este caso se han realizado pruebas con cada sujeto en dos sesiones distintas. En cada sesión se ha ordenado al usuario atender a ciertos estímulos e ignorar otros, generando datos de manera simétrica. Posteriormente se ha calculado la curva ROC para cada usuario y para cada sesión realizada, eligiendo como umbrales óptimos los umbrales típicos de las curvas ROC. Finalmente, se ha optado por calcular el promedio de ambos umbrales y utilizar éste para clasificar entre atención e indiferencia ante los estímulos.

DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE NAVEGACIÓN WEB

4.1. Objetivo de la aplicación

El objetivo principal de la aplicación a desarrollar es el de proveer una herramienta de ayuda para aquellas personas que sufren una grave discapacidad, cuya finalidad sea controlar un navegador web únicamente con las ondas cerebrales, sin la intervención de los mecanismos normales de los nervios y músculos periféricos.

El método seleccionado para registrar la actividad cerebral que permitirá controlar la aplicación es el EEG debido al bajo coste que implica su obtención, a que es un método no invasivo, portátil y de fácil uso. En el segundo capítulo se realizó una revisión del estado del arte sobre los navegadores desarrollados anteriormente con señales EEG, los cuales utilizaban diversas señales de control. Las señales de control seleccionadas para esta aplicación son los potenciales P300 (al igual que en [22], [23] y [24]), generados a través del paradigma *odd-ball*, puesto que no requieren entrenamiento por parte del usuario y su procesamiento es rápido y sencillo.

El navegador web a desarrollar deberá ser una aplicación en la que el usuario pueda ser independiente, sin necesidad de contar con ningún supervisor (a diferencia de [19]) y que permita una navegación completa y libre, sin tiempos de lectura predeterminados. Las selecciones deberán ser rápidas y el sistema deberá adaptarse al número de hipervínculos de una página web, adecuando la velocidad del sistema en función de éstos. En futuras versiones se podrá incluir una lista de marcadores y un historial para el usuario.

A continuación se detallan las partes y el desarrollo de la aplicación, además de la evaluación por parte de los usuarios finales y las tareas propuestas con ese fin.

4.2. Partes de la aplicación

Como advierte la Figura 4-1, se pueden distinguir tres partes principales de la aplicación: la adquisición de la señal, el *BCI Browser: P3Speller* y el *BCI Browser: Navigator*.

La parte de adquisición de la señal registra el EEG del usuario a través de los electrodos y el amplificador proporcionados por la empresa *G·Tec*.

La segunda parte de la aplicación, identificada como *BCI Browser: P3Speller*, se encarga de procesar la señal EEG a través del sistema de propósito general BCI2000 y una aplicación programada en lenguaje C++. Esta parte de la aplicación filtra la señal y extrae las características de los potenciales P300 generados atendiendo a un paradigma *odd-ball* visual compuesto por dos matrices que intercalará el usuario a voluntad. Para ello aplica, en primer lugar, un filtrado espacial CAR para reducir el difuminado espacial en los electrodos deseados y, en segundo lugar, un promediado temporal referenciando las épocas para distinguir los potenciales de la señal EEG



Figura 4-1. Estructura principal de la aplicación de navegación web desarrollada.

basal, tal y como se explicó en el anterior capítulo. Posteriormente traduce las características con un algoritmo SWLDA y clasifica los potenciales para determinar qué comando ha deseado seleccionar el usuario. Finalmente compara la intensidad del potencial para determinar si el usuario ha querido, efectivamente, seleccionar la celda o por el contrario deseaba ignorar los estímulos. Si se supera el umbral, envía una pulsación de teclado correspondiente al comando deseado hacia la siguiente parte de la aplicación; en caso contrario, indica que el umbral no ha sido superado.

La tercera y última parte de la aplicación, identificada como *BCI Browser: Navigator*, se encarga de procesar los comandos de teclado recibidos y traducirlos en comandos del navegador *Google Chrome*. Para ello se ha desarrollado una extensión del mismo basada en el lenguaje JavaScript, cuyo funcionamiento se detalla más tarde.

4.3. Funcionamiento general de la aplicación

Como ya se ha advertido en el apartado anterior, la aplicación consta de tres partes principales que se comunican entre sí para poder proveer al usuario de una navegación completa y libre.

En términos generales, la interfaz gráfica se compone de una matriz de comandos a la derecha de la pantalla y el navegador web *Google Chrome* a la izquierda de la misma. Debido al gran número de comandos necesarios para controlar el navegador web, se han desarrollado dos matrices que se intercalan a voluntad del usuario. Esta interfaz se muestra en la Figura 4-2.

Estas matrices se encargarán de seleccionar el comando que desee el usuario a través del paradigma *odd-ball*. Para ello, se iluminan de manera aleatoria sus filas y sus columnas y se genera un potencial evocado P300 en el EEG de usuario cada vez que la celda que quiera seleccionar se ilumine. Detectando esos potenciales evocados P300 se determina cuál es el comando seleccionado. Sin embargo, éstas matrices no dejan de iluminarse en ningún momento, provocando que continuamente se seleccionen comandos no deseados. Por esta razón se ha implementado un umbral que determina si la intensidad del potencial es lo suficientemente alta como para considerarlo como una selección, puesto que el usuario podría estar leyendo la página web tranquilamente. Este umbral varía entre usuarios y se determina en una sesión de entrenamiento previa.

Para seleccionar los hipervínculos de una web y poder navegar entre varias sin necesidad de ningún supervisor, la aplicación codifica y etiqueta los hipervínculos y los formularios de la página web que se esté visualizando. De esta manera, cuando el usuario desee seleccionar un hipervínculo o introducir texto en un formulario, solamente tendrá que introducir la codificación del mismo, la cual estará formada por números del 0 al 5, a través de las matrices del *P3Speller*.

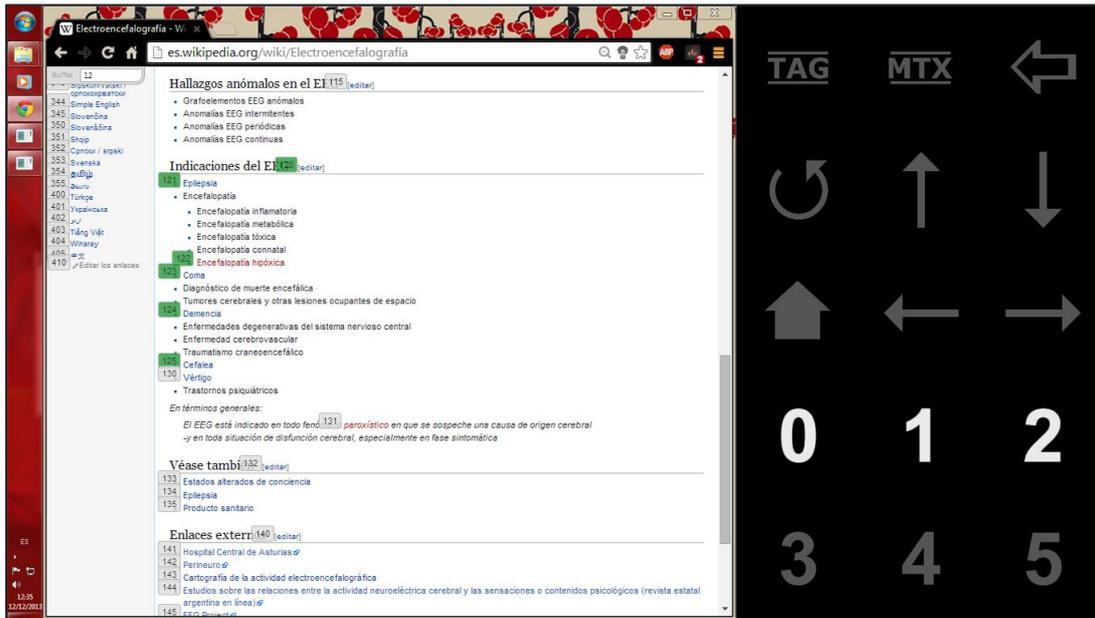


Figura 4-2. Interfaz gráfica de la aplicación desarrollada.

A continuación, se detallan el diseño y el desarrollo de las tres partes principales de la aplicación, en términos más específicos.

4.4. Adquisición de la señal EEG

El gorro que recoge la señal EEG del cuero cabelludo del usuario utiliza los siguientes electrodos: Fz, Cz, Pz (básicos para obtener correctamente los potenciales evocados P300), P3, P4, PO7, PO8 y Oz; cuya localización puede observarse en la Figura 4-3.

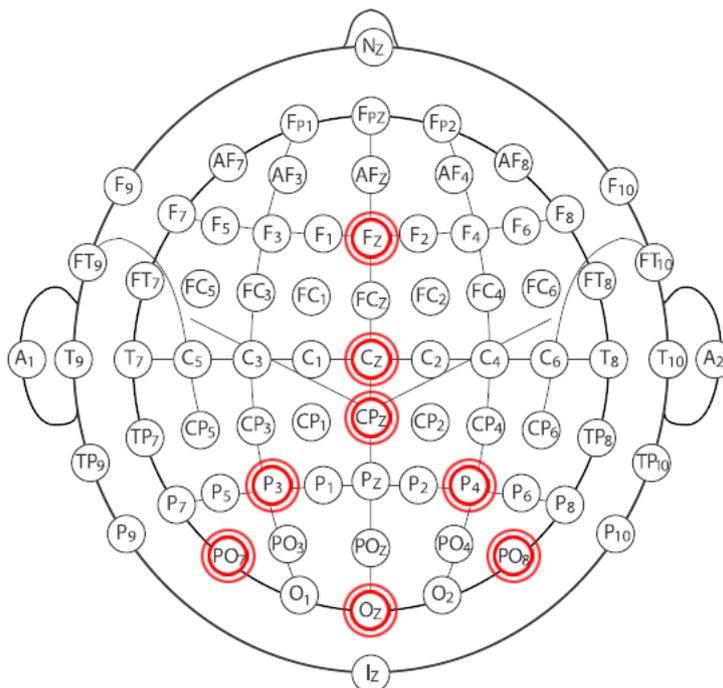


Figura 4-3. Disposición de los electrodos utilizados en la aplicación según el sistema internacional 10/20: Fz. Cz. Pz. P3. P4. PO7. PO8 v Oz.

Tal y como se ha explicado anteriormente, es necesario colocar un gel conductor entre los electrodos y el cuero cabelludo para reducir la impedancia del cráneo y poder amplificar la señal correctamente. La amplificación se realiza a manos de *g.USBamp*, un amplificador desarrollado por la empresa *G.Tec*. En cuanto a la frecuencia de muestreo, esta aplicación la fija en 256Hz, adecuada para el análisis del EEG puesto que cumple con creces el teorema de Nyquist.

4.5. BCI Browser: P3Speller

La siguiente parte de la aplicación, *BCI Browser: P3Speller*, se encarga de permitir al usuario configurar los distintos parámetros que caracterizarán a la aplicación, de generar los potenciales evocados P300, de procesar la señal EEG adecuadamente y de determinar si realmente el usuario deseaba seleccionar un comando determinado a través de un umbral.

4.5.1. Sistema de propósito general BCI2000

La implementación del *P3Speller* se realizó con la ayuda del sistema de propósito general BCI2000, junto con una aplicación implementada en C++ [34]. La finalidad del sistema BCI2000 es facilitar el desarrollo, la investigación y la monitorización de aplicaciones en tiempo real que requieren adquirir una señal EEG y procesarla utilizando diversas señales de control.

La estructura de BCI2000 se compone de cuatro módulos que se comunican entre ellos: la “Fuente”, el “Procesado de la Señal”, la “Aplicación” y el “Operador”. Estos módulos son programas independientes y se comunican entre sí a través de un protocolo basado en TCP/IP, sin embargo, dado que esta aplicación no requiere un sistema distribuido, los módulos se comunican en la máquina local. La estructura se puede observar en la Figura 4-4.

La señal EEG se adquiere en el módulo “Fuente” y se procesa de manera síncrona, en bloques que contienen un determinado número de muestras. Cada vez que se adquiere un nuevo bloque de datos, el módulo “Fuente” lo envía al módulo de “Procesado de la Señal”, a su vez subdividido en dos partes: extracción y traducción de características, que se encarga de producir señales de control que envía al módulo de “Aplicación”. Finalmente, el módulo de “Aplicación” envía la señal EEG analizada y marcada con eventos que facilitan la distinción de las diversas señales de control a utilizar al módulo “Fuente”, con el fin de poder realizar un análisis *off-line* gracias al almacenamiento en disco de la señal por parte de este último módulo. El módulo

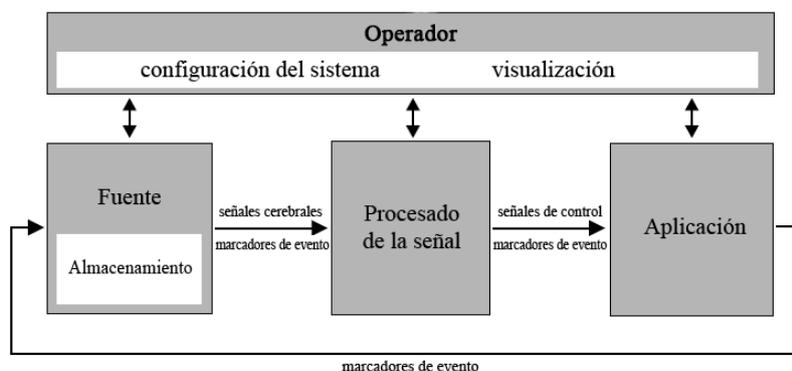


Figura 4-4. Estructura del sistema de propósito general BCI2000, compuesta por cuatro módulos independientes.

“Operador” se encuentra apartado de este análisis EEG, sin embargo, provee al usuario una interfaz gráfica que permite configurar los distintos parámetros del sistema y visualizar en tiempo-real el análisis de la señal. Por esta razón, el módulo “Operador” se comunica con el resto de módulos y permite al usuario iniciar, parar, suspender o continuar las operaciones que realiza el sistema [34].

El sistema BCI2000 propone ya una solución para implementar un paradigma *odd-ball* visual con el fin de generar los potenciales evocados P300 a través de una matriz cuyas filas y columnas se iluminan aleatoriamente. Esta solución sugiere utilizar un filtro CAR y un promediado sincronizado como métodos de extracción de características y un algoritmo SWLDA como método de traducción de las mismas. En el presente trabajo se ha mantenido la utilización de estos métodos pero, sin embargo, se ha adaptado la aplicación *P3Speller* para producir los comandos específicos para la aplicación y para establecer un umbral que permita identificar la intención del usuario en cada momento. Para ello se ha modificado principalmente el módulo de “Aplicación” y parte del “Operador”.

4.5.2. Configuración de los parámetros

Nada más iniciar la aplicación se inicia el módulo “Operador”, donde el usuario puede configurar diversos parámetros como la ruta de los archivos *.dat* que generará, el número de canales a analizar, el nombre de los electrodos utilizados, el filtro espacial a utilizar, los pesos del clasificador del usuario, el tamaño y las celdas de las matrices que generan los P300, la duración y el número de las secuencias, la duración de los estímulos o el modo de funcionamiento o el umbral de cada usuario, entre otros. En la Figura 4-5 aparece la interfaz gráfica del “Operador”.

La aplicación está optimizada para pantallas panorámicas, donde aproximadamente 2/3 de la longitud horizontal de la pantalla los ocupará el navegador web *Google Chrome* y el 1/3 restante lo ocupará la matriz del *P3Speller*, utilizada para generar los potenciales evocados P300 y poder controlar el navegador web. La estructura de la interfaz gráfica aparece representada en la Figura 4-6. Teniendo en cuenta estos aspectos, se recomienda modificar los parámetros oportunos en el “Operador” como sigue, si consideramos como X a la longitud horizontal y como Y a la longitud vertical de la pantalla en píxeles.

$$WindowHeight = Y; WindowWidth = \left\lfloor \frac{3}{5} \cdot Y \right\rfloor \text{ y } WindowLeft = X - WindowWidth \quad (4.1)$$

Antes de iniciar la aplicación es necesario establecer ciertos parámetros que determinarán el funcionamiento del paradigma *odd-ball*. Considerando como un intento (o *trial*) la selección de un comando determinado, éste puede descomponerse

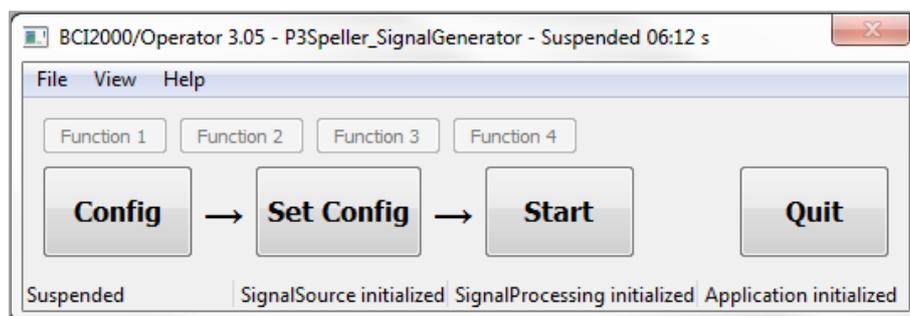


Figura 4-5. Interfaz gráfica del “Operador”, donde se configuran los parámetros relativos a la aplicación, además de permitir el control de ésta a tiempo real.

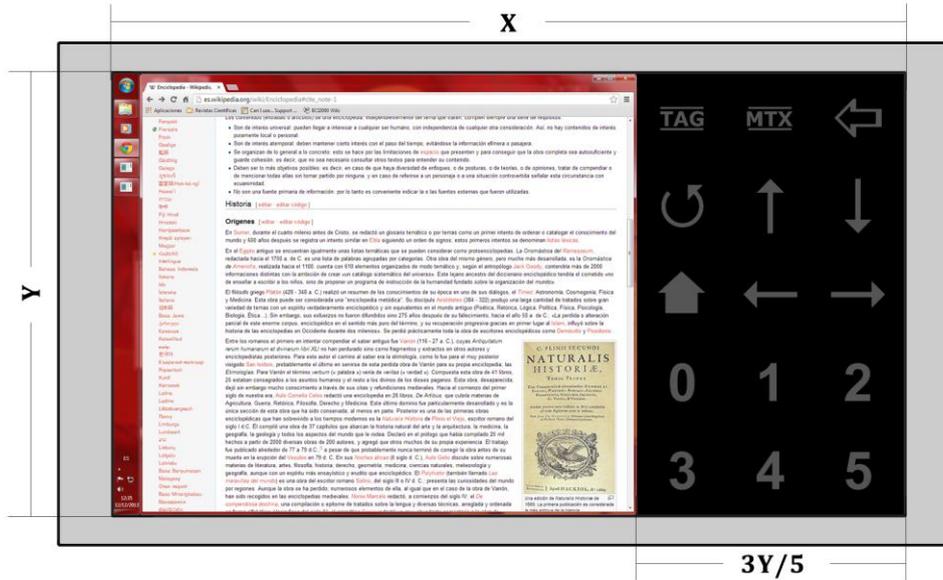


Figura 4-6. Optimización de la aplicación para pantallas panorámicas con relación de aspecto 16:9.

en una serie de fases cuya duración programa el usuario en función de la finalidad que tenga cada sesión. A continuación se exponen los parámetros involucrados y en la Figura 4-7 se muestra la línea de tiempo de cada intento.

- **PreRunDuration.** Establece la pausa que inicia el intento.
- **PreSequenseDuration.** Establece la pausa que precede a la primera secuencia. Se considera como una secuencia la iluminación de todas las filas y de todas las columnas que componen la matriz y, por tanto, el número de secuencias de cada intento es equivalente al número de veces que se va a iluminar cada fila y cada columna antes de determinar la selección del usuario. Por ejemplo, para la matriz de navegación, de tamaño 5x3, una secuencia se corresponde con “5+3 = 8” iluminaciones.
- **StimulusDuration.** Establece la duración del estímulo visual, es decir, cada iluminación.
- **ISIMinDuration, ISIMaxDuration.** Establecen la duración mínima y máxima del intervalo entre estímulos, respectivamente. Especifican el tiempo que pasa entre dos iluminaciones, siendo éste aleatorio, con una distribución uniforme para los valores intermedios.
- **PostSequenseDuration.** Establece la pausa que sigue a la última secuencia del intento.
- **PostRunDuration.** Establece la pausa que finaliza el intento.
- **NumberOfSequences.** Número de secuencias que compondrán un solo intento. Este valor debe ser el mismo que el del parámetro *EpochsToAverage*, el cual determina el número de épocas que deben considerarse en el promediado sincronizado (la duración de las mismas la establece *EpochLength*).
- **Threshold.** Indica el umbral óptimo de cada usuario para determinar si se debe considerar o no la selección realizada.

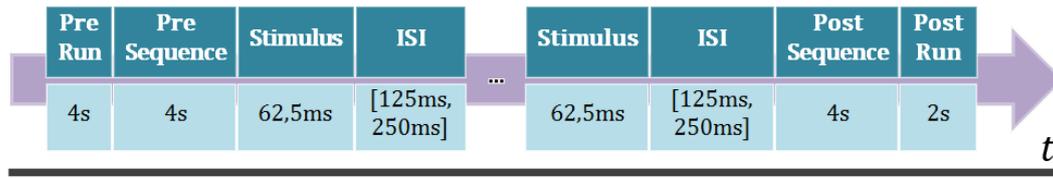


Figura 4-7. Línea de tiempo de cada intento de *BCI Browser: P3Speller*.

4.5.3. Matrices *odd-ball*

A causa del gran número de comandos necesarios para poder experimentar una navegación web libre de restricciones se han desarrollado dos matrices *odd-ball* distintas que se intercalan entre ellas a voluntad del usuario. Estas matrices son la matriz de navegación y la matriz de teclado.

La matriz de navegación es una matriz pequeña, compuesta por 5 filas y 3 columnas, con 15 celdas en total. En esta matriz se incluyen únicamente los comandos que afectan a la navegación web, como los *scrolls*, el retroceso de página, la actualización del navegador, etcétera. El pequeño tamaño de esta matriz provoca que la navegación a través de las distintas páginas web sea rápida debido a que se necesitan un menor número de iluminaciones para seleccionar cada comando deseado, por esta razón se han omitido la mayor parte de los caracteres alfanuméricos habituales, más útiles para la redacción que para la navegación web. En la Figura 4-8 aparece la matriz de navegación y a continuación se explican los distintos comandos que se contiene la misma.

- (1) El primer comando, denominado "TAG", es un conmutador que permite activar o desactivar el etiquetado de nodos (hipervínculos y formularios) por parte de *BCI Browser: Navigator*. De esta manera, el usuario puede leer tranquilamente una web sin etiquetas y activarlas cuando desee navegar a través de los hipervínculos o introducir algo en los formularios.
- (2) El segundo comando, denominado "MTX", es otro conmutador que permite alternar entre la matriz de navegación y la matriz de teclado.
- (3) El tercer comando ("↶", *back*) permite retroceder a la última web visitada.
- (4) El cuarto comando ("↻", *reload*) permite actualizar la web que se está visualizando.
- (5) El quinto comando ("↑", *scroll up*) permite mover la barra de desplazamiento hacia arriba, equivalente a una pulsación de teclado habitual denominada "Re Pág", o retroceso de página.
- (6) El sexto comando ("↓", *scroll down*) permite mover la barra de desplazamiento hacia abajo, equivalente a una pulsación de teclado habitual denominada "Av Pág", o avance de página.
- (7) El séptimo comando ("🏠", *home*) permite ir a la página de inicio.
- (8) El octavo comando ("←") es, posiblemente, el que más funciones puede realizar en toda la aplicación. Este comando varía su función dependiendo del estado del programa: si el conmutador "TAG" no está activo y, por tanto, no se ha activado el etiquetado de los nodos de la web, la función de "←" es la de un desplazamiento horizontal de la web hacia la izquierda (*scroll left*); por el contrario, si el conmutador "TAG" sí está activo y los nodos están etiquetados, la función del comando variará dependiendo de si el cursor se

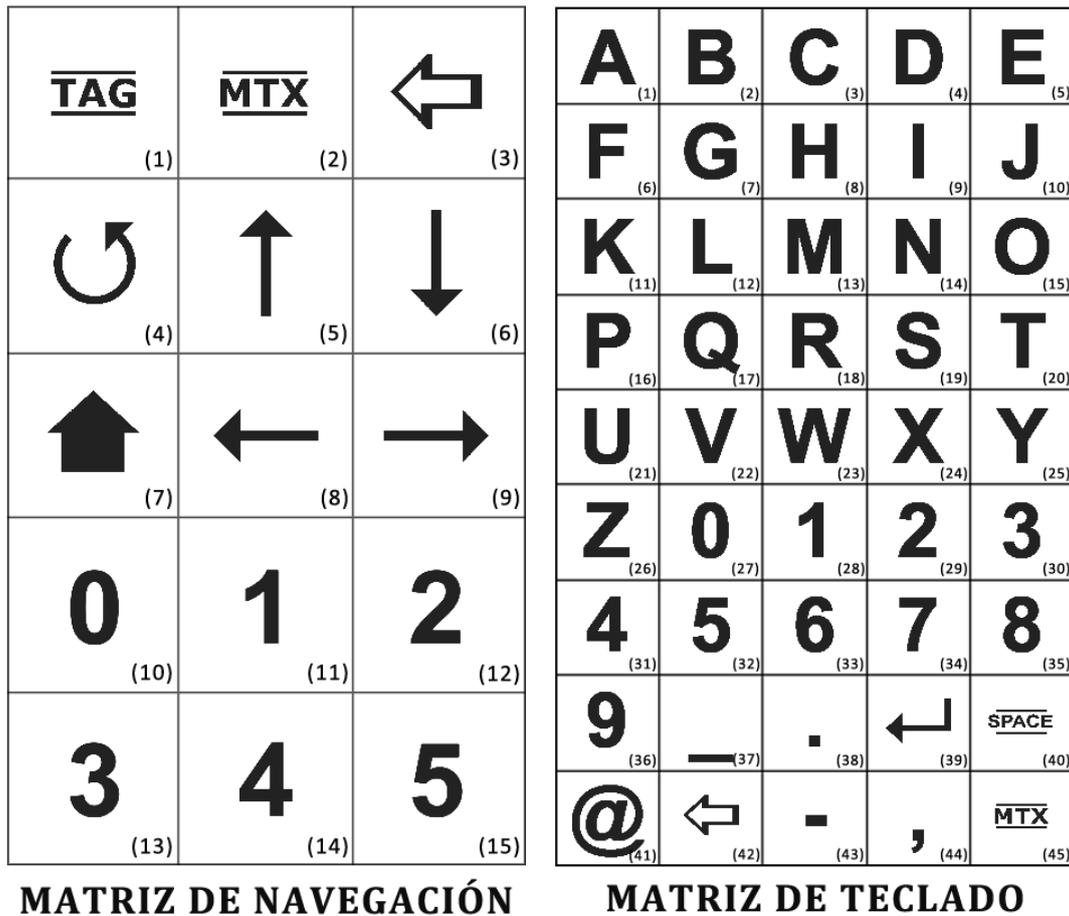


Figura 4-8. Matrices utilizadas en la aplicación *BCI Browser: P3Speller*. A la izquierda se muestra la matriz de navegación, a la derecha, la matriz de teclado.

encuentra dentro de un formulario o no. Si no se ha seleccionado ningún formulario, la función de “←” será la de borrar el último carácter introducido en el *buffer*, sin embargo, si el cursor se encuentra dentro de un formulario, “←” permitirá abandonar el mismo para poder seleccionar otro nodo distinto. Las funciones de “←” vienen recogidas en el diagrama de flujo de la Figura 4-9.

- (9) El noveno comando (“→”, *scroll right*) permite mover la barra de desplazamiento hacia la derecha.
- (10), (11), (12), (13), (14) y (15) Los siguientes comandos simplemente envían una pulsación de teclado de los cinco posibles números que se utilizan para codificar los nodos en *BCI Browser: Navigator*. La finalidad de incluir estos caracteres alfanuméricos en la matriz de navegación no es otra que agilizar las selecciones de nodos, tal y como se ha explicado anteriormente.

A diferencia de la matriz de navegación, la matriz de teclado es una matriz grande, compuesta por 5 columnas y 9 filas, resultando en 45 celdas en total. Esta matriz está especialmente destinada para redactar o rellenar formularios. Cuenta con la mayor parte de caracteres alfanuméricos y con los símbolos más utilizados en la jerga de Internet. Atendiendo a la Figura 4-8, desde el comando (1) hasta el (26) se corresponden con las letras del alfabeto latino y simplemente se encargan de enviar las pulsaciones de teclado adecuadas para cada una. Lo mismo ocurre con todos los

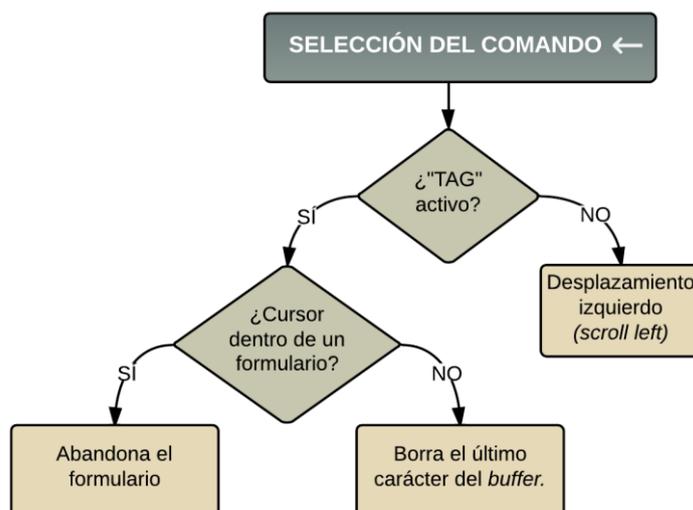


Figura 4-9. Funciones del comando “←” de la matriz de navegación en función del estado de la aplicación en el momento de su selección.

números, desde el (27) hasta el (36). De ahí en adelante encontramos diversos símbolos.

(37), (38), (39), (40), (41), (42), (43), y (44) Envía las pulsaciones de teclado correspondientes a la barra baja (“_”, *underscore*), al punto (“.”, *point*), al retorno (“↵”, *enter*), a la barra espaciadora (“ ”, *spacebar*), a la arroba (“@”, *at*), al retroceso (“↶”, *backspace*), al guión (“-”, *dash*) y a la coma (“,”, *comma*), respectivamente.

(45) Aparece el comando “MTX” de nuevo, conmutador que permite alternar entre la matriz de navegación y la matriz de teclado.

La matriz predeterminada al iniciar la aplicación es la matriz de navegación. No obstante, indistintamente de cuál sea la matriz que está activada en cada momento, ésta iluminará continuamente sus filas y sus columnas y no parará a no ser que se detenga la aplicación. Esto hace necesario la introducción de un umbral de decisión que establezca cuándo una pulsación de teclado debe enviarse a *BCI Browser: Navigator* y cuándo no. Este umbral se basa en las curvas ROC explicadas anteriormente.

4.6. BCI Browser: Navigator

Tal y como se ha explicado anteriormente, *BCI Browser: Navigator* se encarga de procesar los comandos de teclado recibidos del *BCI Browser: P3Speller* y traducirlos en comandos del navegador *Google Chrome*.

Para ello se ha desarrollado una extensión del mismo en lenguaje JavaScript (cuyo código se encuentra en el Anexo B.1) que interpreta las pulsaciones de teclado y actúa en consecuencia. La función más relevante de la extensión es la de etiquetar todos los hipervínculos y formularios existentes en la página web que se está visualizando con uno o varios caracteres numéricos. El usuario podrá seleccionar el hipervínculo o formulario deseado introduciendo esos caracteres, consiguiendo así una navegación libre controlada plenamente por el usuario.

Para facilitar la fluidez de la aplicación, el número de caracteres numéricos que se utilizan para etiquetar los hipervínculos y los formularios, considerados ambos como *nodos* de la web, depende de la página que se esté visualizando. En primer lugar,

la aplicación calcula el número de nodos que contiene la web y codifica éstos con un número de caracteres que permita etiquetarlos todos utilizando únicamente los números 0, 1, 2, 3, 4 y 5.

Una vez codificados y etiquetados se inicializa un *buffer* en la parte superior izquierda del navegador que servirá de realimentación al usuario para poder visualizar los caracteres que escribe mientras intenta seleccionar un nodo determinado. Además de indicar qué caracteres se han escrito anteriormente, el *buffer* indica qué nodos comienzan con la misma codificación, proveyendo al sistema de una realimentación intuitiva y a tiempo real. En la Figura 4-10 se muestra un ejemplo en el que el usuario ha introducido un “0”, el *buffer* lo indica e ilumina los nodos que empiezan con esa codificación y, por tanto, pueden ser selecciones en potencia.

Como medida preventiva para aumentar la rapidez de la selección de nodos, la aplicación simula la introducción de un “ENTER” cuando el usuario ha introducido en el *buffer* el número necesario de caracteres para seleccionarlo, evitando así la creación de un comando adicional para confirmar la introducción de teclas en el *buffer* y disminuyendo el tiempo necesario para realizar la selección. Adicionalmente, el *buffer* permite borrar los caracteres introducidos si el usuario ha fallado en la selección del comando deseado.

Además de las funciones descritas, la activación o no de la extensión también está gobernada por el usuario. El usuario decide si activar el etiquetado de nodos o no activarlo, puesto que éstos pueden ser molestos si el usuario desea leer tranquilamente una página web que está cargada de hipervínculos.



Figura 4-10. Etiquetado de hipervínculos con *BCI Browser: Navigator*. En rojo se indica la posición del *buffer* que contiene el carácter introducido por el usuario, en verde los nodos posibles que concuerdan con la codificación introducida (o posibles selecciones en potencia) y en gris el resto de nodos que no pueden seleccionarse con esa codificación.

4.7. Sujetos y tareas

La aplicación se ha evaluado por parte de cinco sujetos varones, sanos y cuyas edades comprenden desde los 21 años hasta los 33, durante tres sesiones distintas. Durante las tres sesiones los parámetros relativos a la duración de las fases de cada intento se han mantenido constantes, siendo éstos los que aparecen en las celdas inferiores de la Figura 4-7. Los electrodos utilizados aparecen en la Figura 4-2 y su localización tampoco ha sido modificada entre sesiones, al igual que la frecuencia de muestreo, manteniéndose en 256Hz. A continuación se describe el objetivo y las especificaciones de la aplicación para cada sesión, dispuestas en días distintos.

4.7.1. Primera sesión

El primer objetivo de la primera sesión fue determinar los pesos del clasificador para cada usuario, utilizando el algoritmo SWLDA. Este entrenamiento se realizó con la matriz de mayor tamaño, puesto que un clasificador adaptado para una matriz amplia servirá perfectamente para una matriz más pequeña, sin embargo, si se realizase al contrario probablemente se obtendrían errores de selección.

Se ordenó a los usuarios intentar escribir una palabra de seis letras cuatro veces con la matriz de teclado, es decir, se emplearon 4 rondas con 6 intentos cada una de ellas. El número de secuencias fue de 15, en otras palabras, cada fila y cada columna se iluminaban 15 veces; y la duración de cada ronda se encontraba en torno a los 6 minutos. Durante este proceso el usuario debía mirar fijamente a la letra que debía seleccionar. Las especificaciones aparecen recogidas en la Tabla 4-1.

Una vez completadas todas las rondas se estableció el clasificador de cada usuario gracias a la herramienta *P300Classifier*, que provee el sistema BCI2000. Para ello se cargan los 4 ficheros *.dat* de cada ronda realizada, generados para cada usuario, y se calcula el vector de pesos. Además de proveer el vector de pesos, la herramienta también establece el número mínimo de secuencias para que un determinado usuario pueda seleccionar cualquier celda satisfactoriamente, pudiendo resultar inferior a 15. En la Figura 4-11 aparece una captura de la herramienta utilizada.

Adicionalmente, una vez entrenado el clasificador del usuario, se prosiguió con el siguiente objetivo de la sesión: adquirir muestras para determinar el umbral del usuario. Como ya es conocido, el umbral óptimo varía entre usuarios y también varía, en menor medida, entre sesiones. Por esta razón se adquirieron las muestras de entrenamiento requeridas para calcularlo en dos sesiones distintas, en la primera y en la segunda sesión.

Estas muestras de entrenamiento se obtuvieron ordenando a los usuarios realizar dos tareas distintas. La primera tarea consistió en tratar de seleccionar una serie de comandos objetivo con la matriz de navegación, concretamente “↓ → ↑ ∪ 5”. Se registraron 4 rondas, con 6 intentos cada una (los seis comandos) y 10 secuencias por intento. Nótese que el número de secuencias ha disminuido porque en esta prueba los usuarios ya estaban utilizando su propio clasificador.

TABLA 4-1. ESPECIFICACIONES DE LA PRIMERA SESIÓN, UTILIZADA PARA ENTRENAR EL CLASIFICADOR.

PRIMERA SESIÓN				
RONDAS	INTENTOS	SECUENCIAS	TAMAÑO MTX	FILTRO
4	6	15	9x5	CAR

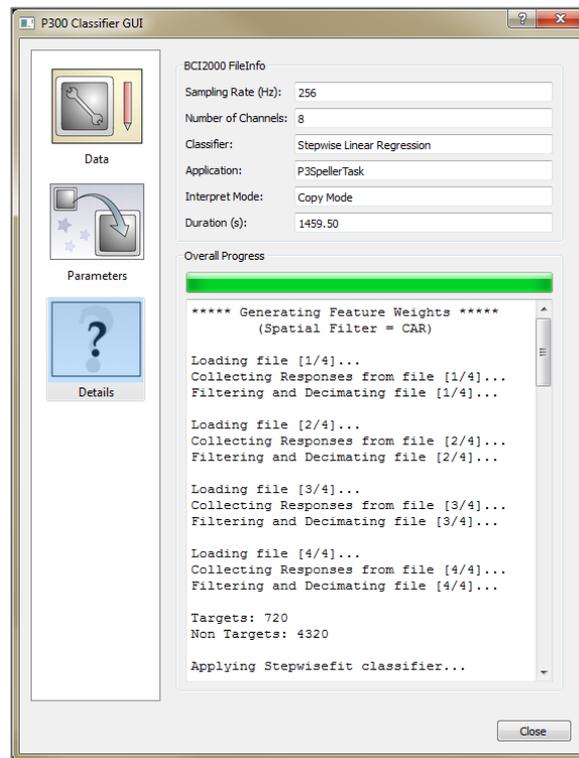


Figura 4-11. Interfaz gráfica de la herramienta *P300Classifier*, utilizada para hallar el vector de pesos adaptado para cada usuario a partir de una serie de muestras de entrenamiento.

La segunda tarea consistió en leer una página web mientras la matriz de navegación seleccionaba comandos de manera aleatoria, es decir, se ordenó al usuario ignorar los estímulos, para ver así la intensidad con la que se recibían éstos y ver las probabilidades de selección con las que trabajaba el clasificador. Se realizaron también 4 rondas con 6 intentos y 10 secuencias por intento. Las especificaciones de esta sesión post-clasificador aparecen recogidas en la tabla 4-2, puesto que son las mismas que se utilizaron en la segunda sesión.

4.7.2. Segunda sesión

El objetivo de la segunda sesión fue determinar el umbral de decisión de cada usuario. Para ello se ordenó a los usuarios volver a realizar la prueba de la sesión 1 post-clasificador, atendiendo a los estímulos e ignorándolos durante 4 rondas, con 6 intentos y 10 secuencias por intento, tal y como se ha comentado en el anterior punto. En la Tabla 4-2 aparecen las especificaciones de este tipo de tarea.

Una vez obtenidas las 8 rondas de estímulo atendido e ignorado, y considerando los datos de ambos días, se trató de estimar la intensidad de la celda seleccionada en cada intento. Para ello, se extrajeron las probabilidades de selección con las que trabaja el módulo de “Procesamiento de la señal” en BCI2000, relacionadas con el vector de pesos de cada usuario. Una vez extraídas se representó gráficamente cada selección y se halló la curva ROC de los datos para cada sesión, tal y como se muestra en la Figura 4-12. Una vez estimado el umbral óptimo para cada sesión se realizó un promedio entre ambos, resultando en el umbral final del usuario, como expresa la ecuación (4.2).

$$Umbral\ final = (UmbralROC_{S1} + UmbralROC_{S2})/2 \quad (4.2)$$

4.7.3. Tercera sesión

El objetivo de la tercera y última sesión fue evaluar completamente el funcionamiento de la aplicación utilizando el umbral y el clasificador óptimos para cada usuario. Para ello se establecieron cuatro tareas con una dificultad progresiva que probaron todas las funcionalidades de la aplicación.

La primera tarea consistía en la visualización simple de una web. El sujeto debía centrar su atención en visualizar una página web haciendo uso de los comandos de la matriz de navegación, además de practicar la selección de hipervínculos. La tarea partía del artículo de la *Wikipedia* sobre los potenciales evocados P300 y, posteriormente, el sujeto debía hacer *scroll* dos veces hacia abajo y hacia arriba, seleccionar un hipervínculo y cambiar de web a través de él. Finalmente, se ordenó al usuario volver a la página anterior. Las selecciones mínimas para realizar esta tarea fueron 10 selecciones, sin embargo, si el usuario erraba debía solventarlo, causando un aumento de las selecciones necesarias para finalizar la prueba exitosamente. Además de eso, es posible que el programa indicase que no se había superado el umbral en alguna selección concreta y, por tanto, debería repetirla para intentar sobrepasar la intensidad necesaria (indicado como un *warning* en el log).

La segunda tarea requería más selecciones que la anterior, concretamente, 14 selecciones. Aun así, para realizarla solamente se necesitaba utilizar la matriz de navegación y, en consecuencia, el tiempo necesario para finalizar la prueba no aumentó de manera significativa con respecto a la primera. El sujeto debía centrar su atención en visualizar un e-mail de su bandeja de entrada y, finalmente, responderlo con un cuerpo de mensaje corto. La tarea partía de la versión HTML básica de *Gmail*, donde el sujeto debía seleccionar el e-mail objetivo, leerlo, seleccionar el formulario de respuesta rápida, introducir un cuerpo de mensaje corto, abandonar el formulario y, finalmente, seleccionar el botón de enviar.

TABLA 4-2. ESPECIFICACIONES DE LA SEGUNDA SESIÓN, UTILIZADA PARA HALLAR EL UMBRAL ÓPTIMO.

SEGUNDA SESIÓN				
RONDAS	INTENTOS	SECUENCIAS	TAMAÑO MTX	OBJETIVO
4	6	10	5x3	↓ → ↑ ∪ 5

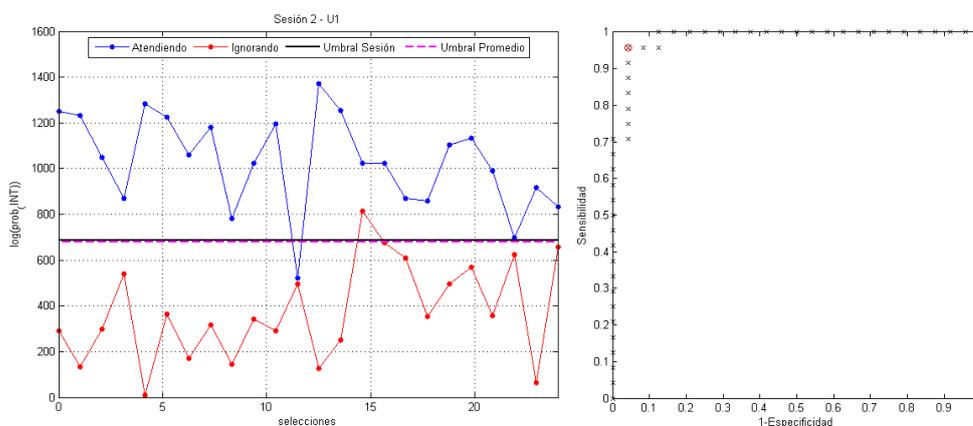


Figura 4-12. Representación gráfica de los datos de la segunda sesión para el sujeto U1. A la izquierda aparecen las selecciones atendiendo e ignorando los estímulos para las 24 selecciones de la segunda sesión. A la derecha aparece la curva ROC para esa sesión concreta y el umbral óptimo calculado.

La tercera tarea fue la más complicada de todas. Aunque solamente requería 13 selecciones, varias de éstas solamente podían realizarse con la matriz de teclado y, dado que tiene un mayor número de celdas, el tiempo necesario para finalizar la prueba aumentó considerablemente. El usuario debía realizar una búsqueda en *Google* introduciendo el acrónimo "EEG" y seleccionar el primer resultado encontrado. Para ello debía seleccionar el formulario de búsqueda, introducir los caracteres alternando entre ambas matrices y seleccionar el primer hipervínculo activando el etiquetado.

La cuarta y última tarea estaba destinada a comprobar el funcionamiento del umbral óptimo de cada usuario. Se ordenó a cada usuario leer una página web durante 5 minutos mientras la aplicación estaba en curso, ignorando los estímulos, con el fin de comprobar si se seleccionaban comandos o no. En todas estas tareas se utilizaron 10 secuencias, como en la segunda sesión.

Para finalizar, se entregó a los cinco usuarios un cuestionario de satisfacción, con el fin de recoger su experiencia personal y las sugerencias para mejorar la aplicación en un futuro. Este cuestionario y la hoja entregada a los usuarios donde aparecían las tareas propuestas se recogen en el Anexo C.

En la Figura 4-13 aparece una imagen de un sujeto realizando la tercera prueba propuesta y en la Tabla 4-3 se recoge la edad de cada uno de los cinco sujetos. En el capítulo siguiente se muestran los resultados de esta evaluación para cada uno de los usuarios.

TABLA 4-3. EDAD DE LOS CINCO SUJETOS MASCULINOS.

EDAD DE LOS SUJETOS				
U1	U2	U3	U4	U5
21 años	25 años	21 años	30 años	33 años

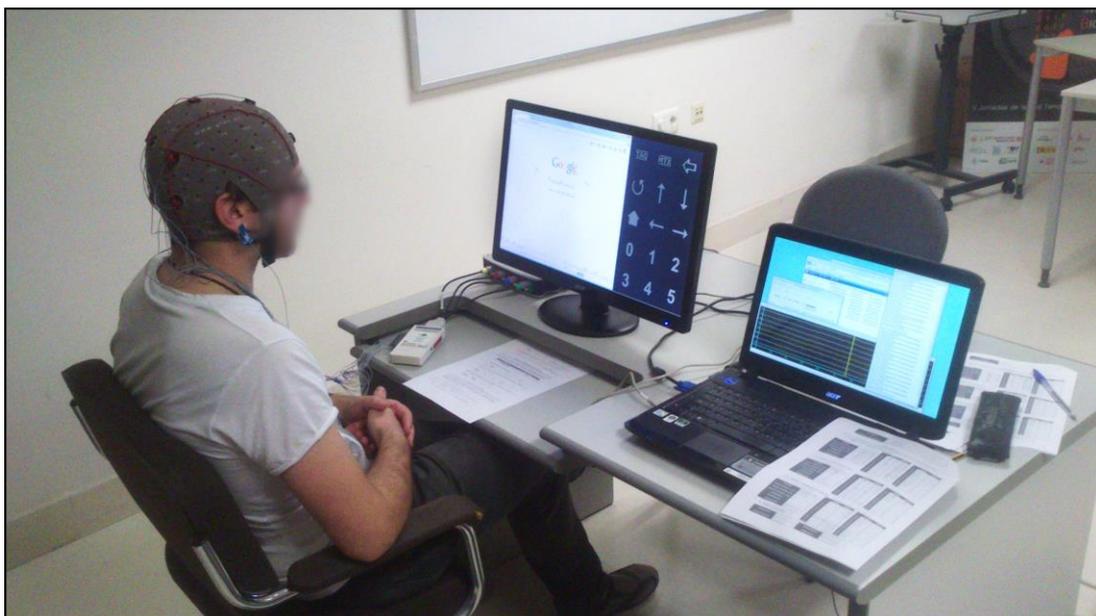


Figura 4-13. El sujeto U5 evaluando la aplicación en la tercera sesión. La pantalla de la izquierda contiene la aplicación mientras que la de la derecha muestra el voltaje de cada electrodo y la información sobre las celdas seleccionadas y el umbral.

5.1. Determinación del umbral

Los primeros resultados a presentar son los obtenidos de la determinación del umbral óptimo para cada sujeto, siguiendo las pautas descritas en el capítulo anterior, para las dos primeras sesiones de cada usuario.

5.1.1. Sujeto U1

El primer sujeto, identificado como sujeto U1, obtuvo unos resultados perfectos para la primera sesión de determinación del umbral. La intensidad de las selecciones atendiendo e ignorando los estímulos no se superpusieron en ningún momento, causando una curva ROC óptima, con un valor diagnóstico perfecto. Por tanto, la precisión, la sensibilidad, la especificidad y el área bajo la curva para un umbral situado en 673.315 fueron máximas.

No obstante, los resultados de la segunda sesión no fueron tan altos, quizás por una variación en la colocación de los electrodos sobre el cuero cabelludo, falta de motivación, etcétera. En la segunda sesión se obtuvo una precisión, una sensibilidad y una especificidad del 95.83% para un umbral situado en 687.994. Aunque los resultados fuesen un poco peores, una precisión del 90% para ese umbral óptimo implica un buen funcionamiento del clasificador hallado anteriormente.

El umbral promedio se situó en 680.6545, no muy alejado de los otros dos. Estos resultados vienen recogidos en la Tabla 5-1 y en la Figura 5-1, a continuación.

TABLA 5-1. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U1.

SUJETO U1	SESIÓN 1	SESIÓN 2
Umbral óptimo	673,315	687,994
Precisión	100%	95,83%
Sensibilidad	1	0,9583
Especificidad	1	0,9583
Área ROC	1	0,9844
PPV	1	0,9583
NPV	1	0,9583
Umbral Promedio	680,6545	

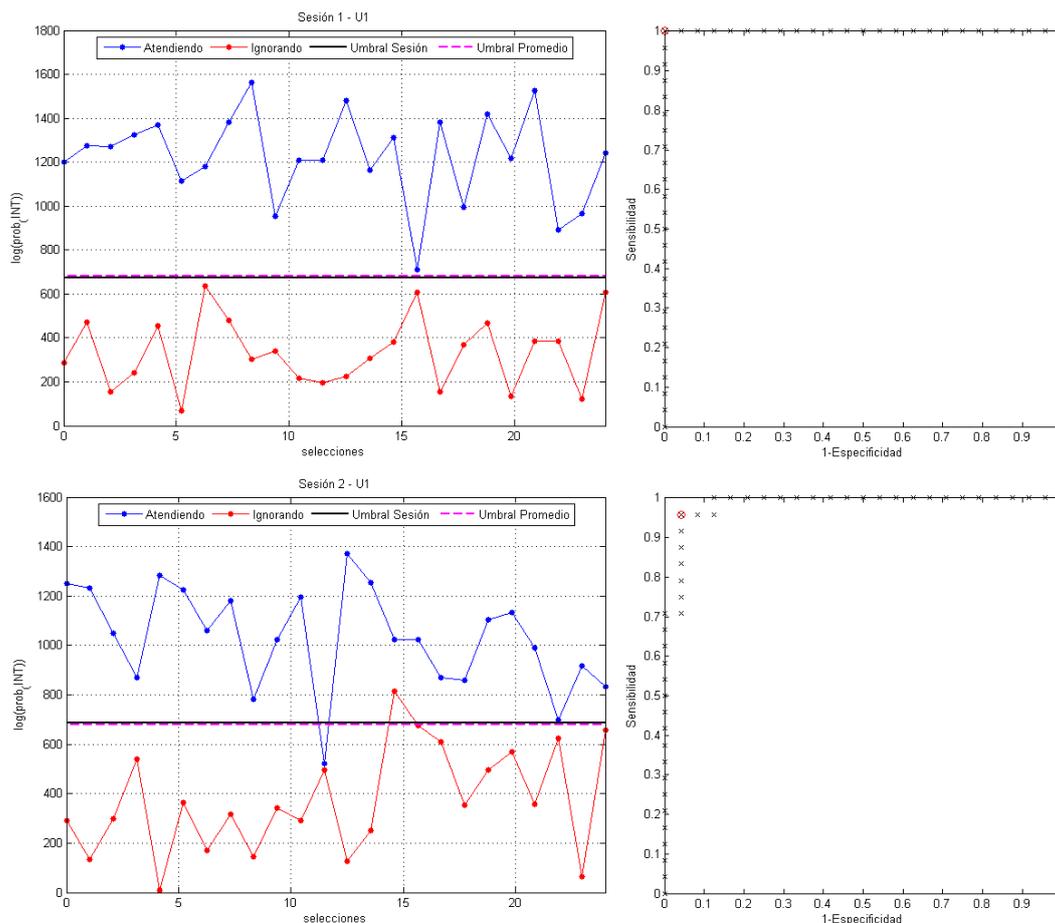


Figura 5-1. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U1. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión.

5.1.2. Sujeto U2

El segundo sujeto, identificado como sujeto U2, obtuvo unos resultados impecables para ambas sesiones. La curva ROC de ambas sesiones fue perfecta, causando una precisión del 100% y una sensibilidad y especificidad máximas.

Los umbrales óptimos para los cuales se consiguieron estos resultados perfectos fueron de 574.7595 para la primera sesión y de 463.043 para la segunda sesión. Estos resultados ponen de manifiesto que el clasificador creado para el sujeto U2 es muy robusto y distingue perfectamente la ausencia o no de atención en los estímulos presentados.

En la Figura 5-2 se puede observar que la diferencia de intensidad entre el punto máximo obtenido para la ausencia de atención y el punto mínimo cuando el sujeto U2 mostraba atención es notable. Sin embargo, aunque en la segunda sesión se mantenga una discriminación perfecta, se observa que la distancia se ha reducido drásticamente. Este resultado lleva a pensar que, probablemente, la curva ROC no sería perfecta si se hubiesen realizado más que cuatro rondas para cada tipo de estimulación.

El umbral promedio se situó en 518.90125, tal y como muestran la Tabla 5-2 y la Figura 5-2.

TABLA 5-2. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U2.

SUJETO U2	SESIÓN 1	SESIÓN 2
Umbral óptimo	574,7595	463,043
Precisión	100%	100%
Sensibilidad	1	1
Especificidad	1	1
Área ROC	1	1
PPV	1	1
NPV	1	1
Umbral Promedio	518,90125	

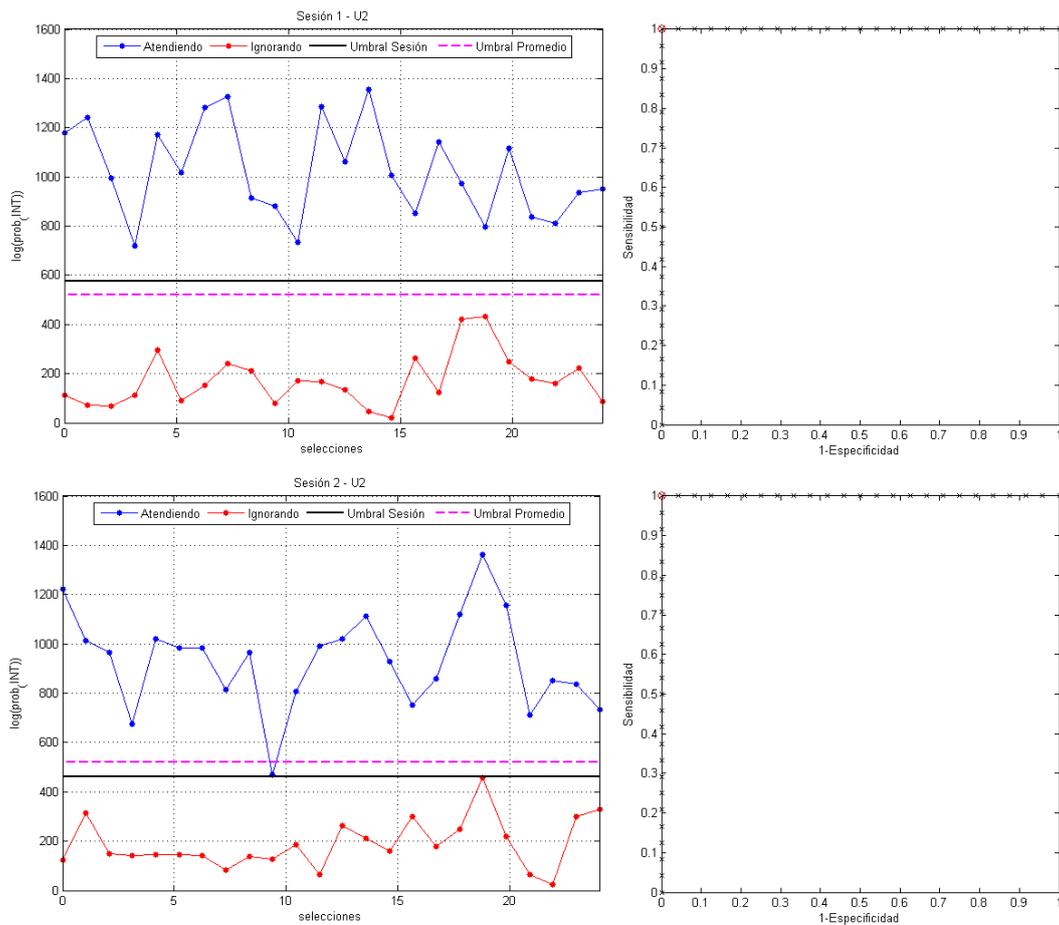


Figura 5-2. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U2. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión.

5.1.3. Sujeto U3

El tercer sujeto, identificado como sujeto U3, obtuvo un resultado perfecto para la segunda sesión y un resultado excelente para la primera.

En la primera sesión no alcanzó el valor óptimo, obteniendo un valor de 97.9167% de precisión para un umbral situado en 0.023319, con un alto valor de sensibilidad, especificidad y de área bajo la ROC. Nótese que el valor del umbral es muy inferior al valor utilizado para el resto de usuarios. Este fenómeno depende únicamente de los pesos asignados en su clasificador, hartos dispares al compararlos con el resto de sujetos. No obstante, la precisión obtenida asegura que el funcionamiento de su clasificador es ejemplar, seleccionando correctamente el 97% de las veces. Según advierte la Figura 5-3, la especificidad para la primera sesión fue máxima, es decir, el umbral óptimo clasificó correctamente todas las selecciones atendidas, fallando únicamente en la clasificación de selecciones cuyo estímulo ha sido ignorado. Realmente sería más apropiado lograr un valor máximo de sensibilidad, en lugar de obtenerlo para la especificidad; la razón reside en que, para esta aplicación concreta, se podría considerar más importante que la aplicación no seleccione comandos cuando el usuario está ignorando los estímulos y, por tanto, cueste más seleccionarlos al atenderlos, que clasificar correctamente los estímulos atendidos y no asegurar que no se seleccione ninguno al ignorarlos.

Tal y como se ha comentado anteriormente, el resultado obtenido para la segunda sesión fue perfecto para un umbral de 0.031577, causando valores máximos en la precisión (siendo 100%), sensibilidad, especificidad, área bajo la ROC, etcétera.

El umbral promedio se situó, por tanto, en 0.027448, tal y como se puede observar en la Tabla 5-3 y la Figura 5-3. Esta vez la precisión de los decimales utilizados al introducir el umbral en el parámetro "Threshold" es primordial, puesto que el umbral se encuentra, aproximadamente, tres órdenes de magnitud por debajo del resto de umbrales hallados para los otros dos sujetos.

TABLA 5-3. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U3.

SUJETO U3	SESIÓN 1	SESIÓN 2
Umbral óptimo	0,023319	0,031577
Precisión	97,9167%	100%
Sensibilidad	0,9583	1
Especificidad	1	1
Área ROC	0,9549	1
PPV	1	1
NPV	0,9600	1
Umbral Promedio	0,027448	

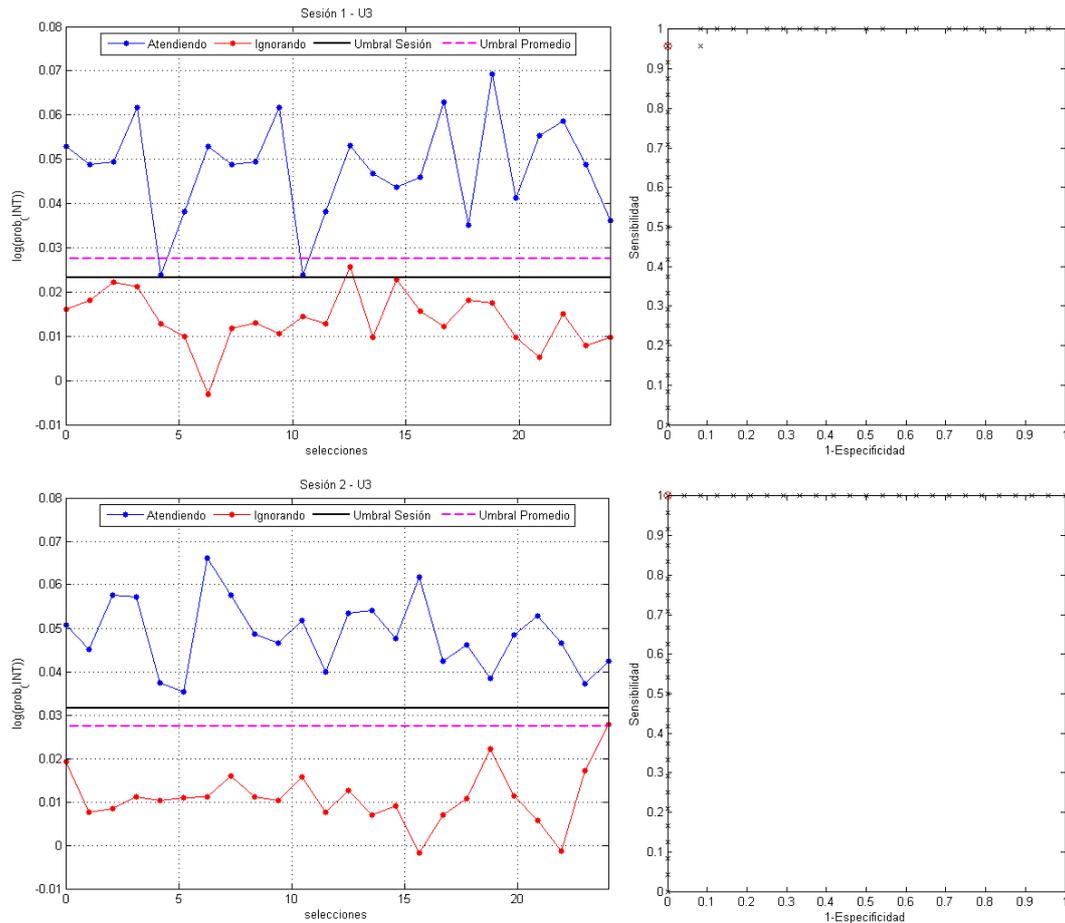


Figura 5-3. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U3. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión.

5.1.4. Sujeto U4

El cuarto sujeto, identificado como sujeto U4, obtuvo un resultados perfectos para ambas sesiones, causando dos curvas ROC con valor diagnóstico perfecto, y obteniendo el 100% de precisión y los valores máximos de sensibilidad y especificidad para los umbrales 724.471 y 901.747 en la primera y la segunda sesión, respectivamente.

Tal y como se comentó en los resultados del sujeto U2, éstos ponen de manifiesto que el clasificador creado para este sujeto es muy robusto y distingue perfectamente la ausencia o no de atención de los estímulos presentados. En esta ocasión el umbral promedio se encuentra un poco por encima del obtenido para el sujeto U2, concretamente, en 813.109 (Tabla 5-4).

De hecho, observando las gráficas de la Figura 5-4, se puede considerar que los resultados del sujeto U4 son mejores que los del sujeto U2, puesto que la diferencia entre el valor mínimo atendiendo y el valor máximo ignorando los estímulos es mucho mayor que para éste último. Este hecho determina que, probablemente, si se hubiesen realizado más sesiones de entrenamiento, es posible que las curvas ROC obtenidas fuesen perfectas igualmente.

TABLA 5-4. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U4.

SUJETO U4	SESIÓN 1	SESIÓN 2
Umbral óptimo	724,471	901,747
Precisión	100%	100%
Sensibilidad	1	1
Especificidad	1	1
Área ROC	1	1
PPV	1	1
NPV	1	1
Umbral Promedio	813,109	

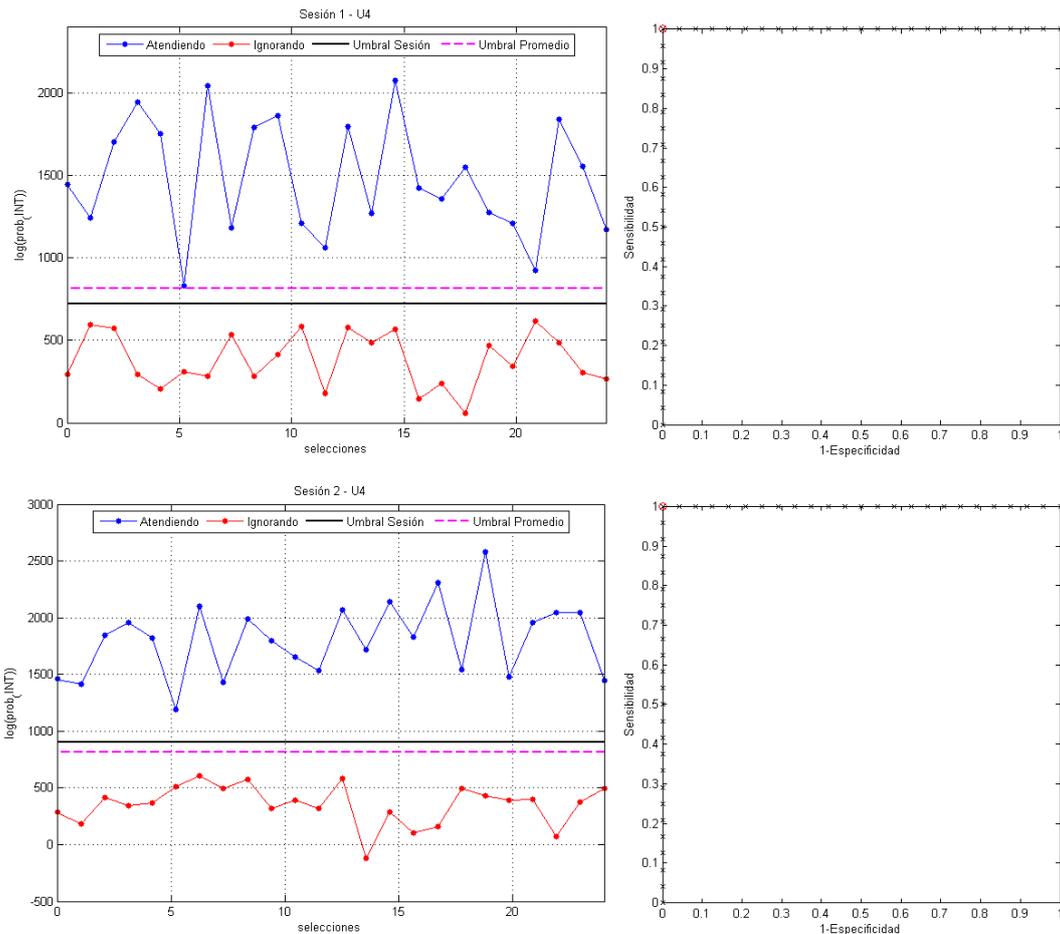


Figura 5-4. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U4. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión.

5.1.5. Sujeto U5

El quinto y último sujeto, identificado como sujeto U5, obtuvo buenos resultados para ambas sesiones, sin embargo, no obtuvo ninguna curva ROC perfecta.

Para la primera sesión, el sujeto U5 obtuvo una precisión del 97.9167% en un umbral situado en 554.361. Además, la sensibilidad obtenida es máxima, produciendo que todos los estímulos ignorados se clasifiquen correctamente y fallando, únicamente, en la clasificación de los estímulos atendidos, puesto que la especificidad es de 0.9583. Tal y como se comentó en el caso del sujeto U3, obtener una sensibilidad máxima podría resultar mucho más útil que obtener una alta especificidad para esta aplicación en concreto, así que puede considerarse como un resultado muy satisfactorio.

En la segunda sesión se obtuvo una precisión del 93.75% para un umbral situado en 457.951. También es destacable el valor máximo obtenido para la especificidad, al contrario que en la sesión anterior. Este valor implica que todos los estímulos atendidos se clasifican correctamente, fallando únicamente en los estímulos ignorados. Atendiendo a este hecho, se considera peor este resultado que el obtenido en la sesión anterior, aunque bajo ninguna circunstancia se puede considerar un resultado desfavorable.

El umbral promedio se situó en 506.156, según advierten la Tabla 5-5 y la Figura 5-5.

5.2. Tareas propuestas

Los siguientes resultados a presentar son los obtenidos de la evaluación de la aplicación por parte de los sujetos. Para ello se realizaron cuatro tareas distintas: visualización de una web, envío de un e-mail, búsqueda en *Google* y lectura de una página web.

Estos resultados se presentan organizados por sujetos, tal y como se ha hecho en el apartado anterior con los resultados sobre la determinación del umbral y

TABLA 5-5. RESULTADOS OBTENIDOS PARA LA DETERMINACIÓN DEL UMBRAL DEL SUJETO U5.

SUJETO U5	SESIÓN 1	SESIÓN 2
Umbral óptimo	554,361	457,951
Precisión	97,9167%	93,75%
Sensibilidad	1	0,875
Especificidad	0,9583	1
Área ROC	0,9948	0,9826
PPV	0,9600	1
NPV	1	0,8
Umbral Promedio	506,156	

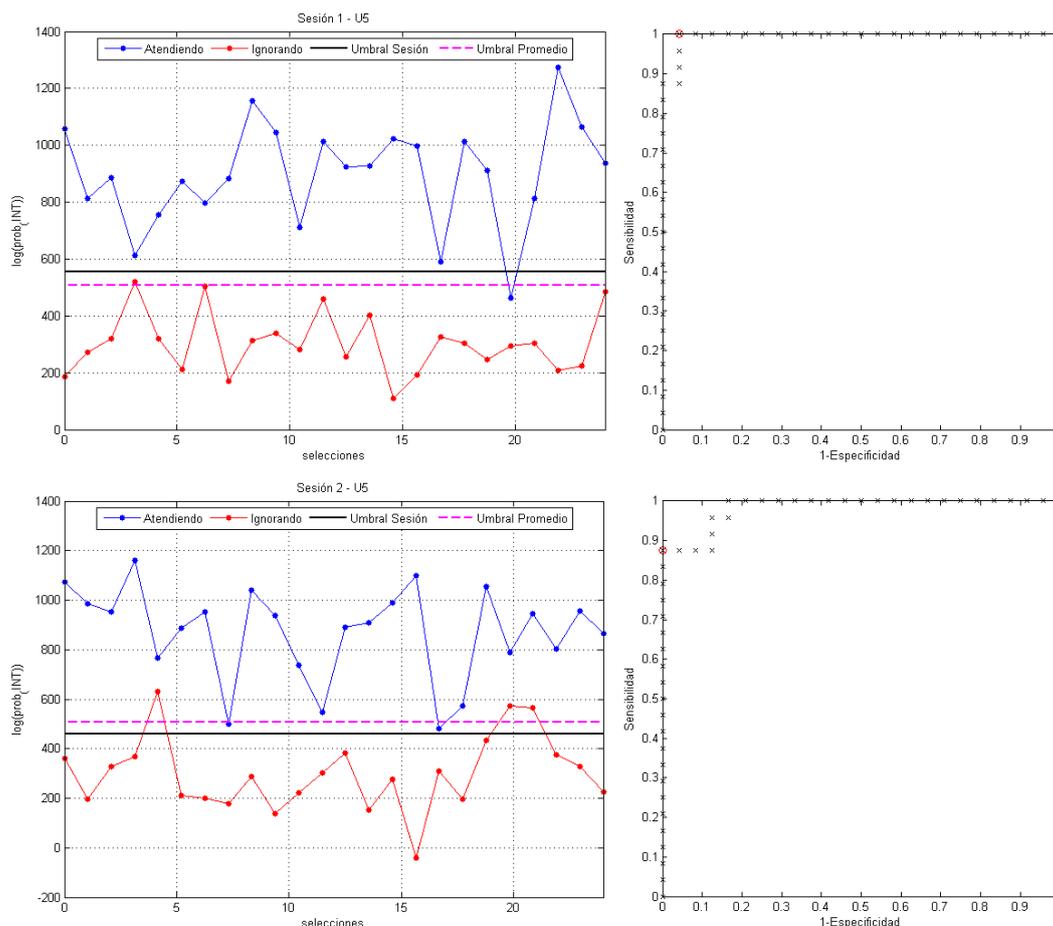


Figura 5-5. Resultados de la determinación del umbral en las dos primeras sesiones para el sujeto U5. A la izquierda aparece la intensidad de las selecciones y a la derecha la curva ROC para cada sesión.

organizados por tablas con los datos necesarios para realizar correctamente la evaluación. Estos datos son los siguientes:

1. **Selecciones óptimas:** número de selecciones mínimas necesarias para terminar la tarea propuesta.
2. **Selecciones finales:** número de selecciones finales realizadas por el usuario para terminar la tarea propuesta. Obviamente, este número debe ser mayor o igual que las selecciones óptimas, concordando con la fórmula (5.1).

$$Sele. Finales = Sele. Óptima + Sele. Adicionales + Errores + Warnings \quad (5.1)$$
3. **Selecciones adicionales:** número de selecciones que fueron necesarias para solventar un error cometido por el usuario.
4. **Errores:** número de selecciones que el clasificador y el umbral discriminaron erróneamente.
5. **Warnings:** número de selecciones que no superaron el umbral y, por tanto, el usuario debió repetir posteriormente.
6. **Duración:** como su nombre indica, la duración de la prueba para ese usuario en concreto.
7. **Precisión:** porcentaje de selecciones correctas sobre el total de selecciones realizadas para una tarea concreta. La precisión se calcula atendiendo a la fórmula (5.2).

8. **Precisión total:** porcentaje de selecciones correctas sobre el total de selecciones realizadas para todas las tareas, atendiendo a la fórmula (5.2) y considerando como $Sele.Finales = \sum_{TAREA=1}^4 Sele.Finales$ y $Errores = \sum_{TAREA=1}^4 Errores$.

$$Precisión = \frac{Sele.Finales - Errores}{Sele.Finales} = 1 - \frac{Errores}{Sele.Finales} \quad (5.2)$$

5.2.1. Sujeto U1

El sujeto U1 cometió cuatro errores en el transcurso de las tareas, sin embargo, una precisión del 92.9825% demuestra que el umbral óptimo para ese usuario y su clasificador funcionan de forma excelente. Este resultado se observa en la Tabla 5-6.

La primera tarea consiguió realizarla sin error alguno, no obstante, se produjo un *warning* cuando la intensidad de una selección no superó el umbral. En la segunda tarea se cometieron dos errores y dos *warnings*, resultando en una precisión del 88.8%. Estos errores no requirieron realizar selecciones adicionales en el momento de su aparición, por tanto, el número de selecciones finales se elevó a 18, cuando las selecciones óptimas hubiesen sido 14. En la tercera tarea se volvieron a cometer dos errores y un solo *warning*; esta vez uno de ellos sí requirió una selección adicional, causando una precisión del 88.2353%. Finalmente, la cuarta tarea se realizó de manera impecable, puesto que el objetivo era conseguir provocar 11 *warnings* sin error alguno.

Aunque los *warning* no afectan negativamente a la precisión del sistema, sí afectan a la velocidad del mismo. El gran número de *warnings* obtenidos para el sujeto U1 (en concreto 4 para las tres primeras sesiones) pone en duda la situación del umbral del usuario. Quizás un umbral un poco inferior al actual hubiese reducido esas cifras significativamente.

5.2.2. Sujeto U2

El sujeto U2 es, probablemente, el sujeto con mejores resultados en estas tareas. Este sujeto ha obtenido una precisión del 100% en todas las tareas, sin producir *warning* o error alguno, tal y como puede observarse en la Tabla 5-7.

Estos resultados implican que el clasificador y el umbral para este sujeto no podrían funcionar mejor, convirtiéndose en resultados inmejorables.

TABLA 5-6. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U1.

SUJETO U1	TAREA 1	TAREA 2	TAREA 3	TAREA 4
Sele. Óptimas	10	14	13	
Sele. Finales	11	18	17	11
Sele. Adicionales	0	0	1	
Errores	0	2	2	0
Warnings	1	2	1	11
Duración	5:10min	8:55min	10:04min	5:00min
Precisión	100%	88,8%	88,2353%	100%
Precisión Total	92,9825%			

TABLA 5-7. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U2.

SUJETO U2	TAREA 1	TAREA 2	TAREA 3	TAREA 4
Sele. Óptimas	10	14	13	
Sele. Finales	10	14	13	11
Sele. Adicionales	0	0	0	
Errores	0	0	0	0
Warnings	0	0	0	11
Duración	4:50min	6:28min	7:20min	5:00min
Precisión	100%	100%	100%	100%
Precisión Total	100%			

5.2.3. Sujeto U3

El sujeto U3 obtuvo unos resultados extraordinarios, sin cometer error alguno en las tres últimas tareas. Sin embargo, en la primera tarea sí cometió un error, causando una precisión global del 97.9167%.

Curiosamente, el sujeto U3 no cometió ningún *warning* para las tres primeras tareas. Este resultado pone ligeramente en duda la elección del umbral, pudiendo, quizás, ser mejorable si éste se hubiese encontrado en valores un poco más altos. Los resultados se recogen en la Tabla 5-8.

5.2.4. Sujeto U4

El sujeto U4 obtuvo una precisión del 100%, al igual que el sujeto U2. Estos resultados ponen de manifiesto que la elección del clasificador y del umbral ha sido correcta y el funcionamiento de la aplicación es excelente.

Sin embargo, en la primera tarea se cometió un *warning*, el cual no afecta a la precisión pero sí a la velocidad del sistema, tal y como se comentó anteriormente.

TABLA 5-8. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U3.

SUJETO U3	TAREA 1	TAREA 2	TAREA 3	TAREA 4
Sele. Óptimas	10	14	13	
Sele. Finales	10	14	13	11
Sele. Adicionales	0	0	0	
Errores	1 (*)	0	0	0
Warnings	0	0	0	11
Duración	4:43min	6:32min	7:21min	5:00min
Precisión	90%	100%	100%	100%
Precisión Total	97,9167%			

(*) El error cometido permitió seguir con la tarea sin requerir selecciones adicionales.

TABLA 5-8. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U4.

SUJETO U4	TAREA 1	TAREA 2	TAREA 3	TAREA 4
Sele. Óptimas	10	14	13	
Sele. Finales	11	14	13	11
Sele. Adicionales	0	0	0	
Errores	0	0	0	0
Warnings	1	0	0	11
Duración	5:07min	6:36min	7:23min	5:00min
Precisión	100%	100%	100%	100%
Precisión Total	100%			

5.2.5. Sujeto U5

El sujeto U5 obtuvo la peor precisión de todas: 92.5926%, siendo muy alta pero inferior al resto de sujetos.

La primera tarea fue impecable, sin cometer ningún error o *warning* y produciendo, por tanto, un precisión del 100%. En la segunda tarea se cometió un error, causando una selección adicional necesaria para poder finalizar la tarea correctamente, concluyendo con una precisión del 93.75%. La tercera tarea fue en la que obtuvo peores resultados, cometiendo tres errores que requirieron una selección adicional. Estos errores se cometieron intentando manejar la matriz de teclado, probablemente por impaciencia, dado que el mayor número de celdas de la matriz de teclado causa que las selecciones requieran más tiempo. La precisión para la tercera tarea fue de 82.3529%. Para finalizar, los resultados de la cuarta tarea fueron perfectos también, al igual que el resto de usuarios, concluyendo que el umbral funciona perfectamente cuando el sujeto no atiende a los estímulos.

Los resultados obtenidos para este usuario se muestran en la Tabla 5-9, incluida a continuación.

TABLA 5-9. RESULTADOS OBTENIDOS PARA LAS TAREAS PROPUESTAS DEL SUJETO U5.

SUJETO U5	TAREA 1	TAREA 2	TAREA 3	TAREA 4
Sele. Óptimas	10	14	13	
Sele. Finales	10	16	17	11
Sele. Adicionales	0	1	1	
Errores	0	1	3 (*)	0
Warnings	0	0	0	11
Duración	4:39min	7:30min	8:30min	5:00min
Precisión	100%	93,75%	82,3529%	100%
Precisión Total	92,5926%			

(*) Dos de los errores cometidos permitieron seguir con la tarea sin requerir selecciones adicionales.

5.3. Cuestionario de satisfacción

Según se ha comentado en apartados anteriores, al finalizar las tres sesiones se pidió a los usuarios que rellenasen un cuestionario de satisfacción (recogido en el Anexo C.2).

En la Tabla 5-10 se muestran los resultados de dicho cuestionario, cuyas respuestas vienen etiquetadas de la siguiente manera:

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

La última pregunta realizada (no recogida en la tabla) tenía como intención el recoger las sugerencias de los usuarios para mejorar la aplicación, formulada como: “¿Qué cree usted que debería mejorarse para una futura versión?”. Las respuestas de los tres usuarios que contestaron a la misma vienen recogidas a continuación.

El sujeto U2 recalcó que, en su opinión, hay poco tiempo para pensar los comandos a utilizar en cada momento y que, si el usuario no lo piensa rápidamente, hay que esperar a la finalización del intento para poder llevar a cabo la selección. Es decir, demanda más tiempo de reacción entre los intentos.

TABLA 5-10. RESULTADOS OBTENIDOS PARA EL CUESTIONARIO DE SATISFACCIÓN DE LOS USUARIOS.

Cuestiones planteadas	Sujeto U1	Sujeto U2	Sujeto U3	Sujeto U4	Sujeto U5	Media
La interfaz del sistema es intuitiva y fácil de entender.	5	3	3	5	4	4
Me ha resultado sencillo seleccionar comandos de la matriz de navegación.	4	5	4	5	4	4.4
El efecto de parpadeo de los botones de la aplicación resulta molesto.	3	3	3	3	4	3.2
Me ha resultado complicado seleccionar comandos de la matriz de teclado.	3	1	4	1	2	2.2
El tiempo necesario para seleccionar un enlace es muy grande.	2	3	5	3	4	3.4
Habitualmente se seleccionan comandos que no deseo que se seleccionen, complicando el uso de la aplicación.	2	1	1	1	2	1.4
La adquisición de la actividad cerebral mediante EEG con el gorro y electrodos activos es poco práctica.	1	3	4	2	1	2.2
La duración de las sesiones realizadas es adecuada.	4	5	3	5	2	3.8
¿Volvería a participar en un proyecto de características similares?	Sí	Sí	Sí	Sí	Sí	Sí

El sujeto U3 incidió en la necesidad de poder intercalar entre letras mayúsculas y minúsculas en una futura versión, comando útil y en ocasiones necesario para los sitios web que requieren contraseñas. Además, destacó que la sesión utilizada para entrenar el clasificador se hizo un poco pesada y que el gel conductor es molesto.

Finalmente, el sujeto U5 subrayó que existen ciertas celdas de las matrices, correspondientes a la parte inferior o superior de la pantalla, que la visión no detecta con la misma intensidad debido a variaciones de contraste o a la posición de la pantalla panorámica y que, probablemente, podrían influir en la selecciones del usuario. También sugirió la inclusión del comando "TAG" en ambas matrices, no solamente en la matriz de navegación y demandó una mejora en el etiquetado de nodos, recalcando que existen nodos muy cercanos y que sus codificaciones pueden llegar a superponerse en ciertas ocasiones.

6.1. Determinación del umbral

La determinación del umbral, para distinguir entre ausencia o no de atención a la hora de seleccionar comandos de las matrices de navegación y de teclado, es una tarea de suma importancia debido a que el rendimiento de la aplicación depende directamente de éste. En las dos primeras sesiones de evaluación de la aplicación se trató de determinar el umbral óptimo para cada usuario. La discusión de los resultados obtenidos se muestra a continuación.

Tal y como se expresó anteriormente en el capítulo 5, el área bajo la curva ROC (AUC, *Area Under Curve*) es una medida global e independiente del umbral óptimo seleccionado en cada sesión realizada. Por esta razón, en el ámbito sanitario las curvas ROC también se denominan curvas de rendimiento diagnóstico. Estableciendo el máximo valor (AUC=1) como una representación del valor diagnóstico perfecto, es decir, una capacidad discriminatoria inmejorable, y el mínimo valor (AUC=0.5) como una prueba sin capacidad discriminatoria alguna, se pueden comparar los valores obtenidos en ambas sesiones de determinación del umbral (sesión 1 y sesión 2) para esclarecer el significado de los resultados obtenidos.

Según advierte la Figura 6-1, el AUC obtenida para los sujetos U1, U2 y U4 en la primera sesión fue máxima, representando una capacidad discriminatoria perfecta. El sujeto U5 rozó la perfección obteniendo un valor de 0.9948 y el valor más pequeño lo obtuvo el sujeto U3, con una AUC de 0.9549. Con respecto a la segunda sesión, los sujetos U2, U3 y U4 obtuvieron una AUC máxima, seguidos por el sujeto U1 con una AUC de 0.9844 y el sujeto U5 con una AUC de 0.9826.

A la vista de los resultados, se concluye que el rendimiento del umbral obtenido para los sujetos U2 y U4 es impecable, destacando por encima del resto. Seguidamente se observa que los sujetos U1 y U5 obtuvieron valores muy altos, casi perfectos y, finalmente, se observa que el sujeto U3 se quedó por detrás, obteniendo valores AUC menores que el de sus compañeros. Aun así, el valor discriminatorio de todos los umbrales obtenidos es extraordinario.

En la Figura 6-2 se aprecian los valores de las precisiones obtenidas para ambas sesiones, con un comportamiento similar a la AUC. Se obtienen precisiones perfectas para los sujetos U2 y U4, seguidas por el sujeto U3 con 97.9167% para la primera sesión y 100% para segunda, por el sujeto U1 con 100% para la primera sesión y 95.83% para la segunda y, finalmente, el sujeto U5, con los peores resultados obteniendo 97.9167% para la primera sesión y 93.75% para la segunda.

Sin embargo, el mínimo de precisión se encuentra por encima del 90%. Este resultado implica que la elección del clasificador se puede considerar como excelente para todos los usuarios y que más del 90% de las selecciones a realizar se discriminan correctamente, causando un buen rendimiento de la aplicación en general. No obstante, el rendimiento del umbral fijado se pondrá a prueba en las siguientes tareas.

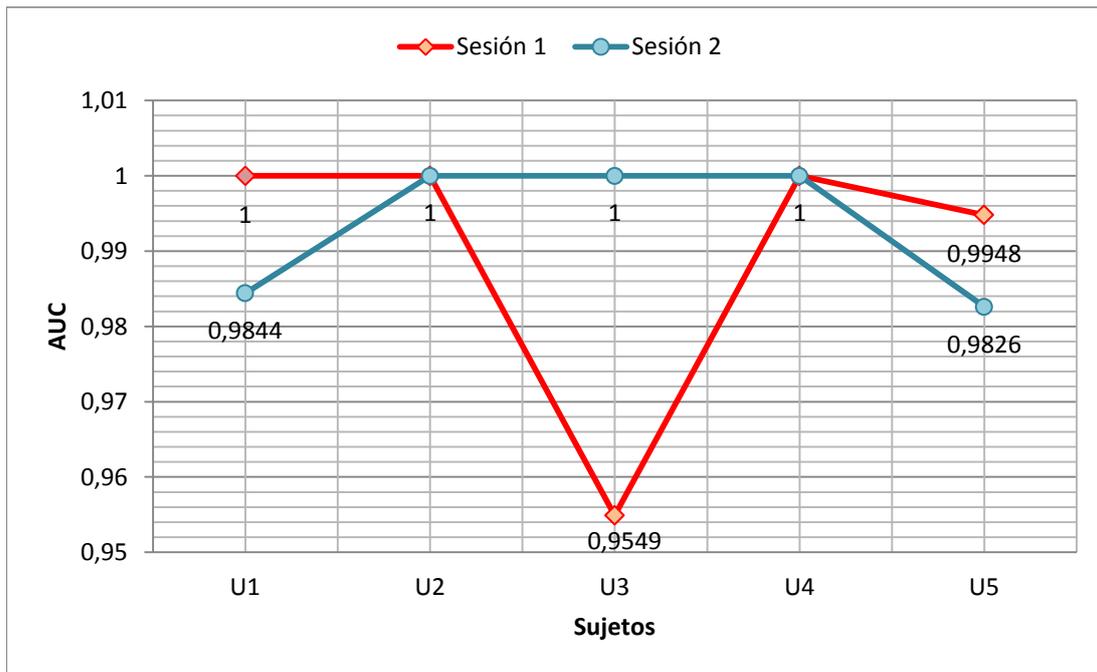


Figura 6-1. Comparación del valor de AUC para las dos primeras sesiones destinadas a la determinación del umbral óptimo de cada sujeto.

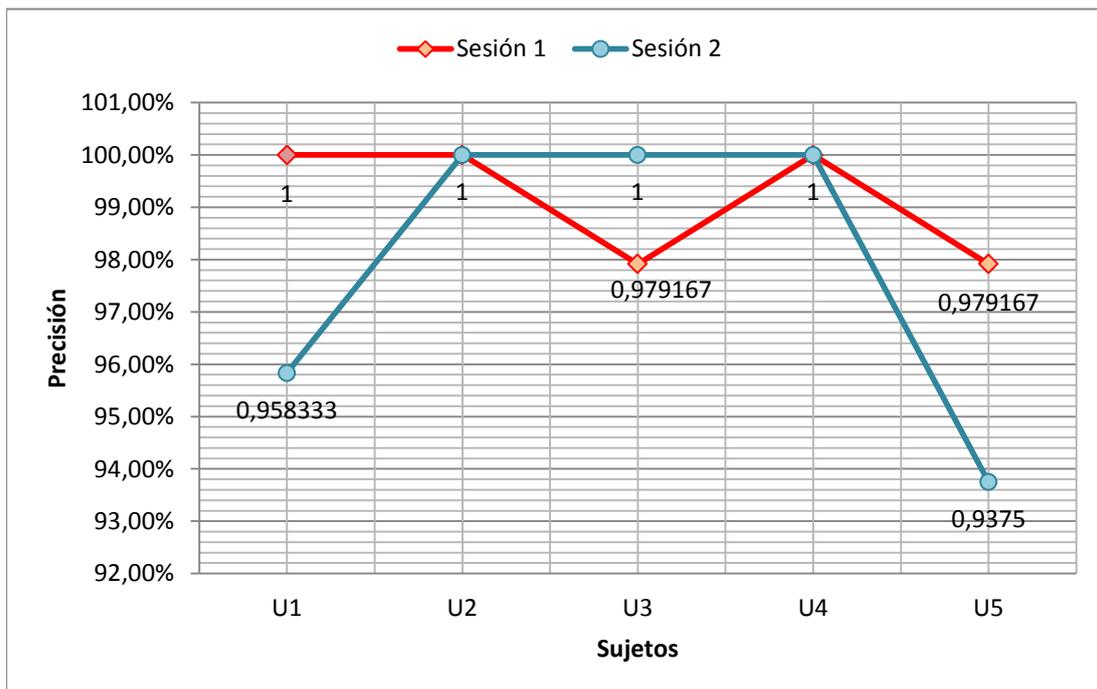


Figura 6-2. Comparación de la precisión obtenida para las dos primeras sesiones destinadas a la determinación del umbral óptimo de cada sujeto.

La elección del umbral óptimo de cada curva ROC se realizó atendiendo al criterio de maximización del par sensibilidad-especificidad. Sin embargo, podría mejorarse la aplicación si el umbral se seleccionase intentado maximizar la sensibilidad por encima de la especificidad, puesto que se podría considerar más importante que la aplicación no seleccione comandos cuando el usuario está ignorando los estímulos y, por tanto, cueste más seleccionarlos al atenderlos, que clasificar correctamente los estímulos atendidos y no asegurar que no se seleccione ninguno al ignorarlos.

6.2. Tareas propuestas

Las tareas con dificultad progresiva propuestas para evaluar la aplicación por parte de los usuarios han puesto de manifiesto que la aplicación funciona adecuadamente, con un bajo porcentaje de error.

En la Figura 6-3 se muestran una serie de gráficos circulares que recogen la cantidad de selecciones que realizó cada usuario en las tres primeras tareas. Tal y como puede observarse, solamente los sujetos U1, U3 y U5 cometieron algún error. El sujeto U2 realizó las tareas de manera perfecta, sin provocar ningún *warning* o error y sin requerir selecciones adicionales. Le sigue el sujeto U4, el cual solamente produjo un *warning* al no superar el umbral en una determinada selección y el sujeto U3, el cual solamente cometió un error en todas las tareas. Está claro que los peores resultados los obtuvieron el sujeto U1 y el sujeto U5, con cuatro errores cada uno.

Curiosamente, los sujetos U3 y U5 no obtuvieron ningún *warning* y en cambio, sí obtuvieron uno o varios errores. Estos resultados ponen ligeramente en duda la elección de los umbrales, pudiendo, quizás, ser mejorable si éstos se hubiesen encontrado en valores un poco más altos.

La cuarta tarea no se ha representado puesto que absolutamente todos los sujetos la realizaron de manera impecable, obteniendo un 100% de precisión al generar cuatro *warnings* mientras ignoraban los estímulos. Se concluye, por tanto, que la elección del umbral generalmente es correcta, puesto que ningún estímulo fue capaz de superarlo mientras el usuario visualizaba una web concreta.

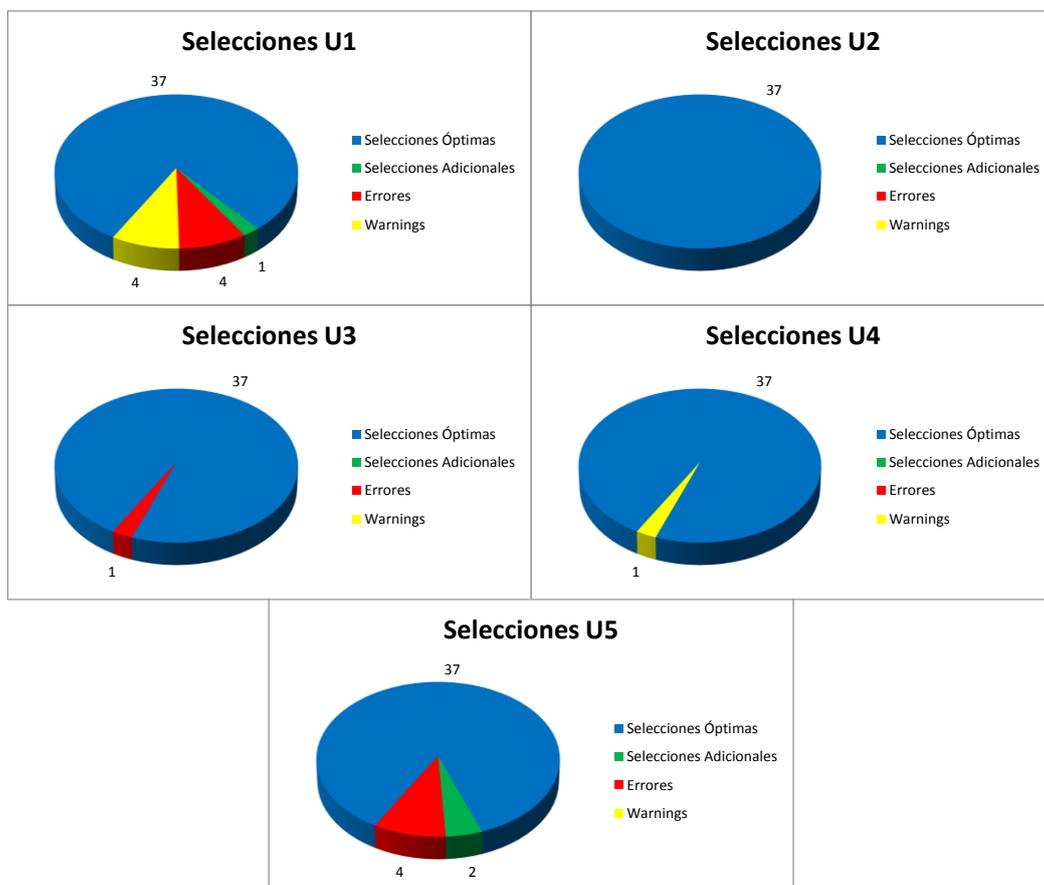


Figura 6-3. Tipo de selecciones realizadas para cada sujeto durante las tres primeras tareas propuestas.

En términos de precisión el sujeto U2 y el sujeto U4 han obtenido resultados perfectos para todas las tareas, tal y como puede observarse en la Figura 6-3. Estos resultados ponen de manifiesto que la elección del clasificador y del umbral para estos sujetos ha sido óptima y la aplicación se ha adaptado completamente a las características individuales de éstos sujetos.

El sujeto U3 también ha obtenido precisiones muy elevadas, siendo éstas máximas para la primera y la tercera tarea, y obteniendo el 90% de precisión para la segunda tarea. Le sigue el sujeto U1, cuya precisión ha sido máxima en la primera tarea, más sencilla, pero ha decaído en las dos tareas siguientes, obteniendo un 88.8% para la segunda y un 88.2353% para la tercera. Finalmente, el sujeto U5 ha vuelto a obtener malos resultados, obteniendo una precisión máxima para la primera tarea pero unas precisiones del 93.75% y del 82.3529% para las tareas segunda y tercera, respectivamente.

Según advierten los datos, la primera tarea la realizaron perfectamente casi todos los sujetos, posiblemente gracias a la sencillez de la misma (exceptuando al sujeto U3). La segunda y la tercera tarea, más complicadas, dieron problemas a los sujetos U1 y U5, los cuales obtuvieron precisiones bastante inferiores. Sin embargo, la precisión más baja es de 82.3529%, la cual implica que más del 80% de las selecciones realizadas fueron correctas.

Para finalizar, la Figura 6-5 muestra una comparación entre las precisiones obtenidas para todas las sesiones, incluyendo la determinación del umbral y las tareas propuestas. Tal y como se puede observar, los sujetos U2 y U4 quedaron completamente por encima del resto de sujetos, obteniendo los mejores resultados en todas las sesiones realizadas. No obstante, las precisiones obtenidas por el resto no son bajas y, por tanto, se concluye que el funcionamiento de la aplicación, en términos técnicos, es satisfactorio.

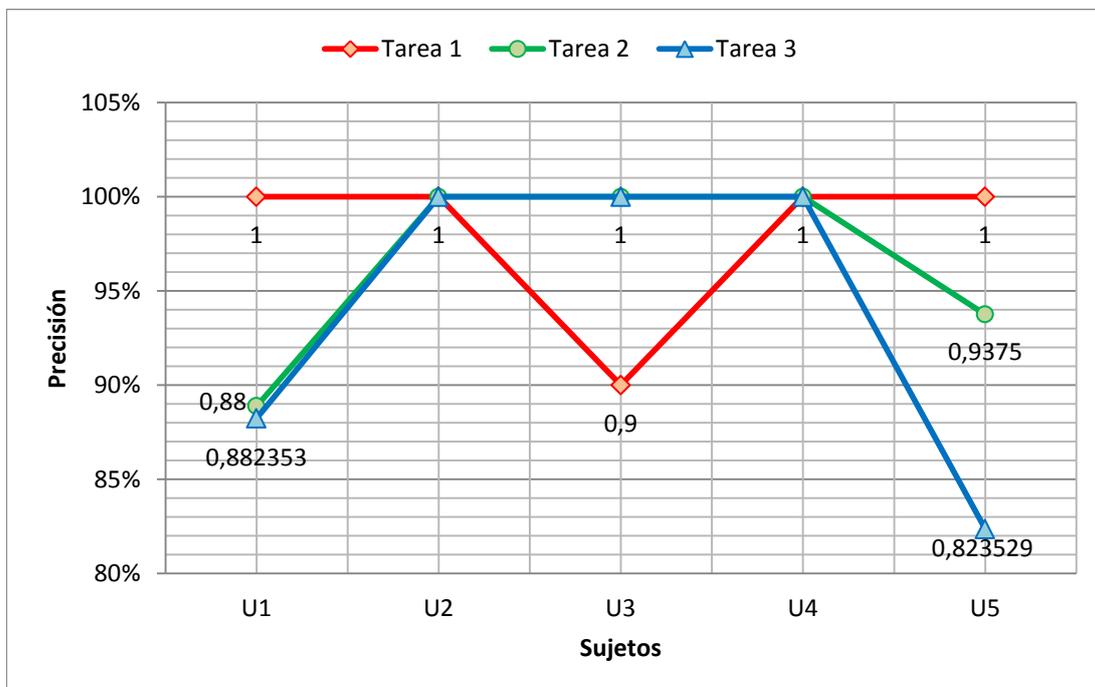


Figura 6-4. Comparación de la precisión obtenida para las tres primeras tareas destinadas a evaluar el funcionamiento general de la aplicación.

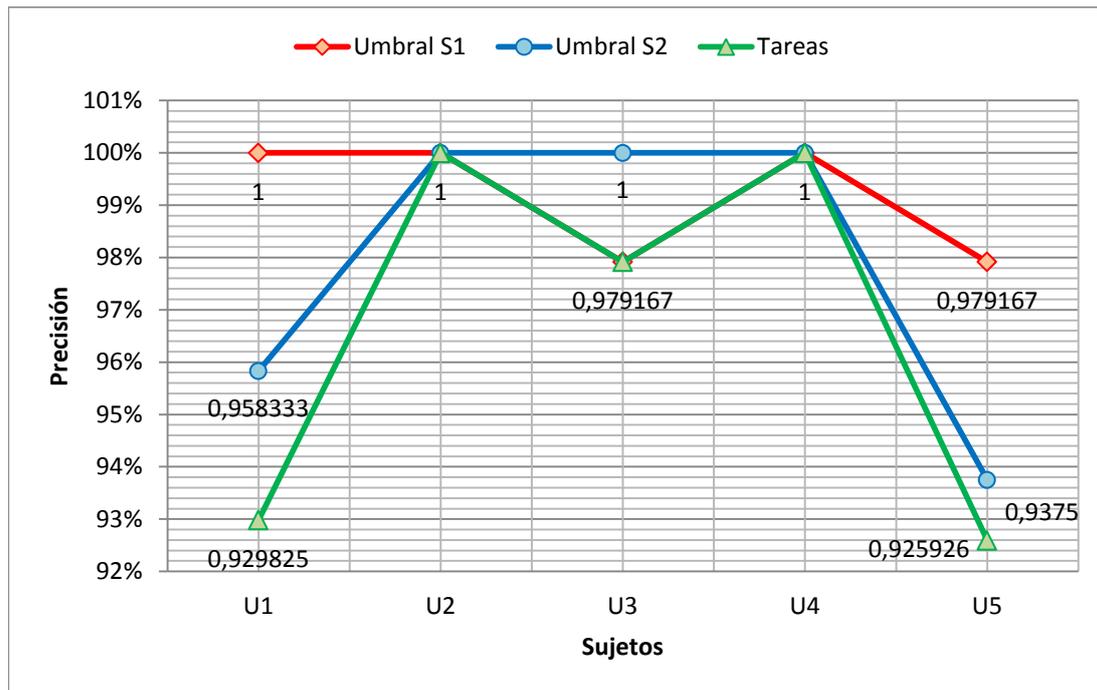


Figura 6-5. Comparación de la precisión obtenida para todas las sesiones, tanto la determinación del umbral (sesión 1: UmbralS1 y sesión 2: UmbralS2) como las tareas propuestas.

6.3. Cuestionario de satisfacción

El cuestionario de satisfacción de los sujetos refleja tanto aspectos en los que todos los usuarios se ponen de acuerdo como el fenómeno contrario.

Los cinco sujetos están de acuerdo en que la interfaz del sistema es intuitiva y fácil de aprender, en que les ha resultado sencillo seleccionar comandos de la matriz de navegación, en que el efecto de parpadeo de las celdas de las matrices no es del todo molesto pero que acaba cansando y en que normalmente no se seleccionan comandos que no desean seleccionar.

Sin embargo, difieren en opiniones con respecto a la dificultad de seleccionar comandos de la matriz de teclado. Los sujetos U2, U4 y U5 declaran que no les ha resultado difícil seleccionarlos, mientras que los sujetos U1 y U3 indican que sí existe cierta dificultad. Estos resultados concuerdan bastante con las precisiones obtenidas, donde los sujetos U2 y U4 siempre se han encontrado por encima del resto.

Tampoco se ponen de acuerdo a la hora de determinar si el tiempo necesario para seleccionar un nodo es grande o no. Los sujetos U1, U2 y U4 reflejan en el cuestionario que el tiempo necesario no les ha resultado muy grande, sin embargo, los sujetos U3 y U5 opinan que éste tiempo es bastante grande, quizás porque los *warnings* o errores obtenidos ralentizaron las selecciones.

Con respecto a la adquisición de la señal, todos los sujetos exceptuando el sujeto U3 determinaron que la adquisición de la actividad cerebral mediante EEG con el gorro y electrodos activos es muy práctica.

Posteriormente, los sujetos U2 y U4 reflejaron que la duración de las sesiones realizadas es completamente adecuada, mientras que el resto de los usuarios reflejaron ciertas reticencias. Para finalizar, todos los sujetos estuvieron de acuerdo en que volverían a participar en un proyecto de características similares.

Adicionalmente, tres de los sujetos propusieron sugerencias para mejorar la aplicación. Las sugerencias más apropiadas son el aumento del tiempo entre intentos (propuesta por el sujeto U2), el cual podría permitir pensar mejor los comandos a seleccionar en cada momento y tener más tiempo de reacción; la inclusión de un conmutador entre mayúsculas y minúsculas, útil en sitios web que requieren contraseñas (propuesta por el sujeto U3); y finalmente, la optimización de la posición de las etiquetas de los nodos, evitando superposición entre ellas, y de la altura de la pantalla para poder recibir con la misma intensidad las distintas celdas de la matriz (propuestas por el sujeto U5).

Quizás las dos primeras sugerencias son las más importantes de cara a la mejora de las funcionalidades de la aplicación en sí. Establecer mayor tiempo entre cada intento permitiría aumentar el tiempo de reacción de cada usuario, sin embargo, reduciría la velocidad de la aplicación. Es, por tanto, un compromiso que podría adaptar el usuario en los parámetros del “Operador”. La segunda sugerencia radicaba en incluir un conmutador para intercalar entre mayúsculas y minúsculas. Esta sugerencia es de vital importancia para los sitios en los que la distinción entre este tipo de letras es fundamental, por tanto, podría implementarse en la matriz de teclado.

El sujeto U3 demandó una sesión más corta para entrenar el clasificador y una adquisición de la señal menos molesta, sin la utilización del gel conductor. Lamentablemente, estos aspectos están fuera del ámbito de la aplicación puesto que la duración de la sesión de entrenamiento determina completamente el rendimiento posterior y la adquisición de la señal con gel conductor actualmente es necesaria para reducir la impedancia del cráneo. Finalmente, el sujeto U5 sugirió la inclusión del conmutador “TAG” en la matriz de teclado. Sin embargo, la inclusión de éste botón en esa matriz haría inútil la distinción entre matriz de navegación y de teclado, y aumentaría el tiempo necesario para seleccionar un comando en ésta última.

6.4. Comparación con otros estudios

Atendiendo a las funcionalidades de la aplicación, la mayor similitud se encuentra con respecto al estudio de Mugler *et al* [22] puesto que, aunque muchos otros estudios también hayan utilizado los potenciales evocados P300 como señales de control (Sirvent Blasco *et al* [23] y Yu *et al* [24]), es el único que también selecciona los hipervínculos de una web etiquetándolos con una codificación única.

El resto de navegadores utilizan una metodología distinta para seleccionar los hipervínculos de la web. En Bensch *et al* [19] se rodean de un borde rectangular de dos posibles colores, y se seleccionan a través de una selección binaria mediante los potenciales SCP. En Sirvent Blasco *et al* [23] y en Yu *et al* [24] se seleccionan mediante el desplazamiento de un cursor controlado con los potenciales evocados P300 y, adicionalmente, con N2PC y ritmos sensoriomotores, respectivamente.

Sin embargo, también posee diferencias con respecto al estudio Mugler *et al* [22]. Entre ellas destaca el hecho de que en esta aplicación se establece un umbral para distinguir si el usuario desea seleccionar un comando determinado o, por el contrario, desea leer una página web tranquilamente, mientras que en Mugler *et al* [22] se establece un tiempo de lectura que no permite seleccionar nada al usuario, ralentizando la aplicación y convirtiéndose así en un navegador más rígido que el implementado en el presente trabajo de fin de grado. De hecho, ninguno de los estudios comparados anteriormente implementan un umbral de decisión.

Adicionalmente, esta aplicación no requiere supervisión, a diferencia de Bensch *et al* [19]. No obstante, destaca el hecho de que todos los estudios implementan la

funcionalidad de edición de la barra de URL, permitiendo introducir caracteres directamente en ella, funcionalidad no implementada en la aplicación desarrollada.

En términos de precisión, la aplicación desarrollada consigue precisiones bastante mayores que el resto de navegadores, concretamente, se obtiene una precisión media en las tareas propuestas del 96.698% para los cinco sujetos sanos, por encima del 90% de Mugler *et al* [22], el 93.21% de Yu *et al* [24] y el 93% de Sirvent Blasco *et al* [23] y de Liu *et al* [25] (todas ellas para sujetos sanos).

6.5. Limitaciones de la aplicación

Dentro de las limitaciones de la aplicación destaca el hecho de que no incluye los marcadores con las webs más visitadas por el usuario o un historial para consultar qué web ha visitado anteriormente, funcionalidades prescindibles pero que dotarían al navegador de mayor complejidad.

Adicionalmente, según se ha advertido en el apartado anterior, la introducción de una barra editable de URL facilitaría mucho la fluidez de la aplicación. También cabe señalar que la aplicación no diferencia entre mayúsculas y minúsculas, sin embargo, muchas páginas web (normalmente las que requieren contraseñas) lo hacen, convirtiéndose en una limitación que debe solventarse en una futura versión.

Aunque la aplicación obtiene precisiones altas, una mejora en el procesado de la señal podría mejorarlas, tanto en la extracción como en la traducción de características. No obstante, estas precisiones se han obtenido para un grupo relativamente pequeño de usuarios sanos, por tanto, sería conveniente evaluar la aplicación por parte de un mayor número de usuarios y por parte de usuarios con grave discapacidad.

CONCLUSIONES Y LÍNEAS FUTURAS

7.1. Conclusiones

A lo largo del presente 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 del EEG como la mejor solución para desarrollar la aplicación web 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 el algoritmo de traducción, también conocido como traducción de características. Adicionalmente, se ha realizado una búsqueda exhaustiva del estado del arte sobre los navegadores web diseñados anteriormente y se han comparado entre ellos para extraer las ventajas y los inconvenientes de cada uno de ellos. En base a esa comparativa se han propuesto los objetivos de la aplicación y se han elegido los potenciales evocados P300 como la señal de control más apropiada.

Posteriormente se han detallado en profundidad las características de los potenciales evocados P300 y la variación de su forma según los aspectos psicológicos y farmacológicos de los sujetos o las diferencias individuales entre ellos. Una vez estudiada la señal de control a utilizar, se han detallado los diversos métodos de extracción de características para los potenciales evocados P300 utilizados con anterioridad, considerando un análisis espacial, temporal y espectral. Dentro del análisis espacial se han descrito el filtrado Laplaciano, los métodos de referencia de media común, 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, la detección de picos y el cálculo del área. En último lugar, dentro del análisis espectral se han introducido los conceptos de transformada de Fourier de tiempo corto y las transformadas *wavelet*. 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, el análisis discriminante lineal paso-a-paso y las máquinas de vectores de soporte. También se han introducido los conceptos de sensibilidad, especificidad y las curvas ROC con el fin de crear un umbral óptimo que discrimine entre dos poblaciones.

En cuanto al diseño de la aplicación de navegación web basado en la señal EEG y los potenciales evocados P300, el primer paso ha consistido en recoger los objetivos de la aplicación. Una vez establecidos se ha estructurado la misma en dos partes principales: *Navigator* y *P3Speller*. Ésta última parte ha sido desarrollada con la ayuda del sistema de propósito general BCI2000, cuya finalidad es la de facilitar el desarrollo, la investigación y la monitorización de aplicaciones en tiempo real que requieren adquirir y procesar una señal EEG.

La etapa de *Navigator* se ha implementado como una extensión del navegador web a controlar, *Google Chrome*, en lenguaje JavaScript. La finalidad de esta etapa consiste en procesar los comandos de teclado recibidos de la otra parte de la

aplicación y traducirlos en comandos del navegador web. Además, se encarga de implementar la interfaz gráfica del navegador, compuesta por una serie de hipervínculos codificados con ciertos caracteres alfanuméricos y un *buffer* que recoge las pulsaciones recibidas.

La etapa del *P3Speller* 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 CAR seguido de un promediado sincronizado como métodos de extracción de características, y un algoritmo SWLDA como método de traducción. También se encarga de producir los potenciales evocados P300 necesarios para controlar la aplicación mediante el paradigma *odd-ball*, consistente en presentar una serie de estímulos objetivo infrecuentes camuflados entre estímulos frecuentes. Cuando el usuario detecta los estímulos objetivos genera un potencial evocado P300 que detectará la aplicación y determinará el comando que deseaba seleccionar el usuario. Este paradigma *odd-ball* lo controlan dos matrices, bautizadas como matriz de teclado y de navegación, que implementan comandos distintos. Una vez determinado el comando deseado se ha introducido un umbral que clasificará la atención del usuario: si éste ignora los estímulos no se enviará el comando y, en caso contrario, sí se enviarán hacia el *Navigator*.

Una vez desarrollada la aplicación, ésta ha sido evaluada por parte de cinco usuarios sanos durante tres sesiones distintas. La disposición de los electrodos ha seguido la norma internacional 10/20, ubicando los mismos en las localizaciones Fz, Cz, Pz, P3, P4, PO7, PO8 y Oz. Al principio de la primera sesión se ha entrenado el clasificador SWLDA de cada usuario, tratando de estimar el vector de pesos adecuado. En la última parte de la primera sesión y en la segunda se ha determinado el umbral óptimo para cada usuario, monitorizando la intensidad de los potenciales P300 para cuando el usuario atendía o ignoraba los estímulos. Finalmente, en la tercera sesión se han realizado cuatro pruebas con el fin de evaluar el rendimiento general de la aplicación. Al terminar ésta sesión, se ha entregado a los usuarios un cuestionario de satisfacción, donde han reflejado sus experiencias y las sugerencias de mejora.

Para terminar, se han expuesto los resultados obtenidos y se han discutido. Estos resultados reflejan el buen funcionamiento del sistema, cuyas precisiones no bajan del 92.6%, lo cual significa que más del 90% de las selecciones realizadas para todos los usuarios y todas las sesiones han sido correctas. De hecho, dos sujetos han obtenido precisiones del 100% para todas las sesiones realizadas. El cuestionario de satisfacción muestra que los usuarios opinan que la interfaz del sistema es sencilla y fácil de aprender, que la selección de comandos con la matriz de navegación es sencilla y que habitualmente no se seleccionan comandos no deseados. No obstante, difieren en la dificultad de selección en la matriz de teclado, más grande y complicada, y en general a los usuarios que han obtenido mejores resultados les resulta más sencillo que al resto.

Se concluye que el sistema funciona de manera excepcional una vez se han adecuado correctamente el clasificador y el umbral de cada usuario individual, obteniendo precisiones superiores al 90%. Sin embargo, la adecuación de estos parámetros implica varias sesiones de entrenamiento, hecho que a algunos usuarios les resulta incómodo o demasiado tedioso. No obstante, aunque las precisiones obtenidas sean altas, la aplicación aún debe madurar e implementar muchas más funcionalidades, además de mejorar las ya existentes.

Adicionalmente, el trabajo de fin de grado ha cumplido todos los objetivos específicos propuestos anteriormente:

1. Se han estudiado las distintas técnicas para registrar la actividad cerebral, los sistemas BCI y las señales de control utilizadas normalmente. Como resultado, se han elegido el EEG y los potenciales evocados P300 como método de registro y señal de control utilizada, respectivamente.
2. Se ha realizado una revisión del estado del arte de los navegadores web BCI desarrollados con anterioridad y se han comparado atendiendo a sus funcionalidades y sus precisiones.
3. Se han establecido las funcionalidades que debía implementar la aplicación en base a esa comparativa.
4. Se han diseñado las matrices y se ha desarrollado la parte de la aplicación que adquiere la señal, la procesa y envía los comandos necesarios, identificada como *BCI Browser: P3Speller*.
5. Se ha desarrollado la última parte de la aplicación, encargada de interpretar los comandos, identificada como *BCI Browser: Navigator*.
6. Se ha diseñado la metodología para implementar un umbral que dependa de cada usuario para discriminar entre ausencia o no de atención. Finalmente, se ha optado por realizar el promedio de los umbrales óptimos obtenidos a través de las curvas ROC en sesiones distintas.
7. Se ha evaluado el clasificador, el umbral óptimo y las funcionalidades de la aplicación por parte de cinco sujetos sanos en tres sesiones distintas.
8. Se han discutido los resultados y se han comparado con otros estudios, declarando el buen funcionamiento de la aplicación, cuya precisión media es del 96.698%.
9. Se han extraído conclusiones a partir de los resultados obtenidos, como advierte este punto en concreto.

7.2. Líneas Futuras

Las líneas futuras de la aplicación web desarrollada tratan de solventar sus limitaciones añadiendo nuevas funcionalidades o modificando las ya existentes. Como introducción de nuevas funcionalidades se podría hablar de añadir una serie de marcadores que pueda editar el usuario. De esta manera, el usuario podrá encontrar más fácilmente sus páginas web favoritas, sin necesidad de invertir tiempo buscándolas en *Google*. Otra funcionalidad podría ser la introducción de un comando que permita modificar la barra de URL, para aumentar la velocidad a la hora de desplazarse a una web concreta, y también se podría implementar una funcionalidad que mostrase el historial reciente y permitiese visitar las webs incluidas en el mismo. Además, se podría optimizar la localización de las etiquetas que codifican los nodos de la web para evitar, a toda costa, una superposición entre ellas.

Atendiendo al procesado de la señal EEG, se podrían implementar técnicas de procesado diferentes, tanto para la extracción de características como para la traducción de las mismas, y comparar los resultados para determinar qué técnica son las más adecuadas.

Adicionalmente, un usuario determinó que sería útil poder alternar entre mayúsculas y minúsculas, sobre todo a la hora de navegar por webs que requieren la introducción de contraseñas con ciertos requisitos de seguridad; y otro usuario insistió en introducir un tiempo mayor entre cada intento en las matrices P300 para pensar mejor el comando que se debe seleccionar, sugerencias a tener en cuenta.

Para finalizar, una línea futura de la aplicación podría ser la evaluación de la misma por parte de sujetos con grave discapacidad, puesto que esa es la finalidad de la misma y, probablemente, generaría varias propuestas de mejora.

REFERENCIAS

- [1] J. López, E. Martos, “Caracterización de Señales Aleatorias”, in *Señales Aleatorias*, 1st ed. Malaga: Marcombo S.A, pp. 116–119, 2011.
- [2] L. Sörnmo, P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*, 1st ed. Sweden: Academic Press, pp. 1-24, June 2005.
- [3] S. M. Landau, “Comparing rates of change in cognitive measurements and regional glucose metabolism in Alzheimer’s disease and mild cognitive impairment: Data from the Alzheimer’s disease neuroimaging initiative”, *Neurology*, vol. 70, no. 11, Philadelphia: Lippincott Williams & Wilkins, 2008.
- [4] H. H. Jasper, “The ten twenty electrode system of the International Federation”, *Electroenceph. Clin. Neurophysiol.*, vol. 10, pp. 371–375, 1958.
- [5] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, “Brain–computer interfaces for communication and control”, *Clin. Neurophysiol.*, vol. 113, pp. 767–1 79, 2002.
- [6] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, T. M. Vaughan, “Brain-computer Interface Technology: A review of the First International Meeting”, *IEEE Trans. Rehab. Eng.*, vol. 8, pp. 164–173, June 2000.
- [7] J. J. Vidal, “Real-Time Detection of Brain Events in EEG”, *Proc. IEEE*, vol. 65, pp. 633–664, May 1977.
- [8] J. R. Wolpaw, G. E. Loeb, B. Z. Allison, E. Donchin, O. Feix do Nascimento, W. J. Heetderks, F. Nijboer, W. G. Shain, J. N. Turner, “BCI Meeting 2005–Workshop on Signals and Recording Methods”, *IEEE Trans. Rehab. Eng.*, vol. 14, no. 2, pp.138–141, June 2006.
- [9] O. Friman, I. Volosyak, A. Gräser, “Multiple Channel Detection of Steady-State Visual Evoked Potentials for Brain-Computer Interfaces”, *IEEE Trans. Biomed. Eng.*, vol. 54, no. 4, pp. 742–750, April 2007.
- [10] M. Middendorf, G. McMillan, G. Calhoun, K. S. Jones, “Brain–computer interfaces based on steady-state visual evoked response”, *IEEE Trans. Rehab. Eng.*, vol. 8, no.2, pp. 211–214, June 2000.
- [11] J. Guan, Y. Ghen, J. Lin, Y. Yuan, M. Huang, “N2 Components as Features for Brain Computer Interface”, *Proc. 2005 First International Conference on Neural Interface and Control*, pp. 45–49, May 2005.
- [12] N. Birbaumer, “Slow cortical potentials: their origin, meaning, and clinical use”, in *Brain and behavior past, present, and future*, 1st ed. The Netherlands: Tilburg University Press, 1997. pp. 25–39.
- [13] T. Hinterberger, N. Weiskopf, R. Veit, B. Wilhelm, E. Betta, N. Birbaumer, “An EEG-Driven Brain-Computer Interface Combined With Functional Magnetic Resonance Imaging (fMRI)”, *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 971–974, June 2004.
- [14] G. Pfurtscheller, C. Neuper, “Motor Imagery and Direct Brain-Communication”, *Proc. of the IEEE*, vol. 89, no. 7, pp. 1123–1134, July 2001.
- [15] J. Hu, J. Si, B. P. Olson, J. He, “Principle Component Feature Detector for Motor Cortical Control”, *Proc. 26th Annual International Conference of the IEEE EMBS*, pp. 4021–4024. September 2004.

- [16] D. J. McFarland, C. W. Anderson, K.R. Müller, A. Schlögl, D. J. Krusienski, "BCI Meeting 2005–Workshop on BCI Signal Processing: Feature Extraction and Translation", *IEEE Trans. Rehab. Eng.*, vol. 14, no. 2, pp. 135–138, June 2006.
- [17] K. R. Müller, C. W. Anderson, G. E. Birch, "Linear and Nonlinear Methods for Brain-Computer Interfaces", *IEEE Trans. Rehab. Eng.*, vol. 11, no. 2, pp. 165–169, June 2003.
- [18] G. E. Fabiani, D. J. McFarland, J. R. Wolpaw, G. Pfurtscheller, "Conversion of EEG Activity Into Cursor Movement by a Brain-Computer Interface (BCI)", *IEEE Trans. Rehab. Eng.*, vol. 12, no. 3, pp. 331–338, September 2004.
- [19] M. Bensch, A. A. Karim, J. Mellinger, T. Hinterberger, M. Tangermann, M. Bogdan, "Nessi: An EEG-Controlled Web Browser for Severely Paralyzed Patients", *Computational intelligence and neuroscience*, vol. 2007, June 2007.
- [20] J. A. Pineda, D. S. Silverman, A. Vankov, J. Hestenes, "Learning to Control Brain Rhythms: Making a Brain-Computer Interface Possible," *IEEE Trans. Rehab. Eng.*, vol. 11, no. 2, pp. 181-184, June 2003.
- [21] L. J. Trejo, R. Rosipal, B. Matthews, "Brain-Computer Interfaces for 1-D and 2-D Cursor Control: Designs Using Volitional Control of the EEG Spectrum or Steady-State Visual Evoked Potentials", *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 14, no 2, pp. 225–229, June 2006.
- [22] Mugler E. M., C. A. Ruf, S. Halder, M. Bensch, A. Kübler, "Design and Implementation of a P300-Based Brain-Computer Interface for Controlling an Internet Browser", *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no 6, pp. 599-609, December 2010.
- [23] J. L. Sirvent Blasco, E. Lláñez, A. Úbeda, J. M. Azorín, "Visual Evoked Potential-Based Brain-Machine Interface Applications to Assist Disabled People", *Expert Systems with Applications*, vol. 39, no 9, pp. 7908-7918, 2012.
- [24] T. Yu, Y. Li, J. Long, Z. Gu, "Surfing the Internet with a BCI Mouse", *Journal of neural engineering*, vol. 9, no 3, pp. 036012, May 2012.
- [25] T. Liu, L. Goldberg, S. Gao, B. Hong, "An Online Brain-Computer Interface Using Non-Flashing Visual Evoked Potentials", *Journal of neural engineering*, vol. 7, no 3, pp. 036003, April 2010.
- [26] J. Yin, D. Jiang, J. Hu, "Design and Application of Brain-Computer- Interface Web Browser Based on VEP", in *Biomedical Information Engineering, FBIE 2009. International Conference on Future. IEEE*, pp. 77-80, 2009.
- [27] ITU-R Recommendation BT.709: "Parameter Values for the HDTV Standards for Production and International Programme Exchange" (4.2 Signal format: derivation of luminance signal).
- [28] A. Kübler, A. Furdea, S. Halder, E. M. Hammer, F. Nijboer, B. Kotchoubey, "A Brain-Computer Interface Controlled Auditory Event-Related Potential (P300) Spelling System for Locked-In Patients", *Annals of the New York Academy of Sciences*, vol. 1157, no 1, pp. 90-100, 2009.
- [29] T. W. Picton, "The P300 wave of the human event-related potential", *Journal of clinical neurophysiology*, vol. 9, no 4, pp. 456–479, 1992.
- [30] R. M. Rangayyan, "Filtering for Removal Artifacts", in *Biomedical Signal Analysis: A Case-Study Approach*, Canada: IEEE Press & Wiley, pp. 73–176, 2002.
- [31] N. K. Squires, K. C. Squires, S. A. Hillyard, "Two Varieties of Long-Latency Positive Waves Evoked by Unpredictable Auditory Stimuli in Man", *Electroencephalogr. Clin. Neurophysiol.*, vol. 8, pp. 387–401, 1975.
- [32] R. M. Stelmack, M. Houlihan, P. A. McGarry-Roberts, "Personality, reaction time, and event-related potentials". *Journal of Personality and Social Psychology*, vol. 65, no 2, pp. 399, 1993.
- [33] M. Klein, M. G. H. Coles, E. Donchin, "People with absolute pitch process tones without producing a P300". *Science*, vol. 223, no 4642, pp. 1306-1309, 1984.
- [34] G. Schalk, J. Mellinger, *A Practical Guide to Brain-Computer Interfacing with BCI2000*, London: Springer, 2010.

- [35] D. J. McFarland, L. M. McCane, S. V. David, J. R. Wolpaw, "Spatial filter selection for EEG-based communication," *Electroenceph. Clin. Neurophysiol.*, vol. 103, pp. 386–394, September 1997.
- [36] K. Li, R. Sankar, Y. Arberl, E. Donchin, "Single trial independent component analysis for P300 BCI system", *Engineering in Medicine and Biology Society, EMBC 2009. Annual International Conference of the IEEE*, pp. 4035-4038.
- [37] R. T. C. Lee, Y. H. Chin, S. C. Chang, "Application of Principal Component Analysis to Multikey Searching", *IEEE Transactions on Software Engineering*, vol. SE-2, no.3, pp.185-193, September 1976.
- [38] E. Donchin, K. M. Spencer, R. Wijesinghe, "The Mental Prosthesis: Assessing the Speed of a P300-Based Brain-Computer Interface", *IEEE Trans. Rehab. Eng.*, vol. 8, no. 2, pp. 174–179, June 2000.
- [39] L. A. Farwell, E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials". *Electroencephalography and clinical Neurophysiology*, vol. 70, no 6, p. 510-523, 1988.
- [40] M. Kaur, P. Ahmed, M. Q. Rafiq, "Analysis of Extracting Distinct Functional Components of P300 using Wavelet Transform", *Proceedings of 4th MMES and the 2nd ICDDCC*, pp. 57-62, June 2013.
- [41] D. J. Krusienski, E.W. Sellers, F. Cabestaing, S. Bayouth, D. J. McFarland, T. M. Vaughan, J. R. Wolpaw, *A comparison of classification techniques for the P300 Speller. Journal of neural engineering*, vol. 3, no 4, pp. 299, 2006.
- [42] S. Haykin, "Support Vector Machines", in *Neuronal networks. A comprehensive foundation*, 2nd ed. Canada: Prentice Hall International, pp. 340–372, 1999.
- [43] M. Thulasidas, C. Guan, J. Wu, "Robust classification of EEG signal for brain-computer interface", *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 2006, vol. 14, no 1, p. 24-29, 2006.
- [44] N. R. Draper, H. Smith, "Selecting the Best Regression Equation", in *Applied Regression Analysis*, 3rd ed. New York: John Wiley & Sons, pp. 335–339, 1981.

ÍNDICE DE ACRÓNIMOS

A continuación se recogen todos los acrónimos empleados a lo largo de este trabajo de fin de grado por orden alfabético, acompañados de su descripción tanto en inglés como en español.

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
AUC	<i>Area Under Curve</i> Área Bajo la Curva
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
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
FLD	<i>Fisher's Linear Discriminant</i> Discriminante Lineal de Fisher
fMRI	<i>Functional Magnetic Resonance Imaging</i> Imagen por Resonancia Magnética Funcional
HMM	<i>Hidden Markov Model</i> Modelo Oculto de Markov
ICA	<i>Independent Component Analysis</i> Análisis de Componentes Independientes
LDA	<i>Linear Discriminant Analysis</i> Análisis Discriminante Lineal
LMS	<i>Least Mean Square</i> Mínimos Cuadrados Promediados
LVQ	<i>Linear-Vector Quantization</i> Aprendizaje por Cuantificación Vectorial
MEG	<i>Magnetoencephalography</i> Magnetoencefalografía
MSE	<i>Mean Square Error</i> Error Cuadrático Medio
mVEP	<i>Motion-onset Visual Evoked Potentials</i> Potenciales Evocados Visuales de Movimiento
NPV	<i>Negative Predictive Value</i> Valor Predictivo Negativo
PCA	<i>Principal Component Analysis</i> Análisis de Componentes Principales
PCM	<i>Pearson's Correlation Method</i> Método de Correlación de Pearson
PET	<i>Positron Emission Tomography</i> Tomografía por Emisión de Positrones
PLS	<i>Partial Least Squares</i> Mínimos Cuadrados Parciales
PPV	<i>Positive Predictive Value</i> Valor Predictivo Positivo
QDA	<i>Quadratic Discriminant Analysis</i> Análisis Discriminante Cuadrático
RLS	<i>Recursive Least Squares</i> Mínimos Cuadrados Recursivos
RT	<i>Reaction Time</i> Tiempo de Reacción
SCP	<i>Slow Cortical Potentials</i> Potenciales Corticales Lentos
STFT	<i>Short-Time Fourier Transform</i> Transformada de Fourier de Tiempo Corto

SWLDA	<i>StepWise Linear Discriminant Analysis</i> Análisis Discriminante Lineal Paso-a-paso
SVM	<i>Support Vector Machines</i> Máquinas de Vectores de Soporte
TTD	<i>Thought Translation Device</i> Dispositivo de Interpretación del Pensamiento
VEP	<i>Visual Evoked Potentials</i> Potenciales Evocados Visuales
WPT	<i>Wavelet Packet Transform</i> Transformada de Paquetes Wavelet

Anexo B

CÓDIGO GENERADO EN C++ Y JAVASCRIPT

A continuación se recoge el código correspondiente a la aplicación desarrollada durante el presente trabajo fin de grado.

B.1. Aplicación BCI Browser: Navigator

Para la programación de la extensión de *Google Chrome* se ha utilizado el lenguaje JavaScript, cuyo código se indica a continuación.

```
/*-----*/
/*--          B C I   B R O W S E R   :   N A V I G A T I O N          --*/
/*-----*/
// Aplicación que se encarga de procesar los comandos de teclado //
// recibidos del BCI Browser: P3Speller y de traducirlos en comandos//
// del navegador Google Chrome. //
//      Versión: 1.5 //
//      Fecha:   27/03/2014 //
// //
// // Universidad de Valladolid (Spain)//
// // Grupo de Ingeniería Biomédica//
// // Víctor Martínez Cagigal//
// //-----//

/* DEFINICIÓN DE CONSTANTES */
//Clase predeterminada
const HINT_CLASS = "chrome_bcibrowser_hint",//
// Clase de la selección
HINT_CLASS_SEL = "chrome_bcibrowser hintsel",
HINTS_ID = "chrome_bcibrowser_hints"; // IDs de los tags
BUFFER_ID = "chrome_bcibrowser_buffer"; // ID del buffer
BUFFER_FORM = "buffer_form"; // ID del formulario

// Códigos de las dos teclas a usar
const KEY_OEM_PLUS = 187, KEY_LEFT= 37;
// Caracteres de codificación
const DEF_CHARS = "012345";

/* DEFINICIÓN DE VARIABLES LOCALES */
var hintChars = localStorage["hintchars"];
if (!hintChars) {hintChars = "012345";}
var mode = undefined;
var maxCod = 0;
var prevSelected = new Array();

function isTextInput(e) {
return ((e.tagName=="INPUT"&&(e.type=="text"||e.type=="password"))
|| e.tagName == "TEXTAREA")
}
```

```

        || e.getAttribute("g_editable") // google groups text box
    );
}

/*****
/** Función emulateMouseClicked(): **/
/** Emula un click del ratón para seleccionar el tag deseado. **/
/** • Parámetros: **/
/** (hint) target: tag deseado. **/
/** ctrlKey, altKey, shiftKey, metaKey: flags. **/
/** • Retorno: null. **/
*****/
function emulateMouseClicked(target,ctrlKey, altKey, shiftKey, metaKey){
    if (isTextInput(target)||target.tagName=="SELECT"){target.focus();}

    else {
        var event = document.createEvent("MouseEvents");
        event.initMouseEvent("click", true, true, window,0,0,0,0,0,
            ctrlKey, altKey, shiftKey, metaKey, 0, null);
        target.dispatchEvent(event);
    }
}

/*****
/** Función HintMode(): **/
/** Función main de la aplicación. **/
/** • Parámetros: null. **/
/** • Retorno: **/
/** (string) mode: modo de operación. **/
*****/
function HintMode() {
    var nodes, hints, hintsDiv, input;

    /* INICIALIZACIÓN */
    this.init = function() {
        hints = [];
        input = "";

        // Busca todos los nodos posibles
        var i, j, sel, elem;
        nodes = [document.getElementsByTagName("a"),
            document.getElementsByTagName("input"),
            document.getElementsByTagName("textarea"),
            document.getElementsByTagName("select"),
            document.getElementsByTagName("img"),
            document.getElementsByTagName("button"),
            // google new forms
            document.getElementsByClassName("gbqfif"),
            // github dropdown menus
            document.getElementsByClassName("select-menu-button"),
            // bbc news titles
            document.getElementsByClassName("headline"),
            // google groups edit box
            [document.getElementById("p-b-0")]
        ];
        sel = [];
        for (i = 0; i < nodes.length; i++) {
            // Grupos de elementos
            elem = nodes[i];
            // Se analiza cada elemento individual
            for (j = 0; j < elem.length; j++) {

```

```

        if(!elem[j]) continue;
        sel.push(elem[j]);
    }
}

nodes = [];
var rep = 1;

// Detecta si estamos en Gmail o en una búsqueda de Google
var url = document.URL;
var gmail = false;
var google = false;
if( url.search("mail.google.com") != -1 ){ gmail = true; }
if( url.search("www.google") != -1 ){    google = true; }

// Optimiza la visualización de los hipervínculos
for (i = 0; i < sel.length; i++) {
    elem = sel[i];
    if ((elem.tagName == "A" && !elem.href) ||
        (elem.tagName == "INPUT" && elem.type == "hidden") ||
        (elem.tagName == "IMG" && !elem.onclick)) {continue; }

// Evita que aparezcan etiquetas de más en los formularios de Google
if (elem.className == "gbqfb"){continue; }
if (elem.id == "gs_htif0"){continue; }
if (rep == 1){
    if (elem.className == "gbqfif" && elem.id == "gbqfq"){
        rep = 0;
        continue;
    }
}
    // Interfaz de Gmail
if (elem.id == "gs_tai50"){continue; }

    // Elementos que sí son útiles
// En Gmail los a.href de los mails no tienen longitud
if( gmail ){ nodes.push(elem); }
else{
    if ( google ) {
        // Optimización de las búsquedas de Google
        if(elem.className == "ab_dropdownlnk" ||
            elem.className == "fl" || elem.className ==
            "clickable-dropdown-arrow ab_button")
            {continue; }
    } // Fin if
    if(elem.offsetWidth>0&&elem.offsetHeight>0)
    {nodes.push(elem); }
} // Fin else
} // Fin for

// Crea el buffer
bufferDiv = document.createElement("div");
bufferDiv.id = BUFFER_ID;
document.body.appendChild(bufferDiv);
bufferDiv.style.position = "fixed";
bufferDiv.style.top = "0px";
bufferDiv.style.left = "0px";
bufferDiv.style.width = "160px";
bufferDiv.style.height = "28px";
bufferDiv.style.backgroundColor = "#DFDFDF";
bufferDiv.style.borderBottom = "1px solid #AAAAAB";

```

```

bufferDiv.style.borderRight = "1px solid #AAAAAB";
bufferDiv.style.borderBottomRightRadius = "15px";
bufferDiv.style.zIndex = "2147483647";
bufferDiv.innerText = "Buffer";
bufferDiv.style.textAlign = "center";
bufferDiv.style.fontFamily = "Segoe UI";
bufferDiv.style.color = "#AAAAAB";
bufferDiv.style.fontSize = "10pt";
bufferDiv.style.lineHeight = "26px";

// Div hijo del formulario
bufferForm = document.createElement("div");
bufferForm.id = BUFFER_FORM;
bufferDiv.appendChild(bufferForm);
    bufferForm.style.float = "right";
    bufferForm.style.backgroundColor = "white";
    bufferForm.style.margin = "3px";
    bufferForm.style.width = "95px";
    bufferForm.style.height = "20px";
    bufferForm.style.border = "1px solid #AAAAAB";
    bufferForm.style.borderBottomRightRadius = "11px";
    bufferForm.style.textAlign = "left";
    bufferForm.style.fontFamily = "Segoe UI";
    bufferForm.style.color = "black";
    bufferForm.style.fontSize = "10pt";
    bufferForm.style.lineHeight = "20px";
    bufferForm.style.paddingLeft = "5px";
// Fin del buffer

// Crea los div que portarán los hipervínculos
hintsDiv = document.createElement("div");
hintsDiv.id = HINTS_ID;
document.body.appendChild(hintsDiv);

// Crea las etiquetas
var node, cr, span, hint;
for (var i = 0; i < nodes.length; i++) {
    node = nodes[i];
    cr = node.getBoundingClientRect();
    span = document.createElement("span");
    hint = hints[i]={ "str": nts(i), "node": node, "span": span };

// Cada etiquetado muestra el valor de nts(i)
span.innerText = hint["str"];
span.className = HINT_CLASS;
hintsDiv.appendChild(span);

span.style.border = "1px solid #AAAAAB";
span.style.opacity = "0.8";

var topUnderBuff=Math.max(window.pageYOffset+cr.top-10,0);
var leftUnderBuff=Math.max(window.pageXOffset+cr.left-
span.offsetWidth - 10, 0);

if( topUnderBuff < 30){
// Evita que las etiquetas queden debajo del buffer
span.style.top = "22px";
span.style.left = Math.max(window.pageXOffset +
cr.left - span.offsetWidth + 20, 0) + "px";
span.style.border = "1px solid #318BC4";
if( leftUnderBuff < 160 ){

```

```

// Si los links los oculta el buffer que no aparezcan
span.style.opacity = "0";
    }
}
else{
span.style.top = topUnderBuff + "px";
span.style.left = leftUnderBuff + "px";
}

span.style.position = "absolute";
span.style.color = "black";
span.style.backgroundColor = "#DFDFDF";
span.style.paddingLeft = "5px";
span.style.paddingRight = "5px";
span.style.borderRadius = "5px";
span.style.margin = "0";
span.style.fontSize = "11pt";
span.style.lineHeight = "18pt";
span.style.position = "absolute";
span.style.zIndex = "2147483646";
} // Fin del bucle
}

/* GESTIÓN DE PULSACIONES DE TECLADO */
this.handleKeyDown = function(e, modifier) {
    switch (e.keyCode) {
        case KEY_OEM_PLUS: // Tecla '+'
            if (!modifier) {
                e.preventDefault();
                this.hide();
                mode = undefined;
            }else{sleep(100);}
            break;
        case KEY_LEFT: //Tecla que simula el 'Backspace'
            if(!modifier){
                var buffer = document.getElementById(BUFFER_FORM);
                // Si no está activado el buffer debería tener
                su función original: volver a la anterior web
                if(buffer){
                    e.preventDefault();
                    // Borrar un carácter del buffer
                    var buff="";
                    if (input.length == 1| input.lengt == 0)
                        {input="";}
                    else{
                        for(var i=0;i<(input.length-1);i++)
                            {buff+=input[i];}
                        input = buff;
                    }
                    buffer.innerText = input;
                    bufferString = buffer.innerText;
                    selectingTags.call(this, bufferString);
                }
            }
            break;
    }
} // Fin de HandleKeyDown

// Para teclas acordes con DEF_CHARS [012345]
var c = String.fromCharCode(e.keyCode);
if (!modifier && hintChars.indexOf(c) >= 0) {
    e.preventDefault();
}

```

```

        input += c;

        // Se actualiza el buffer según la input
        var buffer = document.getElementById(BUFFER_FORM);
        buffer.innerText = input;
        bufferString = buffer.innerText;

        // Cuando se haya seleccionado un tag retorna su "hint" y
        se emula el click del ratón
        var hint = selectingTags.call(this, bufferString);
        if(hint){
            emulateMouseClicked(hint["node"], e.ctrlKey, e.altKey,
e.shiftKey, e.metaKey);
            // Se ocultan los spans anteriores
            this.hide();
            e.preventDefault();
            // Se establece un nuevo modo
            mode = new HintMode();
            mode.init();
            // Se guarda el buffer anterior por estética
            var buffer = document.getElementById(BUFFER_FORM);
            buffer.innerText = bufferString;
        }
    }
}

/* ELIMINACIÓN DE LAS ETIQUETAS */
this.hide = function() {
    document.body.removeChild(hintsDiv);
    document.body.removeChild(bufferDiv);
    hints = hintsDiv = null;
    input = "";
}

/*****
/** Función selectingTags(buffer):                               **/
/** Se encarga de seleccionar los tags en cada momento y detectar**/
/** si es una selección final o intermedia, además de adaptar el **/
/** color de los mismos.                                         **/
/** • Parámetros:                                               **/
/** (string) buffer: buffer que determina la entrada del teclado.**/
/** • Retorno:                                                  **/
/** (hint): en el caso de que la selección sea una selección   **/
/** final, retorna el hint a seleccionar.                       **/
*****/
function selectingTags(buffer) {
    var numArray = new Array();
    numArray = searchLight(buffer);
    var dist = 0;

    // Si se ha borrado un carácter y el buffer no contiene nada,
    que no se ilumine nada
    if(buffer == ""){dist = 1;}

    //Compruebo si la selección no es la misma y si no lo es
    recupero el color original de los tags
    if(numArray.length != prevSelected.length){dist = 1;}
    else{

```

```

        for(var g = 0; g<prevSelected.length; g++){
            if(numArray[g] != prevSelected[g]){dist = 1;}
        }
    }
    if(dist == 1){
        for(var d = 0; d < prevSelected.length ; d++){
            var hintSel = hints[parseInt(prevSelected[d])];
            if (hintSel) {
                hintSel["span"].className = HINT_CLASS;
                hintSel["span"].style.backgroundColor="#DFDFDF";
            }
        }
    }
    prevSelected.length = 0; //Limpio el array
    if(buffer == ""){return null;}

    //Ilumino los tags
    for(var v = 0; v < numArray.length ; v++){
        var num = parseInt(numArray[v]);
        if (!isNaN(num)) {
            // Paso a base decimal
            var dec = 0;
            for(var n=0; n<numArray[v].length; n++){
                var cifra =
                parseInt(numArray[v].charAt(numArray[v].length-1-n));
                var op = 1;
                for(var l=0; l<n; l++){ op *= hintChars.length;}
                dec = dec + cifra * op;
            }
        }
        // Guardo las selecciones de todos los tags para poder borrarlas más
        tarde

        prevSelected[v] = dec;

        // Ilumino los tags correspondientes
        var hint = hints[dec];
        if (hint) {
            //Color verde
            hint["span"].style.backgroundColor = "#239A39";
        }
    }
}

// Si solamente se ha seleccionado uno es el deseado
if(prevSelected.length == 1){
    var hint = hints[parseInt(prevSelected[0])];
    if (hint) {
        hint["span"].className = HINT_CLASS_SEL;
        // Color rojo
        hint["span"].style.backgroundColor = "#B83636";
        return hint;
    }
}
return null;
}

```

```

/*****
/** Función nts(num):
/** Halla la string que contendrá cada uno de los tags con el
/** formato adecuado.
/** • Parámetros:
/** (int) num: índice del tag.
/** • Retorno:
/** (string): string que contendrá el tag
*****/
function nts(num) {
    // Convierte a num en su correspondiente base no decimal según
    // el número de caracteres de codificación
    var tmp = num.toString(hintChars.length);
    var str = "";
    // Número de caracteres disponibles para codificar
    var n_cod = hintChars.length;
    // Número de nodos a etiquetar
    var n_tag = nodes.length;

/**Detección del número de nodos y del rango de codificación a usar**/
    var end = 0;
    // El bucle comparará el n_tag con el n_cod^k y cuando sea
    // menor o igual que éste parará
    for(var k = 1; end == 0; k++){

        var op = 1; //var op = n_cod^k;
        for(var j = 0; j < k; j++){op *= n_cod;}

        if(n_tag <= op){ //Determina el rango a usar
            end = 1;
            var rango = k;
        } //El rango determina: n_cod^(rango-1) < n_tag <= n_cod^(rango)
    }

    // Valor máximo a codificar (var maxi)
    var maxi = 1;
    for(var m = 0; m < rango; m++){maxi *= n_cod;}

    // Número de caracteres necesarios máximos por TAG
    var zero = maxi.toString().length;
    // Ceros a añadir en cada codificación
    var add = zero - tmp.length;
    maxCod = zero;

/**Introducción del número de ceros previos si son necesarios**/
    var head = "";
    for(var u = 0; u < add; u++){head += "0";}
    var tmp = head + tmp;

/**Introducción de la codificación**/
    for (var i = 0; i < tmp.length; i++) {
        str += hintChars[DEF_CHARS.indexOf(tmp.charAt(i))];
    }
    return str;
}

```

```

/*****
/** Función searchLight(buffer): **/
/** Halla el número de tags a seleccionar según el buffer y lo **/
/** devuelve como array. **/
/** • Parámetros: **/
/** (string) buffer: buffer que determina la entrada de teclado. **/
/** • Retorno: **/
/** (Array) num: array que contiene todos los tags seleccionados.**/
*****/
function searchLight(buffer) {
    var spans = document.getElementsByClassName(HINT_CLASS);
    var text = new Array();
    var num = new Array();
    var count = 0;
    var bufferLength = buffer.length;

    for(var i = 0; i < spans.length; i++){
        text[i] = spans[i].innerText;
        var comp = "";
        for(var l=0; l < buffer.length; l++)
            {comp+=text[i].charAt(l);}
        if( buffer == comp ){
            num[count] = text[i];
            count++;
        }
    }
    return num;
}
/* ESTABLECIMIENTO DE INTERVALO DE GUARDA */
function sleep(milliSeconds){
    var startTime = new Date().getTime(); // Hora actual
    while (new Date().getTime() < startTime + milliSeconds);
}
}
/* DETECCIÓN DE PULSACIONES */
document.addEventListener('keydown', function(e) {
    var modifier = (e.ctrlKey || e.altKey || e.shiftKey || e.metaKey);
    var active = document.activeElement;
    if (active && isTextInput(active)) {
        //Quita el focus a un formulario si se pulsa VK_LEFT
        if (e.keyCode == KEY_LEFT && !modifier) {
            e.preventDefault();
            active.blur();
        }
        // No escribe un + al pulsar TAG en un formulario
        if(e.keyCode==KEY_OEM_PLUS && !modifier){e.preventDefault();}
    }
    else if (mode == undefined) {
        if (e.keyCode == KEY_OEM_PLUS && !modifier) {
            e.preventDefault();
            mode = new HintMode();
            mode.init();
        }
    }
    } else {mode.handleKeyDown(e, modifier);}
}, true);

/* ESTABLECIMIENTO DE INTERVALO DE GUARDA */
function sleep(milliSeconds){
    var startTime = new Date().getTime(); // Hora actual
    while (new Date().getTime() < startTime + milliSeconds);
}
}

```

B.2. Aplicación BCI Browser: P3Speller

Para la programación del *P3Speller* se han modificado una serie de archivos dentro del código fuente de BCI2000, concretamente '*KeystrokeFilter.cpp*', '*P3SpellerTask.cpp*', '*Association.cpp*' y '*Association.h*'.

B.2.1. KeystrokeFilter.cpp

```

////////////////////////////////////
// $Id$
// Autor:           Víctor Martínez Cagigal
// E-mail:          victorlawliet@gmail.com
// Fecha:           4 de Marzo de 2014
// Descripción: Filtro que se encarga de detectar los cambios de estado de la
//               aplicación y genera pulsaciones de teclado correspondientes a
//               teclas concretas que se traducirán en comandos posteriormente.
//
////////////////////////////////////
#include "PCHIncludes.h"
#pragma hdrstop

#include "KeystrokeFilter.h"

#include <string>
#include <windows.h>

using namespace std;
RegisterFilter( KeystrokeFilter, 3.1 );

double preRes; //Usado para evitar enviar varias copias de la misma keystroke
int cur;

/*Constructor*/
KeystrokeFilter::KeystrokeFilter()
: mPreviousStateValue( 0 ),
  mKeystrokeStateName( "" )
{
  BEGIN_PARAMETER_DEFINITIONS
    "Application:Human%20Interface%20Devices string KeystrokeStateName= % "
    "// State to be translated into keystrokes (0-F), empty for off",
    "Application:Human%20Interface%20Devices string KeystrokeExpression= % "
    "// Expression that evaluates to a virtual keycode, empty for off",
    "Application:Human%20Interface%20Devices string
KeystrokeExpressionOnStartRun= % "
    "// Expression that initializes expression variables on StartRun",
  END_PARAMETER_DEFINITIONS
}

/*Destructor*/
KeystrokeFilter::~KeystrokeFilter()
{
}

void
KeystrokeFilter::Preflight( const SignalProperties& Input,
                           SignalProperties& Output ) const
{
  //Compruebo la existencia de los estados a utilizar
  State("SpellerMenu");
  State("SelectedRow");
}

```

```

    State("SelectedColumn");

    Output = Input;
}

void
KeystrokeFilter::Initialize( const SignalProperties&, const SignalProperties& ){

void
KeystrokeFilter::StartRun(){
    preRes = 0; //Inicializo el 'preRes'
    cur = 1;
}

void
KeystrokeFilter::Process( const GenericSignal& Input, GenericSignal& Output )
{
    /*Envío una u otra key según la celda seleccionada*/
    double result = 0;
    /*_____Matriz de Navegación_____*/
    if( State("SpellerMenu") == 1 ){
        //Fila #1
        if ( State("SelectedRow") == 1 ){
            switch ( State("SelectedColumn") ){
                case 1:
                    result = 187; //TAG [VK_OEM_PLUS]
                    break;
                case 2:
                    result = 0; //MTX Cambio de Matriz -> Matriz Teclado
                    break;
                case 3:
                    result = 166; //BROWSER BACK KEY [VK_BROWSER_BACK]
                    break;
            }
        }
        //Fila #2
        if ( State("SelectedRow") == 2 ){
            switch ( State("SelectedColumn") ){
                case 1:
                    result = 168; //RELOAD [VK_BROWSER_REFRESH]
                    break;
                case 2:
                    result = 33; //RE PAG [VK_PRIOR]
                    break;
                case 3:
                    result = 34; //AV PAG [VK_NEXT]
                    break;
            }
        }
        //Fila #3
        if ( State("SelectedRow") == 3 ){
            switch ( State("SelectedColumn") ){
                case 1:
                    result = 172; //HOME [VK_BROWSER_HOME]
                    break;
                case 2:
                    result = 37; //LEFT ARROW KEY [VK_LEFT]
                    break;
                case 3:
                    result = 39; //RIGHT ARROW KEY [VK_RIGHT]
                    break;
            }
        }
    }
}

```

```
    }
    //Fila #4
    if ( State("SelectedRow") == 4 ){
        switch ( State("SelectedColumn") ){
            case 1:
                result = 48; //Número 0
                break;
            case 2:
                result = 49; //Número 1
                break;
            case 3:
                result = 50; //Número 2
                break;
        }
    }
    //Fila #5
    if ( State("SelectedRow") == 5 ){
        switch ( State("SelectedColumn") ){
            case 1:
                result = 51; //Número 3
                break;
            case 2:
                result = 52; //Número 4
                break;
            case 3:
                result = 53; //Número 5
                break;
        }
    }
} //Fin de la navegación
/*_____Matriz de Teclado_____*/
if( State("SpellerMenu")==2 ){
    //Fila #1
    if ( State("SelectedRow") == 1 ){
        switch ( State("SelectedColumn") ){
            case 1:
                result = 65; //Letra A
                break;
            case 2:
                result = 66; //Letra B
                break;
            case 3:
                result = 67; //Letra C
                break;
            case 4:
                result = 68; //Letra D
                break;
            case 5:
                result = 69; //Letra E
                break;
        }
    }
    //Fila #2
    if ( State("SelectedRow") == 2 ){
        switch ( State("SelectedColumn") ){
            case 1:
                result = 70; //Letra F
                break;
            case 2:
                result = 71; //Letra G
                break;
        }
    }
}
```

```
        case 3:
            result = 72; //Letra H
            break;
        case 4:
            result = 73; //Letra I
            break;
        case 5:
            result = 74; //Letra J
            break;
    }
}
//Fila #3
if ( State("SelectedRow") == 3 ){
    switch ( State("SelectedColumn") ){
        case 1:
            result = 75; //Letra K
            break;
        case 2:
            result = 76; //Letra L
            break;
        case 3:
            result = 77; //Letra M
            break;
        case 4:
            result = 78; //Letra N
            break;
        case 5:
            result = 79; //Letra O
            break;
    }
}
//Fila #4
if ( State("SelectedRow") == 4 ){
    switch ( State("SelectedColumn") ){
        case 1:
            result = 80; //Letra P
            break;
        case 2:
            result = 81; //Letra Q
            break;
        case 3:
            result = 82; //Letra R
            break;
        case 4:
            result = 83; //Letra S
            break;
        case 5:
            result = 84; //Letra T
            break;
    }
}
//Fila #5
if ( State("SelectedRow") == 5 ){
    switch ( State("SelectedColumn") ){
        case 1:
            result = 85; //Letra U
            break;
        case 2:
            result = 86; //Letra V
            break;
        case 3:
            result = 87; //Letra W
```

```
        break;
    case 4:
        result = 88; //Letra X
        break;
    case 5:
        result = 89; //Letra Y
        break;
    }
}
//Fila #6
if ( State("SelectedRow") == 6 ){
    switch ( State("SelectedColumn") ){
    case 1:
        result = 90; //Letra Z
        break;
    case 2:
        result = 48; //Número 0
        break;
    case 3:
        result = 49; //Número 1
        break;
    case 4:
        result = 50; //Número 2
        break;
    case 5:
        result = 51; //Número 3
        break;
    }
}
//Fila #7
if ( State("SelectedRow") == 7 ){
    switch ( State("SelectedColumn") ){
    case 1:
        result = 52; //Número 4
        break;
    case 2:
        result = 53; //Número 5
        break;
    case 3:
        result = 54; //Número 6
        break;
    case 4:
        result = 55; //Número 7
        break;
    case 5:
        result = 56; //Número 8
        break;
    }
}
//Fila #8
if ( State("SelectedRow") == 8 ){
    switch ( State("SelectedColumn") ){
    case 1:
        result = 57; //Número 9
        break;
    case 2:
        result = 334; //Barra baja (Underline) [UNICODE 95]
        break;
    case 3:
        result = 190; //Punto (Point) [VK_OEM_PERIOD]
        break;
    case 4:
```

```

        result = 13; //Enter [VK_RETURN]
        break;
    case 5:
        result = 32; //Barra Espaciadora [VK_SPACE]
        break;
    }
}
//Fila #9
if ( State("SelectedRow") == 9 ){
    switch ( State("SelectedColumn") ){
    case 1:
        result = 335; //Arroba (At) [UNICODE 64]
        break;
    case 2:
        result = 8; //Retroceso (BackSpace)
        break;
    case 3:
        result = 189; //Guión (Dash) [VK_OEM_MINUS]
        break;
    case 4:
        result = 188; //Coma (Comma) [VK_OEM_COMMA]
        break;
    case 5:
        result = 0; //MTX Cambio de Matriz-> Matriz Navegación
        cur = 1;
        break;
    }
}
} //Fin del teclado

/*Se envía el resultado*/
if( preRes != result){
    // Evita que la transición entre estados genere keystrokes no deseadas
    if( !( result==66 && cur!=State("SpellerMenu") ))
        {SendKeystroke( result );}
}
cur = State("SpellerMenu");

preRes = result; //Evita el enviar copias de la misma keystroke
Output = Input;
}

void
KeystrokeFilter::SendStateKeystroke( State::ValueType s )
{
    const short unicodeChars[] =
    {
        '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
        'A', 'B', 'C', 'D', 'E', 'F',
    };
};
const int numChars = sizeof( unicodeChars ) / sizeof( *unicodeChars );
if( s < 0 || s >= numChars )
{
    bciout << "Only state values between 0 and "
        << numChars - 1
        << " may be sent as keystrokes"
        << endl;
    return;
}
}

```

```

KEYBDINPUT keyEvents[] =
{
    { 0, unicodeChars[ s ], KEYEVENTF_UNICODE, 0, 0 },
    { 0, unicodeChars[ s ], KEYEVENTF_UNICODE | KEYEVENTF_KEYUP, 0, 0 },
};
const int numInputs = sizeof( keyEvents ) / sizeof( *keyEvents );
INPUT inputs[ numInputs ];
for( int i = 0; i < numInputs; ++i )
{
    inputs[ i ].type = INPUT_KEYBOARD;
    inputs[ i ].ki = keyEvents[ i ];
}
if( ::SendInput( numInputs, inputs, sizeof( *inputs ) ) != numInputs )
    bciout << "Could not send keystroke for state value '"
        << s << "'" << endl;
}

void
KeystrokeFilter::SendKeystroke( double inKeyCode )
{
    //Tratamiento normal de la key
    WORD keyCode = static_cast<WORD>( inKeyCode );
    if( keyCode > 0 && keyCode < 255 )
    {
        KEYBDINPUT keyEvents[] =
        {
            { keyCode, 0, 0, 0, 0 },
            { keyCode, 0, KEYEVENTF_KEYUP, 0, 0 },
        };
        const int numInputs = sizeof( keyEvents ) / sizeof( *keyEvents );
        INPUT inputs[numInputs];

        for( int i = 0; i < numInputs; ++i )
        {
            inputs[i].type = INPUT_KEYBOARD;
            inputs[i].ki = keyEvents[i];
        }
        //Busca el Google Chrome
        HWND chromeWindow = FindWindow("Chrome_WidgetWin_1", NULL);
        HWND chrome = GetWindow(chromeWindow, GW_HWNDNEXT);
        SetForegroundWindow(chrome);

        if( ::SendInput( numInputs, inputs, sizeof( *inputs ) ) != numInputs )
            bciout << "Could not send keystroke for virtual key code 0x" << hex
                << keyCode << endl;
    }
    else{
        //Tratamiento para keys UNICODE ( _ , @ )
        double key = 0;
        if( inKeyCode == 334) {key = 95;} //Tecla _
        if( inKeyCode == 335) {key = 64;} //Tecla @
        WORD keyUnicode = static_cast<WORD>( key );
        KEYBDINPUT keyEvents[] =
        {
            { 0, keyUnicode, KEYEVENTF_UNICODE, 0, 0 },
            { keyCode, 0, KEYEVENTF_UNICODE | KEYEVENTF_KEYUP, 0, 0 },
        };
        const int numInputs = sizeof( keyEvents ) / sizeof( *keyEvents );
        INPUT inputs[numInputs];

        for( int i = 0; i < numInputs; ++i )
        {

```

```

        inputs[i].type = INPUT_KEYBOARD;
        inputs[i].ki = keyEvents[i];
    }

    //Busca el Google Chrome
    HWND chromeWindow = FindWindow("Chrome_WidgetWin_1", NULL);
    HWND chrome = GetWindow(chromeWindow, GW_HWNDNEXT);
    SetForegroundWindow(chrome);

    if( ::SendInput( numInputs, inputs, sizeof( *inputs ) ) != numInputs )
        bciout << "Could not send keystroke for virtual key code 0x" << hex
                << keyCode << endl;
    }
}

```

B.2.2. P3SpellerTask.cpp

Debido al gran número de líneas de código que incluye este archivo, a continuación se muestran únicamente las modificaciones que se han realizado para optimizar el funcionamiento de la aplicación desarrollada en el presente trabajo fin de grado.

Para empezar, se incluye el parámetro “Threshold” para poder introducir el umbral dentro del operador, dentro de `BEGIN_PARAMETER_DEFINITIONS`.

```

"Application:Speller%20Targets float Threshold= 0 883 0 % // "
"threshold which determines the selections",

```

Posteriormente, se inicializan los estados que determinan la fila seleccionada (*SelectedRow*), la columna seleccionada (*SelectedColumn*) y la celda seleccionada (*SelectedTarget*).

```

void
P3SpellerTask::OnSequenceBegin()
{
    State( "SelectedRow" ) = 0;
    State( "SelectedColumn" ) = 0;
    State( "SelectedTarget" ) = 0;
}

void
P3SpellerTask::OnPostRun()
{
    State( "SelectedRow" ) = 0;
    State( "SelectedColumn" ) = 0;
    State( "SelectedTarget" ) = 0;
}

```

Para finalizar, se modifica el método *onClassResult* para que determine el valor del umbral, almacenado en un objeto de la clase *Association* (ver el código de *Association.cpp* y *Association.h*). Previamente se ha introducido: `#include <math.h>`

```

Target*
P3SpellerTask::OnClassResult( const ClassResult& inResult )
{
    Stimulus* pClickedStimulus = NULL;
    while( !Display().ObjectsClicked().empty() )
    {
        Stimulus* pStimulus=dynamic_cast<Stimulus*>(Display().ObjectsClicked().front() );
        if( pStimulus != NULL )
            pClickedStimulus = pStimulus;
    }
}

```

```

    Display().ObjectsClicked().pop();
}

Target* pTarget = NULL;
if( mTestMode && pClickedStimulus != NULL )
{ // Fake an ideal ERP result, i.e. a binary response to the clicked
stimulus.
    // This allows for testing result processing as well.
    ClassResult fakeResult;
    GenericSignal fakeSignal( 1, 1 );
    for( AssociationMap::const_iterator i = Associations().begin();
        i != Associations().end(); ++i )
    {
        fakeSignal( 0, 0 ) = i->second.Contains( pClickedStimulus );
        fakeResult[ i->first ].push_back( fakeSignal );
    }
    pTarget = StimulusTask::OnClassResult( fakeResult );
    // fakeSignal vale 1 siempre que se clique en una celda, ya sea
fakeSignal(0,0) o fakeSignal(1,1)
}
else{
    pTarget = StimulusTask::OnClassResult( inResult );
}

// Determina si se supera el umbral
Association ass;
float logprob = ass.GetLogprob(); // Consigue el valor de logprob
de la celda seleccionada
bool selection = false;

if( Parameter("Threshold") > logprob ){ // No se supera el umbral
    State( "SelectedTarget" ) = 0;
    State( "SelectedRow" ) = 0;
    State( "SelectedColumn" ) = 0;
    selection = false;
}
else{
    // Se supera el umbral
    selection = true;
    int targetID = pTarget ? pTarget->Tag() : 0;
    State( "SelectedTarget" ) = targetID;

    /*Corrección del valor de SelectedRow*/
    float rowID = ( targetID - 1 ) / mNumMatrixCols + 1;
    State( "SelectedRow" ) = targetID ? floor(rowID) : 0;
    /***/

    State("SelectedColumn")=targetID ? (targetID-1)%mNumMatrixCols+1: 0;
}

// Write classification signal details into the application log.
size_t numAverages = 0;
for( ClassResult::const_iterator i = inResult.begin(); i !=
inResult.end(); ++i )
    numAverages += i->second.size();
AppLog << "This is the end of this sequence: "
    << numAverages * mNumberOfSequences << " total intensifications"
    << endl;

// Muestra si se ha seleccionado algo realmente o no.
if(!selection) {
    AppLog << "WARNING: Stimulus does not exceed the threshold" << endl;}

```

```

// Report mean responses to log file but not to screen log.
AppLog.File << "Mean responses for each stimulus:\n";
for( ClassResult::const_iterator i = inResult.begin(); i !=
inResult.end(); ++i )
{
float mean = 0.0;
for( size_t j = 0; j < i->second.size(); ++j )
mean += i->second[ j ]( 0, 0 );
mean /= i->second.size();
AppLog.File << "Response for Stimulus Code " << i->first << ": "
<< setprecision( 2 ) << fixed << mean
<< "\n";
}
return pTarget;
}

```

B.2.3. Association.cpp

Debido al gran número de líneas de código que incluye este archivo, a continuación se muestran únicamente las modificaciones que se han realizado para optimizar el funcionamiento de la aplicación.

Antes de modificar el archivo *Association.cpp* se debe incluir la declaración de la función "float GetLogprob();" dentro de la clase "class Association{}" en el archivo *Association.h*.

Una vez incluida la declaración se modifica el archivo *Association.cpp*. En la definición de constantes y declaración de librerías se incluye lo siguiente.

```

#include <fstream>
#include <math.h>
int trial = 1;
double logprob = 0;

```

Después de incluir las declaraciones se introduce la función que se ha introducido en el archivo de cabecera.

```

// Devuelve el valor de la variable local logprob
float
Association::GetLogprob()
{
float log = (float)logprob;
return log;
}

```

Finalmente, se modifican las funciones *MostLikelyTarget* y *ClassifyTargets* para almacenar los valores de la intensidad de los estímulos que han determinado la celda seleccionada, para modificar el valor del umbral acorde con los resultados almacenados localmente.

```

// TargetClassification
Target*
TargetClassification::MostLikelyTarget( double& outEvidence ) const
{
Target* result = NULL;

double max = std::numeric_limits<double>::quiet_NaN(),
next = max;

```


TAREAS Y CUESTIONARIO

A continuación se recogen los dos folios que se entregaron a cada usuario en el transcurso de la tercera sesión.

C.1. Tareas propuestas

TAREAS PROGRESIVAS

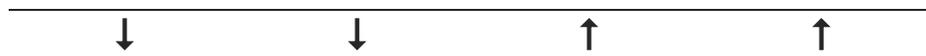
Cada usuario final de la aplicación llevará a cabo tres tareas con dificultad progresiva para evaluar la funcionalidad de la misma.

TAREA 1: VISUALIZACIÓN DE UNA WEB

En la primera tarea el usuario se centrará en visualizar una página web haciendo uso de los comandos de navegación. Además, practicará el salto de una web a otra mediante hipervínculos.

INICIO	SELECCIONES
es.wikipedia.org/wiki/P300	10

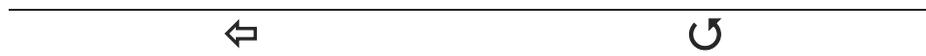
1. Haga *scroll* dos veces hacia abajo y dos veces hacia arriba para simular una visualización completa de la web.



2. Active la detección de hipervínculos mediante el comando "TAG" y seleccione el hipervínculo "electroencefalografía".

TAG	0	0	3
-----	---	---	---

3. Una vez cargada la página de electroencefalografía, vuelva a la página anterior y recárguela.



TAREA 2: ENVIAR UN E-MAIL

Ésta segunda tarea es la que más selecciones requiere, sin embargo, todas se pueden realizar con la matriz de navegación. El usuario se centrará en visualizar un e-mail de su bandeja de entrada y, finalmente, en responderlo.

INICIO	SELECCIONES
mail.google.com/?ui=html	14

1. Seleccione el primer e-mail de su bandeja de entrada para poder leerlo.

TAG	0	1	1	2
-----	---	---	---	---

Nota: La codificación del e-mail puede variar.

2. Lea el e-mail y respóndalo introduciendo un '0' como cuerpo de mensaje. Para ello, seleccione el formulario de respuesta rápida e introduzca el '0'.

TAG	1	5	2	0
-----	---	---	---	---

Nota: La codificación del formulario puede variar.

3. Abandone el formulario seleccionando '←' y seleccione el botón de enviar.

←	1	3	5
---	---	---	---

Nota: La codificación del botón puede variar.

TAREA 3: BÚSQUEDA EN GOOGLE

La tercera tarea es la más complicada, puesto que utiliza ambas matrices. El usuario deberá realizar una búsqueda en Google y seleccionar el primer resultado encontrado.

INICIO	SELECCIONES
www.google.es	13

1. Seleccione el formulario de búsqueda.

TAG	3	0
-----	---	---

2. Cambie de matriz e introduzca la palabra "EEG" en el formulario. Finalmente, pulse '↵' para realizar la búsqueda.

MTX	E	E	G	↵
-----	---	---	---	---

3. Seleccione el primer resultado. Es decir, la página de la Wikipedia en español. Para ello será necesario volver a la matriz de navegación.

MTX	TAG	0	2	3
-----	-----	---	---	---

TAREA 4: LECTURA

La cuarta y última tarea consiste en leer una página web sin fijarse en la matriz durante 5 minutos. Las selecciones no deberán superar el umbral del usuario, provocando, aproximadamente, 11 *warnings*.

C.2. Cuestionario de satisfacción

CUESTIONARIO DE SATISFACCIÓN NAVEGACIÓN WEB MEDIANTE SISTEMAS BCI

Valore las siguientes afirmaciones sobre la aplicación BCI de navegación web:

1. La interfaz del sistema es intuitiva y fácil de entender.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

2. Me ha resultado sencillo seleccionar comandos de la matriz de navegación.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

3. El efecto de parpadeo de los botones de la aplicación resulta molesto.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

4. Me ha resultado complicado seleccionar comandos de la matriz de teclado.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

5. El tiempo necesario para seleccionar un enlace es muy grande.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

6. Habitualmente se seleccionan comandos que no deseo que se seleccionen, complicando el uso de la aplicación.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

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

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

8. La duración de las sesiones realizadas es adecuada.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

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

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

.....

10. ¿Qué cree usted que debería mejorarse para una futura versión?

PLIEGO DE CONDICIONES

Este anexo contiene las condiciones legales que guiarán a la realización del trabajo de fin de grado titulado como “Diseño y desarrollo de una aplicación para navegar por Internet mediante sistemas *Brain Computer Interface* (BCI) orientada a personas con grave discapacidad”.

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 trabajos 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 partidaalzada 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 Intel® Core™ 2 Duo – CPU E4500, con 2.2GHz, 32 bits y 1GB 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 7 Professional (SP1).
Copyright© 2009, Microsoft Corporation.
- Microsoft® Word (14.0.7116.5000).
Microsoft® Office Home and Student 2010.
Copyright© 2010, Microsoft Corporation.
- 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 35.0.1916.114m).
Copyright© 2014, Google Inc.
- Microsoft® Visual Studio Professional (Versión 10.0.30319.1).
Copyright© 2010, Microsoft Corporation.
- CMake (2.8.12.1) con Qt (4.6.2).
Creative Commons (CC), Kitware Inc.
- g.USBamp Simulink High-Speed Online Processing.
- Plataforma de propósito general BCI2000.

Anexo E

PRESUPUESTO

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

EJECUCIÓN MATERIAL

Equipo Intel® Core™ 2 Duo – CPU E4500, 2.2GHz, 32 bits y 1GB de RAM	700€
Equipo BCI	13.300€
Sistema Operativo: Microsoft® Windows 7 Professional (SP1)	110€
Software: Microsoft® Office Home and Student 2010	80€
Software: MATLAB® Student R2012a	427€
Software: Adobe Photoshop CS5 Extended	80€
TOTAL	14.697€

GASTOS GENERALES

16% sobre la ejecución material	2.351,52€
---------------------------------	-----------

BENEFICIO INDUSTRIAL

6% sobre la ejecución material	881,82€
--------------------------------	---------

MATERIAL FUNGIBLE

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

HONORARIOS DEL PROYECTO

450 horas a 30€/hora	13.500€
----------------------	---------

SUBTOTAL DEL PRESUPUESTO

Subtotal del presupuesto	31.752,34€
--------------------------	------------

I.V.A. APLICABLE

21% sobre el subtotal de presupuesto 6.667,99€

TOTAL DEL PRESUPUESTO

Total del presupuesto 38.420,33€

El total del presupuesto, en euros, asciende a:

**TREINTA Y OCHO MIL CUATROCIENTOS VEINTE EUROS CON TREINTA Y TRES
CÉNTIMOS DE EURO.**

En Valladolid, Junio de 2014.

Fdo. Víctor Martínez Cagigal.