



Universidad Nacional de Entre Ríos
Facultad de Ingeniería

Diseño y elaboración de una base de datos pública de registros electroencefalográficos orientados a la clasificación de habla imaginada

Proyecto Final para la obtención
del título de Bioingeniero
de

Germán Alberto Pressel Coretto

Director

Dr. Bioing. Leonardo Rufiner

CoDirector

Bioing. Iván Gareis

Evaluadores

Ms. Bioing. Carlos Pais

Dr. Bioing. César Martínez

Ms. Bioing. Rubén Acevedo

Oro Verde - Entre Ríos - 2016

Agradecimientos

En primer lugar quiero agradecer al director del proyecto, Dr. Bioing. Leonardo Rufiner, y al Co-Director, Bioing. Iván Gareis, por haberme introducido y guiado en el fascinante campo del habla imaginada. A Iván, especialmente, por haberme acompañado durante largas jornadas mientras realizaba los registros. También quiero agradecer al grupo de investigación del LIRINS por haber prestado su tiempo, sus instalaciones, así como los equipos para permitir la realización de esta tesina.

A mis amigos de la facultad que me “prestaron sus cerebros” por cuatro horas para hacer los registros y pusieron todo su esfuerzo para que salieran lo mejor posible. Agradezco también a todos los amigos y compañeros que conocí en la facultad y a mis amigos de la infancia que siempre brindaron sus palabras de aliento a lo largo de la carrera y de la realización de este proyecto.

Finalmente, me resta agradecer a mi familia, que son el eje fundamental de la vida. A ellos les quiero dar las gracias por haberme brindado su apoyo incondicional, por acompañarme a lo largo de esta etapa de mi vida y por siempre creer en mi.

Resumen

Los pacientes en estado de enclaustramiento o en estadios avanzados de enfermedades degenerativas como la esclerosis múltiple, no poseen control sobre la gran mayoría de los músculos, incluidos los que están involucrados con el habla. Con la intención de brindar una nueva vía de comunicación independiente de los nervios periféricos y músculos, se desarrollaron los sistemas de BCI. En el caso de las interfaces basadas en EEG, se utilizan las señales recolectadas a partir de electrodos dispuestos sobre el cuero cabelludo para el control de dispositivos, como sintetizadores de voz; pero debido a diversos factores estos suelen ser relativamente lentos. Un nuevo enfoque que permitiría mejorar la tasa de transmisión de datos es el registro del habla no pronunciada. Sin embargo, no existe en la actualidad una base de datos pública que contenga registros de este tipo y que permita la generación y validación de nuevos algoritmos de procesamiento.

Este proyecto da solución a dicha problemática al generar una base de datos de libre acceso de registros de EEG durante la imaginación o la pronunciación de palabras. Para ello se diseñó un protocolo de adquisición para el registro de la señal de EEG mediante 6 canales (F3, F4, C3, C4, P3 y P4) bajo dos modalidades, habla pronunciada o imaginada. Además se empleó un vocabulario compuesto por dos categorías de palabras: las 5 vocales del español y palabras que indican comando (arriba, abajo, derecha, izquierda, adelante y atrás). Para la implementación del protocolo diseñado en este proyecto se desarrolló una aplicación sobre la plataforma BCI2000 que permite la presentación de las palabras, seguido de una ventana de 4 segundos para la imaginación o la pronunciación de la misma. También esta aplicación posibilita el registro de la señal de audio sincronizada con la de EEG.

El protocolo diseñado se practicó sobre 17 individuos de entre 24 y 28 años de edad, todos voluntarios, sanos e hispanohablantes (argentinos nativos), registrándose 50 repeticiones por palabra, de las cuales 10 correspondían a la modalidad pronunciada

y el resto a la imaginada. La adquisición de la señal de EEG se realizó utilizando un amplificador Grass 8-18-36 conectado a una placa convertora DT9816 de marca DataTranslation; mientras que para el audio se usó el preamplificador MobilePre USB de la marca M-Audio y un micrófono Shure SM58. Las frecuencias de muestreo para cada una de estas señales fueron de 1024 Hz y 44100 Hz, respectivamente.

Para la conformación de la base de datos, se utilizaron solo los registros de 15 individuos, los cuales se inspeccionaron visualmente. Posteriormente, los registros se seccionaron y se filtraron entre 2 Hz y 40 Hz utilizando filtros FIR. Por cada repetición de las palabras, se concatenaron los 6 canales de EEG y se le adicionaron 3 etiquetas correspondientes a la modalidad, al estímulo y a la presencia o no de artefactos. En el caso de la señal de audio, solo se le adicionaron 2 etiquetas, una con el estímulo y otra con el número de registro de EEG al que correspondía. Las repeticiones se almacenaron en archivos con formato MATLAB. La base de datos está constituida por carpetas, una por cada sujeto, que contienen los archivos de la señal de audio y de EEG.

Por último, en este proyecto se evaluó el desempeño de un algoritmo, que emplea para la extracción de características de la señal de EEG a la transformada Wavelet utilizando como función madre las Daubechies de orden 2, 4, 6 y 8. A partir de los coeficientes Wavelet, se calculó la energía relativa Wavelet para cada uno de los niveles de descomposición y para la clasificación se empleó el algoritmo Random Forest (RF) y una Máquina de Vectores de Soporte (SVM). Los resultados obtenidos fueron superiores al azar en varios sujetos para 6 clases, lo que dio indicios de que existe información en la señal de EEG que permite identificar la palabra imaginada.

Índice general

Agradecimientos	I
Resumen	II
Lista de figuras	VI
Lista de tablas	VIII
1. Introducción	1
1.1. Introducción	1
1.2. Problemática	2
1.3. Motivación	2
1.4. Objetivos	3
1.4.1. Objetivo general	3
1.4.2. Objetivos específicos	3
1.5. Estructura del documento	4
2. Electroencefalografía	5
2.1. Introducción	5
2.2. La neurona	5
2.3. Generación del potencial de acción	7
2.4. La corteza cerebral	9
2.5. Áreas corticales asociadas al lenguaje	10
2.6. Electroencefalograma	11
2.7. Ritmos característicos del EEG	12
2.8. Sistema de adquisición de EEG	13
2.9. Artefactos	15
2.10. BCI	16
3. Técnicas de procesamiento de la señal de EEG	18
3.1. Introducción	18
3.2. Referencia promedio común	19
3.3. Filtros FIR	19
3.4. Análisis de componentes independientes	21
3.5. Transformada Wavelet	23
3.6. Random Forest	26
3.7. Máquina de Vectores de Soporte	27

4. Protocolo de adquisición	29
4.1. Introducción	29
4.2. Estado del arte	29
4.3. Protocolo propuesto	32
4.4. Aplicación diseñada para el proceso de adquisición: ImaginedSpeech . .	35
4.5. Resultados de la prueba piloto	39
5. Obtención de la base de datos	41
5.1. Sujetos de prueba	41
5.2. Equipamiento	42
5.3. Procedimiento previo al registro	44
5.4. Registros	47
5.5. Conformación de la base de datos	49
5.6. Estructura y publicación de la base de datos	53
6. Experimentación y resultados	55
6.1. Introducción	55
6.2. Experimentación con la base de datos	55
6.3. Resultados	57
7. Enfoque económico	61
7.1. Introducción	61
7.2. Fuente de financiamiento	61
7.3. Requisitos de PICT 2015	62
7.4. Presentación del proyecto	64
8. Conclusiones y trabajo futuro	71
8.1. Conclusiones	71
8.2. Trabajo futuro	72
A. Consentimiento Informado	73
B. Registros por sujeto	75
C. Aplicación ImaginedSpeech	78
C.1. ImaginedTask.h	79
C.2. ImaginedTask.cpp	83
C.3. ImaginedSpeechTask.h	91
C.4. ImaginedSpeechTask.cpp	93
Bibliografía	109

Índice de figuras

2.1. Esquema de una neurona	6
2.2. Sumación espacial de estímulos	8
2.3. Localización de los lóbulos cerebrales	10
2.4. Sistema internacional 10-20	14
2.5. Esquema de un sistema BCI	17
3.1. Ventanas más usuales para el diseño de filtro FIR por enventaneo	21
3.2. Dilatación de una función Wavelet del tipo Morlet	24
3.3. Esquema de descomposición de la transformada Wavelet discreta	25
3.4. Hiperplano óptimo en un clasificador binario	28
4.1. Esquema de la plataforma BCI2000	36
4.2. Sucesión de bloques en la aplicación ImaginedSpeech	37
4.3. Ventana de configuración de la aplicación ImaginedSpeech	38
5.1. Gráfica de isotonicidad de un sujeto de prueba	42
5.2. Dispositivos para la adquisición de la señal de EEG	43
5.3. Dispositivos para la adquisición de la señal de audio	44
5.4. Lugar de registro	45
5.5. Distribución de los electrodos para el registro	46
5.6. Secuencia de presentación de los estímulos	49
5.7. Interfaz gráfica del toolbox EEGLAB	50
5.8. Respuesta en frecuencia de los filtros FIR implementados	51
5.9. Estructura de la base de datos	54
6.1. Resultados obtenidos para tres sujetos	59

7.1. Porcentaje por rubro sobre el total del subsidio solicitado	70
--	----

Índice de cuadros

2.1. Bandas de frecuencia de los ritmos característicos del EEG	12
4.1. Diccionarios propuestos en el protocolo	33
5.1. Configuración de intervalos para ImaginedSpeech	48
5.2. Valor de la etiqueta estímulo y la palabra del diccionario correspondiente	52
6.1. Bandas de frecuencia por nivel de descomposición	57
6.2. Diferencia aritmética entre los porcentajes de acierto obtenidos sin CAR y con CAR.	58
6.3. Porcentaje de aciertos para cada sujeto de la base de datos	60
7.1. Montos máximos por año y total para los proyectos de categorías I, II y III	63
7.2. Montos máximos por año y total para los proyectos de categorías I, II y III que solicitan beca	63
7.3. Porcentajes máximos por rubro	64
7.4. Costos de insumos	66
7.5. Costos de bibliografía	66
7.6. Costos de publicación de resultados	67
7.7. Costos de viajes e inscripciones	67
7.8. Costos de equipamiento	68
7.9. Costos de mano de obra	69
7.10. Costos totales por rubro	69
B.1. Información personal de los sujetos de prueba	75
B.2. Cantidad de registros por sujeto y modalidad para cada una de las vocales	76

B.3. Cantidad de registros por sujeto y modalidad para cada una de las direcciones	77
--	----

Capítulo 1

Introducción

1.1. Introducción

La forma de comunicación típica entre individuos incluye información visual y auditiva proveniente de la vocalización del habla junto con la producción de gestos. En el caso de los pacientes que padecen discapacidades físicas severas, como el síndrome de locked-in, esta forma de comunicación se ve imposibilitada debido a la carencia de control sobre la mayoría, o la totalidad, de la musculatura voluntaria del cuerpo. En consecuencia estos pacientes, que poseen su capacidad cognitiva intacta, se encuentran aislados de su entorno.

Con la finalidad de dar solución a esta problemática se desarrollaron las interfaces cerebro-computadora (BCI) que proveen al cerebro de un nuevo canal de comunicación, independiente de los nervios periféricos y los músculos. A pesar de que existen diversas técnicas, en la mayoría de los sistemas BCI, el control se realiza a partir de señales electroencefalográficas (EEG). Estos sistemas procesan los potenciales adquiridos por electrodos sobre el cuero cabelludo, traduciéndolos a señales de control que posibiliten el comando de dispositivos como sintetizadores de voz.

Si bien grandes avances se han realizado en este área, los sistemas son aún relativamente lentos, ya que permiten solo la producción de una palabra por minuto [1]. El otro problema que presentan es que el tiempo de entrenamiento, para que el usuario logre controlar correctamente el dispositivo de salida, es prolongado debido al uso de características de la señal de EEG que se producen de manera poco cociente. Investigaciones recientes tratan de sortear estos problemas utilizando como medio de control el habla imaginada, entendiéndose por ésta a la imaginación de la pronunciación

de las palabras sin emitir sonidos o articular movimientos musculares.

1.2. Problemática

Para la implementación de sistemas BCI que permitan una comunicación más rápida, se requiere poder discriminar las palabras a partir de cambios o características particulares de la señal EEG durante el habla imaginada. Si bien lograr este objetivo presenta grandes dificultades en cuanto a encontrar los parámetros que caracterizan a la señal, uno de los problemas principales es la falta de registros de electroencefalograma adquiridos mientras sujetos de prueba imaginan distintos vocabularios bajo un protocolo preestablecido. La inexistencia de bases de datos de acceso público de este tipo de registros imposibilita, o al menos dificulta, la generación de nuevos algoritmos de clasificación o la mejora de los ya existentes.

1.3. Motivación

El desarrollo de una interfaz BCI que sea capaz de traducir las señales de EEG durante la imaginación del habla permitiría mejorar la calidad de vida de aquellos pacientes que padecen severas discapacidades físicas. Mediante estos sistemas podrían relacionarse con su entorno y comunicarse con sus familiares, así como también indicar sus necesidades a los agentes de salud posibilitando un enfoque más preciso en el tratamiento de este tipo de patología.

Para lograr una interfaz de estas características es necesario generar algoritmos robustos y precisos tanto en las etapas de preprocesamiento, extracción de parámetros de interés y clasificación. Para ello, la utilización de bases de datos de registros realizados sobre sujetos sanos brinda una primera aproximación hacia este objetivo. Además, el empleo de grandes bases de datos permite lograr una mayor generalización de estos sistemas, reduciendo los tiempos dedicados al ajuste de las interfaces al usuario final. Asimismo, dado que el habla imaginada se corresponde con una respuesta espontánea en la actividad cerebral, la tasa de transmisión de datos sería mayor a la de otros sistemas BCI que utilizan potenciales evocados.

En todas las investigaciones recientes sobre habla imaginada, como paso previo

a la prueba de un algoritmo de procesamiento, era necesaria la obtención de registros de EEG bajo un protocolo determinado. La adquisición de estos registros requiere de equipamiento específico, de un grupo de voluntarios y una gran cantidad de tiempo para la recolección de los datos y su acondicionamiento. Es por ello, que la disponibilidad de una base de datos pública de registros de EEG durante habla imaginada facilitaría las investigaciones tendientes al desarrollo de nuevos algoritmos de procesamiento, así como también permitiría la comparación de los resultados obtenidos a partir de ellos.

1.4. Objetivos

1.4.1. Objetivo general

Obtención de una base de datos de libre acceso de registros de electroencefalograma adquiridos sobre sujetos sanos orientados a la detección del habla imaginada.

1.4.2. Objetivos específicos

- Investigar el estado del arte respecto al estudio del habla imaginada.
- Diseñar y definir un protocolo de ensayo para la adquisición de las señales de EEG.
- Conectar, configurar y programar el sistema de adquisición a nivel de software y hardware.
- Obtener los registros de EEG durante el habla no pronunciada.
- Conformar una base de datos con los registros de EEG procesados y segmentados.
- Evaluar el desempeño de la base de datos mediante un algoritmo desarrollado e implementado por otro autor.
- Publicar la base de datos.

1.5. Estructura del documento

Además de este capítulo introductorio, el documento esta conformado por 7 capítulos. El Capítulo 2 brinda un breve marco teórico donde se describen las áreas de la corteza cerebral involucradas en la producción del habla así como los principales procesos biológicos que generan los potenciales medidos mediante electroencefalografía y las características más relevantes de esta técnica. Así mismo, en dicho capítulo se detallan los componentes de un sistema BCI.

El Capítulo 3 continúa con el marco teórico y en él se explican los diferentes métodos de preprocesamiento, extracción de características y clasificación utilizados en la realización de esta tesina. En el Capítulo 4 se presenta una versión resumida de las investigaciones que conforman el estado del arte en el estudio del habla imaginada, así como también el protocolo de ensayo diseñado junto con los resultados de la prueba piloto realizada.

El Capítulo 5 detalla las condiciones bajo las cuales se realizaron los registros, la selección de los sujetos de prueba y el equipamiento utilizado. La conformación de la base de datos, su estructura y formas de publicación también se describen. En el Capítulo 6 se presenta el experimento realizado sobre la base de datos y los resultados obtenidos.

Un análisis económico para la búsqueda de financiamiento orientado a futuras investigaciones se reporta en el Capítulo 7. Mientras que, las conclusiones y proyecciones a futuro de este trabajo se puntualizan en el Capítulo 8.

Capítulo 2

Electroencefalografía

2.1. Introducción

El cerebro es la estructura de mayor complejidad del cuerpo humano y la comprensión acerca de su funcionamiento es solo parcial, a pesar de los avances que se han hecho en esta área. Una de las señales de las cuales se puede inferir nueva información acerca de cómo el cerebro realiza el procesamiento de distintos estímulos es el electroencefalograma. Esta palabra fue acuñada por el científico alemán Hans Berger en la década de 1920, quien realizó el primer registro de señales eléctricas cerebrales mediante la colocación de electrodos sobre el cuero cabelludo de un humano. Desde ese momento la técnica se ha ido perfeccionando, así como también se han mejorado los sistemas para la adquisición y almacenamiento de las señales.

En este capítulo se desarrollarán los conceptos y bases fisiológicas detrás de la generación de los potenciales eléctricos registrados mediante electroencefalografía, así como también las características actuales de la técnica y una de sus aplicaciones, las interfaces cerebro computadora.

2.2. La neurona

La neurona es la unidad funcional del sistema nervioso y es un tipo de célula excitable y secretora que permite la generación de impulsos eléctricos y su comunicación mediante la liberación de neurotransmisores. En el SNC, se encuentran dentro de una estructura de soporte brindada por las células gliales, quienes aseguran la irrigación y cumplen un papel importante en la transmisión de información. Según su

funcionalidad, las neuronas se pueden clasificar en neuronas sensoriales, que se encuentran unidas a receptores sensitivos; neuronas motoras, que hacen sinapsis sobre la placa motora de los músculos o sobre glándulas secretoras; e interneuronas, las cuales transmiten información de una neurona a otra [2,3].

A pesar de que los distintos tipos de neuronas presentan morfologías distintas, en todas ellas se pueden distinguir tres regiones con funciones diferentes: el cuerpo o soma, las dendritas y el axón. La Figura 2.1 muestra las secciones principales de la neurona. El soma contiene las organelas y elementos citoplasmáticos, tales como el núcleo, el aparato de Golgi, mitocondrias, ribosomas y lisosomas, encargados de suplir las necesidades metabólicas y sintéticas de toda la neurona, ya que no se producen proteínas en el axón y solo muy pocas en las dendritas. Las dendritas son prolongaciones del citoplasma neuronal, que están especializadas en la recepción de señales químicas transmitidas por los axones de otras neuronas. En promedio, una neurona tiene entre 2 y 9 dendritas principales, las cuales se ramifican formando lo que se conoce como árbol dendrítico y que tiene como objetivo aumentar la superficie para la unión sináptica con otras neuronas.

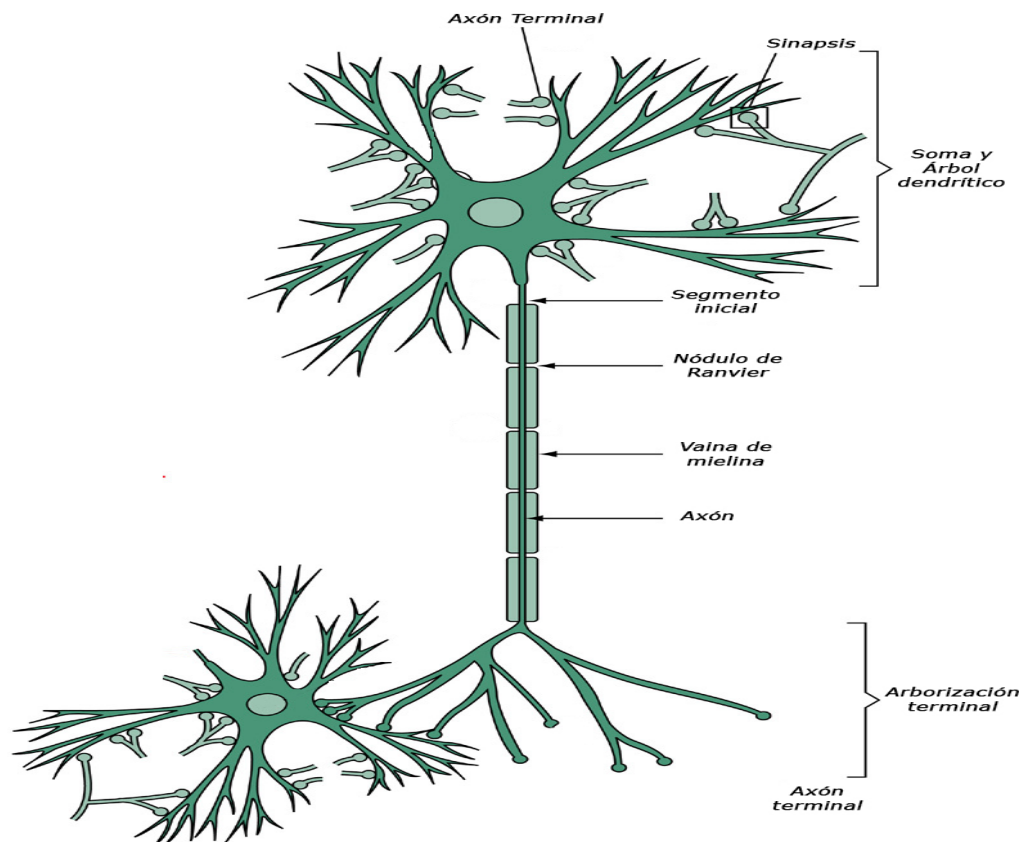


Figura 2.1: Esquema de una neurona. Modificado de [4]

Mientras que las dendritas se encargan de transmitir los impulsos hacia el interior de la neurona, los axones lo hacen en sentido inverso, es decir, transmiten información desde el soma hacia otra neurona u órgano efector. Cada neurona posee un axón, que se origina en una región especializada del soma en forma de embudo, denominada cono axónico. A partir de este punto, el axón se puede extender solo unos pocos milímetros, en el caso de las interneuronas, o alcanzar longitudes cercanas a un metro, como en algunos nervios del SNP. Ciertos axones se encuentran recubiertos por una espiral de membranas, las vainas de mielina, que tienen como función aumentar la velocidad de conducción de los potenciales de acción. Estas vainas son producidas en el SNP por las células de Schwann y por los oligodendrocitos en el SNC [4].

2.3. Generación del potencial de acción

Al igual que el resto de las células, las neuronas están rodeadas por una membrana semipermeable que permite la difusión de ciertos iones a través de ella e inhibe la de otros. En el interior de la neurona, existe una baja concentración iones Na^+ debido a que la membrana admite la salida pasiva de K^+ pero restringe el ingreso de Na^+ . Esta separación de cargas entre el interior celular y el líquido tisular genera un potencial de reposo de alrededor de -60 mV. Frente al ingreso de cationes Na^+ , la membrana aumenta su potencial, es decir, se despolariza. Mientras que si se produce una apertura de canales K^+ , el egreso de estos iones provoca que el potencial disminuya por debajo del de reposo, lo que se denomina hiperpolarización. Los cambios en la permeabilidad de la membrana, posibilitan la generación y transmisión de impulsos eléctricos [5,6].

Cuando una neurona es estimulada, a nivel de sus dendritas o directamente en el soma, el potencial generado es local y se transmite a través del cuerpo neuronal hacia el cono axónico. Allí se suman, espacial y temporalmente, los distintos estímulos locales que ha recibido la neurona y si se genera una despolarización superior al nivel umbral se produce un potencial de acción. Esto se puede apreciar en la Figura 2.2. Los potenciales de acción son respuestas estereotipadas del tipo “todo o nada”, cuya duración y amplitud son idénticas para cada tipo de neurona [5,7].

Al superar el potencial transmembrana el umbral, se desencadena la apertura de los canales de Na^+ controlados por voltaje, permitiendo el ingreso de una corriente

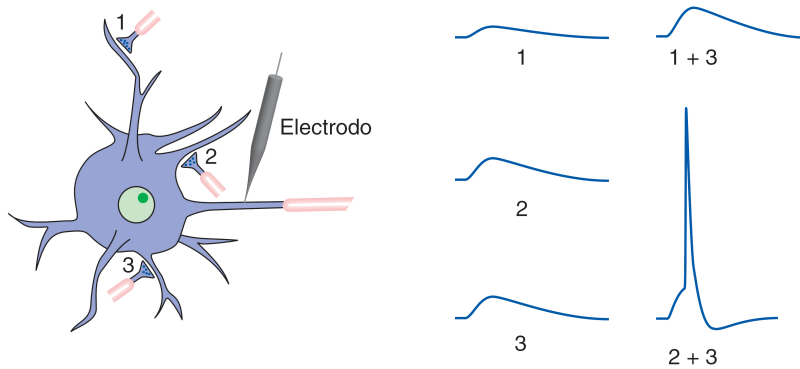


Figura 2.2: Sumación espacial de estímulos: Solo cuando se estimula la neurona con 2 y 3, se supera el umbral de excitación y se desencadena un potencial de acción. Modificado de [8]

de cationes hacia el interior del segmento proximal del axón. Este movimiento de cargas modifica el potencial de membrana hasta un valor positivo de 40mV, amplitud pico del potencial de acción. En ese momento, se abren los canales de K^+ , que tienden a restaurar el balance de cargas a su condición de reposo.

Después de producirse un potencial de acción, se sucede un periodo refractario, dividido en absoluto y relativo. El absoluto inicia al alcanzar el pico del potencial de acción y en él, sin importar la intensidad del estímulo, no se suscitará ningún otro potencial. Esto es debido a que los canales de Na^+ se encuentran en estado inactivo. A medida que los canales retornan a su estado de reposo, el valor de despolarización requerido para generar un nuevo potencial de acción va disminuyendo. Este es el periodo refractario relativo y finaliza al regresar el nivel umbral hasta los -45mV. [8].

Los potenciales de acción producidos en la porción proximal del axón, se transmiten a lo largo de este hasta alcanzar sus ramas terminales. En este proceso, la corriente de iones creada por la despolarización inicial es propagada a una sección adyacente de la membrana y así sucesivamente. La velocidad de conducción depende del diámetro del axón y de la presencia de mielina. La última permite que el potencial de acción sea transmitido electrotónicamente en las vainas, mientras que genera un nuevo potencial de acción cuando alcanza un nodo de Ranvier, regenerando el estímulo en cada nodo y posibilitando su conducción a mayores distancias.

Cuando el estímulo alcanza el botón terminal del axón, los canales de Ca^{+2} controlados por voltaje se activan y desencadenan la liberación de vesículas de neuro-

transmisores hacia el espacio de unión entre las células, denominado sinapsis. En esta juntura, los neurotransmisores difunden pasivamente y se unen a neuroreceptores en la membrana de la célula postsináptica. La unión del neurotransmisor puede generar un potencial local excitatorio, en el caso de que la membrana se despolarize, o inhibitorio, si provoca una hiperpolarización [4, 5].

2.4. La corteza cerebral

El sistema nervioso humano se divide en dos partes principales, el sistema nervioso central (SNC) y el sistema nervioso periférico (SNP). El primero está conformado por el encéfalo y la medula espinal, mientras que el segundo por los nervios craneales y periféricos, y los ganglios asociados. El SNC es el encargado de la correlación e integración de la información nerviosa, aportada a través de las vías sensitivas, y la coordinación de los impulsos eferentes hacia los órganos efectores. La transmisión de dichos impulsos, hacia y desde el sistema nervioso central, es función del SNP [7, 8].

El cerebro es el órgano más voluminoso del encéfalo y se encuentra alojado en la cavidad craneana. La fisura longitudinal del cerebro lo divide en dos hemisferios casi simétricos, uno derecho y otro izquierdo; cada uno formado por una corteza de sustancia gris sobre una masa de sustancia blanca. La sustancia gris está constituida por el soma y los árboles dendríticos de millones de neuronas, mientras que la sustancia blanca es rica en axones mielinizados. A nivel de la corteza es donde se llevan a cabo las funciones mentales superiores, tales como la percepción, el control de los movimientos, el pensamiento abstracto, la memoria y el lenguaje. Entre los hemisferios y por debajo de la fisura, se encuentra el cuerpo caloso, una estructura de fibras nerviosas que comunica e integra la información proveniente de ambos hemisferios [9, 10].

La apariencia arrugada de la superficie del cerebro, caracterizada por surcos y convoluciones, le permite aumentar el área destinada a la corteza cerebral. Estas cisuras, a su vez, delimitan cuatro lóbulos en cada hemisferio: frontal, parietal, temporal y occipital; siendo los nombres correspondientes con el hueso del cráneo debajo del cual se encuentran. En la Figura 2.3 se puede apreciar la localización de cada lóbulo y los surcos que los limitan.

El lóbulo frontal contiene la corteza motora primaria, encargada de controlar

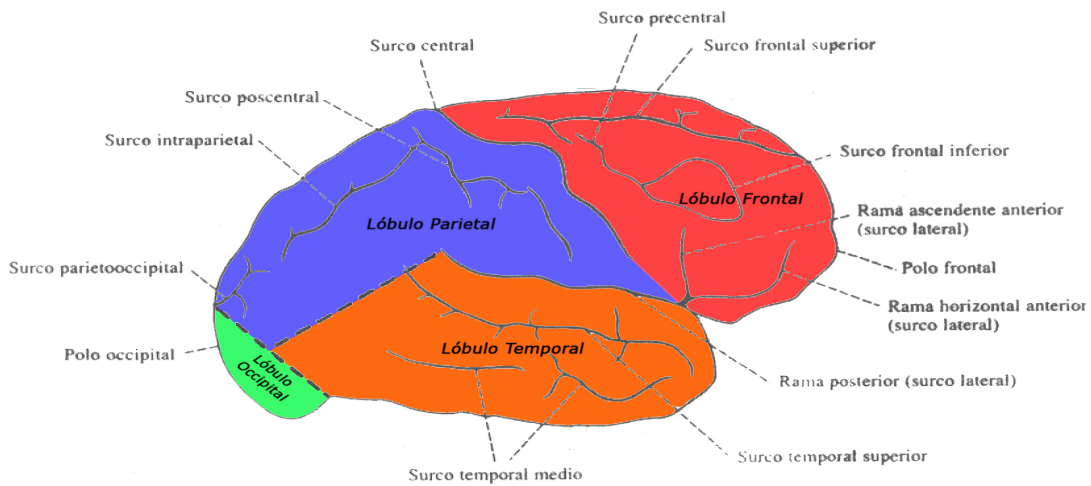


Figura 2.3: Localización de los lóbulos cerebrales y de los surcos principales. Modificado de [7]

los movimientos del lado opuesto del cuerpo; la corteza prefrontal, asociada a la personalidad y la inteligencia; y el área de Broca, relacionada con la producción del lenguaje. La información táctil y propioceptiva proveniente del lado contralateral del cuerpo, es procesado en la corteza somatosensorial primaria del lóbulo parietal; mientras que en el lóbulo temporal se encuentra la corteza auditiva primaria; el área de Wernicke, asociada a la comprensión del lenguaje; y una región dedicada al procesamiento de la información visual. El lóbulo occipital, en cambio, es responsable del procesamiento de los estímulos visuales, su asociación y almacenamiento [6].

2.5. Áreas corticales asociadas al lenguaje

La producción del habla es un proceso complejo y de múltiples etapas, que convierte una idea en una señal acústica, capaz de ser comprendida por otros individuos. En este proceso se destacan tres regiones corticales: el área de Broca, el área de Wernicke y la corteza motora primaria. El área de Broca se encuentra en la parte posterior del giro frontal inferior y limita posteriormente con la corteza motora primaria. Se considera que esta región está involucrada en varios aspectos del lenguaje, tales como el procesamiento semántico, fonológico y en la articulación de palabras [11, 12]. En cambio, el área de Wernicke se encarga del análisis y comprensión de los estímulos auditivos y visuales relacionados con el habla [13]. Según Indefrey [14], también está asociado al almacenamiento de las codificaciones fonológicas de las palabras. La corte-

za motora primaria, fundamentalmente el tercio inferior, coordina el posicionamiento y los movimientos de la lengua, los labios, la mandíbula y las cuerdas vocales, para traducir las ideas en sonidos.

Si bien se desconocen las vías fisiológicas reales, existe una amplia variedad de modelos que teorizan el flujo de información a través de las distintas áreas del cerebro. Uno de los primeros modelos fue el de Geschwind – Wernicke, presentado por Geschwind en 1967, que propone un sistema serial lateralizado en el hemisferio izquierdo del cerebro [15]. El proceso se inicia con la recepción de un estímulo auditivo o visual, a nivel de la corteza auditiva primaria o de la corteza visual primaria, respectivamente. En el área de Wernicke se interpreta y se le asocia un significado, transmitiendo la información hacia el área de Broca mediante el fascículo arqueado. La respuesta motora necesaria para articular la palabra se define en el área de Broca y se envía hacia la corteza motora primaria, encargada de la producción del habla [16].

Un modelo más reciente es el propuesto por Hickok y Poeppel [17], en el cual el procesamiento de un estímulo auditivo, se lleva a cabo por dos vías, una ventral y otra dorsal. La ventral se asocia al reconocimiento del sonido y se proyecta a estructuras del lóbulo temporal; mientras que la vía dorsal, cuyo flujo de información se dirige a los lóbulos frontales y parietales, es la interfaz entre el estímulo auditivo y la articulación del habla. Además, en este modelo se propone una mayor actividad bilateral que en el de Gestwind.

2.6. Electroencefalograma

El primer registro de actividad espontánea cerebral fue realizado en 1875 por Richard Canton, quien mediante un galvanómetro pudo observar impulsos eléctricos medidos sobre la superficie del cerebro de un animal. En 1924, casi cincuenta años más tarde, Hans Berger registra el primer electroencefalograma de un humano, mediante el uso de tiras metálicas pegadas sobre el cuero cabelludo y utilizando, al igual que Canton, un galvanómetro como elemento de medida. Durante dicho procedimiento pudo medir el patrón irregular y de poca amplitud de la señal de EEG, generando las bases para el desarrollo de la electroencefalografía [18].

La electroencefalografía consiste en el registro de la actividad eléctrica con-

junta de un grupo de neuronas, principalmente, de la corteza cerebral [19]. Esta puede ser registrada mediante electrodos profundos, denominándose electrograma; sobre la corteza cerebral o electrocorticograma (ECoG); o desde el cuero cabelludo (EEG). Tanto en el ámbito clínico como científico, el método que emplea electrodos sobre el cuero cabelludo es el más utilizado, debido a que no es invasivo, es de fácil colocación y el instrumental de adquisición es de bajo costo [20].

Debido a la atenuación producida por las capas de tejido, hueso y piel, el campo eléctrico producido por una única neurona cortical no puede ser registrado. Sin embargo, la despolarización sincronizada de un conjunto de neuronas produce un campo eléctrico lo suficientemente fuerte como para ser medido desde la superficie. La orientación perpendicular a la corteza de las neuronas piramidales propicia que los potenciales post-sinápticos, producidos en la membrana de las dendritas, o del soma, sean los que más contribuyen a la señal de EEG [19]. Además, las características temporales y espaciales de los potenciales registrados dependen no solo de la localización de la fuente, sino también, de la geometría del cerebro y de las propiedades eléctricas de los tejidos que los conducen [21].

2.7. Ritmos característicos del EEG

En general, las señales de EEG presentan amplitudes que varían desde $1\mu\text{V}$ hasta cerca de $100\mu\text{V}$ y abarcan un rango de frecuencias de 0.5Hz a 40Hz [2]. Del análisis espectral del EEG es posible distinguir 5 bandas dominantes, identificadas con las letras griegas α , β , δ , θ y γ , cuyos límites frecuenciales se detallan en el Cuadro 2.1.

Cuadro 2.1: Bandas de frecuencia de los ritmos característicos del EEG

Ondas	Frecuencias
Delta δ	1 - 4Hz
Tita θ	4 - 8Hz
Alfa α	8 - 13Hz
Beta β	13 - 30Hz
Gamma γ	> a 30Hz

Los ritmos delta (δ) caracterizan el EEG durante las etapas de sueño profundo y son ondas lentas de entre 1Hz y 4Hz. Su amplitud aumenta en sujetos privados del sueño y marcadamente en niños. Sin embargo, en un adulto despierto, la presencia

de estas ondas solo es asociada con cuadros patológicos [22]. Las ondas tita (θ), en cambio, son predominantes durante estados de somnolencia y en las primeras etapas del sueño; aunque también se han registrado durante tareas que involucran la memoria funcional. Las frecuencias comprendidas por este ritmo van de 4Hz a 8Hz y la potencia de esta banda se encuentra inversamente relacionada con las del ritmo alfa [23].

En un sujeto despierto y relajado, las ondas alfa (α) se registran principalmente en las áreas occipitales y se las asocia como un ritmo característico del estado inactivo [24]. Abarcan frecuencias de 8Hz a 13Hz y su actividad puede ser bloqueada por la apertura y movimiento de los ojos, al evocar una imagen o realizar una actividad mental. En el mismo rango que las ondas alfa, se ha descrito otro ritmo relacionado a la activación del área somatosensorial en el lóbulo parietal del cerebro, llamado ritmo μ . Este exhibe una morfología más puntiaguda que las ondas alfa y está relacionado con la inmovilización de los músculos esqueléticos [25].

Las ondas beta (β) son ondas más rápidas, de entre 13Hz y 30Hz, dominantes en el EEG de un adulto despierto y con los ojos abiertos. La amplitud de estas ondas es mayor durante contracciones sostenidas, disminuye al imaginar movimientos y es prominente durante la anticipación a un evento [26, 27]. La actividad mayor a 30Hz se denomina ritmo gamma (γ) y está relacionada con la atención, la respuesta a ciertos estímulos sensoriales y procesos que involucran a la memoria a corto plazo [28].

2.8. Sistema de adquisición de EEG

Las señales de EEG, como ya se mencionó, poseen amplitudes muy pequeñas, del orden de los microvoltios, y para ser registradas se requiere un sistema que tenga elevada amplificación, un adecuado ancho de banda, inmunidad al ruido y alta impedancia de entrada. Los sistemas para el registro de EEG consisten en electrodos, un amplificador, filtros analógicos, un conversor analógico-digital y un dispositivo para el almacenamiento de las señales adquiridas [18].

Los electrodos son elementos conductores, que convierten las corrientes iónicas en corrientes eléctricas, conectados por cables al amplificador. Existen de diversas formas y tipos, pero los más usuales son los electrodos superficiales. Dentro de los superficiales, encontramos a los electrodos de contacto, que son placas delgadas de

metal en forma de discos planos o de copa [29]. Generalmente, están hechos de plata u oro y tienen la ventaja de no ser invasivos y ser de fácil colocación.

En la práctica clínica, se estableció el sistema internacional 10-20, estándar basado en puntos de referencia anatómicos, para el posicionamiento de electrodos [30]. El sistema emplea las distancias de nasión a inion y de pretrago derecho a pretrago izquierdo como base; colocándose el primer electrodo al 10 %, y los restantes al 20 % de la longitud de referencia. Dicha distribución de los electrodos se puede ver en la Figura 2.4. Además existen dos tipos de montaje para el registro de EEG, unipolar y bipolar. En el denominado unipolar, se registra la diferencia entre cada uno de los electrodos y un potencial de referencia; siendo la apófisis mastoides, el lóbulo de la oreja, la nariz y el vertex algunas de las posiciones elegidas [21]. El montaje bipolar, en cambio, mide la diferencia de potencial registrada entre un electrodo y otro vecino [20].

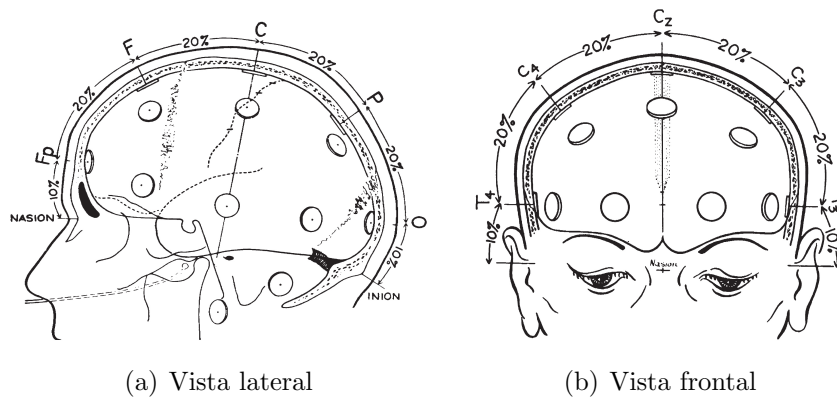


Figura 2.4: Posicionamiento de los electrodos según el sistema internacional 10-20. Extraído de [30]

La amplificación de la señal de EEG se lleva a cabo, normalmente, en dos etapas, debido a la alta ganancia requerida para que alcance los rangos del convertor analógico- digital (A/D). La señal proveniente de los electrodos pasa por un amplificador de instrumentación con una relación de rechazo del modo común elevada, mejorando la relación señal-ruido y con una ganancia variable; y luego a través de un amplificador de aislación para que el circuito al que se conecta el paciente este aislado de la red eléctrica [31]. Con el fin de que el ruido no llegue a saturar la segunda etapa de amplificación, la señal es filtrada mediante un filtro pasabanda o una combinación de filtros pasabajo y pasaalto.

Finalmente, se amplifica hasta cubrir el rango de entrada del convertor A/D, se digitaliza y almacena. Las características que definen la exactitud del convertor

A/D al representar digitalmente la señal analógica son la frecuencia de muestreo, la cantidad de bits y el rango de voltaje de entrada; en tanto que el almacenamiento, por lo general, se realiza mediante una computadora.

2.9. Artefactos

Los artefactos son potenciales indeseados que contaminan la señal de EEG, provocando cambios en su morfología. Su correcta detección y la implementación de técnicas para reducirlos son cruciales al realizar registros de electroencefalograma. Dependiendo de su origen, los artefactos se pueden clasificar en fisiológicos y no fisiológicos o técnicos [32]. Dentro de la primera categoría encontramos a los movimientos oculares, el parpadeo y los producidos por la actividad muscular. Mientras que el ruido de línea y la desconexión momentánea de los electrodos son los artefactos técnicos más frecuentes [18].

Los movimientos oculares son la principal fuente de ruido en el EEG y se deben a la actividad eléctrica de la retina (EOG). El ojo se puede modelizar como un dipolo eléctrico, estando la dirección del dipolo alineada con la línea de visión y el tamaño del mismo acorde a la cantidad de luz que impacta sobre la retina. Cuando se realizan movimientos sacádicos, ya sean voluntarios o involuntarios, se produce un cambio en el potencial entre la córnea y la retina, cuya amplitud es proporcional al desplazamiento angular de la línea de visión [33]. El parpadear provoca una modificación abrupta del potencial de la retina, por lo que la señal producida es de mayor amplitud que en el caso del movimiento de ojos [34]. Ambos tipos de artefactos oculares contaminan principalmente los canales posicionados sobre el lóbulo frontal.

La actividad mioeléctrica (EMG) generada por la contracción de un grupo de músculos, provoca grandes perturbaciones en el electroencefalograma [32]. Se pueden observar al tragar, masticar, hablar o mover la cabeza y manifiestan una respuesta menos estereotipada que el resto de los artefactos fisiológicos, lo que dificulta su remoción. Además el espectro de la señal de EMG muestra contenidos frecuenciales que se solapan con las ondas beta y alfa del EEG [35].

Una de las fuentes de ruido no fisiológico son los cables que conectan los electrodos con el amplificador, ya que, si no presentan aislación suficiente, son susceptibles

a interferencias electromagnéticas producidas por las corrientes que circulan por las líneas eléctricas. Este tipo de artefacto contamina la EEG con señales sinusoidales de igual frecuencia a la utilizada en el suministro eléctrico; siendo en Argentina de 50 Hz. Otro ruido técnico es el generado por el movimiento de los electrodos, que provoca cambios en el potencial de contacto de la interfaz piel- electrodo. Usualmente, se manifiesta como un pico abrupto seguido del regreso progresivo hacia la línea base [36].

2.10. BCI

Una interfaz cerebro-computadora (Brain Computer Interface, BCI) es un método de comunicación basado en el registro de la actividad eléctrica del cerebro para la generación de una nueva vía de transmisión de información, distinta de las vías eferentes nerviosas y los músculos [28]. El objetivo de estos sistemas es proveer de una interfaz que permita, a individuos con disfunciones motoras severas, controlar de manera efectiva dispositivos como computadoras, sintetizadores de voz, prótesis y sillas de ruedas. Las BCI han generado especial interés en el ámbito clínico, como medio alternativo de comunicación para pacientes que padecen cuadros agudos, entre los que se pueden nombrar infarto del tallo cerebral, arteritis basilar infecciosa y tumor de bulbo raquídeo; o severos casos de esclerosis lateral amiotrófica y esclerosis múltiple [37].

Las BCI traducen los impulsos eléctricos medidos desde el cuero cabelludo en señales de control, sin la necesidad de realizar movimientos voluntarios y basándose en la actividad espontánea o evocada del cerebro. Cuando son tareas evocadas o exógenas, el usuario enfoca su atención en estímulos que generan una respuesta autónoma en forma de patrones específicos. Mientras que en el registro de la actividad espontánea, el usuario realiza una tarea mental que provoca cambios en la señal de EEG, pudiendo ser detectados mediante la interfaz. Estas no requieren de estímulos y el usuario puede entrenarse para mejorar la respuesta mediante retroalimentación [38].

Como cualquier sistema de control, un sistema BCI tiene una señal de entrada, un elemento que produce una señal de salida a partir de la señal de entrada, una señal de salida y un protocolo que determina los tiempos de operación [1]. El esquema de la Figura 2.5 muestra el desglose de un sistema BCI.

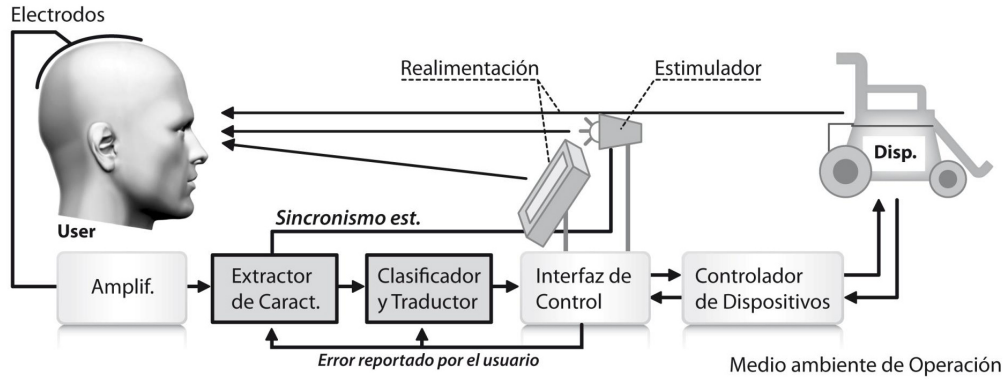


Figura 2.5: Esquema de un sistema BCI. Extraído de [39]

El sistema de adquisición, como ya se explicó en la sección anterior se encarga de adquirir la señal de EEG, filtrarla y digitalizarla. Junto con la señal de interés, también se registrara ruido proveniente de la actividad eléctrica de otras partes del cerebro o de fuentes externas. Es por ello, que la primera etapa consiste en el preprocesamiento del EEG, para eliminar los artefactos y ruidos que acompañan a la señal, mediante métodos espaciales y temporales [40]. Posteriormente, se lleva a cabo la extracción de características, cuyo objetivo es encontrar las variables que son similares a todos los registros de la misma clase, pero distintos en las otras categorías. El vector de características debe tener la menor dimensión posible para no aumentar la complejidad del sistema; pero siempre teniendo en cuenta que, con la cantidad de información proporcionada, sea posible clasificar la señal [1].

La etapa de clasificación emplea algoritmos para la conversión del vector de características en señales lógicas de control, seleccionadas dentro de un número determinado de estados posibles. La correcta clasificación de la señal depende de las características empleadas, de la efectividad de los algoritmos y de la presencia de ruidos o artefactos en la señal de entrada que no fueron reducidos o eliminados en las etapas previas [38].

Por último, las señales lógicas son traducidas en el sistema de control a señales adecuadas para comandar algún dispositivo de salida, como puede ser una silla de ruedas, un sintetizador de voz, un brazo robótico, entre otros. Sin embargo, la mayoría de las BCI actuales tienen como dispositivo de salida el monitor de una computadora, permitiendo mediante retroalimentación mantener y mejorar la exactitud y velocidad de la comunicación [41].

Capítulo 3

Técnicas de procesamiento de la señal de EEG

3.1. Introducción

Las interfaces cerebro-computadora son sistemas constituidos por una secuencia de etapas que comienza con la adquisición de la señal de EEG mediante electrodos en el cuero cabelludo y se continua con el preprocesamiento, la extracción de características y la clasificación.

Las señales son preprocesadas para reducir el contenido de ruido y mitigar artefactos presentes en ella. Algunos de los métodos existentes, tales como filtrado, referencia promedio común y análisis de componentes independientes se describen en este capítulo. Además, se brinda una breve explicación del análisis multiresolución implementado mediante la transformada Wavelet, que permite extraer características del EEG en el plano temporal y frecuencial. En cuanto a los clasificadores, se hace hincapié en las máquinas de vectores de soporte y en el algoritmo Random Forest.

La correcta combinación de técnicas efectivas para acondicionar la señal y liberarla de artefactos; de estrategias para extraer aquellas características que la definen y de algoritmos que sean eficientes en la separación de las clases, permitirá obtener sistemas BCI más robustos, con gran capacidad de generalización y cuya tasa de detección sea óptima.

3.2. Referencia promedio común

El registro de EEG unipolar mide la diferencia de voltaje entre un electrodo activo y el electrodo de referencia, siendo uno de los mayores inconvenientes encontrar una región de la cabeza donde la actividad eléctrica sea constante en función del tiempo. Aún aquellos sitios presumiblemente inactivos presentan variaciones en su potencial, no solo debido a fuentes localizadas en el lóbulo cerebral adyacente sino que también producto de fuentes distantes transmitidas por el conductor de volumen [42].

La referencia promedio común (Common Average Reference, CAR) ha sido propuesta como un método que provee de una referencia inactiva. En cada instante de la señal se sustrae, del canal de interés, el valor promedio de todos los electrodos del montaje [43]. Esto se puede expresar matemáticamente como

$$V_i[n] = V_i[n] - \frac{1}{N} \sum_{j=1}^N V_j[n] \quad (3.1)$$

donde V_i es el i -ésimo electrodo y N es la cantidad total de electrodos activos. Una distribución espacial de voltaje con media cero se obtiene si toda la cabeza se encuentra cubierta por electrodos equidistantemente espaciados y si los potenciales son generados por fuentes puntuales. A pesar de que en la práctica, esta condición normalmente no se cumple, igual se considera que el registro de EEG obtenido es independiente de la referencia. El método actúa como un filtro pasaalto espacial, reduciendo los componentes comunes a la gran mayoría de los electrodos [44].

3.3. Filtros FIR

La señal de EEG, como se ya se expresó en el capítulo anterior, se encuentra contaminada por ruido y artefactos. Un método para mitigar el ruido es mediante filtros digitales, entendiéndose por filtro digital a un sistema de tiempo discreto que aplica una transformación a la señal de entrada $X[n]$, generando una secuencia de salida $Y[n]$. Una forma clásica de clasificar filtros es en base a las características de su respuesta al impulso, en filtros de respuesta finita al impulso (Finite Impulse Response, FIR) y de respuesta infinita al impulso (Infinite Impulse Response, IIR).

Para los filtros FIR, la ecuación en diferencias se corresponde con la convolución entre la respuesta al impulso y la señal de entrada.

$$Y[n] = h[n] * X[n] = \sum_{j=0}^{N-1} h_j X[n-j] \quad (3.2)$$

donde N es el orden del filtro y h_j son los coeficientes de la respuesta al impulso $h[n]$ [45].

Los filtros FIR presentan varias ventajas frente a los IIR. Una de ellas es la posibilidad de diseñar un filtro FIR de manera que cumpla con los requisitos de magnitud, obteniendo a su vez un retardo de grupo constante. El retardo de grupo se define como $\tau = -(d\theta/d\omega)$, donde θ es la respuesta de fase del filtro. Al ser τ constante, la respuesta de fase es lineal, es decir, todas las frecuencias se transmiten con el mismo retardo sin generar ninguna distorsión. Otra ventaja de estos filtros es que son siempre estables debido a que todos sus polos se encuentran en el origen [46].

El método más simple de diseño de filtros FIR se denomina método de ventanas. La respuesta en frecuencia deseada ideal se puede expresar como

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n} \quad (3.3)$$

siendo $h_d[n]$ la respuesta al impulso deseada. A partir de $h_d[n]$ se puede obtener una aproximación causal del filtro FIR, primero multiplicando la secuencia por una ventana de longitud finita $w[n]$ y luego introduciendo la causalidad al retardar en el tiempo la respuesta al impulso de la ventana [47]. Esto se puede expresar como

$$h[n] = h_d[n] w[n] \quad (3.4)$$

De modo que, la respuesta en frecuencia esta dada por la convolución periódica de la respuesta al impulso deseada con la transformada de Fourier de la ventana.

$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta \quad (3.5)$$

Por lo tanto, la respuesta en frecuencia será una versión distorsionada de la respuesta deseada. La selección de la ventana está sujeta a intentar que la duración

de $w[n]$ sea lo más corta posible, pero de modo que $W(e^{j\omega})$ se aproxime a un impulso; entrando ambos requerimientos en conflicto [48]. Las funciones de ventana más usuales son rectangular, Bartlett, Hanning, Hamming y Blackman y se pueden apreciar en la Figura 3.1.

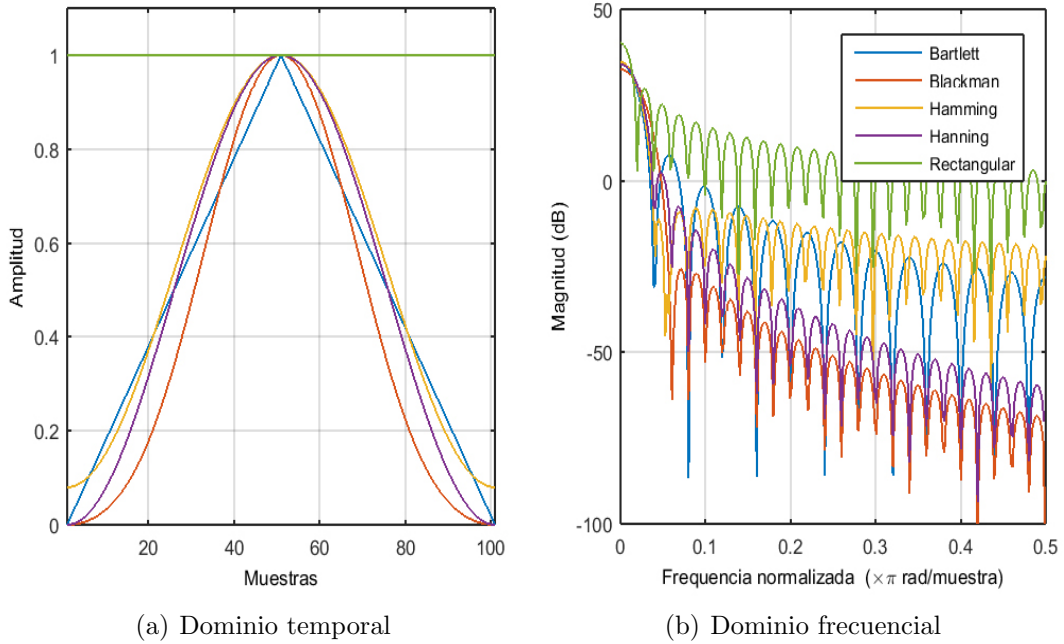


Figura 3.1: Ventanas más usuales para el diseño de filtro FIR por enventaneo: rectangular, Bartlett (triangular), Hanning, Hamming y Blackman.

3.4. Análisis de componentes independientes

El análisis de componentes independientes (Independent Component Analysis, ICA) es un método estadístico que tiene como objetivo expresar un conjunto de variables aleatorias como una combinación lineal de variables estadísticamente independientes [49]. ICA es una solución particular al problema denominado separación ciega de fuentes (Blind Source Separation, BBS), que consiste en obtener las señales originales solo a partir de la información contenida en la mezcla registrada desde cada uno de los canales. El uso de ICA en señales electroencefalográficas tiene como finalidad la remoción de artefactos, tales como el movimiento de ojos, el parpadeo y el ruido muscular, entre otros.

Uno de los ejemplos clásicos de BBS es el problema de la “fiesta de cóctel”, el cual plantea que en una habitación se encuentran dos individuos hablando simultáneamente, de modo que solo es posible escuchar la mezcla de voces sin poder identificarlas

por separado. Si se colocan dos micrófonos en la habitación, cada uno registraría una combinación lineal de las voces en función de la distancia respecto de la fuente [50]. Designando con $S_1(t)$ y $S_2(t)$ a las voces de cada individuo, se puede expresar la señal registrada en los micrófonos como

$$\begin{aligned} X_1(t) &= a_{11}S_1(t) + a_{12}S_2(t) \\ X_2(t) &= a_{21}S_1(t) + a_{22}S_2(t) \end{aligned} \tag{3.6}$$

En el caso de la señal de EEG, cada electrodo registra la combinación de múltiples potenciales generados en diferentes áreas de la corteza cerebral. Siendo, para un tiempo t , el valor de cada uno de los m canales $X(t) = [X_1(t), X_2(t), \dots, X_m(t)]^T$ una combinación de n fuentes desconocidas $S(t) = [S_1(t), S_2(t), \dots, S_n(t)]^T$, con $n \leq m$. La mezcla de las fuentes generalmente se considera lineal y la matriz de combinación \mathbf{A} , de $n \times m$, es tal que

$$X(t) = \mathbf{A}S(t) \tag{3.7}$$

El objetivo de ICA es encontrar una matriz de separación W que permita obtener un vector $Y(t)$ que se aproxime lo mejor posible a $S(t)$ [51].

$$Y(t) = \mathbf{W}X(t) \cong S(t) \tag{3.8}$$

Para emplear ICA en las señales de EEG, ciertas suposiciones se deben cumplir. La principal suposición es que las fuentes son independientes, lo que se considera cierto debido a la organización de la corteza cerebral en regiones compactas que realizan funciones especializadas. También se asume que los potenciales se propagan instantáneamente hacia los electrodos y se suman linealmente. Otras consideraciones son que las fuentes corticales de EEG permanecen fijas espacialmente durante el registro, que a lo sumo una presenta una función de densidad de probabilidad del tipo gaussiana y que la señal no presenta ruido [51, 52].

Existe una gran variedad de algoritmos que se emplean para estimar el vector $S(t)$ y la matriz \mathbf{A} , siendo los más comúnmente empleados INFOMAX y FastICA. El algoritmo INFOMAX, presentado por [53], utiliza el método del gradiente natural

para lograr minimizar la información mutua de los componentes independientes. En cambio, FastICA es un esquema iterativo de punto fijo que busca la dirección, dada por el vector \mathbf{w} , para la cual la proyección de $\mathbf{w}^T X(t)$ maximiza la no gaussianidad [50, 54].

3.5. Transformada Wavelet

Las señales de EEG son señales no estacionarias cuyo contenido frecuencial varía durante el tiempo de registro y de un electrodo a otro. Si se empleara la transformada de Fourier para el análisis, obtendríamos el espectro de la señal pero no podríamos identificar en que momento determinada frecuencia fue dominante en el registro. Tratando de superar esta limitación, se desarrolló la transformada de Fourier de tiempo corto (Short Time Fourier Transform, STFT). La STFT produce un análisis espectral local al desplazar una función ventana en el tiempo y realizar la transformada de Fourier del producto entre la señal y la ventana. La desventaja que presenta está dada por la resolución, la cual es la misma para todas las frecuencias, ya que el ancho de las ventanas es fijo [55].

La transformada Wavelet, a diferencia de la STFT, es una herramienta que permite un análisis multiresolución, al descomponer la señal mediante un conjunto de funciones obtenidas a partir de la traslación y de la dilatación, o contracción, de un prototipo denominado Wavelet madre [56]. Al dilatar la función madre obtenemos mejor precisión en frecuencia, mientras que si la contraemos mejoramos la resolución temporal. Una función para ser considerada Wavelet madre debe satisfacer el criterio de admisibilidad, que establece que la función posea media cero y se encuentre bien localizada tanto en tiempo como en frecuencia. Este criterio se puede expresar como

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (3.9)$$

Una familia Wavelet $\psi_{a,b}$ es un conjunto de funciones elementales obtenidas a partir de escalar y trasladar una función Wavelet madre $\psi(t)$.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (3.10)$$

donde $a, b \in \mathfrak{R}$, con $a \neq 0$. El parámetro a determina el grado de dilatación o con-

tracción y b define la traslación temporal; mientras que el factor $1/\sqrt{a}$ se utiliza para normalizar la energía [57]. La transformada de Fourier de la función Wavelet madre posee una morfología del tipo pasabanda, con centro en f_0 . Como se puede apreciar en la Figura 3.2, a medida que se dilata, la frecuencia central del filtro varía de f_0 a f_0/a .

Entonces, dada una señal $f(t)$ de energía finita, se define la transformada Wavelet continua (CWT) de f como

$$W_\psi(a, b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}^*(t)dt \quad (3.11)$$

donde $W_\psi(a, b)$ representa los coeficientes Wavelet y el asterisco representa el complejo conjugado [58].

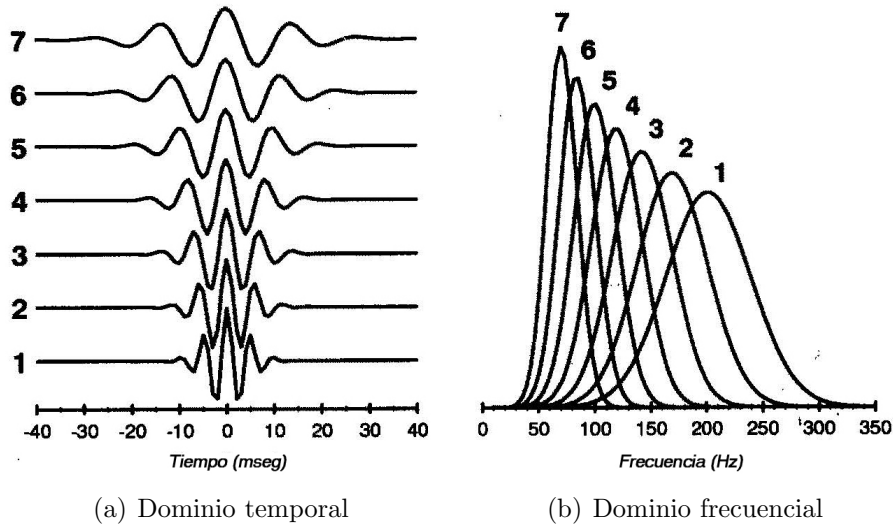


Figura 3.2: Dilatación de una función Wavelet del tipo Morlet. A medida que aumenta el valor de a , la función Wavelet se dilata y la banda de paso de la transformada de Fourier se vuelve más angosta. Extraído de [59]

Debido a la complejidad computacional y a la información redundante que aporta la transformada continua, se desarrolló la transformada Wavelet discreta (DWT). Esta transformada permite obtener una representación eficiente al emplear un esquema recursivo de filtros y, al mismo tiempo, elimina la información repetida al discretizar los parámetros de traslación y escala [57]. Usualmente, la discretización se realiza en potencias de 2, siendo a y b definidos como $a_j = 2^j$ y $b_{j,k} = k2^j$, $\forall j, k \in \mathbb{Z}$; de manera

que la Wavelet madre queda expresada por

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi(2^{-j}t - k) \quad (3.12)$$

Es posible expresar a la transformada Wavelet discreta de una función $f(t)$ como

$$WD_{\psi}(j, k) = \int_{-\infty}^{\infty} f(t) \psi_{j,k}^*(t) dt \quad (3.13)$$

Para la implementación de la TWD se emplea un esquema simple de filtros recursivos, compuesto por un par de filtros digitales complementarios cuyos coeficientes están determinados por la Wavelet madre elegida para el análisis. Este esquema se puede apreciar en la Figura 3.3.

Supongamos que una señal de n muestras con frecuencia de Nyquist f_N pasa a través de dichos filtros. A la salida se tendrán n coeficientes Wavelet por cada filtro, de los cuales se descarta la mitad al decimar por 2. La salida del pasaalto esta asociada a los coeficientes de detalle $d_{j,k}$ del nivel de descomposición j y comprende a las frecuencias mayores a $f_N/2^j$ de la señal de entrada. Mientras que del pasabajo se obtienen los coeficientes de aproximación $A_{j,k}$, que representan las componentes de baja frecuencia (menores a $f_N/2^j$) de la señal [60]. La salida del pasabajo ingresa a un nuevo esquema de filtros y este proceso se repite tantas veces como niveles de descomposición se requieran. Al finalizar, la señal se puede representar como un conjunto de coeficientes de detalle para cada uno de los niveles y un conjunto de aproximación correspondiente al último nivel.

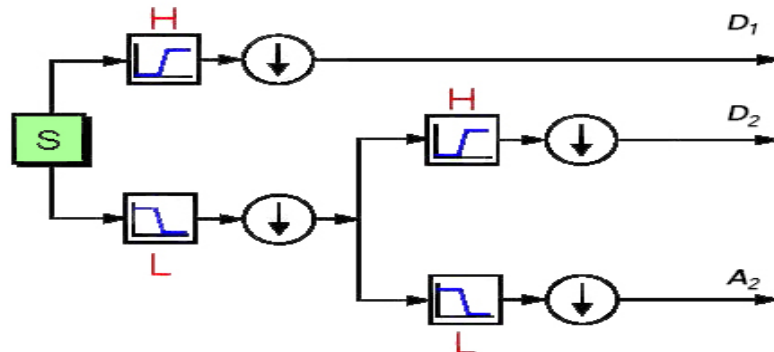


Figura 3.3: Esquema de descomposición de la transformada Wavelet discreta: la S indica la señal de entrada, las flechas hacia abajo representan un proceso de decimación y las H y las L son los filtros pasaalto y pasabajo, respectivamente. Modificado de [61]

3.6. Random Forest

El método de clasificación Random Forest (RF) emplea una combinación de árboles predictores, entendiéndose por estos a un procedimiento de clasificación que divide el conjunto de datos en pequeñas subdivisiones a partir de reglas definidas en cada rama de manera recursiva. En RF, cada árbol depende de los valores de un subconjunto aleatorio del vector de características, muestreado independientemente y con la misma distribución para todos los árboles. Cada árbol aleatorio es construido sin podar, adicionando nodos hasta que el nodo terminal posee no más de un determinado número de características o hasta que la clasificación sea única. Para cada vector de observación, cada uno de los árboles predice una clase y la salida del clasificador será aquella que posea la mayor cantidad de votos [62].

El algoritmo de un clasificador Random Forest, se puede resumir en las siguientes etapas [63]:

1. Crear un bosque de N árboles.
2. Construir cada árbol a partir de un subconjunto de vectores de características, seleccionados aleatoriamente y con reemplazo.
3. En cada nodo:
 - a) Seleccionar aleatoriamente m_{try} predictores.
 - b) Eligir la mejor separación de estas variables, minimizando el índice de impureza de Gini.
4. Predecir la clase para un nuevo dato de entrada mediante voto mayoritario.

El número de características m_{try} a emplear en cada árbol, así como también la cantidad de árboles N en el bosque, son los parámetros que debe definir el usuario al construir el clasificador. A medida que aumenta el número de árboles, el error de generalización siempre converge a un valor límite y no sufre de sobreentrenamiento. Mientras que la selección aleatoria de variables, permite disminuir la correlación entre los árboles y, en consecuencia, reducir la tasa de error del clasificador RF [64,65]. Otras ventajas que presenta son su simpleza, rapidez, capacidad para manejar gran cantidad de predictores y que es inherentemente multiclase.

3.7. Máquina de Vectores de Soporte

Un clasificador lineal es un algoritmo discriminante que utiliza funciones lineales para distinguir entre las clases. Dentro de este grupo de clasificadores se encuentran las máquinas de vectores de soporte (Support Vector Machine, SVM) que mapean un vector de entrada en un espacio de características multidimensional e intentan separar los datos mediante un hiperplano [66, 67]. A diferencia del análisis discriminante lineal, SVM intenta minimizar el error de generalización al aumentar los márgenes de separación entre el hiperplano y los datos.

Considérese un problema de clasificación binario como el que se ilustra en la Figura 3.4 y un set de entrenamiento de N datos $\{x_k, y_k\}_{k=1}^N$, donde $x_k \in \mathfrak{R}^N$ es el k -ésimo vector de entrada y $y_k \in \{1, -1\}$ es la clase. Se dice que dicho conjunto es linealmente separable si existe un vector \mathbf{w} y una constante b , tal que

$$y_k(\mathbf{w} \cdot \mathbf{x}_k + b) \geq 1, \quad k = 1 \dots N \quad (3.14)$$

El hiperplano óptimo $\mathbf{w}_0 \cdot \mathbf{x} + b_0 = 0$ es el único capaz de separar los datos de entrenamiento maximizando los márgenes. La dirección en la que el margen es máximo es $\mathbf{w}/|\mathbf{w}|$ y la distancia está dada por

$$\rho(\mathbf{w}_0, b_0) = \frac{2}{|\mathbf{w}_0|} \quad (3.15)$$

Los vectores \mathbf{x}_k para los cuales se satisface que $y_k(\mathbf{w} \cdot \mathbf{x}_k + b) = 1$ se denominan vectores de soporte y es posible expresar al hiperplano óptimo en función de ellos como:

$$\mathbf{w}_0 = \sum_{k=1}^N y_k \alpha_k \mathbf{x}_k \quad (3.16)$$

siendo α positiva y solo distinta de cero para los vectores de soporte [67, 68]. Se emplean una gran variedad de algoritmos para encontrar el hiperplano, entre los que se puede nombrar a los multiplicadores de Lagrange.

Cuando los vectores de características no son linealmente separables en su espacio, se emplea una transformación $\varphi(\mathbf{x})$, cuyo objetivo es llevar los vectores de entrada a un nuevo espacio (normalmente de dimensión mayor al original) que mejore

la separabilidad. Dentro de las transformaciones más usuales se encuentran la lineal, polinomial, función de base radial y sigmoide [69].

Los clasificadores SVM son robustos, presentan gran capacidad de generalización, comúnmente no sufren de sobreentrenamiento. La desventaja que poseen es que son clasificadores binarios, sin embargo, dos implementaciones se pueden usar para resolver problemas multiclase: «uno contra uno» y «uno contra el resto». Considerando N clases, en el primero, se crean $N(N - 1)/2$ clasificadores, cada uno evaluando una clase contra otra, siendo elegido el que presenta mayor cantidad de votos. En cambio, en el segundo se construyen N clasificadores considerando a la clase para la cual fue construido positiva y a las restantes negativas. La salida del clasificador será la clase que presente más amplio margen [70].

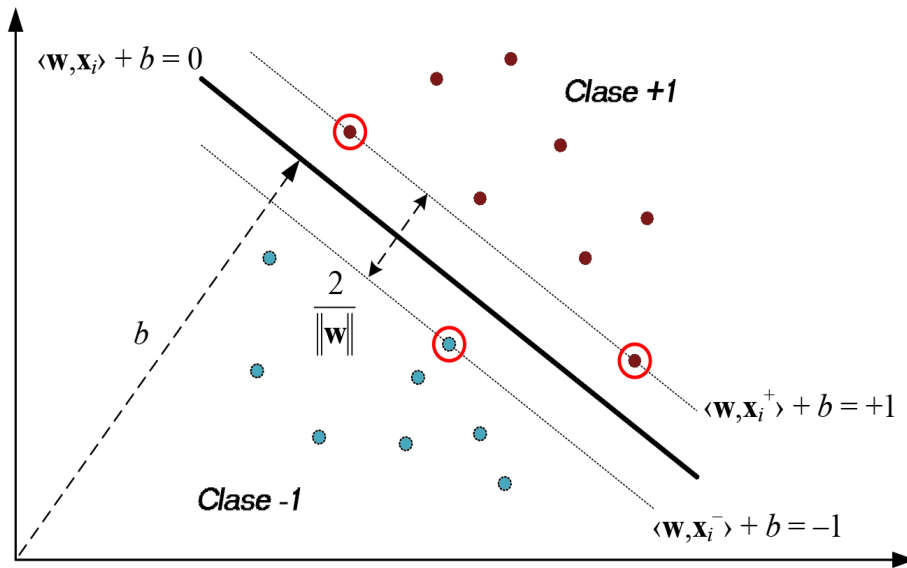


Figura 3.4: Hiperplano óptimo en un clasificador binario: encerrados en círculos rojos se destacan los vectores de soporte, mientras que el hiperplano está demarcado con línea gruesa. Modificado de [71]

Capítulo 4

Protocolo de adquisición

4.1. Introducción

La investigación en el área de habla imaginada es relativamente reciente y pocos son los trabajos que han versado sobre esta temática. Las características más relevantes de los protocolos de adquisición y los resultados obtenidos por algunos de ellos, se detallan en este capítulo. Para la obtención de la base de datos fue necesario diseñar un protocolo de ensayo que asegurase que todos los sujetos de prueba estuvieran bajo las mismas condiciones de registro.

En este capítulo, se describe en profundidad el protocolo diseñado, desde el posicionamiento de los electrodos hasta los distintos intervalos de adquisición. Su implementación incluyó el desarrollo de una aplicación sobre la plataforma BCI2000, que permitiese tener flexibilidad en cuanto a la configuración de los tiempos y estímulos. Además, para identificar falencias se realizó una prueba piloto y a partir de sus resultados se realizaron las modificaciones pertinentes, estableciendo el protocolo definitivo.

4.2. Estado del arte

Uno de los primeros trabajos acerca de habla imaginada fue el realizado por Suppes en 1997, quien utilizó los registros de EEG y magnetoencefalografía (MEG) para la clasificación de 7 palabras del idioma inglés: first, second, third, yes, no, right and left [72]. Para el registro de EEG de 7 sujetos empleó 16 canales y analizó las contribuciones de cada uno a la tasa de detección, siendo los preponderantes los que se hallaban en la región temporal(T3, T4, T5 y T6) seguidos de los de la región parie-

tal(C3 y C4). Durante el experimento evaluó tres situaciones: comprensión auditiva, discurso imaginado y discurso pronunciado. La tarea de comprensión auditiva fue realizada por la totalidad de la muestra; mientras que 5 hicieron la prueba de discurso imaginado y solo 2 la de discurso pronunciado. Para los registros de habla imaginada, se presentaron estímulos visuales de 200 *mseg* de duración, período luego del cual el sujeto debía imaginar inmediatamente la pronunciación de la palabra. Se obtuvieron 100 registros por cada una de las 7 palabras y alcanzaron una tasa de acierto del 52 % al emplear prototipos obtenidos a partir de la transformada de Fourier de la señal promedio de esa clase.

D’Zmura [73], al igual que Brigham y Kumar [74], empleó un protocolo basado en la adquisición de registros de EEG durante la imaginación de las sílabas */ba/* y */ku/* utilizando 128 canales de registro. El primero tuvo como muestra a 4 sujetos, mientras que el segundo realizó el protocolo sobre 7 individuos. En el trabajo de D’Zmura, las sílabas eran imaginadas a intervalos determinados por tres ritmos presentados en forma de clics, determinando seis condiciones a identificar y siendo cada una registrada 120 veces. En la etapa clasificación, se usaron filtros adaptativos, logrando tasas de detección del 62 % al 87 %. En cambio, Brigham, utilizó un solo ritmo en el cual la sílaba era repetida cada 1.5 segundos en un período de 6 segundos. Para la extracción de características se usó un modelo autoregresivo y se clasificó mediante el algoritmo *k*-vecinos más cercanos, resultando en tasas de acierto muy variables y con solo 4 sujetos superando el azar.

Un enfoque orientado al análisis de vocales imaginadas es el presentado por DaSalla [75], en el cual se registró la señal de EEG de tres sujetos en tres situaciones diferentes: al imaginar la pronunciación de las vocales */a/* y */u/* en forma sostenida, y durante un estado de control. El registro comenzaba con un estímulo auditivo, seguido de entre 2 y 3 segundos de preparación en el cual se mostraba en la pantalla una cruz. Luego se presentaba uno de los tres estímulos en un intervalo de 2 segundos y, a continuación, una señal de descanso durante 3 segundos. En este caso, como método de extracción de características se emplearon patrones espaciales comunes (Common Spatial Pattern, CSP) y como clasificador un SVM no lineal, produciendo un porcentaje de detección máximo de 78 % para dos clases.

Cuatro tesis han versado sobre el habla imaginada cuyo objetivo era la identi-

ficación de palabras del idioma inglés. La primera de ellas fue la de Wester [76], quien captó la señal de EEG de 6 sujetos bajo 5 modalidades de pronunciación, entre las que se encontraba el discurso normal y el imaginado. El protocolo consistía en la presentación de la palabra durante 1 segundo, seguido de otro intervalo de igual duración para que el individuo inspirara y exhalara. Posteriormente se mostraba una pantalla negra por 2 segundos, período en el cual el sujeto debía expresar la palabra según la modalidad seleccionada. Utilizando la STFT para la extracción de características y modelos ocultos de Markov como clasificador, Wester obtuvo un 21.8% de acierto para 10 clases.

Tanto la tesis de Wand [77] como la de Calliess [78], definen un protocolo de adquisición muy parecido, en el cual las palabras a imaginar son las primeras cinco letras del alfabeto radiofónico y que se presentan en bloque, de manera secuencial o en forma aleatoria. Si bien existen diferencias en el procesamiento, los resultados obtenidos para la modalidad aleatoria apenas superan al azar, mientras que durante la presentación en bloques la tasa de detección fue dos veces mayor. Debido a esta discrepancia, Porbadnikg [79] intentó determinar si el porcentaje de acierto se debía a artefactos temporales, utilizando un protocolo que contaba con la presentación de las mismas palabras con las mismas modalidades, pero registradas de manera intercalada en una misma sesión. Los resultados a los que arriba muestran una elevada influencia del orden en el cual se presentan las palabras en relación a la tasa de detección; siendo siempre superior en el caso de la presentación en bloques.

El único trabajo orientado a la imaginación de palabras del idioma español es el propuesto por Torres García [80]. En este se seleccionó un diccionario compuesto por las palabras: arriba, abajo, derecha, izquierda y seleccionar. Se realizó el registro de EEG a 27 individuos y el protocolo se basó en permitir que el sujeto indique, mediante clics del ratón, la ventana de tiempo en la cual imaginaba. La cantidad de repeticiones por palabra fue de 33, realizadas en forma de bloque. En dicha investigación se utilizaron para el registro 16 electrodos y se analizó la contribución de 4 posicionados en las áreas funcionales definidas en el modelo de Geschwind- Wernicke. Para la caracterización de la señal se empleó la DWT y se formó un vector con las energías relativas Wavelet en cada nivel de descomposición. En cuanto a la clasificación, se usaron tres clasificadores de los cuales el que mejor exactitud tuvo fue el algoritmo Random Forest, alcanzando

una tasa promedio de 44 %.

4.3. Protocolo propuesto

El objetivo de esta tesina es la elaboración de una base de datos de registros de EEG durante el habla imaginada, así como también, en el discurso pronunciado. En esta última situación, además de la señal electroencefalográfica, se debe poder adquirir la señal de audio emitida por cada sujeto. Para ello, se diseñó un protocolo de ensayo, con la finalidad de que todos los sujetos se encuentren sistemáticamente sometidos a las mismas condiciones de registro.

En este protocolo se definió como universo de estudio a jóvenes sanos de entre 20 y 30 años de edad, residentes en la ciudad de Paraná, o localidades vecinas, y que sean hablantes nativos del idioma español. Dichos sujetos de prueba no deben presentar deficiencias auditivas, debido a que esto produce modificaciones en el habla y, posiblemente, en la señal de EEG; siendo necesario descartar patologías graves mediante la realización de una audiometría tonal sencilla.

Al iniciar el ensayo, se le explicará al sujeto las características del experimento y se le brindará la posibilidad de hacer preguntas acerca de la finalidad del mismo, del destino de los datos recabados y de las condiciones de anonimato bajo las que se van a realizar los registros. Cada sujeto, además deberá completar un pequeño formulario con sus datos personales, tales como nombre completo, nacionalidad, domicilio, edad y si era diestro o zurdo. También se le pedirá que firme el documento de consentimiento informado, en el que explícitamente se expresa el carácter confidencial de los datos personales proporcionados durante el ensayo.

Posteriormente, el sujeto deberá sentarse en un sillón confortable, ubicado a una distancia determinada de un monitor LCD, siendo esta distancia dependiente del espacio disponible, pero constante para todos los sujetos. El lugar debe estar alejado de fuentes de ruido, a fin de evitar interferencias en los registros de audio y desconcentrar al sujeto durante la prueba. También deberá ser una zona de poco tránsito, de manera que el desplazamiento de personas en los alrededores no perturbe el estado de concentración del individuo. Para evitar la sudoración y, por consiguiente, cambios en la impedancia de la juntura electrodo-piel, las condiciones ambientales dentro del

recinto deben ser las adecuadas para que el sujeto se sienta cómodo.

Los registros se adquirirán mediante electrodos de copa de Ag-ClAg bañados en oro, sin la utilización de una gorra elásticas. Esto último se definió en función de que, en registros prolongados, el uso de gorras provoca molestias y dolor en los sujetos, ya que los electrodos están afirmados a presión al cuero cabelludo. La cantidad de electrodos a utilizar será de 8; siendo seis usados como canales activos, uno como referencia y el restante como tierra. La limitación a usar solo 8 electrodos está impuesta por los equipos de registro disponibles.

El posicionamiento de los electrodos se hará en base al sistema internacional 10-20 y estará dado por las posiciones F3, F4, C3, C4, P3 y P4 para los canales activos. Esta elección se justifica en que las mismas se encuentran cercanas a las áreas cerebrales relacionadas con la producción del lenguaje según el modelo de Geschwind-Wernicke y que, a su vez, permiten que durante el registro de habla pronunciada no se capte gran cantidad de ruido muscular. Como referencia se utilizará un electrodo en la apófisis mastoides izquierda; mientras que la tierra del equipo estará ubicada sobre la mastoides derecha.

El protocolo propuesto correspondería a un modelo de BCI de naturaleza exógena, dado que se empleará estimulación visual y auditiva para la producción de respuestas fisiológicas moduladas por el usuario mediante tareas cognitivas. En cuanto a los estímulos a presentar, el vocabulario seleccionado está compuesto por tres diccionarios, uno que contiene a las vocales del idioma español, otro que incluye palabras indicando comandos, y el tercero formado por un conjunto de palabras monosilábicas, cada una conteniendo a una de las vocales. El último grupo se seleccionó de una lista de monosílabos fonéticamente balanceados en vocales [81]. En el Cuadro 4.1 se presentan las palabras comprendidas por cada diccionario.

Cuadro 4.1: Diccionarios propuestos en el protocolo

Diccionario	Componentes
Vocales	«a», «e», «i», «o» y «u»
Monosílabos	«mar», «sed», «crin», «voz» y «tul»
Comandos	«arriba», «abajo», «derecha», «izquierda», «adelante» y «atrás»

Con respecto a la adquisición, se definieron dos modalidades. Una en la que el sujeto debe imaginar la pronunciación de los componentes del corpus, sin la rea-

lización de movimientos ni emisión de sonidos; mientras que en la otra, las deberá pronunciar. En este último caso será primordial que el sujeto, sin dejar de pronunciar correctamente, minimice los movimientos para la producción del habla, permitiendo esto reducir los artefactos musculares captados en la señal de EEG. En la modalidad pronunciada se grabará mediante un micrófono la señal de audio, debiendo estar la misma sincronizada con los registros electroencefalográficos. Además, para no introducir sesgo durante la adquisición, el orden de presentación de las modalidades como el de los componentes de los diccionarios deberá ser aleatorio. Otra razón de que el orden sea aleatorio, es que representa más naturalmente el proceso de habla imaginada.

En el protocolo se definió que la cantidad de adquisiciones por palabra será 50, de las cuales 40 se harán bajo la modalidad imaginada mientras que las restantes(10) serán pronunciadas. De modo que, al finalizar el ensayo, se tendrán 800 registros por cada sujeto. Las señales serán inspeccionadas visualmente y se anotarán aquellas en las cuales se observen artefactos. Dependiendo de la cantidad de los registros contaminados, se realizarán registros adicionales para intentar cumplimentar con la cuota propuesta de adquisiciones.

La secuencia de presentación de los diferentes estímulos estará dividida en 4 intervalos.

1. **Intervalo de concentración** (2 seg): Este período tiene como finalidad indicar al sujeto que el intervalo de descanso ha culminado y que pronto se presentará un nuevo estímulo.
2. **Presentación del estímulo** (2 seg): Durante este lapso de tiempo se presenta la palabra del vocabulario seleccionada. El estímulo puede ser solo visual o visual y auditivo.
3. **Período de imaginación/pronunciación** (4 seg): Dependiendo de la modalidad, el sujeto en este período deberá imaginar la pronunciación, o llevar a cabo la pronunciación, del estímulo indicado en el intervalo anterior.
4. **Intervalo de descanso** (4 seg): Este intervalo le permite al sujeto tener una pausa entre estímulos, pudiendo realizar movimientos, masticar o parpadear con normalidad.

La duración de cada uno de los intervalos fue elegida arbitrariamente, considerando que debía ser lo suficientemente larga para permitir una correcta adquisición de la señal de EEG, pero sin prolongar innecesariamente el tiempo que el sujeto debía estar con los electrodos puestos. Se definieron ventanas de tiempo fijas, de modo de facilitar el seccionamiento de los registros que conformaran la base de datos, así como también, el preprocesamiento y la extracción de características.

En base a lo expuesto en [75], para la imaginación/pronunciación de las vocales se definió que fuera en forma sostenida durante los cuatro segundos de duración del intervalo. Con respecto a las restantes palabras, se emitirán, dentro de la ventana, una sucesión de 3 clics que indiquen el momento en el cual el sujeto debe realizar la imaginación o pronunciación; siendo un sistema parecido al implementado en [73].

4.4. Aplicación diseñada para el proceso de adquisición: ImaginedSpeech

Para la implementación del protocolo propuesto en la Sección 4.3, se decidió utilizar el software BCI2000, el cual es una plataforma de propósitos generales para la investigación en interfaces cerebro-computadora [82]. Dentro de las ventajas que presenta esta plataforma, se destacan que: es gratuita para fines de investigación, permite la generación de aplicaciones en lenguaje C++ acorde a las necesidades del protocolo y tiene gran aceptación por parte de la comunidad científica. Otra razón por la que se eligió dicho software es que tiene soporte para los equipos de registro disponibles. El sistema de BCI2000 está compuesto por cuatro módulos interconectados e independientes, los cuales se pueden apreciar en la Figura 4.1.

El primero de ellos es el módulo «Fuente», compuesto por el sistema de adquisición de datos y por el sistema de almacenamiento. Este módulo se comunica directamente con el hardware de adquisición y se encarga de recibir los datos, calibrarlos y enviarlos al módulo de «Procesamiento de señales»; siendo este quien realiza el preprocesamiento y la extracción de características en las investigaciones con BCI online. Las señales procesadas avanzan hasta el programa «Aplicación» y pueden ser empleadas para la predicción de una clase. También, en este se define la secuencia de presentación de estímulos y la generación de marcadores que indican la fase dentro de

la secuencia, algún tiempo específico o señales de sincronización. Las señales, luego de pasar por el bloque de aplicación, son enviadas hacia el módulo «Fuente» y son almacenadas en un archivo, en cuya cabecera se describen todos los parámetros empleados durante el registro. El módulo «Operador» intercomunica los distintos módulos y provee de una interfaz gráfica que le permite al investigador visualizar los parámetros del experimento en curso, así como también, el análisis de la señal en tiempo real. Además, este último controla el inicio, pausa y fin del registro.

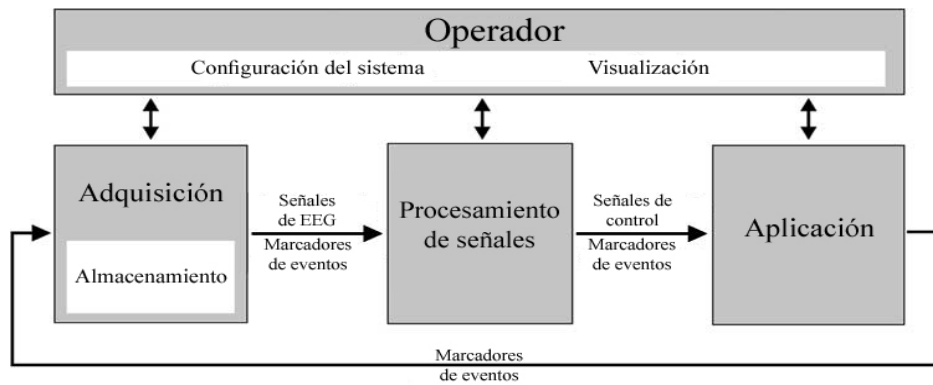


Figura 4.1: Esquema de la plataforma BCI2000: el sistema está compuesto por cuatro módulos denominados Fuente, Procesamiento de Señales, Aplicación y Operador. Modificado de [82]

Con el objetivo de obtener la secuencia de presentación de los estímulos descrita en la sección anterior, se programó un nuevo módulo de aplicación, denominado ImaginedSpeech; que emplea como referencia la aplicación StimulusPresentation provista por defecto con el código fuente de BCI2000. El programa fue escrito en lenguaje C++, utilizando el entorno de desarrollo VisualStudio 2010 Express Edition, de distribución gratuita. El código de la aplicación diseñada se encuentra en el Apéndice C. Durante la etapa de diseño se tuvo en cuenta que: la duración de los intervalos debía ser variable y definible por el usuario, que se almacenara una variable que indicase el estímulo presentado y que fuese posible grabar audio de manera sincronizada con los registros de EEG para la modalidad pronunciada. Asimismo, se consideró importante que el orden de aparición de los estímulos pudiera ser de manera aleatoria como secuencial; siendo el último modo más práctico en el caso de requerir la repetición de solo algunas palabras.

El programa creado emplea una sucesión de 5 bloques configurables llamados *PreRun*, *PreSequence*, *Imagine*, *PostSequence* y *PostRun*; estableciéndose la denomi-

nación en inglés, dado que es en ese idioma en el que se encuentra escrita toda la documentación de BCI2000. El orden en el que se suceden dichos bloques se puede ver en la Figura 4.2.



Figura 4.2: Sucesión de bloques en la aplicación ImaginedSpeech

El bloque *PreRun* es invocado al inicio del ensayo y provee de una pausa durante el tiempo determinado por *PreRunDuration*. *PreSequence* es el bloque siguiente y se subdivide en dos partes, denominadas *FocusOn* y *Stimulus*. En *FocusOn* se presenta una imagen en la pantalla, pudiendo o no estar acompañada de audio, que le indica al sujeto que debe prestar atención, dado que la pista de la palabra a evocar está próxima. Mientras que en *Stimulus* se presenta el estímulo, tanto de manera visual como auditiva. La duración de *FocusOn* y *Stimulus*, está determinada por *FocusOnDuration* y *StimulusDuration*, siendo también posible definir la duración de *PreSequence* mediante el parámetro *PreSequenceDuration*. En el caso de que este último sea mayor a la del conjunto *FocusOn* y *Stimulus*, se visualizará una pantalla negra durante el tiempo restante; mientras que si es menor, se advertirá al usuario y se modificará este parámetro para que coincida con la suma de *FocusOnDuration* y *StimulusDuration*.

El bloque *Imagine* demarcará el tiempo en el cual el sujeto debe imaginar o pronunciar el estímulo provisto en *Stimulus*. Para identificar si ese ensayo tenía como objetivo la imaginación o la pronunciación, se definió una variable booleana, denominada *PronounceSwitch*, que toma valor 0 en el primer caso y 1 en el segundo. Del mismo modo que para todos los bloques del sistema, existe un parámetro que establece su duración y se llama *ImagineDuration*.

A continuación, se inicia la etapa *PostSequence* y por un lapso igual *PostSequenceDuration* el sujeto tiene permitido acomodarse y descansar; a la espera de la aparición de un nuevo *PreSequence*. Dentro de este bloque se encuentra el intervalo *Rest*, de duración *RestDuration*, en el cual, mediante una imagen, se le indica al sujeto que se encuentra dentro del período de descanso. La secuencia *PreSequence-Imagine-PostSequence* se repite hasta cumplir con la cantidad de palabras definidas para el

ensayo y, luego, da comienzo al bloque *PostRun* que introduce una pausa de largo determinado por *PostRunDuration*, antes de finalizar el experimento. En la Figura 4.3 se puede observar la interfaz de configuración de los tiempos de cada bloque provista por la aplicación *ImaginedSpeech*.

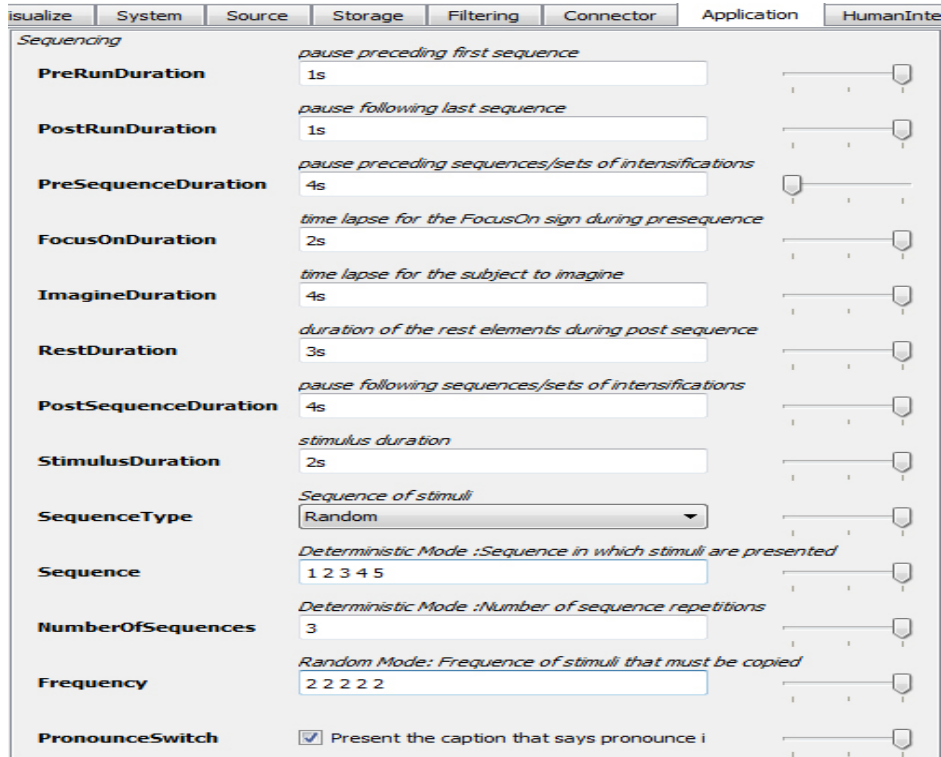


Figura 4.3: Ventana de configuración de la aplicación *ImaginedSpeech*: Sequencing es el apartado del esquema de configuración asociado con la duración de cada uno de los intervalos

Las imágenes y sonidos que se asocian a cada uno de los bloques se encuentran almacenados en las matrices *FocusOn*, *Stimuli*, *Imagine* y *Rest*. Cada matriz cuenta con igual cantidad de columnas que elementos se vayan a presentar y con 3 filas, denominadas *Caption*, *Icon* y *Sound*. En *Caption* se define una línea de texto que describe el elemento; mientras que en *Icon* y *Sound*, se especifican las rutas a los archivos de imagen y audio, respectivamente. En particular, en la matriz *Imagine* existen dos columnas, habla imaginada y pronunciada, siendo seleccionada una u otra en función del valor de *PronounceSwitch*.

La variable booleana *SequenceType* permite elegir entre dos modalidades para la presentación de los estímulos: determinística, si su valor es 0, y aleatoria, en el caso de ser igual a 1. En la primera, la secuencia se visualizará en el orden especificado en el parámetro *Sequence* y la cantidad de reiteraciones estará determinada por la variable

NumberOfSequences. En cambio, bajo la segunda modalidad, los estímulos establecidos en el vector Sequence se repetirán el número de veces indicado por el valor del vector Frequency en la posición correspondiente, pero el orden de presentación será aleatorio.

Para la adquisición de audio en forma sincronizada con el registro de EEG, se utilizó la contribución MicRecorderFilter realizada por Tim Blakely¹. Esta contribución crea un archivo de audio .wav, con el mismo nombre que el archivo .dat generado para almacenar las señales de EEG, y un estado, llamado MicSample, que indica el número de datos cargados al archivo .wav, pudiendo ser empleado como índice de sincronización entre la señal de EEG y el audio. La configuración de MicRecorderFilter se realiza mediante la pestaña HumanInterfaceDevices y posibilita la selección de la frecuencia de grabación y de la cantidad de canales; así como también del micrófono a emplear.

Para facilitar el seccionamiento de la señal en las etapas posteriores se definieron tres estados: InitImagine, StimulusCode y MicSampleImagine. El estado InitImagine es cero en todos los intervalos, excepto durante el bloque *Imagine*, en el cual toma el valor 1; permitiendo definir la sección en la cual se realiza la imaginación o pronunciación de las palabras del diccionario dentro de la señal continua de EEG. StimulusCode, en cambio, toma el valor del estímulo presentado durante Stimulus y asocia las señales de EEG con alguna de las palabras del diccionario. Por último, MicSampleImagine almacena la cantidad de bloques que se registraron en el archivo .wav al momento del cambio de fase, de *PreSequence* a *Imagine*; siendo de utilidad para determinar cuando, dentro del archivo de audio, comienza el discurso pronunciado.

4.5. Resultados de la prueba piloto

Con la finalidad de comprobar la practicidad y los criterios de selección de diversos parámetros del protocolo se llevó a cabo una prueba piloto. En ella, un sujeto de prueba se sometió al protocolo establecido en la Sección 4.3, empleando los mismos equipos y bajo las mismas condiciones que se utilizaron para adquirir los registros que conforman la base de datos y que se detallan en el siguiente capítulo.

A partir de dicha prueba se comprobó que la realización de registros usando los

¹Para más detalles dirigirse a <http://www.bci2000.org/wiki/index.php/Contributions:micRecorderFilter>

tres diccionarios, en una única sesión, era una tarea extremadamente agotadora para el sujeto; quien se reportaba distraído luego de la realización de aproximadamente el 50 % del experimento. En base a esto se decidió eliminar el diccionario de monosílabos y agregar tiempos de descanso, durante los que el sujeto pudiera caminar y despejarse. Además, se resolvió llevar registro del estado del sujeto, preguntándole, en una escala del 1 al 3, cuan cansado y cuan concentrado estaba.

Se pudo comprobar que se adquiriría una señal con menor ruido, durante el habla pronunciada, utilizando la configuración de electrodos propuesta que si se empleaban las posiciones más cercanas al lóbulo temporal. Un aspecto del protocolo que se corrigió fue el uso de auriculares tipo vincha, ya que producían ruido en la señal de EEG al moverse; por lo que se decidió utilizar para los estímulos auditivos auriculares intraaurales. Los parámetros de configuración del programa de aplicación ImaginedSpeech, encargados de definir la duración de cada intervalo, no sufrieron modificaciones.

Capítulo 5

Obtención de la base de datos

5.1. Sujetos de prueba

El protocolo propuesto en el Capítulo 4 se le practicó a 17 sujetos de prueba, de los cuales 9 eran varones y 8 mujeres. Todos eran estudiantes universitarios, la mayoría de la carrera Bioingeniería, excepto por dos que realizan sus estudios en la Universidad Nacional del Litoral, cursando uno Arquitectura y otro Medicina. Todos argentinos hablantes nativos del idioma español y tenían, al momento del ensayo, entre 24 y 28 años de edad, siendo la edad promedio de 25 años. La media para los varones fue de 25,3 años; mientras que la de las mujeres fue de 24,7 años. De los 17 sujetos que conforman la muestra, solo uno reportó ser zurdo, mientras que los restantes (16) informaron ser diestros.

Para descartar la presencia de alteraciones auditivas que produjeran modificaciones en el habla y en la señal de EEG, a cada individuo se le practicó una audiometría tonal; en particular, la propuesta por la Universidad de New South Wales en su página de Internet². Esta audiometría intenta definir una curva de isotonicidad, es decir, encontrar para las distintas frecuencias el valor de amplificación, con respecto a una señal patrón seleccionada por el usuario, al que se percibe la misma intensidad. Debido a esto, permite detectar alteraciones en la audición a determinadas frecuencias. No se identificó problemas en la audición en ninguno de los sujetos de prueba. La gráfica de isotonicidad de uno de los sujetos de prueba se puede observar en la Figura 5.1.

Cabe destacar que la respuesta en frecuencia de la placa de sonido de la PC, así como también el sistema de amplificador y auriculares empleado, tiene mucha

²Disponible en <http://newt.phys.unsw.edu.au/jw/hearing.html>

influencia en la curva obtenida. Esto resulta evidente tanto en las muy bajas frecuencias (30 y 45 Hz) como en las más altas (12 y 16 kHz), dado que saturaban los altavoces de los auriculares.

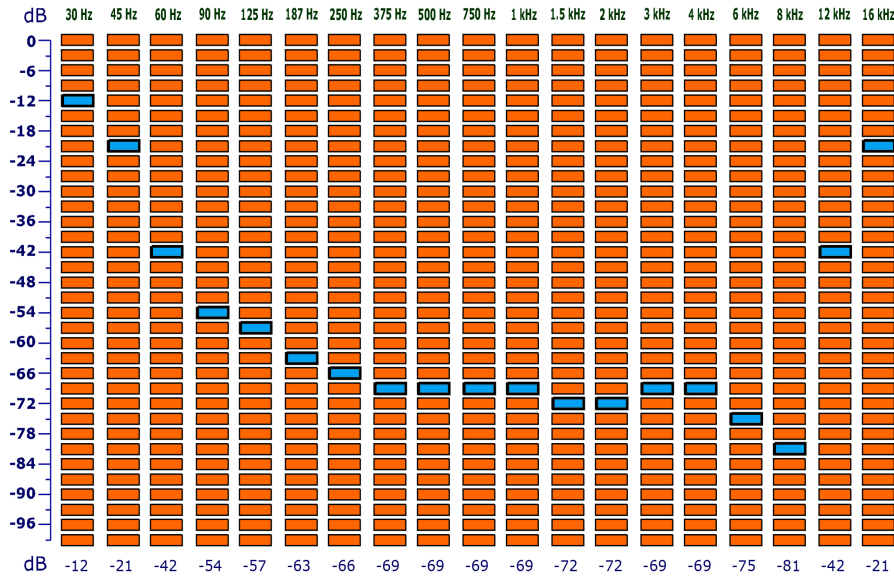


Figura 5.1: Gráfica de isotonicidad obtenida a partir de la audiometría practicada a un sujeto de prueba

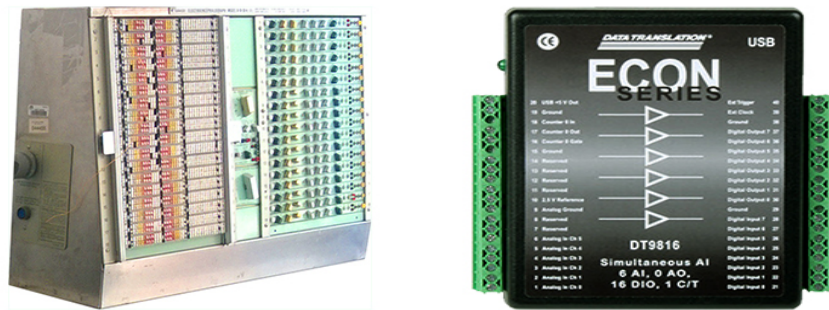
5.2. Equipamiento

Los electrodos empleados para la realización de los registros de la señal de EEG fueron 8 electrodos superficiales, tipo copa, de Ag-ClAg y recubiertos en oro. Cada uno de ellos se colocó, sobre el cuero cabelludo limpio, utilizando pasta conductora Ten20 que, además de reducir la impedancia de la unión entre el electrodo y la piel, brinda soporte para que los mismos no se despeguen.

Para la adquisición de las señales de EEG se utilizó un sistema compuesto por el amplificador Grass 8-18-36 en conjunto con la placa convertora analógico - digital DT9816 de la marca DataTranslation, ambos ilustrados en la Figura 5.2. El primero permite la amplificación de hasta 18 canales de adquisición simultáneos y presenta filtros analógicos pasaalto y pasabajo; siendo las frecuencias de corte seleccionables de 0.1, 0.3, 1 y 5 Hz y de 15, 35 y 70 Hz, respectivamente. Para la realización de los registros de EEG se utilizaron solo 6 de los canales del amplificador y se fijaron las frecuencias de corte en 0.3 Hz para el pasaalto y 35 Hz para el pasabajo.

Las señales analógicas amplificadas son convertidas a señales digitales me-

diante la placa convertora A/D DT9816. La misma posee 6 entradas analógicas y 8 entradas/salidas digitales, lo que condiciona que solo se puedan emplear 6 electrodos activos del amplificador. La máxima tasa de conversión es de 50 KS/s y la resolución es de 16 bits; además, el uso de igual cantidad de conversores A/D que de canales analógicos permite la conversión en simultaneo de todas las entradas. Se eligió para la adquisición de los registros de EEG una frecuencia de muestreo de 1024 Hz, cumpliendo ampliamente con el criterio Nyquist dado que la señal de interés tiene como frecuencia máxima 40 Hz. La placa presenta una interfaz USB, que permite la comunicación con una PC y su integración con el software base BCI2000 y la aplicación *ImaginedSpeech*.



(a) Grass 8-18-36

(b) DT 9816

Figura 5.2: Dispositivos para la adquisición de la señal de EEG

En cuanto a la adquisición de audio en simultáneo con la señal de EEG, se empleó el preamplificador MobilePre USB M-Audio junto con un micrófono, modelo SM58, de la marca Shure, como los que se muestran en la Figura 5.3. El MobilePre USB es un dispositivo que permite la amplificación y digitalización de las señales de audio. Este equipo admite un máximo de dos micrófonos y, a su vez, posibilita la amplificación de las señales de audio producidas por la plataforma BCI. Las señales digitalizadas poseen una resolución de 16 bits y es posible conectar el equipo a la PC, mediante su puerto USB.

El micrófono SM58 tiene una respuesta en frecuencia adaptada a voces y un filtro esférico que reduce significativamente el ruido producido por la respiración; siendo ambas características relevantes durante el registro en la modalidad de habla pronunciada. Además, para el montaje del micrófono se empleó un trípode, de manera que el sujeto de prueba no tenga que sostener el micrófono con la mano y, por consiguiente, realizar movimientos que introducirían ruido mioeléctrico en la señal de EEG.



(a) M-Audio MobilePre USB



(b) Shure SM58

Figura 5.3: Dispositivos para la adquisición de la señal de audio

Para la configuración de todo el sistema de adquisición al igual que el almacenamiento de los registros se empleó una notebook Lenovo G470, con un microprocesador Core i3 de 2.20 GHz y con 4 GB de memoria RAM. Sobre la misma se corría el programa BCI2000 utilizado para los registros y por medio de sus puertos USB estaba conectada al preamplificador y a la placa convertora. Además, se usó un monitor LCD para la presentación de la secuencia de intervalos propuesta en el protocolo.

5.3. Procedimiento previo al registro

Los registros fueron realizados en las oficinas del Laboratorio de Ingeniería en Rehabilitación e Investigaciones Neuromusculares y Sensoriales (LIRINS), ubicadas en el Centro de Prototipado de la Facultad de Ingeniería, perteneciente a la Universidad Nacional de Entre Ríos (UNER). Las oficinas se encuentran ubicadas en una zona tranquila y de poco tránsito, por lo que el desplazamiento de personas no fue un factor que influyese en la concentración de los sujetos. Sin embargo, en horas cercanas al mediodía, aumentaba notablemente el bullicio en los alrededores; siendo necesario, en el caso de detectarse algún ruido fuerte, la identificación del segmento y su posterior eliminación. En la Figura 5.4 se puede ver el espacio físico en el cual se llevaron a cabo los registros y un sujeto de prueba realizando el protocolo.

Con la finalidad de optimizar el tiempo disponible para el uso de los equipos, así como del espacio físico, se definieron dos turnos, uno por la mañana y otro por la tarde, en los cuales llevar a cabo la adquisición. Al arribar al lugar de registro, a cada sujeto se le indicó que tomara asiento en un sillón cómodo, frente a un monitor LCD. Luego, se le informó que podía retirarse del experimento en el momento que quisiera



Figura 5.4: Espacio físico en el que se llevaron a cabo los registros. En la imagen se observa a un sujeto de prueba realizando el protocolo de adquisición

y sin tener que dar explicaciones, así como también tomar la cantidad de descansos que necesitara. Debido a la larga duración del ensayo, se concedió al menos una pausa, aún sin ser pedida por el sujeto.

A su vez, se explicó el propósito del experimento y las condiciones bajo las cuales se iban a resguardar sus datos, y se brindó un espacio de tiempo en el cual pudiera preguntar lo que no haya quedado claro. Posteriormente, en el caso de aceptar someterse al estudio, se procedió a la firma del documento de consentimiento informado, siendo el modelo utilizado el que se encuentra en el apéndice A. Además, se les acercó un breve formulario que debían completar con sus datos personales, tales como nombre completo, nacionalidad, .

Luego, se continuó con la colocación de los electrodos en las posiciones F3, F4, C3, C4, P3 y P4, tal como se expuso en el capítulo anterior y que se corresponden con los canales 1, 2, 3, 4, 5 y 6 del registro, respectivamente. La demarcación de los lugares donde se debían colocar los electrodos se realizó utilizando una cinta métrica y un marcador al agua, teniendo en cuenta las proporciones estipuladas bajo el sistema internacional 10-20. Una vez definidas todas las posiciones, se limpió con alcohol cada zona y se procedió a la colocación de los electrodos, en conjunto con la pasta conductora. La Figura 5.5 muestra la distribución de los electrodos, de manera esquemática y sobre uno de los sujetos.

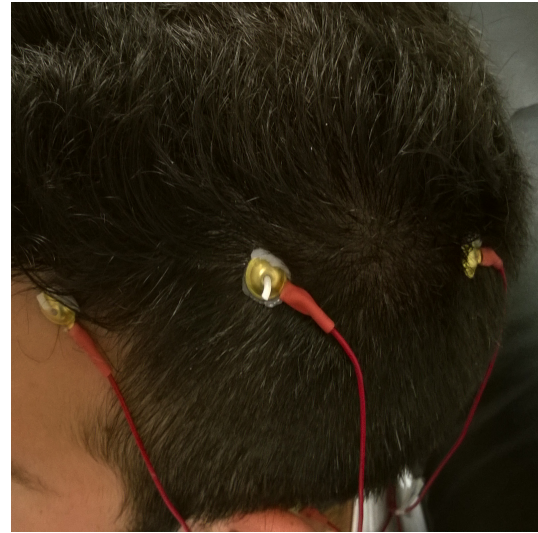
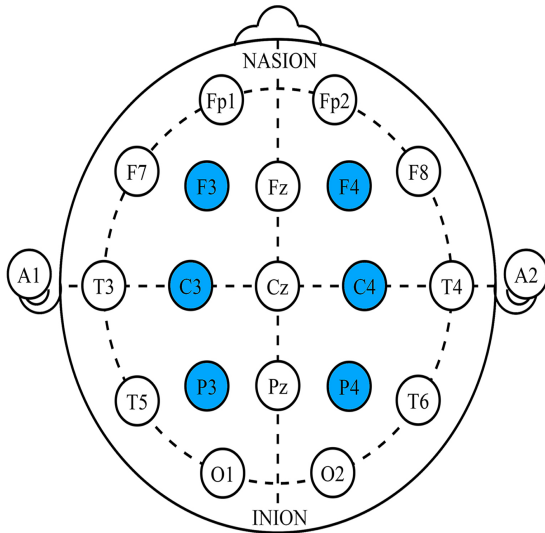


Figura 5.5: Distribución de los electrodos para el registro

Mientras se le colocaban los electrodos, a cada sujeto se le explicó como era el protocolo para la adquisición de los registros y las consideraciones a tener en cuenta. Se le informó que bajo la modalidad imaginada debía imaginar que pronunciaba la palabra indicada, sin realizar ningún movimiento; mientras que en la pronunciada debía decir la palabra de manera precisa. Debido a que los principales artefactos registrados en la señal de EEG, son provocados por el sujeto de prueba, se hizo especial énfasis en que debía permanecer lo más quieto posible y que, además, evitara tragar o parpadear durante el intervalo de imaginación/pronunciación. Otro requisito, era que debían mantener la mirada fija en la pantalla, evitando el movimiento de los ojos y, del mismo modo, se le pidió que durante el habla pronunciada, tratara de realizar la menor gesticulación posible pero manteniendo la inteligibilidad del discurso.

Por último, se revisó que las señales estuviesen siendo adquiridas con poca cantidad de ruido y de manera sincronizada. Para ello se empleó el sistema de visualización del programa BCI2000 y se le indicó al sujeto que parpadeara, evaluándose a simple vista la latencia con la que aparecía la onda negativa que caracteriza a este artefacto. También se realizaron pruebas con el sistema de adquisición de audio, pero dado que BCI2000 no posee una interfaz para la inspección de la señal de audio, estas fueron realizadas con el programa Audacity. En las mismas, el sujeto debía hablar normalmente, verificándose la correcta amplitud de la señal y su sincronía.

5.4. Registros

Previo al inicio formal del registro, se realizaron unas pruebas implementando el protocolo de presentación de estímulos, de manera que los sujetos se adaptaran a la metodología propuesta, como así también comenzaran todos bajo las mismas condiciones. Los registros de prueba fueron realizados bajo la modalidad pronunciada, ayudando al sujeto a encontrar la forma de pronunciación correcta que permitiese que la señal de EEG estuviese libre de ruido muscular. Particularmente, esto fue más complicado en sujetos con voz grave, por lo que fue necesario entrenar al sujeto para que usara un tono de voz más agudo. Luego de realizados dos registros con cada uno de los diccionarios, se procedió a los registros que conformarían la base de datos.

Tal como se propuso en el protocolo, se realizaron 50 registros de cada uno de los elementos, pero de solo dos de los diccionarios, el de vocales y el de comandos. Las señales de EEG obtenidas eran visualizadas mediante el sistema BCI2000 y se anotaban si las señales presentaban algún artefacto durante la ventana de imaginación/pronunciación. La modalidad, así como también el diccionario, eran seleccionados aleatoriamente por el supervisor, teniendo en cuenta que debía cumplirse con la proporción del 20% en modalidad pronunciada, es decir, 10 de los 50 registros.

Además, cada cinco registros, se le consultaba al sujeto cuan cansado y cuan concentrado estaba, a lo que respondía en una escala del 1 al 3; siendo 1 poco, 2 medio y 3 mucho. En los casos en que el sujeto estaba muy cansado y poco concentrado, se procedió a realizar un descanso de 5 minutos. Así mismo, aproximadamente habiendo cumplimentado la mitad de los registros, casi todos los sujetos pidieron un descanso de 10 minutos en el cual podían caminar y salir a tomar aire. A su regreso, se inspeccionó si los electrodos se habían salido y, si era necesario, se volvían a colocar.

En cuanto a la configuración de los intervalos para la aplicación *ImaginedSpeech*, los parámetros introducidos se resumen en el Cuadro 5.1. Para agilizar el proceso de adquisición, se decidió que por cada registro se presentaran dos veces cada uno de los estímulos de un diccionario; siendo para las vocales 10 estímulos y para los comandos 12. Esto se configuró mediante los parámetros `SequenceType`, `Sequence` y `Frequency`. A modo de ejemplo, para las vocales la configuración estaría dada por `SequenceType=1`, `Sequence=1 2 3 4 5` y `Frequency=2 2 2 2 2`, dado que la secuencia es aleatoria, los

números de identificación de las imágenes de las vocales van de 1 a 5 en la matriz Stimuli y que se estipula que cada estímulo sea presentado 2 veces.

Cuadro 5.1: Configuración de intervalos para ImaginedSpeech

Variable	Valor
PreRunDuration	1
PostRunDuration	1
PreSequenceDuration	4
FocusOnDuration	2
StimulusDuration	2
ImagineDuration	4
PostSequenceDuration	4
RestDuration	3

A partir de los valores indicados en el Cuadro 5.1, quedó definida la duración de cada uno de los registros; siendo para las vocales de 2 minutos y para los comandos de 2,43 minutos. Considerando que cada diccionario tendría 25 de estos registros, la duración total de la adquisición sería de aproximadamente dos horas. Sin embargo, dado que había pequeñas pausas entre cada uno de los registros, a lo cual se le adicionaban los tiempos de descanso y la colocación de los electrodos, la duración total del ensayo superaba las 3,5 horas.

Como paso inicial, al sujeto se le indicaba la modalidad y el diccionario, de modo que la primera palabra no lo tomara por sorpresa y condicionara el primer intervalo de imaginación/pronunciación de cada registro. La secuencia que se utilizó para la estimulación y repetición del vocabulario propuesto, se ilustra en la Figura 5.6. Comenzaba con la presentación de la imagen de concentración, durante dos segundos; seguido por un lapso de igual duración donde se indicaba la palabra objetivo. Posteriormente, se mostraba la modalidad que debía utilizar por un espacio de 4 segundos. Si el diccionario estaba compuesto por las vocales, las debían decir o pensar de manera sostenida durante el largo del intervalo; mientras que para el caso de los comandos se emitieron una sucesión de clics y la palabra tenía que ser repetida cada vez que los escuchaban. El primer clic comenzaba a los 0.8 segundos del inicio del intervalo y se repetía cada 1 segundo, siendo el propósito de la latencia inicial brindar un margen de tiempo para que el sujeto alcance a reconocer la modalidad. Por último, el sujeto tenía 4 segundos en los cuales descansar hasta la aparición de una nueva imagen de concentración, donde estaba permitido tragar, parpadear y acomodarse en el asiento.

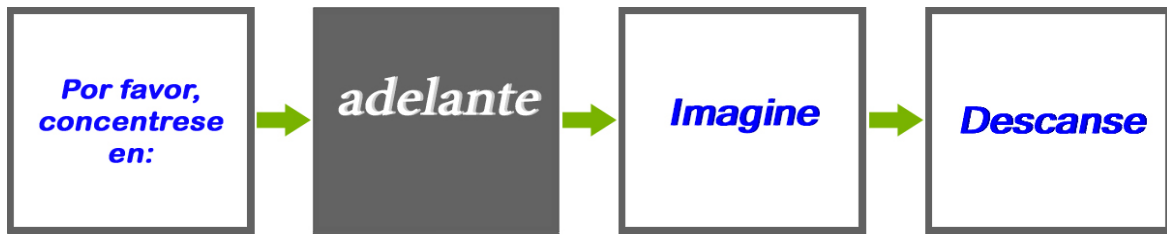


Figura 5.6: Secuencia de presentación de los estímulos

Esta secuencia se repetía hasta que todos los elementos del diccionario fuesen presentados dos veces. Cuando el sujeto no alcanzaba a leer la palabra objetivo, se distraía o imaginaba otra palabra, lo hacía saber al supervisor levantando el dedo índice. Dicho segmento del registro se marcaba para ser excluido posteriormente. Una vez realizadas las 50 repeticiones por palabra, se evaluaba la cantidad de segmentos y registros que habían presentado artefactos musculares, o que debían ser eliminados, y se estimaba el número de registros adicionales necesarios. Si bien esto dependía del sujeto, normalmente eran entre 6 y 8. La cantidad de registros realizados por sujeto, se puede encontrar en el Apéndice B.

5.5. Conformación de la base de datos

De los 17 sujetos de prueba sobre los que se realizó el protocolo propuesto, solo los registros de 15 de ellos pasaron a conformar la base de datos, 8 varones y 7 mujeres. De los registros que se descartaron, uno fue debido a que, durante la realización del experimento, surgieron problemas técnicos que prolongaron considerablemente el tiempo de registro con respecto a los restantes sujetos. El otro se desestimó debido a la presencia de gran cantidad de artefactos musculares en la señal de EEG, con la consecuencia de tener que descartar gran cantidad de registros.

Cada uno de los registros de los sujetos de prueba fueron inspeccionados visualmente mediante el toolbox de Matlab, EEGLAB. Este provee de una interfaz gráfica para el procesamiento de las señales de EEG y se lo eligió ya que permite la apertura de los archivos .dat obtenidos a partir de BCI2000, así como también presenta una gran cantidad de herramientas para la detección de artefactos. Uno de ellos, RUNICA, que realizó un análisis de componentes independientes, permitió discriminar de manera correcta los artefactos producidos por movimientos oculares. Solo se identi-

caron aquellos que ocurrían dentro de la ventana de imaginación/pronunciación. En la Figura 5.7, se puede observar la interfaz y la ventana de visualización de la señal de EEG, con los marcadores implementados por la aplicación ImaginedSpeech.

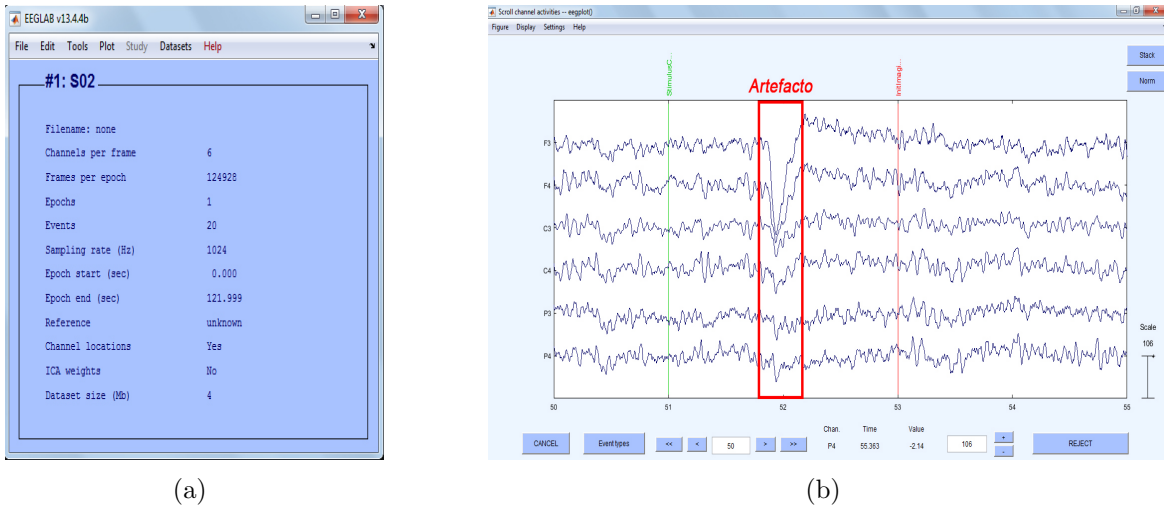


Figura 5.7: Interfaz gráfica del toolbox EEGLAB: En la imagen (A) se observa la pantalla principal de EEGLAB. La ventana de visualización de registros se muestra en (B), estando remarcado en rojo un artefacto producido por el cierre de ojos.

A partir de esto se generó una tabla de Excel por cada sujeto, en la que se identificó la modalidad y diccionario utilizado durante el registro de cada archivo .dat y se consignó el estímulo y la presencia de artefactos por cada segmento mediante etiquetas. El estímulo fue obtenido del marcador de evento StimulusCode, generado por la aplicación ImaginedSpeech, de cada uno de los registros; mientras que como etiqueta se utilizaron las siglas B, CO y M para identificar los segmentos correctos, los que contenían artefactos oculares y los que se debían descartar, respectivamente. Aquellos que poseían artefactos oculares no fueron descartados debido a que los mismos podían ser removidos mediante ICA.

Una vez clasificado cada uno de los segmentos, se procedió a filtrar estas señales utilizando filtros FIR. Esta elección está basada en que este tipo de filtro posee fase lineal y retardo de grupo constante, con lo cual no genera distorsión en la señal. La frecuencia de paso inferior se fijó en 2Hz, permitiendo la remoción de derivas; mientras que la superior se estableció en 40 Hz, que es la frecuencia máxima de interés y posibilita la reducción del ruido de línea presente en la señal. Cabe destacar que no se utilizó un filtro Notch para disminuir el ruido de línea, debido a que introduce una importante distorsión en fase.

La implementación de los filtros digitales se realizó en Matlab, utilizando un filtro pasabajo de orden 372 y un filtro pasaalto de orden 1204, obtenidos a partir de la ventana Kaiser. El orden de cada uno de estos filtros es elevado dado que la frecuencia de muestreo de la señal es 1024Hz, por lo que las bandas de transición en frecuencia relativa son muy estrechas. El filtro pasabajo se diseñó para lograr una atenuación de aproximadamente 60dB en 50Hz, lo cual se justifica debido a que la señal de EEG, por lo general, estaba contaminada con mucho ruido de línea. El retardo de grupo introducido por los filtros es de 788 muestras, pero dado que cada registro posee un segundo disponible al final, correspondiente a la etapa PostRun, esto no modificó ninguno de los segmentos de imaginación/pronunciación. Dicho retardo fue compensado mediante desplazamiento a la izquierda. En la Figura 5.8 se muestran las respuestas en frecuencia de los filtros pasaalto y pasabajo.

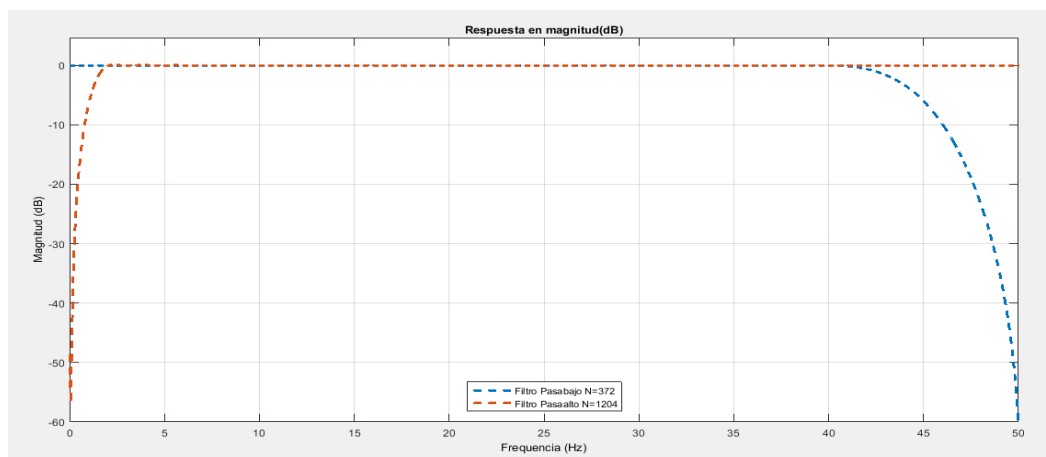


Figura 5.8: Respuesta en frecuencia de los filtros FIR implementados: En rojo se muestra la respuesta del pasaalto mientras que la del pasabajo se identifica en azul.

La segmentación de los registros filtrados se realizó utilizando las tablas de Excel, confeccionadas a partir de la inspección visual de los registros sin procesar. En cada uno de ellos, a partir de cada marcador InitImagine, se cortaban 4 segundos de la señal de todos los canales. En el caso de que ese segmento hubiera sido clasificado como M durante la inspección visual, se descartaba de la base de datos; mientras que si pertenecía a la clase B o CO se concatenaban cada uno de los canales, para obtener un vector por observación. En el último caso, se le adicionaban en orden tres etiquetas: «Modalidad», «Estímulo», «Artefactos». La etiqueta Modalidad está definida como 1 para habla imaginada y 2 para pronunciada; mientras que para Artefactos el valor 1

indica que no posee y 2 que presenta artefactos oculares. La correspondencia entre el valor de las etiquetas Estímulo y la palabra del vocabulario se detalla en el Cuadro 5.2. De este modo, el formato de cada observación está dado por un vector que contiene la señal registrada por los canales F3, F4, C3, C4, P3 y P4, de 4 segundos cada una, concatenadas en dicho orden y tres etiquetas; determinando la longitud de dicho vector en 24579 muestras. La totalidad de los registros de cada sujeto se almacena en un archivo Matlab, de extensión .mat, cuyas filas corresponden a cada una de las observaciones. El nombre del archivo está definido como Sxx_EEG, donde xx es el número de identificación del sujeto.

Cuadro 5.2: Valor de la etiqueta estímulo y la palabra del diccionario correspondiente

Estímulo	Vocal	Estímulo	Dirección
1	A	6	Arriba
2	E	7	Abajo
3	I	8	Adelante
4	O	9	Atrás
5	U	10	Derecha
		11	Izquierda

Con respecto a la señal de audio, se le aplicó un filtro pasabajo de 10kHz de frecuencia de corte con 12dB de atenuación por octava al archivo original y se procedió a realizar la segmentación. Para ello se empleó el marcador MicSampleImagine, el cual almacena el número de la muestra de la señal de audio que está sincronizada con la señal de EEG, cuando se produce la transición de 0 a 1 del marcador InitImagine. Cada uno de los segmentos de audio, de 4 segundos de duración y registrados por un único canal, fue cortado y se le adicionaron 2 etiquetas. La primera corresponde al estímulo presentado, utilizando los mismos valores que se muestran en el Cuadro 5.2, mientras que la segunda identifica la fila, en el archivo de EEG, del vector de observación que se corresponde con el audio. Una vez segmentadas todas las señales de audio, se creó un archivo, de extensión .mat, en el cual se almacenaron cada uno de los vectores de audio y cuyo nombre se define como Sxx_Audio, donde xx corresponde al número de identificación del sujeto.

5.6. Estructura y publicación de la base de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y organizados en una estructura que debe permitir el agregado de nueva información sin la necesidad de ser modificada. Para este proyecto se decidió organizar los datos obtenidos de los registros de EEG en una estructura de carpetas, similar a la empleada por el sitio Physionet, el cual es de amplia aceptación por parte de la comunidad científica. La simpleza de este esquema permite obtener bases de datos grandes, sin ser demasiado pesadas y además permite que la base de datos pueda ser ampliada de manera sencilla.

La carpeta principal contiene un archivo de texto que identifica los equipos empleados, así como también el protocolo utilizado, incluyendo la correspondencia entre los canales de registro y las señales almacenadas. Dentro de esta carpeta, se dispone de un archivo de texto con los sujetos que se sometieron al protocolo, la cantidad de registros por cada uno de ellos y cuantos por estímulo y modalidad.

Para cada sujeto, se creó una carpeta cuyo nombre se define bajo el formato Sxx, donde xx corresponde al número de identificación del sujeto. Dentro de esta última, se dispone de un archivo de texto, de nombre igual a la carpeta que lo contiene, con datos personales del sujeto de prueba, tales como sexo, edad y si era diestro o zurdo. Cada una de estas carpetas contienen los archivos .mat de los registros de EEG y audio, luego de ser filtrados y segmentados, siendo sus nombres iguales a los de la carpeta seguidos de _EEG y _Audio, respectivamente. En la Figura 5.9 se ilustra el esquema de almacenamiento de datos generado.

Como soporte físico para el almacenamiento de la base de datos se empleó un único DVD de 4,7 GB de capacidad. La misma tiene un peso total de 2,48 GB y contiene 15 carpetas (una por cada sujeto) y 47 archivos. La cantidad de registros que constituyen la base de datos es de 4201 para el diccionario de vocales y de 5113 para el de comandos. Los registros por sujeto, por estímulo y para cada modalidad se detallan en el Apéndice B. Dado que cada registro contiene 4 segundos, la duración total de la señal de EEG es de 37256 segundos (aproximadamente 10,3 horas), mientras que, en cuanto a la señal de audio, se almacenaron 7892 segundos (alrededor de 2,2 horas). El DVD original se dejará a la cátedra de Señales y Sistemas y una copia se dispondrá

en la biblioteca de la Facultad de Ingeniería.

Con respecto a la publicación de la base de datos todavía está en etapa de definición, debido a que se presentan dos posibles espacios de publicación para la misma. Uno es por medio de la base de datos de registros biológicos denominada BaSeBio, implementada en la Facultad de Ingeniería y descrita en [83]. Esta presenta la ventaja de la facilidad de acceso a las señales, así como también, provee de una estructura ya implementada y funcional para la carga de los archivos. Como desventaja se encontró que la estructura de la base de datos obtenida en este proyecto, debía ser modificada para amoldarse al formato de BaSeBio. La otra posibilidad es publicarla en la página del Laboratorio de Cibernética, de la misma unidad académica. En esta, no sería necesario modificar la estructura, pero si se debería generar un sistema de gestión, para administrar los datos e identificar usuarios y administradores. Por todo esto, aún no se determinó cual será la forma de publicación que permitirá mayor accesibilidad a los datos.

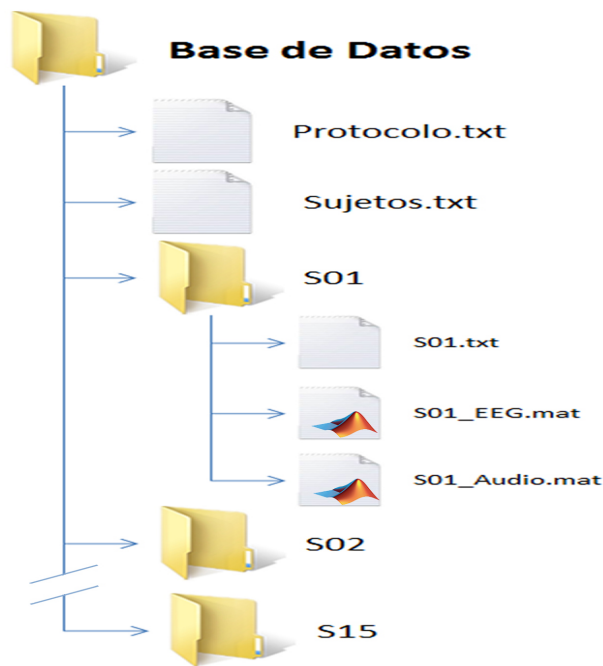


Figura 5.9: Estructura de la base de datos

Capítulo 6

Experimentación y resultados

6.1. Introducción

En este capítulo se describe una primera aproximación al procesamiento de las señales de habla imaginada utilizando los registros que conforman la base de datos. En la primera sección se describe el procesamiento realizado a los registros, tanto en la etapa de preprocesamiento, extracción de características y clasificación, mientras que en la segunda sección se hará el análisis de los resultados obtenidos empleando solo la señal de EEG del vocabulario que indica comandos.

6.2. Experimentación con la base de datos

En cuanto a la experimentación, dado que no es objetivo de esta tesina generar nuevos algoritmos para el procesamiento y clasificación de las palabras imaginadas, se decidió utilizar el método propuesto por Torres Garcia [80]. Dos consideraciones favorecieron la elección: la primera fue el uso de diccionarios similares y en el mismo idioma y la segunda la tasa de detección obtenida. Como se mencionó en la Sección 4.2, en ese trabajo las palabras utilizadas fueron «arriba», «abajo», «derecha», «izquierda» y «seleccionar»; obteniendo tasas de detección promedio del 44 %. Si bien se empleará el mismo método, la modalidad usada en el proceso de adquisición difiere entre dicho trabajo y esta tesina, especialmente en lo relacionado a la presentación de estímulos. En los registros utilizados por Torres la modalidad de imaginación era en bloque, es decir, se repetían la cantidad de veces necesaria una de las palabras del diccionario y luego recién se procedía a registrar otra.

Para la etapa de preprocesamiento, Torres propone emplear la referencia promedio común para cada uno de los registros, teniendo como objetivo eliminar los valores de la señal comunes a todos los canales. Esto último es ventajoso siempre que se utilicen gran cantidad de electrodos (mayor a 64), pero dado que ese no es el caso de los registros obtenidos para la base de datos, se decidió comparar los resultados obtenidos con y sin este método de preprocesamiento.

En cuanto a la extracción de características, se propuso el análisis multiresolución mediante la descomposición de la señal de EEG por medio de la transformada Wavelet discreta (DWT). Al igual que Torres, se utilizaron 5 niveles de descomposición, usando como Wavelet madre a una Daubechies de segundo orden (db2). En nuestro procedimiento, además de utilizar db2, se probó con las funciones Daubechies de orden 4, 6 y 8. Los coeficientes de detalle y aproximación obtenidos, se emplearon para el cálculo de la energía Wavelet relativa de cada una de los niveles.

La energía Wavelet relativa para el i -ésimo nivel se calcula como

$$EWR_i = \frac{E_i}{E_{Total}}, \quad i = 1 \dots N + 1 \quad (6.1)$$

donde N es el número de niveles de descomposición, E_i es la energía del i -ésimo nivel y E_{Total} es la energía total de los coeficientes wavelet. La energía wavelet del i -ésimo nivel está definida por

$$E_i = \begin{cases} \sum_j |d_{i,j}|^2 & \text{para } i \leq N \\ \sum_j |a_{i,j}|^2 & \text{en otro caso} \end{cases} \quad (6.2)$$

mientras que la energía total se obtiene como $E_{Total} = \sum E_i$.

Debido a la alta frecuencia de muestreo de la señal (1024Hz) se realizó un procedimiento de decimación (con $M=8$), para obtener una tasa de muestreo de 128Hz y una menor cantidad de datos por canal de registro. A partir de esta frecuencia se calcularon las bandas para cada uno de los niveles de descomposición, quedando descritas en el Cuadro 6.1.

Cuadro 6.1: Bandas de frecuencia por nivel de descomposición

Frecuencia[Hz]	Coefficientes
32-64	D1
16-32	D2
8-16	D3
4-8	D4
2-4	D5
<2	A5

Dado que el nivel de coeficientes D1, se asocia con ruido mioeléctrico, no se empleó para la obtención del vector de características. De esta forma cada uno de los canales estaría representado por un vector de cinco elementos correspondientes a las energías relativas wavelets de los niveles de detalle D2 a D5 y de aproximación A5.

Para la etapa de clasificación, se emplearon dos clasificadores: máquinas de vectores de soporte (SVM) y Random Forest (RF). Ambos clasificadores se implementaron utilizando WEKA, que es una plataforma desarrollada por la Universidad de Waikato, para el aprendizaje automático y la minería de datos. En el caso de Random Forest se probaron con 10, 50, 100 y 500 árboles, seleccionando 4, 5 o 6 características aleatoriamente. Para SVM se usó el kernel lineal y se probaron distintos coeficientes de complejidad C. Debido a que SVM es un clasificador binario, para permitir el uso en problemas multiclase se implementó un esquema de clasificadores «uno contra uno».

La medida para evaluar el desempeño de los clasificadores fue el porcentaje de exactitud, que mide la cantidad de aciertos entre el número total de observaciones pertenecientes a esa clase. Además, estos porcentajes fueron estimados empleando el sistema de validación cruzada de 10 particiones. En este sistema el conjunto de observaciones se divide en un número de subconjuntos, determinado por la cantidad de particiones, de los cuales uno se usa como prueba y el resto se emplea para entrenar al clasificador; y el proceso es reiterado hasta que cada una de las particiones haya sido usado como prueba una sola vez. Dicho proceso se repitió 10 veces con distintas semillas y luego se consignó el promedio.

6.3. Resultados

Los experimentos se realizaron utilizando el diccionario de comandos que contiene las palabras «abajo», «arriba», «derecha», «izquierda», «adelante» y «atrás», y

38 observaciones por cada una de ellas en la modalidad imaginada. Por cuestiones de complejidad computacional, se decidió realizar pruebas preliminares solamente con tres individuos S01, S02 y S03 de la base de datos y emplear luego el mejor clasificador con la totalidad de los sujetos.

Como primera premisa se evaluó la influencia del uso de CAR como método de preprocesamiento, cuando los registros se realizaron utilizando matriz dispersa de electrodos. Los resultados se obtuvieron usando como vector de características las energías relativas obtenidas a partir de la DWT de tres sujetos (S01, S02, S03), empleando como wavelet madre las funciones Daubechies de orden 2, 4, 6 y 8; y utilizando como clasificadores una máquina de vectores de soporte con coeficiente de complejidad 1 y el algoritmo Random Forest con 100 árboles y 5 características seleccionadas al azar. Los resultados se obtuvieron usando validación cruzada de 10 particiones y se computo el promedio de 10 iteraciones.

Los resultados obtenidos se muestran en el Cuadro 6.2, donde se consigna la diferencia aritmética entre el porcentaje de aciertos sin CAR y con CAR (%SinCAR - %ConCAR).

Cuadro 6.2: Diferencia aritmética entre los porcentajes de acierto obtenidos sin CAR y con CAR.

Clasificador	Random Forest				SVM			
	Wavelet	Db2	Db4	Db6	Db8	Db2	Db4	Db6
S01	4,55	4,01	3,22	1,52	6,53	5,33	5,59	-0,89
S02	0,68	1,12	1	0,63	1,93	1,38	1,4	8,18
S03	6,55	6,92	3,37	-0,5	4,21	5,53	3,33	-0,49
Media	3,92	4,01	2,53	0,55	4,22	4,08	3,08	2,26

Se observa que el uso de CAR disminuye la tasa de detección de palabras en la mayoría de los casos, y en los que no lo hace, no mejora de manera significativa el porcentaje de aciertos. Esto se corrobora con lo expuesto en [43], donde expresa que solo se recomienda el uso de esta técnica en aquellos esquemas de adquisición que emplean mallas de electrodos que cubren gran parte de la superficie del cuero cabelludo. En el caso de las señales de la base de datos, se observó que esta metodología de preprocesamiento, utilizaba como promedio las señales de los canales C3 y C4, eliminando la información en dichos canales. Es por ello, que solo se presentarán los resultados obtenidos sin el empleo de esta técnica de preprocesamiento.

En cuanto a los porcentajes de clasificación, no se encontraron diferencias usando los distintos ordenes de la función wavelet madre Daubechies. Dado que fue apenas mejor la tasa de detección con Db4, esta se utilizó para comparar los diferentes clasificadores. Al usar diferentes valores de C para el clasificador SVM, se observó que los mejores resultados se obtenían empleando $C=0,9$, mientras que para Random Forest, empleando una combinación de parámetros de 10, 50, 100, 200 y 500 árboles junto con 4, 5 o 6 características seleccionadas al azar, se registró una mejor clasificación para la configuración con 200 árboles y 5 características. Los resultados de ambos para tres sujetos se muestran en el Figura 6.1

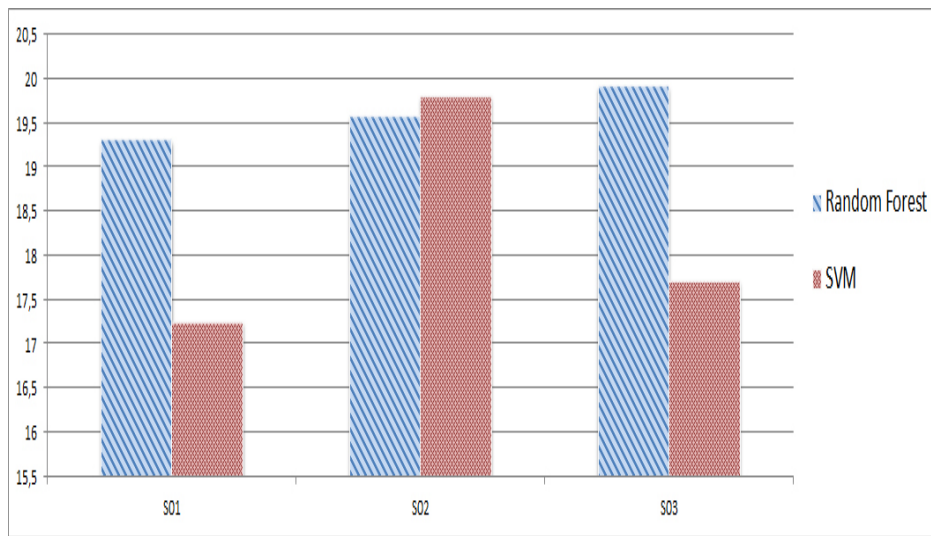


Figura 6.1: Resultados obtenidos para tres sujetos de prueba

A partir de la Figura 6.1, se puede observar que los resultados obtenidos son superiores a la chance para todos los sujetos (16,6%) y con ambos clasificadores. Dado que mejores porcentajes se alcanzaron empleando el algoritmo Random Forest, este se utilizará para la totalidad de los sujetos. La tasa de aciertos conseguida por cada uno de los sujetos, así como la media del total de la muestra, se presentan en el Cuadro 6.3. En la misma, los porcentajes en **negrita** corresponden a aquellos sujetos que superaron el porcentaje de acierto determinado por el azar. Si bien la gran mayoría de los sujetos obtuvieron un porcentaje de detección por encima de la chance, dicho porcentaje no fue significativamente más alto.

Cuadro 6.3: Porcentaje de aciertos para cada sujeto de la base de datos

Sujeto	Porcentaje de aciertos
S04	19,11 \pm 3,95
S05	16,58 \pm 4,25
S06	16,73 \pm 3,01
S07	17,34 \pm 3,17
S08	16,05 \pm 2,84
S09	20,18 \pm 1,72
S10	20,63 \pm 2,73
S11	17,49 \pm 4,30
S12	19,14 \pm 3,18
S13	18,77 \pm 1,97
S14	17,75 \pm 2,01
S15	20,18 \pm 1,68
Media	18,32 \pm 1,55

Comparando los resultados con los obtenidos por Torres, la tasa de aciertos es sustancialmente menor utilizando la base de datos generada en esta tesina. Una hipótesis acerca de esta discrepancia se fundamenta en la investigación realizada en [79], en la cual se compara la detección utilizando la modalidad de registro en forma de bloques y en forma aleatoria. En ella, los resultados indican que durante la primera se registran artefactos temporales lentos que ayudan al proceso de clasificación de las palabras y es debido a ellos que las tasas de acierto son elevadas.

En cambio, al comparar los valores alcanzados en el Cuadro 6.3, con los obtenidos para la modalidad aleatoria en los trabajos [76] y [78], se puede ver que los resultados son mejores a los reportados en dichos trabajos; para los cuales se consiguieron tasas de acierto por debajo del azar. Es importante destacar que el procesamiento implementado es solo una primera aproximación hacia clasificación de las señales de EEG durante el discurso imaginado, pudiéndose inferir que existe información que permitiría discriminar el habla en la señal de EEG; pero que para obtener tasas de acierto significativas es menester explorar nuevos algoritmos de extracción de características que tengan en cuenta también los parámetros temporales de la señal.

Capítulo 7

Enfoque económico

7.1. Introducción

La base de datos de registros de EEG, obtenida en esta tesina, no tiene como finalidad transformarse en un producto final comercializable, sino que intenta dar solución a la problemática de la inexistencia de bases de datos durante el proceso de imaginación de palabras y ser punto de partida para sucesivas investigaciones.

Teniendo esto en cuenta, el enfoque económico está dirigido hacia la obtención de financiamiento por parte de empresas u organismos nacionales e internacionales, que estén interesados en subvencionar proyectos de desarrollo científico. Este financiamiento sería utilizado para ampliar la base de datos y realizar nuevas investigaciones que permitan disminuir las tasas de error, con el objetivo final de lograr una implementación práctica de un sistema BCI basado en habla imaginada.

7.2. Fuente de financiamiento

Para la ampliación de la base de datos se buscó financiamiento en distintos entes, recurriendo finalmente a la Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT). Este organismo, dependiente del Ministerio de Ciencia, Tecnología e Innovación Productiva, tiene como misión apoyar proyectos y actividades cuya finalidad es la generación de nuevos conceptos científicos y tecnológicos desarrollados por investigadores pertenecientes a instituciones públicas y privadas sin fines de lucro radicadas en el país.

La Agencia posee cuatro fondos de financiamiento, denominados FONTAR,

FONARSEC, FONSOFT y FONCyT. El primero tiene como objetivo apoyar aquellos proyectos dirigidos a la mejora del proceso productivo en empresas privadas por medio de innovación tecnológica; mientras que FONARSEC financia proyectos que desarrollen capacidades críticas en áreas de alto impacto y que permitan su transferencia al sector productivo. El FONSOFT, en cambio, se orienta a la generación de nuevos emprendimientos y al fortalecimiento de las PyMES pertenecientes al sector de tecnologías de la información y la comunicación. Por último, el FONCyT tiene como finalidad subvencionar proyectos que permitan la generación de nuevos conocimientos, tanto científicos como tecnológicos.

De estos fondos de financiamiento, el único que se encuadra dentro de la temática del proyecto, es el FONCyT, y, a su vez, dentro de este existen diversos programas de los cuales el más adecuado es el que apoya a Proyectos de Investigación Científica y Tecnológica (PICT). Para solicitar dicho financiamiento se debe completar una serie de formularios y cumplimentar ciertas pautas, que se encuentran detalladas en la página web de la Agencia ³.

7.3. Requisitos de PICT 2015

Los PICT presentan cuatro categorías I, II, III y V. En nuestro caso, el proyecto se encuadraría en la categoría I, que corresponde a TEMAS ABIERTOS e incluye a aquellos que tienen por objeto la generación de nuevos conocimientos en todas las áreas científico-tecnológicas. Los resultados de dichas investigaciones estarán destinados al dominio público y no están sujetos a condiciones de confidencialidad comercial.

Dentro de la categoría I, se definen tres tipos de presentación. El tipo A está conformado por un equipo de trabajo, el B por un único investigador joven y el C por un grupo de investigación en formación. De estos, el proyecto se enmarcaría en los de tipo A, donde existe un grupo responsable conformado por uno o más investigadores que asumen la responsabilidad científico, tecnológica y administrativa ante la ANPCyT, en conjunto con la institución beneficiaria. Dentro de este grupo se debe seleccionar un Investigador Responsable, quien será el encargado de realizar las gestiones técnico-administrativas con la Agencia.

³Disponible en <http://www.agencia.mincyt.gob.ar/>

Con respecto a la duración del proyecto, esta puede ser de 1, 2 o 3 años para los de Tipo A, Categoría I, siendo los montos máximos para los subsidios a solicitar presentados en el cuadro 7.1 y 7.2, según se establecen en las bases de la convocatoria PICT 2015 y en función de si se solicita o no beca.

Cuadro 7.1: Montos máximos por año y total para los proyectos de categorías I, II y III

Categorías	Áreas Temáticas	Monto 1° año Hasta	Monto 2° año Hasta	Monto 3° año Hasta	Monto Total Hasta
I, II y III	Todas las Áreas, excepto Cs. Económicas, Cs. Sociales y Cs. Humanas	\$ 250.000	\$ 250.000	\$ 250.000	\$ 750.000
	Cs. Económicas, Cs. Sociales y Cs. Humanas	\$ 160.000	\$ 160.000	\$ 160.000	\$ 480.000

Cuadro 7.2: Montos máximos por año y total para los proyectos de categorías I, II y III que solicitan beca

Categorías	Áreas Temáticas	Monto 1° año Hasta	Monto 2° año Hasta	Monto 3° año Hasta	Monto Total Hasta (sin incluir beca)
I, II y III	Todas las Áreas, excepto Cs. Económicas, Cs. Sociales y Cs. Humanas	\$ 210.000	\$ 210.000	\$ 210.000	\$ 630.000
	Cs. Económicas, Cs. Sociales y Cs. Humanas	\$ 125.000	\$ 125.000	\$ 125.000	\$ 375.000

En las bases de la convocatoria se especifica los rubros en los cuales se pueden utilizar los subsidios; siendo estos consignados en el cuadro 7.3, junto con los porcentajes máximos sobre el total del subsidio solicitado. En el caso de excederse el presupuesto en alguno de ellos, se deberá justificar la excepcionalidad del pedido.

En cuanto a los subsidios brindados por la FONCyT, estos bajo ningún concepto se podrán emplear para el pago de becas, salarios, sueldos o sobresueldos. Los gastos de administración del subsidio, hasta un máximo del 5% del total, serán reconocidos por el FONCyT y calculados directamente a partir del monto otorgado y no deben ser incluidos en el presupuesto.

Como contrapartida, la institución beneficiaria deberá realizar un aporte igual o superior al subsidio otorgado, ya sea con fondos propios o a través de terceros. Dentro de esta podrán consignarse los salarios de los integrantes del equipo de trabajo

Cuadro 7.3: Porcentajes máximos por rubro

Rubros	Valores indicativos topes para cada rubro (% de los montos máximos sobre el total de subsidio solicitado)
1. Insumos	60 %
2. Bibliografía	20 %
3. Publicaciones de resultados del proyecto	20 %
4. Gastos de servicios técnicos especializados	20 %
5. Viajes y viáticos relacionados con el proyecto que realicen los integrantes del Equipo de Trabajo y los investigadores invitados. Gastos de inscripción a Congresos y otras Reuniones Científicas y para la realización de trabajos de campo en el país	30 %
6. Equipamiento	40 %

residentes en Argentina. No se podrá computar como contrapartida costos derivados del uso de las instalaciones e infraestructura existente.

7.4. Presentación del proyecto

La presentación del proyecto para la obtención del financiamiento se realiza mediante formularios en línea, provistos a través del Sistema de Gestión y presentación de Proyectos (SGP) del FONCyT. Estos deben ser completados con la información requerida en cinco pestañas distintas: Identificación, Impacto y otros, Presupuesto, Grupos y Adjuntos. A continuación se presenta la información requerida para completar los campos de las primeras tres pestañas.

Identificación del proyecto

Categoría: I

Tipo: A

Título: Identificación del habla imaginada a partir de la señal de EEG

Área temática principal: Medicina Humana y Salud Pública

Área temática secundaria: Tecnología Informática, de las Comunicaciones y Electrónica

Duración: 1 año

Provincia: Entre Ríos

Institución beneficiaria: Facultad de Ingeniería (UNER)

Resumen:

Las investigaciones orientadas a la identificación del habla imaginada mediante señales de EEG son recientes y escasas. Es por ello que el presente proyecto tiene como finalidad la exploración de nuevas técnicas de procesamiento y clasificación que permitan mejorar las tasas de detección del habla imaginada a partir de señales electroencefalográficas de sujetos sanos.

Se utilizarán los registros de una base de datos obtenida previamente, así como también, se adquirirán nuevas señales de EEG de sujetos durante el discurso imaginado y pronunciado. Con estos registros, se desarrollarán nuevos algoritmos para el preprocesamiento, la extracción de características y clasificación, evaluando la capacidad de discriminación de palabras. El proyecto no incluirá el diseño e implementación de un software para un sistema BCI ni el desarrollo de un prototipo para estas interfaces.

Impacto del proyecto

Este proyecto tiene como objetivo a futuro el desarrollo de sistemas BCI comandados con la imaginación de la pronunciación de determinadas palabras. Estos sistemas brindarían una alternativa de comunicación e interacción a pacientes que sufren del síndrome de enclaustramiento, o patologías similares, ya que permitiría que sin realizar movimiento alguno puedan comandar una silla de ruedas, un brazo robótico o un sintetizador de voz.

Además el uso de esta tecnología se podría extender a otros ámbitos, como por ejemplo en situaciones donde la interpretación del habla se ve dificultada por el ruido ambiente; proveyendo de una manera más natural de comunicación y que no requiere de entrenamiento por parte del usuario. Esto último, permitiría que las tasas de transmisión de datos fuesen superiores a las de otras metodologías implementadas para el control de sistemas BCI.

Presupuesto

En este apartado se debe hacer una descripción detallada del presupuesto planteado para la realización del proyecto, teniendo en cuenta la división por rubros

planteada en las bases de la convocatoria. Además, en el caso del presente proyecto no se solicitara beca, por lo que los montos máximos son los descriptos en el cuadro 7.1.

Costos de insumos

En el manual de administración de operaciones, se define como insumo a aquel material consumible, no inventariable necesario para la realización del proyecto. En el caso de este proyecto, se consideran insumos los electrodos empleados durante la adquisición de las señales de EEG, junto con la pasta conductora y otros elementos varios como alcohol y algodón. También se requerirá de la compra de un disco externo para el almacenamiento de una copia de seguridad de los registros obtenidos y los algoritmos realizados.

Los insumos para impresión de documentos tales como cartuchos de impresora y resmas de papel, así como otros insumos de librería no se pueden consignar en este ítem, por lo que deberán ser provistos por la unidad beneficiaria. En el cuadro 7.4 se indican los valores aproximados de los insumos.

Cuadro 7.4: Costos de insumos

Artículo	Cantidad	Monto financiado	Monto cofinanciado	Total
Electrodos	20	\$4.000	\$0,00	\$4.000
Pasta Conductora	2	\$2.000	\$0,00	\$2.000
Insumos Varios	1	\$1.000	\$0,00	\$1.000
Disco Externo	1	\$3.000	\$0,00	\$3.000
Total		\$10.000	\$0,00	\$10.000

Costos de bibliografía

En este rubro se deben consignar la adquisición de libros, publicaciones, accesos y suscripciones a revistas electrónicas que se empleen durante el desarrollo del proyecto. En el cuadro 7.5, se realiza un desglose de estos gastos.

Cuadro 7.5: Costos de bibliografía

Artículo	Cantidad	Monto financiado	Monto cofinanciado	Total
Artículos científicos	30	\$15.000	\$0,00	\$15.000
Libros	7	\$7,000	\$0,00	\$7.000
Total		\$22.000	\$0,00	\$22.000

Costos de publicación de resultados del proyecto

En los costos de publicación se debe indicar los gastos correspondientes a la compra de espacios de publicación en revistas o a la difusión de los resultados obtenidos en el proyecto en congresos. El cuadro 7.6 se consigna una estimación de los gastos para la publicación de los resultados.

Cuadro 7.6: Costos de publicación de resultados

Item	Monto financiado	Monto cofinanciado	Total
Publicaciones	\$20.000	\$0,00	\$20.000
Inscripciones a congresos	\$0,00	\$5.000	\$5.000
Total	\$20.000	\$5.000	\$25.000

Costos de servicios técnicos especializados

En este rubro se incluyen aquellos gastos surgidos de servicios profesionales prestados por personal especializado y servicios técnicos indispensables para la realización del proyecto. Dado que en el presente trabajo no se requerirán de servicios técnicos especializados, estos gastos no serán consignados.

Costos de viajes y viáticos

Según lo expuesto en el manual de administración de operaciones, esta categoría comprende los costos correspondientes a viajes y viáticos de los integrantes del grupo de investigación para la asistencia a congresos y simposios, para el cumplimiento de las tareas del proyecto. Los costos aproximados se pueden apreciar en el cuadro 7.7. Los gastos de inscripción a congresos y jornadas serán solventados por la unidad beneficiaria.

Cuadro 7.7: Costos de viajes e inscripciones

Item	Monto financiado	Monto cofinanciado	Total
Inscripción a Jornadas y Congresos	\$5.000	\$0,00	\$5.000
Viajes y Viáticos	\$30.000	\$0,00	\$30.000
Total	\$35.000	\$0,00	\$35.000

Costos de equipamiento

Como costos de equipamiento se consideran los gastos surgidos de la compra de una computadora personal para el almacenamiento de las señales y su posterior procesamiento en conjunto con una impresora. En el cuadro 7.8, se muestran los aportes solicitados a la entidad financiadora. Todos los valores expuestos en la tabla son aproximados y se obtuvieron al realizar una toma de precios.

Cuadro 7.8: Costos de equipamiento

Artículo	Cantidad	Monto financiado	Monto cofinanciado	Total
Computadora Portátil	1	\$20.000	\$0,00	\$20.000
Impresora	1	\$4.000	\$0,00	\$4.000
Micrófono	1	\$5.000	\$0,00	\$5.000
Preamplificador de audio	1	\$11.000	\$0,00	\$11.000
Total		\$40.000	\$0,00	\$40.000

Para la obtención de un subsidio que permita la compra de un equipo para adquisición de señales de EEG, se debería aplicar un Proyecto de Mejora de Equipamiento (PME) perteneciente al FONCyT. Este tiene como finalidad permitir adquisición o mejora de equipamiento de las unidades ejecutoras pertenecientes a instituciones dedicadas a la investigación científica. Debido a que el precio de uno de estos equipos es superior al límite impuesto para el rubro equipamiento en la presente convocatoria, no se lo podrá incluir en el presupuesto.

Costos de mano de obra

Para el cálculo de este costo se consideran los salarios de los investigadores que conforman el grupo de investigación. Los valores se deben consignar por un año de trabajo, incluyendo el aguinaldo. Además, en estos gastos se debe considerar el aporte jubilatorio y la obra social de los investigadores.

El grupo de investigación estará conformado por un profesor adjunto de dedicación semi-exclusiva y por un jefe de trabajos prácticos con dedicación parcial, ambos con una antigüedad del 50%. También se incorporará al presente proyecto un bioingeniero recién recibido, quien realizará la obtención de datos y el procesamiento, bajo

lo tutela de los restantes integrantes del grupo de investigación. Este último se debe encontrar dentro de un programa formal de doctorado acreditado por la CONEAU y con lo cual podrá solicitar una beca de doctorado al CONICET. El estipendio de dicha beca, se computará como un aporte realizado por la institución beneficiaria como se estipula en las bases de la convocatoria.

Los gastos referidos a mano de obra se muestran en el cuadro 7.9, siendo los montos indicados correspondientes a salarios brutos anuales.

Cuadro 7.9: Costos de mano de obra

Cargo	Monto financiado	Monto cofinanciado	Total
Profesor adjunto con dedicación semi- exclusiva	\$0,00	\$150.000	\$150.000
Jefe de trabajos prácticos con dedicación parcial	\$0,00	\$65.000	\$65.000
Becario CONICET	\$0,00	\$145.000	\$145.000
Total	\$0,00	\$360.000	\$360.000

Costo total del proyecto

En este apartado se realiza el cálculo total estimado del costo del proyecto, tanto por parte de la entidad beneficiaria como del ente de financiamiento, para los diferentes rubros detallados anteriormente. Dichos valores se muestran en el cuadro 7.10.

Cuadro 7.10: Costos totales por rubro

Rubro	Monto financiado	Monto cofinanciado	Total
Insumos	\$10.000	\$0,00	\$10.000
Bibliografía	\$22.000	\$0,00	\$22.000
Publicaciones de resultados	\$20.000	\$5.000	\$25.000
Gastos de servicios técnicos especializados	\$0,00	\$0,00	\$0,00
Viajes y Viáticos	\$35.000	\$0,00	\$35.000
Equipamiento	\$40.000	\$0,00	\$40.000
Mano de obra	\$0,00	\$360.000	\$360.000
Total	\$127.000	\$365.000	\$492.000

Considerando los límites máximos presentados en las bases de la convocatoria, se puede apreciar a partir de los datos consignados en el cuadro 7.10, que cada rubro cumplimenta con los topes establecidos. Los porcentajes del total del subsidio asignados a cada rubro se pueden observar en la figura 7.1. En cuanto al monto total solicitado en forma de subsidio, este no supera el valor máximo de \$250.000 para los proyectos de categoría I del tipo A, con duración total de 1 año.

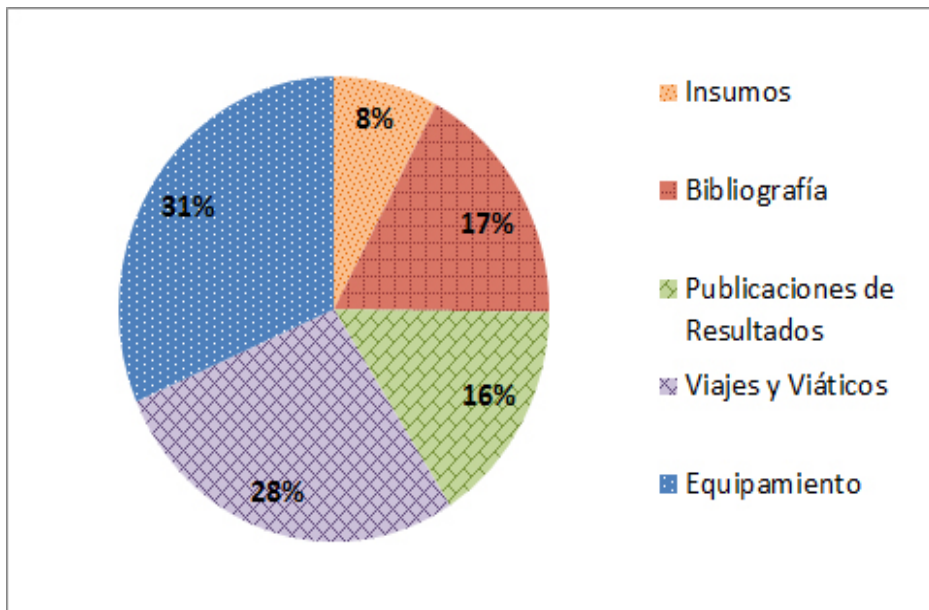


Figura 7.1: Porcentaje por rubro sobre el total del subsidio solicitado

Cabe destacar que si bien el valor cofinanciado supera con creces al subsidiado, el 98 % del mismo corresponde a costos de mano de obra. Además, dentro de los \$360.000 consignados a dicho rubro, el 40,2 % representa el monto de una beca doctoral durante 1 año, que será solventada por el CONICET.

A partir de este análisis, el costo total estimado para la implementación del proyecto sería de \$492.000. El 25,8 % del mismo, correspondiente a \$127.000, será subsidiado por el FONCyT a través del programa PICT; mientras que 74,1 % restante estaría a cargo de la unidad ejecutora y del CONICET.

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones

A partir de la presente tesina, se obtuvo una base de datos de registros de EEG de sujetos sanos durante la imaginación de la pronunciación, o la pronunciación de palabras. Esta base de datos se conformó con los registros de 15 sujetos pertenecientes al entorno universitario y que voluntariamente se prestaron para el ensayo. Se diseñó un protocolo de ensayo, en el cual el vocabulario utilizado corresponde a dos categorías, una que contempla las vocales del idioma español y otra que usa palabras que indican comandos. Además, en él se definieron dos modalidades, imaginada y pronunciada.

Para la adquisición de los registros, se llevó a cabo el diseño e implementación de una aplicación integrada a la plataforma BCI2000 que proveyese de una interfaz para el control de los intervalos impuestos por el protocolo de ensayo. Esta aplicación demostró ser útil para la adquisición de las señales de EEG y de audio, tanto en las modalidades pronunciada e imaginada.

Se comprobó a partir de este trabajo que la recolección de datos de EEG, es una tarea demandante de tiempo y que para su realización se requiere de equipamiento especializado en la adquisición de estas señales. A esto se le debe sumar el tiempo de inspección de los registros para descartar aquellos que se encuentran contaminados por artefactos, así como también el tiempo dedicado a la segmentación y acondicionamiento de las señales.

Por otra parte, se evaluó el resultado obtenido utilizando los registros de la base de datos mediante un algoritmo diseñado por otro autor, obteniéndose porcentajes de detección por encima del azar para 6 clases en varios sujetos. En cuanto a la

publicación de la base de datos, se propusieron dos posibles lugares de publicación; pero en ambos se requiere diseñar una interfaz para la gestión de la base de datos, lo cual ameritaría un trabajo posterior.

8.2. Trabajo futuro

Con la obtención de la base de datos objeto de esta tesina, se desprenden dos áreas para investigaciones futuras. La primera tendiente a la obtención de una mayor cantidad de registros de sujetos durante el habla imaginada, con la finalidad de brindar mayor generalidad a la base de datos, así como también una ampliación en cuanto al vocabulario empleado. A su vez, sería de interés el registro de señales de EEG durante el habla imaginada en sujetos que presentan discapacidades físicas severas, tales como aquellos pacientes en estado de enclaustramiento. También se podrían realizar registros tanto en la modalidad aleatoria como en bloques, de manera de comparar el impacto del orden en el que se presentan los estímulos.

La segunda línea de investigación correspondería al uso de la base de datos para la evaluación de otras técnicas de preprocesamiento, extracción de características y clasificación. En cuanto a las técnicas de preprocesamiento se podría probar la utilidad de descomponer la señal mediante ICA o PCA. Para la extracción de características, se podrían utilizar distintos tipos de Wavelet madre, emplear la transformada de Fourier de tiempo corto o modelos autoregresivos. Con respecto a la clasificación se podría considerar el uso de redes neuronales o la utilización de ensambles de clasificadores.

Apéndice A

Consentimiento Informado

Yo

DNI/Pasaporte:

He sido informado acerca del protocolo del estudio

He tenido oportunidad de efectuar preguntas sobre el estudio

He recibido respuestas satisfactorias

He recibido suficiente información del estudio

Entiendo que la participación es voluntaria

Entiendo que puedo abandonar el estudio:

- Cuando lo desee.
- Sin tener que dar explicaciones

Doy consentimiento para la realización de registros de electroencefalograma, con el fin de obtener una base de datos de dichos registros, que será empleada en estas y futuras investigaciones. No se utilizará mi nombre, DNI o domicilio en la base de datos.

Declaro que he leído y conozco el contenido del presente documento, comprendo los compromisos que asumo y los acepto expresamente. Y, por ello, firmo este consentimiento informado de forma voluntaria para manifestar mi deseo de participar en este estudio de investigación denominado **Obtención de una base de datos de registros de EEG orientados a la clasificación de habla imaginada.**

Nombre del sujeto:

DNI/Pasaporte:

Firma:

Fecha:

Nombre del investigador:

Firma:

Fecha:

Apéndice B

Registros por sujeto

En el cuadro B.1 se consignan los datos recabados al inicio de cada uno de los registros, que refieren a información personal de los sujetos de prueba. En los cuadros B.2 y B.3 se indican la cantidad de registros por sujeto, bajo ambas modalidades, para las vocales y las direcciones, respectivamente.

Cuadro B.1: Información personal de los sujetos de prueba

ID	Sexo	Edad	Diestro/Zurdo	Nacionalidad	Provincia
S01	Femenino	24	Diestro	Argentina	Entre Ríos
S02	Femenino	28	Diestro	Argentina	San Luis
S03	Femenino	24	Diestro	Argentina	Entre Ríos
S04	Masculino	25	Diestro	Argentina	Tierra del Fuego
S05	Femenino	24	Zurdo	Argentina	Entre Ríos
S06	Masculino	25	Diestro	Argentina	Entre Ríos
S07	Masculino	25	Diestro	Argentina	Neuquén
S08	Masculino	25	Diestro	Argentina	Santa Fe
S09	Masculino	24	Diestro	Argentina	Buenos Aires
S10	Masculino	26	Diestro	Argentina	Buenos Aires
S11	Femenino	25	Diestro	Argentina	Entre Ríos
S12	Masculino	28	Diestro	Argentina	Santa Fe
S13	Femenino	25	Diestro	Argentina	Entre Ríos
S14	Masculino	25	Diestro	Argentina	Entre Ríos
S15	Femenino	24	Diestro	Argentina	Entre Ríos

Cuadro B.2: Cantidad de registros por sujeto y modalidad para cada una de las vocales

ID	A		E		I		O		U		Total
	Imaginada	Pronunciada	Imaginada	Pronunciada	Imaginada	Pronunciada	Imaginada	Pronunciada	Imaginada	Pronunciada	
S01	56	14	53	10	55	12	55	10	55	10	330
S02	44	12	44	12	44	12	44	12	44	12	280
S03	42	13	44	12	44	13	44	14	42	13	281
S04	41	11	43	12	43	12	41	11	41	11	266
S05	43	11	43	11	44	9	42	9	43	12	267
S06	43	12	43	12	38	12	43	11	44	12	270
S07	44	12	37	11	38	12	39	12	42	10	257
S08	44	12	42	12	44	12	43	11	45	12	277
S09	49	11	48	12	48	14	47	13	48	13	303
S10	44	14	42	14	44	14	42	13	43	14	284
S11	43	12	44	12	43	12	44	12	44	12	278
S12	44	12	44	11	43	12	44	11	44	11	276
S13	44	11	43	10	43	12	43	12	44	12	274
S14	46	11	46	11	46	11	46	12	45	11	285
S15	45	11	44	11	45	12	38	12	43	12	273
Total	672	179	660	173	662	181	655	175	667	177	4201

Cuadro B.3: Cantidad de registros por sujeto y modalidad para cada una de las direcciones

ID	Arriba		Abajo		Adelante		Atrás		Derecha		Izquierda		Total
	Imaginada	Pronunciada	Imaginada	Pronunciada	Imaginada	Pronunciada	Imaginada	Pronunciada	Imaginada	Pronunciada	Imaginada	Pronunciada	
S01	51	11	51	10	51	7	51	9	51	7	49	9	357
S02	43	11	44	11	44	12	43	12	43	12	44	12	331
S03	46	13	47	14	48	13	45	14	47	14	45	14	360
S04	44	10	43	13	42	13	45	13	43	13	46	12	337
S05	41	12	39	8	41	12	42	9	40	11	39	12	306
S06	44	13	39	13	44	14	44	14	44	14	41	13	337
S07	41	12	41	14	43	11	43	13	41	13	43	12	327
S08	46	13	47	13	47	14	44	14	47	12	47	13	357
S09	46	12	45	12	46	12	45	12	43	12	45	12	342
S10	45	14	46	14	45	12	43	11	46	14	43	14	347
S11	47	12	47	12	45	12	47	12	46	12	46	12	350
S12	45	11	46	11	45	11	46	11	45	10	45	12	338
S13	43	13	43	14	44	14	45	14	46	14	43	14	347
S14	48	12	48	12	47	12	48	11	48	11	45	12	354
S15	45	11	42	12	43	11	41	12	44	11	40	11	323
Total	675	180	668	183	675	180	672	181	674	180	661	184	5113

Apéndice C

Aplicación ImaginedSpeech

En este apartado se incluyen los archivos fuente para la aplicación ImaginedSpeech diseñada para la plataforma BCI2000. Estos archivos se deben compilar en conjunto con la distribución estándar de BCI2000 y deben ser agregados al proyecto con el fin de permitir que sean reconocidos por CMake. Una vez realizado esto se debe hacer un build del proyecto completo. La dirección relativa de debería estar dada, a partir de la dirección raíz, de BCI2000 `src\shared\modules\application` y `src\custom\ImaginedSpeech`. Además es necesario permitir la instalación de la contribución MicRecorder, que se emplea para la grabación en simultaneo de la señal de audio. Cabe destacar que la versión de Qt instalada en la PC debe ser inferior a las 5.3 para que compile correctamente BCI2000, en este proyecto se uso la versión 4.8.5.

En los dos archivos, `ImaginedTask.cpp` y `ImaginedTask.h`, se define el orden de presentación de las diferentes etapas de la aplicación, así como los prototipos de las funciones. En los dos restantes, `ImaginedSpeechTask.cpp` y `ImaginedSpeechTask.h`, se redefinen las funciones virtuales establecidas en `ImaginedTask.h` para cumplir con los intervalos propuestos en el protocolo y se establecen las variables que el usuario puede modificar.

C.1. ImaginedTask.h

```
////////////////////////////////////
// $Id: ImaginedTask.h $
// Author: Pressel Germán
// Description: A base class for application modules that present a sequence
// of stimuli.
// Inheriting from ApplicationBase, descendants of ImaginedTask have access
// to the AppLog, AppLog.File and AppLog.Screen streams declared in
// ApplicationBase.
//
// This class performs sequencing, and dispatches GenericFilter::Process()
// calls to its virtual member functions, depending on its current state.
//
// Child classes (descendants) of ImaginedTask implement event
// handlers by overriding its virtual functions, and populate the stimulus
// map associating stimulus codes with sets of stimuli.
//
// Sequence of events          Typical application behavior
//
// OnPreflight
// OnInitialize
// OnStartRun                  display initial message
// DoPreRun*
// Loop {
//   OnNextStimulusCode        provide a stimulus code, or 0 to finish run
//   OnPreSequence             determine attended target
//   DoPreSequence*
//   OnSequenceBegin
//   Loop {
//     OnStimulusBegin          present stimulus
//     DoStimulus*
//     OnStimulusEnd            hide visual stimulus
//     OnNextStimulusCode       provide a stimulus code, or 0 to finish sequence
//                               sequence, and during the post sequence phase
//   }
//   OnSequenceEnd
//   DoPostSequence*
// }
// OnPostRun
// DoPostRun*
// OnStopRun                   display final message
// OnHalt
//
// Events marked with * may occur multiple times in a row.
// Progress from one state to the next will occur according to the sequencing
// parameters, or if requested by a handler via its doProgress output argument.
//
// $BEGIN_BCI2000_LICENSE$
//
// This file is part of BCI2000, a platform for real-time bio-signal research.
// [ Copyright (C) 2000-2012: BCI2000 team and many external contributors ]
//
```

```
// BCI2000 is free software: you can redistribute it and/or modify it under the
// terms of the GNU General Public License as published by the Free Software
// Foundation, either version 3 of the License, or (at your option) any later
// version.
//
// BCI2000 is distributed in the hope that it will be useful, but
// WITHOUT ANY WARRANTY
// - without even the implied warranty of MERCHANTABILITY or FITNESS FOR
// A PARTICULAR PURPOSE. See the GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along with
// this program. If not, see <http://www.gnu.org/licenses/>.
//
// $END_BCI2000_LICENSE$
////////////////////////////////////
#ifdef IMAGINED_TASK_H
#define IMAGINED_TASK_H

#include "ApplicationBase.h"
#include "ApplicationWindow.h"
#include "Association.h"
#include "TextField.h"

// Values of the PhaseInSequence state:
namespace PhaseInSequence
{
    enum
    {
        None = 0,
        PreSequence = 1,
        Sequence = 2,
        PostSequence = 3,
    };
}

class ImaginedTask : public ApplicationBase
{
private:
    enum TaskPhases
    {
        none,
        preRun,
        preSequence,
        stimulus,
        postSequence,
        postRun,
    };
protected:
    // Events to be handled by ImaginedTask descendants.
    // Events triggered by the GenericFilter interface
    virtual void OnPreflight( const SignalProperties& Input ) const {}
};
```

```
virtual void OnInitialize( const SignalProperties& Input ) {}
virtual void OnStartRun() {}
virtual void OnStopRun() {}
virtual void OnHalt() {}
// Events triggered during the course of a sequence
virtual void OnPreSequence() {}
virtual void OnSequenceBegin() {}
virtual void OnSequenceEnd() {}
virtual void OnStimulusBegin(){}
virtual void OnStimulusEnd(){}
virtual void OnPostSequenceBegin(){}
virtual void OnPostRun() {}
virtual void OnClassInput( int stimulusCode, const GenericSignal& ) {}

// Events
// Dispatching of the input signal.
// Each call to GenericSignal::Process() is dispatched to one of these
// events, depending on the phase in the sequence.
// There, each handler function corresponds to a phase.
// If a handler sets the "progress" argument to true, the application's
// state will switch to the next phase independently of the phases' pre-set
// durations.
virtual void DoPreRun(      const GenericSignal&, bool& doProgress ) {}
virtual void DoPreSequence( const GenericSignal&, bool& doProgress ) {}
virtual void DoStimulus(   const GenericSignal&, bool& doProgress ) {}

virtual void DoPostSequence( const GenericSignal&, bool& doProgress ) {}
virtual void DoPostRun(     const GenericSignal&, bool& doProgress ) {}
// A virtual interface for sequence access.
// An implementation of OnNextStimulusCode() should return the next element of
// the current sequence, or zero to indicate the end of the sequence.
// A null sequence indicates the end of a run, i.e. the current run will end when
// OnNextStimulusCode() returns two zeros in sequence.
virtual int OnNextStimulusCode() = 0;

protected:
// Access to the Associations property.
// An association map maps stimulus codes to sets of stimuli and sets of
// targets. Stimuli are used for presentation, while targets are used for
// classification, and selected according to the classification result.
AssociationMap& Associations()
{ return mAssociations; }
const AssociationMap& Associations() const
{ return mAssociations; }

// Access to the AttendedTarget property.
// The StimulusType state indicates whether the current stimulus is associated
// with the target given by the AttendedTarget property.
ImaginedTask& SetAttendedTarget( Target* inTarget )
{ mpAttendedTarget = inTarget; return *this; }
Target* AttendedTarget() const
{ return mpAttendedTarget; }
```

```
// Access to the Display property.
GUI::GraphDisplay& Display()
{ return mrDisplay; }
const GUI::GraphDisplay& Display() const
{ return mrDisplay; }

// User message display.
void DisplayMessage( const std::string& );

protected:
    ImaginedTask();

public:
    virtual ~ImaginedTask();

// Implementation of the GenericFilter interface.
// Normally, there should be no need to override these functions from
// a descendant class.
virtual void Preflight( const SignalProperties&, SignalProperties& ) const;
virtual void Initialize( const SignalProperties&, const SignalProperties& );
virtual void Process( const GenericSignal&, GenericSignal& );
virtual void StartRun();
virtual void StopRun();
virtual void Halt();

private:
    int mPhase,
        mBlocksInPhase,
        mStimulusCode;

    int mPreRunDuration,
        mPostRunDuration,
        mPreSequenceDuration,
        mPostSequenceDuration,
        mFocusOnDuration,
        mStimulusDuration,
        mImagineDuration,
        mRestDuration;

// Display elements.
ApplicationWindow& mrDisplay;
TextField* mpMessageField;

// Stimulus/Target configuration
AssociationMap mAssociations;

Target* mpAttendedTarget;
};

#endif // IMAGINED_TASK_H
```

C.2. ImaginedTask.cpp

```
////////////////////////////////////
// $Id: ImaginedTask.cpp $
// Author: Pressel Germán
// Description: A base class for application modules that provide feedback in a
// trial-based paradigm.
// Inheriting from ApplicationBase, descendants of ImaginedTask have access
// to the AppLog, AppLog.File and AppLog.Screen streams declared in
// ApplicationBase.
//
// This class performs sequencing, and dispatches GenericFilter::Process()
// calls to its virtual member functions, depending on its current state.
// Child classes (descendants) of ImaginedTask implement event
// handlers by overriding its virtual functions.
//
// $BEGIN_BCI2000_LICENSE$
//
// This file is part of BCI2000, a platform for real-time bio-signal research.
// [ Copyright (C) 2000-2012: BCI2000 team and many external contributors ]
//
// BCI2000 is free software: you can redistribute it and/or modify it under the
// terms of the GNU General Public License as published by the Free Software
// Foundation, either version 3 of the License, or (at your option) any later
// version.
//
// BCI2000 is distributed in the hope that it will be useful, but
// WITHOUT ANY WARRANTY
// - without even the implied warranty of MERCHANTABILITY or FITNESS FOR
// A PARTICULAR PURPOSE. See the GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along with
// this program. If not, see <http://www.gnu.org/licenses/>.
//
// $END_BCI2000_LICENSE$
////////////////////////////////////
#include "PCHIncludes.h"
#pragma hdrstop

#include "ImaginedTask.h"
#include "MeasurementUnits.h"
#include "BCIException.h"

#include <iomanip>
#include <set>
#include <limits>

using namespace std;

ImaginedTask::ImaginedTask()
: mPhase( none ),
  mBlocksInPhase( 0 ),
  mPreRunDuration( 0 ),
```

```

mPostRunDuration( 0 ),
mPreSequenceDuration( 0 ),
mPostSequenceDuration( 0 ),
mStimulusDuration( 0 ),
mImagineDuration( 0 ),
mRestDuration( 0 ),
mFocusOnDuration( 0 ),
mpMessageField( NULL ),
mpAttendedTarget( NULL ),
mrDisplay( Window() )
{
BEGIN_PARAMETER_DEFINITIONS
  "Application:Sequencing float PreRunDuration= 1s 1s 0 % "
  "  // pause preceding first sequence",
  "Application:Sequencing float PostRunDuration= 1s 1s 0 % "
  "  // pause following last sequence",
  "Application:Sequencing float PreSequenceDuration= 4s 4s 0 % "
  "  // pause preceding sequences/sets of intensifications",
  "Application:Sequencing float FocusOnDuration= 2s 4s 0 % "
  "  // time lapse for the FocusOn sign during presequence",
  "Application:Sequencing float ImagineDuration= 4s 4s 0 % "
  "  // time lapse for the subject to imagine",
  "Application:Sequencing float RestDuration= 3s 4s 0 % "
  "  // duration of the rest elements during post sequence",
  "Application:Sequencing float PostSequenceDuration= 4s 4s 0 % "
  "  // pause following sequences/sets of intensifications",
  "Application:Sequencing float StimulusDuration= 2s 2s 0 % "
  "  // stimulus duration",
  "Application:Rest int DisplayRest= 1 1 0 1 //"
  "  //Display rest (boolean)",
END_PARAMETER_DEFINITIONS

BEGIN_STATE_DEFINITIONS
  "StimulusCode  16 0 0 0",
  "StimulusBegin  1 0 0 0", // 1: first block of stimulus presentation
  "PhaseInSequence 2 0 0 0", // 1: pre-sequence, 2: during sequence, 3: post-sequence
  "PauseApplication 1 0 0 0",
END_STATE_DEFINITIONS
}

ImaginedTask::~ImaginedTask()
{
  Halt();
}

void
ImaginedTask::Preflight( const SignalProperties& Input, SignalProperties& Output ) const
{
  /* For parameters defining a time value, issue a warning if limited temporal
  resolution leads to a discrepancy greater than 1ms.*/
  const char* timeParams[] =
  {

```

```
"PreRunDuration",
"PostRunDuration",
"PreSequenceDuration",
"PostSequenceDuration",
"StimulusDuration",
"FocusOnDuration",
"ImagineDuration",
"RestDuration",
};

float oneMillisecond = MeasurementUnits::TimeInSampleBlocks( "1ms" );
for( size_t i = 0; i < sizeof( timeParams ) / sizeof( *timeParams ); ++i )
{
    for( int j = 0; j < Parameter( timeParams[ i ] )->NumValues(); ++j )
    {
        float value = Parameter( timeParams[ i ] )( j ).InSampleBlocks();
        if( ( value > numeric_limits<float>::epsilon() && value < 1.0f ) || ::fmod( value,
1.0f ) > oneMillisecond )
            bciout << "Due to a sample block duration of "
                << 1.0f / oneMillisecond << "ms,"
                << " the actual value of " << timeParams[ i ]
                << " for stimulus " << j + 1 << " will be "
                << ::floor( value ) / oneMillisecond << "ms"
                << " rather than "
                << value / oneMillisecond << "ms"
                << endl;
    }
}

float preRunDuration = Parameter( "PreRunDuration" ).InSampleBlocks();
if( preRunDuration < 1 )
    bcierr << "PreRunDuration must be >= 1 data block" << endl;

/*If Presequence Duration is less than the sum of FocusOn Duration and Stimulus Duration
the value of Presequence is changed to the minimum valid value */
int mMinPreSequenceDuration= Parameter("FocusOnDuration") + Parameter("StimulusDuration");
if(Parameter("PreSequenceDuration") < mMinPreSequenceDuration)
{
    Parameter("PreSequenceDuration") = mMinPreSequenceDuration;
    bciout<<"Value of PreSequenceDuration was changed to "
        <<Parameter("PreSequenceDuration")<<"due to it was"
        <<"inconsistent"
        <<endl;
}

/*If PostSequence Duration is less than Rest Duration
the value of Postsequence is changed to the minimum valid value */
int RestDuration= Parameter("RestDuration");
if(Parameter("PostSequenceDuration") < RestDuration)
{
    Parameter("PostSequenceDuration") = RestDuration;
}
```

```

    bciout<<"Value of PostSequenceDuration was changed to "
        <<Parameter("PostSequenceDuration")<<"due to it was"
        <<"inconsistent"
        <<endl;
}

State( "Running" );

bcidbg( 2 ) << "Event: Preflight" << endl;
/*Calls OnPreflight in ImaginedSpeechTask*/
OnPreflight( Input );
Output = Input;
}

void
ImaginedTask::Initialize( const SignalProperties& Input,
                        const SignalProperties& Output )
{
    ApplicationBase::Initialize( Input, Output );

    mPreRunDuration = static_cast<int>( Parameter( "PreRunDuration" ).InSampleBlocks() );
    mPostRunDuration = static_cast<int>( Parameter( "PostRunDuration" ).InSampleBlocks() );
    mPreSequenceDuration = static_cast<int>( Parameter( "PreSequenceDuration" ).InSampleBlocks() );
    mPostSequenceDuration = static_cast<int>( Parameter
    ( "PostSequenceDuration" ).InSampleBlocks() );
    mStimulusDuration = static_cast<int>( Parameter( "StimulusDuration" ).InSampleBlocks() );
    mFocusOnDuration = static_cast<int>( Parameter( "FocusOnDuration" ).InSampleBlocks() );
    mImagineDuration = static_cast<int>( Parameter( "ImagineDuration" ).InSampleBlocks() );
    mRestDuration = static_cast<int>( Parameter( "RestDuration" ).InSampleBlocks() );

    bciidbg( 2 ) << "Event: Initialize" << endl;
    OnInitialize( Input ); /*Calls OnInitialize in ImaginedSpeechTask*/
}

void
ImaginedTask::StartRun()
{
    mBlocksInPhase = 1;

    /*Setting states occurs _before_ triggering events.*/
    State( "PhaseInSequence" ) = PhaseInSequence::None;
    State( "StimulusCode" ) = 0;
    State( "StimulusBegin" ) = 0;

    bciidbg( 2 ) << "Event: StartRun" << endl;
    OnStartRun(); /*Calls OnStartRun() in ImaginedSpeechTask*/

    mStimulusCode = 0;
    mPhase = preRun;
}

```

```
void
ImaginedTask::StopRun()
{
    bcidbg( 2 ) << "Event: StimulusEnd" << endl;
    OnStimulusEnd();
    mPhase = none;

    State( "PhaseInSequence" ) = PhaseInSequence::None;
    State( "StimulusCode" ) = 0;
    State( "StimulusBegin" ) = 0;

    bcidbg( 2 ) << "Event: StopRun" << endl;
    OnStopRun(); /*Calls OnStopRun in ImaginedSpeechTask*/
}

void
ImaginedTask::Halt()
{
    bcidbg( 2 ) << "Event: Halt" << endl;
    OnHalt(); /*Calls OnHalt in ImaginedSpeechTask*/
}

void
ImaginedTask::Process( const GenericSignal& Input, GenericSignal& Output )
{
    if( State( "PauseApplication" ) )
    {
        Output = Input;
        return;
    }

    bool doProgress = true;
    while( doProgress )
    {
        switch( mPhase )
        {
            case preRun:
                doProgress = ( mBlocksInPhase >= mPreRunDuration );
                DoPreRun( Input, doProgress );
                break;

            case preSequence:
                doProgress = ( mBlocksInPhase >= mPreSequenceDuration );
                DoPreSequence( Input, doProgress );
                break;

            case stimulus:
                {
                    doProgress = ( mBlocksInPhase >= mImagineDuration );
                    State( "StimulusBegin" ) = ( mBlocksInPhase == 0 && !doProgress );
                }
            }
        }
    }
}
```

```

        DoStimulus( Input, doProgress );
    } break;

    case postSequence:
        doProgress = ( mBlocksInPhase >= mPostSequenceDuration );
        DoPostSequence( Input, doProgress );
        break;

    case postRun:
        doProgress = ( mBlocksInPhase >= mPostRunDuration );
        break;

    case none:
        doProgress = false;
        break;

    default:
        throw bciexception( "Unknown phase value: " << mPhase );
}
if( doProgress )
{
    mBlocksInPhase = 0;
    switch( mPhase )
    {
        case postSequence:
            { /* Leave post sequence phase*/
                bcidbg( 2 ) << "Event: SequenceEnd" << endl;
                OnSequenceEnd();
            }
            /* fall through */
        case preRun:
            mStimulusCode = OnNextStimulusCode();
            if( mStimulusCode > 0 )
            { /* Enter pre sequence phase */
                State( "PhaseInSequence" ) = PhaseInSequence::PreSequence;
                bcidbg( 2 ) << "Event: PreSequence" << endl;
                bciout<<"Event: PreSequence"<<endl;
                OnPreSequence();
                mPhase = preSequence;
            }
            else
            { /* Enter post run phase*/
                State( "PhaseInSequence" ) = PhaseInSequence::None;
                bcidbg( 2 ) << "Event: PostRun" << endl;
                bciout<<"Event: PostRun"<<endl;
                OnPostRun();
                mPhase = postRun;
            }
        }
    } break;
}

```

```
case preSequence:
{ /* Leave pre sequence phase*/
    State( "PhaseInSequence" ) = PhaseInSequence::Sequence;
    bcidbg( 2 ) << "Event: SequenceBegin" << endl;
    bciout<<"Event: SequenceBegin"<<endl;
    OnSequenceBegin();

    if( mStimulusCode > 0 )
    { OnStimulusBegin();
      mPhase = stimulus;
    }
    else
    { /* Enter post sequence phase*/
      State( "PhaseInSequence" ) = PhaseInSequence::PostSequence;
      bciout<<"Event: PostSequence"<<endl;
      mPhase = postSequence;
    }
}
break;

case stimulus:
{ /* Enter PostSequence*/
    State( "StimulusCode" ) = 0;
    bciout << "Event: StimulusEnd" << endl;
    OnStimulusEnd();
    OnPostSequenceBegin();
    State( "PhaseInSequence" ) = PhaseInSequence::PostSequence;
    bciout<<"Event: PostSequence"<<endl;
    mPhase = postSequence;
} break;

case postRun:
    State( "Running" ) = 0;
    bciout<<"Event: None"<<endl;
    mPhase = none;
    break;

default:
    throw bciexception( "Unknown phase value: " << mPhase );
}
}
}
Output = Input;
++mBlocksInPhase;
}

void
ImaginedTask::DisplayMessage( const string& inMessage )
{
    delete mpMessageField;
    mpMessageField = NULL;
}
```

```
if( !inMessage.empty() )
{
    mpMessageField = new TextField( mrDisplay );
    GUI::Rect rect = { 0.5f, 0.4f, 0.5f, 0.6f };
    mpMessageField->SetText( inMessage )
        .SetTextHeight( 0.8f )
        .SetColor( RGBColor::Gray )
        .SetTextColor( RGBColor::Yellow )
        .SetAspectRatioMode( GUI::AspectRatioModes::AdjustWidth )
        .SetObjectRect( rect );
    while( mpMessageField->BoundingRect().right - mpMessageField->BoundingRect().left > 1 )
        mpMessageField->SetTextHeight( mpMessageField->TextHeight() * 0.9 );
}
}
```

C.3. ImaginedSpeechTask.h

```
// Description: ImaginedSpeechTask header
// Author:German Pressel
//
// $BEGIN_BCI2000_LICENSE$
//
// This file is part of BCI2000, a platform for real-time bio-signal research.
// [ Copyright (C) 2000-2012: BCI2000 team and many external contributors ]
//
// BCI2000 is free software: you can redistribute it and/or modify it under the
// terms of the GNU General Public License as published by the Free Software
// Foundation, either version 3 of the License, or (at your option) any later
// version.
//
// BCI2000 is distributed in the hope that it will be useful, but
//
// WITHOUT ANY WARRANTY
// - without even the implied warranty of MERCHANTABILITY or FITNESS FOR
// A PARTICULAR PURPOSE. See the GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along with
// this program. If not, see <http://www.gnu.org/licenses/>.
//
// $END_BCI2000_LICENSE$
////////////////////////////////////

#ifdef INCLUDED_ImaginedSpeechTask_H // makes sure this header is not included more than
once
#define INCLUDED_ImaginedSpeechTask_H

#include "ApplicationBase.h"
#include "ImaginedTask.h"
#include <vector>

namespace SequenceTypes
{
    enum
    {
        Deterministic = 1,
        Random,
    };
}

class ImaginedSpeechTask : public ImaginedTask
{
public:
    ImaginedSpeechTask();
    virtual ~ImaginedSpeechTask();
protected:
    virtual void OnPreflight( const SignalProperties& Input ) const;
    virtual void OnInitialize( const SignalProperties& Input );
    virtual void OnStartRun();
    virtual void OnStopRun();
};
```

```

virtual void    OnPreSequence();
virtual void    OnSequenceBegin();
virtual void    OnStimulusBegin();
virtual void    OnStimulusEnd();
virtual void    OnPostSequenceBegin();
virtual void    OnPostRun();
virtual int     OnNextStimulusCode();

virtual void    DoPreSequence( const GenericSignal&, bool& doProgress );
virtual void    DoPostSequence( const GenericSignal&, bool& doProgress );

private:
    ParamRef StimulusProperty( const ParamRef& inMatrixParam,
                               int inColIndex,
                               const std::string& inPropertyName ) const;

    // Configuration parameters.
    int mNumberOfSequences,
        mSequenceType;

    // Internal state.
    int mPreSequenceBlockCount,
        mPostRestBlockCount,
        mSequenceCount;

    std::vector<int> mToBeCopied;
    std::vector<int>::const_iterator mToBeCopiedPos;

    std::vector<int> mSequence;
    std::vector<int>::const_iterator mSequencePos;

    // Sets of stimuli
    SetOfStimuli mStimuli;
    Association mFocusAnnouncement,
               mImagineAnnouncement,
               mRestAnnouncement;
};

#endif // INCLUDED_ImaginedSpeechTask_H

```

C.4. ImaginedSpeechTask.cpp

```
//ImaginedSpeechTask.cpp
//Created by German Pressel
//$BEGIN_BCI2000_LICENSE$
//
// This file is part of BCI2000, a platform for real-time bio-signal research.
// [ Copyright (C) 2000-2012: BCI2000 team and many external contributors ]
//
// BCI2000 is free software: you can redistribute it and/or modify it under the
// terms of the GNU General Public License as published by the Free Software
// Foundation, either version 3 of the License, or (at your option) any later
// version.
//
// BCI2000 is distributed in the hope that it will be useful, but
// WITHOUT ANY WARRANTY
// - without even the implied warranty of MERCHANTABILITY or FITNESS FOR
// A PARTICULAR PURPOSE. See the GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along with
// this program. If not, see <http://www.gnu.org/licenses/>.
//
// $END_BCI2000_LICENSE$
////////////////////////////////////
#include "PCHIncludes.h"
#pragma hdrstop

#include "ImaginedSpeechTask.h"

#include "TextStimulus.h"
#include "ImageStimulus.h"
#include "AudioStimulus.h"
#include "Localization.h"

#include <algorithm>

using namespace std;

RegisterFilter( ImaginedSpeechTask, 3 );

/* Constructor for the ImaginedSpeechTask class*/
ImaginedSpeechTask::ImaginedSpeechTask()
: mNumberOfSequences( 0 ),
  mSequenceType( SequenceTypes::Deterministic ),
  mPreSequenceBlockCount( 0 ),
  mPostRestBlockCount( 0 ),
  mSequenceCount( 0 ),
  mToBeCopiedPos( mToBeCopied.begin() ),
  mSequencePos( mSequence.begin() )
{
  /* Definition of parameters for the application module */
  BEGIN_PARAMETER_DEFINITIONS
```

```

"Application:Sequencing int SequenceType= 1 1 1 2 // "
"Sequence of stimuli: 1 Deterministic, 2 Random(enumeration)",

"Application:Sequencing intlist Sequence= 4 1 3 4 2 % % % // "
"Deterministic Mode :Sequence in which stimuli are presented",

"Application:Sequencing int NumberOfSequences= 3 1 0 % // "
"Deterministic Mode :Number of sequence repetitions",

"Application:Sequencing intlist Frequency= 3 1 2 3 1 1 % // "
"Random Mode: Frequence of stimuli that must be copied",

"Application:Sequencing int PronounceSwitch= 1 1 0 1 "
"// Present the caption that says pronounce if true, else present imagine (boolean)",

```

END_PARAMETER_DEFINITIONS

/* Definition of the matrixes that contain the stimulus */

BEGIN_PARAMETER_DEFINITIONS

```

"Application:Stimuli matrix Stimuli= "
"{ caption icon audio } " // row labels
"{ stimulus1 stimulus2 stimulus3 stimulus4 stimulus5 } " // column labels
" A E I O U "
"images\\A.bmp images\\E.bmp images\\I.bmp images\\O.bmp images\\U.bmp "
"sounds\\A.wav sounds\\E.wav sounds\\I.wav sounds\\O.wav sounds\\U.wav "
" // captions and icons to be displayed, sounds to be played for different stimuli",

"Application:Stimuli matrix FocusOn= "
"{ caption icon audio } " // row labels
"{ focuson } " // column labels
"Please%20imagine " // caption
"images\\focuson.bmp " // video
"sounds\\uh-uh.wav " // audio
" // initial announcement what to focus on",

"Application:Stimuli matrix Imagine= "
"{ caption icon audio } " // row labels
"{ Imagined Pronounce} " // column labels
" Imagine Pronounce " // caption
"images\\imagine.bmp images\\pronuncie.bmp " // video
"sounds\\imagine.wav sounds\\imagine.wav " // audio
" // imagine/pronounce announcement ",

"Application:Stimuli matrix Rest= "
"{ caption icon audio } " // row labels
"{ Rest } " // column labels
"Please%20rest " // caption
"images\\rest.bmp " // video
"sounds\\imagine.wav " // audio
" // final announcement to rest",

```

```

END_PARAMETER_DEFINITIONS

/* Definition of the options for the application modules */
BEGIN_PARAMETER_DEFINITIONS
  "Application:Stimuli int StimulusWidth= 50 0 0 100 // "
  "StimulusWidth in percent of screen width (zero for original pixel size)",
  "Application:Stimuli int CaptionHeight= 10 0 0 100 // "
  "Height of stimulus caption text in percent of screen height",
  "Application:Stimuli string CaptionColor= 0x00FFFFFF 0x00FFFFFF 0x00000000 0xFFFFFFFF // "
  "Color of stimulus caption text (color)",
  "Application:Stimuli string BackgroundColor= 0x00FFFF00 0x00FFFF00 0x00000000 0xFFFFFFFF // "
  "Color of stimulus background (color)",
  "Application:Stimuli int CaptionSwitch= 1 1 0 1 // "
  "Present captions (boolean)",
  "Application:Stimuli int IconSwitch= 1 1 0 1 // "
  "Present icon files (boolean)",
  "Application:Stimuli int AudioSwitch= 1 1 0 1 // "
  "Present audio files (boolean)",
  "Application:Stimuli float AudioVolume= 100 100 0 100 // "
  "Volume for audio playback in percent",
END_PARAMETER_DEFINITIONS

/* Languages definition */
LANGUAGES "German", "Spanish",
BEGIN_LOCALIZED_STRINGS
  "TIME OUT !!!",
  "Zeit abgelaufen!",
  "SE TERMINO EL TIEMPO",
  "Waiting to start ...",
  "Warte ...",
  "Esperando para comenzar",
END_LOCALIZED_STRINGS

BEGIN_STATE_DEFINITIONS
  "InitImagine 2 0 0 0",
  "MicSampleImagine 32 0 0 0",
END_STATE_DEFINITIONS
}
/* Destructor of the ImaginedSpeechTask class*/
ImaginedSpeechTask::~ImaginedSpeechTask()
{
  mStimuli.DeleteObjects();
}

/* OnPreflight: Check the parameters against the limits of each characteristic*/
void
ImaginedSpeechTask::OnPreflight( const SignalProperties& /*Input*/ ) const
{
  ParamRef Sequence = Parameter( "Sequence" );
  ParamRef Stimuli = Parameter( "Stimuli" );

```

```

/* Switch between the different sequence types */
switch( int( Parameter( "SequenceType" ) ) )
{
    case SequenceTypes::Deterministic:

        bciout<<"NumColumnsStimulus"<<Stimuli->NumColumns()<<endl;
        for( int i = 0; i < Sequence->NumValues(); ++i )
        { /* Checks if the number in the sequence list exceeds the number
           of Stimulis in the Stimuli matrix*/
            if ( Sequence( i )> Stimuli->NumColumns() ){
                bcierr<< "Invalid stimulus code,"
                    << "exceeds the number of stimulus"
                    << endl;
            }
            /* Checks if the number in the sequence list is greater than zero */
            if( Sequence( i ) < 1 ){
                bcierr << "Invalid stimulus code "
                    << "(" << Sequence( i ) << " ) "
                    << "at Sequence(" << i << " )"
                    << endl;
            }
        }
        break;

    case SequenceTypes::Random:
        for( int i = 0; i < Sequence->NumValues(); ++i )
        { /* Checks if the number in the sequence list exceeds the number
           of Stimulis in the Stimuli matrix*/
            if ( Sequence( i )> Stimuli->NumColumns() ){
                bcierr<< "Invalid stimulus code,"
                    << "exceeds the number of stimulus"
                    << endl;
            }
            /* Checks if the frequency of a stimuli is less than zero */
            if( Sequence( i ) < 0 ){
                bcierr << "Invalid frequency "
                    << "(" << Sequence( i ) << " ) "
                    << "at Sequence(" << i << " )"
                    << endl;
            }
        }
        break;

    default:
        bcierr << "Unknown value in SequenceType parameter"
            << endl;
}
/* Generates an object of every type of stimulus */
GUI::GraphDisplay preflightDisplay;
ImageStimulus* pImageStimulus = NULL;
if( Parameter( "IconSwitch" ) == 1 )

```

```

pImageStimulus = new ImageStimulus( preflightDisplay );

AudioStimulus* pAudioStimulus = NULL;
if( Parameter( "AudioSwitch" ) == 1 )
  pAudioStimulus = new AudioStimulus;

/*Check states SourceTime and MicSample*/
int sourcetime=State("SourceTime");
int micsample=State("MicSample");
int micsampleimagine=State("MicSampleImagine");
/* Initialize the variables for FocusOn and Rest */
bool presentFocusOn = false,
     presentRest = false;
/* Sets the FocusOn and asks if the DisplayRest is true */
presentFocusOn = true;
presentRest = ( Parameter( "DisplayRest" ) == 1 );

/* Defines the time settings and presents them in the BCI shell*/
int focusDuration = static_cast<int>(Parameter( "FocusOnDuration" ).InSampleBlocks() ), //
Toma los valores de duracion de focuson y result
restDuration = static_cast<int>( Parameter( "RestDuration" ).InSampleBlocks() ), //
los convierte a partir de la tasa de adquisicion
imagineDuration=static_cast<int>(Parameter("ImagineDuration").InSampleBlocks()),
stimulusDuration=static_cast<int>(Parameter("StimulusDuration").InSampleBlocks()),
preSequenceDuration = static_cast<int>( Parameter
( "PreSequenceDuration" ).InSampleBlocks() ), //Lo mismo hace con Pre y
PostStimulusDuration
postSequenceDuration = static_cast<int>( Parameter
( "PostSequenceDuration" ).InSampleBlocks() ); // con la diferencia que solo tiene
que acceder al parametro dado que es el unico valor que posee.

bciout<< " Focus Duration" <<focusDuration<<endl
        << " Rest Duration" <<restDuration<<endl
        << " Imagine Duration" <<imagineDuration<<endl
        << " Stimulus Duration" <<stimulusDuration
        << endl;

/* Checks for errors in the time settings */
if( presentFocusOn && preSequenceDuration < focusDuration + stimulusDuration)
  bcierr << "When FocusOn message and target stimulus are presented, "
        << "PreSequenceDuration must at least be equal to the FocusDuration "
        << "applying for the FocusOn message plus the StimulusDuration"
        << endl;
if( presentRest && postSequenceDuration < restDuration )
  bcierr << "When Result message and result stimulus are presented, "
        << "PostSequenceDuration must at least be the RestDuration "
        << "applying for the Rest message"
        << endl;

/* Generates a vector with all the matrix elements
to compare then to the minimum requirement of each.*/

```

```

vector<string> stimParams;
stimParams.push_back( "Stimuli" );
stimParams.push_back("Imagine");
if( presentFocusOn )
    stimParams.push_back( "FocusOn" );
if( presentRest )
    stimParams.push_back( "Rest" );

/* For parameters defining a time value, issue a warning if limited temporal
   resolution leads to a discrepancy greater than 1ms. */
const char* timeParams[] =
{
    "StimulusDuration",
};
double oneMillisecond = MeasurementUnits::TimeInSampleBlocks( "1ms" );
int minStimDuration = 0;

enum { caption, icon, audio };
for( size_t i = 0; i < stimParams.size(); ++i )
    for( int j = 0; j < Parameter( stimParams[ i ] )->NumColumns(); ++j )
        {
            int stimDuration = static_cast<int>( StimulusProperty( Parameter( stimParams[ i ] ),
            j, "StimulusDuration" ).InSampleBlocks() );
            if( minStimDuration > stimDuration )
                minStimDuration = stimDuration;
            /*Check individual stimulus duration*/
            for( size_t k = 0; k < sizeof( timeParams ) / sizeof( *timeParams ); ++k )
                {
                    double value = StimulusProperty( Parameter( stimParams[ i ] ), j, timeParams
                    [ k ] ).InSampleBlocks();
                    if( value > 0 && value < 1.0 || ::fmod( value, 1.0 ) > oneMillisecond )
                        bciout << "Due to a sample block duration of "
                        << 1.0f / oneMillisecond << "ms,"
                        << " the actual value of " << timeParams[ k ]
                        << " for stimulus " << j + 1 << " will be "
                        << ::floor( value ) / oneMillisecond << "ms"
                        << " rather than "
                        << value / oneMillisecond << "ms"
                        << endl;
                }
            /* Test availability of icon and audio files. */
            if( pImageStimulus != NULL )
                {
                    string file = Parameter( stimParams[ i ] )( icon, j );
                    if( !file.empty() )
                        pImageStimulus->SetFile( file );
                }
            if( pAudioStimulus != NULL )
                {
                    string file = Parameter( stimParams[ i ] )( audio, j );
                    if( !file.empty() )
                        {

```



```
        pAudioStimulus->SetSound( file );
        if( !pAudioStimulus->Error().empty() )
            bcierr << pAudioStimulus->Error() << endl;
    }
}
}

delete pImageStimulus;
delete pAudioStimulus;

}

void
ImaginedSpeechTask::OnInitialize( const SignalProperties& /*Input*/ )
{
    /*Read sequence parameters.*/
    mNumberOfSequences = Parameter( "NumberOfSequences" );
    mSequenceType = Parameter( "SequenceType" ); /*Deterministic or random*/

    bool presentFocusOn = true,
         presentResult = true;

    /*Configure Stimulus stimuli*/
    mStimuli.DeleteObjects();
    Associations().clear();

    /*Definition of variables and asignment of the list of stimulus in the matrix Stimulus*/
    bool captionSwitch = ( Parameter( "CaptionSwitch" ) == 1 ),
         iconSwitch = ( Parameter( "IconSwitch" ) == 1 ),
         audioSwitch = ( Parameter( "AudioSwitch" ) == 1 );

    RGBColor backgroundColor = RGBColor::NullColor;
    if( !iconSwitch )
        backgroundColor = RGBColor( Parameter( "BackgroundColor" ) );

    enum { caption, icon, audio };
    ParamRef Stimuli = Parameter( "Stimuli" );
    for( int i = 0; i < Stimuli->NumColumns(); ++i )
    {
        Associations()[ i + 1 ].SetStimulusDuration( static_cast<int>( StimulusProperty
            ( Stimuli, i, "StimulusDuration" ).InSampleBlocks() ) );

        double stimulusWidth = StimulusProperty( Stimuli, 1, "StimulusWidth" ) / 100.0,
             captionHeight = StimulusProperty( Stimuli, 1, "CaptionHeight" ) / 100.0,
             audioVolume = StimulusProperty( Stimuli, 1, "AudioVolume" ) / 100.0;
        RGBColor captionColor = RGBColor( StimulusProperty( Stimuli, 1, "CaptionColor" ) );

        /*If CaptionSwitch is active, define a rectangle and assigned the frase in the caption
        to the object StimuliAssociation*/
        if( captionSwitch )
        {
```

?

```
GUI::Rect captionRect =
{
    0.5, ( 1 - captionHeight ) / 2,
    0.5, ( 1 + captionHeight ) / 2
};
TextStimulus* pStimulus = new TextStimulus( Display() );
pStimulus->SetText( Stimuli( caption, i ) )
    .SetTextHeight( 1.0 )
    .SetTextColor( captionColor )
    .SetColor( backgroundColor )
    .SetAspectRatioMode( GUI::AspectRatioModes::AdjustWidth )
    .SetObjectRect( captionRect );
pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
mStimuli.Add( pStimulus );
Associations()[ i + 1 ].Add( pStimulus );
}

/*If IconSwitch is active, define a rectangle, adjust the aspect ratio and assigned
the icon in that column to the object StimuliAssociation*/
if( iconSwitch && Stimuli( icon, i ) != "" )
{
    int iconSizeMode = GUI::AspectRatioModes::AdjustHeight;
    if( stimulusWidth <= 0 )
        iconSizeMode = GUI::AspectRatioModes::AdjustBoth;
    GUI::Rect iconRect =
    {
        ( 1 - stimulusWidth ) / 2, 0.5,
        ( 1 + stimulusWidth ) / 2, 0.5
    };
    ImageStimulus* pStimulus = new ImageStimulus( Display() );
    pStimulus->SetFile( Stimuli( icon, i ) )
        .SetRenderingMode( GUI::RenderingMode::Opaque )
        .SetAspectRatioMode( iconSizeMode )
        .SetObjectRect( iconRect );
    pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
    mStimuli.Add( pStimulus );
    Associations()[ i + 1 ].Add( pStimulus );
}

/*If AudioSwitch is active, define a new audioStimulus and assigned
the sound to the object StimuliAssociation*/
if( audioSwitch && Stimuli( audio, i ) != "" )
{
    AudioStimulus* pStimulus = new AudioStimulus;
    pStimulus->SetSound( Stimuli( audio, i ) )
        .SetVolume( audioVolume );
    mStimuli.Add( pStimulus );
    Associations()[ i + 1 ].Add( pStimulus );
}
}

/*Configure FocusOn stimuli*/
```

```

mFocusAnnouncement.DeleteObjects();

/*Definition of variables and asignment of the list of stimulus in the matrix FocusOn*/
ParamRef FocusOn = Parameter( "FocusOn" );
mFocusAnnouncement.SetStimulusDuration( static_cast<int>(Parameter
("FocusOnDuration").InSampleBlocks() ) );

for( int i = 0; i < FocusOn->NumColumns(); ++i )
{
    float stimulusWidth = StimulusProperty( FocusOn, i, "StimulusWidth" ) / 100.0,
          captionHeight = StimulusProperty( FocusOn, i, "CaptionHeight" ) / 100.0,
          audioVolume = StimulusProperty( FocusOn, i, "AudioVolume" ) / 100.0;
    RGBColor captionColor = RGBColor( StimulusProperty( FocusOn, i, "CaptionColor" ) );

    /*If CaptionSwitch is active, define a rectangle and assigned the frase in the caption
    to the object FocusOnAnnouncement*/
    if( captionSwitch )
    {
        GUI::Rect captionRect =
        {
            0.5, ( 1 - captionHeight ) / 2,
            0.5, ( 1 + captionHeight ) / 2
        };
        TextStimulus* pStimulus = new TextStimulus( Display() );
        pStimulus->SetText( FocusOn( caption, i ) )
            .SetTextHeight( 1.0 )
            .SetTextColor( captionColor )
            .SetColor( backgroundColor )
            .SetAspectRatioMode( GUI::AspectRatioModes::AdjustWidth )
            .SetObjectRect( captionRect );
        pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
        mFocusAnnouncement.Add( pStimulus );
    }

    /*If IconSwitch is active, define a rectangle, adjust the aspect ratio and assigned
    the icon in that column to the object FocusOnAnnouncement*/
    if( iconSwitch && FocusOn( icon, i ) != "" )
    {
        int iconSizeMode = GUI::AspectRatioModes::AdjustHeight;
        if( stimulusWidth <= 0 )
            iconSizeMode = GUI::AspectRatioModes::AdjustBoth;
        GUI::Rect iconRect =
        {
            ( 1 - stimulusWidth ) / 2, 0.5,
            ( 1 + stimulusWidth ) / 2, 0.5
        };
        ImageStimulus* pStimulus = new ImageStimulus( Display() );
        pStimulus->SetFile( FocusOn( icon, i ) )
            .SetRenderingMode( GUI::RenderingMode::Opaque )
            .SetAspectRatioMode( iconSizeMode )
            .SetObjectRect( iconRect );
        pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
    }
}

```

```

        mFocusAnnouncement.Add( pStimulus );
    }
    /*If AudioSwitch is active, define a new audioStimulus and assigned
    the sound to the object FocusOnAnnouncement*/
    if( audioSwitch && FocusOn( audio, i ) != "" )
    {
        AudioStimulus* pStimulus = new AudioStimulus;
        pStimulus->SetSound( FocusOn( audio, i ) )
            .SetVolume( audioVolume );
        mFocusAnnouncement.Add( pStimulus ); //Lo pone en la cola.
    }
}

/*Configure Imagine stimuli*/
mImagineAnnouncement.DeleteObjects();

/*Definition of variables and asignment of the list of stimulus in the matrix Imagine*/
ParamRef Imagine = Parameter( "Imagine" );
int PronounceSwitch= Parameter("PronounceSwitch");
mImagineAnnouncement.SetStimulusDuration( static_cast<int>( Parameter
("ImagineDuration") .InSampleBlocks() ) );

float stimulusWidth = StimulusProperty( Imagine, PronounceSwitch , "StimulusWidth" ) /
100.0,
    captionHeight = StimulusProperty( Imagine, PronounceSwitch, "CaptionHeight" ) /
100.0,
    audioVolume = StimulusProperty( Imagine, PronounceSwitch, "AudioVolume" ) /
100.0;
RGBColor captionColor = RGBColor( StimulusProperty( Imagine, PronounceSwitch,
"CaptionColor" ) );

/*If CaptionSwitch is active, define a rectangle and assigned the frase in the caption
to the object ImagineAnnouncement*/
if( captionSwitch )
{
    GUI::Rect captionRect =
    {
        0.5, ( 1 - captionHeight ) / 2,
        0.5, ( 1 + captionHeight ) / 2
    };
    TextStimulus* pStimulus = new TextStimulus( Display() );
    pStimulus->SetText( Imagine( caption, PronounceSwitch ) )
        .SetTextHeight( 1.0 )
        .SetTextColor( captionColor )
        .SetColor( backgroundColor )
        .SetAspectRatioMode( GUI::AspectRatioModes::AdjustWidth )
        .SetObjectRect( captionRect );
    pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
    mImagineAnnouncement.Add( pStimulus );
}

```

```
/*If IconSwitch is active, define a rectangle, adjust the aspect ratio and assigned
the icon in that column to the object ImagineAnnouncement*/
if( iconSwitch && Imagine( icon, PronounceSwitch ) != "" )
{
    int iconSizeMode = GUI::AspectRatioModes::AdjustHeight;
    if( stimulusWidth <= 0 )
        iconSizeMode = GUI::AspectRatioModes::AdjustBoth;
    GUI::Rect iconRect =
    {
        ( 1 - stimulusWidth ) / 2, 0.5,
        ( 1 + stimulusWidth ) / 2, 0.5
    };
    ImageStimulus* pStimulus = new ImageStimulus( Display() );
    pStimulus->SetFile( Imagine( icon, PronounceSwitch ) )
        .SetRenderingMode( GUI::RenderingMode::Opaque )
        .SetAspectRatioMode( iconSizeMode )
        .SetObjectRect( iconRect );
    pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
    mImagineAnnouncement.Add( pStimulus );
}

/*If AudioSwitch is active, define a new audioStimulus and assigned
the sound to the object ImagineAnnouncement*/
if( audioSwitch && Imagine( audio, PronounceSwitch ) != "" )
{
    AudioStimulus* pStimulus = new AudioStimulus;
    pStimulus->SetSound( Imagine( audio, PronounceSwitch ) )
        .SetVolume( audioVolume );
    mImagineAnnouncement.Add( pStimulus );
}

/*Configure rest stimuli*/
mRestAnnouncement.DeleteObjects();

/*Definition of variables and assignment of the list of stimulus in the matrix Rest*/
ParamRef Rest = Parameter( "Rest" );
mRestAnnouncement.SetStimulusDuration( static_cast<int>( Parameter
( "RestDuration" ).InSampleBlocks() ) );
for( int i = 0; i < Rest->NumColumns(); ++i )
{
    float stimulusWidth = StimulusProperty( Rest, i, "StimulusWidth" ) / 100.0,
        captionHeight = StimulusProperty( Rest, i, "CaptionHeight" ) / 100.0,
        audioVolume = StimulusProperty( Rest, i, "AudioVolume" ) / 100.0;
    RGBColor captionColor = RGBColor( StimulusProperty( Rest, i, "CaptionColor" ) );

    /*If CaptionSwitch is active, define a rectangle and assigned the frase in the caption
to the object RestAnnouncement*/
    if( captionSwitch )
    {
        GUI::Rect captionRect =
        {
            0.5, ( 1 - captionHeight ) / 2,
```

```
        0.5, ( 1 + captionHeight ) / 2
    };
    TextStimulus* pStimulus = new TextStimulus( Display() );
    pStimulus->SetText( Rest( caption, i ) )
        .SetTextHeight( 1.0 )
        .SetTextColor( captionColor )
        .SetColor( backgroundColor )
        .SetAspectRatioMode( GUI::AspectRatioModes::AdjustWidth )
        .SetObjectRect( captionRect );
    pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
    mRestAnnouncement.Add( pStimulus );
}

/*If IconSwitch is active, define a rectangle, adjust the aspect ratio and assigned
the icon in that column to the object RestAnnouncement*/
if( iconSwitch && Rest( icon, i ) != "" )
{
    int iconSizeMode = GUI::AspectRatioModes::AdjustHeight;
    if( stimulusWidth <= 0 )
        iconSizeMode = GUI::AspectRatioModes::AdjustBoth;
    GUI::Rect iconRect =
    {
        ( 1 - stimulusWidth ) / 2, 0.5,
        ( 1 + stimulusWidth ) / 2, 0.5
    };
    ImageStimulus* pStimulus = new ImageStimulus( Display() );
    pStimulus->SetFile( Rest( icon, i ) )
        .SetRenderingMode( GUI::RenderingMode::Opaque )
        .SetAspectRatioMode( iconSizeMode )
        .SetObjectRect( iconRect );
    pStimulus->SetPresentationMode( VisualStimulus::ShowHide );
    mRestAnnouncement.Add( pStimulus );
}

/*If AudioSwitch is active, define a new audioStimulus and assigned
the sound to the object RestAnnouncement*/
if( audioSwitch && Rest( audio, i ) != "" )
{
    AudioStimulus* pStimulus = new AudioStimulus;
    pStimulus->SetSound( Rest( audio, i ) )
        .SetVolume( audioVolume );
    mRestAnnouncement.Add( pStimulus );
}
}
}

void
ImaginedSpeechTask::OnStartRun()
{
    DisplayMessage( LocalizableString( "Waiting to start ..." ) );
    State("InitImagine")=0;
    State("MicSampleImagine")=0;
}
```

```
/* Create a sequence for this run. */
mSequence.clear();
ParamRef Sequence = Parameter( "Sequence" );
int numSubSequences = mNumberOfSequences;
int numToBeCopied= Sequence->NumValues();

switch( mSequenceType )
{
    /*Sequence type deterministic: Generates a vector fill with the association's number
    in the exact order that it's specify in the Sequence
    parameter, and as many times as specify in
    NumberofSequence parameter.

    */
    case SequenceTypes::Deterministic:
        for( int i = 0; i < numSubSequences; ++i )
        {
            for( int j = 0; j < numToBeCopied; ++j )
                mSequence.push_back( Parameter( "Sequence" )( j ) );
        }
        mSequence.push_back( 0 );
        mToBeCopied.clear();
        mToBeCopied.insert(mToBeCopied.end(),mSequence.begin(),mSequence.end());
        break;

    /* Sequence type random: Generates a vector fill with the association's number
    as many times as defined in the frequency parameter for
    that particular association. Then it is scrambled to
    make the sequence alleatory

    */
    case SequenceTypes::Random:
        {
            vector<int> seq;
            for( int i = 0; i < numToBeCopied; ++i ){
                for( int j = 0; j < Parameter("Frequency")(i) ; ++j )
                    {seq.push_back( Parameter("Sequence")(i));
                }
            }
            random_shuffle( seq.begin(), seq.end(), RandomNumberGenerator );
            mSequence.insert( mSequence.end(), seq.begin(), seq.end() );
            mSequence.push_back( 0 );

            mToBeCopied.clear();
            mToBeCopied.insert(mToBeCopied.end(),mSequence.begin(),mSequence.end());
        }
        break;
}
/* Vector's pointers are positioned at the begin */
mSequencePos = mSequence.begin();
mToBeCopiedPos = mToBeCopied.begin();
}
```

```

void
ImaginedSpeechTask::OnStopRun()
{ /* String display during the Stop phase */
  DisplayMessage( LocalizableString( "TIME OUT!!" ) );
}

void
ImaginedSpeechTask::OnPreSequence()
{ /*Deletes the display message and calls the focus announcement presentation*/
  DisplayMessage( "" );
  mFocusAnnouncement.Present();
  mPreSequenceBlockCount = 0;
  bciout<<State("MicSample")<<endl;
  bciout<<State("SourceTime")<<endl;
}

/* Called when phase changes to PreSequence */
void
ImaginedSpeechTask::DoPreSequence( const GenericSignal&, bool& /*doProgress*/ )
{ /* If time elapsed exceeds the FocusAnnouncement duration then
   conceals the FocusAnnouncement and presents the Stimuli selected
   */
  if( mPreSequenceBlockCount == mFocusAnnouncement.StimulusDuration() )
  {
    mFocusAnnouncement.Conceal();
    if( mToBeCopiedPos == mToBeCopied.end() )
    {mToBeCopiedPos=mToBeCopied.begin();}
    Associations()[ *mToBeCopiedPos ].Present();
    State("StimulusCode")= *mToBeCopiedPos;
  }
  /* If time elapsed is equal to the sum of the FocusAnnouncement and the Stimulus
   duration, then it conceals the presentation of the Stimuli
   */

  if( mPreSequenceBlockCount == mFocusAnnouncement.StimulusDuration()
      + Associations()[*mToBeCopiedPos].StimulusDuration() )
  {
    if( mToBeCopiedPos == mToBeCopied.end() )
    {mToBeCopiedPos=mToBeCopied.begin();}
    Associations()[ *mToBeCopiedPos ].Conceal();
    ++mToBeCopiedPos;
  }
  /* If non of the above has occurred then increment the block count*/
  ++mPreSequenceBlockCount;
}

/* Called when Stimuli phase starts */
void
ImaginedSpeechTask::OnSequenceBegin()
{
  mFocusAnnouncement.Conceal();
}

```



```
}

/* Called when phase change to PostSequence */
void
ImaginedSpeechTask::OnPostSequenceBegin()
{ /* Restarts the block count and presents the RestAnnouncement */
  mPostRestBlockCount=0;
  State("MicSampleImagine")=0;
  mRestAnnouncement.Present();
}

/* Called when the program stops */
void
ImaginedSpeechTask::OnPostRun()
{
}

/* Called during Post Sequence */
void
ImaginedSpeechTask::DoPostSequence( const GenericSignal&, bool& /*doProgress*/ )
{ /* If time elapsed is equal to the duration of the RestAnnouncement, conceals it
  */
  if( mPostRestBlockCount == mRestAnnouncement.StimulusDuration() )
  {
    mRestAnnouncement.Conceal();
  }
  /* Increments the count of PostSequence */
  ++mPostRestBlockCount;
}

/* Called during OnPreSequence and OnStimulus
  gives the position of the vector Sequence.
  When ends, it send a zero that corresponds
  with the signal to end the run
  */
int
ImaginedSpeechTask::OnNextStimulusCode()
{
  return mSequencePos == mSequence.end() ? 0 : *mSequencePos++;
}

/* Is called after the Stimuli is concealed
  and presents the Imagine/Pronounce Announcement
  */
void
ImaginedSpeechTask::OnStimulusBegin()
{ State("InitImagine")=1;
  State("MicSampleImagine")=State("MicSample");
  mImagineAnnouncement.Present();
}
```

```
/*Is called when time elapsed is equal to the Imagine/  
PronounceAnnouncement Duration  
*/  
void  
ImaginedSpeechTask::OnStimulusEnd()  
{ State("InitImagine")=0;  
  State("MicSampleImagine")=100;  
  mImagineAnnouncement.Conceal();  
}  
  
/*Designed by the BCI2000 group, it looks for a Parameter  
in a particular matrix  
*/  
ParamRef  
ImaginedSpeechTask::StimulusProperty( const ParamRef& inMatrixParam,  
                                       int inColIndex,  
                                       const std::string& inProperty ) const  
{  
  return inMatrixParam->RowLabels().Exists( inProperty )  
    ? inMatrixParam( inProperty, inColIndex )  
    : Parameter( inProperty );  
}
```

Bibliografía

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Brain–computer interfaces for communication and control,” *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.
- [2] L. Sornmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Elsevier Academic Press, 1 ed., 2005.
- [3] H. Lodish, A. Berk, and S. Zipursky, *Molecular cell biology 4th edition*. National Center for Biotechnology Information’s Bookshelf, 4 ed., 2000.
- [4] C. Hammond, *Cellular and molecular neurophysiology*. Academic Press, 4 ed., 2014.
- [5] P. A. Young, P. H. Young, and D. L. Tolbert, *Basic clinical neuroscience*. Lippincott Williams & Wilkins, 2 ed., 2008.
- [6] K. Dharani, *The Biology of Thought: A Neuronal Mechanism in the Generation of Thought—a New Molecular Model*. Academic Press, 2014.
- [7] R. Snell, *Neuroanatomía clínica*. Editorial Panamericana, 5 ed., 2001.
- [8] J. Nolte, *The human brain: an introduction to its functional anatomy*. Mosby, 4 ed., 2002.
- [9] J. Seikel, D. King, and D. Drumright, *Anatomy & physiology for speech, language, and hearing*. Nelson Education, 4 ed., 2015.
- [10] L. J. Van der Knaap and I. J. Van der Ham, “How does the corpus callosum mediate interhemispheric transfer? a review,” *Behavioural Brain Research*, vol. 223, no. 1, pp. 211–221, 2011.
- [11] K. Amunts, M. Lenzen, A. D. Friederici, A. Schleicher, P. Morosan, N. Palomero-Gallagher, and K. Zilles, “Broca’s region: novel organizational principles and multiple receptor mapping,” *PLoS Biol*, vol. 8, no. 9, p. e1000489, 2010.
- [12] L. Fadiga, L. Craighero, and A. D’Ausilio, “Broca’s area in language, action, and music,” *Annals of the New York Academy of Sciences*, vol. 1169, no. 1, pp. 448–458, 2009.
- [13] J. R. Binder, J. A. Frost, T. A. Hammeke, R. W. Cox, S. M. Rao, and T. Prieto, “Human brain language areas identified by functional magnetic resonance imaging,” *The Journal of Neuroscience*, vol. 17, no. 1, pp. 353–362, 1997.

-
- [14] P. Indefrey and W. J. Levelt, “The spatial and temporal signatures of word production components,” *Cognition*, vol. 92, no. 1, pp. 101–144, 2004.
- [15] M. Catani, D. K. Jones, *et al.*, “Perisylvian language networks of the human brain,” *Annals of neurology*, vol. 57, no. 1, pp. 8–16, 2005.
- [16] A. Kertesz, “Wernicke–geschwind model,” *Encyclopedia of Cognitive Science*, 2001.
- [17] G. Hickok and D. Poeppel, “Dorsal and ventral streams: a framework for understanding aspects of the functional anatomy of language,” *Cognition*, vol. 92, no. 1, pp. 67–99, 2004.
- [18] J. D. Bronzino, *Biomedical engineering handbook*, vol. 2. CRC press, 1999.
- [19] F. L. da Silva, “Eeg: origin and measurement,” in *EEG-fMRI*, pp. 19–38, Springer, 2009.
- [20] M. Teplan, “Fundamentals of eeg measurement,” *Measurement science review*, vol. 2, no. 2, pp. 1–11, 2002.
- [21] M. Laad, “Investigating relative strengths and positions of electrical activity in the left and right hemispheres of the human brain using electroencephalography,” *Biomedical Research*, vol. 24, no. 3, pp. 359–364, 2013.
- [22] M. Ullsperger and S. Debener, *Simultaneous EEG and fMRI: recording, analysis, and application*. Oxford University Press, 2010.
- [23] W. Klimesch, “Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis,” *Brain research reviews*, vol. 29, no. 2, pp. 169–195, 1999.
- [24] T. Ergenoglu, T. Demiralp, Z. Bayraktaroglu, M. Ergen, H. Beydagi, and Y. Uresin, “Alpha rhythm of the eeg modulates visual detection performance in humans,” *Cognitive Brain Research*, vol. 20, no. 3, pp. 376–383, 2004.
- [25] G. Buzsaki, *Rhythms of the Brain*. Oxford University Press, 2006.
- [26] G. Pfurtscheller, A. Stancak, and G. Edlinger, “On the existence of different types of central beta rhythms below 30 hz,” *Electroencephalography and clinical neurophysiology*, vol. 102, no. 4, pp. 316–325, 1997.
- [27] A. K. Engel and P. Fries, “Beta-band oscillations—signalling the status quo?,” *Current opinion in neurobiology*, vol. 20, no. 2, pp. 156–165, 2010.
- [28] P. Fries, “Neuronal gamma-band synchronization as a fundamental process in cortical computation,” *Annual review of neuroscience*, vol. 32, pp. 209–224, 2009.
- [29] W. R. Goff, “Human average evoked potentials: Procedures for stimulating and recording,” *Academic Press, New York, USA*, 1974.
- [30] G. H. Klem, H. O. Lüders, H. Jasper, and C. Elger, “The ten-twenty electrode system of the international federation,” *Electroencephalogr Clin Neurophysiol*, vol. 52, no. 3, 1999.

-
- [31] D. Prutchi and M. Norris, *Design and development of medical electronic instrumentation: a practical perspective of the design, construction, and test of medical devices*. John Wiley & Sons, 2005.
- [32] M. Fatourehchi, A. Bashashati, R. K. Ward, and G. E. Birch, “Emg and eeg artifacts in brain computer interface systems: A survey,” *Clinical neurophysiology*, vol. 118, no. 3, pp. 480–494, 2007.
- [33] A. Schlögl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb, and G. Pfurtscheller, “A fully automated correction method of eeg artifacts in eeg recordings,” *Clinical neurophysiology*, vol. 118, no. 1, pp. 98–104, 2007.
- [34] R. Croft and R. Barry, “Removal of ocular artifact from the eeg: a review,” *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 30, no. 1, pp. 5–19, 2000.
- [35] I. I. Goncharova, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, “Emg contamination of eeg: spectral and topographical characteristics,” *Clinical Neurophysiology*, vol. 114, no. 9, pp. 1580–1593, 2003.
- [36] R. F. Thompson, *Foundations of physiological psychology*. Academic Press, 1967.
- [37] B. Z. Allison and C. Neuper, “Could anyone use a bci?,” in *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction* (S. D. Tan and A. Nijholt, eds.), pp. 35–54, London: Springer London, 2010.
- [38] S. Moghimi, A. Kushki, A. Marie Guerguerian, and T. Chau, “A review of eeg-based brain-computer interfaces as access pathways for individuals with severe disabilities,” *Assistive Technology*, vol. 25, no. 2, pp. 99–110, 2013.
- [39] G. Gentiletti, C. Tabernig, and R. Acevedo, “Interfaz cerebro-computadora: Estado del arte y desarrollo en argentina,” *Revista Argentina de Bioingeniería (SABI)*, vol. 13, no. 1, 2007.
- [40] A. Vallabhaneni, T. Wang, and B. He, “Brain—computer interface,” in *Neural engineering*, pp. 85–121, Springer, 2005.
- [41] V. Gandhi, *Brain-computer Interfacing for Assistive Robotics: Electroencephalograms, Recurrent Quantum Neural Networks, and User-centric Graphical Interfaces*. Academic Press, 2014.
- [42] M. Junghöfer, T. Elbert, D. M. Tucker, and C. Braun, “The polar average reference effect: a bias in estimating the head surface integral in eeg recording,” *Clinical Neurophysiology*, vol. 110, no. 6, pp. 1149–1155, 1999.
- [43] J. Dien, “Issues in the application of the average reference: Review, critiques, and recommendations,” *Behavior Research Methods, Instruments, & Computers*, vol. 30, no. 1, pp. 34–43, 1998.
- [44] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw, “Spatial filter selection for eeg-based communication,” *Electroencephalography and clinical Neurophysiology*, vol. 103, no. 3, pp. 386–394, 1997.

-
- [45] M. Najim, *Digital filters design for signal and image processing*. John Wiley & Sons, 2013.
- [46] M. H. Hayes, *Schaum's outline of digital signal processing*. McGraw-Hill, Inc., 1998.
- [47] J. I. Ababneh and M. H. Bataineh, "Linear phase fir filter design using particle swarm optimization and genetic algorithms," *Digital Signal Processing*, vol. 18, no. 4, pp. 657–668, 2008.
- [48] A. Oppenheim, R. Schaffer, and J. Buck, *Discrete-Time Singal Processing*. Prentice-Hall, Inc, 2 ed., 1999.
- [49] A. Hyvärinen, "A family of fixed-point algorithms for independent component analysis," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 5, pp. 3917–3920, IEEE, 1997.
- [50] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4, pp. 411–430, 2000.
- [51] D. Langlois, S. Chartier, and D. Gosselin, "An introduction to independent component analysis: Infomax and fastica algorithms," *Tutorials in Quantitative Methods for Psychology*, vol. 6, no. 1, pp. 31–38, 2010.
- [52] J. Onton, M. Westerfield, J. Townsend, and S. Makeig, "Imaging human eeg dynamics using independent component analysis," *Neuroscience & Biobehavioral Reviews*, vol. 30, no. 6, pp. 808–822, 2006.
- [53] S.-i. Amari, A. Cichocki, H. H. Yang, *et al.*, "A new learning algorithm for blind signal separation," *Advances in neural information processing systems*, pp. 757–763, 1996.
- [54] N. Hironaga and A. Ioannides, "Localization of individual area neuronal activity," *Neuroimage*, vol. 34, no. 4, pp. 1519–1534, 2007.
- [55] H.-G. Stark, *Wavelets and signal processing: an application-based introduction*. Springer Science & Business Media, 2005.
- [56] M. Akin, "Comparison of wavelet transform and fft methods in the analysis of eeg signals," *Journal of medical systems*, vol. 26, no. 3, pp. 241–247, 2002.
- [57] H. Adeli, Z. Zhou, and N. Dadmehr, "Analysis of eeg records in an epileptic patient using wavelet transform," *Journal of neuroscience methods*, vol. 123, no. 1, pp. 69–87, 2003.
- [58] I. Daubechies *et al.*, *Ten lectures on wavelets*, vol. 61. SIAM, 1992.
- [59] D. Morlet, F. Peyrin, P. Desseigne, P. Touboul, and P. Rubel, "Wavelet analysis of high-resolution signal-averaged ecgs in postinfarction patients," *Journal of electrocardiology*, vol. 26, no. 4, pp. 311–320, 1993.
- [60] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 674–693, 1989.

-
- [61] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, “Wavelet toolbox for use with matlab, the mathworks,” *Natick, MA*, 1996.
- [62] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [63] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R news*, vol. 2, no. 3, pp. 18–22, 2002.
- [64] K. J. Archer and R. V. Kimes, “Empirical characterization of random forest variable importance measures,” *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 2249–2260, 2008.
- [65] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, “Random forest: a classification and regression tool for compound classification and qsar modeling,” *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.
- [66] S.-i. Amari and S. Wu, “Improving support vector machine classifiers by modifying kernel functions,” *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.
- [67] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [68] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.
- [69] S. Abe, *Support vector machines for pattern classification*, vol. 2. Springer, 2005.
- [70] C. Campbell and Y. Ying, “Learning with support vector machines,” *Synthesis lectures on artificial intelligence and machine learning*, vol. 5, no. 1, pp. 1–95, 2011.
- [71] H.-K. Lam and H. T. Nguyen, *Computational intelligence and its applications: evolutionary computation, fuzzy logic, neural network and support vector machine techniques*. World Scientific, 2012.
- [72] P. Suppes, Z.-L. Lu, and B. Han, “Brain wave recognition of words,” *Proceedings of the National Academy of Sciences*, vol. 94, no. 26, pp. 14965–14969, 1997.
- [73] M. D’Zmura, S. Deng, T. Lappas, S. Thorpe, and R. Srinivasan, “Toward eeg sensing of imagined speech,” in *Human-Computer Interaction. New Trends*, pp. 40–48, Springer, 2009.
- [74] K. Brigham and B. Kumar, “Imagined speech classification with eeg signals for silent communication: a preliminary investigation into synthetic telepathy,” in *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on*, pp. 1–4, IEEE, 2010.
- [75] C. S. DaSalla, H. Kambara, M. Sato, and Y. Koike, “Single-trial classification of vowel speech imagery using common spatial patterns,” *Neural Networks*, vol. 22, no. 9, pp. 1334–1339, 2009.

-
- [76] M. Wester and T. Schultz, "Unspoken speech-speech recognition based on electroencephalography," *Tesis de Maestría, Universität Karlsruhe (TH), Karlsruhe, Germany*, 2006.
- [77] M. Wand, "Wavelet-based preprocessing of electroencephalographic and electromyographic signals for speech recognition," *Tesis de grado, Interactive Systems Laboratories Carnegie Mellon University, Pittsburgh, PA, USA and Institut für Theoretische Informatik Universität Karlsruhe (TH), Karlsruhe, Germany*, 2007.
- [78] J.-P. Calliess, "Further investigations on unspoken speech," *Tesis de grado, Universität Karlsruhe (TH), Karlsruhe, Germany*, 2006.
- [79] A. Porbadnikg, "Eeg-based speech recognition: impact of experimental design on performance," *Tesis de grado, Universität Karlsruhe (TH), Karlsruhe, Germany*, 2008.
- [80] A. Torres-García, C. Reyes-García, L. Villaseñor-Pineda, and J. Ramírez-Cortés, "Análisis de señales electroencefalográficas para la clasificación de habla imaginada," *Revista mexicana de ingeniería biomédica*, vol. 34, no. 1, pp. 23–39, 2013.
- [81] L. Aronson, P. Estienne, D. Milone, C. Martínez, H. Rufiner, and M. Torres, "Base de datos batería de evaluación para pacientes con prótesis auditivas (beppa)," *Editorial: FASO (Federación Argentina de Sociedades de Otorrinolaringología)*, vol. 14, no. 1, pp. 17–24, 2007.
- [82] G. Schalk and J. Mellinger, *A Practical Guide to Brain-Computer Interfacing with BCI2000: General-Purpose Software for Brain-Computer Interface Research, Data Acquisition, Stimulus Presentation, and Brain Monitoring*. Springer Science & Business Media, 2010.
- [83] C. Bonell, A. Cherniz, A. Hadad, and B. Drozdowicz, "Basebio: base de datos de señales biomédicas accesible desde internet," in *Proceedings of the XVIII Congreso Argentino de Bioingeniería, Mar del Plata, Argentina*, 2011.