



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

**Potenciando la extracción de características en
sistemas brain-computer interface basados en c-VEP
mediante TRCA**

Alumno: Juan Dueñas Ruiz

**Tutores: Víctor Martínez Cagigal
Aníbal Bregón Bregón**

*Dedicado a
todas aquellas personas
que me han acompañado
en este largo camino.*

Agradecimientos

A mi familia, especialmente a mis padres, por creer cuando yo ni si quiera lo hacía, por soportar berrinches y enfados, por ser el amor más puro y por hacerme ser quien soy.

A mis amigos del colegio, por haber crecido juntos, compartiendo vivencias y experiencias y por enseñarme los verdaderos valores de la amistad.

A mis amigos de la carrera, por acompañarme en este camino, por haber superado todas las dificultades encontradas juntos y por acompañarme día a día desde hace 5 años.

A mis amigos del Erasmus, por llevar la palabra hogar más allá de las fronteras.

Al Grupo de Ingeniería Biomédica, en especial, a mi tutor el Dr. Martínez-Cagigal, por su esfuerzo y dedicación y por enseñarme y aconsejarme todo lo que sé en este campo.

Resumen

La capacidad de poder escribir utilizando única y exclusivamente la ondas cerebrales parece un proyecto utópico fuera del alcance de todos nosotros. Sin embargo, los avances en la neurociencia han desembocado en la creación de interfaces cerebro-ordenador (BCI, *brain-computer interfaces*) las cuales son capaces de traducir las intenciones de los usuarios y de transformarlas en comandos. Sin embargo, estas señales no se pueden decodificar directamente sin utilizar señales de control. Los potenciales evocados visuales modulados por código (c-VEPs, *code-modulated visual evoked potential*) son una innovadora señal de control que permite tiempos de calibración reducidos y una excelente precisión.

Los sistemas BCI basados en c-VEP se fundamentan en la utilización de secuencias binarias pseudoaleatorias para modular los parpadeos de los comandos seleccionables. Es fundamental que la forma en la que se codifiquen estos comandos, es decir, el conjunto de secuencias binarias pseudoaleatorias utilizadas, exhiban buenas propiedades de autocorrelación y/o correlación cruzada. La justificación de codificar los comandos a través de este tipo de secuencias es que, al fijar la atención sobre cada comando, se desencadenen respuestas de electroencefalografía (EEG) que muestren una mayor correlación con el comando al cual el usuario está dirigiendo su atención, en comparación con el resto de estímulos. Uno de los métodos más eficientes para ello es la utilización de versiones desplazadas de una m-secuencia debido a sus excelentes propiedades de autocorrelación. Sin embargo, los Gold Codes han demostrado en varios estudios que, debido a sus magníficas propiedades de correlación cruzada, pueden sustituir a las m-secuencias.

Para garantizar un correcto funcionamiento del sistema BCI es fundamental realizar una etapa de extracción de características para encontrar parámetros correlados con las intenciones del usuario en tiempo real. En este caso, se requiere encontrar características de la señal que nos permitan determinar dónde está mirando el usuario en cada momento. Por tanto, además del filtrado frecuencial, comúnmente se aplican filtros espaciales que ponderan la importancia de cada electrodo. Uno de los métodos más populares para ello es Canonical Correlation Analysis (CCA), cuyo objetivo es maximizar la asociación, medida por la correlación, entre las proyecciones de baja dimensionalidad de dos conjuntos de datos. Sin embargo, Task-Related Component Analysis (TRCA) ha ganado gran popularidad en los últimos años ya que su propósito principal es determinar los coeficientes de las series temporales de manera que se maximice la correlación entre bloques de tarea, asegurando así una alta consistencia entre ellos.

Por todo esto, en este Trabajo de Fin de Grado se desarrolla un sistema BCI basado en c-VEP utilizando desplazamientos circulares de Gold Codes, así como TRCA utilizando Python y Unity. El objetivo es realizar un análisis comparativo con el típico sistema BCI basado en c-VEP que utiliza m-secuencias junto con CCA.

Para ello, se ha realizado un análisis offline a través de dos bases de datos y un análisis online con la aplicación desarrollada, tanto con 31, como con 63 bits. A través de un análisis estadístico, el primero de ellos muestra que no existen diferencias significativas entre el uso de TRCA y CCA cuando se utilizan p -ary m-secuencias, ni en cuanto a las precisiones ni en tiempos de selección, sin embargo, sí que existen diferencias significativas cuando los comandos se modulan a través de Gold Codes. En cuanto al análisis online, los resultados obtenidos en la aplicación desarrollada son excelentes, arrojando una precisión media de $95,31\% \pm 6,13$ con 31 bits y $96,25\% \pm 7,10$ con 63 bits, superior a la de estudios realizados con características muy similares. Además, se ha demostrado que no existen diferencias significativas utilizando 31 y 63 bits en este mismo paradigma lo que permite reducir los tiempos de calibración a la mitad.

Palabras claves: Brain-Computer Interface (BCI), Code-Modulated Visual Evoked Potential (c-VEP), Gold Codes, m-secuencias, CCA, TRCA, Python, Unity.

Abstract

The ability to be able to write solely and exclusively using brain waves seems a utopian project beyond the reach of all of us. However, advances in neuroscience have led to the creation of brain-computer interfaces (BCI), which are capable of translating users intentions and transforming them into commands. However, these signals cannot be decoded directly without the use of control signals. Code-modulated visual evoked potentials (c-VEPs, code-modulated visual evoked potentials) is an innovative control signal that allows reduced calibration times and excellent accuracy.

BCI systems based on c-VEP are founded on the use of pseudorandom binary sequences to modulate the blinks of selectable commands. It is essential that the way in which these commands are encoded, i.e., the set of pseudorandom binary sequences used, exhibit good autocorrelation and/or cross-correlation properties. The rationale for encoding the commands through this type of sequences is that, by fixing attention on each command, electroencephalography (EEG) responses are triggered that show a correlation with the command to which the user is directing his or her attention, compared to the rest of the stimuli. One of the most efficient methods for this is the use of shifted versions of an m-sequence due to their excellent autocorrelation properties. However, Gold Codes have been shown in several studies that, due to their superb cross-correlation properties, they can substitute for m-sequences.

To guarantee a correct operation of the BCI system, it is essential to perform a feature extraction stage to find parameters correlated with the user's intentions in real time. In this case, it is required to find signal characteristics that allow us to determine where the user is looking at any given moment. Therefore, in addition to frequency filtering, spatial filters that weight the importance of each electrode are commonly applied. One of the most popular methods for this is Canonical Correlation Analysis (CCA), which aims to maximize the association, as measured by correlation, between the low-dimensional projections of two data sets. However, Task-Related Component Analysis (TRCA) has gained great popularity in recent years as its main purpose is to determine the coefficients of the time series in such a way as to maximize the correlation between task blocks, thus ensuring high consistency between them.

For all these reasons, in this Final Degree Project a c-VEP based BCI system is developed using Gold Codes and circular shifting, as well as TRCA using Python and Unity.

The objective is to perform a comparative analysis with the typical c-VEP based BCI system using m-sequences along with CCA.

For this purpose, an offline analysis has been performed through two databases and an online analysis with the developed application, both with 31 and 63 bits. Through a statistical analysis, the first one shows that there are no significant differences between the use of TRCA and CCA when p -ary m-sequences are used, neither in terms of accuracies nor in selection times; however, there are significant differences when the commands are modulated through Gold Codes. Regarding the online analysis, the results obtained in the developed application are excellent, yielding an average precision of $95,31\% \pm 6,13$ with 31 bits and $96,25\% \pm 7,10$ with 63 bits, higher than studies carried out with very similar characteristics. In addition, it has been shown that there are no significant differences using 31 and 63 bits in this same paradigm, allowing to reduce calibration times by half.

Key words: Brain-Computer Interface (BCI), Code-Modulated Visual Evoked Potential (c-VEP), Gold Codes, m-sequences, CCA, TRCA, Python, Unity.

Potenciando la extracción de características en sistemas brain-computer interface basados en c-VEP mediante TRCA

Juan Dueñas Ruiz

Índice general

Lista de figuras	IX
Lista de tablas	XIII
Acrónimos	XV
I Memoria del Proyecto	1
1. Descripción del proyecto	3
1.1. Motivación y contexto	3
1.2. Objetivos	5
1.3. Estructura del documento	7
2. Metodología de trabajo	9
2.1. Metodologías ágiles	9
2.1.1. SCRUM	10
2.1.2. SCORE	11
2.1.3. Implementación de SCORE	13
2.2. Herramientas utilizadas	14
3. Planificación	17
3.1. Estimación del esfuerzo	17
3.2. Planificación temporal	20
3.3. Presupuesto económico	21
3.4. Balance final	24
4. Marco Teórico	29
4.1. El sistema nervioso central	29
4.1.1. Aspectos generales de la anatomía cerebral	30
4.1.2. Áreas funcionales del cerebro	31
4.1.3. Adquisición de la actividad cerebral	32
4.2. Sistemas BCI	35
4.2.1. El ciclo cerrado	36

4.2.2.	BCI basados en EEG	37
4.2.3.	Señales de control	39
5.	El paradigma c-VEP	43
5.1.	Bases metodológicas	43
5.2.	Generación de secuencias pseudoaleatorias	44
5.2.1.	LFSR y m-secuencias	45
5.2.2.	Gold Codes	48
5.3.	Extracción de filtros espaciales	50
5.3.1.	Canonical Correlation Analysis	51
5.3.2.	Task-Related Component Analysis	53
5.4.	Procesado de señal	56
5.4.1.	Circular Shifting	57
5.4.2.	Circular Shifting & Gold Codes	59
5.5.	Otros aspectos a tener en cuenta	61
5.5.1.	Ratios de presentación	61
5.5.2.	Raster latencias	62
6.	Estado del arte	63
6.1.	Introducción	63
6.2.	Martínez-Cagigal <i>et al.</i> (2021) -Brain-computer interfaces based on code-modulated visual evoked potentials (c-VEP): a literature review	64
6.3.	Nakanishi <i>et al.</i> (2018) -Enhancing Detection of SSVEPs for a High-Speed Brain Speller Using Task-Related Component Analysis	65
6.4.	Sun <i>et al.</i> (2022) -A 120-target brain-computer interface based on code-modulated visual evoked potentials	68
7.	Diseño y desarrollo del sistema BCI	71
7.1.	Materiales y equipo	71
7.2.	Esquema general del sistema	72
7.3.	Implementación de la aplicación en MEDUSA [©]	73
7.3.1.	Visión general de MEDUSA [©]	73
7.3.2.	Protocolos de comunicación	77
7.3.3.	Aplicaciones dentro de MEDUSA [©]	80
8.	Protocolo de evaluación	83
8.1.	Montaje	83
8.2.	Experimento <i>Circular Shifting</i>	84
8.2.1.	Fase de calibración	85
8.2.2.	Fase de test	86
8.3.	Experimento <i>Gold-Circular</i>	88
8.3.1.	Fase de calibración	92
8.3.2.	Fase de test	93
8.4.	Cuestionario de Satisfacción	93

9. Resultados del experimento	95
9.1. Análisis offline	95
9.1.1. Base de datos de p -ary m-secuencias	95
9.1.2. Base de datos de Thielen <i>et al.</i>	102
9.2. Análisis online	105
9.3. Resultados del cuestionario de satisfacción	116
10. Discusión	119
10.1. Experimento offline	119
10.1.1. Base de datos de p -ary m-secuencias	119
10.1.2. Base de datos de Thielen <i>et al.</i>	120
10.2. Experimento online	120
10.3. Comparación con otros estudios	122
10.4. Fortalezas y limitaciones	124
11. Conclusiones y líneas futuras	127
11.1. Conclusiones	127
11.2. Líneas futuras	129
II Manual de la Aplicación	131
12. Instalación	133
12.1. Manual de instalación	133
12.2. Contenido adjunto	134

Índice de figuras

2.1. Esquema de la metodología SCRUM (Ausum Cloud).	11
3.1. <i>Planning Poker</i> con cartas de tallas de ropa (<i>Planning Poker</i> , Eniun).	18
3.2. Diagrama de Gantt de la planificación temporal del proyecto.	22
3.3. Diagrama de Gantt final de la planificación temporal del proyecto.	27
4.1. Esquema general del sistema nervioso. (A) Representación gráfica del sistema nervioso central y periférico. Imagen adaptada de MedlinePlus, National Library of Medicine. (B) Esquema del sistema nervioso central. Imagen adaptada de Snell (2010).	30
4.2. Esquema general del cerebro humano dividido por lóbulos y áreas (Blink Learning).	32
4.3. Proceso de sinapsis neuronal (University of Maryland, School of Medicine, 2015).	33
4.4. Gráfico que compara las distintas técnicas de adquisición de la actividad cerebral según su resolución espacial y temporal. El grado de invasividad (DI, por sus siglas en inglés) se representa como un gradiente de color, según las capas cerebrales (derecha). ECoG: electrocorticografía, MEG: magnetoencefalografía, EEG: electroencefalografía, fMRI: imagen por resonancia magnética funcional, fNIRS: espectroscopia funcional de infrarrojo cercano, PET: tomografía por emisión de positrones. (Martínez-Cagigal, 2020).	35
4.5. Estructura clásica de un sistema BCI. Imagen adaptada de Millan et al. (2008).	36
4.6. Vistas esquemáticas del estándar del Sistema Internacional 10-10 para EEG (Martínez-Cagigal, 2020).	38
4.7. Oscilaciones neuronales presentes en las señales electroencefalográficas de un sujeto sano y consciente. (A) Segmento de EEG de 5 segundos del canal Pz; (B) densidad espectral de potencia (PSD) de la señal de EEG; (C) desglose del segmento de EEG en las cinco bandas principales: δ (1-4 Hz), θ (4-8 Hz), α (8-13 Hz), β (13-30 Hz) y γ (30-100 Hz) (Martínez-Cagigal, 2020).	40

5.1.	Plantilla principal, obtenida como la promediación de la respuesta tomada por el EEG en el canal Oz, sobre 150 ciclos de calibración (Martínez-Cagigal et al., 2021).	44
5.2.	Esquema general LFSR. Cada $c_j \in \mathbb{Z}_2$, es decir, o bien es 0, o bien es 1; los semicírculos cerrados son compuertas AND; y el bit de retroalimentación s_j es la suma módulo 2 de los contenidos de aquellas etapas i , tal que $0 \leq i \leq n-1$, para las cuales $c_{n-i} = 1$. Imagen adaptada de Menezes et al. (1996).	46
5.3.	Esquema del LFSR definido por $[4, 1 + X + X^4]$ (Menezes et al., 1996).	46
5.4.	Resumen de la codificación de 16 comandos a través del uso de la m-secuencia de 63 bits obtenida del LFSR dado por $[6, 1 + X^5 + X^6]$ y el estado inicial $[1, 1, 1, 1, 1, 0]$, utilizando un desfase de $\tau = 4$. En la parte de la izquierda y en la inferior se observan los valores de la correlación de cada uno de los comandos con la secuencia sin desplazar. En la parte de la derecha, se ve gráficamente la codificación de cada comando, cómo se desplaza un numero entero de τ con respecto a la m-secuencia original, que codifica el carácter A.	48
5.5.	Esquema de generador de Gold Codes (Šajić et al., 2013).	49
5.6.	Diagrama de TRCA en su uso para EEG (Nakanishi et al., 2018).	54
5.7.	La pipeline de procesamiento de referencia para BCI basados en c-VEP con <i>Circular Shifting</i> de m-secuencias y CCA. La etapa de calibración tiene como objetivo calcular las plantillas para cada comando, de acuerdo con sus desplazamientos. En la fase online (o de test), se proyectan las épocas y determina el comando seleccionado mediante un análisis de correlación. Las variables N_s , N_c , k_c y k_t se refieren al número de muestras, canales, ciclos de calibración (o épocas de calibración) y ciclos de test (o épocas de test), respectivamente; mientras que τ indica el tamaño del desplazamiento (Martínez-Cagigal et al., 2021).	57
5.8.	La pipeline de procesamiento por referencia de BCI basados en c-VEP con <i>Circular Shifting</i> de m-secuencias y TRCA. Utiliza la misma notación que la figura 5.7. Imagen adaptada de Martínez-Cagigal et al. (2021).	59
5.9.	La pipeline de procesamiento por referencia de BCI basados en c-VEP con <i>Circular Shifting</i> Gold Codes y TRCA. Utiliza la misma notación que la figura 5.7. Se observa que, en este caso, se repite el proceso una vez por cada Gold Code. Imagen adaptada de Martínez-Cagigal et al. (2021).	60
6.1.	Diseño de los estímulos del sistema BCI de 40 objetivos. Se especifican los valores de frecuencia y fase para cada objetivo (Nakanishi et al., 2018).	66
6.2.	Matriz de comandos utilizada en la fase experimental offline en Sun et al. (2022).	69
6.3.	Matriz de comandos utilizada en la fase experimental online en Sun et al. (2022).	69

7.1. Equipo de g.Tec utilizado. En la parte superior a la derecha, se aprecia el amplificador g.USBamp. En la parte superior a la izquierda, el gorro g.GAMMAcap3 con los electrodos g.LADYbird ya colocados. Finalmente, en la parte inferior, la g.GAMMAbox. Imagen adaptada de Chenane et al. (2019).	72
7.2. Esquema general de la estructura del sistema BCI.	73
7.3. Esquema general MEDUSA [©] (Santamaría-Vázquez et al., 2023).	74
7.4. Ventana principal de MEDUSA [©] Platform. La vista está dividida en paneles para controlar las diferentes funcionalidades. Estos paneles son: aplicaciones (arriba a la izquierda), log de mensajes informativos (abajo a la izquierda) y gráficos en tiempo real (a la derecha). Además, hay más controles y configuraciones disponibles en las barras de tareas.	76
7.5. Panel de configuración de flujos LSL en MEDUSA [©] Platform. En concreto, aparecen dos posibles flujos LSL, que simulan ser dos EEG, llamados <i>Signal_generator</i> y <i>Signal_generator_2</i> , de los que sólo se ha conectado el segundo.	78
7.6. Esquema comunicación TCP/IP (Tyagi, 2020).	79
7.7. Ciclo de vida y workflow de una aplicación en MEDUSA [©] Platform. ** Este estado se alcanza automáticamente cada vez que una ejecución finaliza. Si una ejecución no tiene un final definido, sino que es el usuario quien debe detener la aplicación, este estado no se alcanzará ((alias?)).	81
7.8. Uno de los menús de configuración de la aplicación c-VEP Speller de MEDUSA [©]	82
8.1. Uno de los sujetos antes de realizar el experimento online.	84
8.2. Codificación de matriz de comandos con <i>Circular Shifting</i> de 63 bits. En el eje vertical aparece tanto el comando como el desplazamiento con respecto a la secuencia original (representada por 0τ), donde $\tau = 4$	85
8.3. Interfaz gráfica de la fase de calibración de la aplicación utilizada en el experimento <i>Circular Shifting</i> . La subfigura A muestra el comando iluminado de magenta para fijar la atención del usuario. La subfigura B muestra la iluminación del comando en un frame.	86
8.4. Esquema de la fase de calibración de la aplicación utilizada en el experimento <i>Circular Shifting</i> . t_c representa el tiempo de 10 ciclos, que en este caso es 5,25 s.	86
8.5. Interfaz gráfica de la fase de test de las aplicaciones utilizadas en los experimentos <i>Circular Shifting</i> y <i>Gold-Circular</i> . La subfigura A muestra la matriz de comandos en un frame de ejecución. La subfigura B muestra la submatriz de comandos en la que se ha seleccionado la letra P.	87
8.6. Esquema de la fase de test de los experimentos <i>Circular Shifting</i> y <i>Gold-Circular</i> en la que t_c representa el tiempo de duración de 10 ciclos. Se observa que en este caso se ha seleccionado la letra P.	87

8.7.	Codificación de los comandos en el experimento <i>Gold-Circular</i> 31 bits. Cada una de las subfiguras individuales 1,2,3 y 4 representa la codificación del Gold Code, marcado por 0τ , así como sus versiones desplazadas por $\tau = 7$, junto con la letra que va a codificar. El número de estas subfiguras se corresponde con el número de Gold Code de la tabla 8.2. La subfigura A representa la matriz de comandos divida por zonas según el Gold Code a partir del cual están codificados. La zona roja se corresponde con la subfigura 1 , la amarilla con con la subfigura 2 , la verde con la subfigura 3 y la azul con la subfigura 4 . La subfigura B representa la codificación global de la matriz de comandos.	90
8.8.	Codificación de los comandos en el experimento <i>Gold-Circular</i> 63 bits. Cada una de las subfiguras individuales 1,2,3 y 4 representa la codificación del Gold Code, marcado por 0τ , así como sus versiones desplazadas por $\tau = 15$, junto con la letra que va a codificar. El número de estas subfiguras se corresponde con el número de Gold Code de la tabla 8.3. La subfigura A representa la matriz de comandos divida por zonas según el Gold Code a partir del cual están codificados. La zona roja se corresponde con la subfigura 1 , la amarilla con con la subfigura 2 , la verde con la subfigura 3 y la azul con la subfigura 4 . La subfigura B representa la codificación global de la matriz de comandos.	91
8.9.	Interfaz gráfica de la fase de calibración de la aplicación desarrollada para los experimentos <i>Gold-Circular</i> con 31 y 63 bits. La subfigura A muestra el comando iluminado de magenta para fijar la atención del usuario. La subfigura B muestra la iluminación del comando en un frame.	92
8.10.	Esquema de la fase de calibración de la aplicación desarrollada para los experimentos <i>Gold-Circular</i> con 31 y 63 bits. t_c representa el tiempo de 10 ciclos. Si la longitud de la secuencia es de 63 bits será de 5,25 s pero si es de 31 bits será de 2,58 s. Nótese que en la figura las secuencias parar cada letra son distintas y de 63 bits. El caso con 31 bits es completamente análogo.	92
9.1.	Instantáneas de la aplicación utilizada en el artículo Martínez-Cagigal et al. (2023). (A) Etapa de calibración, en la que se pide a los usuarios que presten atención al comando central, que parpadea de acuerdo con p -ary m-secuencia original. (B) Disposición en línea, en la que los usuarios prestan atención al comando objetivo para realizar tareas de escritura. (C) Durante el modo en línea, las etiquetas de los comandos se reemplazan por un pequeño punto centrado que ayuda a los usuarios a enfocarse en el comando deseado. En esta instantánea, se muestra para cuando $p = 2$. (D) Ejemplo de instantánea de una estimulación cuando $p = 11$. (E) Resumen de la codificación de los comandos utilizada.	96

9.2.	Estímulos y protocolo experimental de la fase offline del artículo de Thielen et al. (2021). La interfaz gráfica muestra una calculadora con 20 comandos seleccionables. Los objetivos se indicaron con un color verde. Se dejó un segundo de descanso entre trials. La duración de cada trials fue de 31,5 s.	103
9.3.	Gráfica de la topografía en un modelo de cabeza 2D, donde se representa el valor del filtro espacial obtenido con CCA en cada uno de los canales utilizados para el experimento <i>Circular Shifting</i>	108
9.4.	Respuesta cerebral de los usuarios en el experimento <i>Circular Shifting</i> . (A) Plantilla principal para cada uno de los usuarios tomada como promedio de la respuesta del EEG en el canal Oz sobre 200 ciclos de calibración. (B) Intervalo de confianza al 95 % para los ciclos de calibración de todos los usuarios, así como la media obtenida.	108
9.6.	Gráfica de la topografía en un modelo de cabeza 2D, donde se representa el valor del filtro espacial obtenido con TRCA en cada uno de los canales utilizados para el experimento <i>Gold-Circular</i> con 31 bits. La figura muestra la topografía de los usuarios por filas y de los Gold Codes por columnas, donde el número de Gold Code se corresponde con los de la tabla 8.2.	110
9.7.	Respuesta cerebral de los usuarios en el experimento <i>Gold-Circular</i> con 31 bits. (A) Plantilla principal para cada uno de los usuarios tomada como promedio de la respuesta del EEG en el canal Oz sobre 200 ciclos de calibración. En azul se muestra la plantilla principal para el primer Gold Code, en naranja para el segundo, en verde para el tercero y en morado para el cuarto. (B) Intervalo de confianza al 95 % para los ciclos de calibración de todos los usuarios, así como la media obtenida, para cada uno de los Gold Codes. El número de Gold Code se corresponde con los de la tabla 8.2.	112
9.9.	Gráfica de la topografía en un modelo de cabeza 2D, donde se representa el valor del filtro espacial obtenido con TRCA en cada uno de los canales utilizados para el experimento <i>Gold-Circular</i> con 63 bits. La figura muestra la topografía de los usuarios por filas y de los Gold Codes por columnas, donde el número de Gold Code se corresponde con los de la tabla 8.3.	114
9.10.	Respuesta cerebral de los usuarios en el experimento <i>Gold-Circular</i> con 63 bits. (A) Plantilla principal para cada uno de los usuarios tomada como promedio de la respuesta del EEG en el canal Oz sobre 200 ciclos de calibración. En azul se muestra la plantilla principal para el primer Gold Code, en naranja para el segundo, en verde para el tercero y en morado para el cuarto. (B) Intervalo de confianza al 95 % para los ciclos de calibración de todos los usuarios, así como la media obtenida, para cada uno de los Gold Codes. El número de Gold Code se corresponde con los de la tabla 8.3.	116

Índice de tablas

3.1. Distribución temporal en bloques de las tareas a realizar.	21
3.2. Costes del proyecto derivados del hardware.	23
3.3. Costes del proyecto derivados de componentes BCI.	23
3.4. Costes del proyecto derivados de los servicios.	24
3.5. Costes del proyecto derivados de las licencias.	24
3.6. Costes del proyecto derivados del personal.	25
3.7. Costes totales del proyecto.	25
3.8. Distribución temporal por bloques final de las tareas realizadas.	26
3.9. Costes del proyecto final derivados de los servicios.	26
3.10. Costes finales del proyecto derivados del personal.	26
3.11. Costes totales finales del proyecto.	28
5.1. Tabla de valores del LFSR dado por $[4, 1 + X + X^4]$	47
5.2. Parejas de polinomios preferidos para generar secuencias de Gold Codes. El grado del polinomio está representado por n y la longitud de las secuencias por N . Tabla adaptada de Sharma et al. (2014).	50
6.1. Secuencias de bits empleadas en los estudios revisados por Martínez-Cagigal et al. (2021). # representa el número de estudios.	65
6.2. Resultados del experimento online de Nakanishi et al. (2018).	67
6.3. m-secuencias de 31 bits, representadas por M_i , con $i = 1, 2$ y los Gold Codes seleccionados, representados por G_j , con $j = 1, \dots, 4$ del texto de Sun et al. (2022).	68
8.1. m-secuencia elegida para la codificación de los comandos para el experimento de <i>Circular Shifting</i>	85
8.2. Gold Codes seleccionados para codificar la matriz de comandos en el experimento online con 31 bits.	88
8.3. Gold Codes seleccionados para codificar la matriz de comandos en el experimento online con 63 bits.	89

9.1. Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de Martínez-Cagigal et al. (2023) cuando la base utilizada para las p -ary m-secuencias es 2.	97
9.2. Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de Martínez-Cagigal et al. (2023) cuando la base utilizada para las p -ary m-secuencias es 3.	98
9.3. Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de Martínez-Cagigal et al. (2023) cuando la base utilizada para las p -ary m-secuencias es 5.	99
9.4. Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de Martínez-Cagigal et al. (2023) cuando la base utilizada para las p -ary m-secuencias es 7.	100
9.5. Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de Martínez-Cagigal et al. (2023) cuando la base utilizada para las p -ary m-secuencias es 11.	101
9.6. Tabla de medias por usuarios de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA para cada ciclo de la base de datos de Martínez-Cagigal et al. (2023).	102
9.7. Precisiones medias obtenidas por usuarios y ciclos tras aplicar CCA y TRCA a la base de datos de Thielen et al. (2021), utilizando <i>5-fold cross-validation</i> sobre el bloque utilizado para la fase de test.	103
9.8. Precisiones obtenidas por usuarios y ciclos tras aplicar CCA y TRCA a la base de datos de Thielen et al. (2021), utilizando <i>5-fold cross-validation</i> sobre el bloque utilizado para la fase de test.	104
9.9. Precisiones obtenidas del análisis online mostrada por ciclos y por usuarios del experimento <i>Circular Shifting</i> con 63 bits y CCA, así como la media y desviación típica de cada ciclo por usuarios.	106
9.10. Precisiones obtenidas del análisis online mostrada por ciclos y por usuarios del experimento <i>Gold-Circular</i> con 31 bits y TRCA, así como la media y desviación típica de cada ciclo por usuarios.	106
9.11. Precisiones obtenidas del análisis online mostrada por ciclos y por usuarios del experimento <i>Gold-Circular</i> con 63 bits y TRCA, así como la media y desviación típica de cada ciclo por usuarios.	107
9.12. Tabla de medias por usuarios de las precisiones obtenidas los experimentos online para cada ciclo. También se ha incluido el p -valor comparando todos los experimentos, donde GC hace referencia a <i>Gold-Circular</i>	107
9.13. Respuestas al cuestionario del análisis online por parte de los 10 sujetos participantes.	117

10.1. Resumen de los estudios con sistemas BCI basados en c-VEP con características similares al desarrollado. La columna “Nº códigos” hace referencia al número de códigos, por lo que, si aparece 1, se han utilizado desplazamientos circulares de m-secuencias. Si aparece un número mayor que 1, quiere decir que se ha utilizado algún otro tipo de secuencia pseudoaleatoria junto con el paradigma <i>Circular Shifting</i>	123
---	-----

Acrónimos

APA	Almost Perfect Autocorrelation
APM	Agile Project Management
BCI	Brain-Computer Interface
BOLD	Blood Oxygenation Level Dependent
CCA	Canonical Correlation Analysis
CSV	Comma Separated Values
c-VEP	Code-modulated Visual Evoked Potential
ECoG	Electrocorticografía
EEG	Electroencefalografía
ELA	Esclerosis Lateral Amiotrófica
ERP	Event-Related Potential
fMRI	Functional Magnetic Resonance Imaging
fNIRS	Functional Near-Infrared Spectroscopy
GIB	Grupo de Ingeniería Biomédica
GUI	Graphical User Interface
IDE	Integrated Development Environment
ITR	Information Transfer Rate
LFSR	Linear Feedback Shift Register
LSL	Lab Streaming Layer
ME	Médula Espinal
MLS	Maximum Length Sequence
MEG	Magnetoencefalografía
NF	Neurofeedback
PET	Positron Emission Tomography
PN	Pseudo-random Noise
PT	Paquete de Trabajo
SMR	Sensorimotor Rhythms
SNC	Sistema Nervioso Central
SNP	Sistema Nervioso Periférico
SSVEP	Steady State Visually Evoked Potential
TRCA	Task-Related Component Analysis

Parte I

Memoria del Proyecto

Capítulo 1

Descripción del proyecto

Este primer capítulo pretende ilustrar al lector con una breve introducción acerca de los sistemas *brain-computer interface*, poner en contexto su importancia en la actualidad y justificar la elección de este tema, así como los objetivos perseguidos por este Trabajo de Fin de Grado.

1.1. Motivación y contexto

La idea de poder comunicarse o de controlar dispositivos utilizando única y exclusivamente las ondas cerebrales parece un proyecto utópico. Sin embargo, hace alrededor de 20 años surgieron los sistemas interfaz cerebro-ordenador, más conocidos por su traducción al inglés, *brain-computer interface* (BCI), convirtiéndose en la actualidad en uno de los campos de estudio más ambiciosos dentro de la Ingeniería Neuronal.

La emersión de esta fascinante disciplina científica, según [Wolpaw and Wolpaw \(2012a\)](#), se basó en tres factores. En primer lugar, la existencia de dispositivos hardware y software con una potente capacidad de análisis de la actividad cerebral y con un coste económico asequible. El segundo factor, se centra en el basto conocimiento del sistema nervioso central y los grandes avances en la medicina, a través de investigaciones en humanos y en animales. Finalmente, el último factor es el reconocimiento y visibilidad de los problemas cotidianos y necesidades de personas con enfermedades como parálisis cerebral, lesión de la médula espinal, accidente cerebrovascular, esclerosis lateral amiotrófica (ELA), esclerosis múltiple o distrofias musculares, entre otros. Los sistemas BCI son apropiados para estas personas con graves discapacidades motoras para sustituir sus respuestas del sistema nervioso central como aplicaciones de comunicación aumentativa o alternativa. Sin embargo, no sólo existen aplicaciones únicamente en el dominio médico, ya que existe un alto potencial para las aplicaciones de BCI que monitorean la carga cognitiva, la atención, la somnolencia y otros aspectos de la mente ([Värbu et al., 2022](#)). En concreto, se han utilizado para pruebas de coeficiente intelectual, para diseñar y llevar a cabo mejor exámenes psicométricos, entrenamiento militar y otros tipos de formación ([Friedman et al., 2019](#)). Finalmente, los sistemas BCI también se pueden utilizar en un ámbito lúdico, por

ejemplo, en el desarrollo de videojuegos, ya sea de forma clásica, como por ejemplo en el representativo *online target shooting game* (Martišius and Damaševičius, 2016) o el *gaming* más innovador utilizando realidad virtual (Hadjaros et al., 2023).

Los sistemas BCI representan una convergencia entre la mente humana y la tecnología informática. En líneas generales, estos sistemas permiten, utilizando únicamente las señales cerebrales del usuario normalmente recogidas a través de electroencefalografía (EEG), el control de dispositivos o aplicaciones (Värbu et al., 2022). Un sensor de EEG es un dispositivo electrónico que puede medir señales eléctricas del cerebro. Los sensores EEG típicamente registran las señales eléctricas variables creadas por la actividad de grandes grupos de neuronas cerca de la superficie del cerebro (Soufneyestani et al., 2020). Es importante destacar que, a través del EEG, resulta imposible decodificar directamente los pensamientos de un individuo. Esto se debe a que la señal obtenida por el EEG presenta ciertas limitaciones que la hacen susceptible a contaminación por artefactos no neurológicos, como el ruido muscular y electromagnético. Estas limitaciones incluyen la baja amplitud de la señal, del orden de microvoltios (μV), efectos de atenuación, conducción de volumen y una resolución espacial limitada (Wolpaw and Wolpaw, 2012a). Por consiguiente, resulta crucial buscar correlaciones entre las intenciones del usuario y cambios medibles en el EEG. Para inducir estos cambios medibles en la actividad cerebral, existen dos enfoques principales: endógeno y exógeno. El enfoque endógeno implica que los usuarios aprendan a autorregular su actividad cerebral, lo cual no solo puede ser útil para generar neuroplasticidad, sino también aplicable en contextos de neurorehabilitación (Wolpaw and Wolpaw, 2012a). Por otro lado, el enfoque exógeno implica el uso de estímulos externos para provocar respuestas cerebrales, siendo comúnmente empleado en sistemas BCI para la comunicación y el control (Wolpaw and Wolpaw, 2012a). Por ello, los sistemas BCI basados en EEG dependen del procesamiento de cambios medibles relacionados con tareas cognitivas, conocidos como *señales de control* (Martínez-Cagigal, 2020).

De esta manera, surgieron los potenciales evocados visuales modulados por código (c-VEPs), una innovadora y reciente señal de control capaz de aumentar la precisión, exactitud, y velocidad de respuesta en tiempo real de los sistemas BCI utilizando además tiempos de calibración muy reducidos (Martínez-Cagigal et al., 2021). Sin embargo, a pesar de su creciente popularidad, todavía no se ha llevado a cabo un análisis exhaustivo de los diferentes métodos de procesamiento de señales y las diversas estrategias de codificación de comandos asociadas a este paradigma.

Por consiguiente, este documento ofrece una investigación que analiza cómo influye en el rendimiento de un sistema BCI basado en c-VEP: (1) la elección de la codificación de los comandos, ya sea mediante *maximum length sequence* (MLS) generadas a partir de *linear feedback shift register* (LFSR) o mediante Gold Codes y (2) la aplicación de diversos filtros espaciales, específicamente aquellos derivados de métodos de análisis multivariante, tales como *Canonical Correlation Analysis* (CCA) y *Task-related Component Analysis* (TRCA).

Este Trabajo de Fin de Grado se ha realizado en el Grupo de Ingeniería Biomédica (GIB, Universidad de Valladolid) que representa un equipo de investigación multidisciplinar dedicado a la exploración científica. Afiliado al Centro de Investigación Biomédica en Red de Bioingeniería, Biomateriales y Nanomedicina (CIBER-BBN), el GIB se posiciona como una Unidad de Investigación Consolidada (UIC-060) respaldada por la Junta de Castilla y León, además de ser un Grupo de Investigación Reconocido (GIR) de la Universidad de Valladolid. Compuesto primordialmente por profesionales de la ingeniería y la medicina, con diversas especializaciones que abarcan neumología, neurología, neurofisiología, psiquiatría y oftalmología, este equipo colabora estrechamente en diversas áreas relacionadas con la ingeniería biomédica. La convergencia de estos campos de trabajo ha llevado a dirigir los esfuerzos hacia diferentes líneas de investigación entre las que se encuentran:

- **Análisis de señales neuronales** a través del estudio de registros de EEG y magnetoencefalografía (MEG) para la detección de patologías neurológicas como el Alzheimer o la esquizofrenia. Esto se consigue a través de diversas técnicas de procesamiento avanzado de señal: métodos espectrales y no lineales, medidas de acoplamiento y parámetros derivados de la teoría de redes complejas.
- **Análisis de señales polisomnográficas** a través de modelos predictivos basados en *machine learning* para la detección de enfermedades como el Síndrome de la Apnea-Hipopnea del Sueño (SAHS) o la Enfermedad Pulmonar Obstructiva Crónica (EPOC).
- **Procesado de retinografías**, a través del uso de inteligencia artificial en el procesamiento de imágenes y señales para la detección de enfermedades relacionadas con la vista, como pueden ser la retinopatía diabética o la degeneración macular asociada a la edad.
- **Diseño, desarrollo y evaluación de sistemas BCI** no invasivos con el objetivo de mejorar la calidad de vida de las personas. Se busca crear aplicaciones que abarquen desde la asistencia a personas con discapacidades motoras graves, hasta la integración de un grado de libertad adicional en videojuegos. También se enfoca en el apoyo a la neurorrehabilitación de individuos que han sufrido accidentes cerebrovasculares y en el entrenamiento cognitivo de personas mayores. Además, desarrollan software específico para la programación de sistemas BCI.

1.2. Objetivos

El objetivo de este Trabajo de Fin de Grado es implementar y evaluar la viabilidad del innovador algoritmo TRCA como alternativa al CCA, así como analizar el rendimiento de distintas codificaciones de comandos mediante MLS y Gold Codes. Para alcanzar este objetivo general, se han planteado los siguientes objetivos específicos:

Obj-1. Diseñar el sistema BCI: esta etapa comprenderá la revisión del estado de la técnica en sistemas BCI, así como el planteamiento y diseño de los paradigmas a desarrollar, incluyendo *Circular Shifting* con MLS y Gold Codes.

Obj-2. Implementar la aplicación: durante esta fase se llevará a cabo el desarrollo de la aplicación de estimulación o paradigma mediante Unity (C#), y de los Gold Codes en Python, así como la implementación del método de procesamiento de señal TRCA, también en Python, a través de MEDUSA[©] (<https://medusabci.com/>), una plataforma software para desarrollar sistemas BCI no invasivos y experimentos neurocientíficos.

Obj-3. Evaluar el sistema: se llevará a cabo un análisis offline y online para evaluar el sistema BCI utilizando un grupo de 10 personas sanas como sujetos de prueba. Se analizarán los resultados obtenidos, se realizará una discusión comparativa con otros estudios relacionados y se extraerán conclusiones sobre la viabilidad y efectividad de los Gold Codes y TRCA en los sistemas BCI actuales.

El producto final de esta aplicación está dirigida a aquellas personas que padezcan alguna patología neurológica que les impida comunicarse a través de palabras o gestos, ya que el sistema será capaz de supervisar y monitorizar las ondas cerebrales que la persona produzca únicamente mirando al comando (en este caso letra) que quiera escribir, sin necesidad de realizar ningún otro movimiento más que el ocular, todo ello a través de un EEG. Además, este sistema también se podrá utilizar en el ámbito del ocio, brindando una forma innovadora y accesible de interactuar con dispositivos electrónicos y aplicaciones de entretenimiento mediante el control mental.

La motivación personal a la hora de la elección de este Trabajo de Fin de Grado ha sido principalmente el interés y la atracción que surge de la intersección del ámbito sanitario, en concreto, de la mente humana y los sistemas computacionales. Este tipo de tecnología representa un campo emergente que ofrece un potencial revolucionario en la manera en que interactuamos con las máquinas y el mundo digital que nos rodea. La posibilidad de traducir la actividad cerebral en comandos ejecutables abre un amplio abanico de aplicaciones fascinantes que varían desde su uso en actividades lúdicas, actividades para mejorar la productividad, o como en este caso, actividades biosanitarias que pueden llegar a suponer un antes y un después en la historia de la humanidad.

Tras la finalización de este documento, se habrán obtenido tanto las [competencias generales del Grado en Ingeniería Informática de Servicios y Aplicaciones](#) como las específicas, incluyendo aquellas comunes a la rama de Informática, como las de Tecnologías de la Información y las de Sistemas de la Información.

1.3. Estructura del documento

La memoria de este Trabajo de Fin de Grado está compuesta por un total de once capítulos, que son: descripción del proyecto, metodología de trabajo, planificación, marco teórico, el paradigma c-VEP, estado de la técnica, diseño y desarrollo del sistema BCI, protocolo de evaluación, resultados del experimento, discusión y conclusiones y líneas futuras.

En el primer capítulo se realiza una breve introducción del tema a tratar, incluyendo la motivación y el contexto, los objetivos que persigue este Trabajo de Fin de Grado y la estructura del documento.

El segundo capítulo trata de exponer la metodología de trabajo seguida desde el inicio del proyecto hasta su finalización. En concreto, se introducen brevemente las metodologías ágiles, profundizando más en SCRUM y SCORE. Finalmente, se detallan las herramientas utilizadas.

La planificación del proyecto se expone en el tercer capítulo, donde se expone una estimación del esfuerzo, una planificación temporal y económica del proyecto, así como un balance final tras la finalización del mismo.

En el cuarto capítulo se trata el marco teórico. En concreto, se plantea una breve exposición sobre el sistema nervioso central, centrado en el cerebro y su especialización funcional. Además, se hablará sobre la adquisición de la actividad cerebral, sobre sistemas BCI en general y, más concretamente, aquellos basados en EEG. Este capítulo termina con una descripción acerca de las señales de control más conocidas.

El quinto capítulo contiene una extensa descripción sobre el paradigma c-VEP. En concreto, se exploran sus conceptos fundamentales, como son la generación de números pseudoaleatorios, la extracción de filtros espaciales o las tuberías de procesado más conocidas.

En el capítulo sexto se realiza una revisión del estado del arte con el objetivo de conocer los conceptos y técnicas más recientes en este campo. En concreto, se han revisado tres artículos que ilustran los aspectos generales tratados en este Trabajo de Fin de Grado.

En el capítulo número siete se esclarece el diseño y desarrollo del sistema BCI, donde se detalla el equipo de EEG utilizado, un pequeño esquema del sistema y todos los detalles acerca de MEDUSA[©].

El octavo capítulo contiene el protocolo de evaluación, donde se tratan temas como el montaje del sistema y los experimentos realizados en el análisis online, incluyendo la fase de calibración y la de test.

A continuación, en el capítulo nueve se presentan los resultados de los experimentos, tanto del análisis offline como del online, así como los del cuestionario de satisfacción.

En el capítulo diez se discuten los resultados obtenidos. En concreto, se analiza la viabilidad de la aplicación desarrollada, la comparación con otros estudios de índole similar y las limitaciones encontradas.

Finalmente, en el undécimo capítulo se concluye y se exponen las líneas futuras de investigación.

Capítulo 2

Metodología de trabajo

Este capítulo pretende ilustrar al lector con la metodología de trabajo seguida desde el inicio del proyecto hasta su finalización. Como es de costumbre en el desarrollo de software, se han seguido marcos de trabajo de Agile, una popular metodología ágil, como pueden ser SCRUM o SCORE. Finalmente, se resumen las herramientas utilizadas, tanto para la documentación, como para el desarrollo de la aplicación.

2.1. Metodologías ágiles

La gestión ágil de proyectos (APM, por sus siglas en inglés) o metodologías ágiles, según [Hidalgo \(2019\)](#), representa un enfoque de gestión de equipos y un *framework* de productividad basado en el progreso continuo e incremental en el trabajo, incluso frente a potenciales cambios. APM tiene sus orígenes en los procesos ágiles de desarrollo de software, tales como SCRUM, XP, DSDM, Cristal, entre otros, los cuales son metodologías de programación basadas en la adaptabilidad a cualquier cambio, buscando aumentar las posibilidades de éxito de un proyecto. La mayoría de los métodos ágiles intentan minimizar los riesgos durante la ejecución de un proyecto mediante el desarrollo de software en iteraciones, que generalmente duran de una a cuatro semanas. Cada iteración trata de ser un proyecto en miniatura del proyecto final e incluye todas las tareas necesarias para implementar nuevas funcionalidades: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Un proyecto de programación ágil tiene como objetivo lanzar nuevo software al final de cada iteración, y entre cada iteración el equipo reevalúa sus prioridades de acuerdo con las necesidades del cliente.

A continuación, se presentan dos metodologías ágiles, como son SCRUM y SCORE. La segunda es una adaptación de la primera a proyectos de investigación universitarios. Por este motivo, y por el hecho de realizar este Trabajo de Fin de Grado en el seno del Grupo de Ingeniería Biomédica, un grupo de investigación reconocido de la Universidad de Valladolid, como se mencionó en sección [1.1](#), se ha escogido la metodología SCORE para la ejecución de este proyecto. Sin embargo, es necesario recordar los conceptos fundamentales de la metodología SCRUM antes de profundizar en SCORE.

2.1.1. SCRUM

SCRUM es un *framework* que facilita la actividad coordinada de los programadores, quienes dividen su trabajo en pequeñas tareas que deben completarse en periodos de duración fija o *sprints*, analizando el progreso y programando reuniones regulares para desarrollar productos de forma incremental (Hidalgo, 2019). SCRUM se enfoca en el trabajo en equipo para mejorar la comunicación y fortalecer la cooperación entre los miembros del equipo, lo que se traduce en una mayor productividad en el futuro. Además, SCRUM permite abordar problemas a corto plazo, reducir los riesgos del proyecto, fomentar una mayor participación del cliente a lo largo de los procesos y ofrecer productos o servicios más funcionales con una frecuencia mayor (Rachid and Stopa, 2019). Esta metodología se puede aplicar en distintos ambientes, independientemente de la naturaleza de la empresa o grupo que lo implementa, ya que Sassa et al. (2023) muestra varios estudios en los que se aplica, por ejemplo, en la industria naval o en una *startup* de tecnología, evidenciando así la flexibilidad de SCRUM.

Equipo SCRUM

Schwaber and Sutherland (2011) explica que en esta metodología, los equipos están formados por menos de 10 personas, en las que al menos una de ellas debe ser el *Scrum Master*, otro el *Product Owner* y el resto desarrolladores. Estos equipos se caracterizan por no tener jerarquías ni dividirse en subequipos y trabajar de forma coordinada hacia el objetivo, o *Product Goal*. El *Scrum Master* es el responsable de establecer la metodología SCRUM como está originalmente definida y velar por su correcto funcionamiento, tanto dentro del equipo SCRUM como en la empresa y/u organización, así como la consecución de los objetivos. Los desarrolladores son los encargados de proporcionar un valor incremental al producto en cada uno de los *sprint*. Finalmente, el *Product Owner* es el punto de unión entre el resto de *stakeholders* y los desarrolladores, además de ser el responsable de la gestión del *Product Backlog*, un documento que recoge una lista ordenada de los aspectos necesarios para mejorar el producto final.

Sprints

Un nuevo *sprint* se inicia nada más terminar el anterior y se realizan tareas como *sprint planning*, *Daily Scrums*, *sprint review* y *sprint retrospective*. Es importante tener en cuenta que durante los *sprints* no se deben realizar cambios que no converjan hacia el *Product Goal*. Además, no se puede disminuir la calidad de lo desarrollado hasta el momento, sin embargo, sí se puede redefinir el *Product Backlog*. Como se muestra en la figura 2.1, este proceso comienza con el *sprint planning*, una reunión que trata de esclarecer el trabajo que se va a realizar. También, el *Product Owner* asegura que todos los participantes están preparados para discutir sobre los aspectos más importantes del

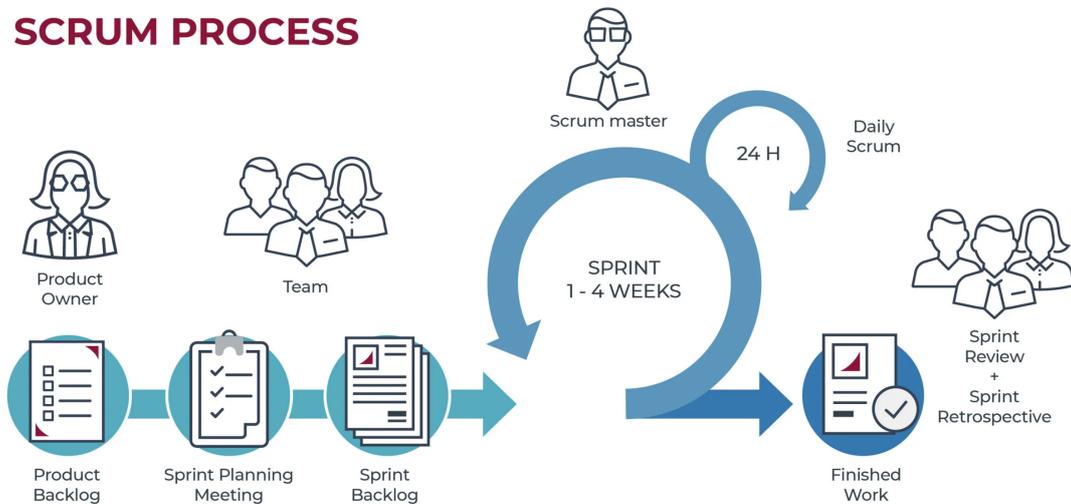


Figura 2.1: Esquema de la metodología SCRUM (Ausum Cloud).

Product Backlog y cómo estos se encaminan hacia el *Product Goal*. Así, en esta reunión, se responden a las siguientes preguntas:

1. ¿Por qué este *sprint* es valioso?
2. ¿Qué se puede hacer en este *sprint*?
3. ¿Cómo se hará el trabajo escogido?

La respuesta a estas preguntas desembocan en el *sprint Backlog*. A continuación, comienza el desarrollo del producto, con una duración entre 1 y 4 semanas, en la que se realizan *Daily Scrums*. Estas son reuniones diarias informales de los desarrolladores, de una duración de aproximadamente 15 minutos, cuyo objetivo es inspeccionar el progreso hasta el momento, adaptando el *sprint Backlog* si fuese necesario. Está recomendado mantener estas reuniones a la misma hora y en el mismo lugar todos los días.

Los *sprints* finalizan con una versión funcional del producto y con dos reuniones, conocidas como *sprint review* y *sprint retrospective*. El objetivo de la primera es analizar el resultado del *sprint* y de determinar futuros cambios. El equipo SCRUM presenta los resultados de su trabajo a los *Stakeholders* principales y su progreso de cara al *Product Goal*. Sin embargo, el objetivo del *sprint retrospective* es planificar formas de mejorar la calidad del producto y efectividad del equipo de trabajo.

2.1.2. SCORE

La metodología SCORE, desarrollada por Hicks and Foster (2010), nace en el seno de un grupo de investigación ante la necesidad de gestionar las reuniones que derivan de la

tarea de tutorizar un doctorado cuando el número de estudiantes es alto. Antes de que desarrollaran esta metodología, estos autores realizaban una o dos reuniones por semana con cada uno de sus alumnos con una duración de entre treinta minutos y una hora. La naturaleza de estas reuniones variaba entre resolución de dudas, comunicación de los avances del proyecto o incluso *brainstorming*. Sin embargo, a medida que aumentaron los alumnos, esta aproximación era inviable y [Hicks and Foster \(2010\)](#) decidieron desarrollar una metodología para gestionar el trabajo y las reuniones derivadas de un proyecto de doctorado o incluso cualquier otro proyecto de investigación.

De esta manera, SCORE adapta muchos de los conceptos de SCRUM, como pueden ser las reuniones cortas, para disminuir la fragmentación del horario. Así, las reuniones de investigación más extensas podían programarse según fuera necesario tras un *Daily Scrum*, en lugar de asignar franjas fijas para cada estudiante de antemano. Además, se consideraba que estas reuniones podían potenciar la comunicación dentro del grupo, puesto que cada miembro del equipo estaría al tanto del progreso de los demás. Por consiguiente, la metodología SCORE se basa en dos tipos de reuniones con características bien diferenciadas, *Status Meetings* y *On-demand Technical Meetings*, sin embargo, SCORE también recomienda otro tipo de prácticas como pueden ser intentar que el alumno desarrolle su proyecto en el mismo lugar de trabajo en el que se encuentra el profesor o facilitar encuentros sociales como un café o una charla distendida entre los miembros del grupo de investigación.

Status Meetings

Para [Hicks and Foster \(2010\)](#) se consideran la pieza central de esta metodología. Las *Status Meetings* son una adaptación de los *Daily Scrum* pero con una frecuencia de tres días por semana. Durante estas reuniones, todos los asistentes deben participar respondiendo tres preguntas:

1. ¿Qué has hecho desde la última reunión?
2. ¿Qué resultados has conseguido y qué dificultades has encontrado?
3. ¿Qué planeas hacer hasta la siguiente reunión?

Es una buena práctica que los profesores también participen en estas reuniones, ya que los estudiantes pueden aprender de sus actividades docentes y de investigación y así aumenten las posibilidades de que quieran optar por un puesto en la Universidad en el futuro. Muchas veces, los estudiantes no han podido avanzar en la correspondiente investigación debido a otros motivos, ya sean académicos o personales, por lo que es tarea del profesor mantener una actitud de comprensión para no desanimar al alumno.

Es importante notar que, aunque parezca que el objetivo de esta metodología es hacer que los estudiantes se centren rígidamente en objetivos a corto plazo, es más bien animarles para que logren algún tipo de progreso en su investigación día a día. Estas reuniones han

de tener una duración de no más de 15 minutos, lo que implica que cada persona tendrá entre 1 y 2 minutos para hablar, por lo que es necesario que los participantes no desarrollen en profundidad sus avances, sino simplemente dar unas pequeñas pinceladas ([Hicks and Foster, 2010](#)).

On-demand Technical Meetings

Como consecuencia de la naturaleza de las *Status Meetings*, las cuestiones o problemas de una profundidad más técnicas deben tratarse en otro tipo de reuniones. Así, surgen las *On-demand Technical Meetings* entre el tutor y el alumno para discutir acerca de cuestiones de investigación, métodos que utilizar, retos tecnológicos o resultados. En contraposición con las *Status Meetings*, estas reuniones no se realizan con una frecuencia acordada de antemano sino que el estudiante deberá pedir las cuando surja la necesidad de ello. Es muy común que se acuerden este tipo de reuniones como consecuencia de diversos aspectos surgidos durante las *Status Meetings* y que en estas no se hayan podido tratar con suficiente profundidad, o que hayan quedado dudas sin resolver.

Las *On-demand Technical Meetings* tratan de imitar las *sprint Reviews* del método SCRUM pero sin tener que realizarlas al terminar cada *sprint*. La frecuencia con las que se realizan depende principalmente de la experiencia del alumno o del progreso que lleva este en el desarrollo del proyecto de investigación. Por ello, [Hicks and Foster \(2010\)](#) asegura que, para aquellos alumnos que ya tienen cierta experiencia, se suelen mantener este tipo de reuniones cada 2 o 3 semanas. Sin embargo, semanas antes de la fecha de entrega del proyecto, se mantienen *On-demand Technical Meetings* casi de forma diaria. Por ello, una de las fortalezas de este enfoque radica precisamente en su capacidad para adaptarse a las necesidades de los estudiantes, de modo que aquellos que necesiten o deseen ayuda o consejo, lo obtengan lo antes posible.

2.1.3. Implementación de SCORE

Para la consecución de este Trabajo de Fin de Grado se ha seguido la metodología SCORE, principalmente por dos razones: (1) se trata de un trabajo de investigación, por lo que SCORE se adapta a la perfección para la confección de este proyecto y (2) se ha integrado también en el desarrollo de las Prácticas de Empresa del Grado en Ingeniería Informática de Servicios y Aplicaciones, que se realizaron en el laboratorio 2L027 de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid dentro del Grupo de Ingeniería Biomédica. Esto proporcionó un ambiente perfectamente adaptable a esta metodología.

El hecho de que tanto el primer tutor de este trabajo como yo nos encontrásemos en el mismo lugar de trabajo facilitó enormemente el planteamiento de las *Status Meetings*. Así, todos los lunes, martes y miércoles comentábamos los avances llevados a cabo y planes para la siguiente reunión. Dado que mi tutor debía compaginar docencia presencial en

la Escuela de Ingeniería Informática de Segovia con tareas de investigación, y yo recibir clases en Valladolid y Segovia por el Programa de Estudios Conjunto, a veces se reemplazaban las reuniones presenciales por correos electrónicos. Sin embargo, esta barrera no supuso ningún problema para seguir estrictamente la metodología SCORE.

A medida que fue avanzando el proyecto, se plantearon diversas *On-demand Technical Meetings* principalmente para la resolución de dudas técnicas. De nuevo, en ocasiones, esta comunicación fue vía correo electrónico, en caso de que se necesitase resolver la duda con prontitud.

Finalmente, además de seguir con precisión estos dos pilares del SCORE, se llevaron a cabo otros métodos secundarios que esta metodología recomienda, como son (1) el desarrollo del proyecto en el mismo lugar de trabajo que el tutor; (2) reuniones informales con el resto de miembros del GIB donde se intercambiaban opiniones tanto del tema a desarrollar en este Trabajo de Fin de Grado, como en otros aspectos de la neurociencia.

2.2. Herramientas utilizadas

A continuación, se detallan las herramientas utilizadas para la realización de este Trabajo de Fin de Grado. Estas varían entre lenguajes de programación, librerías, entornos de desarrollo integrado (IDEs, por sus siglas en inglés) y editores de texto.

- **Overleaf**: plataforma en línea que simplifica el proceso de redacción, edición y publicación colaborativa de documentos científicos. Combina la potencia del editor LaTeX con la conveniencia de la colaboración en tiempo real. Además, Overleaf genera automáticamente la salida compilada del documento en segundo plano mientras se escribe, lo que permite visualizar el formato final mientras se trabaja en el contenido. Esto hace que la redacción y edición de documentos científicos sean procesos más rápidos y sencillos.
- **Microsoft Excel**: es un programa de hojas de cálculo incluido en el conjunto de aplicaciones de Microsoft Office. Se utiliza para crear, manipular y modificar hojas de cálculo de manera eficiente. En concreto, se ha utilizado para almacenar los datos obtenidos tras los estudios offline y online en archivos de extensión csv (Comma Separated Values).
- **PyCharm**: desarrollado por JetBrains, es un IDE ampliamente utilizado en programación informática, especialmente para el lenguaje de programación Python. Ofrece una variedad de herramientas, como análisis de código, un depurador gráfico, un probador de unidades integrado y una integración sólida con sistemas de control de versiones (VCS). Este IDE es multiplataforma, compatible con Windows, macOS y Linux y se distribuye en diferentes ediciones.

- **Microsoft Visual Studio:** es un IDE disponible para Windows y macOS, que ofrece soporte para una amplia gama de lenguajes de programación como C++, C#, Fortran, Visual Basic .NET, F#, Java, Python, Ruby y PHP. En particular, para la consecución de este Trabajo de Fin de Grado se ha utilizado con el objetivo de escribir y *debuggear* scripts en C#.
- **Python:** es un lenguaje de programación interpretado de alto nivel que se destaca por su énfasis en la legibilidad del código. Es conocido por ser multiparadigma, ya que admite en gran medida la programación orientada a objetos, la programación imperativa y, en menor medida, la programación funcional. Además, Python es un lenguaje interpretado y dinámico, lo que permite una mayor flexibilidad durante el desarrollo. Además, es compatible con múltiples plataformas, lo que lo convierte en una opción popular para desarrolladores en diversos entornos.
- **Numpy:** es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para grandes conjuntos de datos. Introduce una nueva clase de objetos denominados **arrays**, que permiten representar colecciones de datos del mismo tipo en múltiples dimensiones, junto con funciones para manipular estos **arrays**. La principal ventaja de NumPy sobre las listas predeterminadas en Python radica en su velocidad de procesamiento, que puede ser hasta 50 veces más rápida que las listas estándar.
- **Pandas:** es librería de Python que ofrece un objeto DataFrame rápido y eficiente, fundamental para la manipulación de datos. Proporciona herramientas para leer y escribir datos en diversos formatos, como CSV, Excel o bases de datos SQL. Además, ofrece capacidades de alineación y manejo de datos faltantes, junto con opciones de reorganización y pivoteo de conjuntos de datos.
- **Matplotlib:** es una librería de Python destinada a la creación de visualizaciones estáticas, animadas e interactivas. Destacando por su versatilidad, esta herramienta permite producir gráficos de calidad para publicaciones, crear figuras interactivas con capacidades de zoom y desplazamiento, personalizar el estilo y diseño visual, exportar a una variedad de formatos de archivo, e integrar fácilmente en entornos como JupyterLab y interfaces de usuario gráficas. Además, Matplotlib se beneficia de una amplia gama de paquetes de terceros que amplían aún más sus funcionalidades.
- **Unity:** anteriormente conocido como Unity 3D, es una plataforma de desarrollo de software que integra todos los elementos necesarios para la creación de videojuegos. Esta herramienta proporciona la capacidad de desarrollar videojuegos para diversas plataformas, incluyendo PC, consolas de videojuegos y dispositivos móviles, mediante un editor visual y programación a través de scripting, permitiendo obtener resultados de alta calidad profesional.
- **C#:** es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft, utilizado para el desarrollo de scripts para controlar las funcionalidades de Unity.

- **Signal Generator:** aplicación desarrollada por el GIB para generar una señal de EEG sintética detectable a través de LSL (Lab Streaming Layer) en la plataforma MEDUSA[©].
- **GitHub:** es una plataforma en la nube que ofrece un sistema de control de versiones conocido como Git. Este servicio permite a los desarrolladores trabajar conjuntamente y realizar modificaciones en proyectos compartidos, manteniendo un registro detallado de los cambios y el progreso.
- **Mendeley:** es un gestor bibliográfico gratuito que combina una versión web con una versión de escritorio. Entre sus principales funcionalidades, se destaca la capacidad de capturar referencias o documentos desde múltiples sitios, organizar una base de datos personal de referencias y permitir la inserción de citas o la creación automática de bibliografías en \LaTeX .

Capítulo 3

Planificación

Todo proyecto necesita una planificación, tanto temporal, como económica. Por ello, este capítulo pretende mostrar al lector la estimación del esfuerzo de cada una de las tareas que conforman el proyecto, la planificación temporal de las mismas y el presupuesto económico inicial. Finalmente, se hará un balance final, comparando lo inicialmente previsto con la materialización final del proyecto.

3.1. Estimación del esfuerzo

En consonancia con la metodología SCRUM, cada historia de usuario incluida en el *Product Backlog* debe ser sometida a una estimación. Esta estimación supone realizar una reunión para identificar posibles dificultades, problemas y riesgos inherentes al desarrollo durante el *sprint*. Normalmente, se tienen en cuenta el esfuerzo, la complejidad y la incertidumbre de la historia de usuario. Para facilitar este proceso de estimación y evitar prolongadas discusiones dentro del equipo de desarrollo, se implementó el *planning poker* o *scrum poker*. El procedimiento es simple: durante la estimación, cada miembro del equipo posee un conjunto de cartas. Cada carta representa una valoración de la dificultad que supone la tarea. Todos los miembros muestran simultáneamente la carta que representa la suma del esfuerzo estimado. Cada equipo puede adaptar el juego de cartas a las numeraciones pertinentes según la unidad de esfuerzo empleada y el tamaño máximo de tarea a estimar. Las tareas que superen este tamaño máximo deben desglosarse en subtareas más pequeñas (*Planning Poker*, [Eniun](#)).

Estas cartas están numeradas siguiendo las tallas de ropa más comunes: *XS*, *S*, *M*, *L* y *XL* (véase la figura 3.1), que representan la dificultad de la tarea en orden creciente. El cliente o *Product Owner* presenta la historia de usuario, proporcionando la información necesaria al equipo de desarrollo, quienes pueden realizar preguntas para clarificar los detalles. Posteriormente, cada miembro del equipo selecciona una carta del mazo sin revelarla a los demás. Una vez que todos han elegido, las cartas se muestran simultáneamente. Luego, se discuten las discrepancias entre las estimaciones más altas y más bajas. Este proceso se repite hasta que se logre un consenso sobre la estimación final.

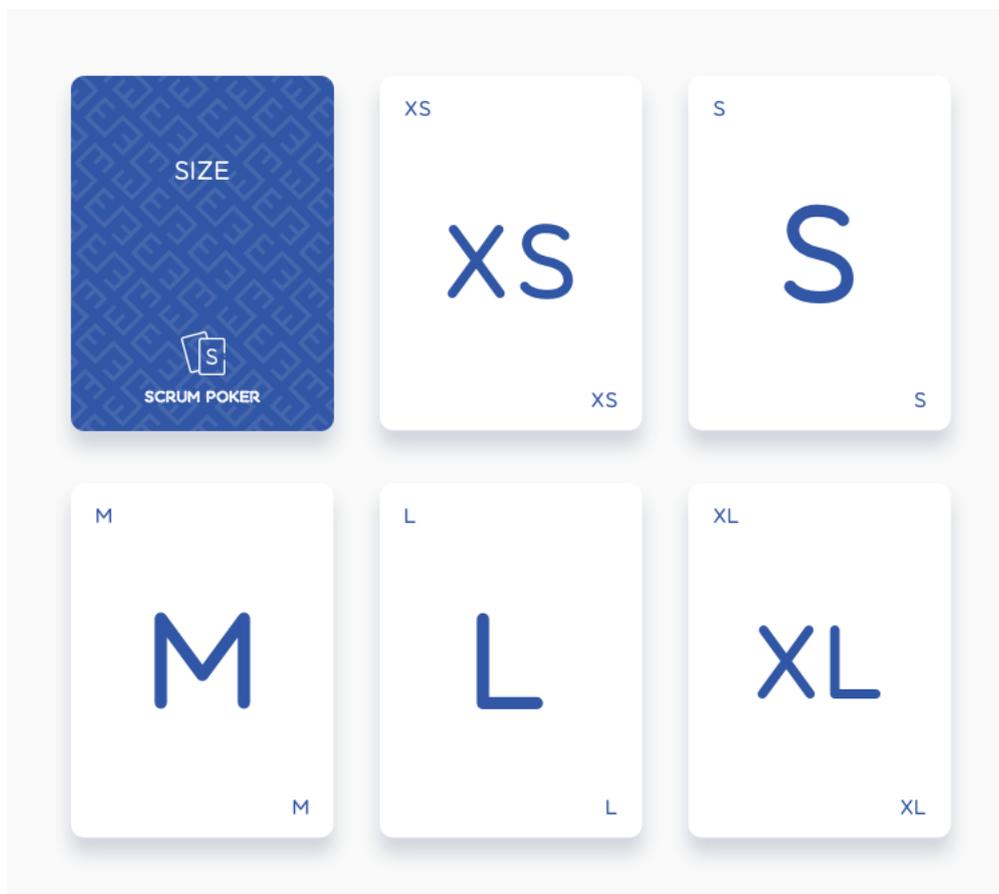


Figura 3.1: *Planning Poker* con cartas de tallas de ropa (*Planning Poker*, Eniun).

Con el objetivo de aplicar este procedimiento a la metodología SCORE explicada en el capítulo anterior, es necesario definir el conjunto de tareas a realizar. En concreto, se han dividido por Épicas y tareas (T). Para poner en práctica el *Planning Poker*, tanto el tutor como el alumno deben asignar una carta a cada una de las subpartes en las que se divide el proyecto. A continuación, se muestran los paquetes de trabajo definidos, las subtareas que los conforman y la carta finalmente elegida.

▪ **Épica-1. Formación.**

- **T-1.1.** Realización de cursos online de Python incluyendo *Crash Course on Python*, Google LLC (32 horas), *Using Python for Research*, Harvard University (48 horas), *Threading in Python*, Dev-Cademy Media Inc. (3 horas), *Speed up Python with concurrency*, Dev-Cademy Media Inc. (4 horas), *Hand-on Python 3 concurrency with the asyncio module*, Dev-Cademy Media Inc. (2.5 horas), *Functional Programming in Python*, Dev-Cademy Media Inc. (2 horas) (L).
- **T-1.2.** Realización del curso online de Unity (C#) *Unity Essentials Pathway*, Unity Technologies (10 horas) (M).

- **Épica-2. Revisión del estado del arte.**
 - **T-2.1.** Estudio general sobre sistemas BCI (Wolpaw and Wolpaw, 2012a) (S).
 - **T-2.2.** Estudio básico sobre teoría de la señal, incluyendo filtros espaciales y frecuenciales, transformada rápida de Fourier, señales digitales y analógicas (Rønning, 2021; Rao and Swamy, 2018) (S).
 - **T-2.3.** Estudio sobre señales de control, focalizado en c-VEP (Martínez-Cagigal et al., 2021; Sun et al., 2022) (M).
 - **T-2.4.** Estudio profundo sobre los métodos de extracción de filtros espaciales CCA (Martínez-Cagigal et al., 2021; Nakanishi et al., 2018) y TRCA (Nakanishi et al., 2018; Sun et al., 2022; Tanaka, 2020) (S).
 - **T-2.5.** Estudio sobre generación de secuencias de números pseudoaleatorios, focalizado en MLS con LFSR y Gold Codes (Kedia, 2012) (XS).
- **Épica-3. Desarrollo e implementación de la aplicación en MEDUSA[©]** (<https://www.medusabci.com/>).
 - **T-3.1.** Estudio y familiarización con MEDUSA[©] Kernel, MEDUSA[©] Platform y la arquitectura de las aplicaciones (Santamaría-Vázquez et al., 2023) (XS).
 - **T-3.2.** Implementación del algoritmo TRCA (S).
 - **T-3.3.** Refactorización de código para adaptar la extracción de filtros espaciales a TRCA (L).
 - **T-3.4.** Implementación de los Gold Codes (S).
 - **T-3.5.** Adaptación de la configuración de la aplicación a los Gold Codes (M).
- **Épica-4. Desarrollo de la aplicación en Unity.**
 - **T-4.1.** Modificación de scripts de C# de la aplicación *c-VEP Speller* (https://www.medusabci.com/market/cvpep_speller/) (L).
- **Épica-5. Evaluación de la aplicación.**
 - **T-5.1.** Análisis offline de las bases de datos de Thielen et al. (2021) y Martínez-Cagigal et al. (2023) (L).
 - **T-5.2.** Análisis online con 10 sujetos (L).
- **Épica-6. Análisis de los resultados y conclusiones** (S).
- **Épica-7. Documentación del proyecto.**
 - **T-7.1.** Descripción del proyecto (S).
 - **T-7.2.** Documentación de la metodología de trabajo (S).
 - **T-7.3.** Documentación de la planificación (S).

- **T-7.4.** Documentación del marco teórico (M).
- **T-7.5.** Documentación del estado de la técnica (S).
- **T-7.6.** Documentación del paradigma c-VEP (L).
- **T-7.7.** Documentación del diseño y desarrollo del sistema BCI (L).
- **T-7.8.** Documentación del protocolo de evaluación (S).
- **T-7.9.** Documentación de los resultados del experimento (M).
- **T-7.10.** Documentación de la discusión del proyecto (S).
- **T-7.11.** Documentación de las conclusiones y líneas futuras (M).

Es importante notar que las épicas 1 y 2 se corresponden con la consecución del objetivo 1, las épicas 3 y 4 con el objetivo 2 y las épicas 5 y 6 con el objetivo 3, descritos en la sección 1.2. Además, si se proporciona un valor numérico a cada una de las cartas en función del orden de dificultad XS : 1, S : 2, M : 3, L : 4 y XL : 5, se obtiene una puntuación total de 72 puntos de historia.

3.2. Planificación temporal

Una vez finalizada la estimación del esfuerzo y la fragmentación de las tareas a realizar, es necesario agrupar estas en bloques de trabajo. Siguiendo la metodología SCORE expuesta en el capítulo anterior, estos bloques son análogos a los *sprints* del SCRUM y tienen una fecha de inicio y fin aproximadas, alcanzando así una duración aproximada de 25 días lectivos, dependiendo el bloque. Como se observa en la tabla 3.1, en la que se utiliza la misma notación previamente expuesta, los bloques de tareas representan un porcentaje similar del proyecto. El inicio de este proyecto es el 1 de agosto de 2023 y se previó finalizarlo el día 1 de junio.

En esta misma tabla aparece la división por bloques de tareas, la fecha de inicio y finalización aproximada, el conjunto de tareas a realizar, y la puntuación asignada a través de *Planning Poker*, así como el progreso que representan. Una vez terminado cada bloque, se realizó una *On-demand Technical Meeting* con el tutor del presente documento para analizar el trabajo realizado, resolver posibles dudas y para realizar modificaciones antes de que los errores se propaguen por el resto del documento.

Hay un total de 174 días lectivos aproximadamente en el total de bloques de trabajo. Para la planificación económica, hay que tener en cuenta el número de horas que se dedica cada día a este proyecto, por ello, se fija en 3 horas diarias, lo que supone un total de 522 horas de trabajo. El Trabajo de Fin de Grado en el Grado en Ingeniería Informática de Servicios y Aplicaciones tiene una carga lectiva de 12 ECTS, lo que supondría un total de 300 horas. Se observa que la dedicación aproximada de 522 horas supera este cota con creces.

Tabla 3.1: Distribución temporal en bloques de las tareas a realizar.

Bloque	Fecha de inicio	Fecha de finalización	Tareas	Puntuación asignada	Progreso
1	01/08/2023	09/10/2023	T-1.1, T-1.2, T-2.1	9	12,5 %
2	10/10/2023	02/11/2023	T-2.2, T-2.3, T-3.1, T-7.1	8	23,6 %
3	03/11/2023	11/01/2024	T-2.4, T-3.2, T-3.3, T-7.2	10	37,5 %
4	12/01/2024	17/02/2024	T-2.5, T-3.4, T-3.5, T-7.3	8	48,6 %
5	18/02/2024	05/03/2024	T-5.1, T-7.4, T-7.5	9	61,1 %
6	06/03/2024	31/03/2024	T-4.1, T-7.6, T-7.7	12	77,8 %
7	01/04/2024	12/05/2024	T-5.2, T-7.8	6	86,1 %
8	13/05/2024	01/06/2024	T-6, T-7.9, T-7.10, T-7.11	10	100 %

La figura 3.2 muestra el diagrama de Gantt de la planificación temporal de las tareas. En él se puede apreciar la agrupación por bloques, las fechas de inicio y fin estimadas para cada una de las tareas, así como la carta asignada para cada subtarea, representada por colores. Finalmente, es importante notar que, debido a que este proyecto solo se realiza por una persona, no es posible realizar tareas en paralelo y, consecuentemente, estas se han configurado de forma secuencial.

3.3. Presupuesto económico

Antes de comenzar con el desarrollo del proyecto, es necesario realizar una estimación de los costes que va a suponer. Para tener cierto rigor a la hora de realizar la planificación económica, es necesario dividir los costes por categorías.

En primer lugar, atendiendo a los costes derivados del hardware, se ha dividido en dos partes. La primera, descrita en la tabla 3.2, hace referencia al material hardware utilizado sin tener en cuenta aquellos dispositivos propios de un sistema BCI. En concreto, se utilizará un ordenador de sobremesa con un procesador Intel[®] Core[™] i7-10700F CPU a 2.90 GHz, una memoria RAM de 32 GB, con un sistema operativo Windows de 64 bits y una tarjeta gráfica GeForce 1050 Ti. La vida útil de este tipo de equipos se ha estimado en 7 años, por lo que, teniendo en cuenta que el proyecto tendrá una duración de 10 meses, se hará un uso del 12 % de su vida útil. También se hará uso de un teclado y un ratón, con

Capítulo 3. Planificación

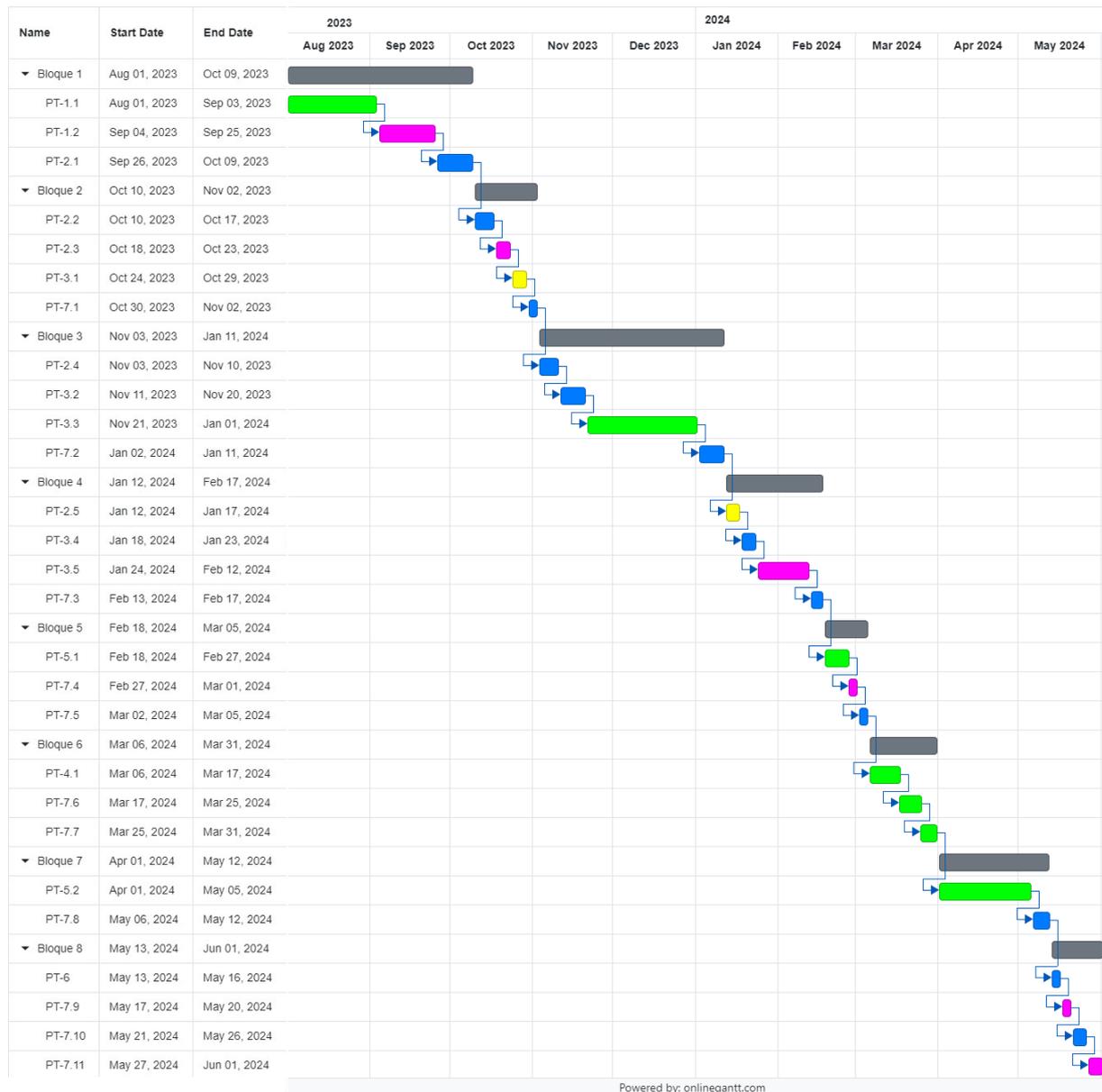


Figura 3.2: Diagrama de Gantt de la planificación temporal del proyecto.

una vida útil media de 3 años y una pantalla de 120 Hz con una vida útil aproximada de 15 años.

Tabla 3.2: Costes del proyecto derivados del hardware.

	Coste (€)	Vida útil (meses)	Uso (%)	Total (€)
Ordenador de sobremesa	1.500,00	84	12	180,00
Teclado y ratón	20,00	36	28	5,60
Pantalla 120 Hz	250,00	180	5	12,50

Tabla 3.3: Costes del proyecto derivados de componentes BCI.

	Coste (€)	Vida útil (meses)	Uso (%)	Total (€)
Amplificador g.USBamp EEG	14.399,00	50	2	287,98
g.GAMMAbox	1.319,00	50	2	26,38
16 g.LADYbird EEG electrodos	2.112,00	50	2	42,24
g.LADYbird electrodo tierra	84,00	50	2	1,68
g.GAMMAsys electrodo referencia	144,00	50	2	2,88
g.GAMMAcap3 gorro EEG	217,00	50	2	4,34
Material limpieza electrodos	47,00	5	20	9,40
gel Neuroline	35,00	5	20	7,00

En esta misma línea, la tabla 3.3 muestra los costes derivados de los componentes BCI para el experimento. Estos incluyen un amplificador *g.USBamp*, una *g.GAMMAbox* para conectar los electrodos al amplificador, 16 electrodos *g.LADYbird*, 1 electrodo tierra *g.LADYbird*, 1 electrodo referencia *g.GAMMAsys*, un gorro *g.GAMMAcap3*, material de limpieza para los electrodos y gel electrofisiológico conductor *g.GAMMAgel*. Nótese que este tipo de componentes sólo se utilizarán durante el paquete de trabajo PT-5.2, por lo que el porcentaje de uso con respecto a su vida útil es solamente del 2%.

La tabla 3.4 muestra los costes derivados de los servicios en los que únicamente se incluye la conexión a Internet y la tabla 3.5 incluye los costes de las licencias software

Tabla 3.4: Costes del proyecto derivados de los servicios.

	Coste (€)	Vida útil (h/meses)	Uso (%)	Duración	Total (€)
Conexión a Internet	50 €/mes	744	12,5	10 meses	62,50

Tabla 3.5: Costes del proyecto derivados de las licencias.

	Coste (€)	Vida útil (%)	Total (€)
Overleaf	0,00	100	0,00
PyCharm (Community Edition)	0,00	100	0,00
MEDUSA [©]	0,00	100	0,00
Unity	0,00	100	0,00

utilizadas.

Finalmente, en cuanto al personal necesario del proyecto, se necesitará un desarrollador de Python junior y un técnico de Laboratorio. Éste último será necesario para realizar el experimento con los sujetos (PT-5.2). Si se fija un sueldo bruto medio del primero de 23.000 € y del segundo de 18.000 €, y se consideran 1750 horas anuales, entonces, la tabla 3.6 muestra el coste en nóminas del proyecto.

Teniendo en cuenta todos estos costes, la tabla 3.7 recoge los costes totales del proyecto, siendo estos 7.383,32 €.

3.4. Balance final

Una vez realizado la planificación tanto temporal como económica, es necesario compararla con el balance final del proyecto, tanto en la planificación de tareas como en los costes.

Tras la finalización de este Trabajo de Fin de Grado, se realizaron una serie de cambios con respecto a la planificación inicial. En concreto, el número total de cambios han sido tres. En primer lugar, debido a los exámenes de la convocatoria ordinaria del primer cuatrimestre se realizó un parón en el proyecto desde el 1 de enero de 2024 hasta el 17 de enero de 2024, quedando las posteriores tareas desplazadas 16 días de su original fecha de inicio. También, la estimación inicial para la realización de la tarea T-5.2 fue de 35 días,

Tabla 3.6: Costes del proyecto derivados del personal.

	Coste (€/hora)	Duración (horas)	Total (€)
Desarrollador Python junior	13,25	461	6.108,25
Técnico de laboratorio	10,37	61	632,57

Tabla 3.7: Costes totales del proyecto.

	Total (€)	% del total
Hardware	198,10	2,68
Componentes BCI	381,90	5,17
Servicios	62,50	0,85
Licencias	0,00	0,00
Personal	6.740,82	91,30
	7.383,32	100

sin embargo, sólo fueron necesarios 20 días. Finalmente, se identificó una nueva tarea no tenido en cuenta antes:

- **T-7.12.** Revisión de toda la documentación (\underline{S}).

Teniendo en cuenta estos tres factores, la fecha de finalización del proyecto fue el 16 de junio de 2024, 15 días más tarde de lo previsto. La distribución temporal final por bloques de las tareas realizadas se puede observar en la tabla 3.8, así como el Diagrama de Gantt final en la figura 3.3.

En cuanto a los costes del proyecto derivados de los cambios en las tareas se observa que ahora la duración del proyecto es de 10,5 meses por lo que aumenta levemente el coste de los servicios (véase tabla 3.9) y de las nóminas (véase tabla 3.10). En concreto, se ha disminuido un 28,6% las horas de trabajo del técnico de laboratorio debido al reajuste del paquete de trabajo PT-5.2. También, se han añadido 45h de trabajo al desarrollador de Python por la nueva tarea T-7.12.

Finalmente los costes del proyecto ascienden a 7.761,40 € (véase tabla 3.11) con un total de 550 horas realizadas aproximadamente.

Tabla 3.8: Distribución temporal por bloques final de las tareas realizadas.

Bloque	Fecha de inicio	Fecha de finalización	Tareas	Puntuación asignada	Progreso
1	01/08/2023	09/10/2023	T-1.1, T-1.2, T-2.1	9	12,1 %
2	10/10/2023	02/11/2023	T-2.2, T-2.3, T-3.1, T-7.1	8	22,9 %
3	03/11/2023	26/01/2024	T-2.4, T-3.2, T-3.3, T-7.2	10	36,4 %
4	27/01/2024	03/03/2024	T-2.5, T-3.4, T-3.5, T-7.3	8	47,2 %
5	04/03/2024	20/03/2024	T-5.1, T-7.4, T-7.5	9	59,4 %
6	21/03/2024	15/04/2024	T-4.1, T-7.6, T-7.7	12	75,6 %
7	16/04/2024	12/05/2024	T-5.2, T-7.8	6	83,7 %
8	13/05/2024	16/06/2024	T-6, T-7.9, T-7.10, T-7.11, T-7.12	12	100 %

Tabla 3.9: Costes del proyecto final derivados de los servicios.

	Coste (€)	Vida útil (h/meses)	Uso (%)	Duración	Total (€)
Conexión a Internet	50 €/mes	744	12,5	10,5 meses	65,62

Tabla 3.10: Costes finales del proyecto derivados del personal.

	Coste (€/hora)	Duración (horas)	Total (€)
Desarrollador Python junior	13,25	506	6704,5
Técnico de laboratorio	10,37	44	456,28

3.4. Balance final



Figura 3.3: Diagrama de Gantt final de la planificación temporal del proyecto.

Tabla 3.11: Costes totales finales del proyecto.

	Total (€)	% del total	Incremento (€)
Hardware	198,10	2,55	0,00
Componentes BCI	381,90	4,92	0,00
Servicios	65,62	0,85	3,12
Licencias	0,00	0,00	0,00
Personal	7.160,78	92,26	419.96
	7.761,40	100	378,08

Capítulo 4

Marco Teórico

En el presente capítulo se explicarán las partes principales del sistema nervioso central, así como la anatomía cerebral. A continuación, se analizarán las áreas funcionales del cerebro y las técnicas más populares de adquisición de la actividad cerebral. Finalmente, se introducirán los sistemas BCI basados en EEG y las principales señales de control.

4.1. El sistema nervioso central

El sistema nervioso está formado por una extensa red de neuronas con funciones tan importantes como recibir, analizar y transmitir la información de todo el organismo. Nos permite realizar acciones complejas como conocer, aprender, elaborar pensamientos y comunicarnos a través del lenguaje oral, escrito y gestual ([Castillo and de Jorge, 2015](#)). Sus estructuras se organizan en dos grandes bloques: el sistema nervioso central (SNC) y el sistema nervioso periférico (SNP). A su vez, el SNC está formado por el encéfalo y la médula espinal (ME), mientras que el SNP lo integra tejido nervioso que pone en contacto el encéfalo y la ME con diferentes estructuras de nuestro organismo. El SNP lo componen nervios, tanto craneales como raquídeos o espinales ([Moore et al., 2013](#)). Nos centraremos únicamente en el estudio detallado del SNC.

El encéfalo está ubicado en el interior del cráneo y está integrado por cuatro estructuras denominadas cerebelo, diencefalo, tronco del encéfalo y cerebro. La ME está situada en el interior del canal vertebral y se extiende desde el tronco del encéfalo hasta la región lumbar. Entre sus funciones destacan coordinar reflejos y transmitir las señales motoras y sensoriales ([Leblanc, 2004](#)). El cerebelo es una gran masa encefálica situada en la parte posterior e inferior del cráneo. Se le considera el área silente del encéfalo ya que su excitación eléctrica no origina ninguna sensación consciente. Su función principal es participar en el control de la postura y del equilibrio, colaborar con la corteza cerebral en la planificación del siguiente movimiento mientras aún se está ejecutando una acción motora, y evaluar cómo se realizan los movimientos iniciados en las áreas motoras del cerebro. Resulta especialmente importante en actividades musculares con movimientos alternativos rápidos como correr o hablar ([Pombano and Evans, 2013](#)). El diencefalo está

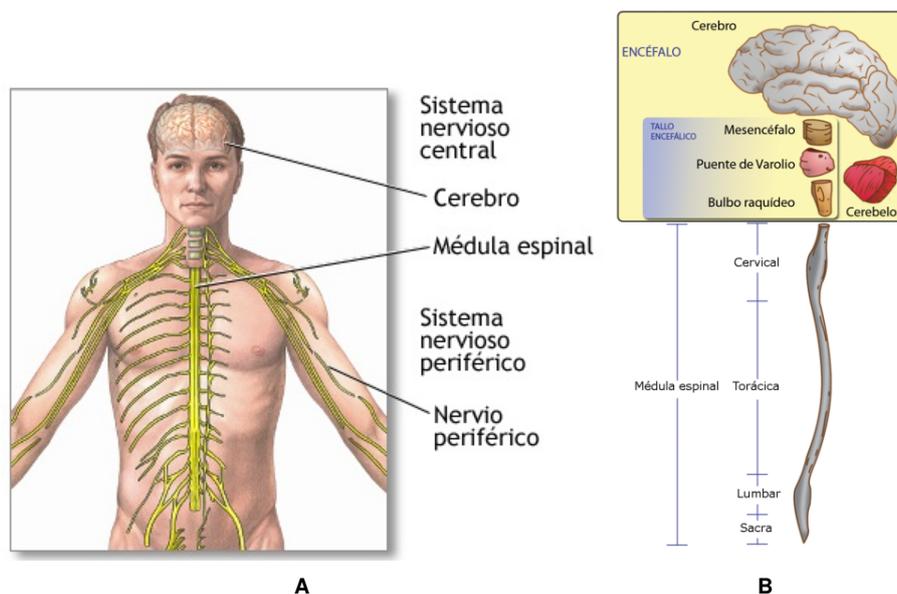


Figura 4.1: Esquema general del sistema nervioso. (A) Representación gráfica del sistema nervioso central y periférico. Imagen adaptada de [MedlinePlus, National Library of Medicine](#). (B) Esquema del sistema nervioso central. Imagen adaptada de [Snell \(2010\)](#).

formado por el tálamo y el hipotálamo, estructuras que son fundamentales en funciones tales como la cognición, el control de las emociones, la memoria, el control de actos motores voluntarios, el despertar, la regulación de la temperatura corporal, de la ingesta de bebidas y alimentos, o de los ritmos circadianos, entre otras. El tronco del encéfalo pone en contacto el cerebro con la médula espinal y está integrado por estructuras como el bulbo raquídeo, la protuberancia y el mesencéfalo. De él salen los 12 pares craneales que son los nervios encargados de inervar parte de las estructuras situadas en la cabeza. Controla funciones vitales como la respiración, el ritmo cardíaco y la presión arterial. Además, está formado por estructuras como la sustancia negra (que participa en la regulación de la actividad muscular) y la formación reticular (que se encarga de mantener la conciencia y el despertar) entre otras [Moore et al. \(2013\)](#). Finalmente, también forma parte del SNC el cerebro (véase la figura 4.1).

4.1.1. Aspectos generales de la anatomía cerebral

El cerebro tiene funciones tan significativas como la memoria, la creatividad, los recuerdos, la toma de decisiones, el lenguaje y la imaginación, entre otras. Consta de una capa superficial de sustancia gris o corteza cerebral, y de la denominada sustancia blanca, ubicada por debajo de la anterior. La corteza cerebral tiene un grosor de 2 a 5 mm y contiene aproximadamente 100.000 millones de neuronas. Su superficie está formada por pliegues denominados circunvoluciones, que tienen partes más profundas llamadas cisuras y otras menos profundas o surcos. Es imprescindible tener en cuenta que, aunque los

procesamientos complejos como el pensamiento involucran conexiones entre multitud de áreas cerebrales, existen áreas más especializadas que otras, las cuales vemos a continuación. La cisura interhemisférica divide el cerebro en dos hemisferios cerebrales, derecho e izquierdo, y cada uno de ellos se organiza en cuatro lóbulos: frontal en su parte anterior; parietal, temporal y occipital en su parte más posterior (Hall and Guyton, 2016).

- **El frontal** decide la conducta motora apropiada en cada momento. También alberga tareas complejas no motoras como la planificación de la conducta, el control de nuestras emociones, el razonamiento y juicio. En el lóbulo frontal izquierdo se encuentra el área de Broca encargada fundamentalmente de la expresión de estructuras sintácticas, del procesamiento verbal y de la planificación y programación motora para la articulación del habla (Hall and Guyton, 2016).
- **El occipital** se encarga específicamente de la visión, elaborando la información visual que posteriormente trasciende a los lóbulos parietales y temporales (Hall and Guyton, 2016).
- **El temporal** es el lóbulo del procesamiento auditivo. En él asientan importantes estructuras de la memoria (hipocampo) y del sistema emotivo inconsciente (sistema límbico). El lóbulo temporal izquierdo contiene el área de Wernicke con funciones en la comprensión auditiva del lenguaje oral y el procesamiento de la selección del léxico (Hall and Guyton, 2016).
- **El parietal** relaciona la información sensorial interoceptiva y exteroceptiva. Se le atribuyen funciones sensitivas, asociativas, así como de reconocimiento del espacio (Hall and Guyton, 2016).

4.1.2. Áreas funcionales del cerebro

La corteza cerebral está formada por tres grandes regiones o áreas funcionales: áreas sensoriales, áreas motoras y áreas de asociación. Todas ellas están conectadas con estructuras más profundas del encéfalo a través de conexiones aferentes y eferentes. A continuación, únicamente se mencionan las áreas sensoriales ya que son aquellas que más se relacionan con este Trabajo de Fin de Grado.

Las **áreas sensoriales**, situadas en la zona más posterior de la corteza cerebral, reciben e interpretan impulsos relacionados con las sensaciones (véase la figura 4.2). Estas áreas engloban a las siguientes (Hall and Guyton, 2016):

- **Área somatosensorial primaria:** localizada en la circunvolución parietal ascendente, justo por detrás de la cisura de Rolando. Su función principal es localizar con exactitud los puntos donde se originan las sensaciones, por ello está especializada en el análisis de la información procedente de los mecanorreceptores, propioceptores, nociceptores y termorreceptores. En ella está representado un mapa de la superficie corporal denominado homúnculo sensitivo. Cada una de las partes del

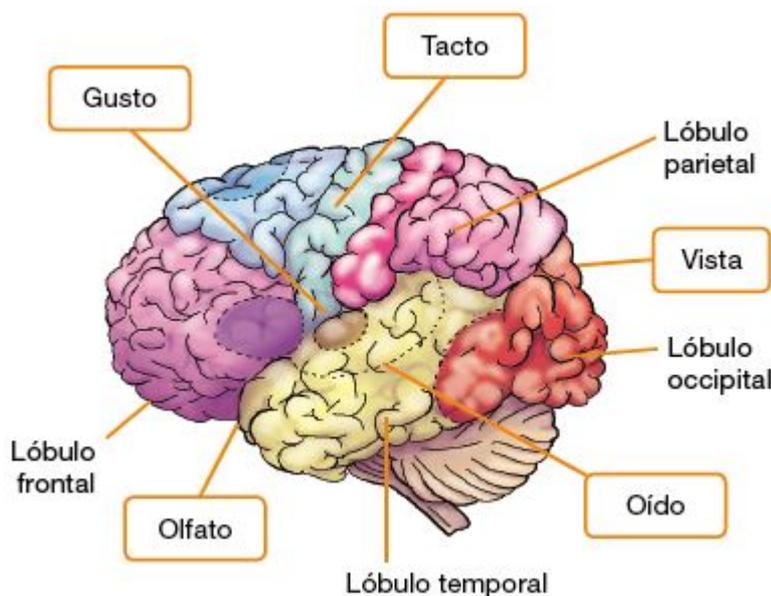


Figura 4.2: Esquema general del cerebro humano dividido por lóbulos y áreas ([Blink Learning](#)).

área capta sensaciones de una parte específica del organismo. El tamaño relativo de estas partes es proporcional al número de receptores sensoriales en la parte corporal respectiva. Así, algunas zonas corporales, como labios, cara, lengua y pulgar, están representadas por áreas más grandes de la corteza somatosensorial, mientras que la representación del tronco es mucho menor.

- **Área visual:** localizada en el lóbulo occipital, recibe impulsos que transmiten información visual como son el color, la forma y el movimiento de los estímulos visuales.
- **Área auditiva:** situada en el lóbulo temporal, se encarga de interpretar las características básicas de los sonidos, como su tonalidad y ritmo.
- **Área gustativa:** está localizada en el lóbulo parietal y percibe estímulos gustativos. Es la responsable de la percepción del gusto.
- **Área olfatoria:** se localiza en el lóbulo temporal. Recibe impulsos relacionados con la olfacción.

4.1.3. Adquisición de la actividad cerebral

La adquisición de la actividad cerebral es un proceso fundamental que implica la captura y el análisis de los patrones eléctricos y químicos generados por las neuronas en el cerebro. Este fenómeno es crucial para comprender cómo se procesa la información y se generan respuestas cognitivas y comportamentales. A la hora de diseñar un sistema BCI en concreto, es necesario conocer las ventajas y desventajas de las distintas técnicas de

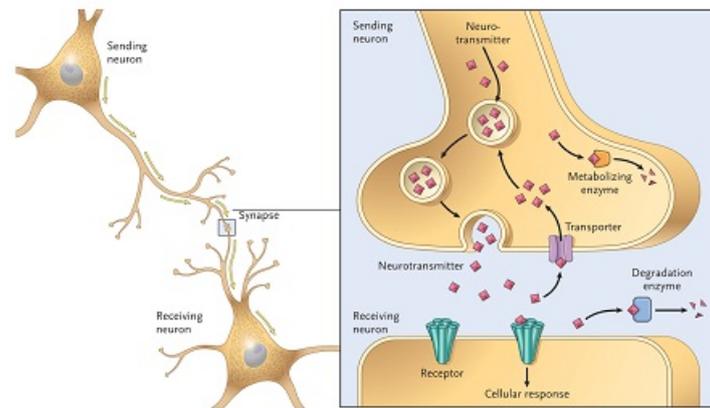


Figura 4.3: Proceso de sinapsis neuronal (University of Maryland, School of Medicine, 2015).

adquisición de la actividad cerebral (Wolpaw and Wolpaw, 2012a).

Desde una perspectiva microscópica, la actividad cerebral surge a través de la interacción de los neurotransmisores de las neuronas. Este proceso químico desencadena un desequilibrio en los niveles de iones entre el interior de una neurona específica y el fluido extracelular, lo que provoca una rápida despolarización que se transmite a lo largo del axón neuronal, como se puede apreciar en la figura 4.3. De forma práctica, este intercambio resulta en variaciones del potencial de membrana de la neurona, generando lo que se conoce como un potencial de acción. Al llegar al extremo del axón, los neurotransmisores son nuevamente liberados desde las vesículas sinápticas, facilitando así la comunicación de célula a célula (Standing, 2016).

Sin embargo, desde un punto de vista macroscópico, los sistemas BCIs, haciendo uso de una serie de estructuras de datos y métodos propios, registran el efecto del proceso neuronal descrito anteriormente para así monitorizar la actividad cerebral. En esta misma línea, según los fundamentos naturales en los que se basan, la adquisición de la actividad cerebral pueden categorizarse en aquellas que detectan la actividad eléctrica, magnética o metabólica.

- **Eléctrica:** la manera más natural de supervisar la actividad cerebral es detectar las fuentes eléctricas ligadas a la actividad sináptica de las neuronas individuales. Según la resolución espacial de los electrodos, esta actividad se puede medir en micro, meso y macroescalas del córtex. Las microescalas reflejan la actividad de pequeños volúmenes de tejido, comparable a neuronas individuales, y pueden ser registradas con microarrays. Las mesoescalas, que cubren módulos neuronales, se registran generalmente mediante electrocorticografía (ECoG), con electrodos sobre la corteza. Las macroescalas, que muestran la actividad eléctrica global en áreas cerebrales, se captan con EEG, colocando los electrodos sobre el cuero cabelludo (Nunez, 2012). Todos estos métodos tienen resoluciones temporales notables, entre

1 y 15 milisegundos (i.e., permiten un registro en tiempo real de la señal, empleando frecuencias de muestreo de 250-1000 Hz), pero hay un compromiso entre la precisión espacial y el grado de invasividad: los microarrays y el ECoG requieren electrodos dentro o sobre la corteza, mientras que el EEG es no invasivo.

- **Magnética:** el fenómeno magnético surge a partir de las mismas corrientes eléctricas que generan el campo eléctrico, siendo detectable mediante la técnica de magnetoencefalografía (MEG). Sin embargo, debido a la baja frecuencia de las señales cerebrales, los campos eléctricos y magnéticos generados por el cerebro se encuentran desacoplados (Srinivasan, 2012). En otras palabras, la MEG se orienta hacia la medición de un campo magnético cuasiestático que no guarda relación con el campo eléctrico reflejado por el EEG. A pesar de la información complementaria que ofrece la MEG, el campo magnético cerebral resulta ínfimo en comparación con las inevitables variaciones magnéticas ambientales. Por consiguiente, se hace necesario emplear un magnetómetro de interferencia cuántica superconductora (SQUID) a una temperatura extremadamente baja y en una cámara especialmente blindada (Srinivasan, 2012). Aunque carece de portabilidad, la MEG es una técnica no invasiva que proporciona una excelente resolución temporal, de 1 a 15 milisegundos y una resolución espacial comparable a la del EEG.
- **Metabólica:** el incremento en la actividad neuronal conlleva un aumento en el consumo de energía, que se suple con glucosa y oxígeno, incrementando así el flujo sanguíneo en la región correspondiente. Técnicas como la tomografía por emisión de positrones (PET), la espectroscopia funcional de infrarrojo cercano (fNIRS) y la resonancia magnética funcional (fMRI) permiten medir indirectamente la actividad cerebral a través del flujo sanguíneo. La PET, aunque invasiva y con exposición a radiación, detecta el flujo sanguíneo mediante la aniquilación electrón-positrón. En cambio, fNIRS y fMRI son no invasivas y se basan en la respuesta dependiente del nivel de oxígeno en la sangre (BOLD) (Ramsey, 2012). La hemoglobina, transportadora de oxígeno en la sangre, presenta diferencias en su espectro de absorción que permiten su seguimiento en fNIRS (Pinti et al., 2020). Mientras tanto, fMRI aprovecha las propiedades magnéticas de la hemoglobina para medir la respuesta BOLD (Ogawa et al., 1990). A pesar de sus ventajas, tanto fNIRS como fMRI tienen limitaciones en la resolución temporal debido a la duración de la respuesta BOLD y requieren equipos costosos y no portátiles para su aplicación efectiva (Ramsey, 2012).

La figura 4.4 muestra la relación entre la resolución temporal y espacial, así como el grado de invasividad de los distintos métodos expuestos. Aquellos BCIs cuyo objetivo principal sea práctico, necesitan operar en tiempo real con excelente resolución temporal. Por ello, la adquisición de la actividad cerebral a través de señales metabólicas se utilizan principalmente en la investigación. En su mayoría, los sistemas BCI priorizan la portabilidad y evitan métodos invasivos, prefiriendo el EEG, como en el desarrollo de este Trabajo de Fin de Grado.

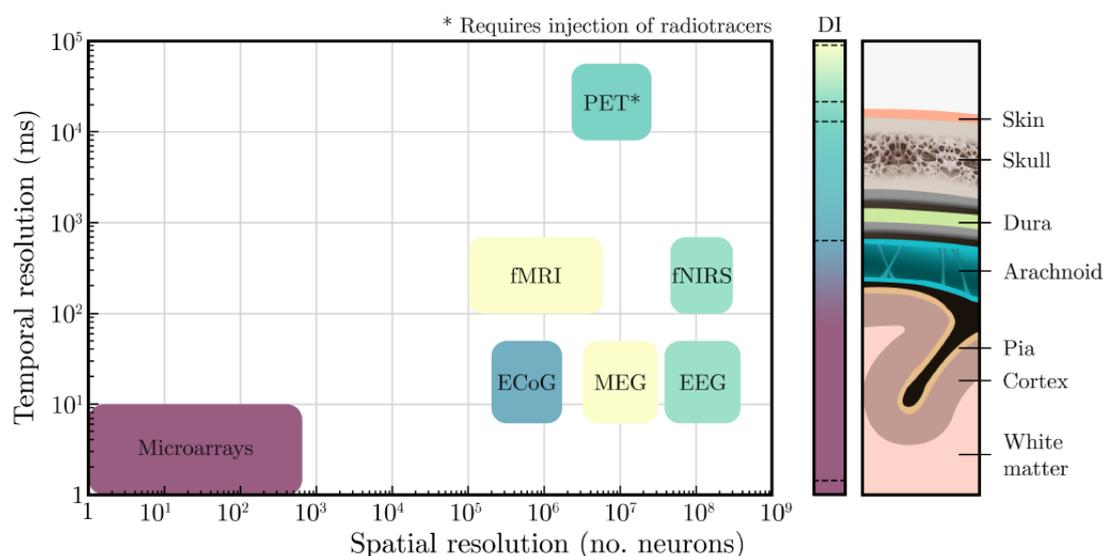


Figura 4.4: Gráfico que compara las distintas técnicas de adquisición de la actividad cerebral según su resolución espacial y temporal. El grado de invasividad (DI, por sus siglas en inglés) se representa como un gradiente de color, según las capas cerebrales (derecha). ECoG: electrocorticografía, MEG: magnetoencefalografía, EEG: electroencefalografía, fMRI: imagen por resonancia magnética funcional, fNIRS: espectroscopia funcional de infrarrojo cercano, PET: tomografía por emisión de positrones. (Martínez-Cagigal, 2020).

4.2. Sistemas BCI

En 1924, el Profesor Hans Berger realizó un descubrimiento crucial al registrar las señales eléctricas del cerebro desde el cuero cabelludo, estableciendo así la electroencefalografía (EEG) como una herramienta esencial para el estudio del cerebro y el diagnóstico clínico. (Berger, 1929). A pesar de su reconocimiento internacional, en 1938 su laboratorio fue cerrado por el gobierno alemán, truncando temporalmente su investigación. Sin embargo, el potencial de las señales EEG para la comunicación ha sido motivo de gran interés tanto para científicos como para el público en general. A medida que nos acercamos al centenario de este descubrimiento, la investigación en interfaces cerebro-computadora está experimentando un crecimiento exponencial, explorando los principios subyacentes, los logros actuales, los desafíos y las prometedoras perspectivas futuras en este campo fascinante y de rápido desarrollo (Wolpaw and Wolpaw, 2012b).

El sistema nervioso central coordina las respuestas del organismo a estímulos externos o internos mediante salidas neuromusculares o hormonales, como se mencionó en la sección anterior. Wolpaw and Wolpaw (2012b) aseguran que un sistema BCI proporciona al CNS una nueva forma de salida artificial. Esta tecnología mide la actividad del CNS y la convierte en salidas artificiales que pueden reemplazar, mejorar o complementar las funciones naturales del CNS. De forma resumida, un sistema BCI comprende las siguientes funciones:

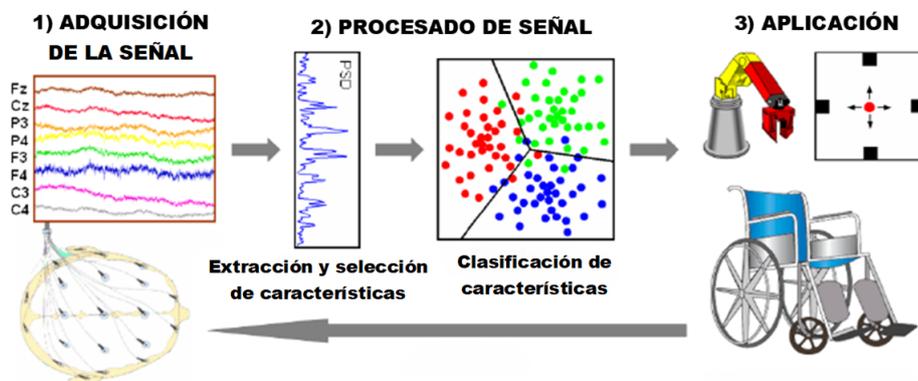


Figura 4.5: Estructura clásica de un sistema BCI. Imagen adaptada de [Millan et al. \(2008\)](#).

1. Sustituir las salidas naturales del CNS que se han perdido debido a lesiones o enfermedades.
2. Restaurar las funciones naturales del CNS perdidas, como el movimiento de las extremidades o la micción.
3. Mejorar la salida natural del SNC al prevenir lapsos de atención que podrían afectar negativamente el desempeño de una tarea.
4. Complementar la salida natural del CNS al agregar una salida artificial adicional como seleccionar objetos con un cursor de ordenador.
5. Mejorar la salida natural del CNS como la optimización de la atención, la memoria o la cognición.

4.2.1. El ciclo cerrado

Independientemente de su objetivo final, los sistemas BCI tienen en común un ciclo estructural cerrado. El funcionamiento de esta tecnología está basado en dicho ciclo y está formado por tres principales escenarios ([Martínez-Cagigal, 2020](#)), como se observa en la figura 4.5:

- 1) **Adquisición de la señal:** como se explicó anteriormente en la sección 4.1.3, es imprescindible registrar la actividad cerebral ya sea por métodos eléctricos, magnéticos o metabólicos. Hoy en día, los sistemas BCI basados en EEG son los que más se usan debido a su precio, portabilidad y facilidad a la hora de usarlo.
- 2) **Procesado de señal:** como se mencionó en la sección 1.1, ninguna técnica de adquisición de las señales producidas por la actividad cerebral es capaz de decodificar directamente las intenciones de los usuarios. Por tanto, el procesado de la señal es esencial para identificar alteraciones cuantificables que guarden correlación con las

intenciones de los usuarios. Por ello, podemos dividir este escenario en tres procesos que se llevan a cabo de forma secuencial.

- 2.1) **Extracción de las características** que estén correlacionadas de alguna manera con las intenciones de los usuarios.
 - 2.2) **Selección de las características** más relevantes.
 - 2.3) **Clasificación de las características** a través de técnicas de *machine learning* para identificar las intenciones de los usuarios y transformarlas en comandos específicos de la aplicación en la que esté integrado el BCI.
- 3) **Aplicación:** el propósito de este escenario consiste en recibir y decodificar los comandos que provienen de la fase anterior con el objetivo de ofrecer una retroalimentación en tiempo real al usuario. Esta retroalimentación resulta fundamental para completar el ciclo, posibilitando que los usuarios reaccionen mediante la selección de nuevos comandos o la modulación de su propia actividad cerebral.

4.2.2. BCI basados en EEG

Desde una perspectiva práctica, los sistemas BCI destinados a la población general deben cumplir con una serie de requisitos para ser efectivos y accesibles. Entre estos requisitos se encuentra la no invasividad, es decir, que no se requiera de intervenciones quirúrgicas para su utilización, portátiles para facilitar su uso en diferentes entornos, económicos para aumentar su accesibilidad, confiables para garantizar resultados precisos y resistentes a diversos ambientes para mantener su funcionamiento óptimo en diferentes condiciones. De hecho, la mayor parte de las técnicas de adquisición de las señales cerebrales se relegan únicamente a tareas de investigación.

Por esta razón, se da prioridad a los BCI basados en el EEG en el presente Trabajo de Fin de Grado. El EEG es una técnica de monitoreo no invasiva que registra la actividad eléctrica del cerebro mediante la colocación de electrodos en el cuero cabelludo del usuario. Cada electrodo captura la actividad de millones de neuronas, lo que permite obtener señales que reflejan la actividad cerebral de forma externa (Soufneyestani et al., 2020; Wolpaw and Wolpaw, 2012a). Estas señales se adquieren utilizando un conjunto mínimo de electrodos, entre los que al menos se debe incluir los electrodos de tierra, referencia y los canales propiamente dichos y se procesan mediante amplificación, filtrado y conversión analógico-digital para su posterior análisis. El electrodo tierra se conecta a la tierra del amplificador para evitar interferencias del amplificador (Srinivasan, 2012). Es importante destacar que el EEG mide la diferencia de voltaje entre los electrodos de los canales y la referencia, que se suele colocar en una zona que no adquiera actividad neurológica, como puede ser el lóbulo de la oreja. El hardware filtra las señales de frecuencias menores de 0,5 Hz. Posteriormente, el amplificador potencia la señal, alrededor de $2 \cdot 10^4$ veces. Finalmente, es necesario tener en cuenta que las variaciones de potencial se originan debido a disparidades en las impedancias entre los electrodos, las cuales se mitigan mediante

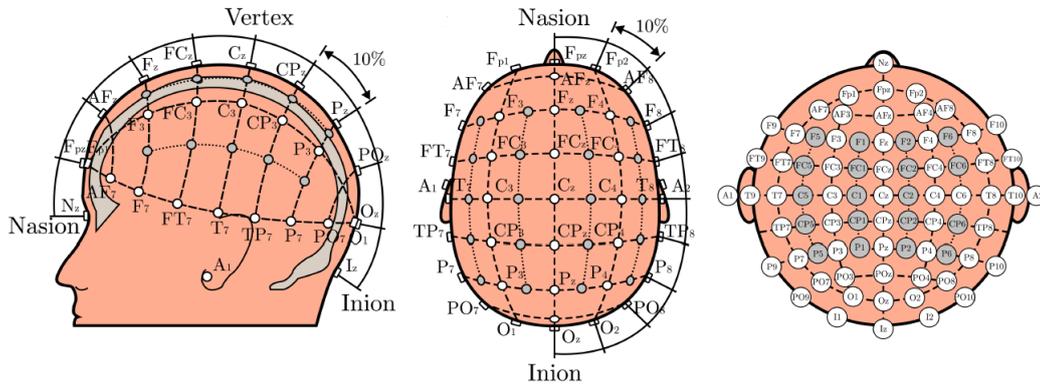


Figura 4.6: Vistas esquemáticas del estándar del Sistema Internacional 10-10 para EEG (Martínez-Cagigal, 2020).

la aplicación de un gel conductor entre los electrodos y la superficie del cuero cabelludo (Ferree et al., 2001).

La disposición de los electrodos canales no es arbitraria, sino que sigue una distribución establecida en diversos estándares. En particular, existen varios estándares internacionales como pueden ser los Sistemas Internacionales 10-20, 10-5 o 10-10 (Srinivasan, 2012). Durante la realización de las pruebas experimentales a lo largo de este Trabajo de Fin de Grado, se utilizará el estándar 10-10, que se caracteriza por la colocación de los electrodos de tal manera que cada punto de referencia se encuentra a una distancia del 10% entre el nasión (el punto anatómico entre los ojos) y el inión (la protuberancia ósea en la parte posterior del cráneo). Además, este estándar especifica la disposición de electrodos adicionales en puntos equidistantes entre los puntos de referencia principales, garantizando una distribución uniforme y reproducible de los electrodos en la superficie del cuero cabelludo (Seeck et al., 2017). Cada posición de estos puntos de referencia se designa con una nomenclatura específica, tal como se ilustra en la figura 4.6, utilizando uno o dos caracteres alfabéticos que reflejan la región de la corteza cerebral en la que se localiza cada punto. En concreto, la letra indica el lóbulo, y los números el hemisferio (números pares en el derecho, e impares en el izquierdo y la Z la línea central que une nasión e inión)

La alta frecuencia de muestro del analog-to-digital converter (ADC), permite una buena resolución temporal (entre 1 y 15 ms). No obstante, la capacidad de resolución espacial del EEG se encuentra restringida por la separación existente entre las fuentes corticales y la superficie del cuero cabelludo (Srinivasan, 2012). Efectivamente, la sensibilidad de los canales para registrar la actividad cerebral está intrínsecamente vinculada a la compleja coordinación de millones de neuronas, cuya eficacia se ve afectada por la distribución de giros y surcos. Este fenómeno, conocido como *efecto fuente*, conlleva la atenuación, distorsión y contaminación por ruido de una fuente cuando atraviesa la corteza cerebral, las

meninges y el cráneo (Wang, 2010). Asimismo, también afecta a la calidad de la señal el concepto de *conducción de volumen*, que describe la extensión de la actividad eléctrica al encontrarse con una superficie distinta, como es el caso del cuero cabelludo (Nunez et al., 1997).

Las señales recogidas por el EEG pueden ser descompuestas en cinco bandas o ritmos de frecuencia, como se ilustra en la figura 4.7.

- 1 Delta** (δ , 1-4 Hz): principalmente manifestada durante el sueño profundo en adultos, aunque también detectada en bebés despiertos, pero en regiones occipitales. Su exceso se vincula con lesiones subcorticales, así como con la hidrocefalia (Tatum et al., 2014).
- 2 Theta** (θ , 4-8 Hz): mayoritariamente presente en regiones frontales o frontocentrales, su aparición se asocia con estados de somnolencia, hiperventilación y estados emocionales. La aparición intermitente puede sugerir una leve encefalopatía difusa o hidrocefalia (Tatum et al., 2014).
- 3 Alpha** (α , 8-13 Hz): presente principalmente en regiones posteriores de la cabeza durante la relajación, aumenta al cerrar los ojos y disminuye al abrirlos o realizar esfuerzos mentales. La ausencia unilateral puede sugerir una anomalía en el lado correspondiente, mientras que la persistencia anormal puede indicar una lesión cerebral (Tatum et al., 2014).
- 4 Beta** (β , 13-30 Hz): ritmos de baja amplitud frontales vinculados con ansiedad, concentración activa, somnolencia o sueño ligero. La activación excesiva se relaciona con el consumo de ciertos medicamentos y patologías como el síndrome Dup15 (Tatum et al., 2014).
- 5 Gamma** (γ , > 30 Hz): relacionados con el procesamiento cognitivo elevado, surgen durante el procesamiento sensorial *crossmodal* y el emparejamiento de la memoria a corto plazo. Estudios recientes sugieren su asociación con interacciones neurotransmisoras excitatorias e inhibitorias (Tatum et al., 2014).

4.2.3. Señales de control

La principal función de un sistema BCI radica en la precisa detección de las intenciones del usuario en tiempo real, un proceso de naturaleza no trivial. Los procesos cognitivos complejos, como el pensamiento o la toma de decisiones, implican la interacción entre diversas áreas cerebrales. Además, la resolución espacial del EEG y su incapacidad para registrar de manera efectiva ciertos aspectos de la actividad neuronal podrían dificultar la focalización en regiones específicas del cerebro Wolpaw and Wolpaw (2012a). Por ende, las intenciones del usuario no quedan reflejadas directamente en las señales EEG, lo que confiere una importancia crucial a los paradigmas y al procesamiento de señales para su

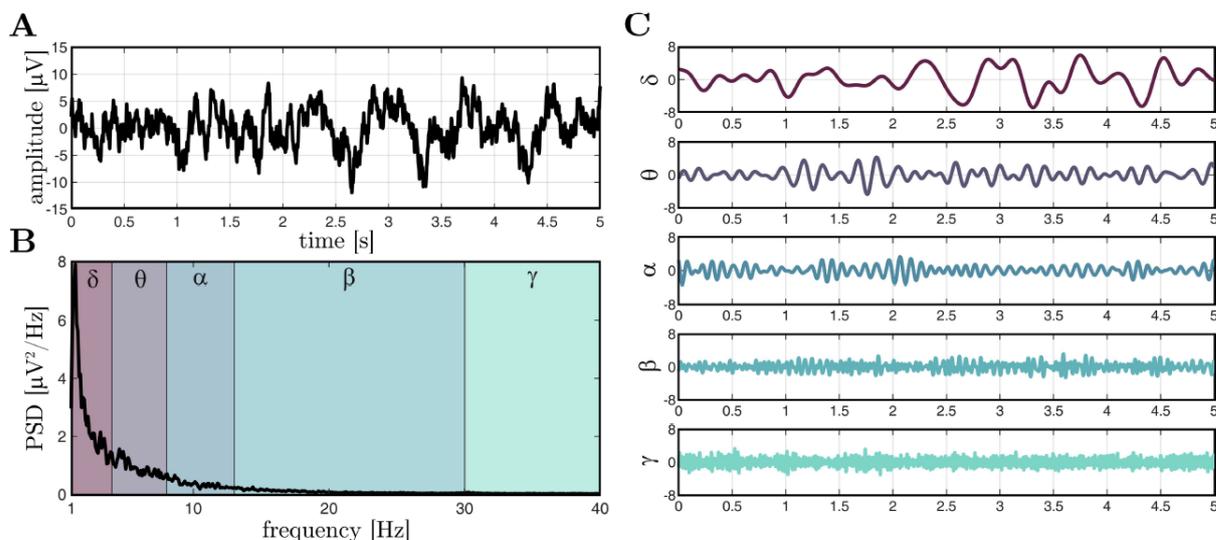


Figura 4.7: Oscilaciones neuronales presentes en las señales electroencefalográficas de un sujeto sano y consciente. **(A)** Segmento de EEG de 5 segundos del canal Pz; **(B)** densidad espectral de potencia (PSD) de la señal de EEG; **(C)** desglose del segmento de EEG en las cinco bandas principales: δ (1-4 Hz), θ (4-8 Hz), α (8-13 Hz), β (13-30 Hz) y γ (30-100 Hz) (Martínez-Cagigal, 2020).

identificación y conversión en comandos utilizables dentro de una aplicación específica (Wolpaw and Wolpaw, 2012a). Concretamente, los sistemas BCI basados en EEG dependen del procesamiento de cambios medibles asociados a tareas cognitivas, denominados señales de control, que se pueden clasificar en:

- **Señales exógenas** que se basan en el procesamiento de respuestas naturales del cerebro a estímulos externos, prescindiendo así de un entrenamiento extenso y requiriendo únicamente una breve fase de calibración (Wolpaw and Wolpaw, 2012a).
- **Señales endógenas** que se fundamentan en la capacidad del usuario para autorregular su actividad cerebral mediante entrenamiento de neurofeedback (NF). El NF implica la presentación en tiempo real de parámetros EEG específicos al usuario, acompañados de refuerzos positivos para favorecer el aprendizaje deseado (Kleber and Birbaumer, 2005). Sin embargo, esta fase de entrenamiento puede llegar a ser muy duradera en el tiempo dependiendo del usuario en cuestión, variando desde horas hasta semanas (Nicolas-Alonso and Gomez-Gil, 2012).

Las señales de control más empleadas en los sistemas BCI basados en EEG se detallan en los siguientes apartados.

Ritmos Sensoriomotores

En inglés, *Sensorimotor rhythms* (SMR), se definen como oscilaciones en las bandas de frecuencia μ y β registradas específicamente sobre la corteza sensoriomotora. Estas oscilaciones, moduladas por la actividad motora endógena, se caracterizan por manifestar

variaciones en el EEG que están sincronizadas temporalmente con el evento, pero no en fase (Pfurtscheller and Lopes Da Silva, 1999). Esta actividad motora endógena viene en respuesta a eventos sensoriomotores, ya sean reales o imaginados. Dichas variaciones se manifiestan mediante lo que se conoce como desincronización relacionada con el evento (ERD) o sincronización relacionada con el evento (ERS) (Pfurtscheller and Lopes Da Silva, 1999). Las variaciones de SMR se alcanzan mediante la práctica de la imaginación motora, haciendo hincapié en la repetición de vivencias kinestésicas en vez de representaciones visuales de movimientos (Nicolas-Alonso and Gomez-Gil, 2012). Estos fenómenos han suscitado un notable interés en la comunidad científica de sistemas BCI, debido a su potencial para permitir a los usuarios regular sus propios SMR mediante técnicas de entrenamiento de neurofeedback (NF), incluso en condiciones de limitación o ausencia de control motor efectivo (Pfurtscheller and McFarland, 2012).

Potenciales Relacionados con Eventos

En inglés, *Event-related Potential* (ERP), constituye una configuración de variaciones de voltaje que se manifiesta en respuesta a un estímulo externo, ya sea visual, auditivo o táctil (Sellers et al., 2012). En particular, cuando el ERP se sincroniza temporalmente con un estímulo visual, como un destello breve, la aparición repentina de imágenes o cambios abruptos de color, se denomina potencial evocado visual (VEP, por sus siglas en inglés). Los VEPs suelen manifestarse con mayor prominencia en áreas occipitales (Allison et al., 2012). No obstante, los ERPs se ven eclipsados por la actividad de fondo del EEG que no guarda relación con el estímulo, lo cual dificulta su identificación en el EEG sin procesar a simple vista. El promedio de muchas repeticiones del mismo ERP permite extraer patrones, anulando la actividad de fondo y fortaleciendo así la respuesta neuronal relacionada con la generación de dicho ERP (Luck, 2005). Estas señales de control se pueden subdividir en P300, SSVEP y c-VEPs.

- **P300:** los potenciales evocados P300 (o P3) constituyen un componente endógeno de gran interés dentro de los VEPs. Estos potenciales se caracterizan por ser deflexiones positivas que emergen en respuesta a estímulos infrecuentes y significativos, aproximadamente 300 ms después de su inicio (Wolpaw et al., 2002). Es importante destacar que la latencia de estos potenciales puede variar en un rango amplio, dependiendo de diversos factores como el filtrado espectral, las características individuales de los sujetos o la complejidad de la tarea cognitiva (Picton, 1992). La generación de las ondas P300 en un ERP sigue un paradigma conocido como *oddball*, el cual se basa en la presentación de estímulos objetivo poco frecuentes que deben ser detectados entre una serie de estímulos de fondo. En este contexto, la tarea del sujeto consiste en discriminar y clasificar estos eventos de manera adecuada, lo que proporciona información valiosa sobre los procesos cognitivos asociados con la detección de estímulos novedosos en el entorno.
- **SSVEP:** en los Potenciales Evocados Visuales de Estado Estable (SSVEPs, por sus siglas en inglés), los estímulos se caracterizan por ser una estimulación visual

repetitiva y rápida, lo que conlleva a una superposición de respuestas evocadas y la consiguiente generación de una oscilación constante a la misma frecuencia (Sellers et al., 2012). La identificación de los SSVEP se facilita mediante un análisis espectral simple, lo que garantiza altos niveles de precisión en la clasificación sin necesidad de un entrenamiento previo, debido a su naturaleza exógena. La configuración típica de un sistema BCI basado en SSVEP implica la implementación de un conjunto de comandos o caracteres que parpadean a diferentes frecuencias, lo que permite que los usuarios puedan seleccionar un comando específico al prestar atención a ese parpadeo, lo que genera un SSVEP en el espectro del EEG a la frecuencia del comando observado. Es importante destacar que las tasas de parpadeo más confiables se encuentran en la banda β baja (13-19 Hz), lo que puede incrementar la fatiga visual y el riesgo de convulsiones epilépticas fotosensibles (Pastor et al., 2003). Además, el número de comandos está limitado por la tasa de actualización vertical estandarizada de las pantallas LCD actuales (e.g., para una pantalla de 60 Hz no se pueden emplear SSVEP a más de 30 Hz) (Volosyak et al., 2009).

- **c-VEPs:** los sistemas BCI basados en Potenciales Evocados Visuales Modulados por Código (c-VEPs) utilizan secuencias/códigos temporales pseudoaleatorios para codificar comandos individuales (Martínez-Cagigal et al., 2021). Este enfoque permite que la estimulación de banda ancha genere respuestas amplias, lo que facilita su decodificación incluso en presencia de interferencias de banda estrecha. Aunque es posible modular cada comando con un código único, la identificación de un conjunto de códigos con propiedades de correlación cruzada lo suficientemente buenas no es una tarea sencilla y requiere de un estudio previo (Thielen et al., 2021). Por ello, este Trabajo de Fin de Grado presenta y compara dos formas de obtener este conjunto de códigos pseudoaleatorios con propiedades adecuadas a través de (1) desplazamientos circulares distintos de cero, codificando cada comando con versiones retardadas en el tiempo de la secuencia original (Bin et al., 2009) y (2) Gold Codes y sus versiones desplazadas (Sun et al., 2022).

El capítulo 5 abordará en detalle la señal de control c-VEP, así como los códigos pseudoaleatorios asociados y sus características, junto con otros elementos esenciales para la implementación exitosa de un sistema BCI basado en c-VEP.

Capítulo 5

El paradigma c-VEP

En este capítulo se exploran diversos aspectos relacionados con la generación de c-VEPs. Se abordan temas clave como la generación de números pseudoaleatorios a través de técnicas como LFSR y m-secuencias, así como la aplicación de Gold Codes para este propósito. Además, se examinan métodos de extracción de filtros espaciales como CCA y TRCA. Se presentan también diferentes pipelines de procesamiento, incluyendo técnicas como el *Circular Shifting* y su combinación con los Gold Codes. Por último, se discuten aspectos relevantes como los ratios de presentación, *raster latencies* y la consideración de vecinos equivalentes en el contexto de este tipo de investigaciones.

5.1. Bases metodológicas

En el transcurso de las últimas décadas, la comunidad científica ha avanzado en la formulación de estrategias innovadoras para la mejora de los sistemas BCI no invasivos y de fácil instalación, tales como la reducción del tiempo de calibración, la implementación de sistemas asíncronos, el desarrollo de sistemas adaptativos, así como la exploración de variaciones en los paradigmas conocidos (Martínez-Cagigal et al., 2021). De esta manera, surgió un creciente interés en la comunidad científica por una señal de control innovadora, destacándose por su capacidad de lograr rendimientos sobresalientes y tiempos de calibración reducidos: los c-VEPs. Bajo este paradigma, los comandos se presentan a través de parpadeos siguiendo secuencias temporales pseudoaleatorias, desencadenando respuestas de EEG que exhiben una mayor correlación con el comando al cual el usuario está dirigiendo su atención, en comparación con el resto de estímulos (Bin et al., 2009).

A pesar de que este enfoque se originó a raíz de la investigación de Sutter (1984), quien empleó un sistema basado en ECoG, no fue sino hasta el año 2009, con el trabajo de Bin et al. (2009), que se evidenció su eficacia a través de la utilización de EEG. Este investigador logró avances significativos, superando de esta manera al paradigma clásico SSVEP, que predominaba hasta entonces.

Como se adelantó en la sección 4.2.3, en el paradigma c-VEP los diversos comandos

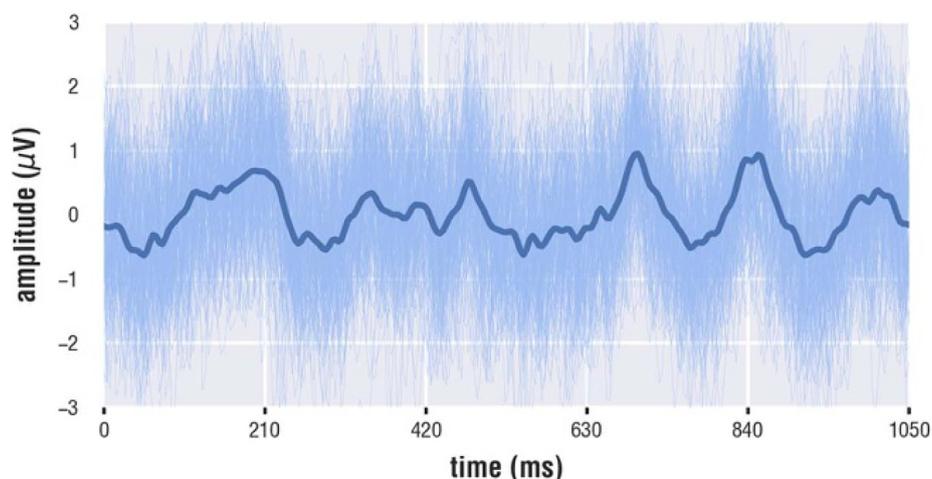


Figura 5.1: Plantilla principal, obtenida como la promediación de la respuesta tomada por el EEG en el canal Oz, sobre 150 ciclos de calibración (Martínez-Cagigal et al., 2021).

disponibles para ser seleccionados por parte del usuario, en tiempo real, tales como letras, números u otros caracteres, se presentan como una fuente parpadeante que sigue una secuencia pseudoaleatoria. El usuario, que debe dirigir su atención hacia el comando que desea seleccionar en cada momento, inducirá una respuesta en su EEG que está condicionada por la secuencia intermitente específica del comando en cuestión. En la figura 5.1 se puede observar la respuesta promediada de un sujeto provocada por el paradigma c-VEP.

5.2. Generación de secuencias pseudoaleatorias

La esencia de los c-VEPs se fundamenta en secuencias binarias, compuestas por 0 y 1, que obedecen a un patrón pseudoaleatorio. Por consiguiente, la elección de tales códigos no es trivial, dado que deben poseer propiedades de correlación adecuadas para asegurar el correcto funcionamiento del sistema BCI. Así, cada comando se codifica mediante una secuencia de bits, donde el parpadeo sigue esta secuencia: negro para 0 y blanco para 1. La velocidad de parpadeo está determinada por la frecuencia de actualización de la pantalla en la que se presentan los estímulos. Por consiguiente, en cada fotograma, el comando adquiere su coloración en consonancia con la secuencia del código pseudoaleatorio, repitiendo este proceso de forma secuencial hasta completar la totalidad de la secuencia (Martínez-Cagigal et al., 2021).

Formalmente, las secuencias pseudoaleatorias (o PN, por sus siglas en inglés *pseudo-random noise*) son secuencias de 1s y 0s que pueden ser generadas por reglas matemáticas, ya que estadísticamente cumple con los requisitos de una secuencia verdaderamente aleatoria en un sentido amplio. Estas secuencias PN, según Sklar (2001), cumplen con tres propiedades fundamentales:

- **Propiedad de balance:** un buen balance asegura que el número de 1s difiere a lo sumo en una unidad con el número de 0s.
- **Propiedad de *run*:** en las secuencias PN, una *run* hace referencia a cualquier subsecuencia contigua de bits que tienen el mismo valor, ya sea 0 o 1. En una secuencia PN, la mitad de las *run* tienen longitud 1, una cuarta parte tienen longitud 2, una octava parte tienen longitud 3, una dieciseisava parte tienen longitud 4, y así sucesivamente, siempre y cuando estas fracciones generen números enteros de secuencias (en cada caso, el número de secuencias de 0 es igual al número de secuencias de 1).
- **Propiedad de autocorrelación:** mide cuánto difiere una secuencia PN, con cualquiera de sus versiones desfasadas. De esta manera, si el desfase τ es igual a 0 o a un múltiplo del periodo de la secuencia, la autocorrelación será evidentemente 1.

Por consiguiente, en la elaboración de este Trabajo de Fin de Grado se han empleado dos métodos para la generación de bits pseudoaleatorios, los cuales se detallan a continuación.

5.2.1. LFSR y m-secuencias

Un registro de desplazamiento de retroalimentación lineal (LFSR, por sus siglas en inglés) de longitud $n \in \mathbb{N}$, está compuesto por n etapas (o elementos de retardo) numeradas de $0, \dots, n-1$, cada una con capacidad para almacenar un bit y equipada con una entrada y una salida; además, cuenta con un reloj que controla el flujo de datos. En cada unidad de tiempo, se llevan a cabo las siguientes operaciones (Menezes et al., 1996), que se pueden ver de forma gráfica en la figura 5.2:

1. El contenido de la etapa 0 se emite, y pasa a formar parte de la *secuencia de salida*.
2. El contenido de la etapa i se desplaza a la etapa $i-1$ para cada i tal que $1 \leq i \leq n-1$.
3. El contenido actualizado de la etapa $n-1$ es el *bit de retroalimentación* s_j , obtenido al sumar módulo 2 los contenidos previos de un conjunto específico de etapas, desde 0 hasta $n-1$.

El LFSR de la figura 5.2, se denota por $[n, P(X)]$, donde $P(X) = 1 + c_1X + c_2X + \dots + c_nX \in \mathbb{Z}_2[X]$ es el *polinomio de conexión*. En la práctica, se supone siempre que $P(X)$ es de grado n , es decir, $c_n = 1$, y por tanto la secuencia de salida, es periódica de periodo p . Si el contenido inicial de la etapa i es $s_i \in \{0, 1\}$ para cada i con $0 \leq i \leq n-1$, entonces $[s_{n-1}, \dots, s_1, s_0]$ se denomina *estado inicial* (o semilla) del LFSR (Menezes et al., 1996).

De hecho, si el estado inicial del LFSR es $[s_{n-1}, \dots, s_1, s_0]$, entonces la secuencia de salida $s = s_0s_1s_2\dots$ viene determinada por la relación de recurrencia siguiente (Tausworthe, 1965)

$$s_j = (c_1s_{j-1} + c_2s_{j-2} + \dots + c_ns_{j-n}) \pmod{2} \quad \text{para } j \geq n$$

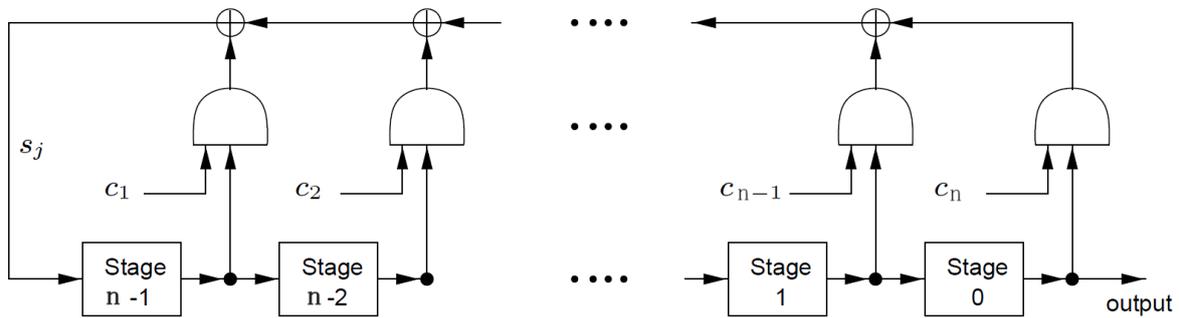


Figura 5.2: Esquema general LFSR. Cada $c_j \in \mathbb{Z}_2$, es decir, o bien es 0, o bien es 1; los semicírculos cerrados son compuertas AND; y el bit de retroalimentación s_j es la suma módulo 2 de los contenidos de aquellas etapas i , tal que $0 \leq i \leq n - 1$, para las cuales $c_{n-i} = 1$. Imagen adaptada de [Menezes et al. \(1996\)](#).

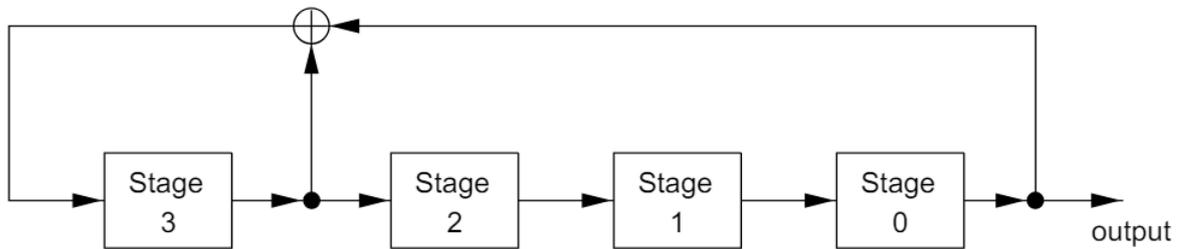


Figura 5.3: Esquema del LFSR definido por $[4, 1 + X + X^4]$ ([Menezes et al., 1996](#)).

Nótese que, por una sencilla cuestión de combinatoria, hay $2^n - 1$ posibles estados iniciales que no sean idénticamente nulos. La condición necesaria y suficiente para que cada una de estas secuencias de salida tenga periodo $p = 2^n - 1$ es que el polinomio de conexión $P(X)$ sea primitivo sobre el cuerpo \mathbb{Z}_2 ([Zierler, 1959](#); [Golomb, 1955](#)). De esta manera, si $P(X) \in \mathbb{Z}_2[X]$ es un polinomio primitivo de grado n , entonces $[n, P(X)]$ es un *LFSR de longitud máxima* y las secuencias de salidas producidas a partir de estados iniciales no nulos se denominan *maximum-length sequence* (MLS) o *m-secuencias*. Evidentemente, en la práctica, las m-secuencias van a tener una longitud de $2^n - 1$ bits ([Menezes et al., 1996](#)).

Para ejemplificar todo esto, tómesese el LFSR dado por $[4, 1 + X + X^4]$ que se muestra en la figura 5.3. A partir de la columna X_0 de la tabla 5.1, se obtiene la secuencia de salida $s = 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, \dots$, y es periódica con período 15, pues la fila correspondiente al tiempo $t = 0$, coincide con la fila de $t = 15$, y además $1 + X + X^4$ es primitivo en \mathbb{Z}_2 .

Aunque, teóricamente, cada comando podría ser codificado por un código pseudoaleatorio único, en la aplicación práctica se utiliza una sola secuencia, pero se aplican

Tabla 5.1: Tabla de valores del LFSR dado por $[4, 1 + X + X^4]$.

t	X_3	X_2	X_1	X_0
0	0	1	1	0
1	0	0	1	1
2	1	0	0	1
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	1	0	0	0
7	1	1	0	0
8	1	1	1	0
9	1	1	1	1
10	0	1	1	1
11	1	0	1	1
12	0	1	0	1
13	1	0	1	0
14	1	1	0	1
15	0	1	1	0

diferentes desplazamientos temporales para cada comando. Esto se hace porque la señal de respuesta EEG asociada estará más estrechamente relacionada con el comando deseado que con otros, lo que permite al sistema BCI determinar la dirección de la atención del usuario mediante un análisis de correlación entre la respuesta EEG registrada y todas las posibles variaciones temporales de la secuencia empleada. Esto se debe a que, para codificar los comandos, es necesario que sus correspondientes códigos tengan correlaciones cruzadas bajas, lo que resulta crucial para evitar interferencias entre las señales EEG. Por consiguiente, el enfoque convencional se centra en la identificación de una secuencia binaria que exhiba niveles reducidos de autocorrelación para desplazamientos circulares distintos de cero, como pueden ser las m-secuencias. Posteriormente, cada comando se codifica utilizando versiones desplazadas temporalmente de esta secuencia original. La principal ventaja del desplazamiento circular es que la etapa de calibración se reduce considerablemente ya que solo es necesario registrar la respuesta cerebral ante la secuencia original (Martínez-Cagigal et al., 2021). En la figura 5.4, obtenida a partir de la aplicación MEDUSA[©], se puede observar un ejemplo práctico de estos desplazamientos y de las excelentes propiedades de autocorrelación. De hecho, el conjunto de m-secuencias desplazadas posee propiedades de aleatoriedad sólidas, entre las cuales se encuentra una función

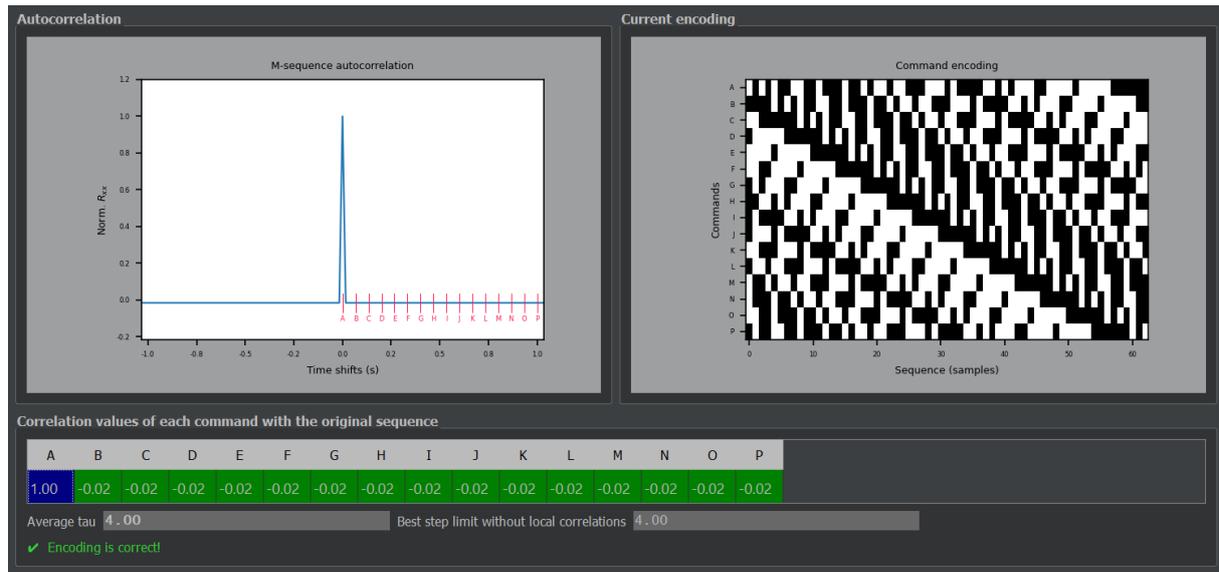


Figura 5.4: Resumen de la codificación de 16 comandos a través del uso de la m-secuencia de 63 bits obtenida del LFSR dado por $[6, 1 + X^5 + X^6]$ y el estado inicial $[1, 1, 1, 1, 1, 0]$, utilizando un desfase de $\tau = 4$. En la parte de la izquierda y en la inferior se observan los valores de la correlación de cada uno de los comandos con la secuencia sin desplazar. En la parte de la derecha, se ve gráficamente la codificación de cada comando, cómo se desplaza un numero entero de τ con respecto a la m-secuencia original, que codifica el carácter A.

de autocorrelación de dos valores, dependiendo del desplazamiento τ (Mitra, 2008):

$$\mathbf{r}(\tau) = \begin{cases} 1, & \text{si } \tau \text{ es múltiplo de } N, \\ -\frac{1}{N}, & \text{en otros casos.} \end{cases}$$

5.2.2. Gold Codes

Las secuencias Gold, o Gold Codes, son aquellas secuencias PN generadas por la operación módulo-2 de dos m-secuencias de la misma longitud (Mitra, 2008), como se aprecia en la figura 5.5. De hecho, para tener las propiedades óptimas de autocorrelación y correlación cruzada entre las secuencias Gold, es necesario escoger dos m-secuencias específicas, que se denominarán par preferido de m-secuencias (Dinan and Jabbari, 1998). Dado que ambas m-secuencias tienen la misma longitud N , el Gold Code que se genera tendrá también la misma longitud. A partir de un par preferido de m-secuencias, el conjunto de Gold Codes se genera mediante la suma módulo-2 de la primera m-secuencia con versiones desplazadas de la segunda, o viceversa. De hecho, si tienen longitud $N = 2^n - 1$, hay N posibles desplazamientos circulares. Así, se pueden obtener N secuencias Gold a partir de un par preferido de m-secuencias (Sklar, 2001).

Formalmente, Mitra (2008) hace el siguiente planteamiento: se parte de dos m-secuencias, es decir, a partir de dos vectores binarios u y v de tamaño $N = 2^n - 1$ que supongamos

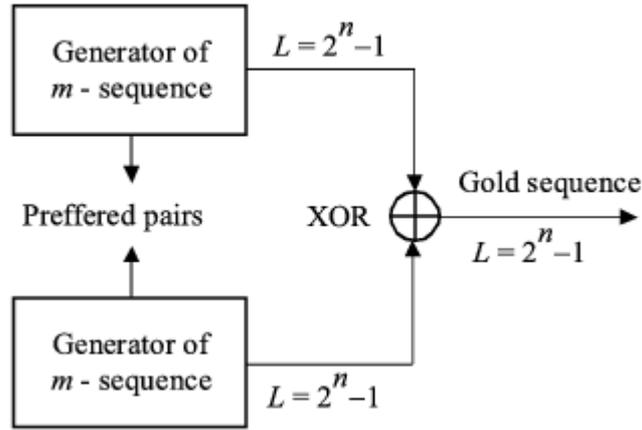


Figura 5.5: Esquema de generador de Gold Codes (Šajjić et al., 2013).

que forman una par preferido. Denotemos por $v(\tau)$ a la secuencia v desplazada un número entero τ , y \oplus a la operación suma módulo-2 (o dicho de otra manera, la operación XOR). Entonces, el conjunto de posibles códigos Gold que se obtiene es:

$$\mathbf{G}(u, v) = \{u, v, u \oplus v, u \oplus v(1), u \oplus v(2), \dots, u \oplus v(N-1), u \oplus v(N)\}.$$

Las propiedades de autocorrelación de las secuencias de Gold no son tan eficientes como las de las m-secuencias. A excepción de las dos secuencias originales, las demás no son m-secuencias, lo que implica que la autocorrelación no se limita a dos valores. El conjunto de secuencias de Gold exhibe un espectro de autocorrelación de tres valores. Estas secuencias destacan por sus excelentes propiedades de correlación cruzada. En particular, si llamamos $\mathbf{r}_{xx}(\tau)$ a la autocorrelación y $\mathbf{r}_{xy}(\tau)$ a la correlación cruzada del conjunto $\mathbf{G}(u, v)$ entonces

$$\mathbf{r}_{xx}(\tau) = \begin{cases} 1, & \text{si } \tau = 0, \\ \left\{ -\frac{t(n)}{N}, -\frac{1}{N}, \frac{t(n)-2}{N} \right\}, & \text{si } \tau \neq 0. \end{cases}$$

$$\mathbf{r}_{xy}(\tau) \in \left\{ -\frac{t(n)}{N}, -\frac{1}{N}, \frac{t(n)-2}{N} \right\},$$

donde

$$t(n) = \begin{cases} 1 + 2^{\frac{n+1}{2}} & \text{para } n \text{ impar,} \\ 1 + 2^{\frac{n+2}{2}} & \text{para } n \text{ par.} \end{cases}$$

Es necesario recalcar que si el conjunto de Gold Codes no se obtiene a partir de un par preferido, no tendrán estas características de correlación y por tanto, su uso en sistemas BCI basados en c-VEP disminuirá la precisión. La tabla 5.2 muestra dichos pares de parejas preferibles de polinomios generadores de m-secuencias en función de su grado.

Tabla 5.2: Parejas de polinomios preferidos para generar secuencias de Gold Codes. El grado del polinomio está representado por n y la longitud de las secuencias por N . Tabla adaptada de [Sharma et al. \(2014\)](#).

n	N	Polinomio preferido 1	Polinomio preferido 2
5	31	$X^5 + X^2 + 1$	$X^5 + X^4 + X^3 + X^2 + 1$
6	63	$X^6 + X + 1$	$X^6 + X^5 + X^2 + X + 1$
7	127	$X^7 + X^3 + 1$	$X^7 + X^3 + X^2 + X + 1$
9	511	$X^9 + X^4 + 1$	$X^9 + X^6 + X^4 + X^3 + 1$
10	1023	$X^{10} + X^3 + 1$	$X^{10} + X^8 + X^3 + X^2 + 1$
11	2047	$X^{11} + X^2 + 1$	$X^{11} + X^8 + X^5 + X^2 + 1$

Siguiendo la misma notación de antes, si se parte de una secuencia binaria $u = u_1u_2u_3\dots$ de periodo $N = 2^n - 1$ y se considera $v = u_{1q}u_{2q}u_{3q}\dots$, donde $q = 2^k + 1$ o $q = 2^{2k} - 2^k + 1$ para algún $k \in \mathbb{N}$. Se dice entonces que u y v forman un par preferido de m-secuencias si $n/\gcd(n, k)$ es impar, donde \gcd es el máximo común divisor ([Sarwate and Pursley, 1980](#)).

Para utilizar los códigos Gold en c-VEP, cada comando a seleccionar debe estar codificado con un Gold Code diferente. Antes de empezar a controlar el sistema, es necesario registrar la respuesta cerebral a todos los comandos codificados. Esto permite asociar cada respuesta específica con su correspondiente comando.

5.3. Extracción de filtros espaciales

Los métodos de neuroimagen y procesamiento de señales están experimentando un rápido avance con el objetivo final de lograr una alta resolución tanto en tiempo como en espacio. El EEG ofrece una excelente resolución temporal, es no invasiva y fácil de utilizar. Sin embargo, las señales observadas en el cuero cabelludo son difíciles de interpretar, ya que provienen de una mezcla de múltiples fuentes corticales y subcorticales que atraviesan diversas capas del cerebro, cada una con diferentes propiedades de conducción de volumen, incluyendo el cuero cabelludo, el cráneo, el líquido cefalorraquídeo y los tejidos cerebrales. Los sensores captan señales mixtas contaminadas por artefactos provenientes de varias estructuras cerebrales activas. Por lo tanto, el análisis directo de EEG a nivel del cuero cabelludo no puede reflejar la dinámica neuronal subyacente ([Rykczewski et al., 2021](#)).

En este documento se presentan dos métodos de análisis multivariante, CCA y TRCA, para ponderar la importancia de los canales a la hora de reflejar las respuestas cerebrales causadas por las iluminaciones de los c-VEP. En concreto, como esta información visual se procesa en el córtex visual primario, se espera que los filtros den más importancia a la

zona occipital, ofreciendo así información de gran valor.

5.3.1. Canonical Correlation Analysis

El Análisis de la Correlación Canónica (CCA, por sus siglas en inglés), originalmente propuesto por [Hotelling \(1935\)](#), es una generalización del coeficiente de correlación de momento del producto de Karl Pearson ([Dattalo, 2014](#)). Las estructuras de datos multivariantes complejas, como puede ser los datos obtenidos a partir del EEG, se vuelven más comprensibles cuando las estudiamos a través de proyecciones de baja dimensionalidad. En un estudio que involucra dos conjuntos de datos, nos preguntamos qué tipo de proyección de baja dimensionalidad nos ayuda a descubrir posibles estructuras compartidas entre las dos muestras. CCA es una herramienta estándar en estadística multivariante que permite descubrir y cuantificar las relaciones entre dos conjuntos de variables ([Härdle and Simar, 2019](#)).

Esta técnica está basada en proyecciones. Se crea un índice (una variable proyectada) que se correlaciona al máximo con el índice de la otra variable, considerando cada muestra por separado. El objetivo de CCA es maximizar la asociación (medida por la correlación) entre las proyecciones de baja dimensionalidad de los dos conjuntos de datos. Encontramos los vectores de correlación canónica mediante un análisis conjunto de la covarianza de las dos variables ([Hancock et al., 2019](#)). Formalmente, CCA se puede formular de la siguiente manera:

Supongamos que partimos de dos variables aleatorias $X \in \mathbb{R}^q$ y $Y \in \mathbb{R}^p$. Entonces CCA trata de buscar los vectores a y b tales que la relación entre las proyecciones $a^T X$ y $b^T Y$ se pueda cuantificar de alguna manera. En particular, se trata de encontrar a y b que maximicen la correlación $\rho(a, b) = \rho_{a^T X b^T Y}$. Si consideremos las matrices de covarianza

$$\begin{aligned} \text{Var}(X) &= \Sigma_{XX}, \\ \text{Var}(Y) &= \Sigma_{YY}, \\ \text{Cov}(X, Y) &= \Sigma_{XY} = \Sigma_{YX}^T, \end{aligned}$$

entonces, se demuestra que

$$\rho(a, b) = \frac{a^T \Sigma_{XY} b}{(a^T \Sigma_{XX} a)^{1/2} (b^T \Sigma_{YY} b)^{1/2}} \quad (\text{Härdle and Simar, 2019}). \quad (5.1)$$

Por lo que basta encontrar a y b que maximicen (5.1). Nótese que hay diferentes variaciones de CCA, sin embargo, todas se basan en la formulación aquí presentada.

Computación del algoritmo CCA

Una vez realizado el análisis teórico sobre este método, es necesario conocer cómo computarlo. A continuación, se muestran los fundamentos en los que se basa la implementación de CCA en MEDUSA[©], adaptada de la implementación hecha en MATLAB

(*canoncorr*) basada en [Everitt and Krzanowski \(1989\)](#) y [Cooper and Seber \(1985\)](#).

El algoritmo calcula el análisis de correlación canónica para las matrices de datos X (con dimensiones $N \times p_1$) e Y (con dimensiones $N \times p_2$). Ambas matrices deben tener el mismo número de observaciones (filas), pero pueden tener una cantidad de variables distinta (columnas). Las columnas j -ésimas de las matrices A y B que se devuelven contienen los coeficientes canónicos, que son las combinaciones lineales de las variables que constituyen la j -ésima variable canónica para X e Y , respectivamente.

Paso 1: es necesario normalizar las matrices X e Y por columnas.

Paso 2: se computa la factorización QR de las matrices X e Y , es decir,

$$\begin{aligned} X &= Q_1 T_{11}, \\ Y &= Q_2 T_{22}, \end{aligned}$$

donde Q_1 y Q_2 son dos matrices ortogonales de dimensiones $N \times k_1$ y $N \times k_2$, respectivamente, y T_{11} y T_{22} son dos matrices triangulares superiores de dimensiones $k_1 \times p_1$ y $k_2 \times p_2$, respectivamente, donde $k_1 = \min\{N, p_1\}$ y $k_2 = \min\{N, p_2\}$. Recuérdese que se dice que Q es una matriz ortogonal si y solamente si $QQ^T = I$, donde I es la matriz identidad.

Paso 3: se computa la descomposición en valores singulares de la matriz $Q_1^T Q_2$, es decir,

$$Q_1^T Q_2 = LDM,$$

donde L y M son dos matrices ortogonales de dimensiones $k_1 \times k_1$ y $k_1 \times k_2$, respectivamente, y D es una matriz diagonal de dimensiones $k_1 \times k_2$.

Paso 4: Se define la matriz de salida A , expresada por columnas, de la siguiente manera:

$$A = [a_1^T; a_2^T; \dots; a_{k_1}^T],$$

donde cada a_j es la solución de mínimos cuadrados del problema $T_{11}a_j = l_j$, siendo l_j la columna j -ésima de L , para cada $j = 1, \dots, k_1$. Análogamente, se define B , también por columnas, de la manera siguiente

$$B = [b_1^T; b_2^T; \dots; b_{k_2}^T],$$

donde cada b_i es la solución de mínimos cuadrados del problema $T_{22}b_i = m_i$, siendo m_i la fila i -ésima de L , para cada $j = 1, \dots, k_2$. Con A y B así construidos, se tiene que $T_{11}A \approx L$ y $T_{22}B \approx M^T$. Teniendo en cuenta esto y los pasos anteriores, se tiene que:

$$\frac{A^T X^T Y B}{\sqrt{A^T X^T X A} \sqrt{B^T Y^T Y B}} = \frac{A^T T_{11}^T Q_1^T Q_2 T_{22} B}{\sqrt{A^T T_{11}^T T_{11}^T A} \sqrt{B^T T_{22}^T T_{22}^T B}} \approx \frac{L^T L D M M^T}{L^T L M M^T} \approx D.$$

De aquí se deduce el fundamento y la explicación de la construcción de A y B . Nótese que el filtro espacial que se busca coincide con la primera columna de B . Finalmente, es

necesario mencionar que el cómputo de CCA descrito es una simplificación, con el objetivo de facilitar su comprensión al lector, ya que, en la implementación real, es necesario tener en cuenta los rangos de las matrices X e Y . Normalmente, estas dos matrices en el contexto del EEG aplicado al paradigma c-VEP, representan (1) la señal del EEG por número de muestras (fila) y número de canales (columna) y (2) la misma matriz pero promediada por épocas y después replicada para igualar las dimensiones de la matriz anterior.

5.3.2. Task-Related Component Analysis

Los datos de EEG del cuero cabelludo registrados con un solo electrodo suelen considerarse como la simple suma de múltiples series temporales provenientes de diferentes fuentes corticales. En otras palabras, las señales de las fuentes corticales pueden reconstruirse mediante la suma lineal ponderada de las señales de EEG del cuero cabelludo de varios electrodos. Basándose en esta premisa, se han empleado varios métodos de filtrado espacial para extraer las actividades relacionadas con la tarea de las fuentes cerebrales a partir de los registros del cuero cabelludo (Nakanishi et al., 2018).

Tanaka et al. (2013) presenta un nuevo enfoque analítico para identificar componentes relacionadas con la tarea a partir de la combinación ponderada lineal de múltiples series temporales, como puede ser un EEG. Este método, que denominamos Análisis de Componentes Relacionadas con la Tarea (TRCA, por sus siglas en inglés), determina los coeficientes de las series temporales de manera que maximizan la covarianza o correlación entre bloques de tarea, asegurando así una alta consistencia entre ellos. Esta aproximación se basa en la suposición de que una señal, que se presenta de forma consistente y sólida en cada bloque de tareas, debe considerarse relevante para la tarea en cuestión (Tanaka et al., 2013). En resumen, como trata de mostrar la figura 5.6, TRCA es el método encargado de extraer las componentes relacionadas con la tarea de forma eficiente maximizando la reproducibilidad durante el periodo de la tarea (Nakanishi et al., 2018; Tanaka et al., 2014).

Si concretamos un poco más, siguiendo como modelo el texto de Nakanishi et al. (2018) y Tanaka et al. (2013), supongamos que partimos de $x_i(t)$, con $1 \leq i \leq N_c$, señales temporales que provienen de N_c canales, por ejemplo, de un EEG. Cada una de estas señales posee N_t bloques de una misma tarea repetida, con periodos fijos dentro del intervalo $t \in [t_h, t_h + T]$, con $1 \leq h \leq N_t$, donde T es la duración de cada bloque de tareas. Se supone también que cada una de las señales temporales es combinación lineal de otras dos señales:

1. una señal relacionada con la tarea, $s(t) \in \mathbb{R}$,
2. una señal no relacionada con la tarea (ruido), $n(t) \in \mathbb{R}$.

De esta manera, se asume que las señales temporales $x_i(t)$ siguen un modelo lineal generativo:

$$x_i(t) = a_{1,i}s(t) + a_{2,i}n(t), \quad i = 1, 2, \dots, N_c,$$

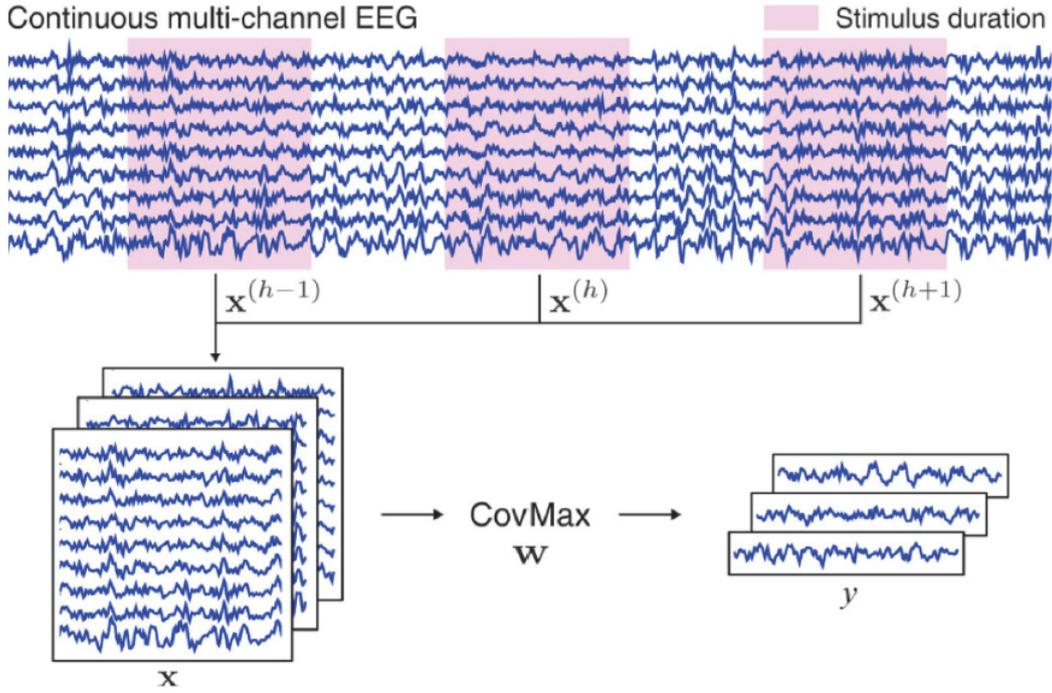


Figura 5.6: Diagrama de TRCA en su uso para EEG (Nakanishi et al., 2018).

donde i es el índice del canal y $a_{1,i}$ y $a_{2,i}$ son coeficientes que permiten proyectar las señales de origen hacia la señal del EEG. De esta manera, el objetivo de TRCA es recuperar la señal $s(t)$ a partir de una combinación lineal de señales observadas $x_i(t)$ dada por:

$$y(t) = \sum_{i=1}^{N_c} w_i x_i(t) = \sum_{i=1}^{N_c} (w_i a_{1,i} s(t) + w_i a_{2,i} n(t)).$$

De forma utópica, el problema tiene una solución ideal dada por $\sum_{i=1}^{N_c} w_i a_{1,i} = 1$ y $\sum_{i=1}^{N_c} w_i a_{2,i} = 0$, es decir $y(t) = s(t)$. Este problema puede ser resuelto a partir de la maximización de la covarianza entre bloques de tareas (*inter-trial covariance maximization*). Las señales temporales restringidas al h -ésimo bloque de tareas se representa por $x_i^{(h)}(t)$ con $1 \leq i \leq N_c$ y $1 \leq h \leq N_t$. Análogamente, $y^{(h)}(t)$ representa la señal y restringida al h -ésimo bloque. Así, la covarianza entre $y(t)$ restringida al bloque de tareas h_1 -ésimo y h_2 -ésimo se obtiene de la siguiente manera:

$$C_{h_1, h_2} = \text{Cov}\left(y^{(h_1)}(t), y^{(h_2)}(t)\right) = \sum_{i_1, i_2=1}^{N_c} w_{i_1} w_{i_2} \text{Cov}\left(x_{i_1}^{(h_1)}(t), x_{i_2}^{(h_2)}(t)\right).$$

Por consiguiente, la suma de todas las combinaciones de covarianzas entre bloques de

tareas distintos tiene la forma siguiente:

$$\sum_{\substack{h_1, h_2=1 \\ h_1 \neq h_2}}^{N_t} C_{h_1, h_2} = \sum_{\substack{h_1, h_2=1 \\ h_1 \neq h_2}}^{N_t} \text{Cov}\left(y^{(h_1)}(t), y^{(h_2)}(t)\right) = \sum_{\substack{h_1, h_2=1 \\ h_1 \neq h_2}}^{N_t} \sum_{i_1, i_2=1}^{N_c} w_{i_1} w_{i_2} \text{Cov}\left(x_{i_1}^{(h_1)}(t), x_{i_2}^{(h_2)}(t)\right). \quad (5.2)$$

Para expresar esta ecuación de forma matricial, es necesario definir $\mathbf{S} = (S_{i_1, i_2})_{1 \leq i_1, i_2 \leq N_c}$ dada por

$$S_{i_1, i_2} = \sum_{\substack{h_1, h_2=1 \\ h_1 \neq h_2}}^{N_t} \text{Cov}\left(x_{i_1}^{(h_1)}(t), x_{i_2}^{(h_2)}(t)\right),$$

y, por tanto, (5.2) será exactamente igual a $\mathbf{w}^T \mathbf{S} \mathbf{w}$, que llamaremos *consistencia de la tarea*. Nótese la simetría de la matriz \mathbf{S} . Con el objetivo de obtener una solución finita, Tanaka et al. (2013) asegura que es necesario imponer que la varianza de $y(t)$ sea igual a 1, es decir:

$$\text{Var}(y(t)) = \sum_{i_1, i_2=1}^{N_c} w_{i_1} w_{i_2} \text{Cov}\left(x_{i_1}^{(h_1)}(t), x_{i_2}^{(h_2)}(t)\right) = \mathbf{w}^T \mathbf{Q} \mathbf{w} = 1,$$

donde $\mathbf{Q} = (Q_{i_1, i_2})_{1 \leq i_1, i_2 \leq N_c}$ es la matriz definida por $Q_{i_1, i_2} = \text{Cov}\left(x_{i_1}^{(h_1)}(t), x_{i_2}^{(h_2)}(t)\right)$. Así, el problema de optimización con restricciones se convierte en un problema de autovalores de Rayleigh-Ritz dado por

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{S} \mathbf{w}}{\mathbf{w}^T \mathbf{Q} \mathbf{w}}.$$

El Teorema de Rayleigh-Ritz (Horn and Johnson, 2012), asegura que el vector de coeficientes óptimos se obtiene como un autovector de la matriz $\mathbf{Q}^{-1} \mathbf{S}$ cuyo autovalor asociado sea el máximo.

Computación del algoritmo TRCA

Uno de los objetivos específicos de este Trabajo de Fin de Grado es la implementación y evaluación del rendimiento de este algoritmo. Por ello, la computación e implementación llevada a cabo son fundamentales. A continuación, se muestran los pasos seguidos para implementar este algoritmo de la forma más óptima posible siguiendo como referencia el texto de Tanaka (2020).

Esta función asume que recibe como parámetros de entrada una matriz \mathbf{X} de dimensiones $N_c \times (\tau N_t)$, donde N_c representa el número de canales, τ la duración de cada bloque de tareas en cuestión y N_t el número de bloques de tareas. Siguiendo una notación similar a la explicación teórica de TRCA, se denotará por $\mathbf{X}^{(h)}$ a la matriz que se obtiene al restringir las columnas de \mathbf{X} de la $[(h-1)\tau + 1]$ -ésima, hasta la h -ésima. Nótese que, cada una de estas submatrices $\mathbf{X}^{(h)}$, tienen dimensiones $N_c \times \tau$ y se corresponden

con el EEG restringido al intervalo de tiempo en el que se produce el estímulo en cuestión.

Paso 1: Normalizar la matriz \mathbf{X} por canales, es decir, hacer que cada fila de \mathbf{X} tenga media 0 y varianza 1.

Paso 2: definición de la matriz \mathbf{S} . Se podría construir de la siguiente manera

$$\mathbf{S} = \frac{1}{N_t(N_t - 1)\tau} \sum_{k \neq h}^{N_t} \mathbf{X}^{(k)} \mathbf{X}^{(h)T}.$$

Sin embargo, esta definición de \mathbf{S} contiene $N_t(N_t - 1)$ sumandos. Otra forma equivalente y más óptima de definir esta misma matriz es la siguiente:

$$\mathbf{S} = \frac{1}{N_t(N_t - 1)\tau} \left(\mathbf{U}\mathbf{U}^T - \frac{1}{N_t}\mathbf{V} \right),$$

donde las matrices \mathbf{U} y \mathbf{V} son

$$\mathbf{U} = \frac{1}{N_t} \sum_{k=1}^{N_t} \mathbf{X}^{(k)},$$

$$\mathbf{V} = \frac{1}{N_t} \sum_{k=1}^{N_t} \mathbf{X}^{(k)} \mathbf{X}^{(k)T}.$$

Esta construcción involucra solo N_t sumandos lo que acelera el cálculo de \mathbf{S} , especialmente cuando N_t es grande.

Paso 3: definición de la matriz \mathbf{Q} de la siguiente manera:

$$\mathbf{Q} = \frac{1}{N_t\tau} \mathbf{X}\mathbf{X}^T$$

Paso 4: cálculo de los autovalores y autovectores de $\mathbf{Q}^{-1}\mathbf{S}$. Para esta operación, se valoró calcularla a través de métodos numéricos como puede ser el Método de la Potencia, sin embargo, no se adquirió la precisión adecuada en un número de iteraciones limitado por lo que se utilizó funciones ya programadas de cálculo de autovalores y autovectores. De hecho, el filtro espacial \mathbf{w} se corresponde con el autovector correspondiente al autovalor principal, es decir, aquel que tiene el mayor valor entre todos los autovalores, por lo que solo bastaría con computar dicho autovector y no todos.

5.4. Procesado de señal

Una vez introducidos los conceptos fundamentales sobre los que se sustentan los sistemas BCI basados en c-VEP, es necesario presentar la *pipeline*, o tubería de procesado de señal, seguida durante los experimentos realizados en este Trabajo de Fin de Grado. A lo largo del tiempo, se han propuesto diversas metodologías para potenciar el desempeño de

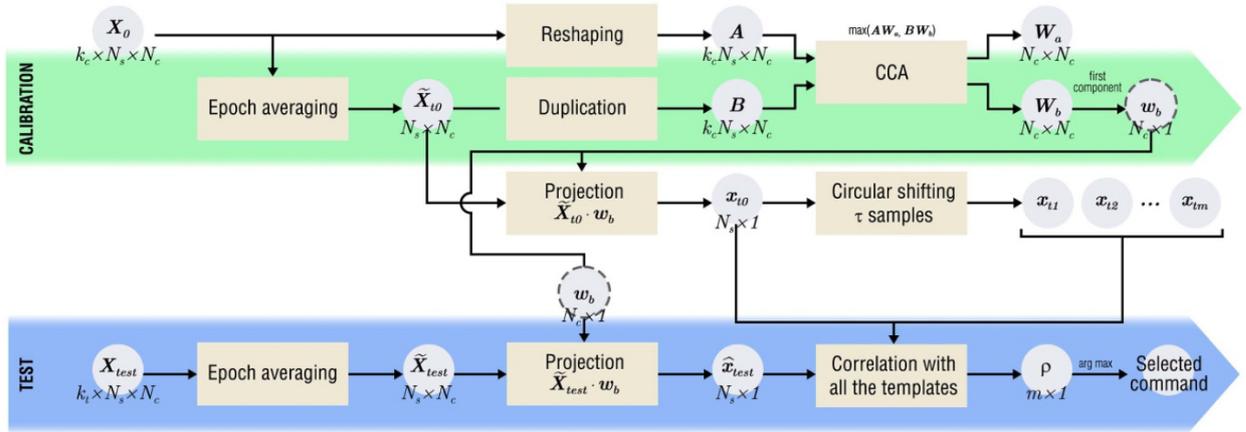


Figura 5.7: La pipeline de procesamiento de referencia para BCI basados en c-VEP con *Circular Shifting* de m-secuencias y CCA. La etapa de calibración tiene como objetivo calcular las plantillas para cada comando, de acuerdo con sus desplazamientos. En la fase online (o de test), se proyectan las épocas y determina el comando seleccionado mediante un análisis de correlación. Las variables N_s , N_c , k_c y k_t se refieren al número de muestras, canales, ciclos de calibración (o épocas de calibración) y ciclos de test (o épocas de test), respectivamente; mientras que τ indica el tamaño del desplazamiento (Martínez-Cagigal et al., 2021).

los BCI basados en c-VEP. Estos enfoques abarcan todas las etapas del sistema, desde variaciones en el paradigma hasta la presentación de estímulos, la extracción y clasificación de características, y la sincronización, entre otros aspectos. Martínez-Cagigal et al. (2021) identifica una pipeline de procesamiento de referencia que coincide exactamente en 24 de 70 estudios que analiza en su *review*. A continuación, se presentan las pipelines de procesamiento implementadas distinguiendo si se usa únicamente m-secuencias o si se usan Gold Codes para la codificación de los comandos.

5.4.1. Circular Shifting

En esta sección se expone la tubería de procesamiento que emplea un sistema BCI basado en c-VEP que utiliza m-secuencias, como producto de salida de un LFSR, para codificar los comandos. Como se discutió en la sección 5.2.1, esta m-secuencia solo puede codificar un comando inicialmente. Sin embargo, haciendo desplazamientos circulares de la misma longitud, es posible codificar más comandos. El tamaño τ de este desplazamiento dependerá de la longitud, es decir, del número de bits de la secuencia y del número de comandos a codificar. Esta técnica recibe el nombre de *Circular Shifting*, de ahí el nombre de la sección. En particular, es necesario distinguir si se utiliza CCA o TRCA como método para la extracción de filtros espaciales.

Método de extracción de filtros espaciales: CCA

Esta aproximación, descrita en la figura 5.7, parte de un EEG de varios canales. Durante el entrenamiento, se pide al usuario que se fije en un objetivo de referencia (que puede ser cualquiera de los m comandos, aunque generalmente es el que no tiene ningún desplazamiento, es decir, la m -secuencia original, para mayor simplicidad) durante k_c ciclos de calibración. Los ciclos son repeticiones de la misma secuencia. Esto se realiza porque ERP de un solo trial o ciclo suelen estar afectados por mucho ruido. Por lo tanto, se repite este proceso varias veces para obtener una respuesta cerebral libre de ruido a través del promediado, el cual elimina las componentes aleatorias o artefactos que no están consistentemente presentes en el ERP. A continuación, las épocas del EEG se remodelan con dimensiones $\mathbf{X}_0 \in \mathbb{R}^{k_c, N_s, N_c}$, donde N_s es el número de muestras de un ciclo completo, y N_c es el número de canales de EEG. La respuesta multicanal se calcula promediando a lo largo de los k_c ciclos, obteniendo $\tilde{\mathbf{X}}_{t0} \in \mathbb{R}^{N_s, N_c}$. Recuérdese de la sección 5.3.1 que el algoritmo de CCA recibe como parámetros de entrada dos matrices, que llamaremos \mathbf{A} y \mathbf{B} , que representan dos señales. El objetivo de CCA es buscar las proyecciones lineales que maximizan la correlación entre dichas señales. En concreto, $\mathbf{A} \in \mathbb{R}^{k_c N_s, N_c}$ es la matriz obtenida al cambiar las dimensiones de \mathbf{X}_0 y $\mathbf{B} \in \mathbb{R}^{k_c N_s, N_c}$ es la matriz $\tilde{\mathbf{X}}_{t0}$ replicada k_c veces para que coincidan las dimensiones (Martínez-Cagigal et al., 2021).

Como salida de CCA se obtienen los filtros espaciales $\mathbf{W}_a \in \mathbb{R}^{N_c, N_c}$ y $\mathbf{W}_b \in \mathbb{R}^{N_c, N_c}$. Sin embargo, para sistemas BCI basados en c-VEP, solamente interesa la primera componente de \mathbf{W}_b , que llamaremos \mathbf{w}_b . A continuación, proyecta la matriz $\tilde{\mathbf{X}}_{t0} \in \mathbb{R}^{N_s, 1}$, es decir, se obtiene la plantilla de referencia \mathbf{x}_{t0} como resultado del producto $\tilde{\mathbf{X}}_{t0} \mathbf{w}_b$. Esta plantilla de referencia se corresponde con la m -secuencia original, es decir, sin desplazar. Por ello, si se desplaza circularmente \mathbf{x}_{t0} acorde con el desplazamiento adecuado, se obtienen $\mathbf{x}_{t1}, \dots, \mathbf{x}_{tm}$ las plantillas para el resto de comandos (Martínez-Cagigal et al., 2021).

A continuación, se realiza la fase online, o fase de test, en la que el usuario debe fijarse en cualquiera de los comandos y el sistema BCI debe adivinar cuál está mirando. Por ello, parte de la señal obtenida del EEG modelada como $\mathbf{X}_{test} \in \mathbb{R}^{k_t, N_s, N_c}$. Tras promediar por épocas, se obtiene la matriz $\tilde{\mathbf{X}}_{test} \in \mathbb{R}^{N_s, N_c}$, que debe ser proyectada utilizando el mismo filtro espacial \mathbf{w}_b . De esta manera, se obtiene $\tilde{\mathbf{X}}_{test} \mathbf{w}_b = \hat{\mathbf{x}}_{test} \in \mathbb{R}^{N_s, 1}$. Para finalizar este proceso, se calcula el coeficiente de correlación de Pearson entre $\hat{\mathbf{x}}_{test}$ y cada una de las plantillas computadas en la fase de calibración, es decir,

$$\boldsymbol{\rho} = \left(\frac{\langle \hat{\mathbf{x}}_{test}, \mathbf{x}_{t0} \rangle}{\sqrt{\langle \hat{\mathbf{x}}_{test}, \hat{\mathbf{x}}_{test} \rangle \langle \mathbf{x}_{t0}, \mathbf{x}_{t0} \rangle}}, \frac{\langle \hat{\mathbf{x}}_{test}, \mathbf{x}_{t1} \rangle}{\sqrt{\langle \hat{\mathbf{x}}_{test}, \hat{\mathbf{x}}_{test} \rangle \langle \mathbf{x}_{t1}, \mathbf{x}_{t1} \rangle}}, \dots, \frac{\langle \hat{\mathbf{x}}_{test}, \mathbf{x}_{tm} \rangle}{\sqrt{\langle \hat{\mathbf{x}}_{test}, \hat{\mathbf{x}}_{test} \rangle \langle \mathbf{x}_{tm}, \mathbf{x}_{tm} \rangle}} \right)$$

Finalmente, se compara cual de las componentes del vector $\boldsymbol{\rho}$ es mayor, ya que el comando seleccionado se corresponde con el que maximiza dicha correlación. Es decir, si el comando seleccionado es el i_0 -ésimo, es porque

$$i_0 = \arg \max_{i=0, \dots, m} \rho_i$$

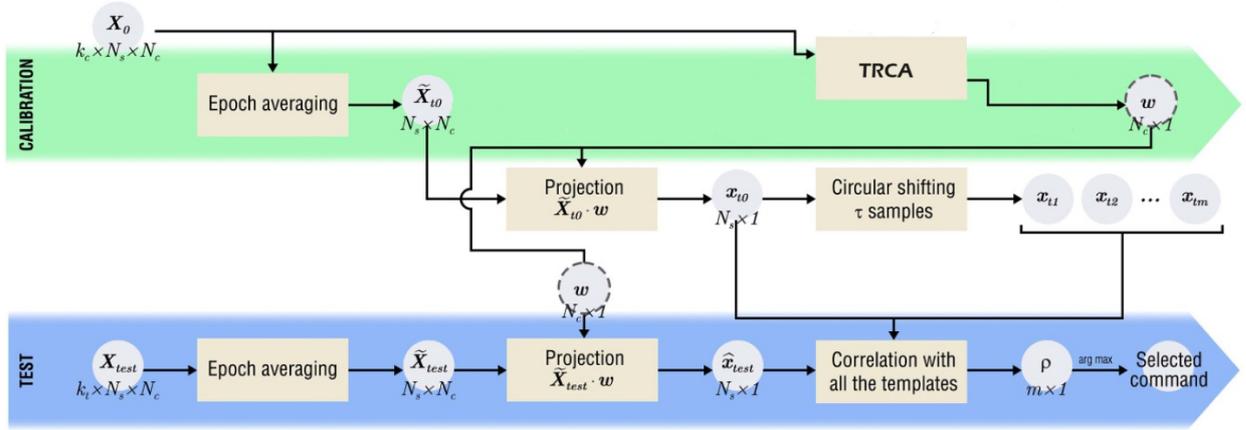


Figura 5.8: La pipeline de procesamiento por referencia de BCI basados en c-VEP con *Circular Shifting* de m -secuencias y TRCA. Utiliza la misma notación que la figura 5.7. Imagen adaptada de Martínez-Cagigal et al. (2021).

donde ρ_i representa la componente i -ésima de ρ (Martínez-Cagigal et al., 2021).

Método de extracción de filtros espaciales: TRCA

En caso de utilizar el algoritmo TRCA en vez de CCA, la tubería de procesamiento es muy similar a la anterior. Como se observa en la figura 5.8, utilizando la misma notación, se parte de las épocas del EEG remodeladas $\mathbf{X}_0 \in \mathbb{R}^{k_c, N_s, N_c}$. Este es el único parámetro que el algoritmo TRCA necesita, como se explicó en la sección 5.3.2, y proporciona como salida el filtro espacial $\mathbf{w} \in \mathbb{R}^{N_c, 1}$. El resto de pasos, tanto en la fase de calibración como en la de test son exactamente iguales que en el procesamiento con CCA pero intercambiando \mathbf{w}_b por \mathbf{w} .

5.4.2. Circular Shifting & Gold Codes

La codificación de comandos en BCI basados en c-VEP puede beneficiarse del uso de Gold Codes y *Circular Shifting* para mejorar la eficiencia y la capacidad de discriminación entre diferentes comandos. En esencia, como se detalló en la sección 5.2.2, los Gold Codes son secuencias pseudoaleatorias que tienen excelentes propiedades de correlación cruzada, lo que los hace ideales para la codificación de señales en sistemas de comunicación y, en este caso, para la codificación de comandos en sistemas BCI.

Imaginemos que deseamos codificar un conjunto de C comandos distintos. Para lograr esto, primero seleccionamos dos m -secuencias generadas a partir de dos polinomios de grado n que formen un par preferido (ver tabla 5.2). Este proceso nos proporciona hasta 2^{n-1} códigos Gold posibles. A continuación, elegimos N_G de estos códigos Gold para codificar nuestros C comandos, donde $1 \leq N_G \leq 2^{n-1}$.

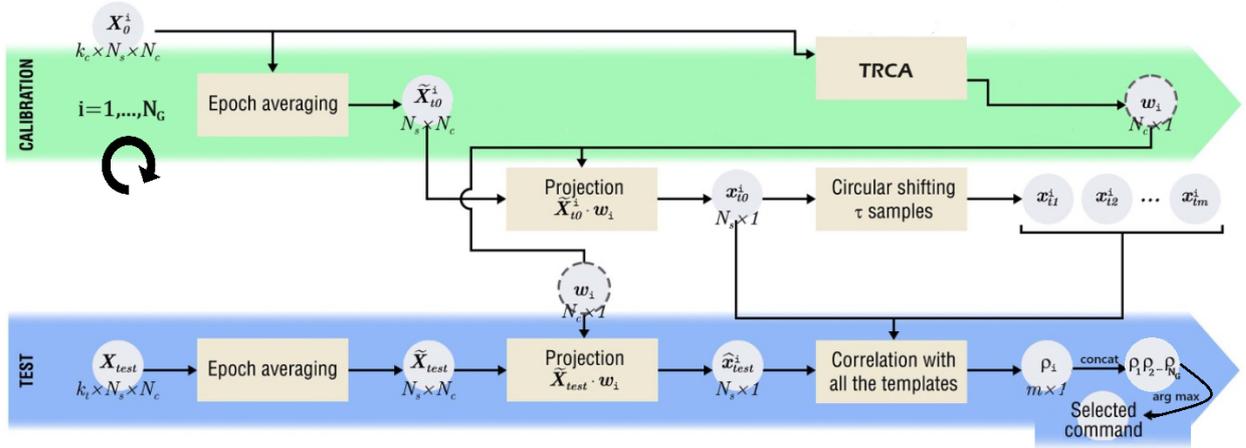


Figura 5.9: La pipeline de procesamiento por referencia de BCI basados en c-VEP con *Circular Shifting Gold Codes* y TRCA. Utiliza la misma notación que la figura 5.7. Se observa que, en este caso, se repite el proceso una vez por cada Gold Code. Imagen adaptada de [Martínez-Cagigal et al. \(2021\)](#).

Cada uno de estos códigos Gold seleccionados tiene la capacidad de codificar 1 comando. Sin embargo, en la práctica, se aplica un proceso de *Circular Shifting* a cada uno de ellos. Este desplazamiento circular implica mover la secuencia en una dirección determinada a través de distintas longitudes de desplazamiento para cada código Gold. Sin embargo, algunos investigadores sugieren que se utilice la misma longitud de desplazamiento ([Sun et al., 2022](#)).

Al aplicar este *Circular Shifting* a cada código Gold, aumentamos aún más la capacidad de codificación. Es decir, ahora podemos codificar hasta $N_G \times m$ comandos, donde m es el número de secuencias obtenidas al desplazar una única secuencia Gold. En la práctica, se busca seleccionar N_G y m de manera que su multiplicación sea igual a C , la cantidad total de comandos que queremos codificar. Otro aspecto crucial a considerar en la selección de estos dos parámetros es que, a medida que aumenta el valor de N_G , se prolonga el tiempo requerido para la fase de calibración. Esto puede representar un obstáculo significativo en numerosos experimentos, ya que puede interferir con el flujo de la investigación.

La figura 5.9 muestra la tubería de procesamiento característica en los sistemas BCI basados en c-VEP que emplean Gold Codes y *Circular Shifting*. En esta descripción, nos enfocaremos exclusivamente en el algoritmo TRCA, ya que fue el método empleado, tanto en las pruebas realizadas con Gold Codes durante la ejecución de este Trabajo de Fin de Grado, como la literatura tomada como referencia en esta sección ([Sun et al., 2022](#)). Se puede observar que la fase de calibración es exactamente igual que en la figura 5.8 pero realizada N_G veces, es decir, una vez por Gold Code seleccionado. Así, se obtienen w_i filtros espaciales y $x_{t_0}^i, x_{t_1}^i, \dots, x_{t_m}^i$ plantillas con $1 \leq i \leq N_G$. La fase de test es también muy similar, ya que, una vez hecha la media por épocas, obteniendo así \hat{X}_{test} , es necesario proyectarlo con cada uno de los filtros espaciales w_i . Es decir, se obtiene $\hat{X}_{test} w_i = \hat{x}_{test}^i$,

para que, a continuación, se calcule la correlación con las plantillas calculadas en la fase de calibración, es decir,

$$\rho_i = \left(\frac{\langle \hat{\mathbf{x}}_{test}^i, \mathbf{x}_{t_0}^i \rangle}{\sqrt{\langle \hat{\mathbf{x}}_{test}^i, \hat{\mathbf{x}}_{test}^i \rangle \langle \mathbf{x}_{t_0}^i, \mathbf{x}_{t_0}^i \rangle}}, \frac{\langle \hat{\mathbf{x}}_{test}^i, \mathbf{x}_{t_1}^i \rangle}{\sqrt{\langle \hat{\mathbf{x}}_{test}^i, \hat{\mathbf{x}}_{test}^i \rangle \langle \mathbf{x}_{t_1}^i, \mathbf{x}_{t_1}^i \rangle}}, \dots, \frac{\langle \hat{\mathbf{x}}_{test}^i, \mathbf{x}_{t_m}^i \rangle}{\sqrt{\langle \hat{\mathbf{x}}_{test}^i, \hat{\mathbf{x}}_{test}^i \rangle \langle \mathbf{x}_{t_m}^i, \mathbf{x}_{t_m}^i \rangle}} \right)$$

donde $1 \leq i \leq N_G$. Finalmente, se concatenan estos vectores ρ_i y se busca aquella componente cuyo valor sea máximo.

5.5. Otros aspectos a tener en cuenta

Una vez expuestos los conceptos fundamentales de los sistemas BCI basados en c-VEP, existen una serie de aspectos adicionales que merecen atención y análisis detallado. En esta sección, exploraremos tres de estos aspectos cruciales: los ratios de presentación y las *raster latencies*. Cada uno de estos elementos desempeña un papel crucial en la eficacia y la precisión de esta tecnología, ofreciendo perspectivas interesantes para mejorar tanto la investigación como las aplicaciones prácticas en este campo.

5.5.1. Ratios de presentación

Para optimizar la rapidez de selección de comandos se suele ajustar la tasa de presentación de estímulos visuales para que coincida con la frecuencia de actualización del monitor. En su análisis del paradigma c-VEP, [Martínez-Cagigal et al. \(2021\)](#) examina diversas investigaciones que han utilizado tasas de presentación que van desde 15 Hz hasta 240 Hz. El objetivo es comparar los resultados obtenidos y evaluar la precisión en relación con la tasa de presentación empleada.

Este mismo autor concluye que no hay un acuerdo unánime sobre cuál es la tasa de presentación más adecuada para los BCI basados en c-VEP. La precisión parece depender de múltiples variables, como la cantidad de ciclos de calibración y prueba, la longitud de la secuencia de bits, el número de comandos y códigos, entre otros factores. Por ejemplo, para algunos usuarios, una tasa de presentación de 120 Hz puede ser óptima, ya que duplica el número de ciclos en comparación con una tasa de 60 Hz, lo que reduce a la mitad los tiempos de calibración. No obstante, es necesario tener precaución al aumentar la tasa de presentación, ya que esto podría resultar en una menor ortogonalidad entre las plantillas c-VEP.

Es relevante destacar que, a pesar de un incremento en la tasa de presentación, las respuestas del EEG mantienen una distribución espectral constante. Esto se atribuye presumiblemente a la no linealidad del sistema visual y a la menor sensibilidad de los conos a estímulos de alta frecuencia. Además, hay un consenso claro en la comunidad científica respecto a que tasas de secuencia más elevadas resultan en una estimulación menos fatigante para el usuario.

5.5.2. Raster latencias

En consonancia con la sección anterior, los sistemas BCI basados en c-VEP suelen necesitar una precisión temporal en la escala de milisegundos para su correcto funcionamiento. Dado que la mayoría de los experimentos se llevan a cabo en un ordenador personal, es crucial considerar y corregir las latencias introducidas por la configuración del sistema (Nagel et al., 2018). La actualización de cada fotograma en los monitores estándar no ocurre de manera simultánea para todos los píxeles de la pantalla. En su lugar, se utiliza un proceso llamado rasterización, que actualiza cada línea de píxeles secuencialmente, de arriba hacia abajo, en lo que se conoce como “barrido vertical”. Este proceso introduce una latencia en la rasterización (en inglés, *raster latencies*) que aumenta de manera sistemática desde el píxel ubicado en la esquina superior izquierda hasta el píxel en la esquina inferior derecha de la pantalla (Martínez-Cagigal et al., 2021).

Estas variaciones en la latencia pueden afectar a la respuesta de comandos c-VEP, especialmente porque estos comandos se ubican en diferentes posiciones a lo largo de la pantalla. Nagel et al. (2018) evidenció que corregir estas variaciones puede mejorar significativamente el rendimiento del sistema. Por ello, se sugiere medir la diferencia de tiempo entre la actualización de las líneas superiores e inferiores utilizando un sensor óptico u otro tipo de dispositivo similar. Estas mediciones de la latencia pueden utilizarse para ajustar temporalmente las plantillas estimadas por la fase de calibración y mejorar la precisión global del sistema.

Capítulo 6

Estado del arte

En este capítulo, a partir de tres artículos indexados de referencia, se analiza el estado de la técnica actual de sistemas BCI basados en c-VEP, filtros espaciales como CCA y la generación de secuencias números pseudoaleatorios.

6.1. Introducción

El presente Trabajo de Fin de Grado, se adentra en un terreno poco explorado en el ámbito de la investigación los sistemas BCI basados en c-VEP: la comparación entre dos enfoques alternativos, CCA y TRCA. A diferencia de la mayoría de los estudios previos, que han utilizado estas metodologías de manera independiente, esta investigación propone un análisis comparativo detallado de ambas técnicas. Destacando la innovación de este estudio, cabe resaltar que nunca antes se ha llevado a cabo una comparación directa entre CCA y TRCA en el contexto específico que abordamos.

Además, es esencial resaltar que, hasta el momento, no se ha encontrado literatura que emplee TRCA como método de procesamiento de señales para la obtención de filtros espaciales al codificar los comandos de c-VEP mediante secuencias no binarias, i.e., de base p . Estas secuencias no se limitan únicamente a los valores binarios 0 y 1 (i.e., estímulos blancos y negros), sino que se generan a partir de polinomios primitivos de \mathbb{Z}_p . Estas secuencias se codifican utilizando p colores que, normalmente, son $p - 2$ tonos de grises, blanco y negro, logrando una mayor comodidad para el usuario ya que generan menor fatiga visual. Específicamente, este documento presenta un análisis que utiliza TRCA para las p -ary m-secuencias a través de la base de datos de [Martínez-Cagigal et al. \(2023\)](#), que utiliza CCA.

En consecuencia, a continuación se exponen tres estudios representativos que sintetizan los precedentes de las investigaciones llevadas a cabo hasta la fecha sobre este tema.

6.2. Martínez-Cagigal *et al.* (2021) -Brain-computer interfaces based on code-modulated visual evoked potentials (c-VEP): a literature review

Martínez-Cagigal *et al.* (2021) realiza una revisión de la literatura actual sobre sistemas BCI basados en c-VEP. Por ello, el objetivo de este artículo es proporcionar una visión general de la literatura para comprender la investigación existente sobre sistemas BCI basados en c-VEP, desde sus inicios (1984) hasta el momento de su publicación (2021), así como identificar líneas de investigación futuras prometedoras. Para ello, se realizó una búsqueda sistemática en distintas herramientas, como son *Web of Science* o *Google Scholar* a través de distintas palabras clave. Así, se consiguieron 70 estudios, de los que se extrajo importante información sobre esta tecnología y que se resume a continuación.

Esta revisión comienza explicando los conceptos básicos más importantes, como pueden ser los sistemas BCI en general, la generación de números pseudoaleatorios basados en m-secuencias o una tubería de procesamiento por excelencia de este tipo de sistemas BCI, utilizada exactamente igual en 24 de los 70 estudios. Aquí, se profundiza en la extracción de filtros espaciales utilizando CCA y en el funcionamiento interno de un BCI basado en c-VEP.

A continuación, explica las diferencias sustanciales encontradas en los diversos estudios analizados. En particular, realiza una comparación entre SSVEP y c-VEP, llegando a la conclusión de que los BCI basados en c-VEP consiguen una velocidad de selección igual o incluso mayor que aquellos basados en SSVEP. Martínez-Cagigal *et al.* (2021) remarca que en SSVEP no es estrictamente necesario realizar una fase de calibración, sin embargo, en c-VEP sí. Por otra parte, c-VEP es menos sensible a actividad basal del EEG no relacionada con la tarea que SSVEP, y además es más sencillo aumentar el número de comandos a seleccionar que en SSVEP.

Los autores también analiza los distintos métodos de generación de números pseudoaleatorios utilizados para codificar los comandos objetivo, como se observa en la tabla 6.1, donde destaca el uso de m-secuencias y de códigos Gold. Sin embargo, algunos de los artículos analizados aseguran que el tipo de secuencia de bits no afecta de forma significativa a los resultados, con tal de que estas tengan buenas propiedades de autocorrelación y correlación cruzada.

Además de otros aspectos que diferencian los estudios analizados en esta revisión, es necesario destacar la comparativa sobre el uso distintos algoritmos para la extracción de filtros espaciales. En concreto, 42 de los 70 estudios utilizan CCA. También se pueden encontrar métodos como *spatio-temporal beamformers*, δ - α *phase-amplitude coupling* o el uso de redes neuronales.

6.3. Nakanishi *et al.* (2018) -Enhancing Detection of SSVEPs for a High-Speed Brain Speller Using Task-Related Component Analysis

Tabla 6.1: Secuencias de bits empleadas en los estudios revisados por [Martínez-Cagigal et al. \(2021\)](#). # representa el número de estudios.

Tipo de secuencia	#
m-secuencias	58
Gold	9
Kasami	2
Barker	2
Golay	5
APA	4
De Bruijn	1
Hand-crafted	5
Random	4

6.3. Nakanishi *et al.* (2018) -Enhancing Detection of SSVEPs for a High-Speed Brain Speller Using Task-Related Component Analysis

Este estudio propone y evalúa un enfoque novedoso de filtrado espacial para mejorar la detección de SSVEPs en sistemas BCI. Para ello, realizan una comparación sobre el rendimiento del BCI utilizando tanto TRCA como CCA. La originalidad de este estudio radica en el hecho de que nunca antes se había aplicado TRCA con SSVEPs, y la motivación para ello se debe a que, a pesar de la fortaleza y buenos resultados que muestra CCA para detectar SSVEPs, su efectividad se ve afectada por la interferencia de actividades espontáneas del EEG.

Se realizaron dos experimentos, uno offline en el que participaron 12 adultos, con una media de 25 años, y uno online, donde participaron 20 adultos, con una edad media también de 25 años. Los estímulos se mostraron en un monitor de 60 Hz, donde la interfaz de usuario era una matriz de dimensiones 5×8 , es decir, con un total de 40 comandos a seleccionar (figura 6.1). Los datos de EEG fueron registrados por nueve electrodos ubicados sobre áreas parietal y occipital (Pz, PO5, PO3, POz, PO4, PO6, O1, Oz, O2) utilizando un sistema Synamp2 (Neuroscan, Inc.) a una tasa de muestreo de 1,000 Hz. El electrodo de referencia se colocó en Cz.

En el experimento BCI offline, los sujetos llevaron a cabo 12 bloques de tareas en los que se les pidió enfocar la mirada en estímulos visuales durante 0.5 segundos, realizando 40 trials correspondientes a los 40 estímulos. Cada trial iniciaba con una señal visual, indicando el estímulo objetivo, seguido de una fase de parpadeo simultáneo de todos los

>> HIGH SPEED BCI							
8.0 / 0.00	9.0 / 1.75	10.0 / 1.50	11.0 / 1.25	12.0 / 1.00	13.0 / 0.75	14.0 / 0.50	15.0 / 0.25
8.2 / 0.35	9.2 / 0.10	10.2 / 1.85	11.2 / 1.60	12.2 / 1.35	13.2 / 1.10	14.2 / 0.85	15.2 / 0.60
8.4 / 0.70	9.4 / 0.45	10.4 / 0.20	11.4 / 1.95	12.4 / 1.70	13.4 / 1.45	14.4 / 1.20	15.4 / 0.95
8.6 / 1.05	9.6 / 0.80	10.6 / 0.55	11.6 / 0.30	12.6 / 0.05	13.6 / 1.80	14.6 / 1.55	15.6 / 1.30
8.8 / 1.40	9.8 / 1.15	10.8 / 0.90	11.8 / 0.65	12.8 / 0.40	13.8 / 0.15	14.8 / 1.90	15.8 / 1.65

Freq.
(Hz)

Phase
(π)

Figura 6.1: Diseño de los estímulos del sistema BCI de 40 objetivos. Se especifican los valores de frecuencia y fase para cada objetivo (Nakanishi et al., 2018).

estímulos durante 0.5 segundos. Se instruyó a los participantes a evitar parpadear durante este periodo. Se realizaron pausas entre bloques para evitar la fatiga visual y permitir el análisis efectivo de los datos offline.

Se realizó un experimento online para validar el BCI de alta velocidad utilizando el método basado en TRCA. Este se dividió en una fase de entrenamiento y una de prueba. La primera constaba de 12 bloques de 40 trials cada uno. Como su propio nombre indica, los datos de esta fase se utilizan como entrenamiento para la fase de prueba. Esta última incluyó 5 bloques de la tarea de selección guiada con trials de 0,8 s cada uno, donde los sujetos debían cambiar la mirada hacia el objetivo tras una estimulación visual de 0.3 s. Se proporcionaron retroalimentaciones visual y auditiva en tiempo real, y tras la identificación correcta de un objetivo, se emitía un breve sonido y se escribía el carácter en la pantalla. El sistema BCI deletreó a 75 caracteres/minuto en la prueba. Además, se realizó una tarea de deletreo libre sin señales visuales, donde los sujetos debían ingresar *HIGH SPEED BCI* tres veces, con la posibilidad de corregir errores usando la tecla ‘Retrosceso’.

La comparación de los métodos TRCA superó a CCA. TRCA, en conjunto, logró el mejor rendimiento independientemente de la longitud de los datos. En cuanto a los resultados obtenidos en el experimento online usando TRCA, se logró una precisión media de $89,83\% \pm 6,07\%$ y una tasa de transferencia de la información (ITR) media de $325,33 \pm 38,17$ bits/min, como se muestra en la tabla 6.2.

6.3. Nakanishi *et al.* (2018) -Enhancing Detection of SSVEPs for a High-Speed Brain Speller Using Task-Related Component Analysis

Tabla 6.2: Resultados del experimento online de [Nakanishi et al. \(2018\)](#).

Sujeto	Trials (Correcto/Incorrecto)	Precisión (%)	ITR (bits/min)
s1	195/5	97,50	376,58
s2	183/17	91,50	333,98
s3	194/6	97,00	372,67
s4	189/11	94,50	354,30
s5	167/33	83,50	285,28
s6	159/41	79,50	263,00
s7	161/39	80,50	268,46
s8	174/26	87,00	305,80
s9	159/41	79,50	263,00
s10	169/31	84,50	291,04
s11	181/19	90,50	327,52
s12	185/15	92,50	340,59
s13	177/23	88,50	314,95
s14	168/32	84,00	288,15
s15	195/5	97,50	376,58
s16	185/15	92,50	340,59
s17	189/11	94,50	354,30
s18	188/12	94,00	350,80
s19	192/8	96,00	365,12
s20	183/17	91,50	333,98
Mean±STD	-	89,83 ± 6,07	325,33 ± 38,17

Tabla 6.3: m-secuencias de 31 bits, representadas por M_i , con $i = 1, 2$ y los Gold Codes seleccionados, representados por G_j , con $j = 1, \dots, 4$ del texto de [Sun et al. \(2022\)](#).

Código	Secuencia
M1	0100101100111110001101110101000
M2	0011100110111110100010010101100
G1	1000011011001010011111011001001
G2	0010010010011100011000010100110
G3	0000111110010010000011101110111
G4	1101111010111001000000001000000

6.4. Sun et al. (2022) -A 120-target brain-computer interface based on code-modulated visual evoked potentials

[Sun et al. \(2022\)](#) llevaron a cabo un estudio innovador, uno de los primeros en la literatura en combinar Gold Codes, *Circular Shifting* y el algoritmo TRCA. En particular, esta investigación implementa un sistema BCI basado en c-VEP de 120 comandos, a partir de cuatro Gold Codes de 31 bits, donde cada código generará 30 secuencias a partir de desplazamientos temporales de un sólo bit.

Este autor, a pesar de las evidencias existentes en otras investigaciones sobre utilizar una longitud de desplazamiento inferior a 2, consigue buenos resultados, resaltando el potencial que tiene el uso de desplazamientos circulares de un conjunto de códigos inicial, los cuales han de tener buenas propiedades tanto de autocorrelación como de correlación cruzada. Los códigos con tales propiedades elegidos por [Sun et al. \(2022\)](#), son los Gold Codes, generados a partir de dos m-secuencias, como se puede observar en la tabla 6.3. Es fundamental notar que, en este estudio, 0 hace referencia al color blanco y 1 al color negro.

En este estudio se realizan dos fases experimentales, una online y otra offline. Participan 22 sujetos sanos, de edades entre 23 y 28 años, donde 13 personas son mujeres. En particular, 11 sujetos participaron en la fase offline y 22 en la online. Se utilizan 9 electrodos (PO3, Pz, PO4, PO5, POz, PO6, O1, Oz, O2) siguiendo el estándar internacional 10/20 para adquirir la actividad cerebral. La frecuencia de refresco del monitor utilizado son 60 Hz y con una resolución lo suficientemente grande (5120×1440) como para poder abarcar los 120 comandos objetivo.

El experimento offline tiene como objetivo seleccionar aquellos Gold Codes para el experimento online que mejores propiedades tengan. La GUI utilizada, mostrada en la figura 6.2, consiste en 30 objetivos, que se corresponden con los desplazamientos del Gold

6.4. Sun *et al.* (2022) -A 120-target brain-computer interface based on code-modulated visual evoked potentials

	C1_1	C1_2	C1_3	C1_4	C1_5	C1_6	C1_7	C1_8	C1_9	C1_10	
	C1_11	C1_12	C1_13	C1_14	C1_15	C1_16	C1_17	C1_18	C1_19	C1_20	
	C1_21	C1_22	C1_23	C1_24	C1_25	C1_26	C1_27	C1_28	C1_29	C1_30	

Figura 6.2: Matriz de comandos utilizada en la fase experimental offline en Sun *et al.* (2022).

C1_1	C1_2	C1_3	C1_4	C1_5	C1_6	C1_7	C1_8	C1_9	C1_10		C2_1	C2_2	C2_3	C2_4	C2_5	C2_6	C2_7	C2_8	C2_9	C2_10
C1_11	C1_12	C1_13	C1_14	C1_15	C1_16	C1_17	C1_18	C1_19	C1_20		C2_11	C2_12	C2_13	C2_14	C2_15	C2_16	C2_17	C2_18	C2_19	C2_20
C1_21	C1_22	C1_23	C1_24	C1_25	C1_26	C1_27	C1_28	C1_29	C1_30		C2_21	C2_22	C2_23	C2_24	C2_25	C2_26	C2_27	C2_28	C2_29	C2_30
C3_1	C3_2	C3_3	C3_4	C3_5	C3_6	C3_7	C3_8	C3_9	C3_10		C4_1	C4_2	C4_3	C4_4	C4_5	C4_6	C4_7	C4_8	C4_9	C4_10
C3_11	C3_12	C3_13	C3_14	C3_15	C3_16	C3_17	C3_18	C3_19	C3_20		C4_11	C4_12	C4_13	C4_14	C4_15	C4_16	C4_17	C4_18	C4_19	C4_20
C3_21	C3_22	C3_23	C3_24	C3_25	C3_26	C3_27	C3_28	C3_29	C3_30		C4_21	C4_22	C4_23	C4_24	C4_25	C4_26	C4_27	C4_28	C4_29	C4_30

Figura 6.3: Matriz de comandos utilizada en la fase experimental online en Sun *et al.* (2022).

Code correspondiente. Así, los sujetos tenían que presionar un botón para empezar cada bloque, concentrándose en el objetivo número 15, que se iluminaba de color rojo. A continuación, todos los objetivos empezaban a parpadear en consonancia con su codificación, durante 5 bloques y cada bloque, durante 12 ciclos. Este proceso se repetía para cada Gold Code, dejando como descanso más de 1 minuto entre secuencias. Tras el análisis offline, se escogieron 4 de los 31 Gold Codes para utilizarlos como base para codificar los comandos en el experimento online.

La parte online se puede dividir en la fase de calibración o entrenamiento y la fase de prueba, utilizando una matriz de comandos de 120 objetivos, como se muestra en la figura 6.3. El objetivo de la fase de entrenamiento es crear las 120 plantillas necesarias para cada comando a partir de 4 filtros espaciales, uno por cada Gold Code. En esta fase, solamente se entrena cada uno de los Gold Codes y, a continuación, se desplaza la plantilla obtenida para entrenar el resto de comandos. Esta fase consiste en 5 trials, de 12 ciclos cada uno, donde el usuario podía descansar entre bloques, repitiendo este proceso secuencialmente, hasta entrenar los cuatro Gold Codes. Así, Sun *et al.* (2022) logra un tiempo récord de entrenamiento de 3.9 minutos aproximadamente. Durante la fase de prueba, los sujetos tenían que seguir un cuadrado rojo que se desplazaba por cada uno de los 120 comandos, durante 30 segundos por objetivo.

Como se ha resaltado previamente, la novedad de este estudio es (1) el uso de desplazamientos de un sólo bit, ya que aumentar dicha longitud haría disminuir el número máximo de comandos a codificar; (2) la utilización del algoritmo TRCA para la extracción de características. Finalmente, tras los correspondientes análisis, se obtiene una precisión media de 76,58 % y una ITR media de 265,74 bits/minuto.

Capítulo 7

Diseño y desarrollo del sistema BCI

En este capítulo se presenta el diseño y desarrollo del sistema BCI. En concreto, se describirán los materiales y el equipo utilizados, un esquema general de la aplicación y todos los detalles acerca de MEDUSA[©]. En particular, se comentarán sus principios fundamentales, la distinción entre MEDUSA[©] Kernel y Platform y los protocolos utilizados para garantizar la correcta comunicación entre sus componentes. Finalmente, se explicará el *workflow* y los estados internos de una aplicación dentro de MEDUSA[©].

7.1. Materiales y equipo

Como se adelantó en la sección 3.3, el material utilizado para la realización de los experimentos es el siguiente:

- Para la adquisición de la señal se han utilizado una serie de dispositivos hardware fabricados por la empresa g.Tec (<https://www.gtec.at/>), entre los que se incluye (véase la figura 7.1) un amplificador g.USBamp que se conecta fácilmente al puerto USB del ordenador. Tiene un rango de entrada de ± 250 mV, lo que permite el registro de señales de corriente continua (DC) sin saturación, utilizando una frecuencia de muestro de 256Hz. También se incluye una g.GAMMAbox que comunica 16 electrodos g.LADYbird, la tierra y la referencia con el amplificador. Los electrodos son activos, es decir, cuentan con un preamplificador integrado en el mismo electrodo. Este preamplificador ayuda a reducir el ruido electromagnético que podría adquirirse a través de los cables. Además, estos electrodos son húmedos de Ag/AgCl, lo que implica que es necesario utilizar gel electrofisiológico. Estos electrodos se acoplan a al gorro g.GAMMAcap3, que posteriormente se colocará a cada uno de los sujetos para poder medir su actividad cerebral. También se utilizó g.GAMMAsys como electrodo de referencia que se coloca en la oreja izquierda o derecha de los sujetos. Finalmente, se utilizó gel electrofisiológico de Neuroline entre cada uno de los electrodos y el cuero cabelludo de los sujetos para reducir las impedancias y mejorar la calidad de la señal.

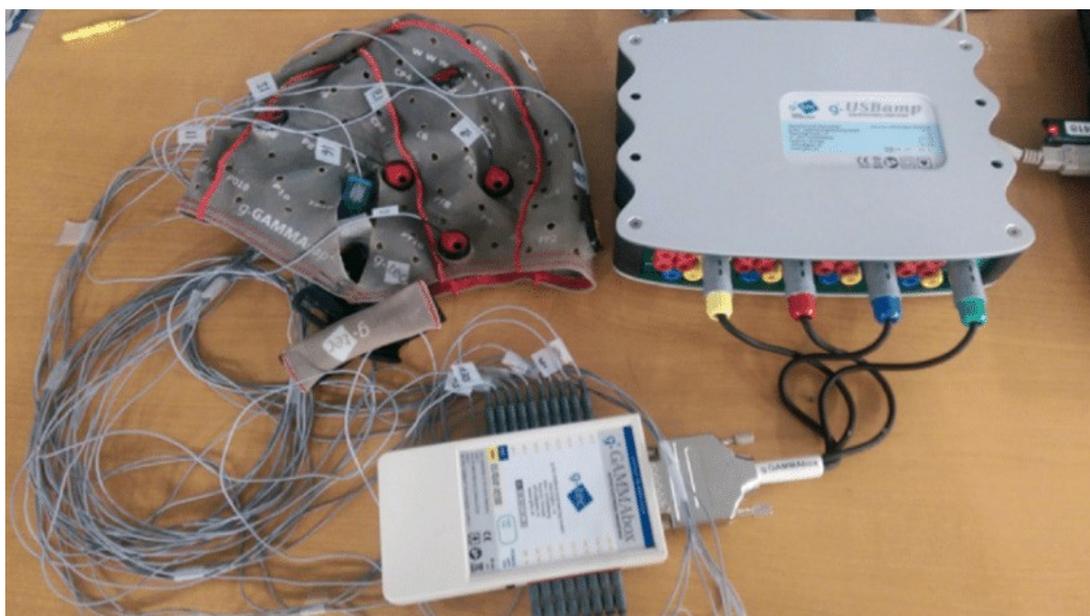


Figura 7.1: Equipo de g.Tec utilizado. En la parte superior a la derecha, se aprecia el amplificador g.USBamp. En la parte superior a la izquierda, el gorro g.GAMMAcap3 con los electrodos g.LADYbird ya colocados. Finalmente, en la parte inferior, la g.GAMMAbox. Imagen adaptada de [Chenane et al. \(2019\)](#).

- Para el análisis de datos y la ejecución de la aplicación, se ha escogido un ordenador de sobremesa con un procesador Intel® Core™ i7-10700F CPU a 2.90 GHz, una memoria RAM de 32 GB, con un sistema operativo Windows de 64 bits y una tarjeta gráfica GeForce 1050 Ti.

Es importante tener en cuenta que para la realización de los experimentos online se han utilizado 16 electrodos, que son Oz, F3, Fz, F4, I1, I2, C3, Cz, C4, CPz, P3, Pz, P4, PO7, POz y PO8 según el estándar 10-10. El electrodo tierra se colocó sobre AFz y la referencia en el lóbulo de la oreja.

7.2. Esquema general del sistema

La figura 7.2 representa de forma esquemática la estructura del sistema BCI. Una vez colocado el gorro sobre el sujeto y conectados todos los dispositivos hardware, la aplicación MEDUSA[©] recibe a través del protocolo LSL el flujo de datos del EEG recogidos por los electrodos. Esto permitiría apreciar en tiempo real las gráficas derivadas del EEG. A continuación, de forma paralela a la ejecución de MEDUSA[©], se lanza la aplicación que se vaya a ejecutar a través de Unity. Es necesario remarcar que la comunicación entre MEDUSA[©] y la aplicación en Unity se realiza mediante un protocolo TCP/IP asíncrono.

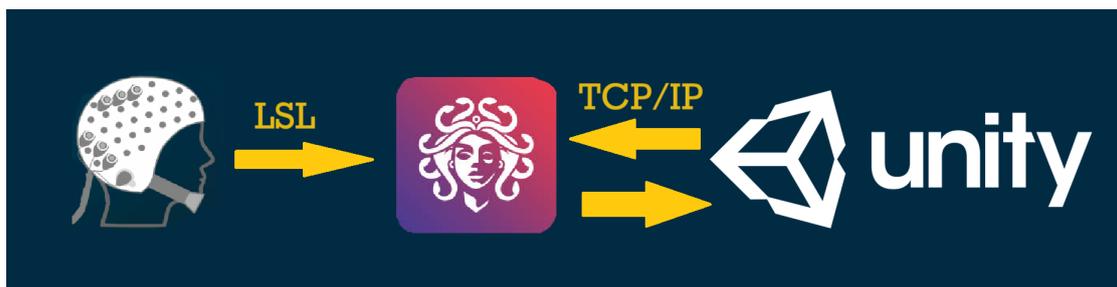


Figura 7.2: Esquema general de la estructura del sistema BCI.

7.3. Implementación de la aplicación en MEDUSA[©]

Las herramientas de software y las aplicaciones diseñadas específicamente para facilitar la implementación de experimentos en neurociencia son de suma importancia para el avance de la investigación cerebral y la neurotecnología. De hecho, no se puede subestimar el impacto que este tipo de herramientas podría tener en campos específicos, dado su potencial para agilizar la experimentación, reducir los costos del proyecto y permitir la participación de investigadores sin conocimientos técnicos (Renard et al., 2010). Aunque existen una variedad de paquetes de herramientas de procesamiento de señales e imágenes con métodos de vanguardia (como EEGLAB, Brainstorm, MNE, todos ellos para análisis offline), las opciones para el diseño e implementación experimental son limitadas. La mayoría de las herramientas actuales ofrecen pocas o ninguna opción de personalización y solo pueden aplicarse a tareas específicas. Además, estos programas suelen distribuirse bajo términos propietarios sin compatibilidad entre los equipos de registro biomédico de diferentes fabricantes, lo que limita la capacidad para aprovechar todos los recursos disponibles en un laboratorio de investigación (Santamaría-Vázquez et al., 2023).

A continuación, se presenta MEDUSA[©], un innovador ecosistema de software basado en Python para implementar experimentos de BCI y neurociencia, que tiene como objetivo superar las limitaciones previas.

7.3.1. Visión general de MEDUSA[©]

MEDUSA[©] ha sido concebido y desarrollado bajo tres principios fundamentales (Santamaría-Vázquez et al., 2023):

- **Modularidad:** MEDUSA[©] está compuesto por estructuras autónomas organizadas en diferentes niveles de abstracción, interconectadas mediante protocolos de comunicación simples. Esto permite realizar correcciones o mejoras rápidas en funcionalidades específicas sin interferir con el resto de los componentes.
- **Flexibilidad:** MEDUSA[©] ha sido diseñado específicamente como una herramienta de investigación, gracias a su arquitectura que permite la realización ágil de experimentos con nuevos métodos de procesamiento de señales y paradigmas de retroalimentación.

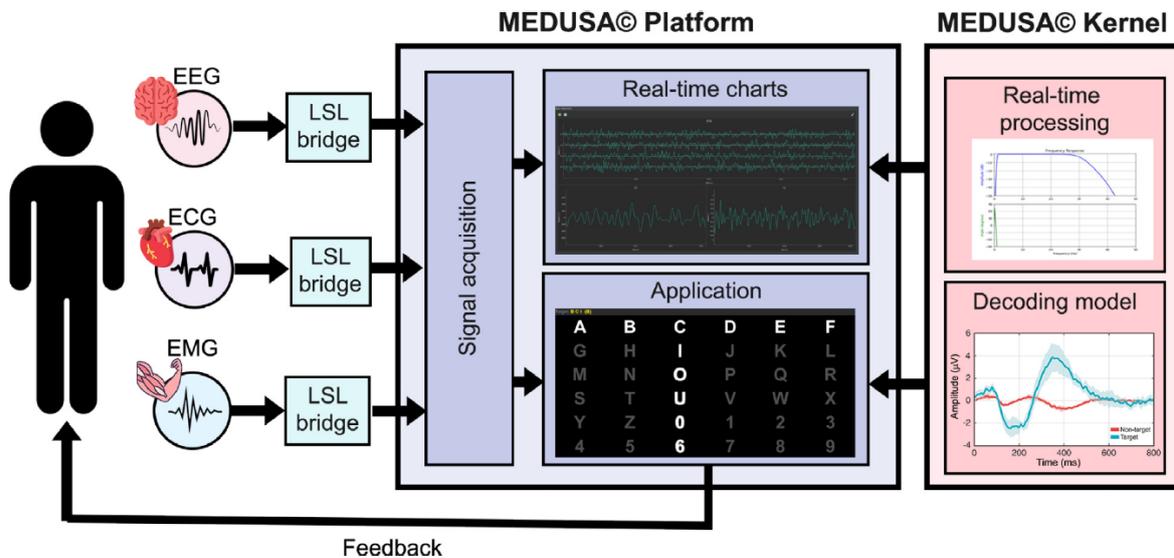


Figura 7.3: Esquema general MEDUSA[©] (Santamaría-Vázquez et al., 2023).

ción. Además, sus desarrolladores han puesto especial atención en la documentación y los comentarios del código, incluyendo ejemplos y tutoriales que ilustran el funcionamiento de la plataforma y cómo desarrollar nuevas aplicaciones.

- **Escalabilidad:** MEDUSA[©] está diseñado para mejorar sus capacidades con el tiempo sin necesidad de modificar partes del código que no estén relacionadas, gracias al uso de metaclasses estandarizadas, lo cual resulta especialmente útil en un entorno de investigación. Este enfoque permite que el software se mantenga al día con los últimos avances en el campo de BCI, que pueden incluir nuevos algoritmos de procesamiento de señales o paradigmas.

La aplicación de MEDUSA[©] se divide en dos componentes principales, con dos objetivos diferenciados, que son, MEDUSA[©] Kernel y MEDUSA[©] Platform, como trata de representar la figura 7.3.

MEDUSA[©] Kernel

Es un paquete de Python que ofrece métodos para utilizar en el análisis de señales cerebrales, abarcando desde el procesamiento avanzado de señales hasta el aprendizaje automático, aprendizaje profundo y una variedad de análisis de alto nivel. Las funciones que se incluyen pueden ser divididas en diferentes niveles de abstracción (Santamaría-Vázquez et al., 2023). El primer nivel consiste en funciones de bajo nivel, que son métodos básicos que pueden usarse para procesar señales en una variedad de situaciones, incluyendo filtros temporales, filtros espaciales (como puede ser CCA), métricas de activación local o métricas de conectividad. En un nivel de abstracción más alto, existen funciones que aplican una serie de procesos a los datos de entrada para analizar ciertas características.

En concreto, MEDUSA[©] no asume la naturaleza de los datos de entrada en las funciones de bajo nivel, pero la mayoría de los análisis de alto nivel que están actualmente implementados están diseñados para trabajar con registros de EEG y MEG. En pocas palabras, las funciones de alto nivel utilizan los métodos de bajo nivel para implementar casos de uso específicos. Estas funciones incluyen algoritmos de procesamiento de señales para BCI basados en P300, *Motor imagery*, *c-VEPs* y *Neurofeedback*.

Esta arquitectura modular, organizada en niveles de abstracción, garantiza la independencia entre los diferentes componentes. Además, el paquete incluye clases y funciones para importar datos de otras herramientas (por ejemplo, MATLAB, MNE), definir el formato de datos de señales y experimentos, guardar registros en varios tipos de archivos (por ejemplo, bson, json, mat) e implementar tuberías de procesamiento de señales personalizadas en tiempo real, lo que ha permitido implementar aquellas expuestas en las secciones 5.4.1 y 5.4.2. Además, algunas de las funciones, incluidos los modelos de BCI, pueden aplicarse tanto en experimentos online como offline. Por lo tanto, MEDUSA[©] se puede utilizar para el análisis con bases de datos públicas, o en tiempo real. De hecho, MEDUSA[©] Platform se basa en esta librería para el procesamiento de señales. Esta es una característica fundamental que permite reproducir exactamente los mismos resultados obtenidos durante la experimentación online que offline, facilitando así la reproducibilidad experimental (Santamaría-Vázquez et al., 2023).

MEDUSA[©] Platform

MEDUSA[©] Platform es una aplicación de escritorio, desarrollada en Python, que ofrece funcionalidades avanzadas para llevar a cabo experimentos en neurociencia. Cuenta con una interfaz gráfica moderna y funciones de adquisición de señales en tiempo real, así como gráficos actualizados al momento. Una de sus características más importantes es la capacidad de instalar y crear aplicaciones que implementan experimentos y paradigmas en el campo de la neurociencia y BCI. Es importante destacar que todas estas funciones dependen del MEDUSA[©] Kernel (Santamaría-Vázquez et al., 2023). De esta manera, esta arquitectura se puede dividir en tres módulos principales: adquisición de señales, gráficos en tiempo real y aplicaciones. Cada uno de estos módulos desempeña un papel esencial durante el proceso experimental, como se observa en la figura 7.4.

- **Adquisición de la señal:** ofrece funciones avanzadas para gestionar equipos de medición fisiológica en tiempo real a través de LSL. Este protocolo de código abierto permite registrar series temporales de forma estandarizada al realizar la sincronización temporal y la transmisión de datos en tiempo real. En MEDUSA[©] Platform, se combinan estas tareas implementadas por los desarrolladores de LSL, con funcionalidades de alto nivel. Esto permite que esta solución sea independiente del hardware de adquisición de datos. Esto es especialmente valioso en entornos de investigación donde se emplean una variedad de dispositivos, tanto comerciales, como no comerciales. Se profundizará más en este protocolo en la subsección 7.3.2.

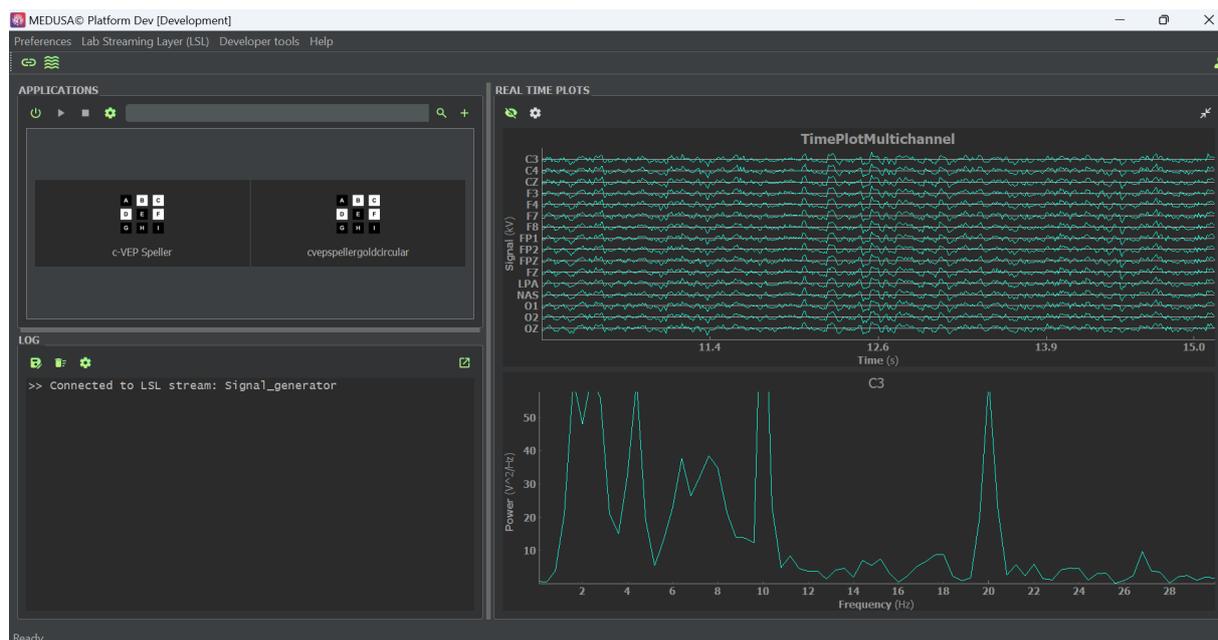


Figura 7.4: Ventana principal de MEDUSA[®] Platform. La vista está dividida en paneles para controlar las diferentes funcionalidades. Estos paneles son: aplicaciones (arriba a la izquierda), log de mensajes informativos (abajo a la izquierda) y gráficos en tiempo real (a la derecha). Además, hay más controles y configuraciones disponibles en las barras de tareas.

- **Gráficas en tiempo real:** Se ofrece varios gráficos para visualizar los flujos de LSL en tiempo real. Este panel es altamente personalizable, permitiendo adaptar los gráficos según las necesidades específicas de cada situación. Para su implementación, se emplea PyQtGraph, una biblioteca de Python de código abierto especialmente diseñada para representaciones en tiempo real, con un óptimo rendimiento y un consumo mínimo de recursos. Actualmente, este módulo es capaz de representar tanto en dominio temporal como en frecuencial, brindando así una amplia gama de opciones para analizar los datos en tiempo real (Santamaría-Vázquez et al., 2023).
- **Aplicaciones:** estos programas realizan tareas o paradigmas de estimulación y brindan retroalimentación en tiempo real mientras monitorean una o más señales. En la actualidad, se admiten aplicaciones basadas en Qt y Unity. En ambos casos, el flujo de trabajo de la aplicación se implementa en Python dentro de MEDUSA[®], pero la presentación de estímulos y retroalimentación se lleva a cabo mediante estos marcos. Qt es una biblioteca ampliamente utilizada para el diseño de interfaces gráficas de usuario, mientras que Unity es un potente motor gráfico que ofrece opciones avanzadas de control de GUI, modelado 3D y animaciones. A pesar de que el desarrollo de aplicaciones visuales y juegos es más complejo en Unity, proporciona un control preciso del tiempo sobre la tasa de actualización y la presentación de estímulos, lo que lo hace adecuado para aplicaciones avanzadas de BCI, como pueden ser los c-VEPs (Santamaría-Vázquez et al., 2023). Por ello, la aplicación desarrollada durante la

realización de este Trabajo de Fin de Grado se realizó en Unity.

7.3.2. Protocolos de comunicación

Los protocolos de comunicación más importantes utilizados en MEDUSA[©] son LSL para poder conectar, en nuestro caso, el EEG y TCP para garantizar la comunicación entre MEDUSA[©] Platform y Unity para el desarrollo de aplicaciones.

LSL

Lab Streaming Layer (LSL) es un ecosistema de middleware en red de código abierto para transmitir, recibir, sincronizar y grabar flujos de datos neurales, fisiológicos y conductuales adquiridos de diversos sensores. Esta tecnología permite a los investigadores trabajar con datos neurales, fisiológicos y de comportamiento de manera más eficiente y estandarizada, reduciendo la complejidad y las barreras técnicas (LSL, [Swartz Center for Computational Neuroscience](#)).

Los controladores de LSL para dispositivos de laboratorio reciben datos transmitidos a través de una red de área local (LAN) utilizando el protocolo UDP, con el fin de recopilar estos datos en una o varias computadoras dentro de la LAN. LSL luego puede almacenar los flujos de datos junto con marcadores de tiempo, lo que permite un análisis conjunto posterior de fenómenos síncronos en más de un flujo de datos, así como hacer que estos flujos estén disponibles para cálculos casi en tiempo real y/o para su visualización, mejorando así el control y la supervisión del experimento. Hasta el momento, se han desarrollado controladores para varios sistemas populares de EEG, rastreadores oculares y sistemas de captura de movimiento, entre otros. Aunque el código eficiente y de bajo nivel de LSL asigna con precisión el tiempo de llegada a las muestras de datos entrantes, es crucial considerar, calcular y aplicar los retrasos entre la entrada de datos del sistema de grabación (EEG, vídeo, etc.) y su recepción por LSL durante el análisis. Esto garantiza la máxima precisión en la sincronización entre flujos (Gramann et al., 2014). Aunque para muchos sistemas estos retrasos son insignificantes, en algunos casos, como en los BCIs basados en c-VEPs, es necesario tener una precisión de milisegundos.

La transmisión entre ordenadores en LSL no está encriptada bajo el supuesto de que los experimentos que implican datos sensibles ocurren en un entorno de red protegido. De esta manera, en caso de hacer uso de este protocolo en una red insegura, es conveniente configurar una Red Privada Virtual (VPN), que funciona incluso a través de Internet. En esta misma línea, también se podrían establecer conexiones remotas entre distintas redes, ilustrando así el potencial que tiene este protocolo para sistemas de este tipo. Evidentemente, en este caso, habría que configurar los *firewall* tanto del ordenador que produce el flujo de datos, como del receptor, y realizar una redirección de puertos en cada una de las redes.

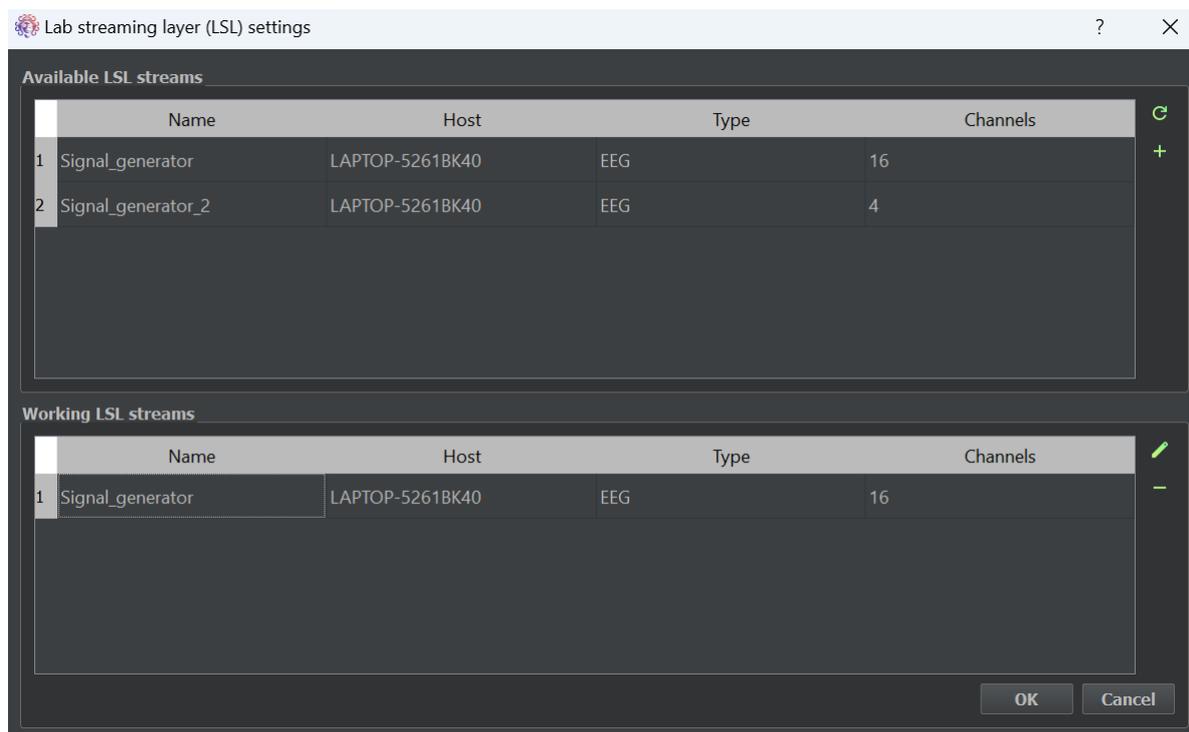


Figura 7.5: Panel de configuración de flujos LSL en MEDUSA[©] Platform. En concreto, aparecen dos posibles flujos LSL, que simulan ser dos EEG, llamados *Signal_generator* y *Signal_generator_2*, de los que sólo se ha conectado el segundo.

En concreto, para conectar un dispositivo, como un EEG, a MEDUSA[©] Platform, se requiere de un puente de LSL (es decir, un *script* o programa independiente) para recibir datos mediante la API específica del dispositivo en cuestión y enviarlos a través de LSL. Luego, se puede configurar MEDUSA[©] Platform para recibir el flujo, seleccionando el tipo de señal e información de los canales. Una vez configurado el flujo, estará disponible para su visualización en tiempo real y para aplicaciones. Es importante destacar que MEDUSA[©] Platform puede manejar varios flujos de LSL simultáneamente (véase la figura 7.5), lo que resulta útil para registrar de manera sincrónica diferentes señales o implementar BCIs colaborativas/competitivas (Santamaría-Vázquez et al., 2023).

TCP/IP

Otro de los protocolos más importantes usados en el ecosistema MEDUSA[©] es TCP/IP. En concreto, este se utiliza para garantizar la correcta comunicación entre las aplicaciones basadas en Unity, como la que se desarrolla en este Trabajo de Fin de Grado, y MEDUSA[©] Platform.

Transmission Control Protocol (TCP) es un protocolo confiable y orientado a la conexión, que incorpora, tanto control de flujo, como control de errores. Al igual que UDP,

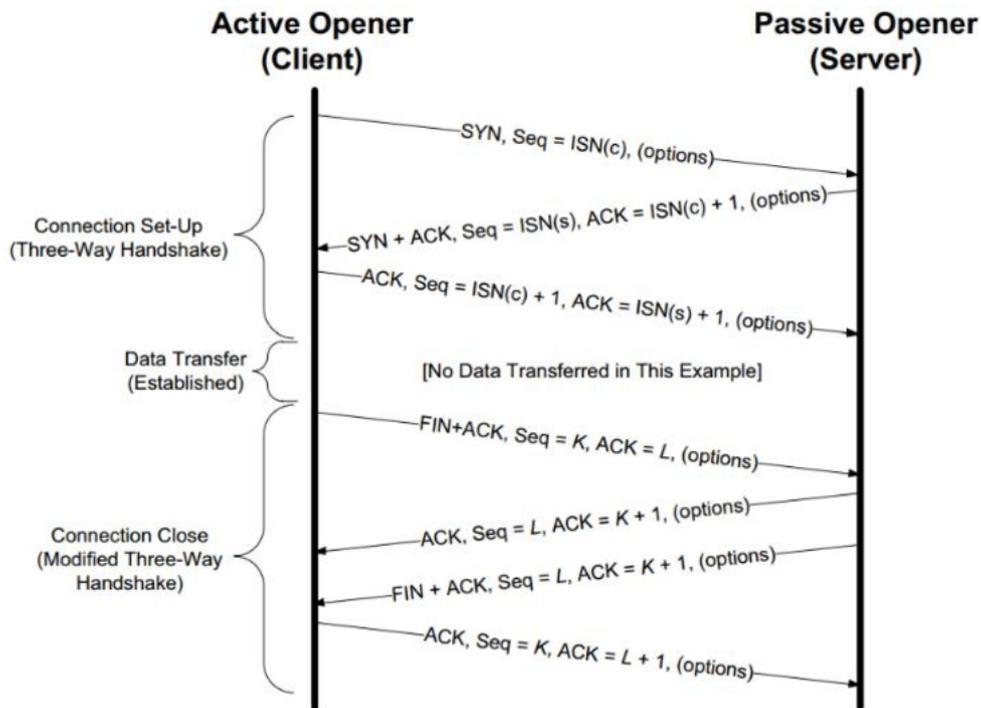


Figura 7.6: Esquema comunicación TCP/IP (Tyagi, 2020).

TCP utiliza números de puertos para garantizar la comunicación. Es un protocolo orientado a secuencias de datos, lo que implica que la información se transmite en forma de flujo de bytes, organizados en segmentos. TCP permite comunicación *full-duplex*, lo que significa que los datos pueden fluir en ambas direcciones simultáneamente. La conexión en TCP se establece mediante un mecanismo de *handshaking* de tres pasos: el remitente, o cliente, envía un segmento SYN para iniciar la conexión, el receptor, o servidor, responde con un SYN + ACK. Finalmente, el cliente responde al Servidor con un paquete ACK en el que confirma la recepción del paquete SYN + ACK, permitiendo así al remitente enviar los datos. Una vez establecida la conexión, tiene lugar la transferencia de datos, y finalmente, la conexión se termina utilizando el mismo proceso de *handshaking* de tres pasos, donde se emplea un segmento FYN para cerrar la conexión (Tyagi, 2020). Este proceso, se muestra de forma esquemática en la figura 7.6.

En concreto, en MEDUSA[©], se trata del protocolo TCP/IP asíncrono y bidireccional. Es decir, se permite la comunicación *full-duplex*, por lo que, una vez completado el *handshake*, la comunicación entre el cliente (Unity) y el servidor (MEDUSA[©] Platform) puede continuar, y ambos dispositivos están sincronizados y listos para intercambiar datos en ambos sentido de manera confiable. Si alguno de los dispositivos no recibe una respuesta adecuada en cualquiera de los pasos del handshake, se puede intentar nuevamente o finalizar la conexión. Este proceso garantiza una comunicación confiable y ordenada en la red. Para la correcta implementación de este protocolo, las aplicaciones de MEDUSA[©]

deben tener los siguientes archivos en Unity ([MEDUSA[©] Official Documentation](#)):

- `Manager.cs`: es la parte principal del proyecto en Unity. Controla el ciclo de vida del lado del cliente de la aplicación. Además, crea una instancia de la clase `MedusaTCPClient` para informar sobre el estado de la aplicación al servidor (MEDUSA[©] Platform). Este script proporciona una descripción detallada de cada elemento que lo compone, así como de sus funciones.
- `MedusaTCPClient.cs`: se encuentra la implementación de la clase `MedusaTCPClient`, la cual establece la conexión con el servidor TCP. Esta conexión permite que la aplicación envíe y reciba mensajes de forma asíncrona.
- `MessageInterpreter.cs`: en este archivo se define la clase `MessageInterpreter`, la cual se encarga de decodificar los distintos mensajes enviados desde el servidor de la aplicación.

7.3.3. Aplicaciones dentro de MEDUSA[©]

Las aplicaciones de MEDUSA[©] son los componentes principales de MEDUSA[©] Platform. Estos programas pueden implementar multitud de experimentos diferentes mientras se monitorea una o más señales. En concreto, se caracterizan por lo siguiente:

- El participante tiene un objetivo claro, en este caso, controlar un sistema BCI o realizar una tarea cognitiva.
- Se registran una o más señales biológicas, como EEG, electrocardiografía, electromiografía, electrooculografía, fNIRS, entre otras.
- La precisión a lo hora de la sincronización fundamental para identificar los diferentes eventos en los registros, para garantizar un correcto funcionamiento del experimento.

Estas aplicaciones cuentan con una arquitectura y flujo de trabajo predefinidos que permiten implementar cualquier experimento que cumpla con estas características ([MEDUSA[©] Official Documentation](#)).

Workflow y estados internos

A priori, el ciclo de vida de las aplicaciones debe estar definido a partir de unos posibles estados internos, que deben ser comunes a todas las aplicaciones. Estos estados están representados por dos variables:

- `app_state`: representa el estado de la aplicación, cambiando entre cuatro valores posibles: `OFF`, `POWERING ON`, `POWERING OFF` y `ON`. La aplicación está en el estado `OFF` cuando aún no se ha iniciado y en el estado `ON` cuando está preparada para comenzar una ejecución. Los estados restantes son transiciones entre `ON` y `OFF`.

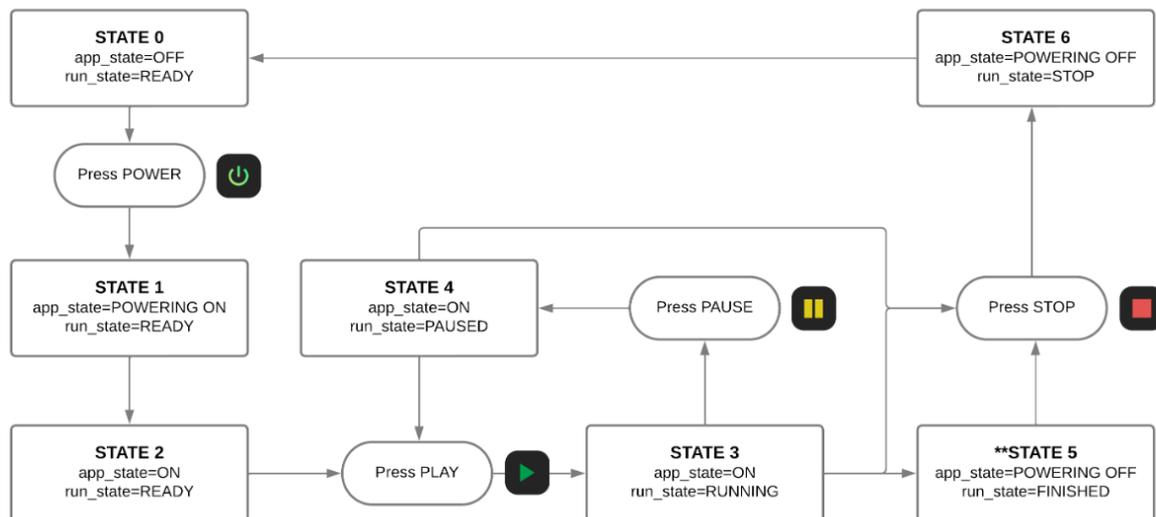


Figura 7.7: Ciclo de vida y workflow de una aplicación en MEDUSA[©] Platform. ** Este estado se alcanza automáticamente cada vez que una ejecución finaliza. Si una ejecución no tiene un final definido, sino que es el usuario quien debe detener la aplicación, este estado no se alcanzará (MEDUSA[©] Official Documentation).

- **run_state**: indica el estado de cada ejecución y puede tomar cinco valores: READY, RUNNING, PAUSED, STOP y FINISHED.

La figura 7.7 representa de forma gráfica el workflow de las aplicaciones MEDUSA[©]. Ambas variables, **app_state** y **run_state**, son objetos cuyo valor se comparte entre todos los hilos y procesos en el ciclo de vida de la aplicación (MEDUSA[©] Official Documentation).

Arquitectura y componentes principales

Las aplicaciones de MEDUSA[©] cuentan con una arquitectura predefinida que incluye una serie de componentes para facilitar la implementación de experimentos personalizados (MEDUSA[©] Official Documentation). Los más importantes son:

- **Clase App**: es el componente central de una aplicación. Implementa el flujo de trabajo principal de la aplicación, que ejecuta las siguientes tareas:
 1. Configura todos los hilos de ejecución secundarios, incluidos los LSL workers, el hilo manager y los hilos de comunicación.
 2. A continuación, se ejecuta la interfaz gráfica, bloqueando el proceso hasta que se cierre dicha interfaz.
 3. Finalmente, cierra los hilos auxiliares y libera todos los recursos para finalizar la ejecución.

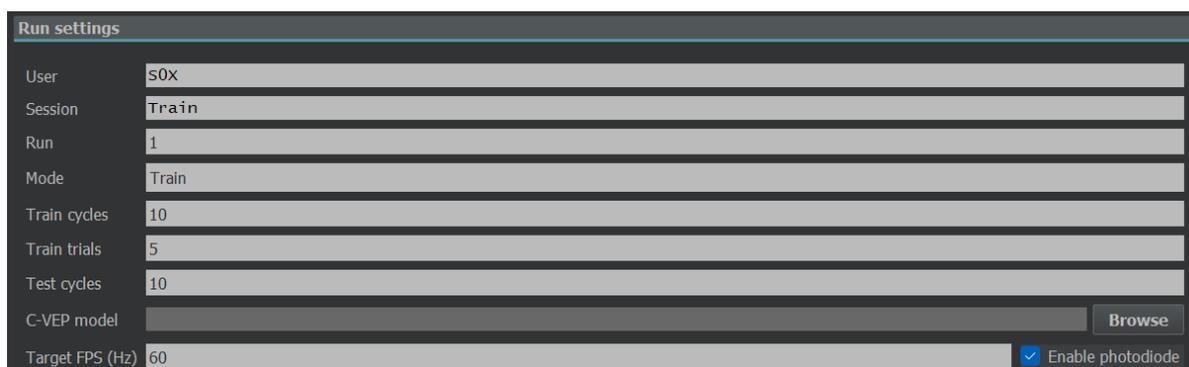


Figura 7.8: Uno de los menús de configuración de la aplicación c-VEP Speller de MEDUSA[®].

- **LSL workers:** para cada flujo LSL del entorno de trabajo, el proceso de App crea un hilo que escucha nuevos flujos entrantes. Cada trabajador LSL mantiene los datos del flujo correspondiente, a los que se puede acceder mediante funciones *thread-safe* desde todos los demás hilos de la aplicación.
- **Hilo Manager:** este hilo espera eventos desde la interfaz gráfica. Una vez que se recibe un evento, lo maneja ejecutando las tareas apropiadas dependiendo de cada aplicación. También tiene acceso a los datos de los LSL workers en caso de que la interfaz requiera alguna tarea relacionada con el procesamiento de señales.
- **Interfaz gráfica:** implementa el código de la interfaz gráfica y el flujo del experimento con un método bloqueante que devuelve los datos en cuestión exclusivamente cuando se completa la tarea. Para el desarrollo de estas aplicaciones, en consonancia con la interfaz gráfica, es común crear hilos auxiliares para controlar el flujo del experimento.
- **Interfaz de comunicación:** implementa el protocolo de comunicación entre la interfaz gráfica y el hilo Manager para controlar el flujo de trabajo de la aplicación. Dependiendo de la interfaz gráfica que estemos utilizando (PyQt o Unity), la interfaz de comunicación cambia sustancialmente. En particular, para aplicaciones desarrolladas en Unity, se implementa el protocolo TCP expuesto en la subsección anterior.
- **Clase Settings:** contiene la información de configuración de la aplicación (por ejemplo, número de ejecuciones y trials, etc.) como se puede apreciar en la figura 7.8.

Capítulo 8

Protocolo de evaluación

Este capítulo trata de ilustrar al lector con el protocolo de evaluación llevado a cabo en el análisis online realizado. En concreto, se explicará la fase de calibración y de test de los tres experimentos llevados a cabo, así como el cuestionario de satisfacción utilizado.

8.1. Montaje

Con el objetivo de comprobar el funcionamiento de TRCA tanto con *Circular Shifting* como con Gold Codes, se va realizar un análisis tanto offline como online. En el primero, se utilizarán los algoritmos de TRCA y CCA con dos bases de datos obtenidas de [Martínez-Cagigal et al. \(2023\)](#) y [Thielen et al. \(2021\)](#) y tiene que objetivo comparar el rendimiento de CCA y TRCA en paradigmas de *Circular Shifting* y de Gold Codes. En cuanto al análisis online, su propósito general es testear si el desarrollo realizado puede aplicarse en un procesado en tiempo real tanto en *Circular Shifting* como Gold Codes y analizar los resultados obtenidos. Para ello, han participado en este estudio un total de 10 sujetos sanos (media de edad: $29,2 \pm 13,95$, 7 hombres, 3 mujeres), donde todos ellos han dado su consentimiento informado para participar. El análisis online consiste de un total de tres experimentos, uno de ellos explicado en la sección 8.2 y los otros dos en la sección 8.3, todos ellos utilizando una frecuencia de refresco de 120 Hz.

La experimentación online comienza colocando el casco del EEG en la cabeza del usuario, comprobando que esté correctamente centrado y que los electrodos estén lo más cerca del cuero cabelludo posible (véase la figura 8.1). A continuación, es necesario ajustar el casco a través de una correa para evitar que este se mueva durante el experimento. Posteriormente, se aplica gel electrofisiológico en cada uno de los electrodos, incluyendo la referencia colocada en una de las orejas para evitar interferencias e impedancias en la señal. Finalmente, se conecta el casco a MEDUSA[©] y se comprueba que la señal que está recogiendo en tiempo real es correcta y no existe ningún tipo de interferencia que pueda afectar negativamente al experimento. Para ello, se suele pedir al usuario que haga fuerza con los dientes o cierre los ojos, para poder observar esos cambios en las gráficas del EEG.



Figura 8.1: Uno de los sujetos antes de realizar el experimento online.

En cuanto al material utilizado, se recuerda que es el siguiente:

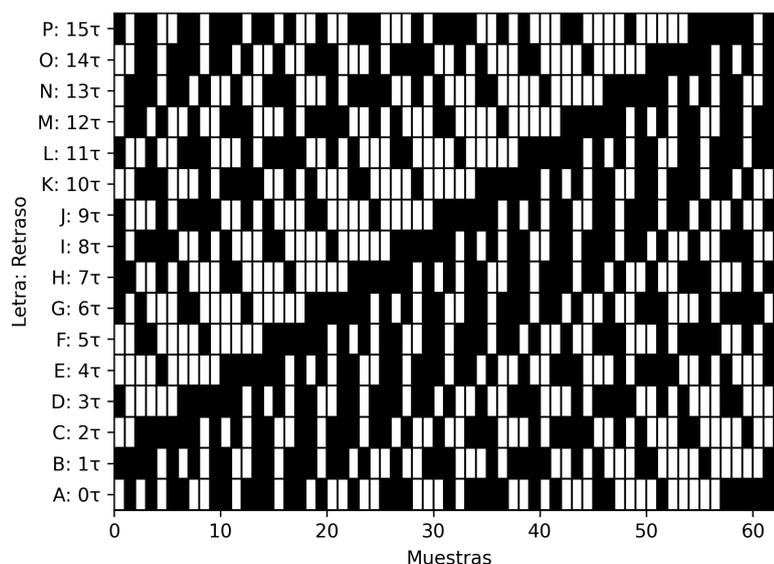
- El amplificador g.USBamp con una frecuencia de muestreo de 256 Hz.
- 16 electrodos g.LADYbird en las posiciones Oz, F3, Fz, F4, I1, I2, C3, Cz, C4, CPz, P3, Pz, P4, PO7, POz y PO8 según el estándar 10-10. El electrodo tierra se colocó sobre AFz y la referencia en el lóbulo de la oreja.
- g.GAMMAbox para comunicar los electrodos con el amplificador
- Gel electrofisiológico de Neuroline.

8.2. Experimento *Circular Shifting*

Para la realización de este experimento, que de ahora en adelante llamaremos experimento *Circular Shifting*, se ha utilizado la aplicación desarrollada por el GIB de *c-VEP Speller* (https://medusabci.com/market/cvep_speller/). Esta aplicación proporciona un *speller* BCI basado c-VEP bajo el paradigma de *Circular Shifting*. Permite desarrollar BCIs de alta velocidad y confiabilidad para comunicación y control al codificar los comandos de la aplicación usando versiones desplazadas de secuencias pseudoaleatorias. En concreto, se han utilizado 63 bits para codificar cada uno de los comandos a través de desplazamientos circulares de la m-secuencia de la tabla 8.1, obtenida a través de LFSR del polinomio primitivo de grado 6 $X^6 + X^5 + 1$. Se quieren codificar un total de 16

Tabla 8.1: m-secuencia elegida para la codificación de los comandos para el experimento de *Circular Shifting*.

m-secuencia (63 bits)
0101100110111011010010011100010111100101000110000100000111111

Figura 8.2: Codificación de matriz de comandos con *Circular Shifting* de 63 bits. En el eje vertical aparece tanto el comando como el desplazamiento con respecto a la secuencia original (representada por 0τ), donde $\tau = 4$.

comandos que se corresponden con las letras de la A a la P, sin incluir la \tilde{N} , por ello, el máximo retraso o *lag* que se puede escoger es de $\tau = 4$ (véase la figura 8.2). Finalmente, como la frecuencia de refresco del monitor utilizado es de 120 Hz, la duración de cada ciclo, es decir, la visualización de una secuencia completa, será de $63/120 = 0,525$ s.

8.2.1. Fase de calibración

Como se explicó en la sección 5, los c-VEP son señales exógenas generadas de manera natural por nuestro cerebro en respuesta a estímulos visuales. Por esa razón, los BCIs basados en c-VEP no requieren que los usuarios reciban entrenamiento, solo una pequeña calibración. Por ello, el protocolo de evaluación comienza por un ejercicio en el que se va a iluminar únicamente un comando con la codificación de la letra A, que recuerde que se corresponde con la m-secuencia original sin desplazamientos (véase la figura 8.3).

Un trial se compone de una serie de repeticiones de la misma secuencia (ciclos) para

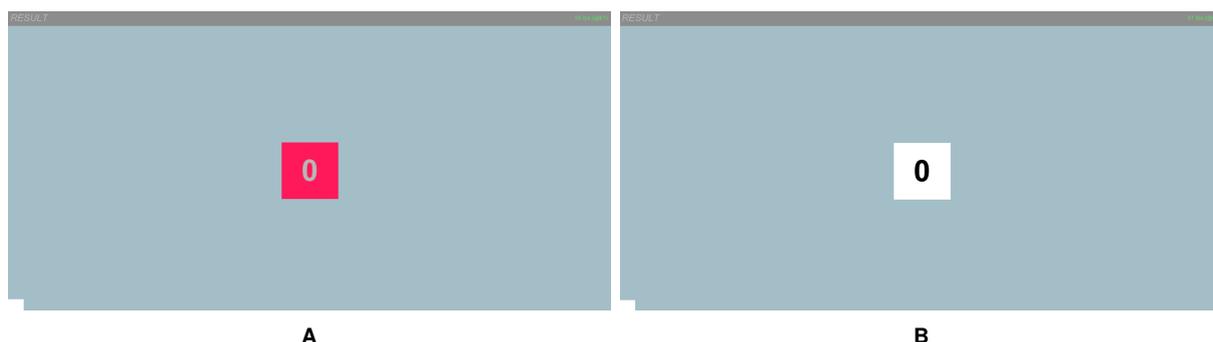


Figura 8.3: Interfaz gráfica de la fase de calibración de la aplicación utilizada en el experimento *Circular Shifting*. La subfigura **A** muestra el comando iluminado de magenta para fijar la atención del usuario. La subfigura **B** muestra la iluminación del comando en un frame.

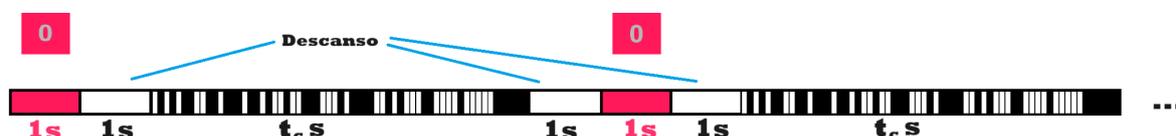


Figura 8.4: Esquema de la fase de calibración de la aplicación utilizada en el experimento *Circular Shifting*. t_c representa el tiempo de 10 ciclos, que en este caso es 5,25 s.

registrar el ERP de una forma confiable, ya que, como se mencionó anteriormente, el promedio de muchas repeticiones del mismo ERP permite extraer patrones, anulando la actividad de fondo y fortaleciendo así la la respuesta neuronal relacionada con la generación de dicho ERP. Por consiguiente, el sujeto tendrá que fijar la vista durante 10 ciclos por cada trial, con un total de 5 trials. Cada ciclo se corresponde a la iluminación de una secuencia completa. Como trata de ilustrar la figura 8.4, se ilumina inicialmente el comando A de color magenta, con el objetivo de indicar al usuario de que debe fijar la vista en él. A continuación, la letra empieza a parpadear siguiendo la secuencia por la que está codificada durante un total de 5,25 s, es decir 10 ciclos \times 0,525 s/ciclo. Esto completaría el primer trial. Posteriormente, se permite 1s de descanso y, a continuación, se vuelve a marcar la letra de color magenta para volver a indicar al usuario que debe fijar la vista en ella y se repite el mismo proceso hasta completar 5 trials. Tras un sencillo cálculo, la duración de la fase de calibración es de 40,25 s. Sin embargo, se repitió este proceso 3 veces más para garantizar una fase de calibración lo suficientemente precisa, por lo que el tiempo de la fase de calibración en total fue de 161 s. Con los datos obtenidos de esta etapa, se entrena un modelo como se explicó en el capítulo 5, a través del algoritmo CCA.

8.2.2. Fase de test

Una vez realizada la fase de calibración y tras haber entrenado el modelo, es necesario probar el funcionamiento de la aplicación en la fase de test. Para ello, como muestra la

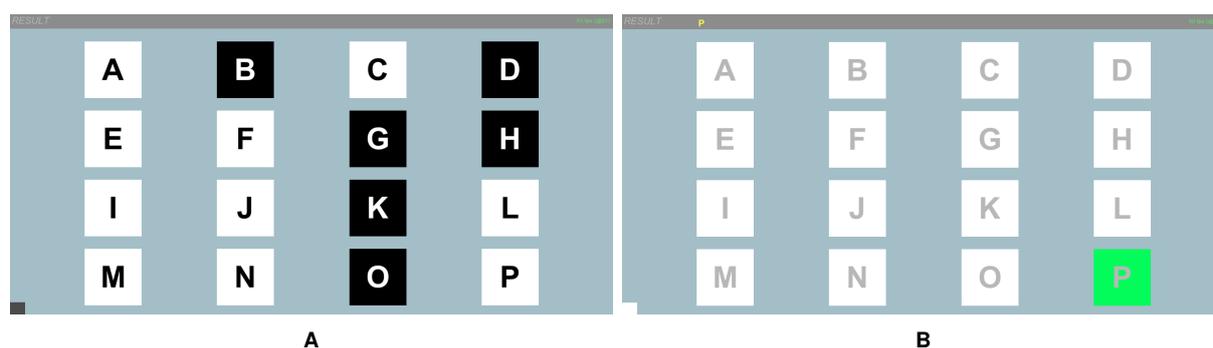


Figura 8.5: Interfaz gráfica de la fase de test de las aplicaciones utilizadas en los experimentos *Circular Shifting* y *Gold-Circular*. La subfigura **A** muestra la matriz de comandos en un frame de ejecución. La subfigura **B** muestra la submatriz de comandos en la que se ha seleccionado la letra P.

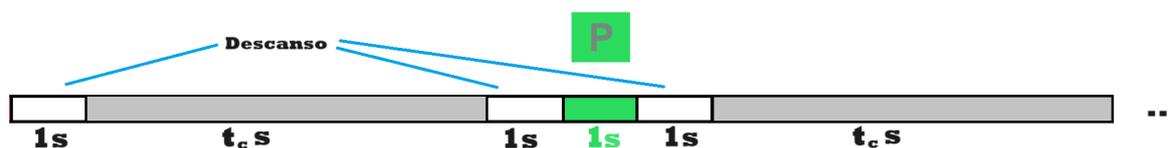


Figura 8.6: Esquema de la fase de test de los experimentos *Circular Shifting* y *Gold-Circular* en la que t_c representa el tiempo de duración de 10 ciclos. Se observa que en este caso se ha seleccionado la letra P.

figura 8.5, se muestra al usuario una matriz 4×4 de comandos. Esta matriz, como se explicó anteriormente, está codificada a través de desplazamientos temporales de 4 muestras de la m -secuencia original, como muestra la figura 8.2. La aplicación se encarga de determinar, en tiempo real, el comando al que está mirando el usuario.

En contraste con la fase de calibración, en el test se iluminan todos los comandos a la vez, siguiendo cada uno su secuencia pseudoaleatoria según estén codificados. Estos parpadearán durante 10 ciclos, es decir, durante un total de 5,25 s. Como muestra la figura 8.6, tras este parpadeo, MEDUSA[©] recibirá los datos recogidos a través del EEG y los clasificará para decodificar las intenciones de los usuarios. A continuación, MEDUSA[©] envía a la aplicación en Unity el comando seleccionado, que se iluminará en verde. Finalmente, se deja 1s de descanso al sujeto, y los comandos vuelven a parpadear otra vez, comenzando de nuevo el mismo proceso.

Es imprescindible que se reduzca el número de artefactos, tanto en la calibración como en el test, por lo que se pidió a los usuarios que no hicieran fuerza con el cuello o la mandíbula, o que no pestañearan. Por ello, se ha pedido a los 10 sujetos que participan en el análisis online que deletreen todas las letras de la A a la P, dos veces. Es decir, en el mejor de los casos, el usuario debería de deletrear ABCDEFGHIJKLMNOPABCDEFGHIJKLMNPNOP.

Tabla 8.2: Gold Codes seleccionados para codificar la matriz de comandos en el experimento online con 31 bits.

Nº	Gold Code (31 bits)
1	1000111001101101101011100001100
2	1100001001000000111010010110010
3	1110010001010110010010101101101
4	0111011101011101000110110000010

8.3. Experimento *Gold-Circular*

Este experimento tiene como objetivo comprobar la viabilidad del algoritmo TRCA tanto con *Circular Shifting* como con los Gold Codes. Para ello, se ha desarrollado e implementado la aplicación *c-VEP Speller Gold-Circular* en MEDUSA[©], la cual combina de forma precisa la técnica de *Circular Shifting* con las propiedades de correlación cruzada excelentes de los Gold Codes. Para codificar las secuencias del c-VEP, se han escogido dos longitudes de 63 y 31 bits, respectivamente. De ahora en adelante, se conocerá a estos experimentos como *Gold-Circular* con 31 y 63 bits. Se codificarán un total de 16 comandos, que se corresponden con las letras de la A a la P, sin incluir la Ñ. En ambos experimentos se han seleccionado 4 Gold Codes y cada uno de ellos se ha desplazado para obtener un 4 secuencias por cada Gold Code. Veamos en profundidad cada uno de estos experimentos.

31 bits

En cuanto al experimento con 31 bits, se han utilizado el par preferido de polinomios dado por $X^5 + X^4 + X^3 + X^2 + 1$ y $X^5 + X^2 + 1$ y un estado inicial de $[1, 1, 1, 1, 1]$ para la codificación de los comandos. De los 31 Gold Codes que se obtienen, se han seleccionado 4 de ellos, mostrados en la tabla 8.2. Para cada una de estas secuencias Gold, se aplica *Circular Shifting* para obtener versiones desplazadas de la original con un retraso o *lag* de $\tau = 7$, codificando así el resto de comandos, como muestra la figura 8.7. Nótese que, siguiendo la misma notación de la sección 5.4.2, $N_G = 4$ y $m = 4$, logrando codificar así 16 comandos. Finalmente, es necesario recalcar que, como la frecuencia de muestro del monitor utilizado es de 120 Hz, la duración de cada ciclo será de $31/120 = 0,2583$ s.

63 bits

Para el segundo experimento de esta sección, se utilizan 63 bits para codificar los comandos. Para ello, se ha utilizado el par preferido de polinomios de grado 6 dado por $X^6 + X^5 + 1$ y $X^6 + X^5 + X^4 + X^3 + X + 1$ y $[1, 1, 1, 1, 1, 0]$ como estado inicial. Como antes, de los 63 Gold Codes que se obtienen, se han escogido 4 de ellos, representados en la tabla 8.3. En este caso, al ser la longitud de cada secuencia más larga, se puede escoger

Tabla 8.3: Gold Codes seleccionados para codificar la matriz de comandos en el experimento online con 63 bits.

Nº	Gold Code (63 bits)
1	010010111111101101001001110100010000001011011100011101101100010
2	101001111110010100000011010010110001001110110011101000011101110
3	110100011110101000100110000001100001101100000100010010100101000
4	111010101110110110110100101000001001111101011111101111111001011

un *lag* de $\tau = 15$, logrando codificar así los 16 comandos de la matriz (véase la figura 8.8). Finalmente, la duración de cada ciclo será de $63/120 = 0,525$ s.

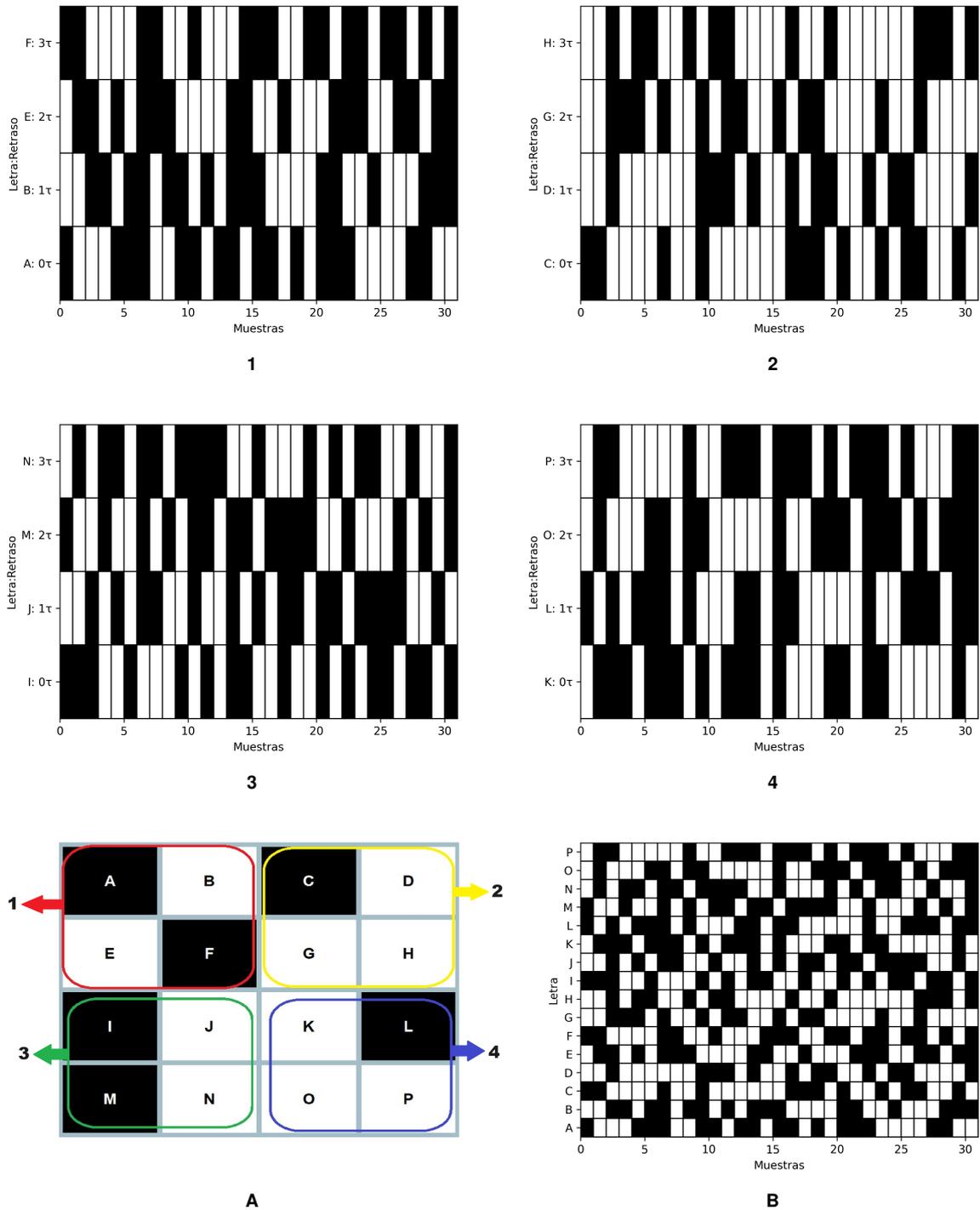


Figura 8.7: Codificación de los comandos en el experimento *Gold-Circular* 31 bits. Cada una de las subfiguras individuales 1,2,3 y 4 representa la codificación del Gold Code, marcado por 0τ , así como sus versiones desplazadas por $\tau = 7$, junto con la letra que va a codificar. El número de estas subfiguras se corresponde con el número de Gold Code de la tabla 8.2. La subfigura A representa la matriz de comandos dividida por zonas según el Gold Code a partir del cual están codificados. La zona roja se corresponde con la subfigura 1, la amarilla con con la subfigura 2, la verde con la subfigura 3 y la azul con la subfigura 4. La subfigura B representa la codificación global de la matriz de comandos.

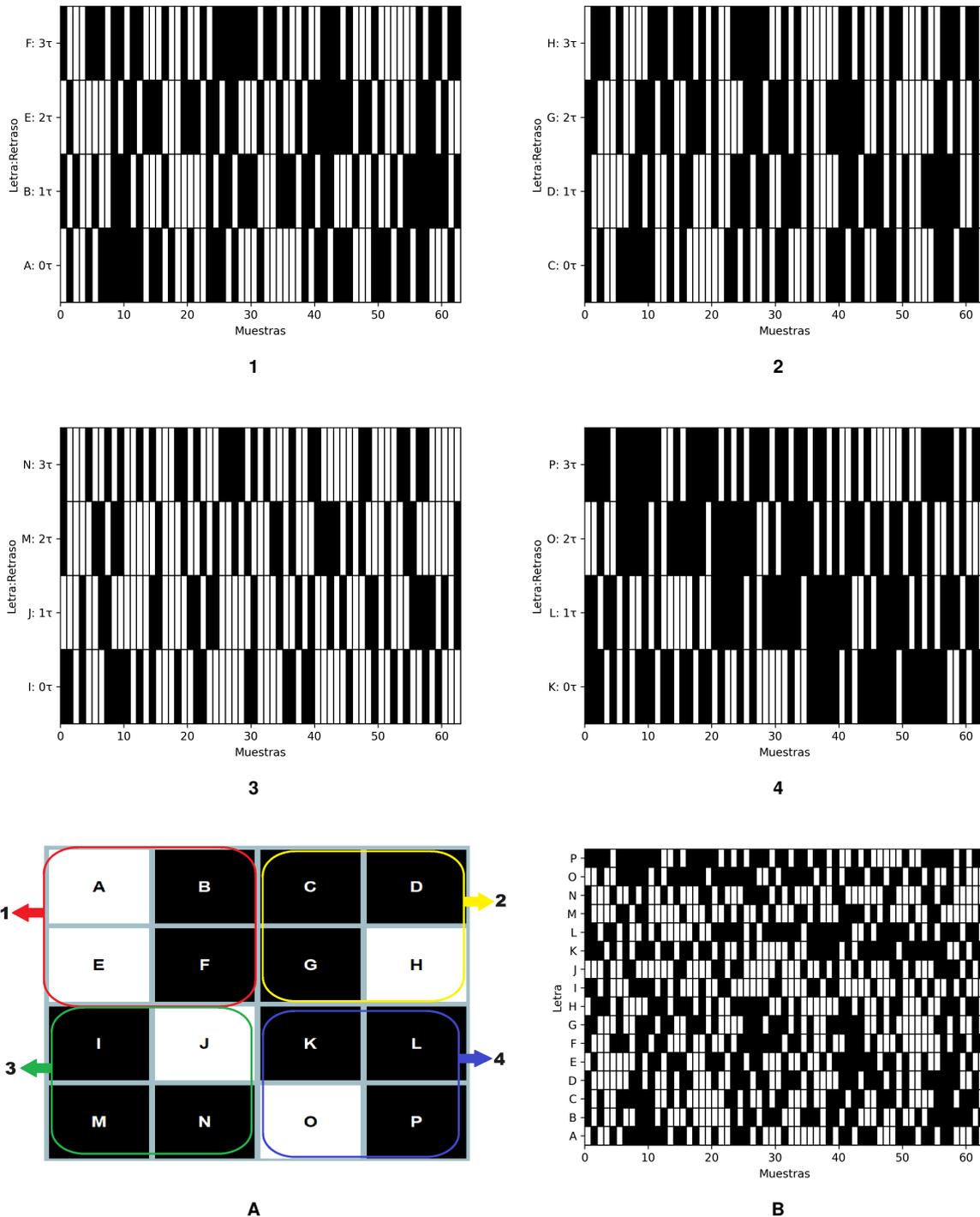


Figura 8.8: Codificación de los comandos en el experimento *Gold-Circular* 63 bits. Cada una de las subfiguras individuales **1,2,3** y **4** representa la codificación del Gold Code, marcado por 0τ , así como sus versiones desplazadas por $\tau = 15$, junto con la letra que va a codificar. El número de estas subfiguras se corresponde con el número de Gold Code de la tabla 8.3. La subfigura **A** representa la matriz de comandos dividida por zonas según el Gold Code a partir del cual están codificados. La zona roja se corresponde con la subfigura **1**, la amarilla con con la subfigura **2**, la verde con la subfigura **3** y la azul con la subfigura **4**. La subfigura **B** representa la codificación global de la matriz de comandos.

Juan Duenas Ruiz

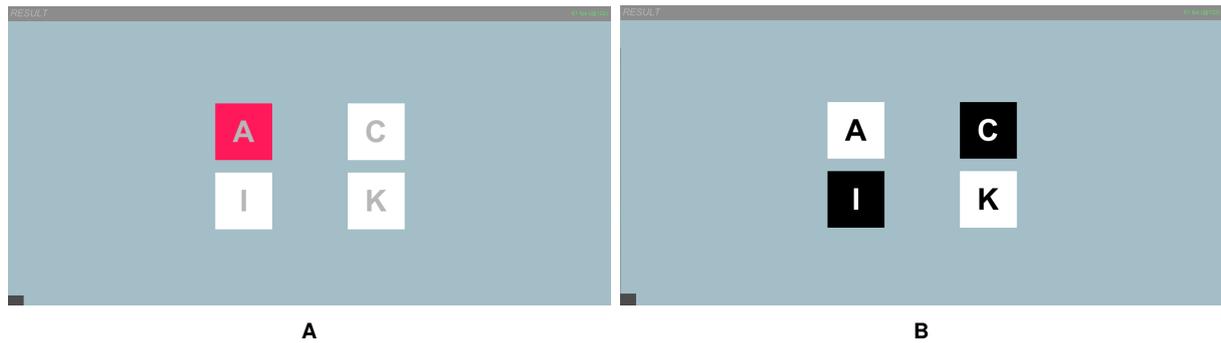


Figura 8.9: Interfaz gráfica de la fase de calibración de la aplicación desarrollada para los experimentos *Gold-Circular* con 31 y 63 bits. La subfigura **A** muestra el comando iluminado de magenta para fijar la atención del usuario. La subfigura **B** muestra la iluminación del comando en un frame.

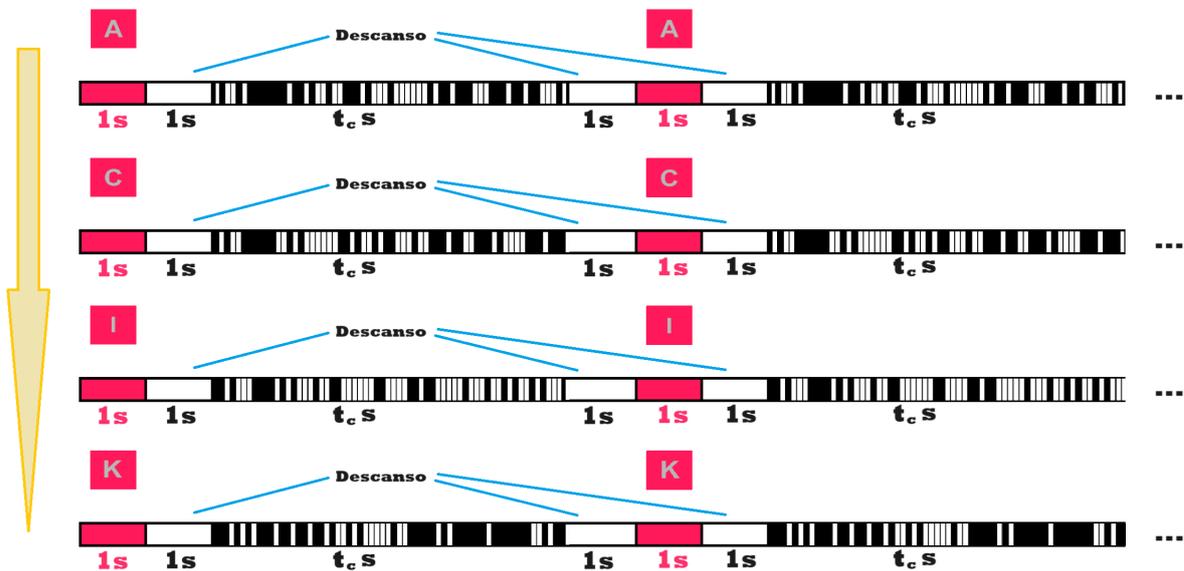


Figura 8.10: Esquema de la fase de calibración de la aplicación desarrollada para los experimentos *Gold-Circular* con 31 y 63 bits. t_c representa el tiempo de 10 ciclos. Si la longitud de la secuencia es de 63 bits será de 5,25 s pero si es de 31 bits será de 2,58 s. Nótese que en la figura las secuencias para cada letra son distintas y de 63 bits. El caso con 31 bits es completamente análogo.

8.3.1. Fase de calibración

La fase de calibración para estos dos experimentos es muy parecida a la explicada en la sección 8.2.1 pero replicada 4 veces. En este caso, como se ha realizado *Circular Shifting* para cada uno de los Gold Codes, es necesario realizar una fase de calibración para cada uno de ellos. En particular, como muestra la figura 8.9, esta fase comienza con una matriz de 4 comandos con las letras A,C,K e I, una por cada Gold Code. Inicialmente, se coloreará la letra A de color magenta para indicar al sujeto que debe prestar atención mirando fijamente este comando. A continuación, tras un segundo de descanso en el que

el comando está blanco, los cuatro comandos empiezan a parpadear durante 10 ciclos con una duración de t_c segundos. Para el experimento de 31 bits, $t_c = 2,58$ s y para 63 bits, $t_c = 5,25$ s. A continuación, se deja un segundo de descanso al usuario y este proceso se repite 4 veces más, logrando así un total de 5 trials. De esta manera, se habría entrenado sólo uno de los Gold Codes, por lo tanto, a continuación se marcará de magenta el comando de la letra C para que el usuario fije su atención en él y se repite el mismo proceso descrito. Lo mismo ocurriría para el comando de la letra I y para el comando de la letra K. De esta manera, la duración total de esta fase es cuatro veces más que la descrita en la sección 8.2.1, durando un total de 161 s para el experimento con 63 bits y 112 s para el experimento con 31 bits. Como antes, se realizarán un total de 4 entrenamientos, por lo que la duración de la fase de calibración es de 10 minutos y 44 segundos para 63 bits y 7 minutos y 28 segundos para 31 bits. La figura 8.10 muestra de forma esquemática la fase de calibración de la aplicación desarrollada para los experimentos *Gold-Circular*.

8.3.2. Fase de test

La fase de test del experimento *Gold-Circular* es exactamente igual a la explicada en 8.2.2, pero cambiando única y exclusivamente la codificación de los comandos dada por las figuras 8.7 y 8.8 para los experimentos con 31 y 63 bits, respectivamente.

8.4. Cuestionario de Satisfacción

Una vez finalizado el protocolo de evaluación, se pidió a cada uno de los sujetos participantes que rellenasen un cuestionario de satisfacción con información acerca del experimento. El objetivo de este cuestionario es registrar la impresión, el efecto y la usabilidad que ha tenido la aplicación sobre los usuarios para posteriormente analizar el nivel de satisfacción durante su uso. Este cuestionario se ha llevado a cabo mediante *Google Forms*, formulando las preguntas de manera que el usuario debe responder asignando una puntuación del 1 al 5, según su grado de acuerdo, donde 1 indica “totalmente en desacuerdo” y 5 “totalmente de acuerdo”. Esta metodología se conoce como una escala de Likert de cinco puntos.

Asimismo, se ha tenido en cuenta el sesgo de acquiescencia al alternar afirmaciones positivas y negativas a la hora de realizar las preguntas a los usuarios. También conocido como sesgo de acuerdo, hace referencia a la tendencia de los encuestados a estar de acuerdo con las afirmaciones positivas presentadas en un cuestionario, independientemente de su contenido. Esto puede llevar a respuestas sesgadas y afectar la validez de los resultados de la encuesta (Kemmelmeier, 2016).

Finalmente, es necesario recalcar que el orden en el que se realizaban los experimentos *Circular Shifting* y *Gold-Circular* con 31 y 63 bits se ha aleatorizado por sujeto con el objetivo de evitar el sesgo de un posible cansancio.

Las preguntas incluidas en el formulario son las siguientes:

- Me ha resultado interesante conocer y usar este sistema BCI.
- Me ha resultado difícil seleccionar los comandos deseados en el experimento *Gold-Circular*.
- Me ha resultado difícil seleccionar los comandos deseados en el experimento *Circular Shifting*.
- Me imagino la utilización de este tipo de sistemas en la vida cotidiana de ciertas personas.
- El sistema responde con suficiente velocidad y fluidez.
- La duración de la sesión me ha parecido excesiva.
- Estaría dispuesto a participar en un estudio de estas características.
- Los estímulos visuales del experimento *Gold-Circular* me han causado fatiga visual.
- Los estímulos visuales del experimento *Circular Shifting* me han causado fatiga visual.
- El paradigma de selección de comandos es intuitivo y fácil de entender.

Capítulo 9

Resultados del experimento

En este capítulo se presentan los resultados obtenidos tras los análisis offline y online.

9.1. Análisis offline

El análisis offline se ha utilizado para comprobar la viabilidad y la efectividad del algoritmo TRCA frente a CCA en distintos casos de uso. En concreto, se ha realizado una comparativa de ambos algoritmos con dos bases de datos de los estudios de [Martínez-Cagigal et al. \(2023\)](#) y [Thielen et al. \(2021\)](#).

9.1.1. Base de datos de p -ary m -secuencias

[Martínez-Cagigal \(2020\)](#) en su estudio realiza una comparativa de un sistema BCI utilizando secuencias de codificación generadas a partir de cierto LFSR, pero no únicamente binarias, sino de base p , con $p \in \{2, 3, 5, 7, 11\}$, lo que se conoce como p -ary m -secuencias. La figura 9.1 muestra un resumen de los polinomios primitivos utilizados para generar dichas secuencias. En este artículo se realizan desplazamientos de dichas secuencias a través de *Circular Shifting* para codificar un total de 16 comandos. Se realizan un total de 5 experimentos, uno para cada valor de p , con un total de 16 usuarios. Finalmente, en este artículo se demostró que cuanto mayor es la base p , mayor es la comodidad de los usuarios al utilizar el sistema BCI y por tanto, menor es la fatiga visual.

Una vez introducido las novedades de este artículo, las tablas 9.1, 9.2, 9.3, 9.4 y 9.5 muestran las precisiones de los resultados por usuarios y por ciclos obtenidos tras aplicar CCA y TRCA, así como la media y la desviación típica para cada ciclo por usuarios para ambos algoritmos. En estas tablas también se incluye el p -valor resultante tras comparar las precisiones por ciclos de CCA y TRCA a través del test de Wilcoxon. Se puede observar que la media para el ciclo 10 de ambos algoritmos en todas las tablas es mayor que 98,05 %. Asimismo, se observa que, para cada una de las bases, la precisión media obtenida en el último ciclo es exactamente igual que para CCA y TRCA (véase la tabla 9.6). Finalmente, se aprecia que ninguno de los p -valores obtenidos es estrictamente menor que 0,05.

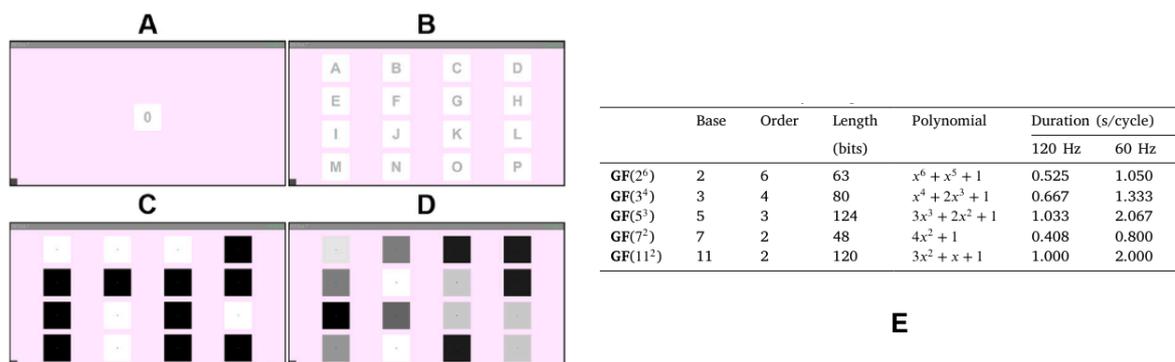


Figura 9.1: Instantáneas de la aplicación utilizada en el artículo [Martínez-Cagigal et al. \(2023\)](#). **(A)** Etapa de calibración, en la que se pide a los usuarios que presten atención al comando central, que parpadea de acuerdo con p -ary m-secuencia original. **(B)** Disposición en línea, en la que los usuarios prestan atención al comando objetivo para realizar tareas de escritura. **(C)** Durante el modo en línea, las etiquetas de los comandos se reemplazan por un pequeño punto centrado que ayuda a los usuarios a enfocarse en el comando deseado. En esta instantánea, se muestra para cuando $p = 2$. **(D)** Ejemplo de instantánea de una estimulación cuando $p = 11$. **(E)** Resumen de la codificación de los comandos utilizada.

Tabla 9.1: Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de [Martínez-Cagigal et al. \(2023\)](#) cuando la base utilizada para las p -ary m -secuencias es 2.

Users	Alg.	Ciclos (%)									
		1	2	3	4	5	6	7	8	9	10
User 1	CCA	56,25	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	56,25	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 2	CCA	34,375	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	34,375	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 3	CCA	93,75	100,0	93,75	96,875	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	93,75	100,0	93,75	96,875	100,0	100,0	100,0	100,0	100,0	100,0
User 4	CCA	81,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	81,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 5	CCA	34,375	75,0	81,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	34,375	75,0	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 6	CCA	53,125	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	56,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 7	CCA	21,875	65,625	93,75	93,75	96,875	96,875	96,875	96,875	96,875	96,875
	TRCA	21,875	65,625	93,75	96,875	96,875	96,875	96,875	96,875	96,875	96,875
User 8	CCA	50,0	68,75	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	53,125	68,75	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 9	CCA	59,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	59,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 10	CCA	68,75	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	68,75	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 11	CCA	28,125	34,375	56,25	65,625	84,375	87,5	87,5	87,5	93,75	93,75
	TRCA	31,25	37,5	53,125	65,625	84,375	84,375	87,5	87,5	93,75	93,75
User 12	CCA	43,75	75,0	87,5	96,875	96,875	96,875	96,875	100,0	100,0	100,0
	TRCA	43,75	78,125	87,5	96,875	96,875	96,875	96,875	100,0	100,0	100,0
User 13	CCA	40,625	90,625	96,875	96,875	96,875	100,0	100,0	100,0	100,0	100,0
	TRCA	37,5	90,625	96,875	96,875	96,875	100,0	100,0	100,0	100,0	100,0
User 14	CCA	18,75	37,5	46,875	62,5	59,375	65,625	71,875	78,125	84,375	78,125
	TRCA	12,5	34,375	50,0	56,25	59,375	65,625	75,0	78,125	84,375	78,125
User 15	CCA	43,75	71,875	90,625	96,875	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	46,875	71,875	84,375	96,875	96,875	100,0	100,0	100,0	100,0	100,0
User 16	CCA	31,25	71,875	87,5	100,0	96,875	100,0	100,0	100,0	100,0	100,0
	TRCA	34,375	71,875	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0
Mean	CCA	47,46	79,49	88,48	94,34	95,70	96,68	97,07	97,66	98,44	98,05
	TRCA	47,85	79,69	88,48	94,14	95,70	96,48	97,27	97,66	98,44	98,05
Std	CCA	20,20	20,52	15,06	11,60	10,12	8,59	7,20	5,90	3,98	5,40
	TRCA	20,66	20,49	14,93	12,74	10,12	8,83	6,52	5,90	3,98	5,40
p-valor	-	0,527	0,564	1,000	0,655	1,000	0,317	0,317	1,000	1,000	1,000

Capítulo 9. Resultados del experimento

Tabla 9.2: Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de [Martínez-Cagigal et al. \(2023\)](#) cuando la base utilizada para las p -ary m -secuencias es 3.

Users	Alg.	Ciclos (%)										
		1	2	3	4	5	6	7	8	9	10	
User 1	CCA	68,75	93,75	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	62,5	90,625	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 2	CCA	50,0	87,5	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	50,0	87,5	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 3	CCA	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 4	CCA	87,5	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	87,5	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 5	CCA	50,0	62,5	84,375	90,625	100,0	96,875	100,0	100,0	100,0	100,0	100,0
	TRCA	50,0	62,5	84,375	93,75	100,0	96,875	100,0	100,0	100,0	100,0	100,0
User 6	CCA	43,75	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	46,875	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 7	CCA	46,875	84,375	90,625	93,75	96,875	96,875	96,875	100,0	100,0	100,0	100,0
	TRCA	40,625	81,25	90,625	93,75	93,75	96,875	96,875	100,0	100,0	100,0	100,0
User 8	CCA	75,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	71,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 9	CCA	46,875	90,625	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	46,875	87,5	84,375	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 10	CCA	65,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	65,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 11	CCA	34,375	56,25	53,125	65,625	71,875	78,125	81,25	87,5	90,625	90,625	90,625
	TRCA	34,375	59,375	56,25	68,75	71,875	71,875	81,25	84,375	87,5	90,625	90,625
User 12	CCA	21,875	43,75	59,375	68,75	81,25	87,5	90,625	96,875	96,875	100,0	100,0
	TRCA	21,875	43,75	50,0	65,625	81,25	84,375	90,625	93,75	96,875	100,0	100,0
User 13	CCA	56,25	87,5	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	56,25	87,5	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 14	CCA	12,5	18,75	43,75	53,125	50,0	56,25	71,875	75,0	84,375	81,25	81,25
	TRCA	9,375	21,875	37,5	53,125	50,0	53,125	68,75	75,0	84,375	81,25	81,25
User 15	CCA	37,5	68,75	87,5	93,75	93,75	93,75	100,0	100,0	100,0	100,0	100,0
	TRCA	34,375	68,75	84,375	93,75	93,75	93,75	100,0	100,0	100,0	100,0	100,0
User 16	CCA	34,375	56,25	62,5	78,125	87,5	93,75	96,875	100,0	100,0	100,0	100,0
	TRCA	34,375	53,125	65,625	78,125	87,5	93,75	100,0	100,0	100,0	100,0	100,0
Mean	CCA	50,98	77,34	84,96	90,04	92,58	93,95	96,09	97,46	98,24	98,24	98,24
	TRCA	50,00	76,95	83,98	90,04	92,38	93,16	96,09	97,07	98,05	98,24	98,24
Std	CCA	20,51	23,11	18,49	14,62	13,62	11,35	7,93	6,56	4,27	4,94	4,94
	TRCA	20,93	22,37	19,76	14,54	13,58	12,70	8,59	6,94	4,67	4,94	4,94
p-valor	-	0,160	0,414	0,236	1,000	0,317	0,102	1,000	0,157	0,317	1,000	1,000

Tabla 9.3: Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de [Martínez-Cagigal et al. \(2023\)](#) cuando la base utilizada para las p -ary m -secuencias es 5.

Users	Alg.	Ciclos (%)									
		1	2	3	4	5	6	7	8	9	10
User 1	CCA	78,125	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	81,25	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 2	CCA	78,125	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	75,0	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 3	CCA	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 4	CCA	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 5	CCA	56,25	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	50,0	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 6	CCA	68,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	68,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 7	CCA	56,25	84,375	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	59,375	84,375	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 8	CCA	71,875	90,625	90,625	96,875	96,875	100,0	100,0	100,0	100,0	100,0
	TRCA	75,0	93,75	93,75	96,875	100,0	100,0	100,0	100,0	100,0	100,0
User 9	CCA	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 10	CCA	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 11	CCA	25,0	43,75	59,375	68,75	81,25	90,625	90,625	90,625	96,875	96,875
	TRCA	25,0	43,75	59,375	68,75	81,25	87,5	87,5	90,625	93,75	96,875
User 12	CCA	46,875	71,875	90,625	93,75	93,75	96,875	100,0	100,0	100,0	100,0
	TRCA	43,75	71,875	90,625	93,75	96,875	96,875	100,0	100,0	100,0	100,0
User 13	CCA	87,5	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	87,5	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 14	CCA	37,5	68,75	71,875	87,5	93,75	93,75	90,625	96,875	100,0	100,0
	TRCA	43,75	65,625	68,75	87,5	87,5	93,75	90,625	96,875	100,0	100,0
User 15	CCA	56,25	84,375	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	56,25	84,375	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 16	CCA	75,0	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	68,75	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
Mean	CCA	69,73	88,28	94,14	96,68	97,85	98,83	98,83	99,22	99,80	99,80
	TRCA	69,53	88,09	94,14	96,68	97,85	98,63	98,63	99,22	99,61	99,80
Std	CCA	21,04	14,80	11,42	7,93	4,78	2,68	3,10	2,34	0,76	0,76
	TRCA	20,95	15,12	11,79	7,93	5,27	3,31	3,66	2,34	1,51	0,76
p-valor	-	0,773	0,564	1,000	1,000	1,000	0,317	0,317	1,000	0,317	1,000

Tabla 9.4: Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de [Martínez-Cagigal et al. \(2023\)](#) cuando la base utilizada para las p -ary m -secuencias es 7.

Users	Alg.	Ciclos (%)									
		1	2	3	4	5	6	7	8	9	10
User 1	CCA	31,25	65,625	84,375	90,625	96,875	100,0	100,0	100,0	100,0	100,0
	TRCA	28,125	68,75	81,25	87,5	96,875	100,0	100,0	100,0	100,0	100,0
User 2	CCA	21,875	56,25	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	21,875	56,25	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 3	CCA	53,125	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	53,125	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 4	CCA	43,75	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	43,75	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 5	CCA	21,875	53,125	78,125	87,5	93,75	93,75	93,75	96,875	93,75	93,75
	TRCA	21,875	53,125	78,125	87,5	93,75	93,75	93,75	96,875	93,75	96,875
User 6	CCA	25,0	78,125	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	25,0	78,125	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 7	CCA	21,875	62,5	87,5	96,875	96,875	96,875	96,875	100,0	100,0	100,0
	TRCA	18,75	62,5	87,5	93,75	96,875	96,875	96,875	100,0	100,0	100,0
User 8	CCA	43,75	84,375	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	43,75	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 9	CCA	46,875	84,375	100,0	100,0	96,875	100,0	100,0	100,0	100,0	100,0
	TRCA	46,875	81,25	100,0	100,0	96,875	100,0	100,0	100,0	100,0	100,0
User 10	CCA	31,25	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	31,25	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 11	CCA	12,5	37,5	37,5	56,25	62,5	59,375	65,625	78,125	75,0	81,25
	TRCA	9,375	37,5	40,625	62,5	62,5	62,5	62,5	71,875	75,0	81,25
User 12	CCA	25,0	43,75	65,625	81,25	90,625	93,75	96,875	96,875	100,0	100,0
	TRCA	21,875	46,875	62,5	78,125	84,375	96,875	96,875	96,875	100,0	100,0
User 13	CCA	34,375	75,0	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	34,375	75,0	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 14	CCA	18,75	43,75	53,125	68,75	90,625	87,5	93,75	93,75	93,75	93,75
	TRCA	18,75	43,75	53,125	71,875	87,5	87,5	93,75	90,625	87,5	90,625
User 15	CCA	21,875	75,0	93,75	100,0	96,875	100,0	96,875	100,0	100,0	100,0
	TRCA	21,875	75,0	93,75	100,0	96,875	100,0	96,875	100,0	100,0	100,0
User 16	CCA	21,875	81,25	90,625	96,875	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	34,375	81,25	93,75	96,875	100,0	100,0	100,0	100,0	100,0	100,0
Mean	CCA	29,69	69,34	85,55	92,38	95,31	95,70	96,48	97,85	97,66	98,05
	TRCA	29,69	69,53	85,55	92,38	94,73	96,09	96,29	97,27	97,27	98,05
Std	CCA	11,27	17,58	17,91	12,69	9,04	10,00	8,26	5,38	6,20	4,80
	TRCA	11,74	17,13	17,88	11,48	9,49	9,28	8,99	6,98	6,62	4,93
p -valor	-	0,480	0,564	1,000	1,000	0,180	0,157	0,317	0,180	0,317	1,000

Tabla 9.5: Tabla comparativa de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA por usuarios y por ciclos de la base de datos de [Martínez-Cagigal et al. \(2023\)](#) cuando la base utilizada para las p -ary m -secuencias es 11.

Users	Alg.	Ciclos (%)									
		1	2	3	4	5	6	7	8	9	10
User 1	CCA	81,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	78,125	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 2	CCA	81,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 3	CCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 4	CCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 5	CCA	75,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	75,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 6	CCA	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 7	CCA	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	87,5	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 8	CCA	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	84,375	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 9	CCA	56,25	65,625	75,0	87,5	84,375	87,5	87,5	87,5	90,625	90,625
	TRCA	56,25	65,625	75,0	87,5	84,375	87,5	87,5	87,5	87,5	90,625
User 10	CCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 11	CCA	59,375	84,375	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	56,25	81,25	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 12	CCA	53,125	84,375	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	56,25	87,5	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 13	CCA	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 14	CCA	15,625	37,5	46,875	68,75	68,75	84,375	84,375	84,375	87,5	87,5
	TRCA	18,75	31,25	46,875	68,75	75,0	90,625	87,5	87,5	93,75	87,5
User 15	CCA	81,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	81,25	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 16	CCA	43,75	75,0	96,875	100,0	96,875	96,875	100,0	100,0	100,0	100,0
	TRCA	46,875	71,875	93,75	100,0	96,875	96,875	100,0	100,0	100,0	100,0
Mean	CCA	74,80	90,43	93,95	97,27	96,88	98,05	98,24	98,24	98,63	98,63
	TRCA	75,20	89,84	94,14	97,27	97,27	98,44	98,44	98,44	98,83	98,63
Std	CCA	22,71	17,20	13,69	7,96	8,19	4,67	4,68	4,68	3,66	3,66
	TRCA	21,94	18,64	13,66	7,96	6,89	3,66	4,13	4,13	3,29	3,66
p-valor	-	0,414	0,257	0,655	1,000	0,317	0,317	0,317	0,317	0,655	1,000

Tabla 9.6: Tabla de medias por usuarios de las precisiones obtenidas entre los métodos de extracción de filtros espaciales CCA y TRCA para cada ciclo de la base de datos de Martínez-Cagigal et al. (2023).

Base	Alg.	Ciclos (%)									
		1	2	3	4	5	6	7	8	9	10
2	CCA	47,46	79,49	88,48	94,34	95,70	96,68	97,07	97,66	98,44	98,05
	TRCA	47,85	79,69	88,48	94,14	95,70	96,48	97,27	97,66	98,44	98,05
	<i>p</i> -valor	0,527	0,564	1,000	0,655	1,000	0,317	0,317	1,000	1,000	1,000
3	CCA	50,98	77,34	84,96	90,04	92,58	93,95	96,09	97,46	98,24	98,24
	TRCA	50,00	76,95	83,98	90,04	92,38	93,16	96,09	97,07	98,05	98,24
	<i>p</i> -valor	0,160	0,414	0,236	1,000	0,317	0,102	1,000	0,157	0,317	1,000
5	CCA	69,73	88,28	94,14	96,68	97,85	98,83	98,83	99,22	99,80	99,80
	TRCA	69,53	88,09	94,14	96,68	97,85	98,63	98,63	99,22	99,61	99,80
	<i>p</i> -valor	0,773	0,564	1,000	1,000	1,000	0,317	0,317	1,000	0,317	1,000
7	CCA	29,69	69,34	85,55	92,38	95,31	95,70	96,48	97,85	97,66	98,05
	TRCA	29,69	69,53	85,55	92,38	94,73	96,09	96,29	97,27	97,27	98,05
	<i>p</i> -valor	0,480	0,564	1,000	1,000	0,180	0,157	0,317	0,180	0,317	1,000
11	CCA	74,80	90,43	93,95	97,27	96,88	98,05	98,24	98,24	98,63	98,63
	TRCA	75,20	89,84	94,14	97,27	97,27	98,44	98,44	98,44	98,83	98,63
	<i>p</i> -valor	0,414	0,257	0,655	1,000	0,317	0,317	0,317	0,317	0,655	1,000

9.1.2. Base de datos de Thielen et al.

Finalmente, la última parte del análisis offline consiste en realizar una comparación entre CCA y TRCA cuando se utilizan Gold Codes. Para ello, se ha hecho uso de la base de datos de Thielen et al. (2021) a través de Donders Data Repository (<https://doi.org/10.34973/9txv-z787>). Este artículo codificó un total de 20 comandos a través del uso exclusivo de Gold Codes. Para ello, el autor escogió el par preferido de polinomios dado por $X^6 + X + 1$ y $X^6 + X^5 + X^2 + X + 1$ para generar un total de $2^6 - 1 = 63$ Gold Codes. También decidió añadir a este conjunto de secuencias las dos originales derivadas de aplicar LFSR a los polinomios antes mencionados, logrando así un total de 65 códigos de los cuales se escogieron 20 para codificar la matriz de comandos mostrada en la figura 9.2. Los 30 participantes completaron cinco bloques idénticos. Cada bloque consistía en 20 trials, una para cada una de las 20 celdas presentadas en orden aleatorio. Al inicio del trial, la celda objetivo se resaltaba en verde durante 1 s. Después de esta señal, todas las celdas comenzaban a parpadear durante 31,5 s. Mientras, el participante mantenía la fijación en la celda objetivo. Una vez finalizado el trial, el siguiente trial se iniciaba directamente, sin dar retroalimentación. Debido a la naturaleza del artículo, no existe una fase de prueba utilizable en este caso. Por lo tanto, se ha empleado la técnica de *k-fold cross-validation*, en este caso con $k = 5$. Esta técnica consiste en utilizar uno de los cinco bloques de entrenamiento como fase de prueba, rotando secuencialmente cada bloque para este propósito. Primero, se usa el primer bloque como fase de test y el resto como entrenamiento del modelo, luego se repite el proceso utilizando el segundo bloque como test, y así sucesivamente hasta completar los cinco bloques. Una vez realizado el análisis variando el

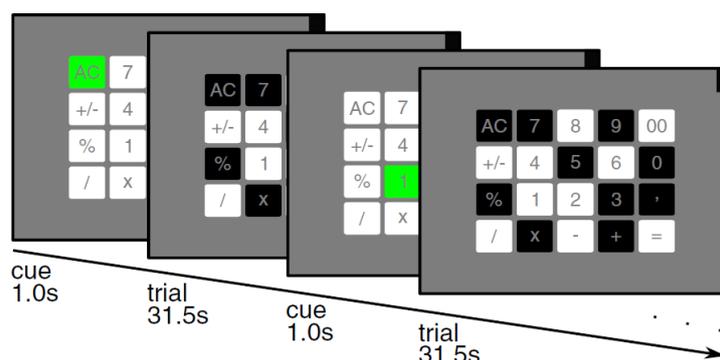


Figura 9.2: Estímulos y protocolo experimental de la fase offline del artículo de [Thielen et al. \(2021\)](#). La interfaz gráfica muestra una calculadora con 20 comandos seleccionables. Los objetivos se indicaron con un color verde. Se dejó un segundo de descanso entre trials. La duración de cada trials fue de 31,5 s.

Tabla 9.7: Precisiones medias obtenidas por usuarios y ciclos tras aplicar CCA y TRCA a la base de datos de [Thielen et al. \(2021\)](#), utilizando *5-fold cross-validation* sobre el bloque utilizado para la fase de test.

	Alg.	Ciclos (%)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean	CCA	81,53	90,33	92,97	93,93	94,13	94,37	94,57	94,53	94,53	94,53	94,53	94,43	94,50	94,37	94,37
	TRCA	83,60	92,90	95,40	96,33	96,50	96,73	96,93	96,93	96,87	96,87	96,87	96,80	96,83	96,70	96,70
p-valor	-	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05

bloque de entrenamiento, se calcula la media de los resultados obtenidos en cada iteración.

La tabla 9.7 resume la tabla 9.8, la cual muestra el resultado final del *cross-validation*. Esta última muestra las precisiones por usuarios y por ciclos usando tanto TRCA como CCA, así como la media sobre los usuarios, para cada uno de los ciclos. Se observa que todas las precisiones en el último ciclo superan el 94 % exceptuando en los usuarios 8 y 12. Sin embargo, en estos dos usuarios tanto CCA como TRCA obtienen precisiones bajas. En líneas generales, observando casi cualquier ciclo y usuario, la precisión de TRCA supera a la de CCA. Este mismo hecho se observa en las medias obtenidas para cada uno de los algoritmos. Finalmente, se aprecia que los *p*-valores obtenidos son, para cada uno de los ciclos, estrictamente menores que 0,05.

Capítulo 9. Resultados del experimento

Tabla 9.8: Precisiones obtenidas por usuarios y ciclos tras aplicar CCA y TRCA a la base de datos de Thielen et al. (2021), utilizando 5-fold cross-validation sobre el bloque utilizado para la fase de test.

Users	Alg.	Ciclos (%)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
User 1	CCA	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 2	CCA	82,0	93,0	94,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0
	TRCA	86,0	98,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 3	CCA	96,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0
	TRCA	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 4	CCA	88,0	95,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0
	TRCA	91,0	98,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0
User 5	CCA	86,0	95,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0
	TRCA	87,0	96,0	97,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0
User 6	CCA	97,0	97,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0
	TRCA	97,0	97,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0
User 7	CCA	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	98,0	98,0	98,0	98,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	99,0	99,0	99,0	99,0
User 8	CCA	55,0	59,0	59,0	60,0	59,0	60,0	61,0	60,0	61,0	61,0	62,0	61,0	62,0	61,0	61,0
	TRCA	57,0	61,0	61,0	61,0	60,0	61,0	62,0	62,0	62,0	62,0	63,0	62,0	63,0	62,0	62,0
User 9	CCA	96,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0
	TRCA	97,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 10	CCA	33,0	56,0	81,0	88,0	88,0	94,0	95,0	96,0	95,0	96,0	96,0	95,0	96,0	96,0	97,0
	TRCA	34,0	59,0	83,0	90,0	90,0	96,0	96,0	98,0	97,0	98,0	98,0	97,0	98,0	98,0	99,0
User 11	CCA	74,0	91,0	97,0	98,0	98,0	98,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0
	TRCA	75,0	92,0	98,0	99,0	99,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 12	CCA	47,0	54,0	56,0	60,0	59,0	58,0	58,0	57,0	55,0	55,0	54,0	53,0	53,0	50,0	49,0
	TRCA	53,0	59,0	58,0	63,0	61,0	60,0	61,0	59,0	56,0	56,0	55,0	55,0	54,0	51,0	50,0
User 13	CCA	73,0	84,0	94,0	96,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0
	TRCA	75,0	87,0	97,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 14	CCA	87,0	94,0	94,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0
	TRCA	91,0	99,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 15	CCA	87,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0
	TRCA	89,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 16	CCA	93,0	96,0	96,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0
	TRCA	95,0	98,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 17	CCA	91,0	93,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0
	TRCA	96,0	98,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 18	CCA	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 19	CCA	81,0	89,0	92,0	94,0	94,0	94,0	94,0	94,0	94,0	94,0	94,0	94,0	94,0	94,0	94,0
	TRCA	85,0	95,0	97,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0
User 20	CCA	62,0	92,0	97,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0	98,0
	TRCA	61,0	94,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 21	CCA	92,0	95,0	96,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0
	TRCA	95,0	98,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 22	CCA	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 23	CCA	69,0	92,0	96,0	95,0	96,0	96,0	96,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0	97,0
	TRCA	70,0	95,0	99,0	98,0	99,0	99,0	99,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 24	CCA	84,0	95,0	96,0	98,0	98,0	98,0	98,0	97,0	98,0	98,0	98,0	97,0	97,0	97,0	97,0
	TRCA	85,0	97,0	98,0	100,0	100,0	100,0	100,0	99,0	100,0	100,0	100,0	99,0	99,0	99,0	99,0
User 25	CCA	89,0	93,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0	95,0
	TRCA	93,0	98,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 26	CCA	43,0	75,0	84,0	90,0	93,0	95,0	96,0	96,0	97,0	97,0	98,0	98,0	98,0	98,0	98,0
	TRCA	44,0	77,0	87,0	92,0	95,0	97,0	98,0	98,0	99,0	99,0	99,0	100,0	100,0	100,0	100,0
User 27	CCA	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0	96,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 28	CCA	92,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
	TRCA	92,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 29	CCA	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0	99,0
	TRCA	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 30	CCA	61,0	90,0	94,0	94,0	97,0	96,0	98,0	98,0	98,0	97,0	97,0	98,0	98,0	98,0	98,0
	TRCA	62,0	91,0	95,0	95,0	98,0	97,0	99,0	99,0	99,0	98,0	98,0	99,0	99,0	99,0	99,0
Mean	CCA	81,53	90,33	92,97	93,93	94,13	94,37	94,57	94,53	94,53	94,53	94,53	94,43	94,50	94,37	94,37
	TRCA	83,60	92,90	95,40	96,33	96,50	96,73	96,93	96,93	96,87	96,87	96,87	96,80	96,83	96,70	96,70
Std	CCA	18,03	12,39	10,28	9,43	9,67	9,60	9,51	9,76	9,92	9,91	9,94	10,17	10,07	10,59	10,74
	TRCA	18,16	12,05	10,29	9,48	9,83	9,75	9,52	9,77	10,18	10,18	10,20	10,31	10,33	10,86	11,01
p-valor		<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05	<0,05

9.2. Análisis online

El procedimiento para realizar el análisis online fue explicado en el capítulo anterior. Así, los resultados del experimento *Circular Shifting* se encuentran en la tabla 9.9. Es importante destacar que la extracción del filtro espacial y la creación del modelo para la decodificación de los comandos en este experimento fue con CCA. Sin embargo, los experimentos *Gold-Circular* fueron realizados con la aplicación que se ha implementado en este Trabajo de Fin de Grado que, como se ha explicado anteriormente, combina el paradigma *Circular Shifting* junto con los Gold Codes. Esta aplicación se ha implementado para que el método de extracción de filtros espaciales sea TRCA. Los resultados obtenidos de los 10 sujetos se encuentran en las tablas 9.10 y 9.11, respectivamente para 31 y 63 bits.

Se puede observar que la precisión media para el experimento *Circular Shifting* con CCA es de 99,38 %, superior a la obtenida en el experimento *Gold-Circular* con TRCA, siendo 95,31 % y 96,25 % para 31 y 63 bits, respectivamente. Se observa que estas dos últimas precisiones utilizan el mismo paradigma y método de extracción de filtros espaciales, siendo la única diferencia la longitud de la secuencia. En la tabla 9.12 se incluye un resumen de las medias de las precisiones obtenidas y el p -valor obtenido al aplicar el test de Wilcoxon por ciclos a las precisiones obtenidas en los experimentos. Se observa que los únicos p -valores estrictamente menores que 0,05 son los de los ciclos 1 y 2, cuando se compara *Gold-Circular* de 31 y 63 bits. Sin embargo, cuando se compara *Circular Shifting* con *Gold-Circular*, únicamente se obtienen p -valores estrictamente menores que 0,05 en los tres primeros ciclos al comparar con el experimento de 31 bits.

Finalmente, las figuras 9.3, 9.6 y 9.9, muestran una gráfica sobre la topografía en un modelo de cabeza 2D, ponderando la contribución de cada uno de los electrodos en el filtro espacial, para cada uno de los usuarios y de los experimentos realizados. Las figuras 9.4, 9.7 y 9.10 muestran, respectivamente, para cada experimento, la plantilla principal obtenida para cada usuario, tomada como la promediación de la respuesta del EEG en el canal Oz, sobre 200 ciclos de calibración.

Tabla 9.9: Precisiones obtenidas del análisis online mostrada por ciclos y por usuarios del experimento *Circular Shifting* con 63 bits y CCA, así como la media y desviación típica de cada ciclo por usuarios.

Users	Ciclos (%)									
	1	2	3	4	5	6	7	8	9	10
User 1	37,5	59,375	68,75	75,0	90,625	90,625	93,75	90,625	93,75	93,75
User 2	43,75	93,75	96,875	96,875	100,0	100,0	100,0	100,0	100,0	100,0
User 3	28,125	81,25	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 4	53,125	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 5	68,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 6	37,5	87,5	96,875	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 7	12,5	40,625	65,625	75,0	90,625	93,75	100,0	100,0	100,0	100,0
User 8	12,5	53,125	65,625	84,375	93,75	96,875	100,0	96,875	96,875	100,0
User 9	53,125	93,75	93,75	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 10	50,0	78,125	81,25	90,625	93,75	96,875	96,875	96,875	100,0	100,0
Mean	39,69	77,81	86,25	92,19	96,88	97,81	99,06	98,44	99,06	99,38
Std	17,18	18,98	13,78	9,91	3,95	3,14	2,00	2,88	2,00	1,88

Tabla 9.10: Precisiones obtenidas del análisis online mostrada por ciclos y por usuarios del experimento *Gold-Circular* con 31 bits y TRCA, así como la media y desviación típica de cada ciclo por usuarios.

Users	Ciclos (%)									
	1	2	3	4	5	6	7	8	9	10
User 1	9,375	46,875	56,25	71,875	93,75	93,75	96,875	96,875	96,875	96,875
User 2	12,5	50,0	68,75	84,375	96,875	100,0	100,0	100,0	100,0	100,0
User 3	9,375	18,75	25,0	56,25	71,875	71,875	68,75	78,125	81,25	87,5
User 4	21,875	75,0	90,625	100,0	93,75	100,0	100,0	100,0	100,0	100,0
User 5	21,875	71,875	93,75	96,875	100,0	100,0	100,0	100,0	100,0	100,0
User 6	15,625	68,75	84,375	90,625	90,625	90,625	93,75	96,875	100,0	100,0
User 7	28,125	65,625	93,75	93,75	100,0	100,0	100,0	100,0	100,0	100,0
User 8	15,625	40,625	56,25	68,75	68,75	68,75	78,125	75,0	78,125	81,25
User 9	6,25	40,625	53,125	75,0	84,375	78,125	90,625	90,625	90,625	93,75
User 10	9,375	43,75	59,375	62,5	81,25	90,625	87,5	87,5	93,75	93,75
Mean	15,00	52,19	68,12	80,00	88,12	89,38	91,56	92,50	94,06	95,31
Std	6,67	16,89	21,28	14,47	10,62	11,54	10,18	8,97	7,84	6,13

Tabla 9.11: Precisiones obtenidas del análisis online mostrada por ciclos y por usuarios del experimento *Gold-Circular* con 63 bits y TRCA, así como la media y desviación típica de cada ciclo por usuarios.

Users	Ciclos (%)									
	1	2	3	4	5	6	7	8	9	10
User 1	40,625	71,875	87,5	90,625	93,75	96,875	96,875	100,0	100,0	100,0
User 2	40,625	90,625	93,75	96,875	96,875	96,875	96,875	96,875	96,875	100,0
User 3	34,375	59,375	81,25	96,875	100,0	100,0	100,0	100,0	100,0	100,0
User 4	53,125	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 5	40,625	96,875	100,0	100,0	96,875	96,875	100,0	100,0	100,0	100,0
User 6	53,125	90,625	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0
User 7	34,375	56,25	68,75	84,375	84,375	84,375	87,5	93,75	96,875	96,875
User 8	12,5	25,0	43,75	56,25	53,125	59,375	65,625	75,0	78,125	78,125
User 9	21,875	59,375	75,0	78,125	81,25	84,375	87,5	84,375	84,375	87,5
User 10	37,5	71,875	87,5	90,625	93,75	93,75	93,75	96,875	96,875	100,0
Mean	36,88	72,19	83,75	89,38	90,00	91,25	92,81	94,69	95,31	96,25
Std	11,84	22,07	16,82	13,05	13,75	12,01	10,18	8,03	7,30	7,10

Tabla 9.12: Tabla de medias por usuarios de las precisiones obtenidas los experimentos online para cada ciclo. También se ha incluido el p -valor comparando todos los experimentos, donde GC hace referencia a *Gold-Circular*.

Experimento	Alg.	Bits	Ciclos (%)									
			1	2	3	4	5	6	7	8	9	10
<i>Circular Shifting</i>	CCA	63	39,69	77,81	86,25	92,19	96,88	97,81	99,06	98,44	99,06	99,38
	p -valor (GC 31)		0,010	0,020	0,037	0,050	0,050	0,075	0,046	0,073	0,078	0,078
	p -valor (GC 63)		0,726	0,492	0,622	0,500	0,072	0,089	0,071	0,225	0,168	0,273
<i>Gold-Circular</i>	TRCA	31	15,00	52,19	68,12	80,00	88,12	89,38	91,56	92,50	94,06	95,31
	TRCA	63	36,88	72,19	83,75	89,38	90,00	91,25	92,81	94,69	95,31	96,25
	p -valor		0,004	0,010	0,064	0,109	0,779	0,720	1,000	0,609	0,914	0,671

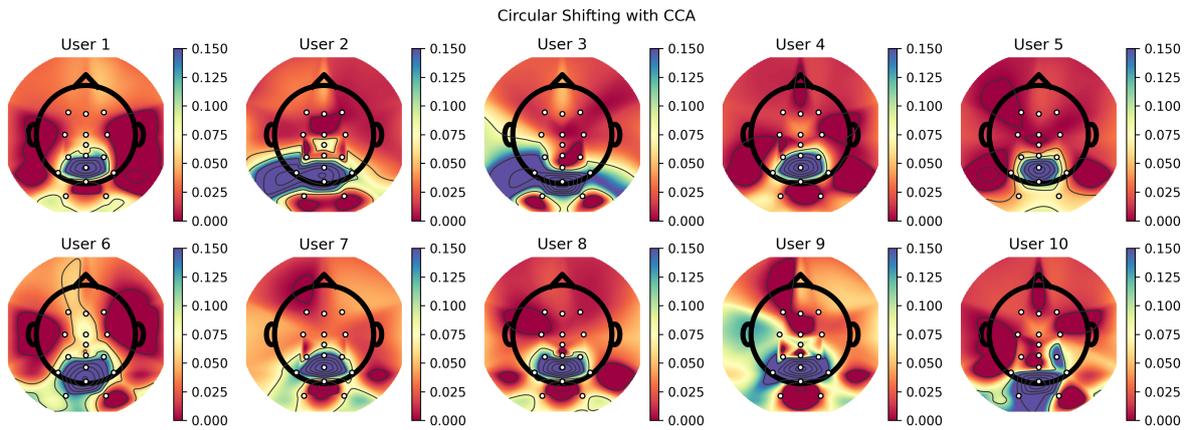


Figura 9.3: Gráfica de la topografía en un modelo de cabeza 2D, donde se representa el valor del filtro espacial obtenido con CCA en cada uno de los canales utilizados para el experimento *Circular Shifting*.

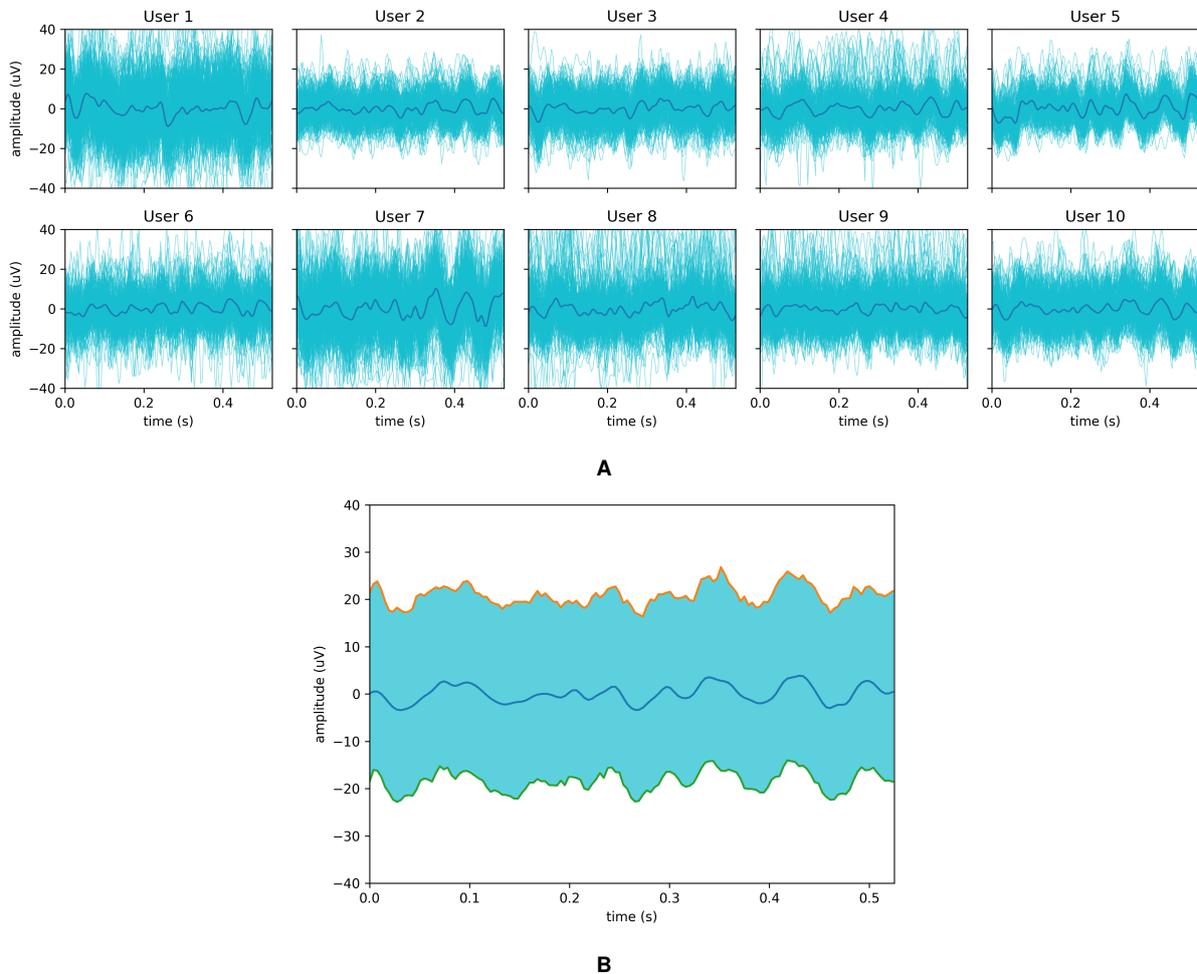
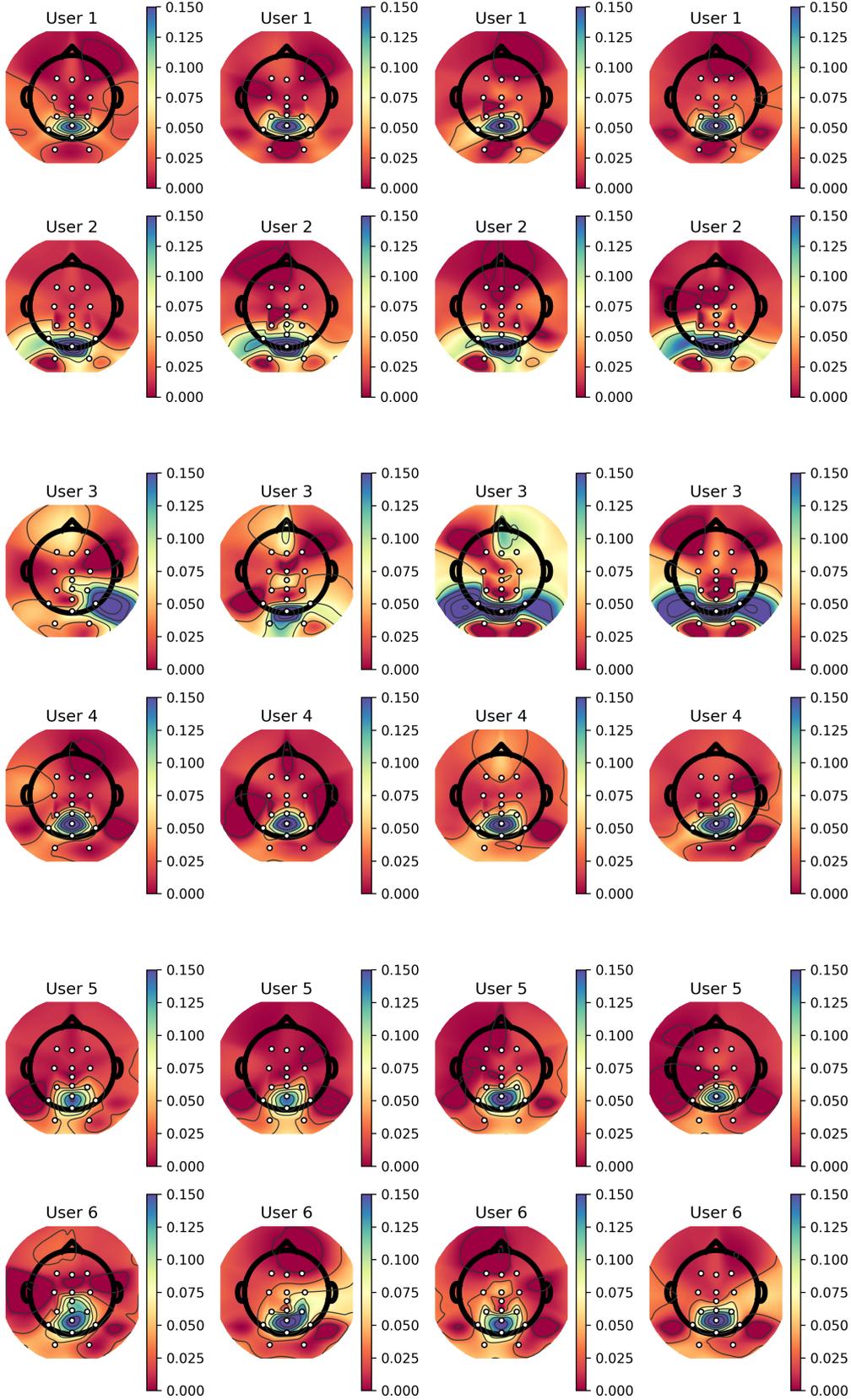


Figura 9.4: Respuesta cerebral de los usuarios en el experimento *Circular Shifting*. **(A)** Plantilla principal para cada uno de los usuarios tomada como promedio de la respuesta del EEG en el canal Oz sobre 200 ciclos de calibración. **(B)** Intervalo de confianza al 95 % para los ciclos de calibración de todos los usuarios, así como la media obtenida.

Gold Codes & Circular Shifting with TRCA and 31 bits



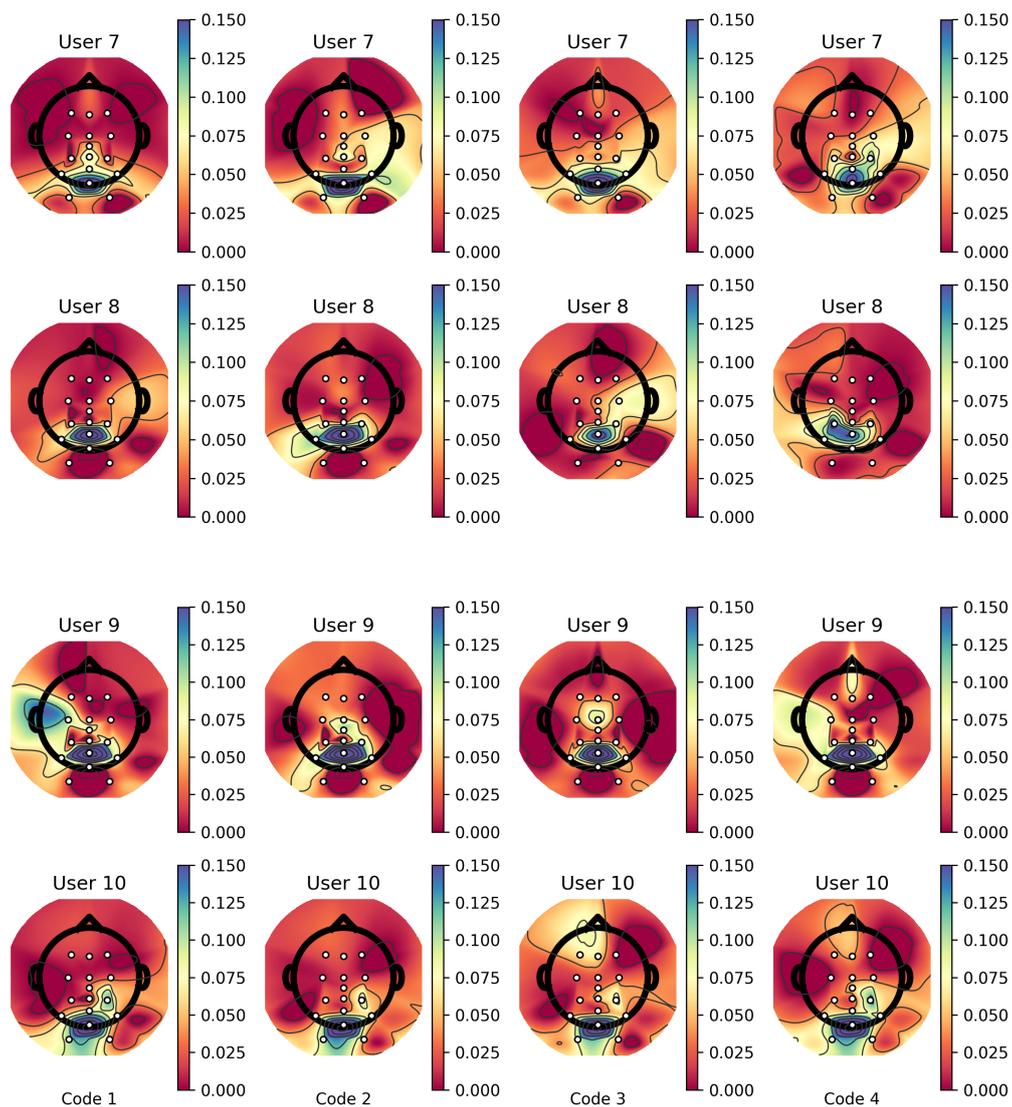
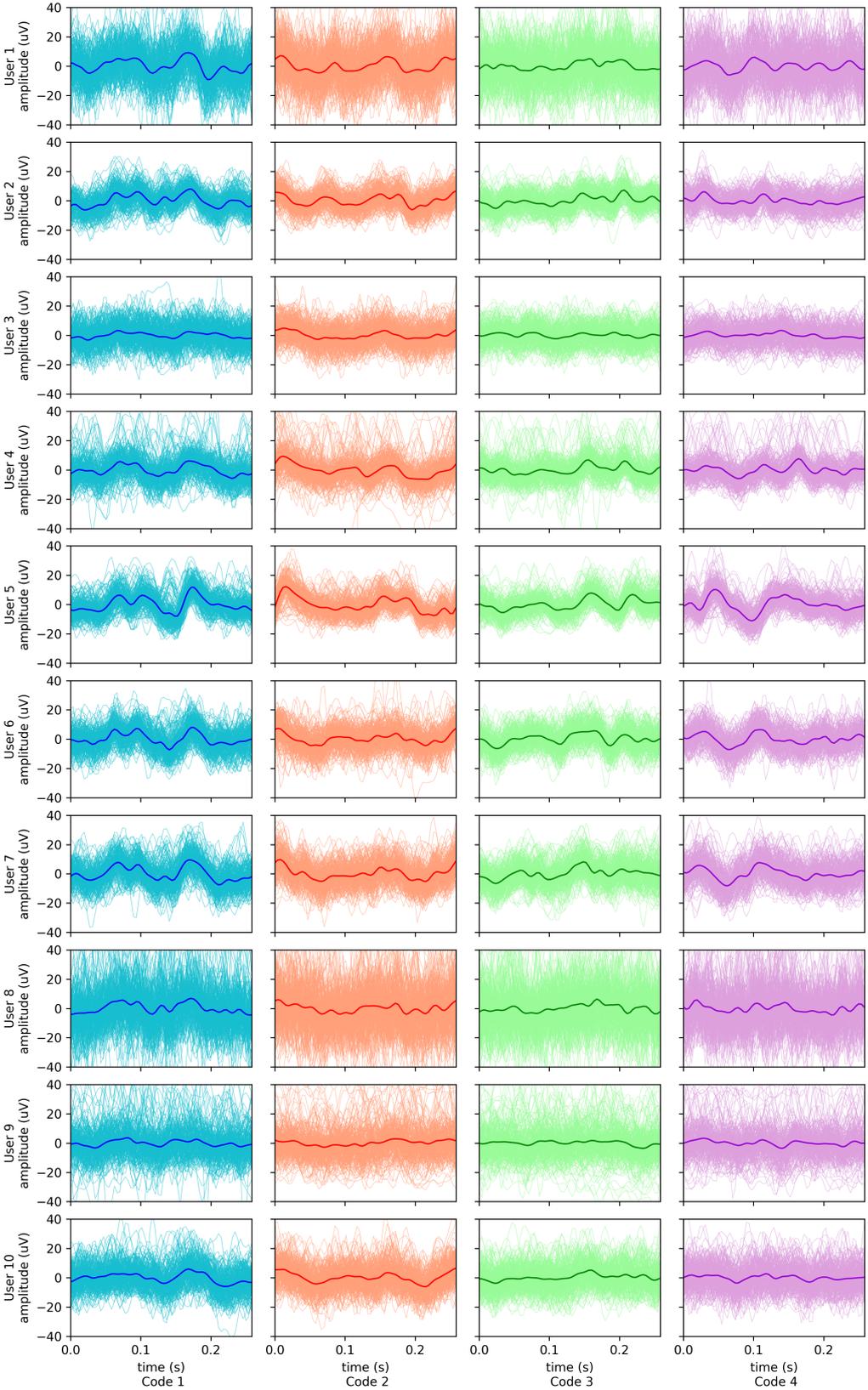


Figura 9.6: Gráfica de la topografía en un modelo de cabeza 2D, donde se representa el valor del filtro espacial obtenido con TRCA en cada uno de los canales utilizados para el experimento *Gold-Circular* con 31 bits. La figura muestra la topografía de los usuarios por filas y de los Gold Codes por columnas, donde el número de Gold Code se corresponde con los de la tabla 8.2.



A

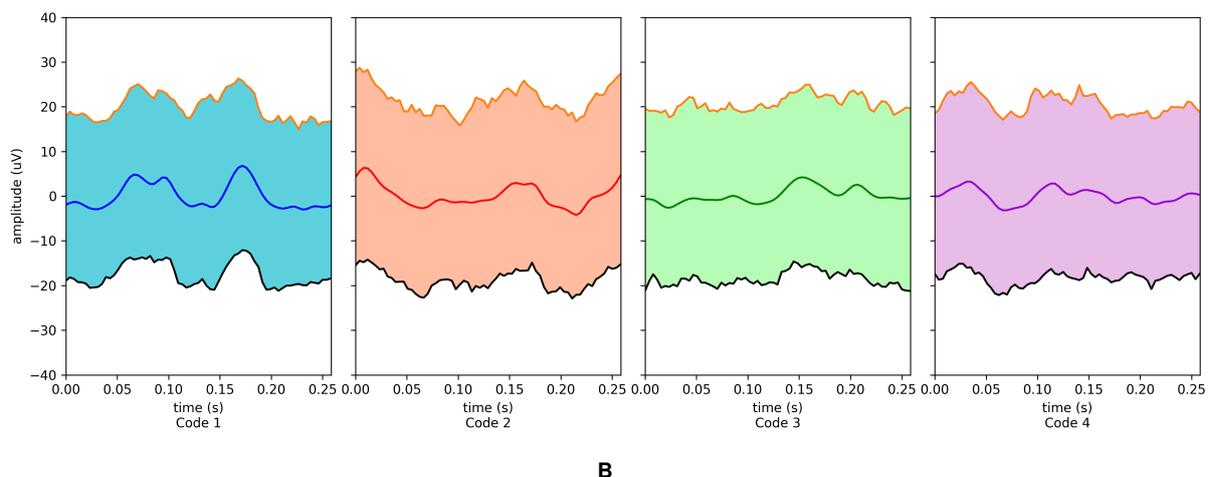
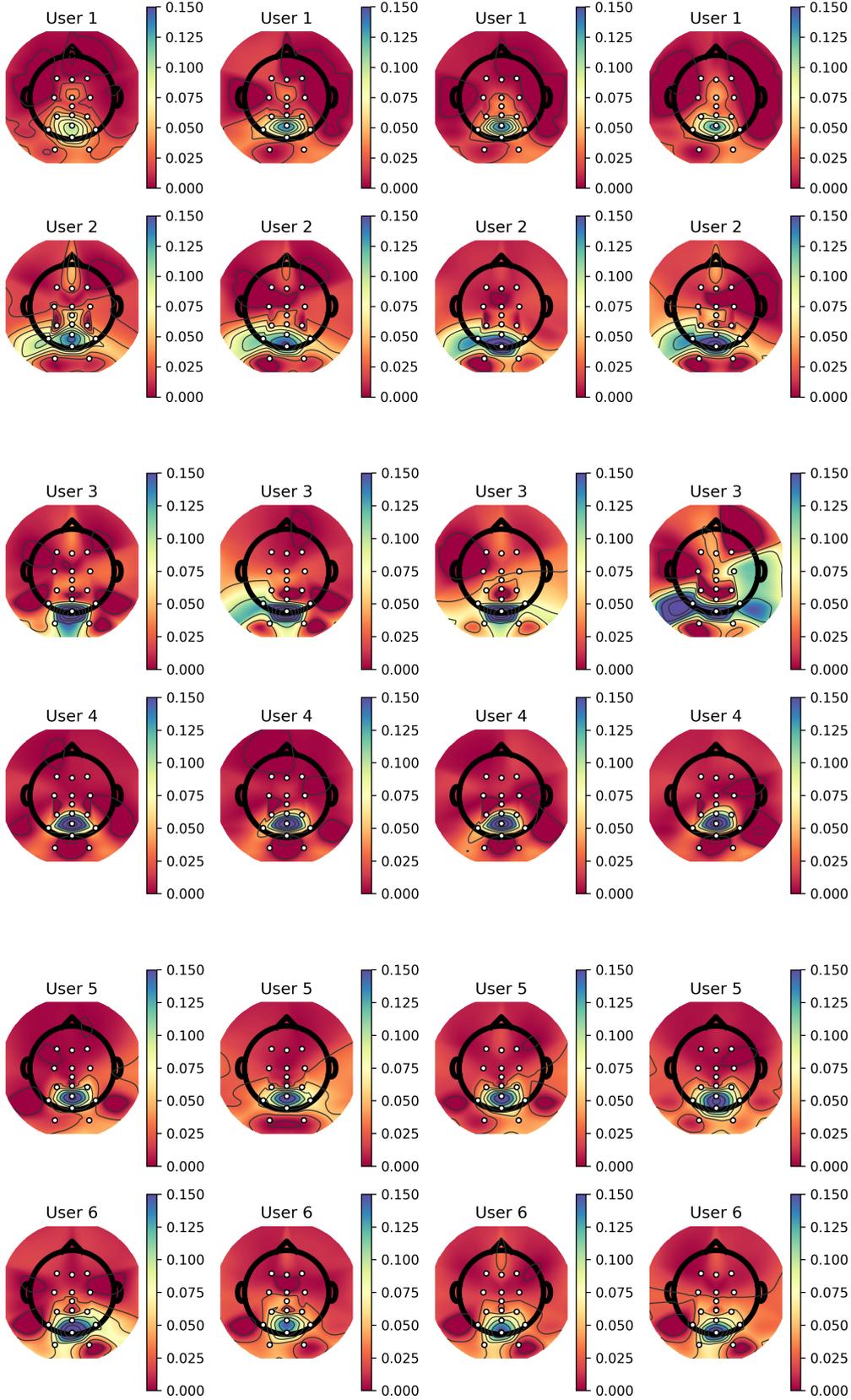


Figura 9.7: Respuesta cerebral de los usuarios en el experimento *Gold-Circular* con 31 bits. **(A)** Plantilla principal para cada uno de los usuarios tomada como promedio de la respuesta del EEG en el canal Oz sobre 200 ciclos de calibración. En azul se muestra la plantilla principal para el primer Gold Code, en naranja para el segundo, en verde para el tercero y en morado para el cuarto. **(B)** Intervalo de confianza al 95 % para los ciclos de calibración de todos los usuarios, así como la media obtenida, para cada uno de los Gold Codes. El número de Gold Code se corresponde con los de la tabla 8.2.

Gold Codes & Circular Shifting with TRCA and 63 bits



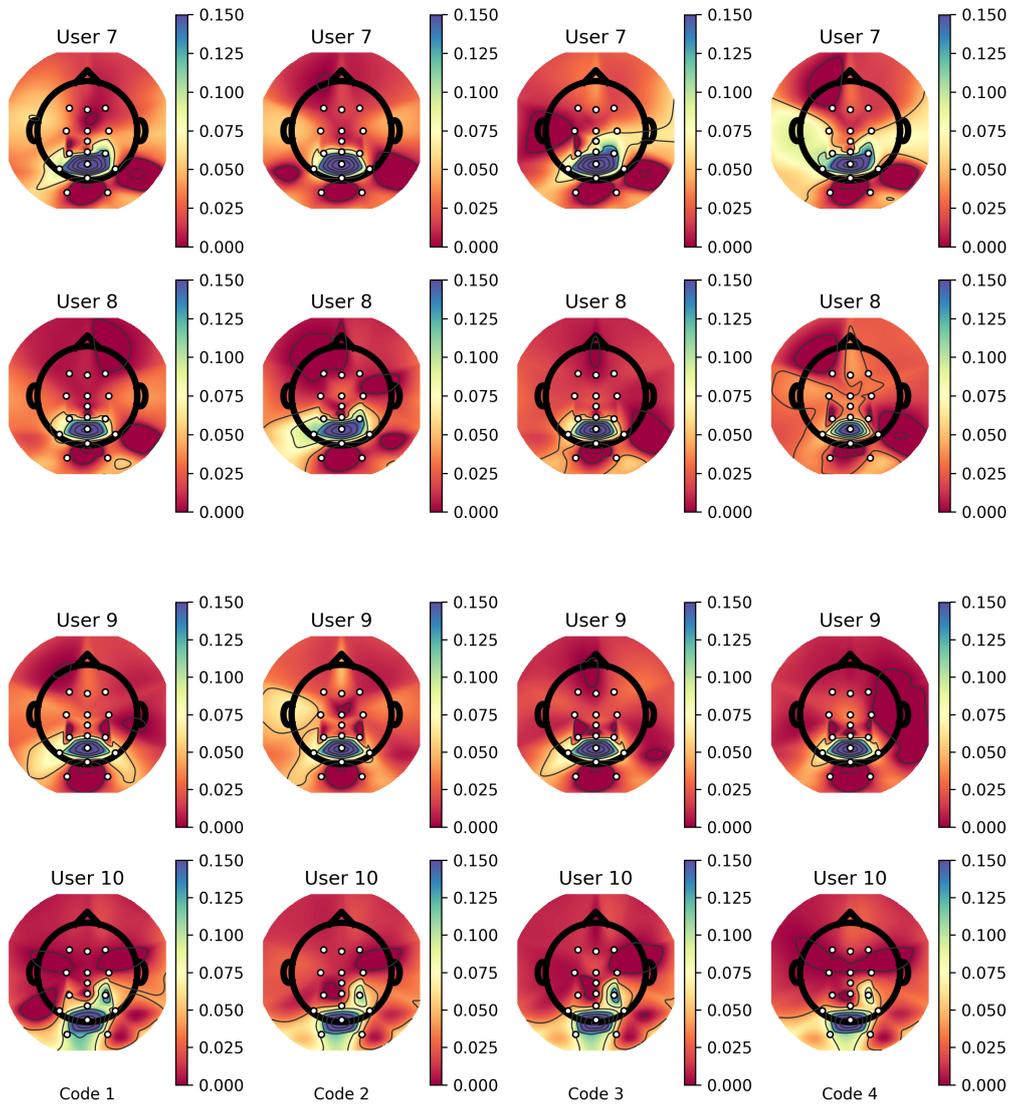
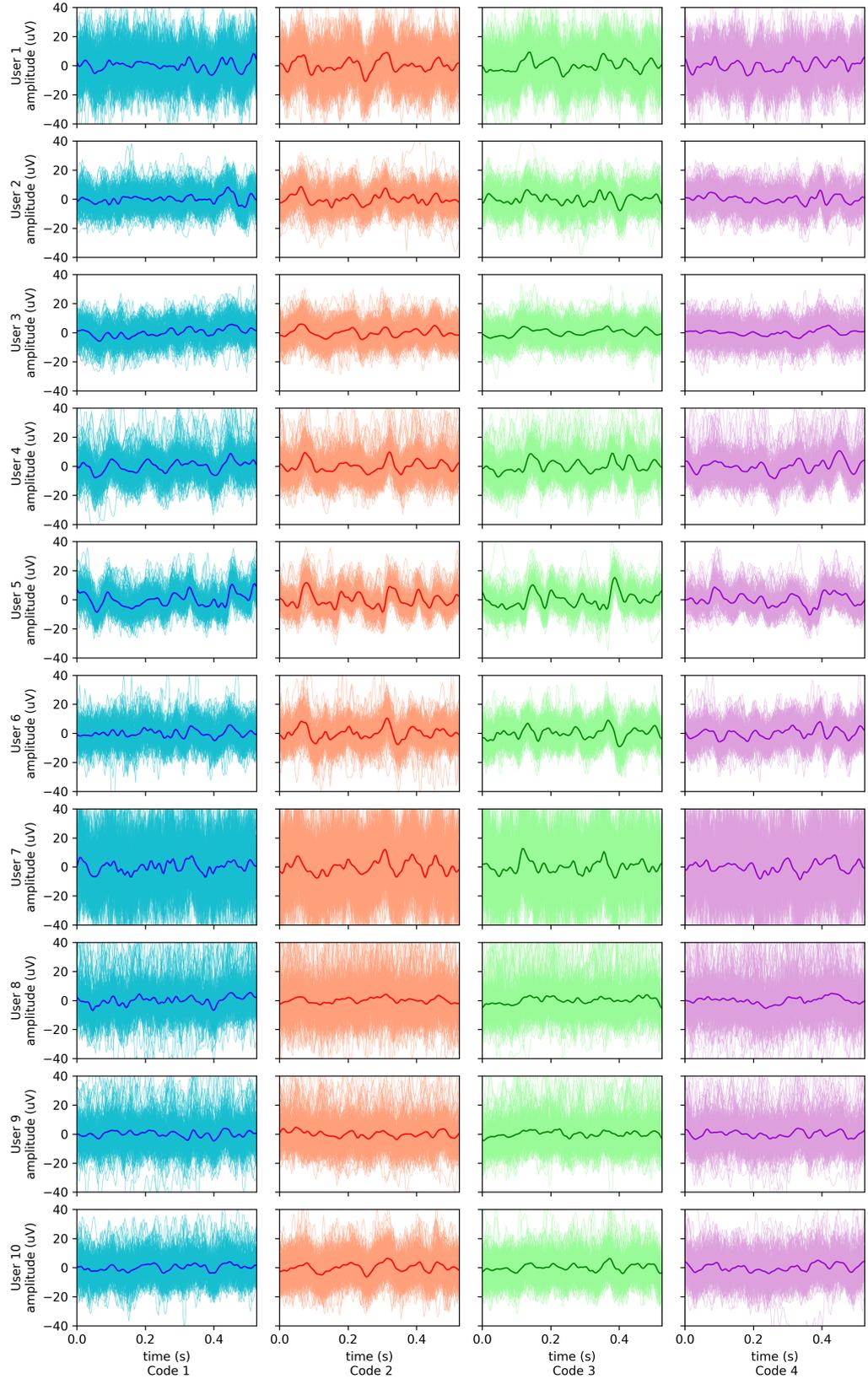


Figura 9.9: Gráfica de la topografía en un modelo de cabeza 2D, donde se representa el valor del filtro espacial obtenido con TRCA en cada uno de los canales utilizados para el experimento *Gold-Circular* con 63 bits. La figura muestra la topografía de los usuarios por filas y de los Gold Codes por columnas, donde el número de Gold Code se corresponde con los de la tabla 8.3.



A

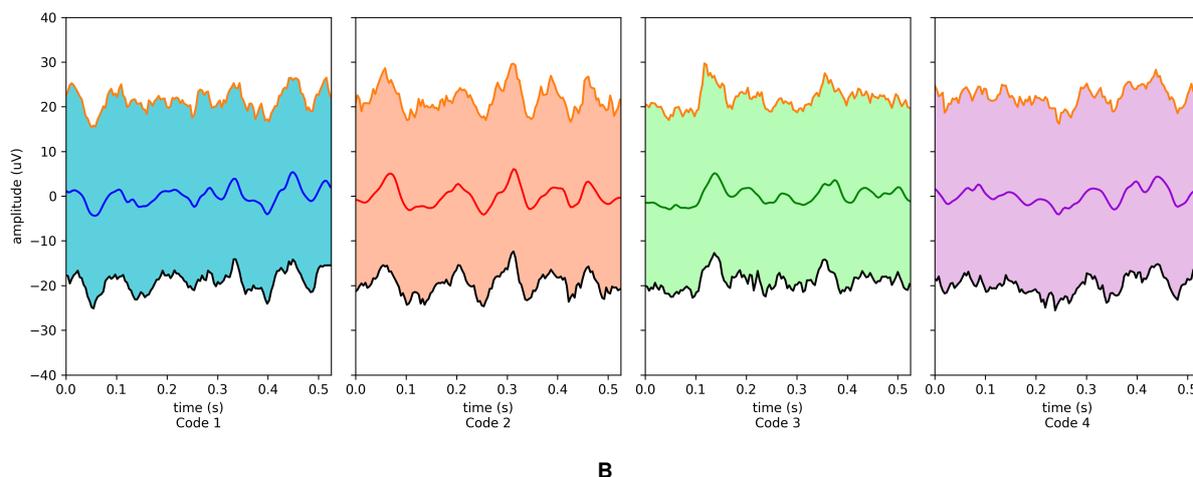


Figura 9.10: Respuesta cerebral de los usuarios en el experimento *Gold-Circular* con 63 bits. **(A)** Plantilla principal para cada uno de los usuarios tomada como promedio de la respuesta del EEG en el canal Oz sobre 200 ciclos de calibración. En azul se muestra la plantilla principal para el primer Gold Code, en naranja para el segundo, en verde para el tercero y en morado para el cuarto. **(B)** Intervalo de confianza al 95 % para los ciclos de calibración de todos los usuarios, así como la media obtenida, para cada uno de los Gold Codes. El número de Gold Code se corresponde con los de la tabla 8.3.

9.3. Resultados del cuestionario de satisfacción

La tabla 9.13 muestra los resultados obtenidos del cuestionario de satisfacción por cada una de las preguntas. Se puede observar que todos los usuarios afirman que les ha resultado interesante el sistema BCI desarrollado y que el paradigma de selección desde el punto de vista del usuario es fácil de entender e intuitivo. También se aprecia que los usuarios participantes creen que la sesión del análisis online les ha parecido excesiva. Es importante destacar que la mayor parte de usuarios consideran haber tenido mayor fatiga visual el experimento *Gold-Circular* que en *Circular Shifting*. En esta misma línea, los usuarios manifiestan que han tenido alguna dificultad más para seleccionar los comandos en *Gold-Circular* que en *Circular Shifting*, lo que está avalado por las precisiones obtenidas en las tablas anteriores (véase la tabla 9.12). Finalmente, hay un pensamiento generalizado de que el sistema responde con velocidad y fluidez, así como la predisposición a participar en estudios similares a este.

9.3. Resultados del cuestionario de satisfacción

Tabla 9.13: Respuestas al cuestionario del análisis online por parte de los 10 sujetos participantes.

Pregunta	Respuesta (%)				
	1	2	3	4	5
Me ha resultado interesante conocer y usar este sistema BCI	0	0	0	0	100
Me ha resultado difícil seleccionar los comandos deseados en el experimento <i>Gold-Circular</i>	60	10	30	0	0
Me ha resultado difícil seleccionar los comandos deseados en el experimento <i>Circular Shifting</i>	70	10	10	10	0
Me imagino la utilización de este tipo de sistemas en la vida cotidiana de ciertas personas	0	0	0	30	70
El sistema responde con suficiente velocidad y fluidez	0	0	10	40	50
La duración de la sesión me ha parecido excesiva	20	20	50	10	0
Estaría dispuesto a participar en un estudio de estas características	0	0	20	30	50
Los estímulos visuales del experimento <i>Gold-Circular</i> me han causado fatiga visual	0	30	30	30	10
Los estímulos visuales del experimento <i>Circular Shifting</i> me han causado fatiga visual	20	30	40	0	10
El paradigma de selección de comandos es intuitivo y fácil de entender	0	0	0	0	100

Capítulo 10

Discusión

Una vez expuestos todos los resultados del análisis realizado, es necesario discutirlos. Este capítulo incluye la viabilidad de la aplicación desarrollada, la comparación con otros estudios de índole similar y las limitaciones presentes en el sistema implementado.

10.1. Experimento offline

10.1.1. Base de datos de p -ary m -secuencias

Tras haber expuesto el conjunto de precisiones obtenidas utilizando CCA y TRCA en esta base de datos, surge la pregunta de qué método es más adecuado para un sistema BCI confiable. Son evidentes las altas precisiones obtenidas con ambos métodos ($> 98\%$ en el ciclo 10), sin embargo, ¿es más preciso CCA o TRCA? En la tabla 9.6, se observa que, para cualquier ciclo y cualquier base, los p -valores obtenidos tras comparar las precisiones utilizando CCA y TRCA son todos mayores que 0,05. Consecuentemente, se deduce que no existen diferencias significativas al utilizar un método u otro cuando se codifican los comandos utilizando versiones desplazadas temporalmente de p -ary m -secuencias. Por tanto, CCA y TRCA son métodos igualmente válidos para extraer filtros espaciales cuando se aplica *Circular Shifting* a m -secuencias de base p . Además [Martínez-Cagigal et al. \(2023\)](#) llegó a la conclusión de que, en términos de precisiones, no existen diferencias significativas en el ciclo 10 al utilizar una base p u otra con CCA, por lo que, como se han obtenido las mismas precisiones con TRCA en este ciclo (98,05%, 98,24%, 99,80%, 98,05% y 98,63% para las bases 2,3,5,7 y 11, respectivamente) es evidente que cualquiera de las bases p son efectivas para un sistema BCI preciso si se utiliza TRCA. Asimismo, se observa que a partir del cuarto ciclo, para cualquier base, se obtienen precisiones elevadas, es decir, $> 90\%$, lo que permitiría reducir el tiempo de selección de comandos un 60%. Por tanto, a la hora de configurar un sistema BCI basado en c-VEP que utilice p -ary m -secuencias que sea capaz de obtener desde el cuarto ciclo una precisión media de 90%, se podría utilizar tanto CCA como TRCA. Sin embargo, la implementación de TRCA es más sencilla que la de CCA, por lo que puede constituirse como un método fiable y fácil de computar.

10.1.2. Base de datos de Thielen *et al.*

Una vez demostrado que TRCA y CCA son igualmente válidos cuando se utiliza el paradigma *Circular Shifting* a m-secuencias, surge la pregunta de si ocurrirá lo mismo cuando se codifican los comandos utilizando Gold Codes. La tabla 9.7 demuestra que, en todos los ciclos, las precisiones medias entre 30 usuarios son todas superiores en TRCA que en CCA. Además, de los p -valores obtenidos se deduce que existen diferencias significativas entre TRCA y CCA cuando se codifican los comandos únicamente a través de Gold Codes (p -valor $< 0,05$). Aunque las precisiones para ambos métodos sean elevadas desde el ciclo 5 (i.e $>94\%$), TRCA se constituye como un método más válido que CCA para sistemas BCI basados en c-VEP utilizando únicamente Gold Codes para la codificación de los comandos.

10.2. Experimento online

Una vez expuestas todas las precisiones por ciclos, para cada uno de los usuarios, se han obtenidos precisiones medias de 95,31 % y 96,25 % para 31 y 63 bits, respectivamente en la aplicación desarrollada (experimento *Gold-Circular*). Estas altas precisiones demuestran fehacientemente la efectividad del sistema BCI basado en c-VEP desarrollado. Estos resultados sugieren que se ha desarrollado una aplicación en la que se han obtenido resultados similares a los del análisis offline, evidenciando así la optimización en tiempo real de la aplicación.

Las dos personas que obtuvieron una precisión en el ciclo 10 menor de 90 %, tanto en *Gold-Circular* con 31 bits como con 63 bits, manifestaron que podía ser causa del orden en el que se ejecutaban los experimentos. Este orden se escogía de forma aleatoria para cada usuario. Uno de los usuarios manifestó que su precisión inferior al resto de usuarios en el experimento pudo deberse a realizarlo el primero y no haber comprendido con exactitud las instrucciones de la persona encargada. Sin embargo, el otro sujeto manifestó que haber realizado este experimento al final pudo haber provocado una disminución de la precisión debido a la fatiga visual acumulada. Por ello, se ha aleatorizado el orden de los experimentos.

Asimismo, como se observa en la tabla 9.12, atendiendo a los p -valores obtenidos al comparar los experimentos *Gold-Circular* con 31 y 63 bits, únicamente existen diferencias significativas en los ciclos 1 y 2, sin embargo, del 3 en adelante, utilizar 31 bits y 63 en *Gold-Circular* es igualmente válido. Consecuentemente, se deduce que, a partir de un ciclo en adelante, no hay diferencias en términos de precisiones en utilizar 31 o 63 bits. Esta es una propiedad muy ventajosa que puede mejorar los sistemas BCI basados en c-VEP, ya que, si se codifican los comandos utilizando versiones desplazadas de varios Gold Codes y el algoritmo TRCA, se puede reducir a la mitad el tiempo de calibración (utilizando 31 bits) para obtener los mismos resultados que si este se duplicase (utilizando 63 bits). Por tanto, la utilización de 31 bits se constituye como una codificación más recomendable que

la de 63 bits al obtener precisiones medias estadísticamente similares pero con un tiempo de calibración reducido.

También es necesario destacar que la ligera superioridad en la media de precisiones del experimento *Circular Shifting* con respecto a *Gold-Circular* se puede deber a diversos factores. Uno de ellos puede ser la duración de la fase de calibración. Al durar menos en el experimento *Circular Shifting* (4 veces menos aproximadamente), los usuarios manifestaron a través de la encuesta de satisfacción una menor fatiga visual con respecto a la aplicación desarrollada. Sin embargo, *Gold-Circular* tiene una ventaja con respecto a *Circular Shifting*: el número de comandos a codificar puede ser mucho mayor. El experimento *Circular Shifting* (63 bits) utiliza $\tau = 4$, lo que permite codificar un total de 16 comandos. Martínez-Cagigal et al. (2021) asegura que τ no debería ser menor que 2 para conseguir precisiones lo suficientemente elevadas. De esta manera, si se redujera el *lag* del experimento *Gold-Circular* 63 bits de $\tau = 15$ a $\tau = 4$, se podrían codificar un total de 64 comandos, muy superior a los comandos codificables con el mismo τ en el experimento *Circular Shifting*.

Sin embargo, a pesar de esta leve superioridad, atendiendo de nuevo a los p -valores obtenidos al comparar *Circular Shifting* con *Gold-Circular* de 31 de la tabla 9.12, se observa que solo existen diferencias significativas en los tres primeros ciclos. Análogamente, las diferencias de precisión entre *Circular Shifting* con *Gold-Circular* de 63 bits no son significativas para ningún ciclo. Por tanto, ambas aplicaciones son igualmente válidas para desarrollar un sistema BCI basado en c-VEP. La elección del paradigma dependerá de las características que se busque en la aplicación. Es decir, si se requiere de menores tiempos de calibración, es recomendable utilizar la aplicación del experimento *Circular Shifting* con CCA, sin embargo, si se busca poder codificar un elevado número de comandos, es recomendable utilizar la aplicación del experimento *Gold-Circular* con TRCA.

Atendiendo a las figuras de las plantillas principales se observa que la respuesta cerebral de aquellos usuarios que han obtenido la máxima precisión carece prácticamente de ruido, para cualquiera de los experimentos. Sin embargo, aquellos usuarios que han obtenido precisiones más bajas, se corresponden con plantillas con más ruido y/o con una media de la respuesta cerebral cercana al cero. Por ejemplo, observando la figura 9.7, los usuarios 2, 4, 5, 6 y 7, que han obtenido la máxima precisión, se observa un patrón claro de la respuesta cerebral sobre los distintos ciclos de calibración, es decir, se perciben distintas componentes relacionadas con los VEP. Sin embargo, en esta misma figura, se aprecia mucho ruido en la respuesta cerebral del usuario 8, que es el que menor precisión ha obtenido en el último ciclo. Por tanto, las plantillas principales son consistentes y características para el mismo usuario.

Asimismo, se observa que las plantillas principales también son consistentes, para un mismo código, entre algunos usuarios. Por ejemplo, atendiendo a la figura 9.10, se observa un claro patrón para todos los códigos cuando se comparan las respuestas obtenidas por

los usuarios 4, 5 y 6.

Finalmente, atendiendo a las figuras de la topografía de cualquier experimento, se aprecia claramente que los filtros espaciales obtenidos por CCA y TRCA cumplen su función de ponderar la importancia de los canales a la hora de reflejar las respuestas cerebrales causadas por las iluminaciones de los c-VEP. Esto se contempla en que se está ponderando más la zona occipital, que es donde se encuentra el cortex visual primario. En estas figuras se observa que el electrodo Oz en todos los usuarios se pondera con un valor absoluto mayor y en algunos usuarios, también los electrodos PO7, POz y PO8.

10.3. Comparación con otros estudios

Como se ha explicado a lo largo de este documento, la aplicación desarrollada implementa un sistema BCI basado en c-VEP en el que se combina la utilización del paradigma *Circular Shifting* con la codificación de comandos a través de Gold Codes, junto con la extracción de filtros espaciales TRCA. A continuación, se muestra una comparación con otros estudios de índole similar, analizando los resultados obtenidos dependiendo de las características del sistema.

En primer lugar, existe bastante literatura que combina la utilización de m-secuencias, junto con sus versiones desplazadas, así como el uso de CCA. En concreto, el estudio de [Bin et al. \(2011\)](#) presenta un sistema BCI basado en c-VEP que utiliza una m-secuencia de 63 bits y *Circular Shifting* para codificar un total de 32 comandos. El *lag* utilizado es de 2 bits y la frecuencia de muestreo del monitor es de 60 Hz. Este autor utilizó CCA como método de procesamiento de señal, logrando alcanzar una precisión media de 85 % entre 12 sujetos. Sin embargo, el sistema desarrollado por [Gembler et al. \(2019\)](#), con las mismas características que el anterior, logró una precisión del 100 % con 18 sujetos. Por lo tanto, se observa que el método utilizado puede llegar a alcanzar una gran precisión, sin embargo, la cantidad de comandos a codificar está limitada al utilizar versiones desplazadas de un único código, en contraste con la aplicación desarrollada.

También existen varios artículos que incluyen versiones desplazadas de varios códigos generados por diversos algoritmos, junto con CCA. [Wei et al. \(2018\)](#) presenta un sistema BCI basado en c-VEP en el que combina la utilización de 3 códigos junto con *Circular Shifting* para codificar un total de 48 comandos. En concreto, utilizó dos códigos Golay y un Almost Perfect Autocorrelation (APA) y desplazando cada uno de ellos con un *lag* de 4. Estos se presentaron en un monitor de 60 Hz, logrando una precisión media de 91,67 % entre 12 sujetos, utilizando CCA. Otro estudio similar es el de [Liu et al. \(2018\)](#). Este autor, desarrolló un sistema BCI basado en c-VEP de 64 comandos, utilizando versiones desplazadas de *lag* 4 de un total de 4 códigos, 2 Golay y 2 APA de 64 bits. Con una frecuencia de muestro de 60 Hz, se logró obtener una precisión de 88,36 % entre 8 usuarios, de nuevo, utilizando CCA. Se observa que las precisiones obtenidas en la aplicación

Tabla 10.1: Resumen de los estudios con sistemas BCI basados en c-VEP con características similares al desarrollado. La columna “Nº códigos” hace referencia al número de códigos, por lo que, si aparece 1, se han utilizado desplazamientos circulares de m-secuencias. Si aparece un número mayor que 1, quiere decir que se ha utilizado algún otro tipo de secuencia pseudoaleatoria junto con el paradigma *Circular Shifting*.

Artículo	Número de comandos	Longitud código (bits)	Lag (bits)	Nº códigos	Nº Sujetos	Nº Ciclos	Precisión (%)	Método
Bin et al. (2011)	32	63	2	1	12	2	85,00	CCA
Gembler et al. (2019)	32	63	2	1	18	30	100,00	CCA
Wei et al. (2018)	48	64	4	3	12	5	91,67	CCA
Liu et al. (2018)	64	64	4	4	8	5	88,36	CCA
Sun et al. (2022)	120	31	1	4	22	3	76,58	TRCA
Sun et al. (2022)	120	31	1	4	22	9	95,02	TRCA
Este estudio	16	31	7	4	10	10	95,31	TRCA
Este estudio	16	63	15	4	10	10	96,25	TRCA

desarrollada, tanto con 31 bits como con 63, son más elevadas que en ambos estudios, lo que parece evidenciar que la combinación de Gold Codes y *Circular Shifting* con un *lag* adecuado junto con TRCA da mejores resultados.

Finalmente, el estudio de [Sun et al. \(2022\)](#) posee características muy similares a la aplicación desarrollada. En concreto, este autor utiliza 4 Gold Codes de 31 bits, junto con sus versiones desplazadas con un *lag* de 1 bit para codificar un total de 120 comandos. Estos estímulos se presentaron en un monitor de 60 Hz y se utilizó TRCA para la extracción de filtros espaciales. En este estudio se obtuvo una precisión media de 76,58% utilizando una sola trial y 95,02% utilizando 3 trials entre 22 sujetos. Se observa que la falta de precisión, posiblemente derivada de la elección de un *lag* tan pequeño, se ve compensada con mayores trials en la fase de test. La aplicación desarrollada en el presente Trabajo de Fin de Grado ha logrado una precisión media en el tercer ciclo en el experimento *Gold-Circular* con 31 bits de 68,12%, y con 63 bits de 83,75%. Asimismo, se alcanzó con 31 bits 94,06% y con 63 bits 95,31% utilizando 9 ciclos. Se observa que, para el experimento con con 31 bits, las precisiones de [Sun et al. \(2022\)](#), tanto para 3 como para 9 ciclos son levemente más elevadas. Esto se puede deber a que este autor utilizó un monitor con una tasa de refresco de 60 Hz y en la aplicación desarrollada 120 Hz.

La tabla 10.1 muestra un resumen de los estudios mencionados previamente, así como

sus características. Se observa que, en todos aquellos estudios que utilizan más de un tipo de código para codificar los comandos, las precisiones obtenidas con la aplicación desarrollada son más elevadas, tanto con 31 como con 63 bits.

10.4. Fortalezas y limitaciones

En este Trabajo de Fin de Grado se han realizado investigaciones innovadoras que han permitido desarrollar un proyecto novedoso acerca de sistemas BCI basados en c-VEP.

En primer lugar, hasta el momento, no se ha encontrado literatura que emplee TRCA como método de procesamiento de señales para la obtención de filtros espaciales al codificar los comandos de c-VEP mediante m-secuencias de base p , como se ha realizado en una de las partes del análisis offline. Además, se ha demostrado que no existen diferencias significativas entre utilizar CCA o TRCA en este tipo de secuencias cuando se aplican desplazamientos circulares. Asimismo, a través de la segunda parte del análisis offline, se ha demostrado la efectividad de TRCA sobre CCA como método de extracción de filtros espaciales cuando únicamente se utilizan Gold Codes. En esta misma línea, se ha desarrollado un innovador sistema BCI basado en c-VEP que combina el paradigma *Circular Shifting* con los Gold Codes, así como TRCA. Tras comprobar la viabilidad, precisión y efectividad de la aplicación, se ha demostrado en un análisis online con 10 sujetos sanos que no existen diferencias significativas a partir de un ciclo en adelante cuando se codifican los comandos con Gold Codes y sus desplazamientos circulares con 31 y 63 bits. Este pionero resultado tiene un potencial muy elevado, ya que utilizando 31 bits se reduce a la mitad el tiempo de calibración necesario para utilizar el sistema BCI.

A pesar de la efectividad del sistema BCI desarrollado avalada por las precisiones es fundamental realizar una autocrítica sobre las carencia y limitaciones de la aplicación.

En primer lugar, la mayor parte de los usuarios manifestaron a través del cuestionario de satisfacción haber padecido fatiga visual a la hora de realizar tanto la fase de calibración como la de test. Esto podría llegar a complicar el uso de la aplicación en la vida real y no únicamente en un entorno de investigación. Se ha demostrado que el uso de secuencias no binarias (p -ary) o el aumento de la frecuencia de refresco de la pantalla favorecen la disminución de fatiga visual. En esta misma línea, se ha demostrado que la utilización de TRCA funciona correctamente con secuencias de base p , con lo cual sería posible emplear el paradigma *Gold-Circular* con ellas para favorecer la comodidad del usuario. Además, otra de las posibles limitaciones que puede presentar este sistema es que funciona de forma síncrona. En el supuesto de que esta aplicación se utilizase en la vida cotidiana para comunicarse, si un usuario quisiera dejar de escribir, sería incapaz de ello, ya que la matriz de comandos seguiría iluminándose siguiendo sus correspondientes secuencias pseudoaleatorias. Dicho de otra manera, no está implementado ningún comando para parar de escribir. Se ha utilizado un equipo con cables para garantizar que todas

las muestras del registro EEG se reciban correctamente, lo que limita la portabilidad. Idealmente, se debería emplear un equipo inalámbrico. Sin embargo, en estos equipos las muestras se etiquetan al llegar al destino, en lugar de hacerlo cuando se generan en el amplificador. Esto causa un pequeño retraso que puede afectar negativamente si los desfases entre desplazamientos son bajos. Finalmente, se ha comprobado la efectividad y viabilidad de la aplicación con sujetos sanos. No obstante, sería apropiado realizar las mismas pruebas con personas que tengan algún tipo de discapacidad motora para comprobar si el sistema obtiene las mismas precisiones o si disminuyen.

Capítulo 11

Conclusiones y líneas futuras

Este capítulo se extraen las conclusiones finales tras la finalización del proyecto y las futuras líneas de investigación.

11.1. Conclusiones

En este Trabajo de Fin de Grado se ha logrado alcanzar los objetivos principales planteados inicialmente, que son: (1) Diseñar un sistema BCI basado en c-VEP utilizando el paradigma *Circular Shifting* con m-secuencias y Gold Codes; (2) Implementar la aplicación en Python y C# (Unity); (3) Evaluar el sistema a través de un análisis offline y online.

Para la consecución de estos objetivos, se planteó una metodología de trabajo llamada SCORE basada en SCRUM que se ha seguido hasta la finalización de este Trabajo de Fin de Grado. En esta misma línea, se realizó una planificación temporal del total de tareas a realizar, a través de Diagramas de Gantt, así como una planificación económica del proyecto.

A continuación, se confeccionó un exhaustivo estudio sobre sistemas BCI basados en c-VEP. Primeramente, se ha realizado un prólogo acerca de la anatomía general del SNC y en concreto del cerebro, donde además se analiza la adquisición de la actividad cerebral, dando mayor importancia al EEG. Debido a la naturaleza de este tipo de sistemas BCI, se introdujeron las señales de control más populares en este campo de trabajo.

Una vez adquiridos e integrados la mayor parte de conceptos teóricos, fue necesario realizar una búsqueda bibliográfica del estado del arte más actual. En concreto, se seleccionó tres artículos de gran impacto que muestran con claridad los avances obtenidos hasta el momento en materia de sistemas BCI basados en c-VEP junto con Gold Codes y TRCA.

Fue necesario elaborar un estudio teórico más profundo sobre el paradigma c-VEP, donde además se analizaron las distintas formas de generación de secuencias de números pseudoaleatorias utilizadas en la aplicación desarrollada, como son m-secuencias y Gold

Codes. Asimismo, se presentaron los algoritmos de extracción de filtros espaciales CCA y TRCA, tanto de forma teórica como su computación, así como las distintas *pipelines* de procesado que posteriormente se implementaron en la aplicación.

En cuanto al diseño y desarrollo del sistema BCI, una vez presentados los materiales y el equipo a utilizar, se realizó una extensa documentación acerca de la aplicación MEDUSA[©] incluyendo sus características principales, la distinción entre MEDUSA[©] Kernel y MEDUSA[©] Platform, los protocolos utilizados para garantizar la comunicación y la arquitectura de las aplicaciones desarrolladas dentro de MEDUSA[©]. Este estudio previo fue esencial para facilitar el desarrollo de la aplicación en Python, donde se implementó la generación de Gold Codes como el algoritmo TRCA y C# (Unity) para implementar la interfaz gráfica de la aplicación a través de una matriz de comandos.

Una vez finalizada la fase de desarrollo de la aplicación, se procedió a evaluar el sistema implementado. Para ello, se realizó un análisis offline con dos extensas bases de datos y un análisis online con un total de 10 sujetos y un protocolo de evaluación previamente definido y establecido. En el primero se demostró que CCA y TRCA tienen comportamientos estadísticamente similares en cuanto a su uso con m-secuencias binarias y no binarias (de base p), tanto en precisiones como en tiempos de selección de comandos. No obstante, se probó que TRCA obtiene mejores resultados significativos que CCA cuando únicamente se utilizan Gold Codes.

En cuanto al análisis online, la aplicación desarrollada obtuvo una precisión media de $95,31\% \pm 6,13$ con 31 bits y $96,25\% \pm 7,10$ con 63 bits, demostrando así que el uso de TRCA como método de extracción de filtros espaciales es factible en el paradigma *Gold-Circular*. Asimismo, se ha demostrado que los resultados obtenidos con esta aplicación al utilizar 31 y 63 bits son estadísticamente similares, lo que permite reducir la fase de calibración a la mitad. A continuación, se realizó una comparativa con respecto a otras literaturas similares, arrojando la aplicación desarrollada resultados superiores, demostrando fehacientemente su efectividad.

Los filtros espaciales obtenidos por CCA y TRCA se focalizan en la región occipital del cerebro. Esto es particularmente relevante dado que la región occipital alberga el córtex visual primario, una estructura crucial para el procesamiento de la información visual. La aplicabilidad de los filtros obtenidos por CCA y TRCA ponderan más los electrodos que se encuentran en esta área específica, lo cual permite captar de manera precisa las variaciones en las señales del córtex visual primario. Esta precisión es de suma importancia en estudios y aplicaciones que involucren la percepción visual, como los sistemas BCI basados en c-VEP, donde la detección y el análisis exacto de los patrones de actividad cerebral relacionados con estímulos visuales son esenciales.

Por todo esto, se concluye que los sistemas BCI basado en c-VEP obtienen resultados precisos, utilizando tiempos de calibración reducidos. Además, la utilización conjunta de

TRCA con Gold Codes y *Circular Shifting* arroja muy buenos resultados con la gran ventaja de que el número de comandos a codificar puede ser mucho mayor que si se utilizase únicamente versiones desplazadas de una m -secuencia.

11.2. Líneas futuras

En consonancia con las limitaciones encontradas para la aplicación desarrollada, a continuación se presentan varias líneas futuras de desarrollo:

- Para reducir la fatiga visual que han padecido los usuarios, una posible línea de investigación es modificar la opacidad de cada uno de los comandos. También se podría modificar el fondo, dando al usuario la capacidad de elegir una imagen que le resulte familiar. Asimismo, se podría aumentar la frecuencia de refresco o utilizar m -secuencias de base p , para las cuales se ha demostrado en el análisis offline que TRCA funciona estadísticamente similar.
- Para lograr el funcionamiento de la aplicación de forma asíncrona y mejorar la interacción del usuario, se podrían implementar las siguientes mejoras. En primer lugar, incorporar una etapa que detecte si el usuario está mirando o no la matriz de comandos de forma dinámica, utilizando tecnología de seguimiento ocular (*eye-tracking*). Cuando el sistema detecte que el usuario no está mirando la matriz, esta podría ocultarse automáticamente y reaparecer cuando el usuario vuelva a mirarla. Además, se podría implementar un sistema híbrido que permita el control de la aplicación mediante múltiples señales de control. Por ejemplo, se puede utilizar el parpadeo o incorporar el uso de movimientos mandibulares como una señal para iniciar o parar la matriz de comandos.
- Finalmente, para hacer la aplicación más interactiva, se podría implementar la codificación de los comandos a través de otro tipo de secuencias aleatorias. Además, en el menú de configuración de la aplicación se añadiría un seleccionable para escoger que tipo de codificación se quiere utilizar. Se haría lo mismo para seleccionar un algoritmo de extracción de filtros espaciales u otro.

Parte II

Manual de la Aplicación

Capítulo 12

Instalación

12.1. Manual de instalación

A continuación, se muestran los pasos a seguir para instalar la aplicación MEDUSA[©] en un dispositivo *Windows*.

- 1) Instalar Python 3.8, 3.9 o 3.10, siendo esta última la versión más recomendable.
- 2) Instalar Git para clonar el repositorio de GitHub de MEDUSA[©] Platform (<https://github.com/medusabci/medusa-platform>).
- 3) Clonar el repositorio de la versión v2023.0.1 a través del siguiente comando en la interfaz de comandos del sistema:

```
git clone -b v2023.0.1 --single-branch https://github.com/medusabci/medusa-platform.git
```

Otra alternativa que permitiría saltarse el paso 2 es acceder al repositorio de GitHub de la versión especificada anteriormente a través de <https://github.com/medusabci/medusa-platform/tree/v2023.0.1>, descargarlo en .zip y descomprimirlo.

- 4) Clonar el repositorio de MEDUSA[©] Kernel necesario para la aplicación:

```
git clone --branch developers https://github.com/medusabci/medusa-kernel.git
```

Alternativamente, descargarlo en zip desde <https://github.com/medusabci/medusa-kernel/tree/developers> y descomprimirlo.

- 5) Acceder a la carpeta `medusa-platform` y crear un directorio que será el *Virtual Environment* llamado `venv`. Para realizarlo a través de la línea de comandos:

```
cd \path\to\medusa-platform
```

```
mkdir venv
```

- 6) Crear un *Virtual Environment* en el directorio `venv` a través de un IDE o de la línea de comandos:

```
py -m venv venv o python -m venv venv
```

- 7) Activar el *Virtual Environment* e instalar los paquetes necesarios almacenados en `requirements.txt` a través de un IDE o de la línea de comandos:

```
venv\Scripts\activate
```

```
pip install -r requirements.txt
```

- 8) Instalar el MEDUSA[©] Kernel clonado ejecutando en la terminal

```
pip install \path\to\medusa-kernel
```

- 9) Ejecutar la aplicación a través de `medusa.exe`. Para poder utilizarla es necesario crearse un usuario y una contraseña.
- 10) Para instalar la aplicación, es necesario crear un proyecto vacío. Esto se consigue clicando sobre `Developer tools\Create app\`. A continuación, se rellena el identificador con `cvep_speller_gold_circular`, el nombre con `c-VEP Speller Gold-Circular`, la extensión con `cvep` y se selecciona el `Empty project`.
- 11) Acceder al directorio
`path\to\medusa-platform\src\accounts\nombre_usuario\apps\
cvep_speller_gold_circular` y sustituir el contenido por los archivos de la única carpeta adjuntada en el *OneDrive*.
- 12) Reiniciar MEDUSA[©].
- 13) Instalar *Signal Generator v2.3*, aplicación desarrollada por el GIB para generar una señal artificial que emula ser un EEG. El instalador se encuentra en https://github.com/medusabci/lsl-signal-generator/blob/master/signal_generator_amd64_v2.3.exe.

12.2. Contenido adjunto

Junto con el presente documento, se adjunta en el enlace *OneDrive* el código de la aplicación desarrollada.

```
|_ LICENSE  
|_ README.md  
|_ app_constants.py  
|_ app_controller.py  
|_ config.py
```

```

├── config.ui
├── config_encoding.ui
├── config_target.ui
├── icon.png
├── info
├── main.py
├── requirements.txt
├── settings.py
├── unity
│   ├── MonoBleedingEdge
│   ├── UnityCrashHandler64.exe
│   ├── UnityPlayer.dll
│   ├── c-VEP Speller.exe
│   └── c-VEP Speller_Data

```

Nótese que únicamente se adjunta en el *OneDrive* el código de la aplicación desarrollado. Debido a la arquitectura de las aplicaciones en MEDUSA[©], no aparece el código desarrollado en Unity sino únicamente el ejecutable correspondiente (`unity/c-VEP Speller.exe`). Asimismo, no aparece adjuntado el código desarrollado acerca de la implementación de TRCA, Gold Codes y el clasificador y modelo del paradigma *Gold-Circular* ya que se ha implementado directamente en MEDUSA[©] Kernel (ver pasos 4 y 8 de la sección anterior). Para comprobar el código correspondiente acceder a:

- `path\to\medusa-kernel\spatial_filtering.py` : Clase TRCA
- `path\to\medusa-kernel\bci\cvcp_spellers.py`: Clases GOLD_CODES, TRCAGoldCodesClassifier y CVEPModelTRCAGoldCodes.

Bibliografía

- Allison, B.Z., Faller, J., Neuper, C., 2012. BCIs That Use Steady-State Visual Evoked Potentials or Slow Cortical Potentials, in: Brain-Computer Interfaces: Principles and Practice. volume 9780195388855. doi:10.1093/acprof:oso/9780195388855.003.0014.
- Ausum Cloud, . Scrum: Qué es y por qué se ha convertido en una de las metodologías ágiles más populares. URL <https://ausum.cloud/scrum-metodologia-agil-mas-popular-en-empresas/> [Accessed: May. 13, 2024].
- Berger, H., 1929. Über das elektrenkephalogramm des menschen. Archiv für Psychiatrie und Nervenkrankheiten 87, 527–570. URL: <https://api.semanticscholar.org/CorpusID:10835361>.
- Bin, G., Gao, X., Wang, Y., Hong, B., Gao, S., 2009. VEP-based brain-computer interfaces: Time, frequency, and code modulations. IEEE Computational Intelligence Magazine 4. doi:10.1109/MCI.2009.934562.
- Bin, G., Gao, X., Wang, Y., Li, Y., Hong, B., Gao, S., 2011. A high-speed BCI based on code modulation VEP, in: Journal of Neural Engineering. doi:10.1088/1741-2560/8/2/025015.
- Biomedical Engineering Group at University of Valladolid, 2022. MEDUSA[©] Official Documentation. URL <https://docs.medusabci.com/platform/> [Accessed: Jan. 20, 2023].
- Blink Learning, . Áreas sensoriales del cerebro. URL <https://www.blinklearning.com/coursePlayer/clases2.php?editar=0&idcurso=1546420&idclase=88349450&modo=0> [Accessed: Feb. 10, 2024].
- Castillo, G., de Jorge, J., 2015. Anatomía y fisiología del sistema nervioso central. Ebook CEU Ediciones, Fundación Univ. San Pablo. URL: <https://books.google.es/books?id=4AfqBgAAQBAJ>.
- Chenane, K., Touati, Y., Boubchir, L., Daachi, B., 2019. Neural net-based approach to EEG signal acquisition and classification in BCI applications. doi:10.3390/computers8040087.
- Cooper, M., Seber, G.A.F., 1985. Multivariate Observations. Journal of Marketing Research 22. doi:10.2307/3151376.
- Dattalo, P.V., 2014. A demonstration of canonical correlation analysis with orthogonal rotation to facilitate interpretation. VCU Scholars Compass, Social Work Publications .
- Dev-Cademy Media Inc., a. Functional programming in python. URL <https://realpython.com/courses/functional-programming-python/> [Accessed: Aug. 30, 2023].
- Dev-Cademy Media Inc., b. Hand-on python 3 concurrency with the asyncio module. URL <https://realpython.com/courses/python-3-concurrency-asyncio-module/> [Accessed: Aug. 25, 2023].
- Dev-Cademy Media Inc., c. Speed up python with concurrency. URL <https://realpython.com/courses/speed-python-concurrency/> [Accessed: Aug. 21, 2023].

- Dev-Cademy Media Inc., d. Threading in python. URL <https://realpython.com/courses/threading-python/> [Accessed: Aug. 15, 2023].
- Dinan, E.H., Jabbari, B., 1998. Spreading codes for direct sequence CDMA and wideband CDMA cellular networks. IEEE Communications Magazine 36. doi:[10.1109/35.714616](https://doi.org/10.1109/35.714616).
- Eniun, . Metodología Scrum para el desarrollo de software ágil. URL <https://www.eniun.com/metodologia-scrum-desarrollo-software-agil/> [Accessed: Dic. 14, 2023].
- Escuela de Ingeniería Informática de Segovia, 2023. Competencias generales y específicas. URL https://www.uva.es/export/sites/uva/2.docencia/2.01.grados/2.01.02.ofertaformativagrados/_documentos/1296564784179_competencias.pdf [Accessed: May. 13, 2024].
- Everitt, B.S., Krzanowski, W., 1989. Principles of Multivariate Analysis. Journal of the Royal Statistical Society. Series A (Statistics in Society) 152. doi:[10.2307/2982935](https://doi.org/10.2307/2982935).
- Ferree, T.C., Luu, P., Russell, G.S., Tucker, D.M., 2001. Scalp electrode impedance, infection risk, and EEG data quality. Clinical Neurophysiology 112. doi:[10.1016/S1388-2457\(00\)00533-2](https://doi.org/10.1016/S1388-2457(00)00533-2).
- Friedman, N., Fekete, T., Gal, K., Shriki, O., 2019. EEG-based prediction of cognitive load in intelligence tests. Frontiers in Human Neuroscience 13. doi:[10.3389/fnhum.2019.00191](https://doi.org/10.3389/fnhum.2019.00191).
- Gemler, F., Benda, M., Saboor, A., Volosyak, I., 2019. A multi-target c-VEP-based BCI speller utilizing n-gram word prediction and filter bank classification, in: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. doi:[10.1109/SMC.2019.8914235](https://doi.org/10.1109/SMC.2019.8914235).
- Golomb, S., 1955. Sequences with Randomness Properties. Martyn. Marietta, Glenn L. Martin Company. URL: <https://books.google.es/books?id=kJKnNAAACAAJ>.
- Google Career Certificates, . Crash course on python. URL <https://www.coursera.org/learn/python-crash-course> [Accessed: Jul. 20, 2023].
- Gramann, K., Ferris, D.P., Gwin, J., Makeig, S., 2014. Imaging natural cognition in action. doi:[10.1016/j.ijpsycho.2013.09.003](https://doi.org/10.1016/j.ijpsycho.2013.09.003).
- Hadjiaros, M., Neokleous, K., Shimi, A., Avraamides, M.N., Pattichis, C.S., 2023. Virtual Reality Cognitive Gaming Based on Brain Computer Interfacing: A Narrative Review. IEEE Access 11, 18399–18416. doi:[10.1109/ACCESS.2023.3247133](https://doi.org/10.1109/ACCESS.2023.3247133).
- Hall, J.E., Guyton, A.C., 2016. Tratado de Fisiología médica. 13 ed.
- Hancock, G.R., Stapleton, L.M., Mueller, R.O., 2019. Canonical correlation analysis, in: The reviewer's guide to quantitative methods in the social sciences. second edition ed.. Routledge, New York. doi:[10.4324/9781315755649-3](https://doi.org/10.4324/9781315755649-3).
- Härdle, W.K., Simar, L., 2019. Applied Multivariate Statistical Analysis. doi:[10.1007/978-3-030-26006-4](https://doi.org/10.1007/978-3-030-26006-4).
- Harvard University, . Using python for research. URL <https://www.edx.org/learn/python/harvard-university-using-python-for-research> [Accessed: Jul. 30, 2023].
- Hicks, M., Foster, J.S., 2010. Adapting Scrum to Managing a Research Group. Communications of the ACM .

- Hidalgo, E.S., 2019. Adapting the scrum framework for agile project management in science: case study of a distributed research initiative. *Heliyon* 5. doi:[10.1016/j.heliyon.2019.e01447](https://doi.org/10.1016/j.heliyon.2019.e01447).
- Horn, R.A., Johnson, C.R., 2012. Hermitian matrices, symmetric matrices, and congruences, in: *Matrix Analysis*. Cambridge University Press, Cambridge, pp. 225–312. doi:[10.1017/CB09781139020411.007](https://doi.org/10.1017/CB09781139020411.007).
- Hotelling, H., 1935. The most predictable criterion. *Journal of Educational Psychology* 26. doi:[10.1037/h0058165](https://doi.org/10.1037/h0058165).
- Katz, J., Smith, Z.T., 2023. Scientific method, in: *Translational Interventional Radiology*, pp. 7–10. doi:[10.1016/B978-0-12-823026-8.00043-2](https://doi.org/10.1016/B978-0-12-823026-8.00043-2).
- Kedia, D., 2012. Comparative analysis of peak correlation characteristics of non-orthogonal spreading codes for wireless systems. *International Journal of Distributed and Parallel systems* 3. doi:[10.5121/ijdps.2012.3307](https://doi.org/10.5121/ijdps.2012.3307).
- Kemmelmeier, M., 2016. Cultural differences in survey responding: Issues and insights in the study of response biases. *International Journal of Psychology* 51, 439–444. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ijop.12386>, doi:<https://doi.org/10.1002/ijop.12386>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/ijop.12386>.
- Kleber, B., Birbaumer, N., 2005. Direct brain communication: Neuroelectric and metabolic approaches at Tübingen. doi:[10.1007/s10339-004-0045-8](https://doi.org/10.1007/s10339-004-0045-8).
- Leblanc, A., 2004. Encephalo-Peripheral Nervous System. doi:[10.1007/978-3-642-56435-2](https://doi.org/10.1007/978-3-642-56435-2).
- Liu, Y., Wei, Q., Lu, Z., 2018. A multi-target brain-computer interface based on code modulated visual evoked potentials. *PLoS ONE* 13. doi:[10.1371/journal.pone.0202478](https://doi.org/10.1371/journal.pone.0202478).
- Luck, S., 2005. *An Introduction to The Event-Related Potential Technique*.
- Martínez-Cagigal, V., 2020. Toward Practical P300-based Brain-Computer Interfaces: Asynchrony, Channel Optimization and Assistive Applications. Ph.D. thesis. Universidad de Valladolid. URL: <https://uvadoc.uva.es/bitstream/handle/10324/47516/Tesis1832-210719.pdf>.
- Martínez-Cagigal, V., Santamaría-Vázquez, E., Pérez-Velasco, S., Marcos-Martínez, D., Moreno-Calderón, S., Hornero, R., 2023. Non-binary m-sequences for more comfortable brain-computer interfaces based on c-VEPs. *Expert Systems with Applications* 232. doi:[10.1016/j.eswa.2023.120815](https://doi.org/10.1016/j.eswa.2023.120815).
- Martínez-Cagigal, V., Thielen, J., Santamaría-Vázquez, E., Pérez-Velasco, S., Desain, P., Hornero, R., 2021. Brain-computer interfaces based on code-modulated visual evoked potentials (c-VEP): A literature review. *Journal of Neural Engineering* 18. doi:[10.1088/1741-2552/ac38cf](https://doi.org/10.1088/1741-2552/ac38cf).
- Martišius, I., Damaševičius, R., 2016. A prototype SSVEP based real time BCI gaming system. *Computational Intelligence and Neuroscience* 2016. doi:[10.1155/2016/3861425](https://doi.org/10.1155/2016/3861425).
- MedlinePlus, National Library of Medicine, . Sistema nervioso central y sistema nervioso periférico. URL https://medlineplus.gov/spanish/ency/esp_imagepages/8679.htm [Accessed: May. 10, 2024].
- Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A., 1996. *Handbook of applied cryptography*. doi:[10.2307/2589608](https://doi.org/10.2307/2589608).
- Millan, J.d.R., Galan, F., Lew, E., Chavarriaga, R., 2008. Non-invasive brain-machine interaction. *International Journal of Pattern Recognition and Artificial Intelligence* 22. doi:[10.1142/S0218001408006600](https://doi.org/10.1142/S0218001408006600).

- Mitra, A., 2008. On Pseudo-Random and Orthogonal Binary Spreading Sequences. *International Journal of Electronics and Communication Engineering* 2.
- Moore, K.L., Dailey, A.F., Agur, A.M., 2013. MOORE Anatomia con orientacion clinica. volume 53. doi:[10.1017/CB09781107415324.004](https://doi.org/10.1017/CB09781107415324.004).
- Nagel, S., Dreher, W., Rosenstiel, W., Spüler, M., 2018. The effect of monitor raster latency on VEPs, ERPs and Brain-Computer Interface performance. *Journal of Neuroscience Methods* 295. doi:[10.1016/j.jneumeth.2017.11.018](https://doi.org/10.1016/j.jneumeth.2017.11.018).
- Nakanishi, M., Wang, Y., Chen, X., Wang, Y.T., Gao, X., Jung, T.P., 2018. Enhancing detection of ssveps for a high-speed brain speller using task-related component analysis. *IEEE Transactions on Biomedical Engineering* 65. doi:[10.1109/TBME.2017.2694818](https://doi.org/10.1109/TBME.2017.2694818).
- Nicolas-Alonso, L.F., Gomez-Gil, J., 2012. Brain computer interfaces, a review. *Sensors (Basel)* 12, 1211–79. doi:[10.3390/s120201211](https://doi.org/10.3390/s120201211).
- Nunez, P.L., 2012. Electric and Magnetic Fields Produced by the Brain, in: *Brain-Computer Interfaces: Principles and Practice*. doi:[10.1093/acprof:oso/9780195388855.003.0003](https://doi.org/10.1093/acprof:oso/9780195388855.003.0003).
- Nunez, P.L., Srinivasan, R., Westdorp, A.F., Wijesinghe, R.S., Tucker, D.M., Silberstein, R.B., Cadusch, P.J., 1997. EEG coherency I: Statistics, reference electrode, volume conduction, Laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalography and Clinical Neurophysiology* 103. doi:[10.1016/S0013-4694\(97\)00066-7](https://doi.org/10.1016/S0013-4694(97)00066-7).
- Ogawa, S., Lee, T.M., Kay, A.R., Tank, D.W., 1990. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences of the United States of America* 87. doi:[10.1073/pnas.87.24.9868](https://doi.org/10.1073/pnas.87.24.9868).
- Pastor, M.A., Artieda, J., Arbizu, J., Valencia, M., Masdeu, J.C., 2003. Human Cerebral Activation during Steady-State Visual-Evoked Responses. *Journal of Neuroscience* 23. doi:[10.1523/jneurosci.23-37-11621.2003](https://doi.org/10.1523/jneurosci.23-37-11621.2003).
- Pfurtscheller, G., Lopes Da Silva, F.H., 1999. Event-related EEG/MEG synchronization and desynchronization: Basic principles. doi:[10.1016/S1388-2457\(99\)00141-8](https://doi.org/10.1016/S1388-2457(99)00141-8).
- Pfurtscheller, G., McFarland, D.J., 2012. 227BCIs That Use Sensorimotor Rhythms, in: *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press. doi:[10.1093/acprof:oso/9780195388855.003.0013](https://doi.org/10.1093/acprof:oso/9780195388855.003.0013).
- Picton, T.W., 1992. The P300 wave of the human event-related potential. *Journal of Clinical Neurophysiology* 9. doi:[10.1097/00004691-199210000-00002](https://doi.org/10.1097/00004691-199210000-00002).
- Pinti, P., Tachtsidis, I., Hamilton, A., Hirsch, J., Aichelburg, C., Gilbert, S., Burgess, P.W., 2020. The present and future use of functional near-infrared spectroscopy (Fnirs) for cognitive neuroscience. *Annals of the New York Academy of Sciences* 1464. doi:[10.1111/nyas.13948](https://doi.org/10.1111/nyas.13948).
- Pombano, L.J., Evans, D.M., 2013. *Cerebellum: Anatomy, functions and disorders*.
- Rachid, C.L., Stopa, G.R., 2019. Scrum: Metodologia Ágil Como Ferramenta De Gerenciamento De Projetos. *CES Revista* 33.
- Ramsey, N.F., 2012. Signals Reflecting Brain Metabolic Activity, in: *Brain-Computer Interfaces: Principles and Practice*. doi:[10.1093/acprof:oso/9780195388855.003.0004](https://doi.org/10.1093/acprof:oso/9780195388855.003.0004).

- Rao, K.D., Swamy, M.N., 2018. Digital signal processing: Theory and practice. doi:[10.1007/978-981-10-8081-4](https://doi.org/10.1007/978-981-10-8081-4).
- Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., Lécuyer, A., 2010. OpenViBE: An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. Presence: Teleoperators and Virtual Environments 19. doi:[10.1162/pres.19.1.35](https://doi.org/10.1162/pres.19.1.35).
- Rønning, F., 2021. The Role of Fourier Series in Mathematics and in Signal Theory. International Journal of Research in Undergraduate Mathematics Education 7, 189–210. doi:[10.1007/s40753-021-00134-z](https://doi.org/10.1007/s40753-021-00134-z).
- Rykaczewski, K., Nikadon, J., Duch, W., Piotrowski, T., 2021. supFunSim: Spatial Filtering Toolbox for EEG. Neuroinformatics 19. doi:[10.1007/s12021-020-09464-w](https://doi.org/10.1007/s12021-020-09464-w).
- Santamaría-Vázquez, E., Martínez-Cagigal, V., Marcos-Martínez, D., Rodríguez-González, V., Pérez-Velasco, S., Moreno-Calderón, S., Hornero, R., 2023. Medusa©: A novel python-based software ecosystem to accelerate brain-computer interface and cognitive neuroscience research. Computer Methods and Programs in Biomedicine 230. doi:[10.1016/j.cmpb.2023.107357](https://doi.org/10.1016/j.cmpb.2023.107357).
- Sarwate, D.V., Pursley, M.B., 1980. Crosscorrelation properties of pseudorandom and related sequences. Proceedings of the IEEE 68, 593–619. URL: <https://api.semanticscholar.org/CorpusID:6179951>.
- Sassa, A.C., de Almeida, I.A., Pereira, T.N.F., de Oliveira, M.S., 2023. Scrum: A Systematic Literature Review. International Journal of Advanced Computer Science and Applications 14. doi:[10.14569/IJACSA.2023.0140420](https://doi.org/10.14569/IJACSA.2023.0140420).
- Schwaber, K., Sutherland, J., 2011. The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game.
- Seeck, M., Koessler, L., Bast, T., Leijten, F., Michel, C., Baumgartner, C., He, B., Beniczky, S., 2017. The standardized EEG electrode array of the IFCN. doi:[10.1016/j.clinph.2017.06.254](https://doi.org/10.1016/j.clinph.2017.06.254).
- Sellers, E.W., Arbel, Y., Donchin, E., 2012. BCIs That Use P300 Event-Related Potentials, in: Brain-Computer Interfaces: Principles and Practice. doi:[10.1093/acprof:oso/9780195388855.003.0012](https://doi.org/10.1093/acprof:oso/9780195388855.003.0012).
- Sharma, A., Anwar, T., Sharma, D.K., 2014. Simulation of gold code sequences for spread spectrum application. URL: <https://api.semanticscholar.org/CorpusID:202896033>.
- Sklar, B., 2001. Digital Communications: Fundamentals and Applications. Signals .
- Snell, R., 2010. Neuroanatomía clínica. Clinical Neuroanatomy 7.
- Soufneyestani, M., Dowling, D., Khan, A., 2020. Electroencephalography (EEG) technology applications and available devices. doi:[10.3390/app10217453](https://doi.org/10.3390/app10217453).
- Srinivasan, R., 2012. Acquiring Brain Signals from Outside the Brain, in: Brain-Computer Interfaces: Principles and Practice. doi:[10.1093/acprof:oso/9780195388855.003.0006](https://doi.org/10.1093/acprof:oso/9780195388855.003.0006).
- Standring, S., 2016. Gray's Anatomy: The Anatomical Basis of Clinical Practice. 41 ed., Elsevier Ltd, London.
- Sun, Q., Zheng, L., Pei, W., Gao, X., Wang, Y., 2022. A 120-target brain-computer interface based on code-modulated visual evoked potentials. Journal of Neuroscience Methods 375. doi:[10.1016/j.jneumeth.2022.109597](https://doi.org/10.1016/j.jneumeth.2022.109597).

- Sutter, E.E., 1984. Visual evoked response as a communication channel.
- Sutter, E.E., 1992. The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Applications* 15. doi:[10.1016/0745-7138\(92\)90045-7](https://doi.org/10.1016/0745-7138(92)90045-7).
- Swartz Center for Computational Neuroscience, University of California, San Diego, . LSL. URL <https://labstreaminglayer.org/> [Accessed: May. 07, 2024].
- Tanaka, H., 2020. Group task-related component analysis (gtrca): a multivariate method for inter-trial reproducibility and inter-subject similarity maximization for eeg data analysis. *Scientific Reports* 10. doi:[10.1038/s41598-019-56962-2](https://doi.org/10.1038/s41598-019-56962-2).
- Tanaka, H., Katura, T., Sato, H., 2013. Task-related component analysis for functional neuroimaging and application to near-infrared spectroscopy data. *NeuroImage* 64. doi:[10.1016/j.neuroimage.2012.08.044](https://doi.org/10.1016/j.neuroimage.2012.08.044).
- Tanaka, H., Katura, T., Sato, H., 2014. Task-related oxygenation and cerebral blood volume changes estimated from NIRS signals in motor and cognitive tasks. *NeuroImage* 94. doi:[10.1016/j.neuroimage.2014.02.036](https://doi.org/10.1016/j.neuroimage.2014.02.036).
- Tatum, W.O., Benbadis, S.R., Nordli, D.R., Kaplan, P.W., Drislane, F.W., Gaspard, N., Hirsch, L.J., Vaughn, B.V., Underwood, W., Husain, A.M., 2014. *Handbook of EEG Interpretation*. Springer Publishing Company. URL: <https://connect.springerpub.com/content/book/978-1-6170-5180-7>, doi:[10.1891/9781617051807](https://doi.org/10.1891/9781617051807).
- Tausworthe, R.C., 1965. Random Numbers Generated by Linear Recurrence Modulo Two. *Mathematics of Computation* 19. doi:[10.2307/2003345](https://doi.org/10.2307/2003345).
- Thielen, J., Marsman, P., Farquhar, J., Desain, P., 2021. From full calibration to zero training for a code-modulated visual evoked potentials for brain-computer interface. *Journal of Neural Engineering* 18. doi:[10.1088/1741-2552/abecef](https://doi.org/10.1088/1741-2552/abecef).
- Tyagi, A., 2020. TCP/IP Protocol Suite. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* doi:[10.32628/cseit206420](https://doi.org/10.32628/cseit206420).
- Unity Technologies, . Unity essentials pathway. URL <https://learn.unity.com/learn/pathway/aspectos-basicos-de-unity> [Accessed: Sep. 10, 2023].
- Universidad de Valladolid, 2015. Grupo de Ingeniería Biomédica. URL <https://gib.tel.uva.es/> [Accessed: Oct. 20, 2023].
- University of Maryland, School of Medicine, 2015. Synapses & Circuits. URL <https://www.lifesciences.umaryland.edu/neuroscience/research-focus-groups/synapses--circuits/> [Accessed: Apr. 8, 2024].
- Värbu, K., Muhammad, N., Muhammad, Y., 2022. Past, Present, and Future of EEG-Based BCI Applications. doi:[10.3390/s22093331](https://doi.org/10.3390/s22093331).
- Volosyak, I., Cecotti, H., Gräser, A., 2009. Optimal visual stimuli on LCD screens for SSVEP based brain-computer interfaces, in: 2009 4th International IEEE/EMBS Conference on Neural Engineering, NER '09. doi:[10.1109/NER.2009.5109329](https://doi.org/10.1109/NER.2009.5109329).
- Wang, X.J., 2010. Neurophysiological and computational principles of cortical rhythms in cognition. doi:[10.1152/physrev.00035.2008](https://doi.org/10.1152/physrev.00035.2008).

-
- Wei, Q., Liu, Y., Gao, X., Wang, Y., Yang, C., Lu, Z., Gong, H., 2018. A Novel c-VEP BCI Paradigm for Increasing the Number of Stimulus Targets Based on Grouping Modulation with Different Codes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26. doi:[10.1109/TNSRE.2018.2837501](https://doi.org/10.1109/TNSRE.2018.2837501).
- Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M., 2002. Brain-computer interfaces for communication and control. doi:[10.1016/S1388-2457\(02\)00057-3](https://doi.org/10.1016/S1388-2457(02)00057-3).
- Wolpaw, J.R., Wolpaw, E.W., 2012a. *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, New York. doi:[10.1093/acprof:oso/9780195388855.001.0001](https://doi.org/10.1093/acprof:oso/9780195388855.001.0001).
- Wolpaw, J.R., Wolpaw, E.W., 2012b. Brain-Computer Interfaces: Something New under the Sun, in: *Brain-Computer Interfaces: Principles and Practice*. doi:[10.1093/acprof:oso/9780195388855.003.0001](https://doi.org/10.1093/acprof:oso/9780195388855.003.0001).
- Zierler, N., 1959. Linear recurring sequences. *Journal of the Society for Industrial and Applied Mathematics* 7, 31–48. doi:[10.1137/0107003](https://doi.org/10.1137/0107003).
- Šajić, S., Maletić, N., Todorovic, B., Sunjevaric, M., 2013. Random binary sequences in telecommunications. *Journal of Electrical Engineering* 64, 230–237. doi:[10.2478/jee-2013-0033](https://doi.org/10.2478/jee-2013-0033).