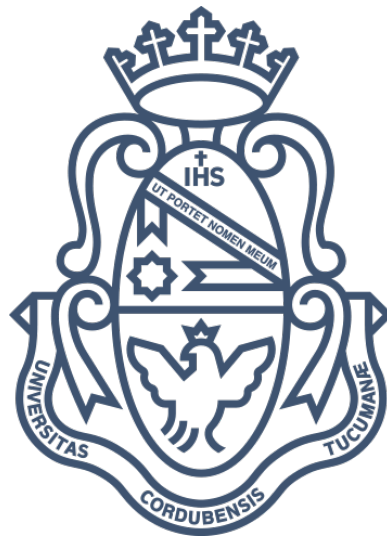


UNIVERSIDAD NACIONAL DE CÓRDOBA - UNC
**FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y
NATURALES**

INGENIERÍA BIOMÉDICA

PROYECTO INTEGRADOR

**Diseño e Implementación de una Herramienta para el
Procesamiento y Optimización del Análisis Cualitativo y
Cuantitativo de Señales de Electroencefalografía**



Alumnos:

Lemos, Dante

Matrícula: 38.139.698

López, Juan Manuel

Matrícula: 39.764.711

Director:

Ing. Garro, Florencia

Co-director:

Ing. Ventre, Luis



FCM
Facultad de
Ciencias Médicas



Septiembre, 2019

Agradecimientos

A Dios, a Jesús y a la Virgen María, por darnos la oportunidad de seguir nuestros sueños y poder estudiar la carrera que nos apasiona.

A nuestra directora Florencia Garro quien confió en nosotros para realizar este proyecto, por el invaluable aporte de su experiencia y por su incondicional apoyo y paciencia para guiarnos en todo momento.

A nuestro director Luis Ventre junto con la contribución del Ing. Orlando Micolini, por brindar sus conocimientos y consejos no sólo en áreas técnicas sino también en aspectos generales de la profesión.

A nuestras familias, por ser un apoyo incansable en los momentos difíciles y los primeros en aplaudir nuestros éxitos y victorias. Especialmente a nuestros padres, que nos enseñaron a creer que un día seríamos autores de nuestros logros, transmitiéndonos valores y siendo en gran parte responsables de quienes somos hoy.

A nuestros compañeros de estudio que compartieron el sueño de ser profesionales, que con sus palabras y compañía hicieron más agradable el día a día a lo largo de toda la carrera.

A nuestras amistades que sin pedirnos nada a cambio han compartido experiencias, disfrutado de alegrías y apoyado en momentos de tristezas, ellos han sido un apoyo muy importante en este camino recorrido.

A la Iglesia de los Capuchinos y sus grupos de jóvenes, que se han convertido en nuestro segundo hogar y en el de tantos estudiantes que se encuentran lejos del suyo.

Finalmente a la Universidad Nacional de Córdoba, por brindarnos educación pública y gratuita de altísimo nivel. Por ser casa de tantos docentes y profesionales que más allá de enseñar y evaluar los temas de alguna materia comparten sus conocimientos y experiencias de vida, invitándonos a superarnos constantemente.

A todos y cada uno de ellos nuestro eterno agradecimiento.

Abstract

La Electroencefalografía es una técnica de imagen médica ampliamente utilizada debido a su bajo costo y no invasividad, tanto en el ámbito clínico como experimental. Estas características le confieren una gama amplia y variada de aplicaciones, no sólo en el área de la medicina sino también en investigación en neurociencia, psicología cognitiva, y psicofisiología.

Aún hoy la interpretación de las mediciones de EEG se realiza, en la mayoría de los casos, mediante el reconocimiento visual de patrones establecidos. Como consecuencia de la baja amplitud de la señal medida y la cantidad de fuentes de ruido que la influyen, este método requiere del entrenamiento del observador y es altamente subjetivo. Por otro lado, la información de interés puede aparecer en las señales de EEG en forma de detalles sutiles o incluso no aparecer a simple vista, por lo que aún un observador entrenado puede no ser capaz de detectarlos.

Por este motivo existe un creciente interés en el desarrollo de herramientas que implementen técnicas digitales de procesamiento para la interpretación de señales electroencefalográficas, que consisten en transformar la señal cruda (sin procesar) en datos numéricos y/o gráficos que faciliten su análisis y comprensión.

En este proyecto se plantea el diseño de una herramienta desarrollada en Python sobre el entorno programación Jupyter. La herramienta brinda a cualquier usuario con mínimos conocimientos de programación la posibilidad de llevar a cabo un pipeline de procesamiento de señales electroencefalográficas con métodos preestablecidos, pudiendo variar algunos de sus parámetros. Se orienta su diseño para archivos de señal generados por el hardware de adquisición de bioseñales Ganglion de OpenBCI, buscando luego su extensión a otros formatos.

PALABRAS CLAVE: ELECTROENCEFALOGRAFÍA – PYTHON – JUPYTER – PROCESAMIENTO DE SEÑALES – TÉCNICAS DIGITALES DE PROCESAMIENTO.

Índice general

Índice de figuras	6
Índice de cuadros	9
Introducción	10
Objetivos	12
Objetivos Generales	12
Objetivos Específicos	12
1. Marco Teórico	13
1.1. El Cerebro	13
1.1.1. Estructura del Sistema Nervioso	15
1.2. Actividad Eléctrica Cerebral	16
1.2.1. Sinapsis	18
1.2.2. Potencial Postsináptico	19
1.3. Electroencefalografía	20
1.3.1. Electroodos	21
1.3.2. Sistema 10-20	23
1.3.3. Patrones Asociados a Funciones Cerebrales	24
1.3.3.1. Ritmos Cerebrales	24
1.3.3.2. Potenciales Evocados	26
1.4. Componentes Eléctricos no Dependientes de la Actividad Cerebral: Artefactos	27
2. Descripción de los Métodos Implementados para el Procesamiento de las Señales de EEG	31
2.1. Filtros	31
2.1.1. Caracterización de un Filtro	33
2.1.1.1. Respuesta al Impulso	33
2.1.1.2. Respuesta al Escalón	33
2.1.1.3. Respuesta en Frecuencia	34
2.1.2. Orden de un Filtro	35
2.1.3. Tipos de Filtros	36
2.1.4. Ejemplos de Filtros	37
2.2. Rreferenciado Digital	39
2.3. Transformada de Fourier	40

2.3.1.	Transformada de Fourier Discreta	40
2.3.2.	Algoritmo FFT - Fast Fourier Transform	41
2.3.3.	Transformada Welch	41
2.4.	Análisis de Componentes Independientes (ICA)	42
2.5.	Espectrograma	45
2.6.	Correlación y Coherencia	46
2.6.1.	Correlación	46
2.6.2.	Coherencia	47
3.	Materiales y Métodos	49
3.1.	Materiales y Herramientas	49
3.1.1.	OpenBCI	49
3.1.2.	Entornos de Programación: Anaconda y Jupyter	50
3.1.2.1.	Anaconda	50
3.1.2.2.	Jupyter	51
3.1.3.	Lenguaje de Programación: Python	51
3.1.4.	Librerías de Python	52
3.1.4.1.	Numpy: Librería Numérica de Python	53
3.1.4.2.	Scipy: Librería Científica de Python	53
3.1.4.3.	Pandas: Librería de Manejo de Datasets	54
3.1.4.4.	Matplotlib: Librería de Gráficos Matemáticos	54
3.1.4.5.	Plotly	55
3.1.4.6.	MNE: Librería para el manejo de Señales de Neuroimage- nología	56
3.1.4.7.	Wyrn	56
3.1.4.8.	Tkinter: Librería de Interfaz Gráfica	56
3.2.	Desarrollo del Proyecto	57
3.2.1.	Requisitos de la Herramienta	57
3.2.2.	Diagrama General	59
3.2.3.	Iteraciones	60
3.2.3.1.	Codificación de Diagramas de Flujo	60
3.2.3.2.	Iteración 1: Adquisición de Datos	61
3.2.3.3.	Iteración 2: Acondicionamiento de la Señal	69
3.2.3.4.	Iteración 3: Análisis en Tiempo	76
3.2.3.5.	Iteración 4: Mapas Topográficos	81
3.2.3.6.	Iteración 5: Análisis en Frecuencia	85
3.2.3.7.	Iteración 6: Visualización	88
3.2.3.8.	Iteración 7: Correlación y Coherencia	91
3.2.3.9.	Iteración 8: Exportación de Datos	94
3.2.4.	Revisión de Requisitos iniciales	95
4.	Verificación	98
4.1.	Señales utilizadas para la verificación:	98
4.2.	Verificaciones:	103

4.2.1. Carga de archivos en distintos formatos:	103
4.2.2. Definición de la Tasa de Muestreo:	104
4.2.3. Intervalos definidos por etiquetas:	105
4.2.4. Filtrado:	107
4.2.5. Submuestreo y Nyquist:	112
4.2.6. Descomposición ICA:	114
4.2.7. CAR:	117
4.2.8. Transformada en Frecuencia: FFT y Welch:	118
4.2.9. Espectrograma:	120
4.2.10. Mapas Topográficos:	121
4.2.11. Correlación:	123
4.2.12. Coherencia:	125
5. Casos de Aplicación	127
5.1. Análisis de Patrones Fisiológicos en Frecuencia	127
5.1.1. Ondas Alfa	127
5.1.2. Potencial Evocado Visual de Estado Estacionario (SSVEP)	131
5.1.3. Conclusiones	133
5.2. Análisis Potencial Evocado P300	133
5.3. Eliminación de ruido	136
Conclusión	139
5.4. Conclusión	139
5.5. Trabajo a futuro	140
Referencias	141

Índice de figuras

1.1. Encéfalo	13
1.2. Lóbulos Cerebrales	14
1.3. Lóbulo de la Ínsula	15
1.4. Neurona	16
1.5. Potencial de Membrana	17
1.6. Potencial de Acción	18
1.7. Sinapsis	19
1.8. Potencial Postsináptico	20
1.9. Electroencefalograma	21
1.10. Técnicas Electroencefalográficas	22
1.11. Electroodos Superficiales	23
1.12. Sistema de posicionamiento 10-20	24
1.13. Ondas Cerebrales	26
1.14. Artefacto Muscular	28
1.15. Artefacto de Pestañeo	29
1.16. Artefacto Cardíaco	29
1.17. Artefacto Movimiento	29
1.18. Artefacto Eléctricos	30
2.1. Filtros según Aceptación y Rechazo	32
2.2. Ejemplo de una Señal Impulso	33
2.3. Diagramas de Bode	34
2.4. Respuestas de Filtros	35
2.5. Orden del Filtro	35
2.6. Filtros FIR vs. IIR	36
2.7. Respuesta Filtro Pasa Bajo	38
2.8. Filtro Chebyshev y Notch	38
2.9. Referenciado Digital	40
2.10. ICA descomposición	44
2.11. ICA composición	44
2.12. Espectrograma	45
2.13. Correlación Cruzada	47
3.1. Hardware OpenBCI	50
3.2. Aplicaciones Científicas de Python	52

3.3. Diferencias entre Numpy <i>arange</i> y <i>linspace</i>	53
3.4. Tabla en Formato Pandas	54
3.5. Gráfico Matplotlib	55
3.6. Ventana Tkinter General	57
3.7. Diagrama General de Etapas	60
3.8. Codificación utilizada para los Diagramas Específicos	61
3.9. Diagrama: Adquisición de Datos	61
3.10. Carga de Archivos	62
3.11. Definición Parámetros de Carga	63
3.12. Configuración de Canales	64
3.13. Ventana de Muestreo	65
3.14. Excel para Intervalos	66
3.15. Definición de Intervalos por Etiquetas	67
3.16. Diccionario de información de la Señal	68
3.17. Comparación de datos Python y MNE	69
3.18. Diagrama: Acondicionamiento de la Señal	69
3.19. Parámetros de Filtrado	70
3.20. Filtros MNE	72
3.21. Ejemplo de Submuestreo de una Señal	73
3.22. Teorema de Nyquist	74
3.23. Confirmación de Submuestreo de la Señal.	74
3.24. Diagrama: Análisis en Tiempo	76
3.25. Ventana de Rereferenciado	77
3.26. Ventana para Análisis de Componentes Independientes de la señal (ICA).	79
3.27. Diagrama: Mapas Topográficos	81
3.28. Montaje de Mapas Topográficos	82
3.29. Creación de Variable Señal Expandida	83
3.30. Ejemplo de Mapa Topográfico	84
3.31. Diagrama: Análisis en Frecuencia	85
3.32. Forma en Tiempo y Respuesta en Frecuencia de la Ventana Hanning	87
3.33. Diagrama: Visualización	88
3.34. Ventana de Visualización	89
3.35. Diagrama: Correlación y Coherencia	91
3.36. Advertencia de uso del P-Valor en Coherencia	92
3.37. Valores Rho de Coherencia	93
3.38. Valores promedio de Coherencia	94
3.39. Ejemplo de exportación de datos	95
4.1. Valores Aleatorios separados por Espacios	99
4.2. Valores Aleatorios separados por Punto y Coma	99
4.3. Valores Aleatorios en Excel	99
4.4. Señales Senoidales a distintas Frecuencias	100
4.5. Señales senoidales de frecuencia creciente	100
4.6. Señales Senoidales desfasadas y escaladas	101

4.7. Señal Ejemplo de Ondas Cuadradas.	102
4.8. Señal Dummy con ruido de Pestaño y aleatorio	103
4.9. Señal Dummy con ruido de Pestaño	103
4.10. Verificación de Carga de Señal	104
4.11. Carga de Señales con dos columnas tipo <i>Time</i>	104
4.12. Verificación de Detección Automática de la Tasa de Muestreo	105
4.13. Señal con columnas de etiquetas	106
4.14. Selección de columna de tipo Tags	106
4.15. Ejemplo de Intervalo generado por etiquetas	107
4.16. Comparativa del cambio de Orden de Filtro Butterworth	108
4.17. Efecto del uso de Filtro Notch	109
4.18. Comparativa del cambio de Orden de Filtro Chebyshev	110
4.19. Comparativa del cambio de Riple de Filtro Chebyshev	111
4.20. Submuestreo en Señales de 10 y 15 Hz	113
4.21. Submuestreo en Señales de 20 y 40 Hz	113
4.22. Submuestreo en Señales de 50 y 60 Hz	114
4.23. Espectro de Frecuencia en Submuestreo	114
4.24. Señales sin aplicación de ICA	115
4.25. Componentes en la aplicación de ICA	116
4.26. Señales con aplicación de ICA	116
4.27. Señal de Ondas Cuadradas luego de la aplicación de CAR.	117
4.28. Aplicación de CAR a Señal Dummy	118
4.29. Transformación por FFT	119
4.30. Transformación por Welch	120
4.31. Espectrograma de Señal Escalonada en Frecuencias	120
4.32. Posicionamiento utilizado para la Verificación de Mapas Topográficos	121
4.33. Verificación de Mapa Topográfico en un instante	122
4.34. Verificación del Mapa Topográfico de Promedio de Señal	123
4.35. Ejemplos de Señal para la Verificación de Correlación	124
4.36. Verificación de Correlación	124
4.37. Espectro en Frecuencia de las Señales usadas al verificar Coherencia	125
4.38. Verificación de Coherencia	126
5.1. Espectro de Frecuencia en Ojos Abiertos	129
5.2. Espectro de Frecuencia en Ojos Cerrados	130
5.3. Espectro en Frecuencia en Estimulación de 20 Hz	132
5.4. Paradigma <i>odd-ball</i>	133
5.5. Posiciones de Canales para el estudio de onda P300	134
5.6. Estimulaciones del estudio de onda P300	135
5.7. Estimulación y onda P300 resultante	136
5.8. Artefacto de Pestaño	137
5.9. Artefacto de Pestaño con CAR	138
5.10. Artefacto de Pestaño con ICA	138

Índice de cuadros

2.1. Comparativa entre Filtros FIR e IIR	36
3.1. Requisitos del Proyecto	58
3.2. Revisión de Requisitos	97

Introducción

La electroencefalografía (EEG) es una técnica de imagen médica que mediante la colocación de electrodos construidos con distintos materiales conductivos, es utilizada para medir la actividad eléctrica del cerebro. La misma puede ser aplicada de manera no invasiva y puede ser utilizada por períodos de tiempo prolongados sin implicar ningún riesgo para el paciente.

La utilidad de esta técnica radica en que las distintas propiedades de las señales de EEG (forma, amplitud, ubicación espacial, contenido frecuencial, etc.), contienen información útil acerca de diferentes estados mentales asociados a actividades motoras, sensoriales y estadios del sueño o de diferentes enfermedades.

La técnica de EEG es ampliamente utilizada debido a su bajo costo y no invasividad, tanto en el ámbito clínico como experimental. En contraste a la Tomografía Computarizada (CT), la cual expone al paciente a radiación; y en comparación con la Resonancia Magnética (MRI), que es mucho más costosa y suele ser menos comfortable. Estas características le confieren una amplia gama y variada de aplicaciones, no sólo en el área de la medicina, sino también en la investigación en neurociencia, psicología cognitiva, y psicofisiología.

Debido a que muchas veces la información relevante no puede observarse de forma explícita, sino oculta entre grandes cantidades de datos, y además puede encontrarse contaminada por artefactos o ruido externo, existe un creciente interés en el desarrollo de técnicas digitales de procesamiento que faciliten la interpretación y análisis de señales electroencefalográficas.

La motivación de este proyecto surge en el marco de la empresa OTTAA Project, dedicada al desarrollo de un sistema aumentativo y alternativo de comunicación basado en el uso de pictogramas. El mismo consiste en una interfaz táctil, quedando su uso limitado a personas que cuentan con algún grado de capacidad motriz. En estos momentos, la empresa se encuentra en desarrollo de un módulo de adquisición de señales electroencefalográficas que permita establecer una interfaz entre la aplicación desarrollada y la actividad cerebral, permitiendo constituir una herramienta alternativa de comunicación para aquellos usuarios que no pueden utilizar el software de manera autónoma debido a discapacidades motrices severas asociadas a distintas patologías.

El presente proyecto abarca el área de las técnicas digitales de procesamiento, específicamente aquellas implementadas para el procesamiento de señales adquiridas mediante electroencefalografía. Se estudia el estado del arte de las mismas y sus características para realizar a continuación la selección de aquellos métodos que mejor se adapten a los requisitos planteados por el usuario. Finalmente se desarrolla una herramienta que

implementa dichas técnicas de procesamiento para el estudio de estas señales.

El trabajo está organizado en cinco capítulos. El Capítulo I desarrolla el marco teórico y conceptual desde el cual se aborda proyecto, donde se introducen conceptos generales sobre la anatomía y fisiología del sistema nervioso para describir como se producen las señales obtenidas mediante la técnica de EEG. En el Capítulo II se describen de manera detallada los métodos estudiados e implementados en el desarrollo de la herramienta. El Capítulo III incluye el marco metodológico del proyecto y describe el proceso de diseño y desarrollo de la herramienta. En el Capítulo IV se describe el proceso de verificación de los métodos implementados para asegurar la correcta utilización de los mismos dentro de parámetros pre establecidos y en el Capítulo V se presentan 3 casos de aplicación. A modo de cierre del proyecto integrador, se enuncian las conclusiones alcanzadas y los nuevos objetivos y/o propuestas a futuro sugeridas.

Objetivos

Objetivo General

Diseñar y desarrollar una herramienta para el procesamiento de señales de electroencefalografía (EEG).

Objetivos Específicos

- Estudiar las bases fisiológicas de las señales obtenidas mediante la técnica de electroencefalografía.
- Estudiar el estado del arte de los métodos más utilizados para el procesamiento de señales adquiridas mediante EEG.
- Seleccionar los métodos más adecuados para la resolución de los requisitos planteados y desarrollar una herramienta que aplique dichos métodos para el estudio de las señales electroencefalográficas.
- Diseñar e implementar una interfaz que permita a usuarios con conocimientos limitados de programación la utilización de dicha herramienta.

Capítulo 1

Marco Teórico

En el presente capítulo se realiza una breve introducción a las señales de electroencefalografía (EEG). Como primer paso, resulta necesario introducir conceptos generales sobre la anatomía y fisiología del sistema nervioso para describir cómo se producen las señales obtenidas mediante la técnica de EEG, en la cual se enfoca este proyecto.

1.1. El Cerebro

El encéfalo es la parte del sistema nervioso central responsable del control del movimiento, respiración, vista, tacto y temperatura, entre otras actividades vitales necesarias para la supervivencia. Se encuentra contenido en la caja craneal y está formado por tres partes bien diferenciadas: el Tronco Cerebral, el Cerebelo y el Cerebro, cuya superficie externa es conocida como Corteza Cerebral (Figura 1.1).

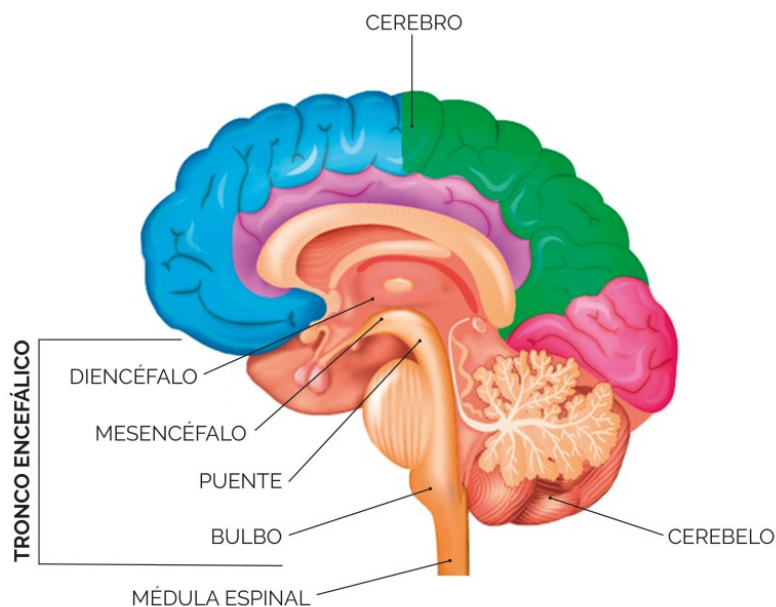


Figura 1.1: Encéfalo¹

¹Fuente: Curso de Neuroanatomía Funcional- Unidad 4.2 - El Sistema Nervioso - Consejo Mexicano de Neurociencias

En la corteza cerebral se presentan prominencias (denominadas circunvoluciones) y cisuras. La ubicación de las cisuras es utilizada para delimitar las distintas zonas o regiones del cerebro. La más importante es la denominada Cisura Longitudinal, que divide al cerebro en hemisferio derecho e izquierdo, pero a su vez cada hemisferio se divide en 5 lóbulos (Figura 1.2):

Lóbulo Frontal: Relacionado con funciones motoras, socialización, el juicio y la producción del lenguaje. Asiste en la planificación, coordinación, control y ejecución de las conductas.

Lóbulo Parietal: Relacionado con el procesamiento de información sensorial procedentes de distintas partes del cuerpo como así también la manipulación de los objetos.

Lóbulo Temporal: Sus principales funciones se relacionan con la memoria, involucrando el recuerdo de palabras y objetos, como así también la memoria visual (caras, imágenes, etc.).

Lóbulo Occipital: Donde reside la corteza visual, relacionado con el procesamiento de estímulos visuales de forma permanente, analizando distancias, formas, colores, etc.

Lóbulo de la Ínsula: Es una estructura de la corteza cerebral situada en la profundidad de la cisura de Silvio, en el punto en que confluyen los lóbulos temporal, parietal y frontal (Figura 1.3). Está implicada en procesos emocionales tan importantes como el amor materno, el amor romántico, el miedo, la tristeza, la felicidad, la empatía o la excitación sexual entre otros.

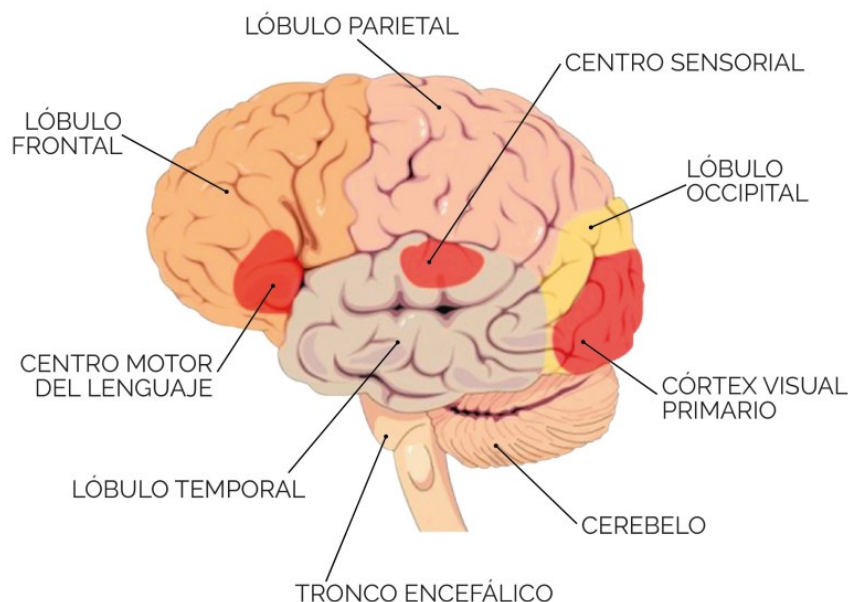


Figura 1.2: Anatomía del cerebro²

²Fuente: Contenidos didácticos Licenciatura en Enfermería y obstetricia - Unidad Didáctica 3 - Universidad de Guanajuato

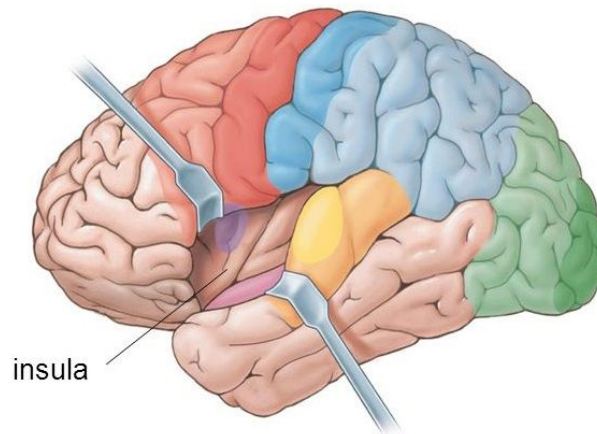


Figura 1.3: Lóbulo de la Ínsula³

1.1.1. Estructura del Sistema Nervioso

El sistema nervioso está compuesto por un tejido altamente especializado, el cual está dividido en dos grandes grupos celulares: las neuronas, y la neuroglia. La neuroglia, constituida por las células de sostén y aislamiento, mantiene a las neuronas en su lugar, se encarga de su nutrición y actúa como una barrera hematoencefálica, regulando el flujo entre el contenido de la sangre y el tejido nervioso.

La neurona es la unidad básica del sistema nervioso debido a que posee en sí misma todos los elementos que le permiten funcionar: recibe información por diferentes vías, evalúa las múltiples entradas, transmite los resultados de este procesamiento como tren de impulsos o secreciones y adapta su morfología según la actividad desarrollada.

El sistema nervioso contiene más de 100.000 millones de neuronas y las mismas han desarrollado en forma notable las propiedades de excitabilidad y conductibilidad. Éstas poseen tres partes principales: el soma o cuerpo celular, las dendritas y el axón (Figura 1.4).

El soma o cuerpo celular es la parte principal de la neurona. Su forma es variable y es allí donde se producen todas las actividades metabólicas de la célula. Las dendritas son prolongaciones que salen de distintas partes del soma y tienen la función de recibir impulsos de otras neuronas y enviarlos hasta el soma. Cada neurona tiene muchas dendritas que se dividen formando un sistema de ramificaciones. En contraste, el axón es una única prolongación del soma que se extiende en dirección opuesta a las dendritas y tiene la función de conducir un impulso nervioso desde el soma hacia otra neurona, músculo o glándula del cuerpo humano.

³Fuente: <https://www.psicoadictiva.com> - Neurología y Ciencias. Último Ingreso : 11/08/19

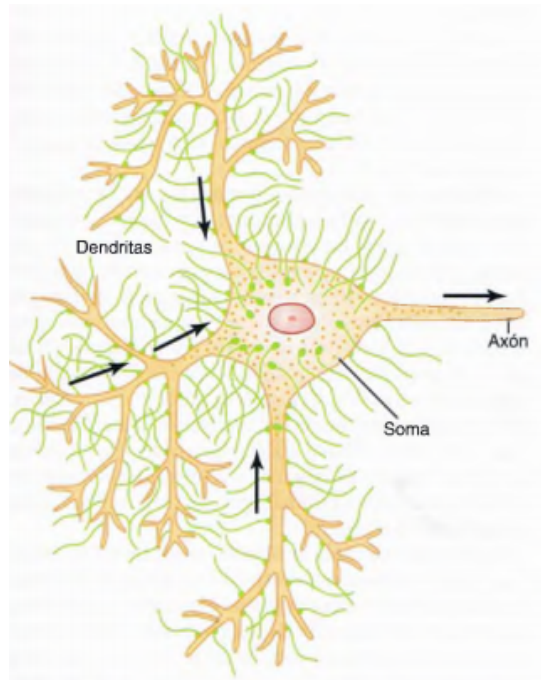


Figura 1.4: Motoneurona típica que muestra los terminales presinápticos sobre el soma neuronal y las dendritas. Se observa también el único axón. [7]

La mayor parte de la información que ingresa al sistema nervioso es mediante un estímulo que excita a los receptores sensitivos. Éstos pueden estar relacionados a sentidos de exterocepción (percepción del exterior del cuerpo), como el visual en los ojos, auditivo en los oídos, táctil en la superficie del organismo, o a sentidos de propiocepción (percepción del estado interno del cuerpo) a través de receptores específicos ubicados en los músculos, tendones, articulaciones e oído interno, denominados propioceptores.

Las señales de entrada llegan a las neuronas a través de las uniones situadas fundamentalmente en las dendritas neuronales, pero también en el soma. Las conexiones aferentes de las neuronas pueden ser tan solo unos cientos o incluso hasta 200.000. En cambio, la señal de salida viaja por el único axón que abandona la neurona y el cual da origen a numerosas ramas independientes que se dirigen a una o varias neuronas del sistema nervioso.

1.2. Actividad Eléctrica Cerebral

Las neuronas en reposo, es decir cuando no transmiten información, mantienen un potencial negativo denominado potencial de membrana (-70 mV aproximadamente) y el mismo corresponde a la diferencia eléctrica entre el interior y el exterior de dicha célula. El interior de la neurona está cargada negativamente debido al predominio de cargas negativas respecto al contenido de K^+ y el exterior cargado positivamente (predominio de iones positivos) debido a concentraciones altas de Na^+ .

Estos potenciales son producidos debido al flujo de iones a través de canales en las membranas que son selectivamente permeables y favorecen la distribución no uniforme de estos iones a ambos lados de la membrana (Figura 1.5).

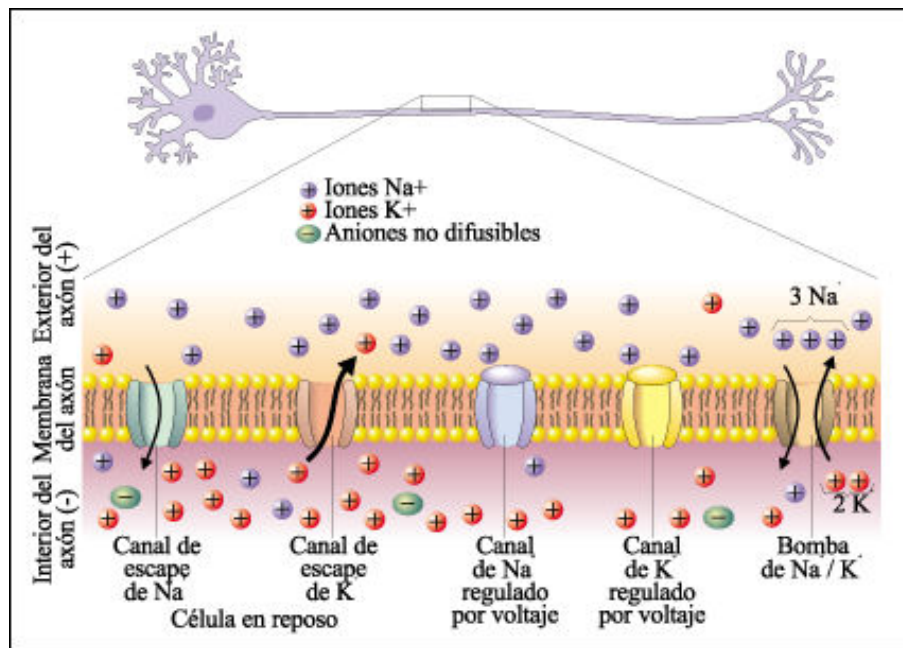


Figura 1.5: Potencial de Membrana provocado por una distribución no uniforme de iones.

4

Las células nerviosas transmiten información mediante potenciales de acción, dando como resultado señales eléctricas. Éstos potenciales son cambios rápidos del potencial de membrana que se extienden a lo largo de toda la fibra nerviosa (Figura 1.6).

El potencial de acción produce un cambio en el potencial de reposo negativo y lo vuelve positivo de manera transitoria. Es un fenómeno de todo o nada, es decir, si un estímulo es suficiente como para desencadenar un impulso nervioso (despolarización por encima de un valor umbral), éste impulso se produce en su totalidad. En cambio, si el estímulo no es suficiente para superar el valor umbral no producirá ningún potencial.

Las fases del potencial de acción son las siguientes:

Fase de reposo: es el estado previo al potencial de acción, se considera a la membrana polarizada durante esta fase debido al potencial negativo de -90mV que está presente.

Fase de despolarización: el estado normal de -90mV se neutraliza inmediatamente debido al ingreso de iones sodio a la célula cargados positivamente, y el potencial aumenta rápidamente en dirección positiva.

Fase de repolarización: Los canales selectivos al ion sodio comienzan a cerrarse y aquellos selectivos al ion potasio se abren más de lo normal. De ésta manera, la rápida difusión de iones potasio hacia el exterior restablece el potencial de membrana en reposo negativo.

⁴Fuente: Neurociencia Cognitiva - Unidad 2.1 - Conceptos Básicos de Neuroanatomía - Consejo Mexicano de Neurociencias

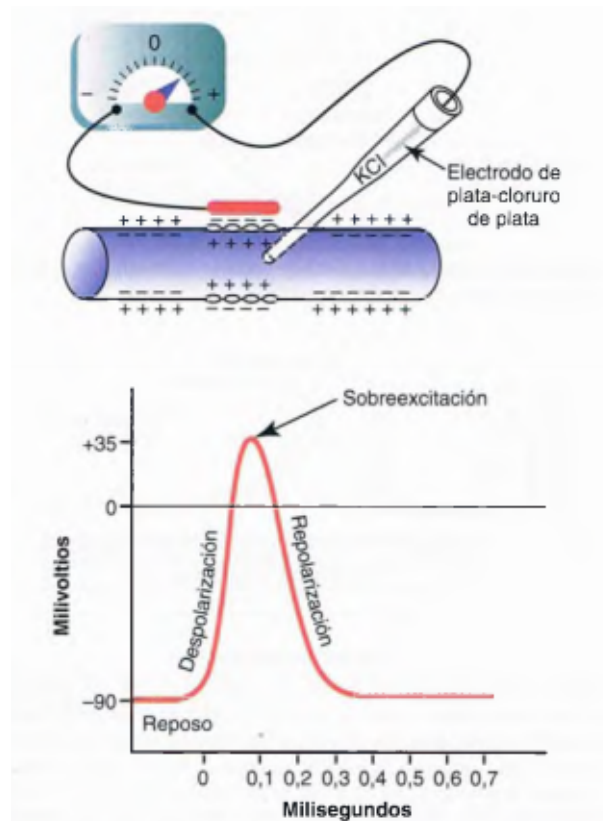


Figura 1.6: Potencial de Acción [7]

1.2.1. Sinapsis

Como se mencionó anteriormente, las señales se transmiten de una neurona a otra por medio de conexiones en sus dendritas o soma. Éstas uniones se denominan sinapsis y las mismas determinan la dirección de propagación que toma cualquier señal por el sistema nervioso (Figura 1.7). Además, efectúan una acción selectiva: en algunos casos son capaces de bloquear los estímulos débiles a la vez que dejan pasar los más potentes, pero en otras circunstancias seleccionan y amplifican estímulos débiles, y los conducen en muchas direcciones en vez de en una sola.

Existen dos tipos principales de sinapsis :

1. Sinapsis Química
2. Sinapsis Eléctrica

Las sinapsis químicas son utilizadas principalmente para la transmisión de señales en el sistema nervioso. En ellas, primero la neurona *presináptica* segrega un producto químico, denominado *neurotransmisor* a nivel de la terminación nerviosa, el cual actúa sobre las proteínas receptoras presentes la membrana de la neurona *postsináptica* para excitarla, inhibirla o modificar su sensibilidad.

Por el contrario, en las sinapsis eléctricas, la transmisión entre una neurona y la otra se caracteriza por el paso de iones a través de “*uniones GAP*”. Las uniones GAP son pequeños canales formados entre células estrechamente adheridas por el acoplamiento

de complejos proteicos. Éstas uniones intercelulares permiten el paso de agua, iones y pequeñas moléculas.

Las sinapsis eléctricas son mencionadas pero no serán descritas en el proyecto debido que son muy escasas y no se encuentran en la corteza cerebral. Por lo tanto no contribuyen de manera significativa el registro de EEG.

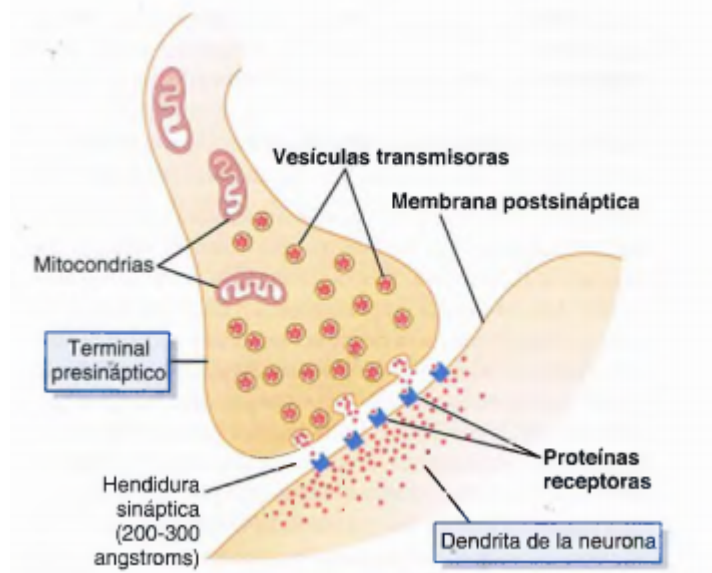


Figura 1.7: Anatomía fisiológica de una sinapsis[7]

1.2.2. Potencial Postsináptico

En la Figura 1.8 A. se muestra una neurona en reposo con un terminal presináptico sin excitar apoyado sobre su superficie. El potencial de reposo en cualquier punto del soma es de -65mV .

La Figura 1.8 B. presenta un terminal presináptico que ha segregado un neurotransmisor, el cual actúa sobre el receptor postsináptico incrementando la permeabilidad de la membrana al ion sodio, elevando así la negatividad eléctrica dentro de la neurona. Su valor cambia de -65mV a -45mV .

Este ascenso positivo en el voltaje desde el potencial de reposo hacia un valor menos negativo, se denomina potencial postsináptico excitador (PPSE), debido a que si aumenta lo suficiente en este sentido desencadenará un potencial de acción en la neurona postsináptica. De la misma manera, si el valor del potencial se modifica hacia un valor más negativo, se denomina potencial postsináptico inhibitorio (PPSI), requiriendo un mayor potencial para desencadenar un potencial de acción. En este caso el PPSE es de 20mV por encima del potencial de reposo[7].

Cabe destacar que la descarga de un solo terminal presináptico no es capaz de incrementar el potencial neuronal en 20mV , un ascenso de tal magnitud requiere el disparo simultáneo de muchos terminales (en muchos casos de unos 40 a 80 terminales) al mismo tiempo o en rápida sucesión.

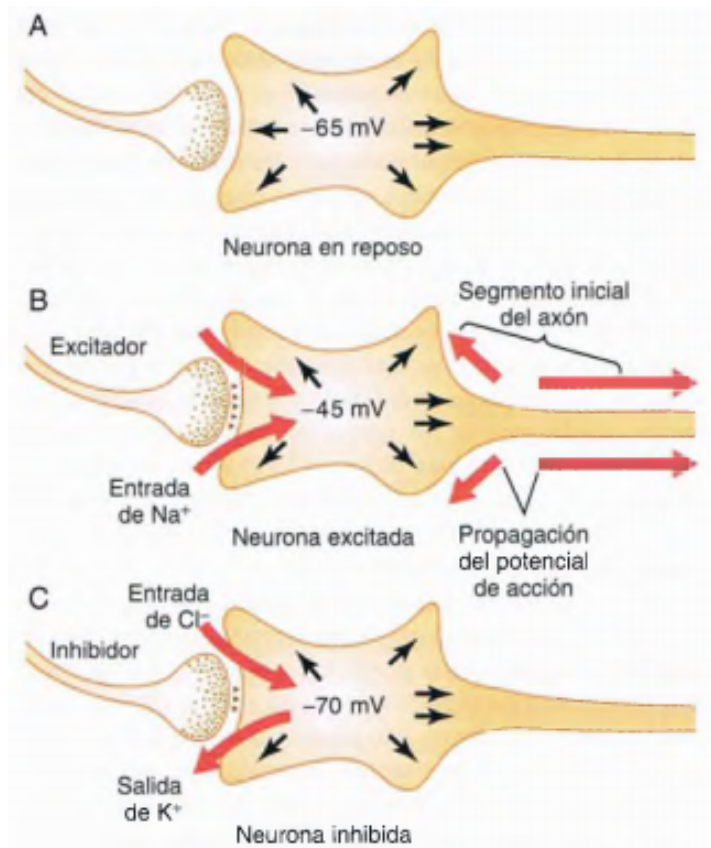


Figura 1.8: Tres estados de una neurona . **A.** Neurona en *reposo*, con potencial intraneuronal normal de -65mV . **B.** Neurona en estado *excitado*, con un potencial intraneuronal menos negativo de -45mV , ocasionado por la entrada de Sodio. **C.** Neurona en estado *inhibido*, con un potencial intraneuronal más negativo de -70mV , ocasionado por la salida del ion Potasio. [7]

1.3. Electroencefalografía

Como se expuso en la sección anterior, la actividad eléctrica del cerebro está formada por biopotenciales que se originan en la membrana externa de las células con capacidad de excitación, como lo son aquellas que componen el tejido nervioso o muscular. Éstas señales eléctricas son de naturaleza iónica.

El Electroencefalograma (EEG) consiste en el registro de la actividad eléctrica cerebral empleando electrodos aplicados sobre el cuero cabelludo (Figura 1.9). En consecuencia, el EEG es el resultado de la suma temporal y espacial de los potenciales postsinápticos procedentes de las neuronas en la corteza cerebral. La actividad del EEG tiene una amplitud pequeña, del orden de $5\mu\text{V}$ a $200\mu\text{V}$, lo que implica que estas señales son difíciles de registrar y pueden distorsionarse con facilidad.

Además, la actividad eléctrica del cerebro debe involucrar de forma simultánea una zona de corteza relativamente extensa (aproximadamente 6 cm^2) para que los potenciales cerebrales puedan ser registrados en el cuero cabelludo [6].



Figura 1.9: Electroencefalograma. ⁵

Como toda señal eléctrica, la actividad cerebral posee dos características medibles esenciales: la amplitud y la frecuencia. La frecuencia es una medida de la rapidez de cambio de la onda, y se mide en Hertz. La amplitud es una medida de la diferencia de potencial, medida en voltios, la cual aumenta o disminuye dependiendo del número de PPS que son generados al mismo tiempo y con la misma dirección (se hace referencia a una dirección eléctrica, excitatoria o inhibitoria, que es dependiente de neurotransmisores y sus receptores de membrana). Así la actividad sincrónica de un gran número de neuronas produce señales eléctricas de mayor amplitud y baja frecuencia. Por el contrario, cuando las neuronas sincronizadas son menos y se encuentran en una región más localizada, se manifiestan a mayores frecuencias y con menores amplitudes.

1.3.1. Electroodos

Los sensores que son aplicados sobre el cuero cabelludo o el cerebro con el fin de registrar los potenciales eléctricos neuronales, constituyen lo que se denomina electrodos de registro. Los electrodos pasan a cumplir entonces la función de transductores y detectan la actividad eléctrica que se propaga a través del tejido circundante, convirtiendo las corrientes iónicas en corrientes electrónicas.

Dentro de las técnicas que emplean la EEG para detectar la actividad eléctrica del cerebro podemos nombrar las siguientes (Figura 1.10):

Electroencefalografía Profunda: se utilizan micro-electrodos implantados en el interior del cerebro mediante una intervención quirúrgica.

Electrocorticograma (ECoG): previa intervención quirúrgica para colocar los electrodos directamente sobre la corteza cerebral.

⁵Fuente: <https://www.electromiografiamexico.com/estudios-de-electroencefalograma-en-mexico/>

Electroencefalograma (EEG): utiliza electrodos superficiales que son colocados sobre la superficie del cuero cabelludo. Se aplicada de manera no invasiva y puede ser utilizada por períodos de tiempo prolongados sin implicar ningún riesgo para el paciente.

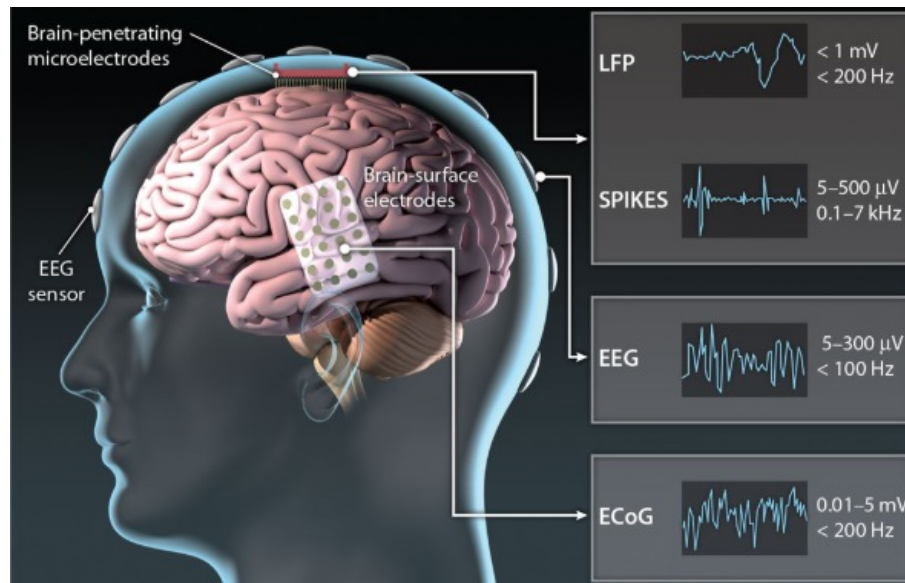


Figura 1.10: Distintas técnicas de EEG, entre ellas: Electroencefalograma (EEG) con electrodos ubicados sobre el cuero cabelludo, Electrocoriografía (ECoG) con electrodos ubicados sobre la corteza cerebral, y microelectrodos implantados en el cerebro. ⁶

Existen muchos tipos de electrodos del cuero cabelludo utilizados para registros electroencefalográficos (Figura 1.11). Los electrodos de plata fina (99.99% de pureza), con o sin capa superficial de plata clorurada, se consideran como los más recomendables para registros no invasivos [6].

Si los registros son tomados más cerca del cerebro reflejando la actividad de un menor número de neuronas, se obtiene una señal con mejor relación señal-ruido y una mejor resolución espacial. En cambio, a medida que los registros son tomados a una mayor distancia reflejando la actividad eléctrica de un gran número de neuronas, las señales obtenidas presentan una menor resolución espacial y relación señal ruido.

⁶Fuente: <http://www.electricalibrary.com/en/2019/03/22/brain-machine-interface>. Brain-machine interface - Pedro Ney Stroski

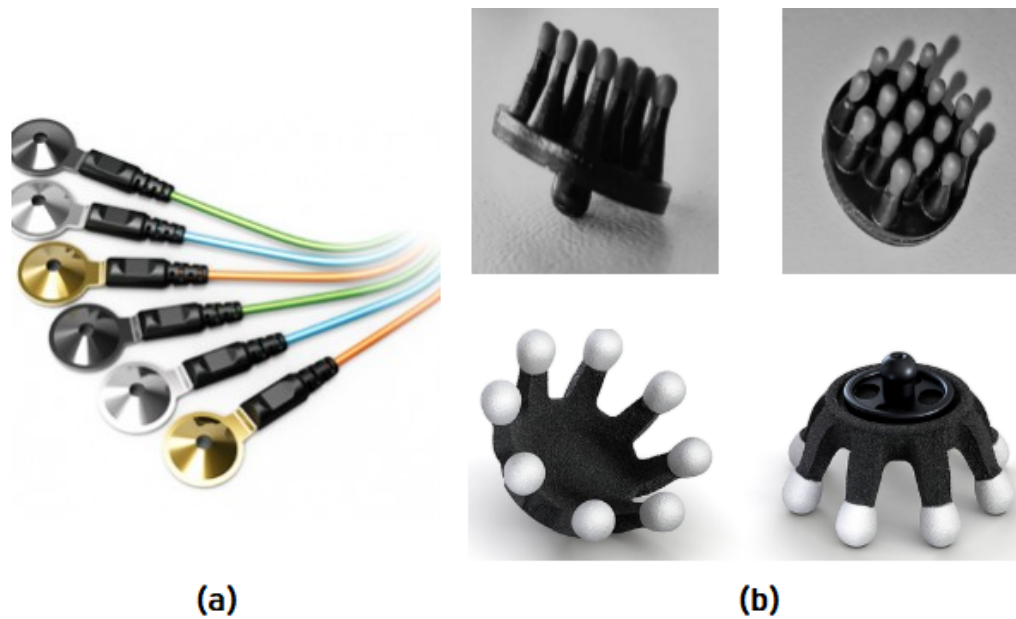


Figura 1.11: Ejemplos de electrodos utilizados en Electroencefalografía superficial no invasiva. A la izquierda (a), se observan electrodos de tipo húmedo de distintos materiales que precisan gel electrolítico para su funcionamiento. A la derecha (b), electrodos secos de plata con cobertura de cloruro de plata, cuya forma ayuda a atravesar el cabello y aumentar la superficie que se encuentra en contacto con el cuero cabelludo.

1.3.2. Sistema 10-20

Para lograr repetitividad en las mediciones, es de gran importancia la utilización de un sistema estandarizado para la ubicación de los electrodos. El sistema de posicionamiento de electrodos más utilizado es el “Sistema Internacional de Posicionamiento de Electrodos 10-20” (Figura 1.12), o bien “Sistema 10-20” recomendado por la Federación Internacional de Sociedades de Electroencefalografía y Neurofisiología Clínica.

Contar con un sistema de posicionamiento permite normalizar la colocación de electrodos y establecer comparaciones de los resultados obtenidos, aún cuando son realizados con distintos equipos y protocolos experimentales.

Se debe tener en cuenta que en los registros de EEG es indispensable la posición de los electrodos debido a la somatotopía del cerebro, es decir, la correspondencia punto a punto de un área del cuerpo con un área específica del sistema nervioso central.

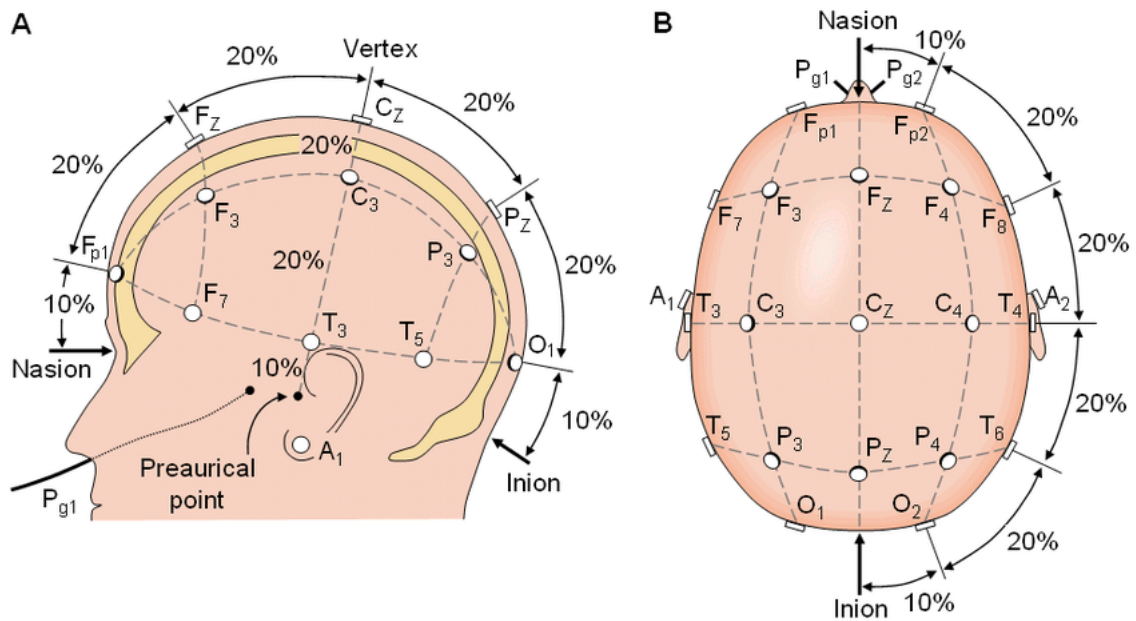


Figura 1.12: Sistema Internacional 10-20 para la colocación de los electrodos extracra-neales. Las letras señalan el área (**Fp**, prefrontal; **F**, frontal; **C**, central; **P**, parietal; **T**, temporal y **O**, occipital), mientras que los números designan el hemisferio (pares hemisferio derecho, impares hemisferio izquierdo). Los electrodos de la línea media se señalan con una "z", por lo que **Fz** se encuentra frontalmente en la línea media. ⁷

1.3.3. Patrones Asociados a Funciones Cerebrales

En los estudios de EEG interesa reconocer patrones asociados a distintas funciones del cerebro, ya sea para diagnóstico o desarrollo de tecnología aplicada al cerebro. Dos de los patrones más utilizados para estos fines son:

1.3.3.1. Ritmos Cerebrales

Los ritmos cerebrales se definen como ondas regulares a lo largo del tiempo que se caracterizan por su frecuencia, localización y asociación con varios aspectos del funcionamiento y el estado del cerebro (Figura 1.13). Dicha actividad se produce o inhibe según los diferentes mecanismos o estados del cerebro (en vigilia, en el sueño, en el estado de coma o en determinadas acciones del sujeto).

Si bien cada ritmo tiene sus características específicas, existen otras que son comunes a todos ellos:

- Los ritmos son bloqueados o atenuados mientras se producen o preparan eventos que afectan la zona cortical involucrada. A este fenómeno se lo conoce como "desincronización relacionada al evento", ERD (Event Related Desynchronization).

⁷Recuperado de <https://medium.com/@neurofeedback/protocolos-y-fundamentos-basicos-del-neurofeedback-d509d8b9ab1f>

- Luego del evento que produjo el ERD, suele reaparecer el ritmo intensificado. A este fenómeno se lo denomina “sincronización relacionada al evento”, ERS (Event Related Synchronization).[13]

Para poder diferenciar los ritmos se han dividido según bandas de frecuencias que ocupan, asignando una letra griega (α , β , δ , θ , γ) a cada uno, y estos pueden observarse en la Figura 1.13.

Banda Gamma (γ): se define como el grupo de ondas que oscilan en el rango de 25 a 100 Hz. Es un tipo de onda que se origina en el tálamo y se mueve desde la parte posterior del cerebro hacia adelante y a una velocidad muy rápida. Se relacionan con tareas de un alto procesamiento cognitivo y la fase del sueño REM, que presentan una alta actividad en este rango de frecuencias.

Banda Beta (β): es una banda que posee rangos variados, se considera la actividad que oscila entre los 12 y 30 Hz y que no supera los 25 μV en el EEG. Se localiza especialmente en regiones frontales y se la relaciona con actividad cerebral, especialmente con tareas cognitivas o especializadas.

Banda Alfa (α): Es aquella banda que agrupa los ritmos de 7-13 Hz, con voltajes que no suelen pasar los 50 μV en adultos en vigilia, con ojos cerrados, y que se atenúa con la apertura de los ojos o con actividad cognitiva. Así, el ritmo α no sólo agrupa un conjunto de frecuencias, sino que se trata de un tipo de actividad cerebral que se da en circunstancias específicas, como lo es el bloqueo del estímulo visual. Esto quiere decir que se trata de un *ritmo funcional* al que se denominó **ritmo de descanso**.

Banda Theta (θ): Se define como el grupo de ondas que oscilan entre 4-7.9 Hz, usualmente con una amplitud de 30-60 μV y junto con los ritmos Delta (δ) son usualmente llamadas ondas lentas. Las ondas θ han sido relacionadas particularmente con actividad límbica, especialmente hipocampal, aquella que está relacionada con funciones espaciales (movimiento, orientación), cognitivas (memoria, atención) y emocionales (ansiedad, defensa).

Banda Delta (δ): Se define como el grupo de ondas en el rango de frecuencias entre 0.5 y 4 Hz. Su actividad, a pesar de siempre estar presente, es de muy baja amplitud y se observa en pacientes mayores de 4 años en estado de vigilia. No suele ser visible en el electroencefalograma normal pero son un rasgo esencial de las etapas del sueño III y IV, también llamadas etapas del sueño de ondas lentas.

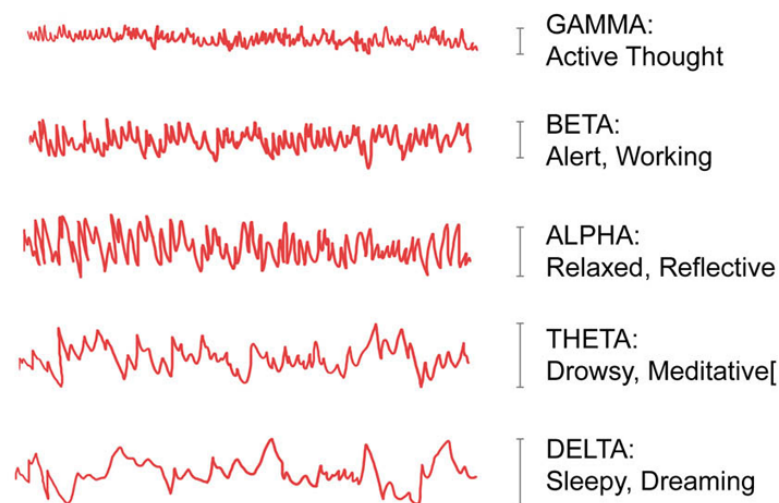


Figura 1.13: Representación de las distintas ondas cerebrales y sus características ⁸

Además de los mencionados, en los últimos tiempos se ha descubierto que existen otros ritmos cerebrales que comparten bandas de frecuencias y se diferencian entre sí según características como su localización o funciones del cerebro asociadas.

1.3.3.2. Potenciales Evocados

Un potencial evocado (PE) es una respuesta del sistema nervioso central o periférico a un estímulo y los mismos pueden realizarse por tres vías: la visual, la auditiva y la táctil, pero también puede ser la respuesta a un evento interno como una actividad cognitiva (atención, la preparación de un patrón motor, etc.). Según la vía escogida reciben distintos nombres, Potenciales evocados visuales (PEV), Potenciales evocados auditivos (PEA), Potenciales evocados somatosensoriales (PESS) o Potenciales evocados cognitivos (PEC).

El análisis de potenciales evocados es una prueba que se utiliza para estudiar la actividad eléctrica cerebral frente a estímulos externos. Cada uno de estos estímulos se recoge por un órgano sensorial diferente, donde se traducen a impulsos eléctricos que viajan a través de las neuronas hasta llegar a áreas específicas del cerebro, donde son interpretados.

Estos estudios permiten identificar lesiones neurológicas en casi cualquier zona del sistema nervioso; además son especialmente útiles en pacientes que, por diferentes motivos, no pueden expresar si son capaces de notar estímulos externos.

Dentro de los potenciales evocados cognitivos más conocidos podemos nombrar:

- **Onda P300 (EP300, P3):** es un potencial evocado que puede ser registrado mediante electroencefalografía como una deflexión positiva de voltaje con una latencia de unos 300ms en el EEG. La presencia, magnitud, topografía y duración de esta señal se utiliza a veces en la medición de la función cognitiva de los procesos de toma de decisiones.

⁸Fuente: Ondas Cerebrales, Estrés y Biomagnetismo Emocional - Sant Pau Centre Terapèutic

- **Onda N100:** es un potencial que aparece como una onda negativa aproximadamente a los 100 ms y la misma parece estar asociada a la atención.
- **Onda N400:** se trata de una onda negativa, con topografía centro-parietal y latencia en torno a los 400ms, considerada un indicador del procesamiento semántico.

1.4. Componentes Eléctricos no Dependientes de la Actividad Cerebral: Artefactos

El primer paso para la interpretación de un electroencefalograma es lograr separar la actividad eléctrica de origen cerebral (EEG) de todo un conjunto de actividades no electroencefalográficas presentes en el registro. Estas actividades externas a la fuente originaria de la señal son denominadas en su conjunto *artefactos*, y causan variaciones no deseadas en la misma pudiendo incluso llegar a modificarla.

La identificación de artefactos constituye una importante tarea previa al estudio del EEG por los siguientes motivos:

- Los artefactos se encuentran superpuestos a la actividad encefalográfica y pueden ocultarla dificultando su valoración.
- Pueden simular fenómenos electroencefalográficos fisiológicos o asociados a patologías, por lo que pueden inducir a errores de interpretación.
- Debido a la interferencia eléctrica de otros dispositivos como también las necesidades especiales del estudio, resulta difícil la eliminación de artefactos e interpretación adecuada del registro.

En un principio los artefactos solo podían corregirse durante el registro (mediante modificación de las características de filtrado, corrección de posición de electrodos o instrucciones al paciente). La introducción de sistemas de registro digitales permitió una gama nueva de tratamiento y eliminación de artefactos con más posibilidades de actuación por parte de la persona que interpreta el registro EEG. En los registros digitales existen métodos para detectar e incluso eliminar de manera automática artefactos, lo que puede mejorar el análisis del registro de la actividad cerebral.[6]

Los artefactos se clasifican en dos categorías según estén producidos por interferencia eléctrica ambiental o estén asociados con actividades fisiológicas de origen no cerebral. A continuación se detallan de manera breve cuales son los artefactos más frecuentes en la práctica electroencefalográfica:

Movimientos oculares (Figura 1.15): al parpadear, en el seguimiento de objetos o incluso cuando se mantienen los párpados cerrados se produce un movimiento ocular que provoca ondas en las regiones frontales, y pueden extenderse a regiones centrales. El potencial de los movimientos oculares disminuye de amplitud desde la región anterior a la posterior.

Artefacto Muscular (Figura 1.14): la actividad eléctrica debida a la contracción de varios músculos del cuero cabelludo, la cara, la mandíbula o el cuello pueden ser registrados en el EEG, y constituyen artefactos musculares. Son potenciales de muy corta duración aislados o en series, generalmente captados en electrodos localizados en regiones temporales y frontales aunque también pueden captarse en otras zonas. Para eliminar este artefacto se solicita al paciente el mantenimiento de una actitud relajada, la apertura ligera de la boca o un cambio de posición que atenúe la contracción.

Electrocardiograma (ECG) (Figura 1.16): son ondas agudas que pueden confundirse con elementos epilépticos. En ausencia de un método específicos de detección, el carácter rítmico de este artefacto permite su identificación. Un usuario con conocimientos técnicos podría identificar (aunque no eliminar) este artefacto mediante el control de pulso del paciente.

Artefactos de movimiento (Figura 1.17): los electrodos funcionan como una interfaz entre el paciente y el sistema de registro. Cualquier modificación en esta interfaz, como la asociada a movimientos, es amplificada por el sistema y aparece como artefacto. Son ondas irregulares de gran voltaje, que puede afectar a uno o todos los canales de medición. A menudo puede evitarse el artefacto solicitando al paciente que mantenga la postura o mejorando la posición de los electrodos.

Artefactos por sistemas eléctricos (Figura 1.18): algunos son fáciles de reconocer por su periodicidad y la regularidad de su forma, como por ejemplo los ruidos de línea que pueden aparecer a una frecuencia de 50 o 60Hz. Los equipos modernos de electroencefalografía disponen de un filtrado selectivo para la eliminación de la actividad a esta frecuencia que no modifica de forma significativa otras actividades del cerebro. También pueden aparecer interferencias eléctricas relacionadas con equipos externos, como pueden ser motores o equipos que se encuentran relativamente cerca y que se atenúan mediante una adecuada derivación a tierra del sistema de registro y el mantenimiento de una impedancia correcta de los electrodos.

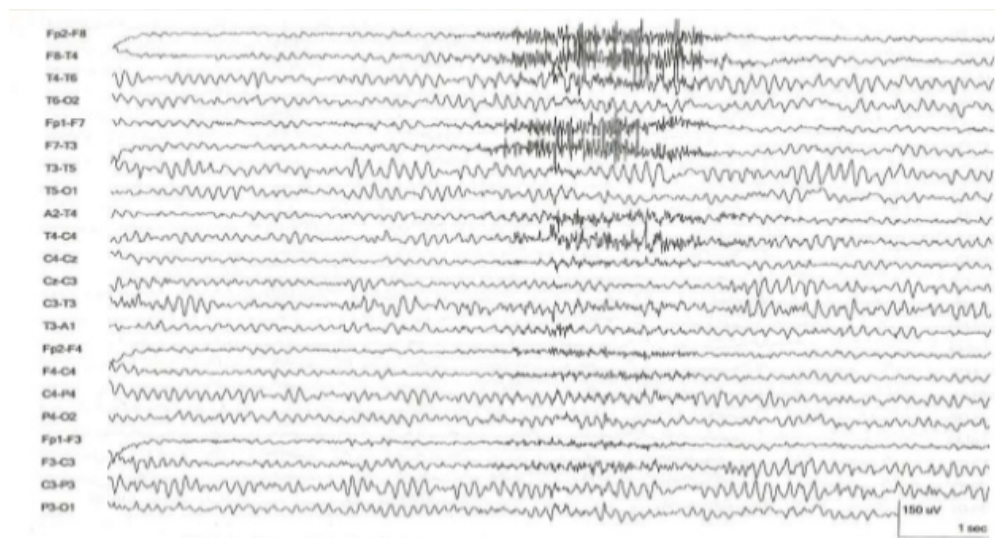


Figura 1.14: Artefactos musculares en registro de EEG⁹

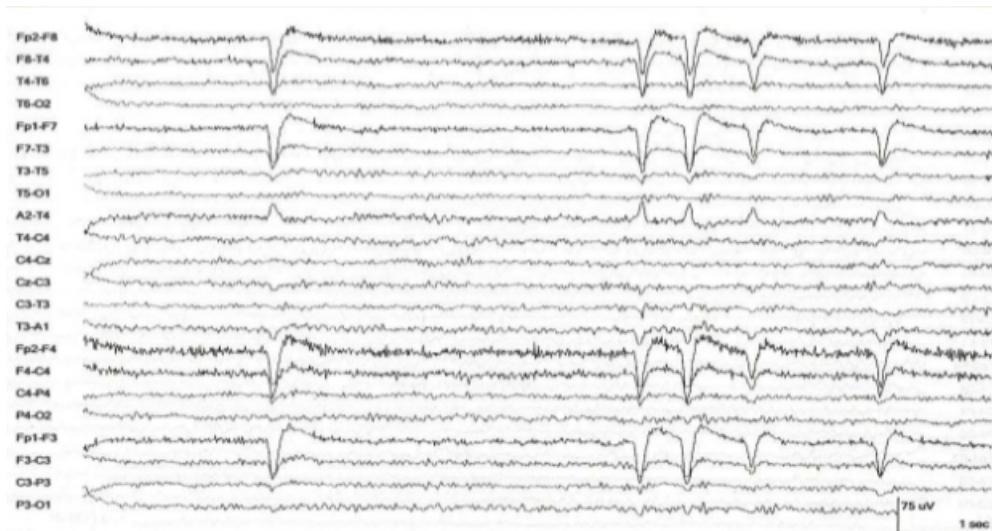


Figura 1.15: Artefactos de pestañeo en registro de EEG⁹

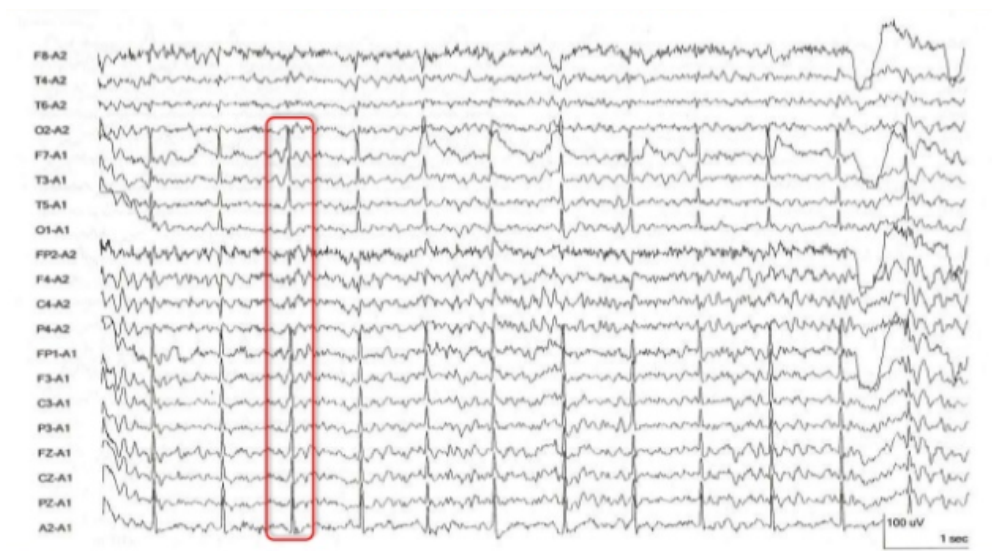


Figura 1.16: Artefactos cardíacos en registro de EEG.⁹

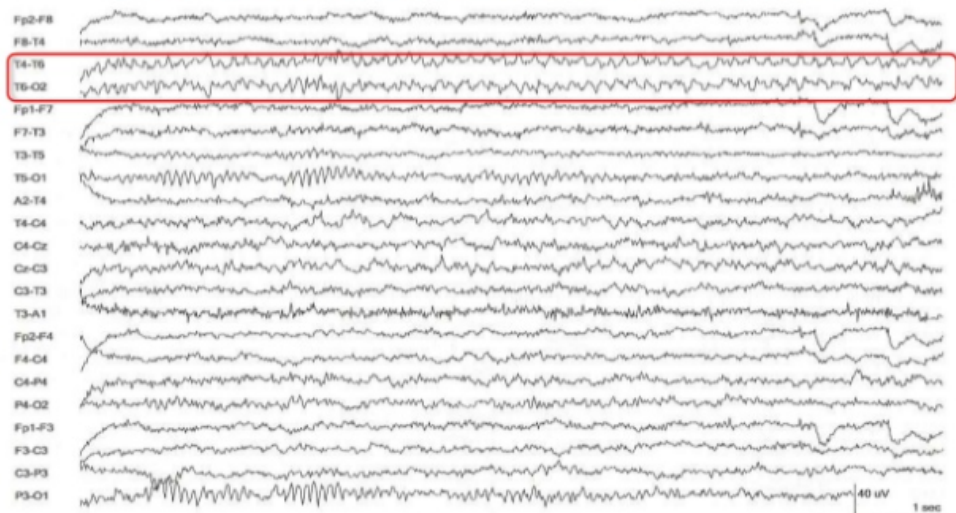


Figura 1.17: Artefactos de movimiento en registro de EEG⁹

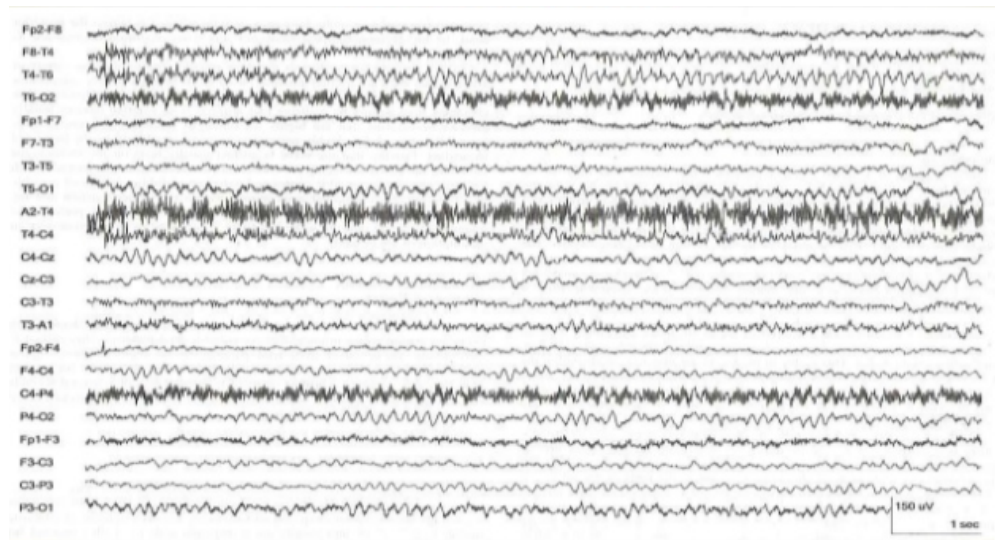


Figura 1.18: Artefactos de interferencia eléctrica a 60Hz en registro de EEG⁹

⁹Fuente: EEG Artifacts - Dr. Sudhakar Marella

Capítulo 2

Descripción de los Métodos Implementados para el Procesamiento de las Señales de EEG

Como se mencionó en el capítulo anterior, la señal obtenida en un electroencefalograma es muy débil (con valores entre 0-200 μV) y está contaminada por una gran variedad de artefactos: tensión de línea eléctrica, acoplamientos capacitivos, otras señales bioeléctricas, etc.

Estos artefactos poseen una amplitud considerablemente mayor a la de la actividad electroencefalográfica, por lo que aparecen más destacados en el registro de la señal, superponiéndose y ocultándola, dificultando así su estudio. Un ejemplo es la actividad eléctrica muscular que presenta potenciales de entre 50 μV y 30 mV, en un rango de frecuencias alrededor de 50 a 150 Hz.

Debido a esto, resulta necesario implementar métodos de procesamiento de señales que permitan detectar y atenuar (o eliminar) el efecto de estos artefactos sobre la señal, como así también resaltar características que resultan de interés para el análisis.

A continuación se describen de manera detallada los métodos estudiados e implementados en el desarrollo de la herramienta.

2.1. Filtros

La etapa de filtrado de una señal de EEG consiste en la aplicación de procesos que destaquen la información que resulta notable para el análisis, y eliminen aquella proveniente de fuentes no deseables. Las características que se buscan destacar o eliminar son componentes de la señal a distintas frecuencias, por lo que conociendo las frecuencias de interés se pueden eliminar todas aquellas que no pertenezcan a dichas bandas.

El filtrado puede desarrollarse a través de hardware denominado Filtrado Analógico, o a través de software denominado Filtrado Digital. El Filtrado Analógico utiliza dispositivos electrónicos para lograr el procesamiento de la señal que luego es digitalizada. En

cambio, en el Filtrado Digital se realiza un procesamiento matemático sobre la señal una vez que la misma fue digitalizada .

Como puede observarse en la Figura 2.1, los filtros se dividen en:

- **Filtro Pasa Alto:** Mantiene la información a frecuencias superiores a un cierto valor, y elimina toda la información que se encuentra a frecuencias inferiores.
- **Filtro Pasa Bajo:** Opuesto al anterior; mantiene la información a frecuencias inferiores a un cierto valor y elimina toda la información que se encuentra a frecuencias superiores.
- **Filtro Pasa Banda:** Mantiene la información dentro de una Banda de Frecuencias entre dos valores extremos y elimina toda información que se encuentra por fuera de la banda seleccionada.
- **Filtro Elimina Banda:** Mantiene la información por fuera de una Banda de Frecuencias entre dos valores extremos y elimina toda la información que se encuentre dentro de la banda seleccionada.

Los valores que definen los extremos entre los cuales se eliminan o se aceptan bandas de frecuencia se denominan **frecuencias de corte** (F_c).

En el presente proyecto se desarrolla una herramienta exclusivamente en software, por lo que sólo se implementan métodos de Filtrado Digital.

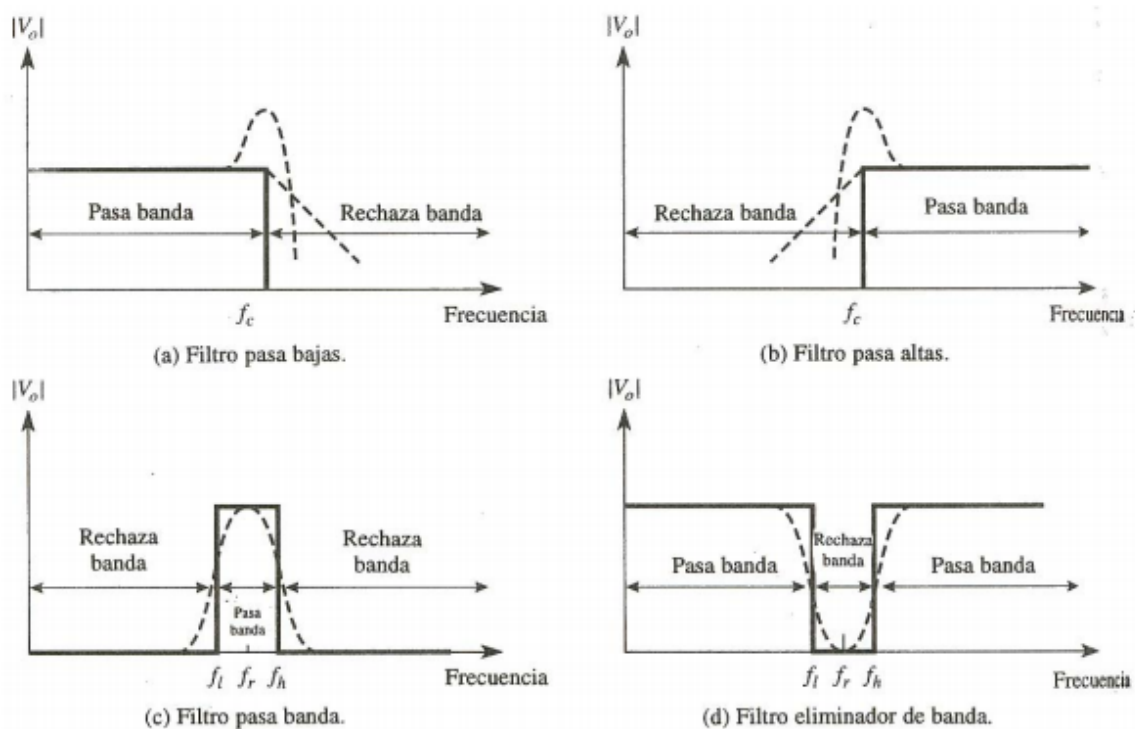


Figura 2.1: Distintos tipos de filtros. En línea continua se visualiza el filtrado ideal, y en línea de puntos el filtrado real que puede ser obtenido tanto digital como analógicamente.

1

¹Fuente: Filtros Activos- Académico - Tecnológico Nacional de México

2.1.1. Caracterización de un Filtro

Son tres las características utilizadas para definir un filtro. Cada una de ellas contiene información completa sobre dicho filtro y si se define un filtro a partir de una de ellas, las otras dos quedan definidas a partir de la primera. Las mismas son: *Respuesta al Impulso*, *Respuesta al Escalón* y *Respuesta en Frecuencia*. [4]

2.1.1.1. Respuesta al Impulso

Una Señal Impulso consta de una muestra de amplitud uno seguida del resto de valor cero, como puede verse en la Figura 2.2. Esta señal se caracteriza por un espectro constante en todas las frecuencias.

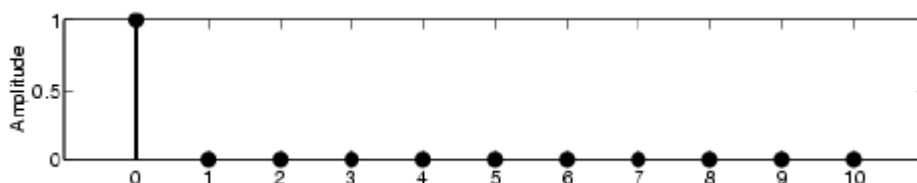


Figura 2.2: Ejemplo de una Señal Impulso, donde la primer muestra tiene amplitud de valor uno y el resto cero. [8]

Si aplicamos el filtro a la señal impulso, obtenemos a la salida la característica Respuesta al Impulso (IR). Es a través de esta respuesta que se clasifican los filtros en no recursivos (de respuesta finita al escalón, FIR) y en filtros recursivos (de respuesta infinita al escalón, IIR). Sin embargo, la principal importancia de esta característica radica en que podemos entender a la aplicación de dicho filtro como la convolución entre la señal a filtrar y la IR del filtro[4]:

$$y(n) = h(n) * x(n),$$

donde $y(n)$ refiere a la señal filtrada, $h(n)$ a la respuesta al impulso del filtro, y $x(n)$ la señal a filtrar, y $*$ representa la operación de convolución.

Esta aplicación le aporta la utilidad más grande a la característica de IR, ya que una de las maneras más comunes y simples de aplicar un filtro es a través de esta operación de convolución.

2.1.1.2. Respuesta al Escalón

La Respuesta al Escalón (SR) es una característica similar a la anterior pero resultante de aplicar el filtro a una señal tipo escalón, que consiste en una señal de amplitud cero hasta un número de muestra a partir del cual empieza a valer uno. Esta señal puede también obtenerse como la integral del impulso desde la muestra en que cambia la amplitud hasta el infinito.

La SR es muy importante para definir la capacidad de segmentación del filtro, o sea, la posibilidad de distinguir una muestra de la otra y que tan diferenciado será el filtrado. Si la respuesta al escalón es muy lenta, se filtrará información importante que esté cercana

a datos que se querían eliminar, en cambio si es muy rápida, puede agregar distorsión por *Overshooting* a la señal de salida. [4]

2.1.1.3. Respuesta en Frecuencia

Otra de las características importantes para definir un filtro es la Respuesta en Frecuencia (FR). Ésta consiste en el estudio de las modificaciones que produce el filtro en la señal, en Amplitud y en Fase.

Como se puede ver en la Figura 2.3, la FR se visualiza en los denominados Diagramas de Bode, que muestran la Respuesta en Amplitud (en decibelios) y la Respuesta en Fase en función de la frecuencia de la señal.

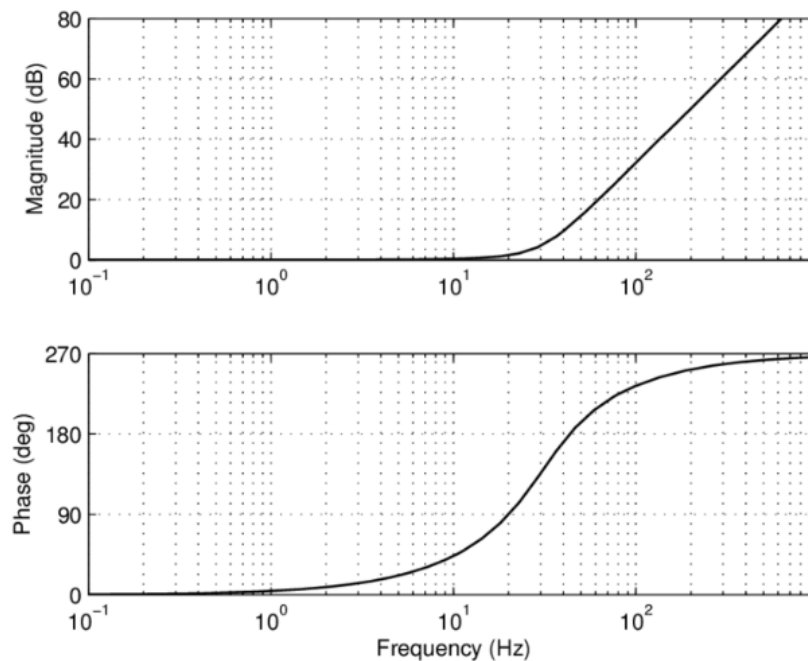


Figura 2.3: Diagramas de Bode que muestran la Respuesta en Frecuencia (FR) de un filtro pasaalto a una frecuencia de corte de 30 Hz. ²

Junto con la IR, la FR es una de las características mas utilizadas al momento de definir un filtro, ya que permite visualizar claramente los efectos que ese filtro produce en la señal en el dominio de la frecuencia. Además, la FR puede obtenerse como la Transformada de Fourier de la IR (Figura 2.4)

²Fuente: Juan Carrion, Billie F. Spencer. "Model-based Strategies for Real-time Hybrid Testing." (2007)

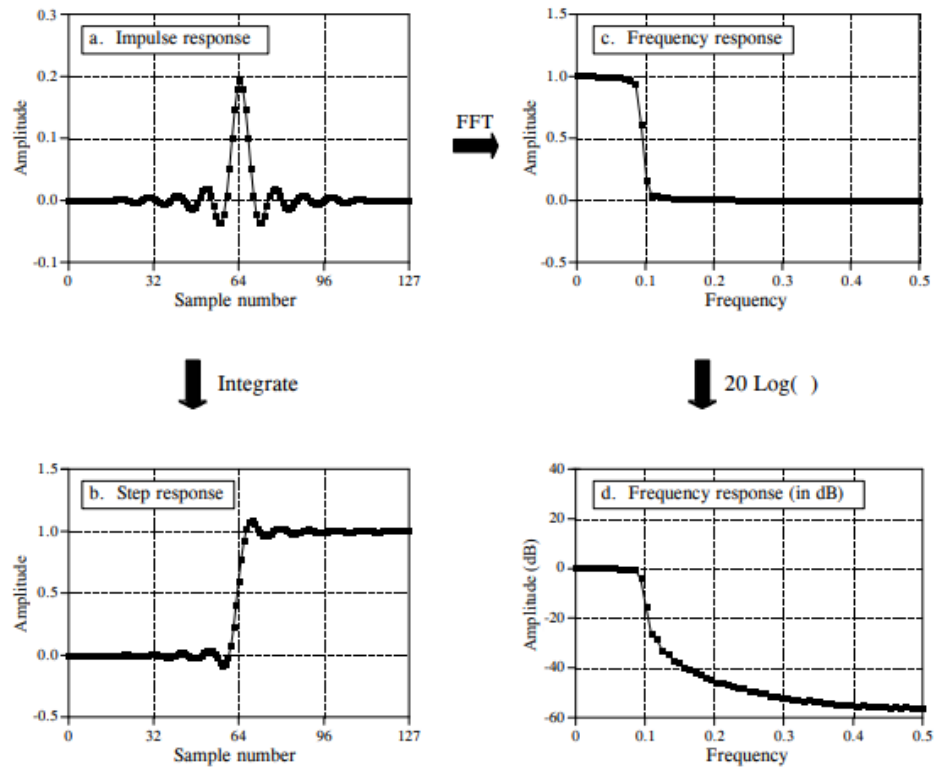


Figura 2.4: Comparación de las distintas características que definen a un filtro [4]

2.1.2. Orden de un Filtro

El orden de un filtro describe el grado con que el filtro acepta o rechaza frecuencias cercanas a las frecuencias de corte. Mientras mayor sea el orden, mayor será la atenuación de frecuencias cercanas a la frecuencia de corte no deseadas. Para lograr una mayor atenuación se requiere una mayor cantidad de etapas de circuito (en filtros analógicos) o etapas de cálculo (en filtros digitales). Cada nivel de orden corresponde a un aumento de 20 dB de atenuación de amplitud por cada década de frecuencia (Figura 2.5). Sin embargo, a medida que se aumenta el orden de un filtro, aumenta el tiempo de cómputo, y por lo tanto el retraso entre la entrada y la salida de dicho filtro.

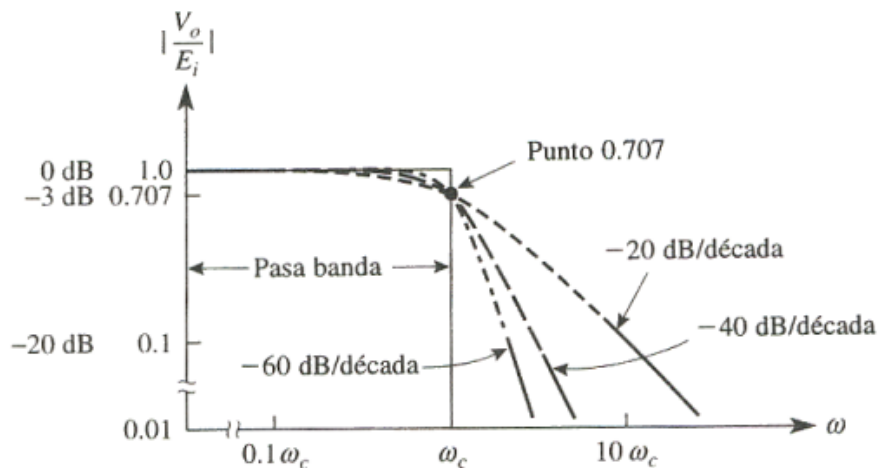


Figura 2.5: Respuesta en frecuencia de filtros Pasabajos Butterworth de distinto orden. ³

2.1.3. Tipos de Filtros

Los filtros se clasifican según su respuesta al impulso (IR) en Filtros de Respuesta Finita al Impulso (FIR) y Filtros de Respuesta Infinita al Impulso (IIR). En los primeros, la IR llega a atenuarse en algún momento, en cambio en los de respuesta infinita esto no sucede (Figura 2.6).

Los filtros FIR no son recursivos (la salida depende únicamente de la entrada en ese instante) por lo tanto son siempre estables. Esto significa que la salida nunca puede aumentar infinitamente si la entrada no lo hace.

Los filtros IIR en cambio poseen partes recursivas (la salida depende de la entrada en ese instante pero también de la salida en instantes anteriores), por lo tanto pueden o no ser estables. Esto significa que ante ciertas señales de entrada estables, la salida del filtro puede aumentar infinitamente.[8]

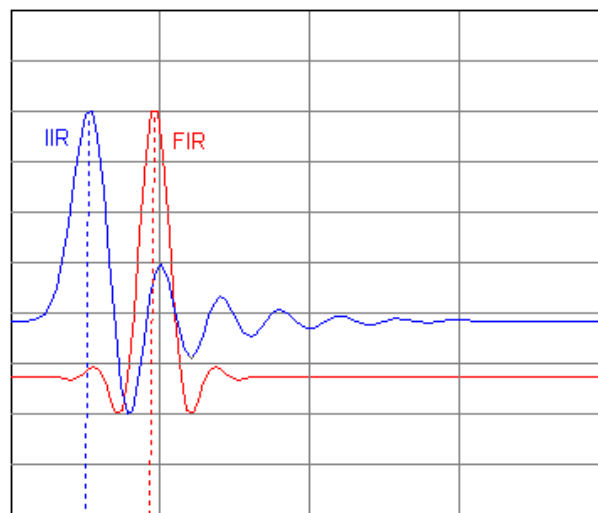


Figura 2.6: Comparación entre filtros FIR e IIR. ⁴

Una ventaja de los IIR es que logran los mismos efectos que los FIR con órdenes de filtro notablemente menores. Los IIR logran atenuaciones de frecuencia similares a los FIR con menores tiempos de cómputo y menor desfase entre la entrada y la salida de los mismos. Sin embargo, el retraso generado por los filtros IIR no es constante ni lineal, por lo que raramente puede predecirse. Esto puede representar una gran desventaja en aplicaciones que requieren la coordinación de eventos temporales de señales.

Se presenta a modo de resumen una tabla comparativa entre filtros FIR e IIR:

Característica	Filtros FIR	Filtros IIR
Velocidad Computacional	Lentos	Rápidos
Retraso/Desfase	Constante	No constante y no lineal
Estabilidad	Siempre estables	Pueden desestabilizarse

Cuadro 2.1: Comparativa entre Filtros FIR e IIR

³Fuente: Filtros Activos - Académico - Tecnológico Nacional de México

⁴Fuente: Difference Between FIR and IIR Filters - Iowa Hills Software

En el procesamiento de señales de EEG se utilizan generalmente filtros IIR, aprovechando la mayor velocidad computacional de los mismos mientras que sus desventajas no afectan de manera notable al análisis. Esto se debe a que:

- Las señales de EEG poseen amplitudes relativamente bajas y por lo tanto pocas veces desestabilizarían la salida del filtro.
- El filtro es aplicado en todos los canales y si bien el desfase no es lineal ni constante, es el mismo en cada uno de ellos.

En cambio, los filtros FIR son preferibles cuando se deben estudiar las señales en coordinación con algún evento externo a la misma. Por ejemplo, al intentar comparar una onda que representa las instancias de estimulación de un sujeto con la señal EEG resultante, en busca de patrones de respuesta.

El uso del paradigma P300 estaría comprendido entre uno de los casos en los que se debe estudiar la señal en coordinación con la onda de estimulación. Sin embargo, incluso en estudios de P300 suelen utilizarse filtros IIR ya que el rango de búsqueda de la onda de P300 es relativamente amplio, por lo que el desfase no constante ni lineal no constituye un problema.

2.1.4. Ejemplos de Filtros

Uno de los filtros más utilizados en el procesamiento digital de señales de EEG es el Chebyshev (referido en bibliografía como Chebychev, Tschebyscheff, Tchebysheff o Tchebichef). Existen diferentes tipos de filtros Chebyshev y éstos se diferencian por el balance entre un nivel de *ripple* (distorsiones en la señal de salida causadas por el uso de sistemas no lineales) y una mejor definición de la frecuencia de corte, denominada *bajada o roll-off* (Figura 2.8). Esto quiere decir que a mayor ripple se logra una mejor bajada, tolerando cierta distorsión en la señal. El filtro Chebyshev con un ripple de 0% es conocido también como Filtro Butterworth y el mismo está diseñado para producir una respuesta más plana hasta la frecuencia de corte (Figura 2.7). Ambos filtros son implementados como filtros IIR y a continuación se detallan algunas de sus diferencias:

-En el Butterworth entre mayor sea el orden del filtro más plana será la banda de paso, en cambio en el Chebyshev entre mayor sea el orden mayores serán las fluctuaciones en la banda de paso .

-La Banda de Transición es mayor en el Butterworth comparada con el Chebyshev.

-El Butterworth posee un mejor dominio del tiempo por lo que su salida es mas estable, en cambio el Chebyshev presenta pequeñas distorsiones debido a las fluctuaciones en la banda de paso.

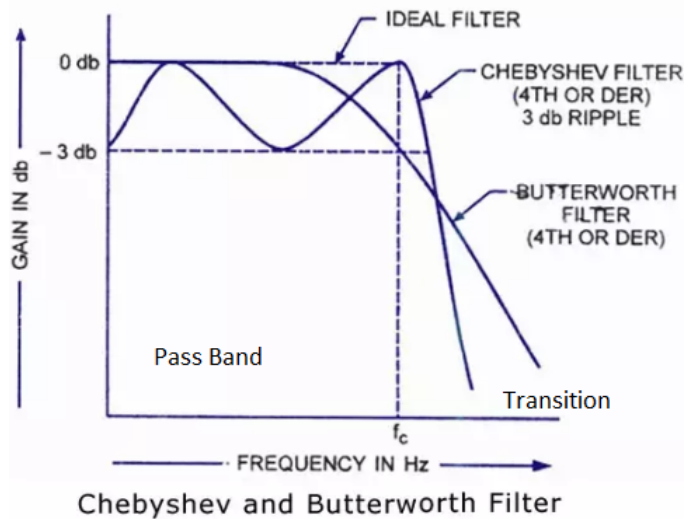


Figura 2.7: Filtro Butterworth vs Chebyshev.⁵

Como puede observarse en la Figura 2.7, la banda de paso en el Butterworth es más plana, esto quiere decir que a frecuencias menores a la Frecuencia de Corte (F_c) no se produce atenuación en la señal de salida. Por lo tanto son implementados en señales donde se requiere la menor distorsión posible.

En cambio en los Filtros Chebyshev, mientras mayor sea el orden y la presencia de ripple en la banda de paso, mayor será el factor de caída (roll-off). Por lo tanto son implementados cuando el contenido de frecuencia de la señal es más importante que un nivel constante de amplitud, ya que aumentar el ripple genera distorsiones.

Otro tipo de filtros son los denominados Filtros Notch (Figura 2.8). Estos son filtros elimina banda con ancho de banda muy pequeños (hasta 1 o 2 Hz) y son implementados en la eliminación de interferencias externas que posean frecuencias estables y estacionarias. Un ejemplo es el ruido electromagnético generado por las líneas de corriente eléctrica a 50 o 60Hz[4].

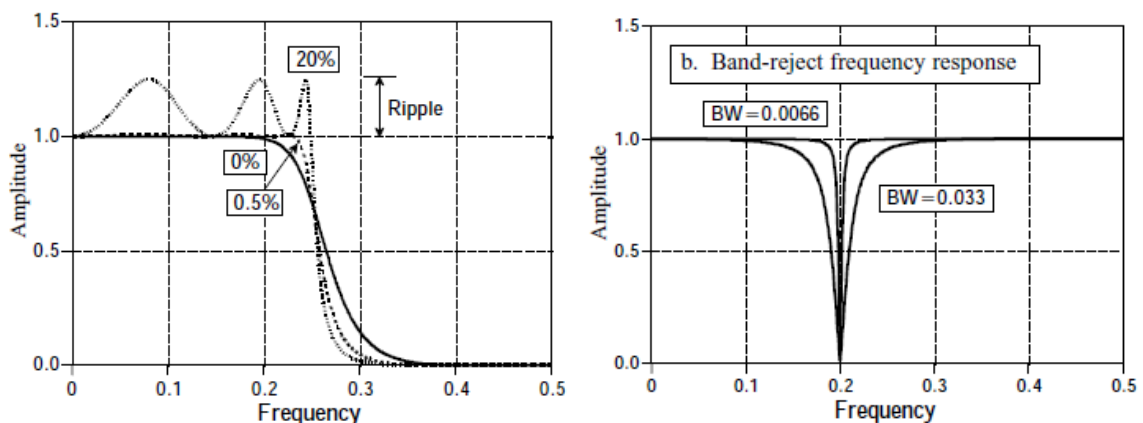


Figura 2.8: A la izquierda se representan distintos filtros Chebyshev con 0% (Butterworth), 0,5% y 20% de ripple . A la derecha se representan distintos filtros Notch con anchos de banda (BW) 0,0066 Hz y 0,033 Hz. [4]

⁵Fuente:Filter types and parameterization - Beckoff Information System. Último Ingreso : 18/07/19.

2.2. Rereferenciado Digital

Al adquirir una señal, cada electrodo se referencia generalmente a algún punto para obtener una diferencia de potencial. Este punto de referencia suele ubicarse en el lóbulo del oído (denominado *Referencia al oído*), también en el mentón, mastoides o en la zona frontal. Este electrodo proporciona un potencial de referencia constante y es utilizado para medir la diferencia de potencial de los demás electrodos ubicados en el cuero cabelludo respecto al mismo.

En las señales electroencefalográficas los artefactos aparecen en patrones espaciales específicos (Sección 1.4). Un artefacto puede ser común en algunas áreas del cuero cabelludo y, por lo tanto, ser captado por múltiples electrodos. Por ejemplo, si es común a dos electrodos, pero la señal deseada sólo es detectada por un electrodo, la diferencia entre las señales respectivas contendrá sólo la señal y no el ruido. Por lo tanto, las combinaciones de señales conocidas como "filtros espaciales" pueden eliminar o filtrar estos artefactos, considerando las distribuciones espaciales del mismo con respecto a los electrodos.

Existe una gran variedad de filtros espaciales, no obstante en este proyecto se implementaron aquellos con mayor utilidad en el área de la Electroencefalografía, y estos son:

- **Referencia Promedio Común:** CAR (Common Average Reference, por sus siglas en inglés) es utilizada para eliminar la información que es común en todas las lecturas simultáneas de los electrodos. CAR puede ser calculada mediante la resta entre el potencial de cada electrodo y el potencial promedio de todos los canales. Esto se repite para cada instante de tiempo, tal como se muestra en la siguiente fórmula:

$$V_i^{CAR} = V_i^{ER} - \frac{1}{n} \sum_{j=1}^n V_j^{ER}$$

Donde V_i^{ER} es el potencial entre el i -ésimo electrodo y la referencia y n es el número de electrodos en el montaje.

- **Laplaciano:** Es un filtro que mejora la actividad localizada mientras que suprime la actividad de difusión. La implementación del Laplaciano se utiliza para sustraer de un electrodo las mediciones de los electrodos circundantes al mismo, ponderadas según la distancia a la que se encuentren.

Existen dos tipos fundamentales de Filtros Laplacianos: Grande y Pequeño (*Large* y *Small*). La diferencia entre ellos reside en cómo se seleccionan los electrodos: para el Laplaciano Pequeño se seleccionan los electrodos vecinos más cercanos, mientras que para el Laplaciano Grande se seleccionan los electrodos vecinos más distantes (Figura 2.9). Si la señal de interés se encuentra menos localizada y/o varía en la ubicación exacta, es recomendable el uso del Laplaciano Grande.

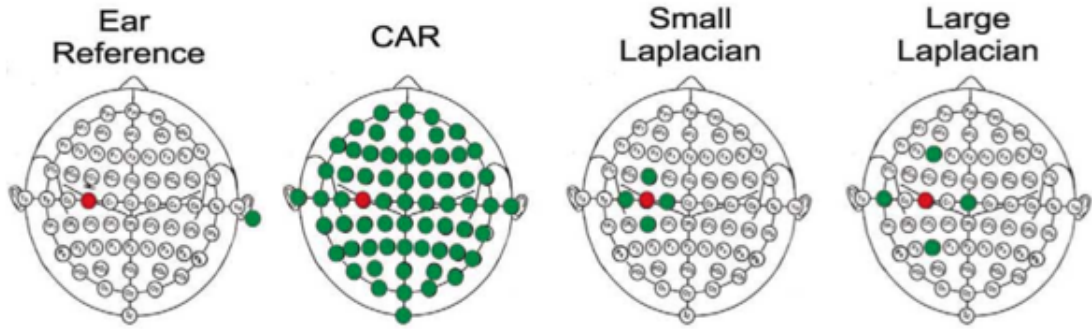


Figura 2.9: EEG para análisis clínico. Los electrodos verdes son tomados como referencia y extraídos para cada electrodo rojo según el filtro espacial utilizado.[3]

2.3. Transformada de Fourier

La Transformada de Fourier (TF) es ampliamente utilizada en el procesamiento de señales de EEG, especialmente en el estudio del comportamiento espectral de eventos (como ritmos cerebrales, crisis epilépticas, movimientos voluntarios, etc.). Un ejemplo es el caso de señales con ritmos cerebrales, donde la TF es utilizada para la caracterización espectral, separación de los diferentes ritmos y la estimación de las frecuencias dominantes de cada uno de ellos.[16]

Dada una señal $s(t)$ de energía finita en el tiempo, su transformada de Fourier es una función de la frecuencia (f), definida por:

$$TF \{s(t)\} = \hat{s}(f) = \int_{-\infty}^{\infty} s(t) \cdot e^{-j2\pi t \cdot f} dt$$

$\hat{s}(f)$ es una nueva representación de $s(t)$ (o $s(n)$) que facilita la visualización de la distribución en frecuencia de la energía. Sin embargo, mediante la TF no se obtiene información acerca del instante en que aparecen/desaparecen las diferentes componentes de frecuencia, es decir que la TF no brinda información de eventos transitorios.

2.3.1. Transformada de Fourier Discreta

Toda señal periódica puede ser representada por la suma de series de Fourier y con un análisis adecuado es posible obtener una representación para señales discretas. Esta representación es la que se conoce como la Transformada de Fourier Discreta (TFD).

Para valores pequeños de muestras la resolución en sí no consume mucho tiempo ni recursos. Sin embargo, para valores de muestra lo suficientemente grandes, el cálculo directo se torna poco eficiente no sólo por el gran tiempo que consume sino también por la búsqueda de los recursos necesarios. Por ejemplo, para $N = 2^{30}$ muestras las operaciones a realizar serían 2^{60} y asumiendo que cada operación toma aproximadamente 1ns el cálculo directo de la TFD tardaría unos 13343 días.

Es así que aparece la Transformada Rápida de Fourier (en inglés Fast Fourier Transform o FFT), un algoritmo para el cálculo eficiente de la TFD. Su importancia radica en eliminar una gran parte de los cálculos repetitivos a los que se ve sometida la TFD,

logrando un cálculo más rápido a menor costo.

2.3.2. Algoritmo FFT - Fast Fourier Transform

La Transformada Rápida de Fourier es un algoritmo para el cálculo de la Transformada Discreta de Fourier basado en la división del tiempo, eliminando así gran parte de los cálculos repetitivos que hay que llevar a cabo si se desea resolver la TFD de forma directa. La FFT presentada asume que la cantidad de muestras es potencia de 2, sin embargo existen métodos para adaptar otros valores de muestras a las condiciones necesarias de este algoritmo.

La FFT divide la Transformada de Fourier Discreta a calcular en dos TFD menores según la paridad de los términos. Dado que se tratan de dos TFD, se puede aplicar el mismo método de división en pares e impares para así obtener dos pares de TFD de $N/4$ muestras. El método es así aplicado hasta que se obtienen TFD de 1 muestra, cuyo cálculo resulta sencillo. Una vez obtenidos los valores de las TFD simples, es cuestión de adicionar los resultados.

Puede observarse que si se tiene inicialmente una TFD de N muestras, se podrán llevar a cabo $p = \log_2 N$ divisiones. Si se calcula el costo de las operaciones que hay que llevar a cabo con este método, se tiene que el algoritmo es de $O(N \cdot \log_2 N)$.

Haciendo una comparación rápida del costo de los dos métodos, el cálculo directo de la TFD y la FFT, se puede observar el factor de mejora que brinda la FFT : Considerando el ejemplo presentado inicialmente para 2^{30} muestras, se tiene que el tiempo de cálculo total era de 13343 días. Ahora, aplicando la FFT resulta que hay que realizar sólo $30 * 2^{30}$ operaciones y asumiendo nuevamente que cada operación tarda 1ns se tiene que el tiempo de cálculo total es de aproximadamente 32 segundos.[16]

2.3.3. Transformada Welch

La TFD de una función muestreada $x(n)$ viene dada por:

$$X_k = T \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad \text{donde } T \text{ es el periodo de muestreo.}$$

Es posible calcular la energía de la señal en función de la frecuencia, la cual se denomina densidad espectral de energía S_k , como:

$$S_k = |X_k|^2$$

Si la cantidad de muestras de la señal es N , la potencia de la señal en función de la frecuencia se obtiene dividiendo la energía por el factor $N \cdot T$:

$$P_k = \frac{|X_k|^2}{N \cdot T}$$

P_k se denomina densidad espectral de potencia, y se la designa como Periodograma de la señal. El periodograma no es un estimador consistente, ya que su varianza no tiende a cero cuando la longitud del registro tiende a infinito.

Para mejorar esta característica, Welch propuso un método según el cual se dividía el registro de N puntos original en segmentos de M puntos solapados entre sí L muestras. Para cada uno de estos segmentos, también denominados *ventanas*, se calcula el periodograma, y luego el periodograma final se obtiene promediando los periodogramas parciales.[1]

La principal ventaja que aporta el método Welch reside en que al promediar los periodogramas de las ventanas, la energía del ruido externo, que es aleatoria e inconstante, tiende a anularse mientras que la energía de la señal de interés durante toda la señal se conserva. Sin embargo, debido a que la transformación o periodograma se calcula sobre segmentos de menor longitud que la señal completa, la transformada tiene menor resolución en frecuencia que la FFT.

Esto no presenta inconvenientes a la hora de implementar la transformada Welch debido a que la información de interés no se encuentra en frecuencias específicas y además permite aprovechar la reducción del ruido. Los parámetros óptimos a seleccionarse dependen del tipo de análisis, aunque en nuestra aplicación se implementa una ventana de 2 segundos con 1 segundo de solapamiento.

2.4. Análisis de Componentes Independientes (ICA)

Los artefactos como los movimientos oculares, los parpadeos, las señales cardíacas, la actividad muscular y el ruido de línea dificultan la interpretación y el análisis electroencefalográfico debido al rechazo de segmentos contaminados que resultan en una pérdida de datos poco eficiente.

Existen muchos métodos para eliminar los artefactos de las señales electroencefalográficas, especialmente los que surgen de los movimientos oculares y los parpadeos. La regresión es aplicada frecuentemente en el dominio de tiempo o frecuencia: para ello es necesario realizar registros electrooculográficos (EOG) para derivar los parámetros que caracterizan la aparición y la propagación de los artefactos oculares en los canales EEG. Debido a que la actividad electroencefalográfica y ocular se mezclan bidireccionalmente, la regresión de los artefactos inevitablemente implica restar también las señales de EEG relevantes de cada registro. Los métodos de regresión se vuelven aún más problemáticos cuando no existe un buen canal de regresión disponible para cada fuente de artefactos, como en el caso de los artefactos musculares.

Se propone luego el uso del análisis de componentes principales (PCA) para eliminar los artefactos oculares del EEG multicanal. Sin embargo, PCA no puede separar completamente los artefactos oculares de las señales cerebrales, especialmente cuando tienen amplitudes comparables.

Surge entonces un nuevo método para eliminar una amplia variedad de artefactos de los registros de EEG se basados en la separación de fuentes por análisis de componentes independientes (ICA).

Resultados de diversos estudios muestran que ICA puede detectar, separar y eliminar una amplia variedad de artefactos de los datos del EEG mediante descomposición lineal, con resultados que se comparan favorablemente con los obtenidos mediante métodos de regresión y PCA.[5]

El método ICA se basa en los supuestos de que las señales registradas en el cuero cabelludo:

1. Son mezclas espacialmente estables de las actividades provenientes de fuentes cerebrales y artificiales temporalmente independientes.
2. La suma de potenciales que surgen de diferentes partes del cerebro, cuero cabelludo y cuerpo es lineal en los electrodos.
3. Los retrasos de propagación desde las fuentes a los electrodos son insignificantes.

El método utiliza filtros espaciales derivados del algoritmo PCA y no requiere un canal de referencia para cada fuente de artefactos. Una vez que las diferentes fuentes cerebrales y de artefactos se extraen de los datos, las señales de EEG con corrección de artefactos pueden obtenerse eliminando las contribuciones de las fuentes de artefactos de las mismas[5].

La Figura 2.10 presenta una ilustración esquemática del procedimiento. En el análisis de EEG, las filas de la matriz de entrada "X" son señales de EEG registradas en diferentes electrodos y las columnas son medidas registradas en diferentes puntos de tiempo (izquierda). ICA encuentra una matriz "W" que descompone linealmente los datos multicanales en una suma de componentes temporalmente independientes y fijos espacialmente.

Las filas de la matriz de datos de salida, $U = WX$, representan activaciones de los componentes ICA en el curso del tiempo. Las columnas de la matriz inversa, $(W)^{inv}$, dan las fuerzas de proyección relativas de los componentes respectivos en cada uno de los sensores del cuero cabelludo (Figura 2.10 derecha). Estas fuerzas dan la topografía del cuero cabelludo de cada componente y proporcionan evidencia de los orígenes fisiológicos de los componentes.

En la práctica, es importante reconocer cuáles son las fuentes de artefactos que contaminan la señal de EEG (Sección 1.4), para poder eliminarlas y en un paso siguiente volver a reconstruir la misma.

Siguiendo con el ejemplo en la Figura 2.11, se obtienen nuevas señales cerebrales relacionadas a eventos y libres de artefactos a partir de proyectar los componentes ICA seleccionados, resultando:

$$\text{Clean Data} = W_{inv}(:, a) * \text{activaciones}(a, :);$$

donde [a] es un vector que contiene los componentes seleccionados que no corresponden a fuentes de artefactos. [5]

ICA decomposition

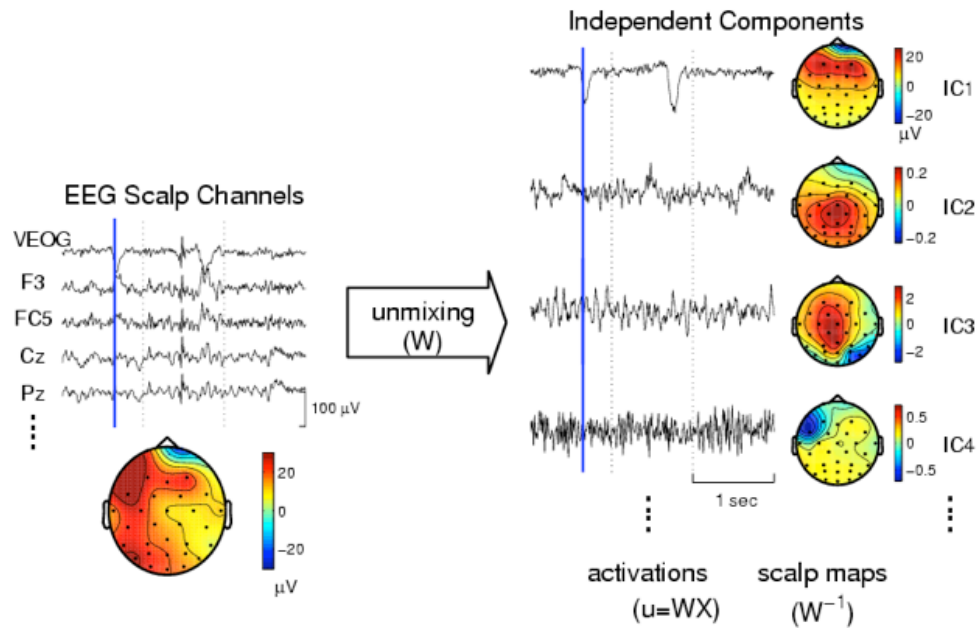


Figura 2.10: Ejemplo de una señal de EEG y la obtención de una matriz W que descomponen linealmente los datos del cuero cabelludo en una suma de componentes temporalmente independientes y fijos espacialmente.[5]

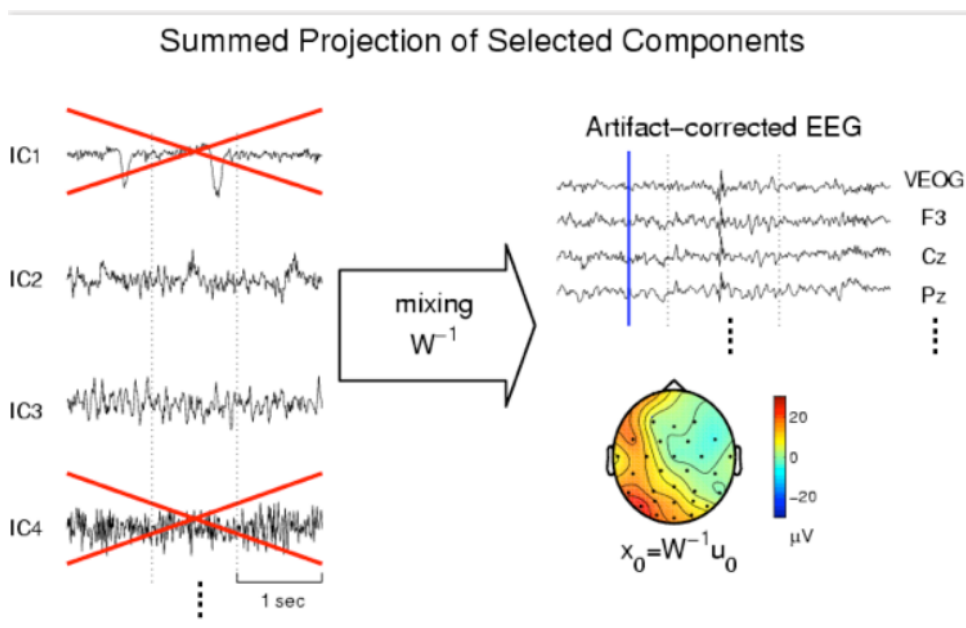


Figura 2.11: Reconstrucción de la señal de EEG libre de artefactos como resultado de la suma de componentes ICA seleccionados sobre el cuero cabelludo.[5]

2.5. Espectrograma

El espectrograma consiste en tomar un determinado número de muestras por medio de una ventana temporal con un tamaño concreto, a continuación hacer el cálculo del contenido frecuencial de las muestras de dicha ventana, y representarlo en una gráfica en tres dimensiones.

Seguidamente se desplaza la ventana a lo largo del tiempo de la señal para seleccionar otro número de muestras diferentes, repetir el proceso y representar los nuevos datos en la misma gráfica anterior. Esta operación se repite sucesivamente a lo largo de toda la señal.

La concatenación de los resultados de las transformadas de Fourier en las ventanas consecutivas permite analizar la energía espectral de la señal en cada instante, aportando información simultánea de los espectros de tiempo y de frecuencia. Esto permite por ejemplo, conocer en qué instante se inició una respuesta fisiológica, estudiando en qué momento la densidad espectral de mayor energía se situó en la frecuencia que corresponde a dicha respuesta.

El espectrograma requiere entonces de tres dimensiones para ser graficado y el mismo puede ser representado de formas diferentes. Generalmente se representa el tiempo en el eje de abscisas, las frecuencias en el eje de ordenadas y una representación de la energía en dB en el plano tridimensional acompañada con una gama de colores que indican la variación en la energía (Figura 2.12).

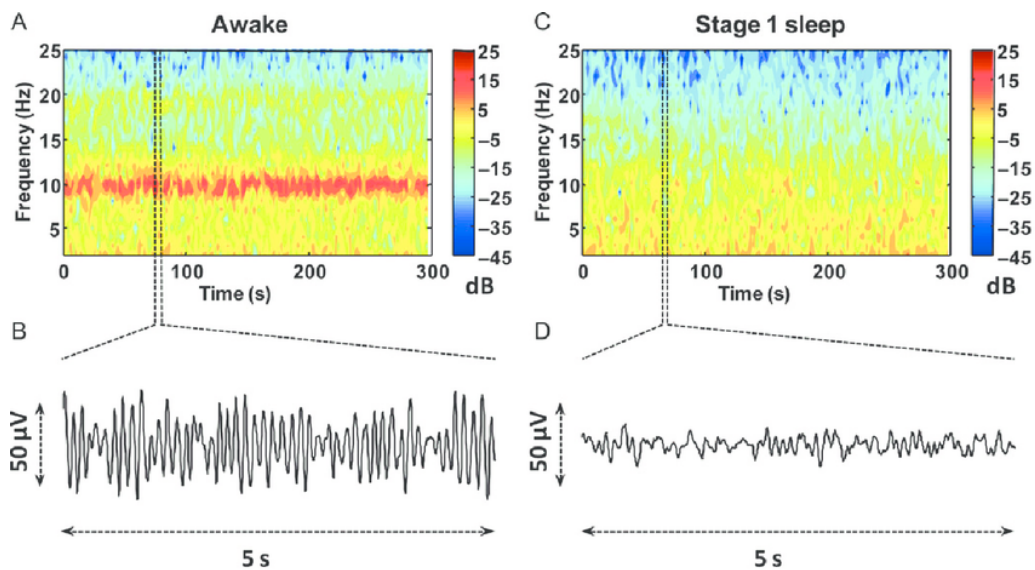


Figura 2.12: Espectros de potencia y formas de onda típicos de EEG después de eliminar artefactos. Tanto (A) como (C) muestran el espectrograma de potencia de toda la época (300 s) del electrodo O1 del mismo sujeto, tanto (B) como (D) muestran segmentos de 5-s de formas de onda EEG entre las 2 líneas de puntos en los espectrogramas. ⁶

⁶Fuente: Efficiency of a "Small-World" Brain Network Depends on Consciousness Level: A Resting-State fMRI Study - Taira Uehara

2.6. Correlación y Coherencia

El estudio de las relaciones funcionales entre dos regiones cerebrales ha sido uno de los temas más importantes desde el desarrollo del Electroencefalograma.

La Correlación y la Coherencia son métodos utilizados para evaluar el grado de similitud que existe entre dos señales y considerando que dos áreas relacionadas funcionalmente presentan respuestas similares ante el mismo estímulo, los mismos pueden constituir un buen indicador del grado de relación entre las áreas circundantes a electrodos ubicados en posiciones distintas.

En un principio la función utilizada para el estudio de la similitud entre dos señales electroencefalográficas fue la Correlación Cruzada (o Correlación). Asumiendo que a una elevada Correlación le corresponde una elevada relación funcional entre ambas regiones cerebrales, es implementada en el estudio de relaciones entre diferentes regiones corticales ante estimulación sensorial, movimientos voluntarios, efecto de drogas, entre un amplio rango de problemas clínicos.

Con el desarrollo digital y de nuevos algoritmos computacionales como la Transformada Rápida de Fourier, el uso de la Correlación fue decreciendo gradualmente y reemplazado por un método matemático alternativo denominado Coherencia de Espectro (o Coherencia). Este método provee información similar a la Correlación pero con la ventaja de indicar la similitud entre dos señales en función de la frecuencia y en un lapso de tiempo relativamente corto.

Ambos métodos, correlación (r) y coherencia (Coh) son considerados al momento de evaluar el grado de similitud entre dos señales aunque lo hacen a través de diferentes cálculos matemáticos.

2.6.1. Correlación

El cálculo de la Correlación es muy similar al de la operación de convolución de señales, con la diferencia de que la señal trasladada no es invertida:

$$\text{Convolución: } y[n] = \sum_{k=0}^N x[k] * h[n - k]$$

$$\text{Correlación: } y[n] = \sum_{k=0}^N x[k] * h[n + k]$$

Esto quiere decir que la Correlación se obtendrá a partir de trasladar una de las señales sobre la otra, realizando en cada instancia de traslado el producto de las señales punto a punto, como puede verse en la Figura 2.13. Por lo tanto, si una de las señales está contenida en la otra, o bien si al menos ambas señales son similares, la Correlación se maximizará en el punto donde la traslación alinee ambas señales. [4]

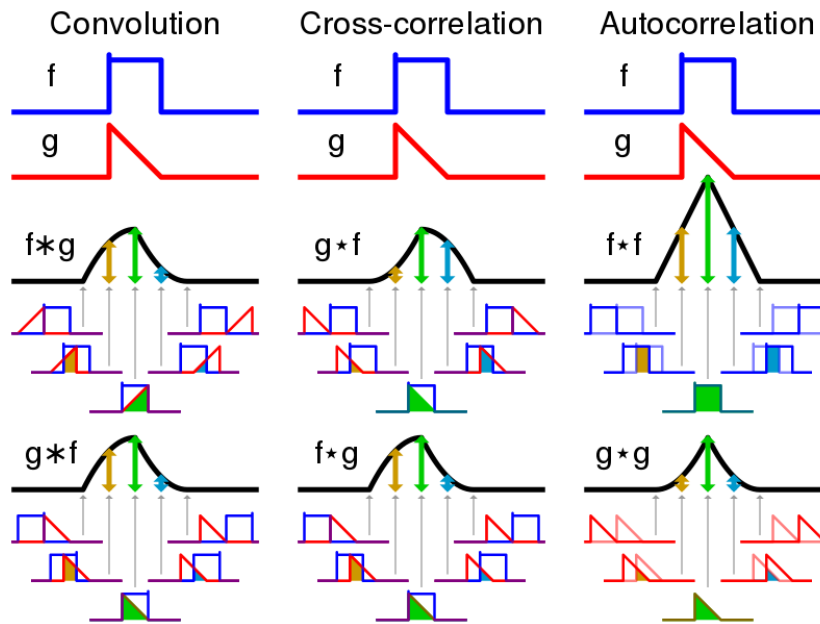


Figura 2.13: Comparación de distintas operaciones entre dos señales, una cuadrada y otra triangular. A la izquierda la operación de convolución (que involucra traslación y rotación de la señal triangular); mientras que en el centro y a la derecha las operaciones de correlación (que involucran traslación pero no rotación) entre ambas señales, denominada Correlación Cruzada y entre la señal cuadrada y sí misma, denominada Autocorrelación.

7

2.6.2. Coherencia

La Coherencia es una medida estadística que se utiliza para encontrar frecuencias comunes entre dos señales y evaluar de esta manera la similitud entre ambas. Sin embargo, la Coherencia no aporta información sobre la similitud en el espectro de tiempo.

La función de Coherencia, respecto a la frecuencia angular (ω), se define como:

$$\text{Coherencia: } C_{xy}[\omega] = \frac{P_{xy}[\omega]}{\sqrt{P_{xx}[\omega]P_{yy}[\omega]}}$$

donde P_{xx} y P_{yy} son las energías espectrales de las señales x e y respectivamente, y P_{xy} es la energía espectral cruzada. Los primeros se calculan como el producto punto entre la transformada de Fourier de cada señal y su complejo conjugado, mientras que la energía espectral cruzada se calcula como el producto punto entre la transformada de Fourier de x y el complejo conjugado de la transformada de y [9]:

$$P_{xx}[\omega] = \hat{x}[\omega]\overline{\hat{x}[\omega]}$$

$$P_{xy}[\omega] = \hat{x}[\omega]\overline{\hat{y}[\omega]}$$

Si bien ambos métodos evalúan el grado de similitud entre dos señales, poseen importantes diferencias.

⁷Fuente: Cross-Correlation. Wikipedia.

La Coherencia es sensible a cambios en la relación de energía y de fase, es decir, que su valor se modifica si ocurre algún cambio en alguna de éstas propiedades de las señales que están siendo comparadas. Para una única época o intervalo, el valor de la Coherencia siempre será 1 (el máximo) sin importar cuán desfasadas o diferenciadas en amplitud estén las señales. En cambio, la Coherencia evaluará cuánto se modifica la fase y la amplitud a lo largo de las distintas épocas o intervalos. Esto significa que la Coherencia no da información directa de la relación de ambas señales sino que informa sobre la estabilidad respecto a las diferencias de energía y la relación de fase. El valor de Coherencia obtenido refleja entonces cambios en la energía o cambios en la morfología de las señales comparadas.

En cambio, la Correlación puede ser calculada sobre una única época o sobre muchas, siendo sensible a la fase y a la polaridad. La Correlación proporciona información sobre el acoplamiento en tiempo y la similitud de forma de onda entre las dos señales independientemente de las amplitudes.

Entonces, cuando el interés es conocer la estabilidad de relación entre dos sitios de registro, que incluye tanto asimetrías de energía como morfología, el mejor método es la Coherencia. Cuando el interés principal se centra en la forma de onda y el acoplamiento de tiempo entre dos sitios, la Correlación es una mejor opción.

Sin embargo, cabe destacar que, en condiciones fisiológicas normales, no es común encontrar grandes asimetrías de energía en el registro electroencefalográfico. Por lo tanto, la influencia de la energía en la coherencia puede considerarse despreciable.[2]

Capítulo 3

Materiales y Métodos

3.1. Materiales y Herramientas

Como se mencionó anteriormente, la motivación de este trabajo es el creciente interés en el desarrollo de herramientas que implementen técnicas digitales de procesamiento para interpretar señales electroencefalográficas. Éstas consisten en transformar datos de la señal cruda (sin procesar) en datos numéricos y/o gráficos que faciliten su análisis y comprensión.

Una vez realizado el estudio de las técnicas y métodos utilizados frecuentemente en el procesamiento de señales de EEG, se decide orientar el diseño de la herramienta a señales con formatos exportados por la placa de adquisición Ganglion de OpenBCI. Dicha placa constituye un requisito del proyecto debido a que es utilizada por la empresa OTTAA Project para la obtención de señales.

Si bien fue necesario el uso de hardware para la adquisición de las señales utilizadas durante la etapas de Verificación y Casos de Aplicación (Secciones 4 y 5), el presente proyecto se desarrolló enteramente con herramientas de software que abarcan: Entornos de programación, Lenguaje y Librerías.

3.1.1. OpenBCI

OpenBCI es una empresa dedicada a la producción de placas para adquisición de bioseñales, especialmente de ondas cerebrales, a un costo accesible junto con un software gratuito para la obtención y el procesamiento de las señales eléctricas del cerebro. El objetivo del proyecto OpenBCI es permitir a las personas adquirir señales cerebrales con el simple uso de una placa de adquisición y una computadora de uso doméstico.

El núcleo de la plataforma es una placa MCP3912 Analog Front End (ADC) de Microchip, alimentado con batería de 3.3V a 6V, posee además un módulo de radio Simblee BLE (compatible con Arduino) y acelerómetro de 3 ejes LIS2DH. Hay que tener en cuenta que el Ganglion posee un filtro de paso de banda que corta cualquier frecuencia por debajo de 0.3 Hz (Figura 3.1).

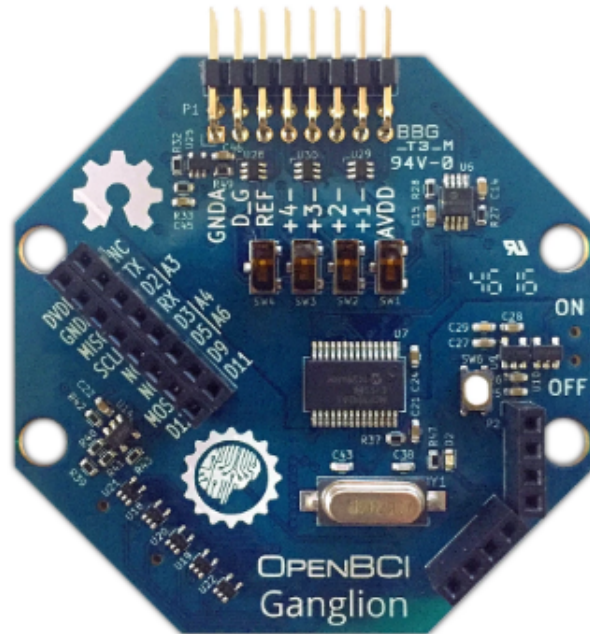


Figura 3.1: Estructura de una placa Ganglion OpcenBCI con sus componentes. ¹

Actualmente el trabajo de OpenBCI se enfoca en lograr que los datos generados por el hardware, puedan ser accedidos por otras aplicaciones de procesamiento de señales de código, como BrainBay y OpenVibe, entre otros. Debido a que se tiene acceso directo a los datos, hacer que sea transferible a cualquier software de EEG existente solo requiere estructurar la forma en que los datos están formateados y relacionados.

3.1.2. Entornos de Programación: Anaconda y Jupyter

3.1.2.1. Anaconda

Para la implementación de la herramienta se utilizó Anaconda, que es una plataforma gratuita y de código abierto que funciona como un administrador de paquetes, administrador de ambientes, y distribuidor de Ciencia de Datos para **Python** y **R**, con una colección de más de 1500 paquetes de acceso libre. Además, es uno de los proveedores de **Jupyter**, plataforma de la que se hablará a continuación.[17]

Sus principales utilidades radican en la creación de distintos ambientes de trabajo sin que se interfieran entre ellos, además de poder trabajar con distintos lenguajes o distintas versiones del mismo lenguaje en cada ambiente.

Los beneficios de utilizar una plataforma como Anaconda surgen al momento de trabajar en equipo o cuando es necesario compartir resultados con miembros de la comunidad, debido a que el uso de una plataforma estandarizada permite replicar rápidamente un entorno como así también una configuración. Esto significa menos tiempo empleado en configurar y modificar bibliotecas.

¹Fuente: <https://openbci.com/> - Ultimo ingreso : 22/07/19

3.1.2.2. Jupyter

Para desarrollar y ejecutar el código, se utilizó la plataforma Jupyter, que es un entorno de trabajo muy utilizado en el ámbito de la investigación y desarrollo, ya que combina en un mismo documento tanto bloques de código como texto, gráficos o imágenes.[18]

Actúa como un *host local* utilizando la memoria propia del sistema para ejecutar código. Jupyter trabaja a través de bloques, incluyendo:

- **Bloques de programación:** Estos son dinámicos y sus salidas se encuentran vinculadas, con la posibilidad de correr de manera independientemente uno del otro. Por defecto trabaja en lenguaje *Python*, pero acepta otros lenguajes como *R* o *Java*.
- **Bloques de texto:** En formato *Markdown* de \LaTeX , incluyendo la inserción de gráficos e imágenes.
- **Herramientas de interactividad:** Como botones o deslizadores, que permiten interactuar con las funcionalidades de programación más intuitivamente.

La elección de Jupyter como entorno de trabajo del proyecto se basó en tres ejes principales:

- La simplicidad de uso, que permite prestar mayor enfoque al desarrollo de código en sí, en vez de en cómo ejecutar dicho código.
- La dinámica de bloques, que posibilita la ejecución de partes del algoritmo sin necesidad de ejecutar el código completo. Esta fue especialmente útil en las primeras etapas de desarrollo, donde cada uno de los autores trabajaba en un bloque independiente para luego unirlos en un único cuaderno Jupyter.
- La universalidad de uso de esta plataforma en el ámbito del desarrollo científico, para poder compartir la herramienta a otros usuarios.

3.1.3. Lenguaje de Programación: Python

Python es un lenguaje de programación interpretado y su importancia aumentó significativamente en las últimas décadas, siendo utilizado en programación web de páginas de alto tráfico como Youtube.

A su vez, se siguen desarrollando aún más sus utilidades en el ámbito científico (lo que es llamado *Scientific Python*) y esto se debe principalmente a su cualidad *open-source* que permite que cualquier persona contribuya con librerías de desarrollo propio, así también como su sintaxis libre de llaves basado en el uso de indentación, lo cual produce un código más limpio, fácil de entender y mantener incluso para aquellas personas sin mucha experiencia.

Su crecimiento en los últimos años puede observarse en la siguiente Figura 3.2 :

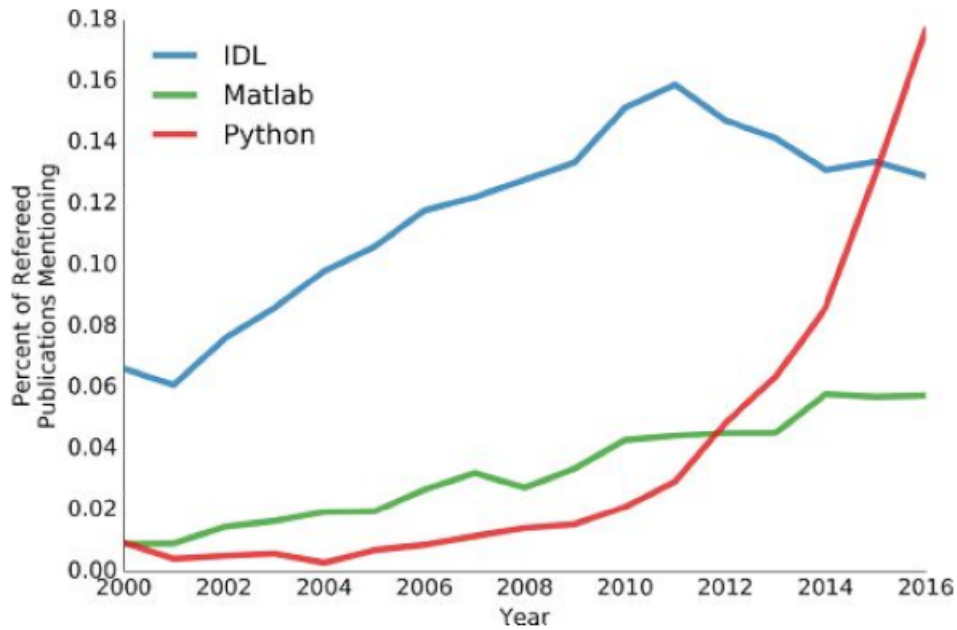


Figura 3.2: Comparativa entre porcentajes en cantidad de menciones de Python, MATLAB e IDL en publicaciones científicas en los últimos años donde puede observarse el evidente crecimiento en el uso de Python. [19]

Entre los lenguajes considerados al momento de diseñar la herramienta (C++, MATLAB y Python), se concluye que el más adecuado para el desarrollo del proyecto es Python debido a las características mencionadas anteriormente y a la gran diversidad de librerías desarrolladas que son gratuitas y de fácil acceso, pero además se encuentran en constante revisión por parte de sus autores. Al mismo tiempo, existen librerías con funciones dedicadas al procesamiento de señales de Electroencefalografía, tanto generales como específicas, que aceleran el avance del proyecto y posibilitan un alcance mucho más amplio.

En contraste, MATLAB al no ser un lenguaje de programación exclusivo sino un entorno de desarrollo integrado con un lenguaje de programación propio, no posee una gran parte de las funcionalidades genéricas de programación. C++ por su parte, posee una sintaxis mas compleja y es un lenguaje compilado (a diferencia de Python que es un lenguaje traducido), generando mayor dificultad al momento de ser implementado por otros usuarios y oponiéndose al objetivo del proyecto de ser open-source y de fácil acceso por los mismos.

3.1.4. Librerías de Python

Siguiendo con lo mencionado en la sección anterior y considerando que Python posee una variada cantidad de librerías, se enumeran todas aquellas que fueron utilizadas en el desarrollo de este proyecto:

3.1.4.1. Numpy: Librería Numérica de Python

Numpy (abreviación de *Numerical Python*), es la librería utilizada con mayor frecuencia para el manejo de arreglos numéricos en Python. Desarrollada por Travis Oliphant y su equipo, busca unificar en una única librería todas las maneras de definir arreglos numéricos a través de Python.

Numpy permite crear arreglos numéricos tipo vectores (columnas o filas) o matrices de tantas dimensiones como se desee, cuyos elementos pueden ser cualquier valor numérico: *integer*, *float*, *double*, *uint8*, etc. Estos pueden ser creados manualmente a partir de listas de Python, o siguiendo parámetros (valores aleatorios; *linspace* o *arange*, como se puede observar en la Figura 3.3).

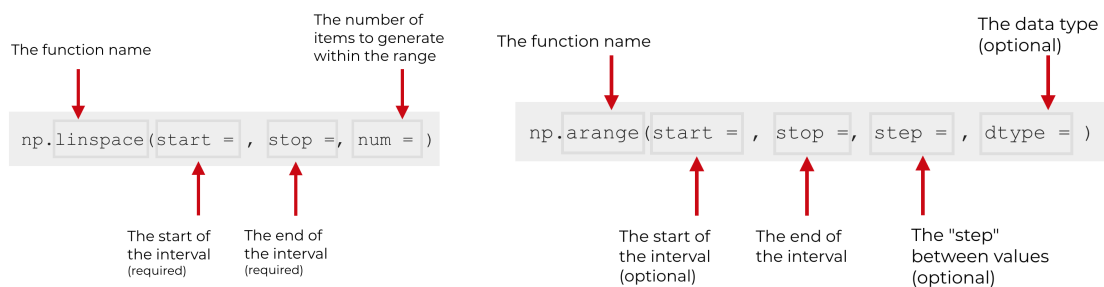


Figura 3.3: Dos métodos implementados para la creación de arreglos numéricos en Numpy. El método *arange*, toma un valor inicial y un valor final, y dado un cierto salto entre valor y valor crea un arreglo de tantos elementos como sea necesario, mientras que el segundo denominado *linspace*, toma un valor inicial y un valor final, y una cantidad de elementos dentro de estos valores, para crear un arreglo con la cantidad de elementos seleccionados.

Además, Numpy incluye funciones para realizar operaciones algebraicas simples y complejas, entre ellas:

- Suma, Resta, Multiplicación, Producto Punto y otros productos entre arreglos.
- Reordenamiento de elementos a partir de alguna condición establecida.
- Transformaciones a espacios algebraicos alternativos (transformada de Fourier, cambio de Base) .

3.1.4.2. Scipy: Librería Científica de Python

Scipy (abreviación de Scientific Python) puede hacer referencia a un Ecosistema de Librerías, Marcos de Referencia y Herramientas destinadas al uso científico, matemático e ingenieril de Python o a la Comunidad que da este uso a Python. En esta sección se describe a Scipy como una Librería utilizada en Python.

La librería Scipy implementa un gran conjunto de funciones para el procesamiento de datos científicos, aplicaciones de estadística, procesamiento de señales e imágenes, optimización de funciones, entre otras, mediante el uso de interfaces amigables con el usuario.

Está desarrollada sobre la librería Numpy, lo que significa que utiliza los métodos y funciones de esta última; pero a su vez fueron sobre ambas, Numpy y Scipy, que se desarrollan la gran mayoría de las librerías utilizadas por la comunidad científica de Python, siendo responsables en gran medida del crecimiento de Python para usos científicos de los últimos años. [12]

3.1.4.3. Pandas: Librería de Manejo de Datasets

Un dataset es un bloque de información formado por un conjunto de valores que pueden adquirir cada una de las variables implicadas, como por ejemplo la altura y el peso de un objeto. Cada uno de estos valores se conoce con el nombre de dato y el conjunto de datos puede incluir valores para uno o más miembros en función de su número de filas.

Pandas provee estructuras de tablas construidas sobre datasets, principalmente orientadas a organizar los datos en columnas para facilitar su visualización (Figura 3.4). Pandas ofrece también:

- Funciones necesarias para generación de tablas, carga manual de datasets a partir de un arreglo de Python o Numpy, o incluso a partir de un archivo.
- Funciones básicas para el reordenamiento, filtrado y análisis de los datasets, para adjuntar o eliminar columnas a una determinada tabla, e incluso una visualización básica de la información en gráficos. [12]

	Sample	Ch1	Ch2	Ch3	Ch4	x	y	z	Time	
0	0	0.00	0.00	0.00	-125.15	0.000	0.000	0.000	12:15:20.870	1547824520870
1	1	0.00	0.00	0.00	-137.52	0.000	0.000	0.000	12:15:20.885	1547824520885
2	2	0.00	0.00	0.00	-138.15	0.000	0.000	0.000	12:15:20.885	1547824520885
3	3	0.00	0.00	0.00	-130.99	0.000	0.000	0.000	12:15:20.899	1547824520899
4	4	0.00	0.00	0.00	-130.16	0.000	0.000	0.000	12:15:20.899	1547824520899
5	5	0.00	0.00	0.00	-136.97	0.000	0.000	0.000	12:15:20.901	1547824520901
6	6	0.00	0.00	0.00	-143.00	0.000	0.000	0.000	12:15:20.901	1547824520901
7	7	0.00	0.00	0.00	-130.37	0.000	0.000	0.000	12:15:20.915	1547824520915
8	8	0.00	0.00	0.00	-125.37	0.000	0.000	0.000	12:15:20.915	1547824520915

Figura 3.4: Tabla provista por Pandas para la visualización de datos.

3.1.4.4. Matplotlib: Librería de Gráficos Matemáticos

Matplotlib es la librería de gráficos y ploteos utilizada por defecto en las aplicaciones científicas de Python. Fue desarrollada individualmente por John Hunter, inspirado en algoritmos de gráficos de MATLAB como una necesidad para aplicaciones similares en Python. Esta similitud con funciones de MATLAB es la que llevó a su amplia utilización por parte de la comunidad científica.[19]

A partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy, puede generar gráficos, histogramas, espectros de

potencia, gráficos de barras, gráficos de error, diagramas de dispersión, etc., con solo unas pocas líneas de código (Figura 3.5).

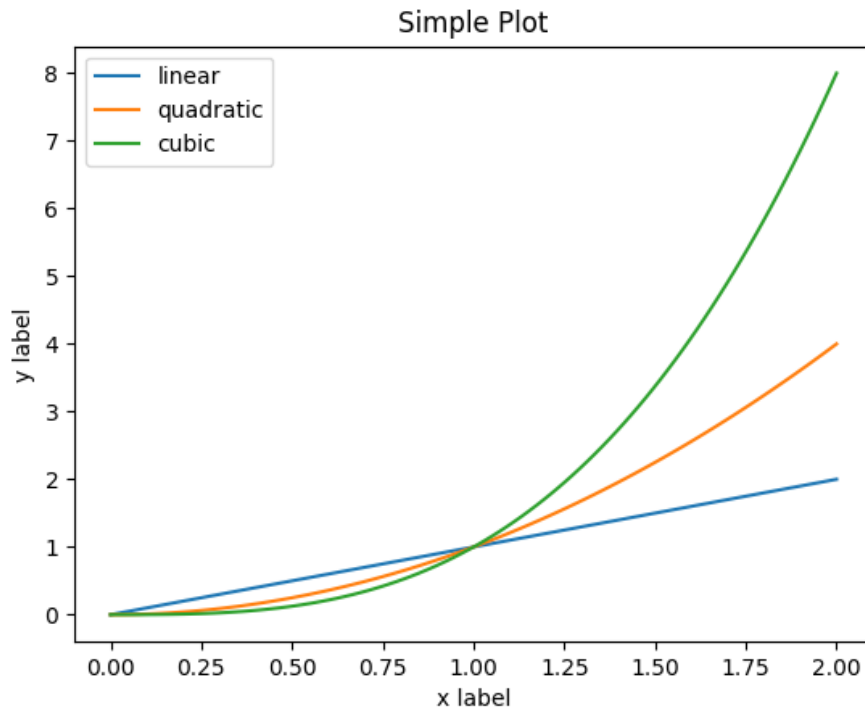


Figura 3.5: Gráfico de funciones generado por Matplotlib²

3.1.4.5. Plotly

Plotly o también denominada Plot.ly (por su URL), es una empresa dedicada al diseño herramientas para aplicaciones de Ciencia de Datos en lenguajes de programación como Java, Python o R. Su principal librería denominada también Plotly, ofrece tanto en R como en Python un recurso para generar gráficos y ploteos.

Se distingue de otras librerías en que permite al usuario crear de manera simple gráficos interactivos a un nivel de calidad de artículo científico. Esta interactividad incluye entre sus funciones:

- Funcionalidad de **Zoom**: Permite acercar o alejar la imagen para resaltar detalles.
- Funcionalidad de **Escala**: Permite ajustar la imagen según nuevas escalas en ambos ejes coordenados.
- Funcionalidad de **Hover**: Se detallan sobre el gráfico los valores de cada coordenada al deslizar el puntero del mouse.
- Funcionalidad de **Captura**: Permite guardar imágenes tipo PNG del gráfico.

²Fuente: <https://matplotlib.org> Ultimo Ingreso : 25/07/19

Si bien la librería utilizada para visualización de datos en el desarrollo de la herramienta es Plotly debido a la posibilidad de interacción dinámica con los gráficos, Matplotlib fue utilizada también durante el desarrollo para la verificación de etapas mediante la generación de gráficos rápidos.

3.1.4.6. MNE: Librería para el manejo de Señales de Neuroimagenología

MNE es una librería desarrollada por el centro Athinoula A. Martinos Center for Biomedical Imaging y posee funciones desarrollada principalmente para Python pero también sobre otros lenguajes como MATLAB y C. Sus funciones están destinadas al procesamiento, análisis y visualización de señales de neuroimagenología (EEG, MEG, sEEG, ECoG, and fNIRS)[14].

Sus aplicaciones incluyen procesamiento de señales, estimación de fuentes, análisis en espectros de tiempo y frecuencia, análisis estadísticos, y estimaciones de interconectividad de áreas cerebrales. [10]

3.1.4.7. Wyrms

Wyrms es una librería desarrollada por Bastian Venthur, usando como base fundamental librerías como Numpy, Scipy y Matplotlib en un intento por proveer herramientas gratuitas y de libre acceso que sirvieran como complemento para desarrollar tecnologías de procesamiento de señales de EEG.[11]

Esta librería implementa decenas de funciones, cubriendo una amplia gama de aspectos para análisis offline y aplicaciones online. La lista de algoritmos incluye: Selección de canales, Filtros IIR, Submuestreo, Espectrogramas, Espectros, Eliminación de la línea de base para procesamiento de señales, Patrones espaciales comunes, Promedio por clase, entre varias funciones más.

3.1.4.8. Tkinter: Librería de Interfaz Gráfica

Tkinter ("Tk Inter face") es un paquete multiplataforma estándar de Python para crear interfaces gráficas de usuario (GUI). Proporciona acceso a un intérprete de Tcl subyacente con el kit de herramientas Tk, que en sí mismo es una biblioteca de interfaz de usuario gráfica multiplataforma.

Tkinter no es la única biblioteca GUI para Python, pero es la que viene instalada de manera predeterminada. Las bibliotecas adicionales que se pueden usar con Python incluyen wxPython, PyQt y kivy .

La mayor fortaleza de Tkinter es su simplicidad. Funciona de forma inmediata en la mayoría de las plataformas (Linux, OSX, Windows) y se completa con una amplia gama de widgets necesarios para las tareas más comunes (botones, etiquetas, lienzos de dibujo, texto de varias líneas, etc.)(Figura 3.6).

Como herramienta de aprendizaje, Tkinter tiene algunas características que son únicas entre los kits de herramientas GUI, como las fuentes con nombre, las etiquetas de enlace y el rastreo de variables.

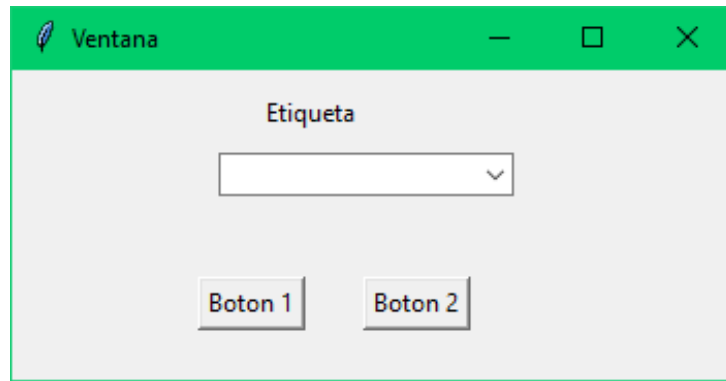


Figura 3.6: Ventana generada por Tkinter con widgets como etiquetas, botones y caja de contenidos. ³

Si bien no es objeto del presente proyecto desarrollar una interfaz gráfica de usuario, Tkinter aporta en muchas etapas del desarrollo una solución para la búsqueda de archivos, ingreso de variables, verificación de resultados obtenidos y errores, simplificando secciones y facilitando de ésta manera el uso a personas con limitados conocimientos en programación.

3.2. Desarrollo del Proyecto

En esta sección se describe el proceso de diseño y desarrollo de la herramienta. Se presentan los requisitos iniciales definidos como entrada de diseño y mediante un diagrama general se detalla de manera breve la solución propuesta a los requisitos planteados.

Ante dichos requisitos, se propone como solución una herramienta desarrollada en bloques y estructurada de manera secuencial, es decir que cada una de los bloques y funciones se ejecutan una a continuación de la otra. Para esto se implementa un diseño en espiral, con evolución parcial, en busca del cumplimiento de los requisitos.

A su vez, cada bloque del diagrama general esta compuesto por funciones con sus respectivas entradas y salidas. La herramienta no permite al usuario controlar qué función se está ejecutando, aunque en algunas instancias otorga la posibilidad de omitir la ejecución de algunas a través de ventanas de advertencia generadas por la librería Tkinter.

A continuación se describe en cada una de las iteraciones: las entradas, el desarrollo del método y sus respectivas salidas, junto con las soluciones pertinentes para cada caso.

Por último, a modo de resumen se presenta una tabla con los requisitos resueltos y los impactos finales como producto del desarrollo de la herramienta.

3.2.1. Requisitos de la Herramienta

En esta sección se presenta la Tabla 3.1 con los requisitos planteados por la empresa OTTAA project para el desarrollo de la herramienta. En la misma se describe el Impacto de Fallo y la Probabilidad de Fallo.

Ambas categorías están clasificadas según las variables cualitativas de valor: Bajo, Medio o Alto. En la categoría **Impacto de Fallo** un valor *Alto* indica que un requisito no resuelto impide la ejecución normal de la herramienta. Un valor *Medio* indica que la

herramienta puede ser ejecutada aunque implica cambios o modificaciones en sus funciones, y un valor *Bajo* indica que en el caso de no resolverse un requisito, esto no afecta la normal ejecución de la herramienta.

En la categoría **Probabilidad de Fallo**, los distintos valores representan grados de ocurrencia de alguna falla en la ejecución. Un valor *Alto* implica un fallo con ocurrencia muy probable, un valor *Medio* indica un fallo que podría ocurrir eventualmente y un nivel *Bajo* representa la ocurrencia de un fallo muy poco probable o imposible.

Una vez asignadas las variables a los distintos requisitos, se genera una escala con índices que representen la prioridad de cada uno de ellos a la hora de resolverlos. A partir de asignar a cada variable Baja un valor de 0, Media 1 y Alta 2, se obtienen los valores del índice de prioridad como la suma del Impacto y Probabilidad de Fallo con un rango de valores resultantes de 0 a 4. Un valor 0 representa requisitos que requieren de menos atención o menos urgencia mientras que un valor 4 representa aquellos requisitos que requieren de mayor dedicación ya que son críticos para el desarrollo efectivo del proyecto.

Cabe aclarar que los índices de prioridad no establecen necesariamente un orden de desarrollo sino una valoración cuantitativa de la importancia al momento de resolver cada uno de los requisitos.

Cuadro 3.1: Requisitos del Proyecto

No.	Requisitos	Impacto de Fallo	Probabilidad de Fallo	Prioridad
Req 1	Lectura de archivos en formato Excel y CSV	Alto	Baja	2
Req 2	Definición de tipo y nombre de canales	Medio	Baja	1
Req 3	Definición de la Tasa de Muestreo del Archivo	Alto	Media	3
Req 4	Definición de los intervalos a analizar	Alto	Media	3
Req 5	Aplicación de filtros tipo Butterworth, Chebichev y Notch con la posibilidad de modificar los parámetros de los mismos(Orden, Frecuencias de Corte, Ripple y Atenuación)	Alto	Media	3
Req 6	Aplicación de Submuestreo	Bajo	Baja	0
Req 7	Modificación digital del punto de referencia de los canales (Rreferenciado Digital)	Medio	Media	2

Continuación de Cuadro 3.1: Requisitos del Proyecto

No.	Requisitos	Impacto de Fallo	Probabilidad de Fallo	Prioridad
Req 8	Eliminación de artefactos por método de Análisis de Componentes Independientes (ICA)	Medio	Alta	3
Req 9	Representación espacial del promedio de los valores de señal durante un Intervalo de tiempo a través Mapas Topográficos	Bajo	Alta	2
Req 10	Transformación al espectro de frecuencias mediante la Transformada FFT y Welch	Alto	Media	3
Req 11	Modificación de los parámetros de ventana de la Transformada Welch	Medio	Baja	1
Req 12	Generación de Espectogramas	Medio	Baja	1
Req 13	Visualización de la señal en los distintos espectros obtenidos, con selección interactiva de datos a mostrar	Alto	Alta	4
Req 14	Escala de ejes modificable de manera interactiva	Medio	Alta	3
Req 15	Estudio Correlación y Coherencia entre canales de la señal	Bajo	Baja	0
Req 16	Exportación de los parámetros de carga e intervalos definidos	Medio	Baja	1
Req 17	Exportación de los Parámetros de Acondicionamiento y Análisis implementados	Bajo	Alta	2
Req 18	Exportación de gráficos generados	Medio	Baja	1

3.2.2. Diagrama General

A lo largo de esta sección, se presentan los métodos y algoritmos implementados en el desarrollo de la herramienta y se detalla mediante un diagrama de flujo la secuencia de

etapas que se suceden al ejecutar el programa. Estas etapas serán descritas con mayor detalle en cada iteración, mediante diagramas específicos. (Figura 3.7)

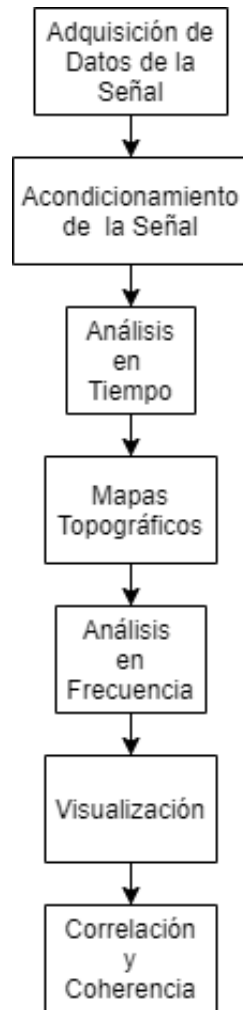


Figura 3.7: Diagrama General con las principales etapas que conforman algoritmo del proyecto.

3.2.3. Iteraciones

Continuando con lo desarrollado en la sección anterior, se describen los métodos y funciones que constituyen cada uno de los bloques del diagrama general, y se analiza cada uno de ellos con sus respectivas entradas, salidas y librerías utilizadas.

3.2.3.1. Codificación de Diagramas de Flujo

En la codificación de los diagramas, se implementan rectángulos para representar a las operaciones, paralelogramos para aquellas suboperaciones que son agregadas a modo de facilitar la comprensión, y rombos para las operaciones de decisión denominadas Si/No.

Los elementos de entrada y salida se representan con rectángulos coloreados de bordes intermitentes. Rectángulos de color amarillo representan aquellos elementos que son considerados únicamente de entradas, en color verde aquellos que son únicamente salidas, y en color azul aquellos elementos que se comportan como entrada y salida de

una misma operación. Esta codificación puede observarse de manera clara en la siguiente figura:



Figura 3.8: Codificación utilizada para los Diagramas Específicos

3.2.3.2. Iteración 1: Adquisición de Datos

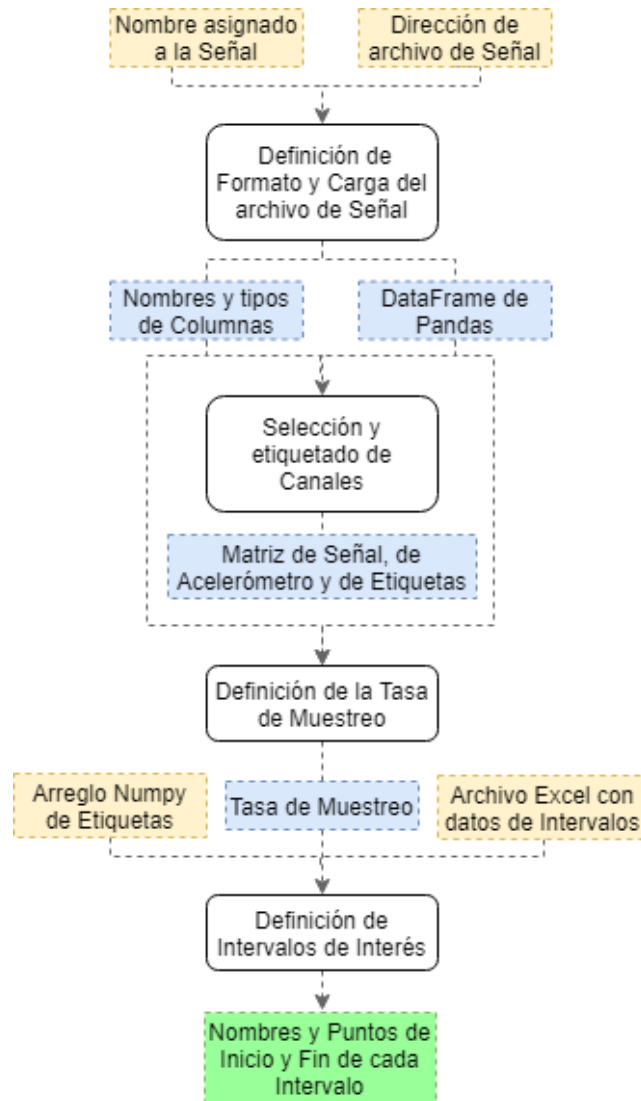


Figura 3.9: Diagrama: Adquisición de Datos

La primera iteración comprende aquellas funciones cuyo objetivo es reunir toda la información de la señal disponible, y utilizarla para crear un formato de señal que pueda ser fácilmente implementado en el resto de las etapas. Además de los datos propios de la señal, se puede mencionar la tasa de muestreo, el montaje y los intervalos de tiempo a estudiar, entre otros.

Definición del Formato y Carga del archivo de Señal

Entradas:

1. Dirección del archivo de Señal.
2. Nombre asignado a la Señal.

Librerías Utilizadas:

1. Tkinter (Sección 3.1.4.8)
2. Pandas (Sección 3.1.4.3)

Desarrollo del Método:

En un primer paso se utiliza una ventana ejecutable mediante la librería Tkinter (Figura 3.10) para ubicar el archivo de señal. Esta ventana permite explorar el ordenador en busca del archivo que contenga la señal (predeterminado en formato .txt o .xlsx).

Si el archivo presenta formato .txt, se solicita al usuario que confirme si el mismo posee la estructura predeterminada de OpenBCI, considerada en el programa como *Estructura Predeterminada* de ahora en adelante . En el caso que posea formato .txt no predeterminado o formato .xlsx, se solicita al usuario que defina la estructura en que se presentan los datos (cantidad de columnas, tipo de datos, separador, etc)(Figura 3.11).

Una vez definido el formato, se utiliza la dirección del archivo junto con la librería Pandas para recopilar los datos de la señal en una variable del tipo DataFrame (Datos de una señal en forma de columnas y propio de la librería antes mencionada).

Ante errores en los parámetros de carga, la herramienta posee mecanismos de verificación que solicitan al usuario el reingreso de los mismos.

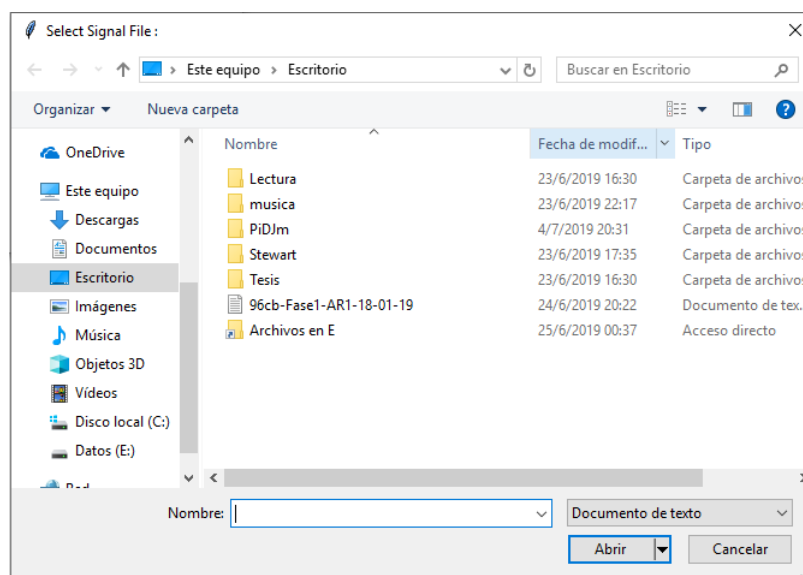
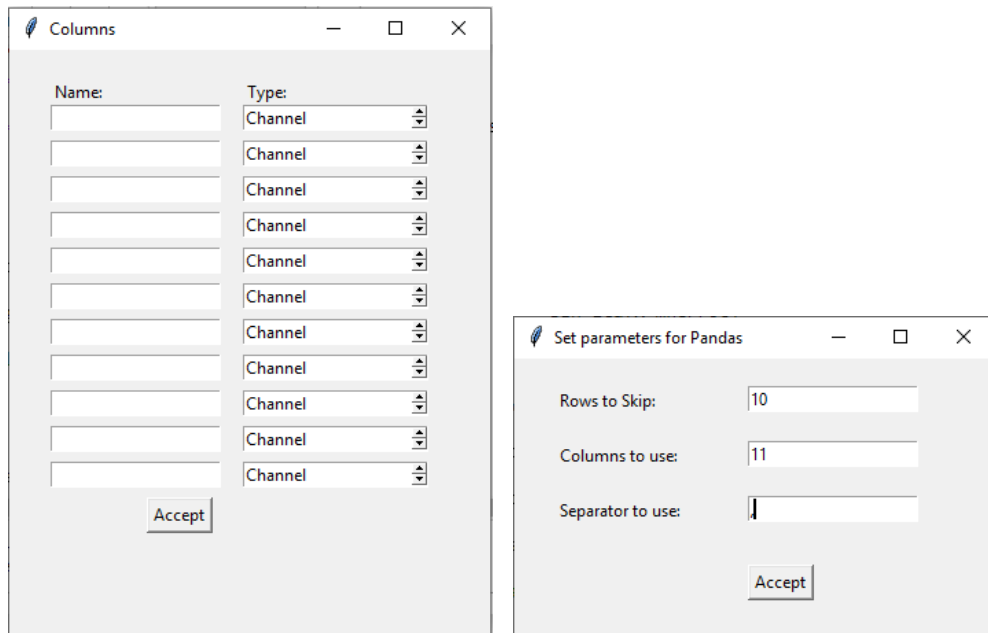


Figura 3.10: Ventana ejecutable en Tkinter para carga de archivos



(a)

(b)

Figura 3.11: Ventanas ejecutables en Tkinter para definición de parámetros de carga

Salidas:

1. DataFrame de Pandas (Sección 3.1.4.3).
2. Nombres y tipos de Columnas del DataFrame.

Selección y Etiquetado de Canales

Entradas:

1. DataFrame de Pandas.
2. Nombres y tipos de Columnas del DataFrame.

Librerías Utilizadas:

1. Tkinter (Sección 3.1.4.8)
2. MNE (Sección 3.1.4.6)
3. Numpy (Sección 3.1.4.1)

Desarrollo del Método:

Una vez definido en el paso anterior el tipo de dato en cada columna, se debe seleccionar cuáles de las columnas asignadas al tipo **Channel** (aquellas que contienen información de los canales o electrodos) van a ser utilizadas durante el procesamiento. Se debe asignar también un nombre y una posición según el lugar de los electrodos en el cuero cabelludo durante el estudio basado en el Sistema 10-20. Estos parámetros son seleccionados manualmente por el usuario como se observa en la siguiente ventana:

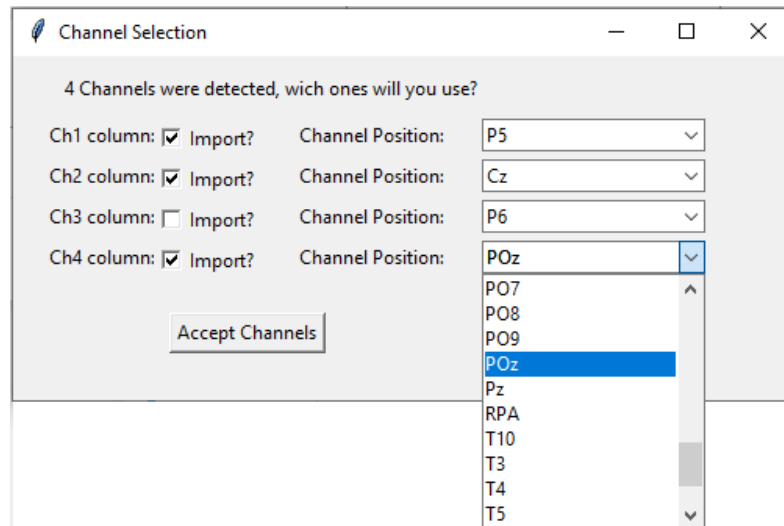


Figura 3.12: Ventana de configuración de canales. Se listan en ella todos los canales según nombres asignados para seleccionar cuales de ellos serán utilizados durante el procesamiento y definir su posición según el sistema 10-20.

La importancia de esta etapa radica en que:

- Permite separar aquellos canales que el usuario no desea utilizar durante el procesamiento de aquellos que si poseen información importante.
- Al asignar posiciones a cada uno de los canales, permite identificarlos con mayor facilidad durante el resto del estudio y construir en una etapa siguiente mapas topográficos.

En última instancia, se crean tres matrices Numpy a partir del DataFrame Pandas mencionado como entrada de esta etapa. La primera es creada para columnas con valores de los electrodos denominados canales, la segunda para posibles columnas que contengan etiquetas o tags y una tercera matriz para los valores de acelerómetros.

Salidas:

1. Arreglos Numpy de canales, etiquetas y acelerómetros.

Tasa de Muestreo

Entradas:

1. Columna de la señal de tipo Tiempo ("Time").

Librerías Utilizadas:

1. Numpy (Sección 3.1.4.1)

Desarrollo del Método:

La función principal de esta etapa es definir la tasa de muestreo con la que fueron obtenidos los datos de la señal. Para esto se debe definir cuánto tiempo transcurre entre la medición de una muestra y la siguiente.

La importancia de la etapa radica en que la mayor parte del análisis se lleva a cabo mediante técnicas de procesamiento en el espectro de tiempo y de frecuencia y para a lo cual es necesario tener un valor de referencia en tiempo. Cada dispositivo de medición puede obtener más o menos datos por segundo dependiendo de las características de hardware.

Cuando una columna del archivo de señal es asignada al tipo tiempo (**Time**), la herramienta intenta detectar automáticamente la tasa de muestreo utilizando los datos de esta columna. Para el calculo se restan los datos de tiempo de la segunda y la primera muestra, y luego se define la tasa de muestreo como el recíproco de dicho valor.

En algunos casos, los valores de la columna tiempo no son totalmente confiables. Un ejemplo de esto son los datos de tiempo provistos por el sistema OpenBCI, cuyo hardware posee una tasa de 200 muestras por segundo y las columnas de tiempo no concuerdan con esta característica debido a que la transmisión vía BLE de los timestamp no es estable. Para solucionar esta problemática, se agrega una opción de ingreso manual para aquellos casos que el usuario conozca de manera previa el valor de la tasa de muestreo y la misma no coincida con la tasa detectada automáticamente (Figura 3.13):

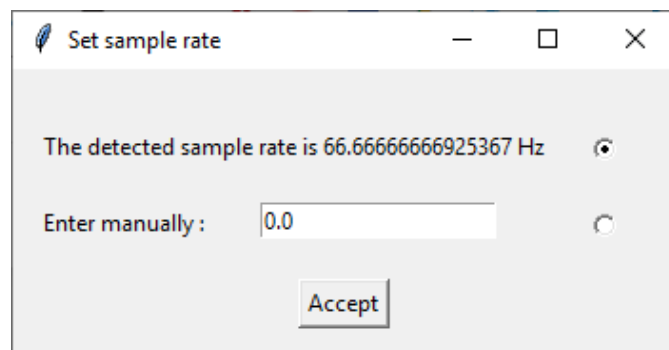


Figura 3.13: Ventana para la definición de la tasa de muestreo. En la parte superior se encuentra la tasa de muestreo detectada de manera automática y en la parte inferior la posibilidad de ingreso manual por parte del usuario.

Salidas:

1. Valor de la tasa de Muestreo

Definición de Intervalos

Entradas:

1. Tasa de Muestreo.
2. Arreglo Numpy de Etiquetas (opcional).
3. Archivo Excel con datos de intervalos (opcional).

Librerías Utilizadas:

1. Numpy (Sección 3.1.4.1)
2. Pandas (Sección 3.1.4.3)
3. Tkinter (Sección 3.1.4.8)

Desarrollo del Método:

Durante la adquisición de señales electroencefalográficas existen períodos de tiempo con información útil denominados épocas, por ejemplo cuando con el sujeto esta siendo estimulado o cuando se solicita algún tipo de respuesta al mismo, y períodos de tiempo sin información relevante, como son el posicionamiento del sujeto, de los electrodos o las etapas de descanso.

Por lo tanto, es importante separar la señal de estudio según intervalos de tiempo para evitar que los datos sin relevancia contaminen los resultados.

Esta etapa posee tres métodos para la definición de los intervalos de interés:

- Ingreso Manual
- Archivo Excel
- Columna asignada a Etiquetas ("*Tags*").

El Ingreso Manual de los datos consiste en solicitar al usuario la cantidad de intervalos a definir junto con el nombre y los puntos de inicio y fin (en segundos) de cada uno de ellos. Para la implementación del método no se requieren archivos ni formatos especiales, solo es necesario que el usuario conozca los intervalos de tiempo a estudiar.

El segundo método utiliza Archivos de Excel y consiste en cargar al programa un archivo de tabla tipo .xls o .xlsx que contenga dos columnas, una con los nombres de los intervalos y otra con los puntos de inicio en segundos.

La herramienta genera la cantidad de intervalos de acuerdo a la cantidad de filas que contenga el Archivo Excel. Además, puede incluirse una columna con los puntos de finalización o con la duración de cada intervalo. En caso de no existir ninguna de estas dos columnas, el usuario debe ingresar un valor de duración que se aplica a todos los intervalos por igual (Figura 3.14).

	Interval Title	Interval Start
0	Ojos Abiertos	40
1	Ojos Cerrados	150
2	Rojo 20Hz Con Estimulación	360
3	Rojo 20Hz Sin Estimulación	390
4	Rojo 25Hz Con Estimulación	480

Figura 3.14: Archivo de Excel para la definición de intervalos. En el ejemplo se definen 5 intervalos con sus nombres y tiempos de inicio. Debido a que no poseen una columna para el tiempo de finalización o de duración, el programa solicita al usuario el ingreso manual de la duración de dichos intervalos.

La herramienta admite como tercera opción la utilización de etiquetas para la definición de intervalos. Este método requiere que el archivo de señal contenga una columna de tipo **Tags**, donde se encuentren distintos valores que identifiquen a cada intervalo.

Una vez detectada la columna de etiquetas, se ejecuta una ventana donde se muestran los valores que fueron encontrados. Ésta ventana permite al usuario seleccionar los intervalos que se incluyen en la definición de intervalos, como así también el nombre asociado a cada uno.

Finalmente se definen los intervalos utilizando numeración creciente por cada vez que un número aparece.

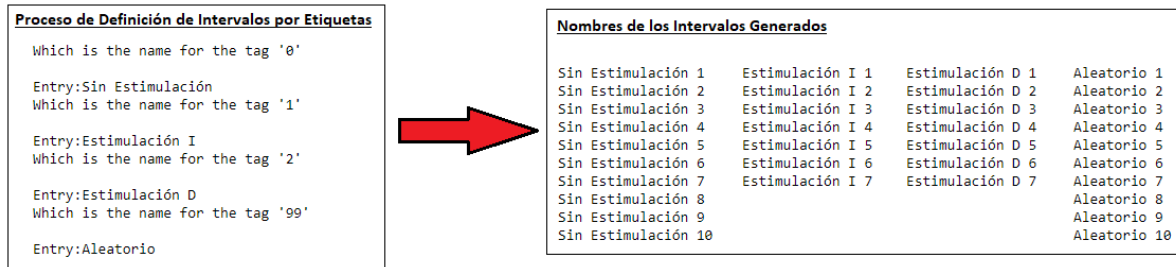


Figura 3.15: Ejemplo de como se solicitan los nombres para cada intervalo según etiquetas existentes, y como en un paso siguiente se definen los nombres con números de manera creciente cada vez que una etiqueta aparece.

Salidas:

1. Nombres y puntos de inicio y fin de cada intervalo a estudiar (en segundos).

Creación de variable MNE

Entradas:

1. Arreglo Numpy de canales de la Señal.
2. Nombres y Posiciones de los canales.
3. Tasa de muestreo

Librerías Utilizadas:

1. MNE (Sección 3.1.4.6)
2. Numpy (Sección 3.1.4.1)

Desarrollo del Método:

La función de ésta etapa consiste en reunir toda la información de la señal recopilada hasta el momento en una única variable denominada RawArray provista por la librería MNE. Ésta variable permite contener tanto los valores de la señal como aquellos

parámetros con los que fue medida (posiciones, tasa de muestreo, línea de tiempo, etc.) en un mismo entorno para que puedan ser fácilmente accedidos.

Además, la variable RawArray posee varias funcionalidades como: filtros aplicables a la señal, submuestreo, transformaciones a otros espectros, ploteo o visualización de la información, entre otros. Muchos de estos métodos son implementados durante el desarrollo de la herramienta.

Para crear una variable RawArray es necesario elaborar un diccionario Python que contenga la información del entorno de la señal (parámetros de medición). Luego se genera dicha variable utilizando el arreglo de canales y el diccionario de información.

Una Lista con todos los parámetros que pueden ser incluidos en el diccionario puede verse en la siguiente Figura:

```

<Info | 16 non-empty fields
  bads : list | 0 items
  ch_names : list | Ch1, Ch2, Ch4
  chs : list | 3 items (EEG: 3)
  comps : list | 0 items
  custom_ref_applied : bool | False
  dev_head_t : Transform | 3 items
  events : list | 0 items
  highpass : float | 5.0 Hz
  hpi_meas : list | 0 items
  hpi_results : list | 0 items
  lowpass : float | 60.0 Hz
  meas_date : NoneType | unspecified
  nchan : int | 3
  proc_history : list | 0 items
  projs : list | 0 items
  sfreq : float | 200.0 Hz
  acq_pars : NoneType
  acq_stim : NoneType
  ctf_head_t : NoneType
  description : NoneType
  dev_ctf_t : NoneType
  dig : NoneType
  experimenter : NoneType
  file_id : NoneType
  gantry_angle : NoneType
  hpi_subsystem : NoneType
  kit_system_id : NoneType
  line_freq : NoneType
  meas_id : NoneType
  proj_id : NoneType
  proj_name : NoneType
  subject_info : NoneType
  xplotter_layout : NoneType
>

```

Figura 3.16: Información contenida en el diccionario de MNE. Algunos de ellos pueden ser: Nombres y posiciones de los canales, fuente de los canales (EEG, EMG o EOG), frecuencias de la señal filtrada, parámetros de plotteo, entre otros.

Cabe destacar que a diferencia del manejo de datos en Python, la librería MNE trabaja con matrices de datos en forma horizontal, es decir, cada fila representa un canal y cada columna un instante de tiempo (o muestra de datos), y ésta es la manera típica de trabajo en software de EEG.

Esta nueva disposición ocasionó muchos inconvenientes durante el desarrollo del proyecto debido a errores producidos o resultados incoherentes al momento de implementar librerías distintas a MNE. Para solucionar este problema, se implementó la funcionalidad Transponer Matrices de Numpy al momento de insertar o extraer datos de la variable RawArray. Una descripción del problema se presenta en la Figura 3.17 a continuación:

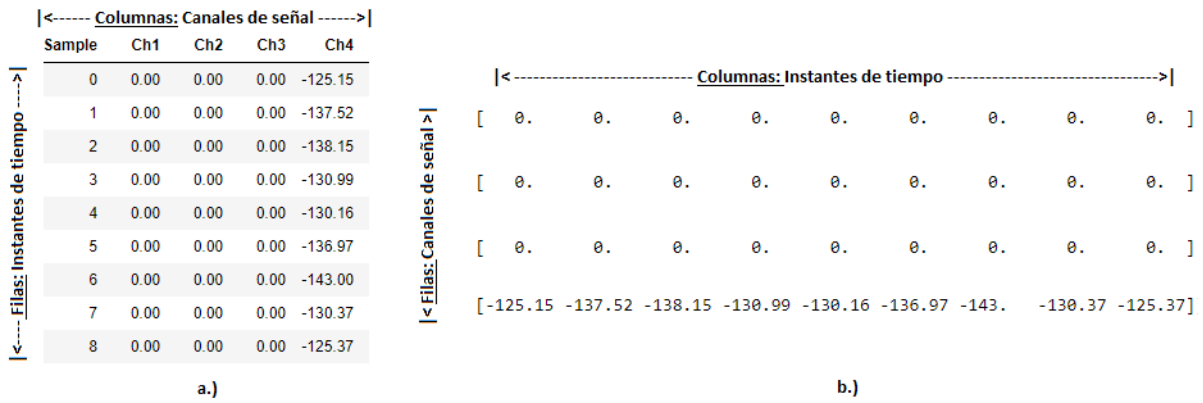


Figura 3.17: La **Figura a** representa un DataFrame de Pandas con disposición vertical y la forma mas frecuente de manejo de datos en la mayoría de las librerías de Python y la **Figura b** representa una variable RawArray de la librería MNE con los datos de una señal de 4 canales. Se observa que las filas del Dataframe representan muestras y cada columna representa un canal, mientras que las filas en el RawArray representan a cada canal y las columnas las muestras.

Salidas:

1. Variable MNE de la señal con datos crudos.

3.2.3.3. Iteración 2: Acondicionamiento de la Señal

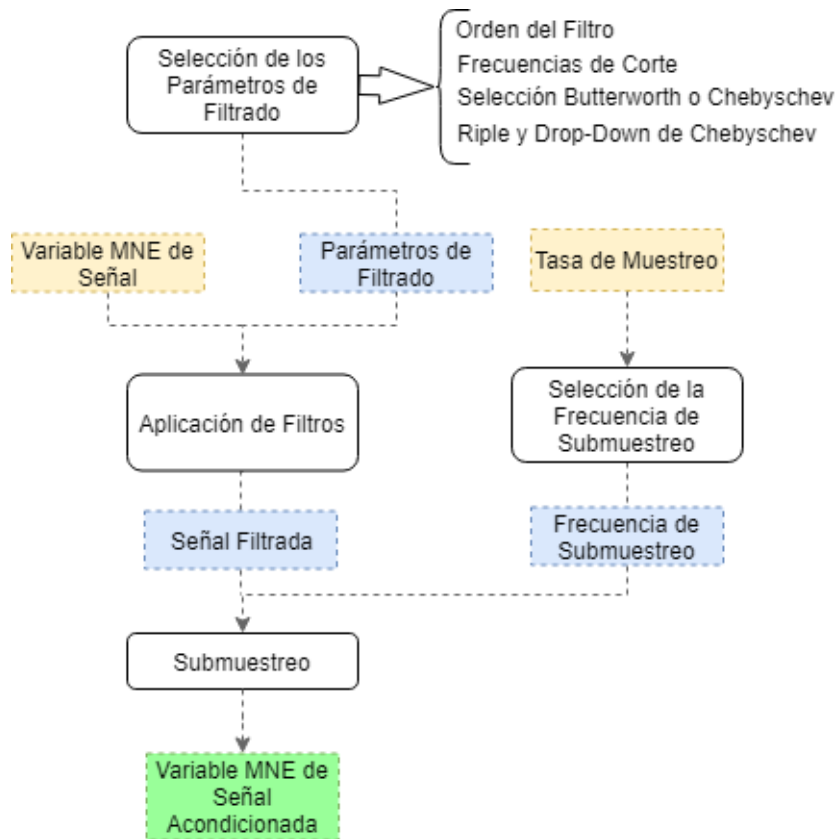


Figura 3.18: Diagrama: Acondicionamiento de la Señal

La segunda iteración comprende aquellas operaciones que realizan una primera etapa de acondicionamiento de la señal. En ella se incluyen dos funciones muy importantes: filtrado en frecuencia, para destacar la información útil y eliminar aquella que no es relevante; y submuestreo que mediante la reducción de la cantidad de datos busca disminuir el trabajo requerido y aumentar la velocidad de ejecución.

Selección de Parámetros de Filtrado

Entradas:

1. Parámetros de filtros establecidos por el usuario

Librerías Utilizadas:

1. Tkinter (Sección 3.1.4.8)

Desarrollo del Método:

La primera función de ésta etapa consiste en definir los parámetros de filtrado de la señal (descritos en la Sección 2.1), entre ellos :

- Tipo de filtro a implementar (Filtro Butterworth o Chebyshev).
- Orden del filtro.
- Frecuencias de corte baja y alta para filtro Pasa Banda.
- Frecuencia de corte para filtro Notch.
- Valores de ripple y dropdown en caso de utilizar filtro Chebyshev.

La herramienta solicita al usuario el ingreso de estos parámetros a través de una ventana ejecutable por la librería Tkinter como puede observarse en la siguiente Figura :

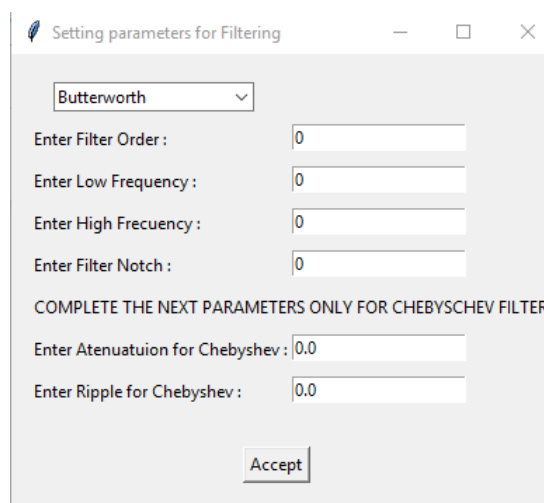


Figura 3.19: Ventana para el ingreso de parámetros de filtrado. La misma permite la selección del tipo de filtro entre un Butterworth y Chevyshev, el orden del mismo, frecuencias de corte alta y baja, y la frecuencia del filtro Notch. En el caso de implementar un filtro Chevyshev se deben ingresar los valores de ripple y atenuación .

Para aplicaciones en EEG, una configuración frecuentemente utilizada es:

- Filtro Butterworth pasa banda de orden 5.
- Frecuencia de corte baja de entre 5 y 15 Hz para eliminar ruido proveniente de actividad eléctrica muscular (EMG).
- Frecuencia de corte alta de 60Hz aproximadamente, que varía de acuerdo al tipo de señal en estudio.
- Filtro Notch en 50 Hz (o 60 Hz) para eliminar ruido de la línea eléctrica.

Salidas:

1. Parámetros de Filtrado

Aplicación de Filtros

Entradas:

1. Variable MNE de la señal con datos crudos (RawArray).
2. Parámetros de Filtrado.

Librerías Utilizadas:

1. MNE (Sección 3.1.4.6)
2. Scipy (Sección 3.1.4.2)

Desarrollo del Método:

Una vez ingresados los parámetros de filtrado, se continúa con la aplicación de los filtros seleccionados. Tanto el filtro Butterworth Pasa Banda como el Notch son aplicados a través de funciones propias de la librería MNE, la cual accede directamente a funciones de la librería Scipy manteniendo la estructura de la variable RawArray.

El motivo por el cual se decide aplicar los filtros Pasa Banda de la librería MNE radica en la capacidad de aplicar cuatro filtros diferentes con el mismo comando, es decir:

- Si la frecuencia de corte inferior es menor que la superior, se aplica un filtro pasa banda entre dichas frecuencias .
- Si el usuario ingresa solamente una frecuencia de corte inferior (manteniendo la frecuencia superior en 0), se aplica un filtro pasa alto en dicha frecuencia.
- Si el usuario solo ingresa una frecuencia de corte superior (manteniendo la frecuencia inferior en 0), se aplica un filtro pasa bajo en dicha frecuencia.
- Si la frecuencia de corte superior es menor que la frecuencia de corte inferior, se aplica un filtro elimina banda entre dichas frecuencias.

- Si ambas frecuencias se mantienen en 0, no se aplica ningún filtro a la señal.

En la Figura a continuación pueden observarse cada uno de los casos posibles:

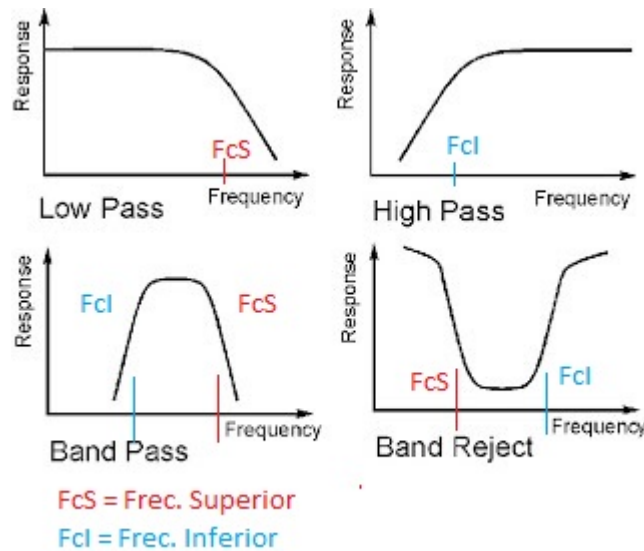


Figura 3.20: Distintos tipos de filtros según valores de frecuencia de corte inferior y superior aplicados por la librería MNE. Si $FcI < FcS$ se aplica un filtro pasa banda, si $FcS < FcI$ se aplica un filtro elimina banda, Si $FcI = Fc$ y $FcS = 0$, se aplica un filtro Pasa Alto, en cambio si $FcS = Fc$ y $FcI = 0$ se aplica un Filtro Pasa Bajo.

Para la aplicación del filtro Notch se solicita únicamente el ingreso de la frecuencia de corte. Esto ocurre debido a que la librería MNE no trabaja con el orden del filtro sino con la distancia (en Hz) desde la frecuencia de corte seleccionada hasta el punto donde la atenuación se vuelve nula. El valor de distancia se fijó en 1 Hz teniendo en cuenta que el objetivo del filtro Notch es eliminar el ruido de línea, cuya desviación en frecuencia no suele ser de mayor de 1 Hz.

Salidas:

1. Señal Filtrada

Selección de la Frecuencia de Submuestreo

Entradas:

1. Sin entradas específicas.

Librerías Utilizadas:

1. Tkinter (Sección 3.1.4.8)

Desarrollo del Método:

Con el objetivo de reducir la cantidad de datos y optimizar así los tiempos de cálculos, es posible submuestrear la señal. La función Submuestreo consiste en crear una nueva señal a partir de extraer muestras periódicas de la señal original (Figura 3.21).

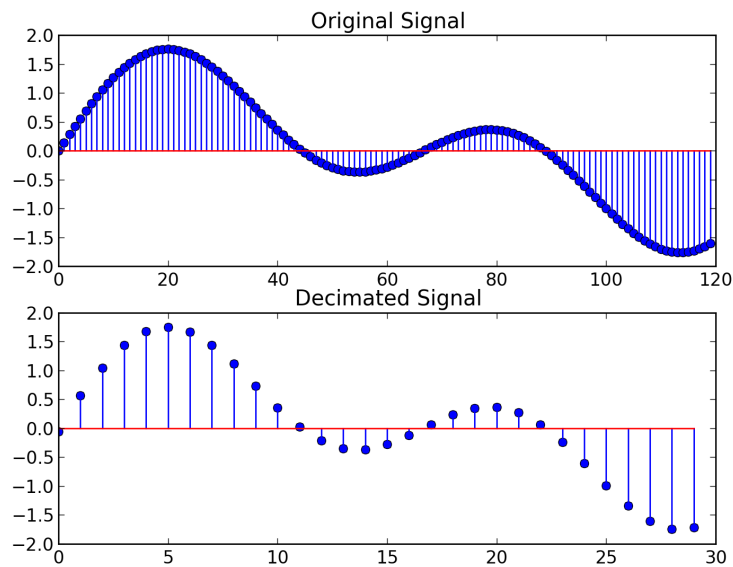


Figura 3.21: Ejemplo de una señal submuestreada. ⁴

Un requisito fundamental es que la frecuencia de submuestreo sea siempre menor que la tasa de muestreo, debido a que es imposible generar mas datos que los obtenidos a la velocidad original de muestreo.

Al momento de aplicar submuestreo es necesario tener en cuenta que para estudiar una cierta frecuencia f contenida en una señal, la frecuencia de submuestreo mínima f_s debe ser el doble de f según lo enuncia el Teorema de Nyquist - Shannon.

$$\text{Teorema de Nyquist: } f_s = 2 * f$$

Esto se debe a que una frecuencia menor a una frecuencia denominada crítica, no lograría la reconstrucción completa de la frecuencia f que se desea estudiar. Para comprender este concepto se visualiza un ejemplo en la Figura 3.22. En la misma puede observarse una señal submuestreada a distintas tasas f_s/f , y destacar que si la tasa toma valores menores a 2.0 (donde f_s es el doble que f), la señal representada por dicho submuestreo (puntos rojos) es de una de menor frecuencia (en verde) con respecto a la señal original (en azul).

⁴Fuente: Gráficos obtenidos utilizando la función *resample* de Scipy sobre una señal de prueba.

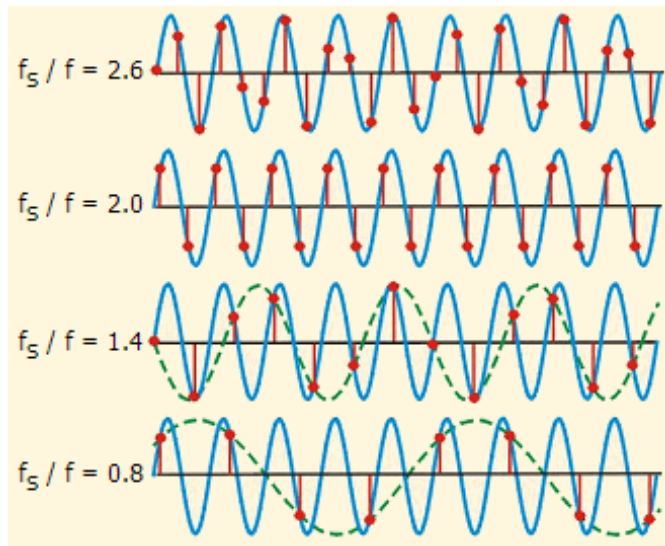
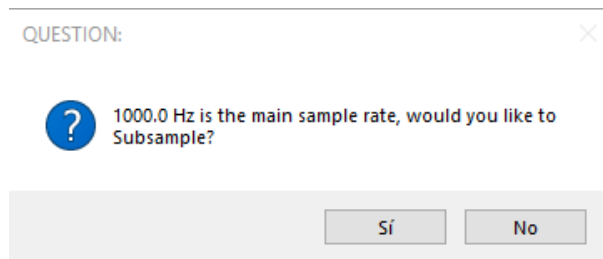


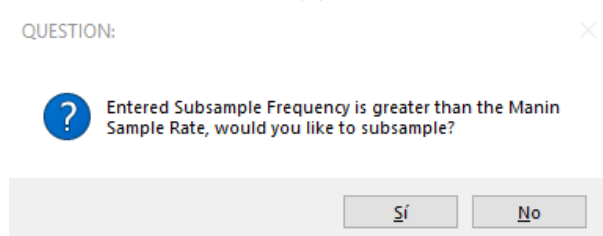
Figura 3.22: Ejemplo de aplicación del Teorema de Nyquist. En las dos primeras imágenes, donde la tasa de submuestreo f_s/f es mayor o igual a 2 la señal submuestreada se corresponde con la señal original. Sin embargo cuando la tasa disminuye por debajo de éste valor crítico, la señal representada difiere de la original disminuyendo su frecuencia.
5

Mediante una ventana ejecutable en Tkinter se pregunta al usuario si desea realizar el submuestreo de la señal (Figura 3.23a). Si la respuesta es afirmativa, se genera un **input** de Python para el ingreso de la frecuencia de submuestreo junto con un mensaje de advertencia sobre la frecuencia máxima.

Si el usuario ingresa un valor igual o mayor al de la tasa de muestreo, se despliega una nueva ventana (Figura 3.23b) mostrando el error y preguntando si el usuario desea submuestrear nuevamente.



(a)



(b)

Figura 3.23: Confirmación de Submuestreo de la Señal.

⁵Fuente: Imágenes obtenidas utilizando Applet diseñada por el Prof. C.F. Efstathiou, para la demostración del efecto Nyquist sobre el Muestreo y Submuestreo.

Como existen señales donde la cantidad de datos eliminados no justifica la pérdida de información, el usuario puede elegir no submuestrear la señal en análisis.

Salidas:

1. Valor de la Frecuencia de Submuestreo

Submuestreo

Entradas:

1. Señal Filtrada.

Librerías Utilizadas:

1. MNE (Sección 3.1.4.6)

Desarrollo del Método:

El submuestreo se implementa utilizando una funcionalidad propia de la librería MNE y la misma está desarrollada sobre una función de la librería Scipy (Sección 3.1.4.2).

Las ventajas de utilizar una función propia de una librería consiste en que ésta puede detectar a la señal contenida en la variable RawArray sin necesidad de extraer sus datos, y obtener así la tasa de muestreo fundamental contenida en el diccionario de la misma variable (Creación de Variable MNE en Sección 3.2.3.2).

Salidas:

1. Variable MNE de la Señal Acondicionada (filtrada y submuestreada)

3.2.3.4. Iteración 3: Análisis en Tiempo

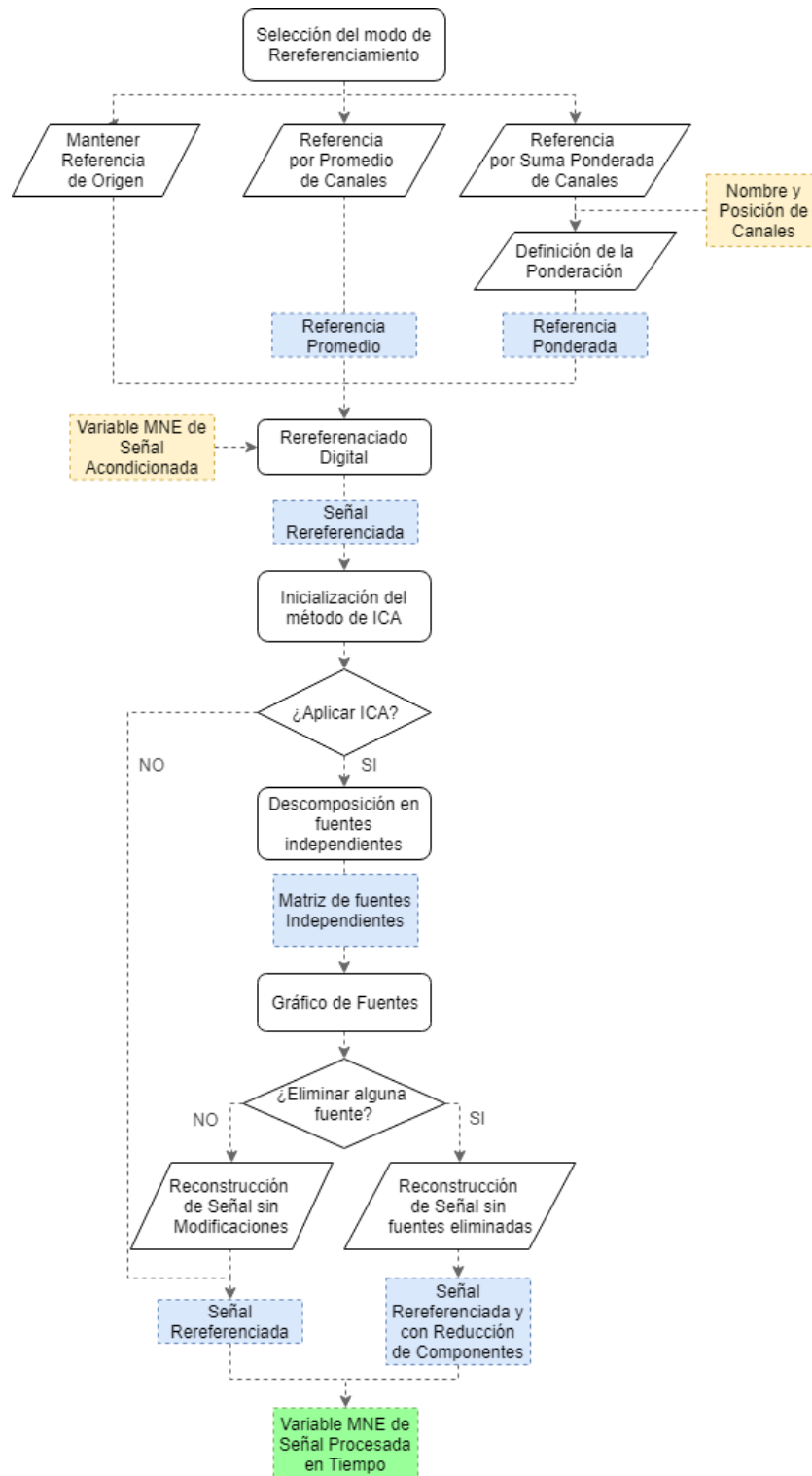


Figura 3.24: Diagrama: Análisis en Tiempo

En la tercera iteración se incluyen aquellas técnicas de eliminación o atenuación del ruido en el espectro del tiempo. Existe una gran cantidad de métodos para la eliminación de ruido en el Procesamiento de Señales, pero en el presente proyecto se decide implementar dos de los más utilizados en el área de la Electroencefalografía: el Rereferenciado Digital y la Descomposición en Componentes Independientes (ICA).

Selección del Método de Re-referenciado

Entradas:

1. Sin entradas específicas

Librerías Utilizadas:

1. Tkinter (Sección 3.1.4.8)

Desarrollo del Método:

Como se menciona en la Sección 2.2, existen dos tipos de re-referenciado: implementando el **Promedio** de los canales (CAR), o **Laplaciano** a los mismos. La herramienta permite seleccionar uno de los dos métodos, además de la opción de no generar ninguna nueva referencia.

Al momento de implementar **Laplaciano** a un canal, éste requiere contar con canales adyacentes al mismo y asignar una ponderación a cada uno para atenuar así la influencia de los mismos sobre el canal seleccionado.

Debido a que la variable Montage de la librería MNE utiliza coordenadas dependientes de la posición del electrodo anterior, resulta muy complejo establecer las distancias relativas entre electrodos y por lo tanto asignar ponderaciones a cada uno de ellos. Debido a esto, se opta por no implementar Laplacianos exclusivamente y si permitir al usuario seleccionar la **Suma Ponderada** que sea conveniente al momento de analizar la señal.

La ventana diseñada para la selección de uno de los 3 procesos y sus parámetros se observa en la siguiente Figura :

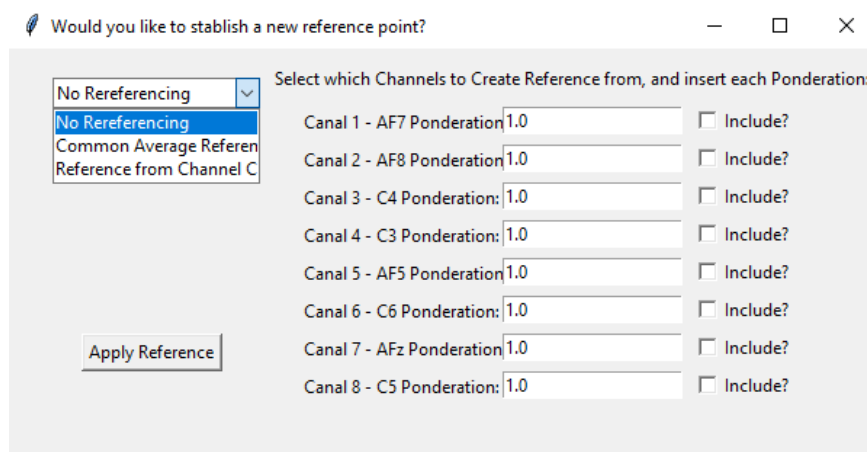


Figura 3.25: Ventana para definición del método de Re-referenciado. En la esquina superior izquierda se encuentra un selector del método a utilizar y debajo del mismo un botón para aceptar la selección. En la derecha de la ventana se encuentran los espacios para definir las ponderaciones en caso de elegir la re-referencia por suma ponderada de canales.

El sector derecho de la ventana, donde se encuentra la lista de canales detectados, se utiliza únicamente en el caso que el usuario desee utilizar la re-referencia por suma

ponderada, donde es necesario seleccionar los canales a incluir en la suma y la ponderación asignada.

Salidas:

1. Método de Re-referenciado Seleccionado.

Re-referenciado

Entradas:

1. Variable MNE de la señal acondicionada.
2. Método de rereferenciado designado.

Librerías Utilizadas:

1. MNE (Sección 3.1.4.6)
2. Numpy (Sección 3.1.4.1)

Desarrollo del Método:

De acuerdo al método de re-referenciado seleccionado en la función anterior, se implementa:

- Re-referenciado por promedio (CAR) utilizando la función CAR propia de la librería MNE.
- Re-referenciado por suma ponderada utilizando la librería Numpy para extraer arreglos de los canales, aplicar la referencia manualmente, y devolver finalmente la información re-referenciada en una variable RawArray de la librería MNE.
- Sin nueva referencia, utilizando una funcionalidad de la librería MNE para designar a la variable RawArray como *no re-referenciable* e impedir de esta manera que otra función modifique la referencia establecida.

Salidas:

1. Señal Re-referenciada

Eliminación de Artefactos por Método ICA

Entradas:

1. Señal Rereferenciada.

Librerías Utilizadas:

1. MNE (Sección 3.1.4.6)

2. Numpy (Sección 3.1.4.1)
3. FastICA

Desarrollo del Método:

En esta etapa se implementa la Función ICA de la librería MNE. La misma utiliza filtros espaciales y no requiere un canal de referencia para cada fuente de artefactos. El método predeterminado en dicha librería es FastICA, que junto con Infomax son dos de los algoritmos más utilizados en separación de componentes independientes. Picard en cambio, es un algoritmo actualmente en desarrollo y se espera converja más rápido que FastICA e Infomax, especialmente cuando el objetivo es recuperar mapas precisos con un parámetro de baja tolerancia.

La función es capaz de detectar las diferentes fuentes cerebrales y aquellas provenientes de artefactos. A continuación, el usuario decide si elimina o no la contribución de estas fuentes a la señal, para finalmente combinar aquellos componentes que no han sido eliminados y volver a generar las señales sin artefactos.

En una primera instancia se pregunta al usuario si desea aplicar el método ICA a la señal procesada y en caso de ser afirmativa, se despliega una segunda ventana para la ejecución del método (Figura 3.26).

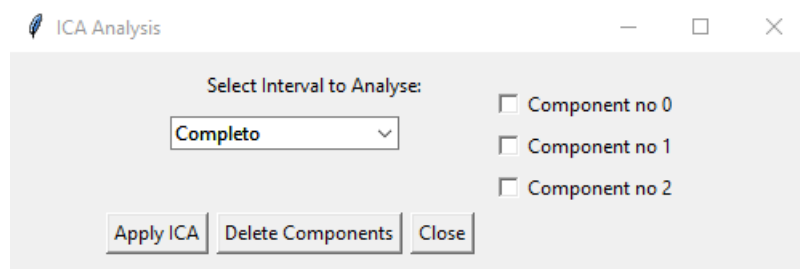


Figura 3.26: Ventana para Análisis de Componentes Independientes de la señal (ICA).

La ventana de implementación posee varios botones y parámetros a tener en cuenta:

- **Selector de Intervalo de Análisis:** Contiene la lista de todos los intervalos seleccionados en etapas anteriores.
- **Botón Apply ICA:** Ejecuta el algoritmo ICA y calcula de manera estadística los componentes independientes de la señal. Genera 3 Gráficos: uno para la visualización de la señal original con artefactos en tiempo, otro para la visualización de componentes independientes en tiempo y un tercero para la visualización de componentes independientes en el espectro de frecuencia.
- **Botón Delete Components:** Elimina los componentes seleccionados en la lista ubicada en la derecha de la ventana. Se eliminan solamente aquellos componentes seleccionados y en caso de no seleccionar ninguno de ellos, la señal no se modifica. Finalmente se genera un gráfico en tiempo de la señal resultante, permitiendo la comparación de la misma con la señal original.

- **Botón Close:** Permite cerrar la ventana y seguir con la ejecución del método.
- **Lista de Componentes:** Debido a que el método es capaz de detectar tantas fuentes de señal como receptores de la misma estén involucrados, la cantidad de componentes coincide con los canales seleccionados en etapas previas.

Como ICA es un método basado en estadística, es posible que una fuente de artefacto sea detectada algunas veces en un determinado componente y al volver a ejecutarlo, el mismo artefacto sea detectado en un componente distinto. Debido a esto, el Botón **Apply ICA** permite al usuario detectar los componentes independientes las veces que desee hasta detectar el artefacto buscado en alguno de ellos.

A modo de utilidad se generan gráficos de la señal original contaminada y de los componentes independientes en tiempo. Además, se decide generar un gráfico de dichos componentes en el espectro de frecuencia para facilitar el análisis de los mismos.

Salidas:

1. Variable MNE de la señal procesada en tiempo.

3.2.3.5. Iteración 4: Mapas Topográficos

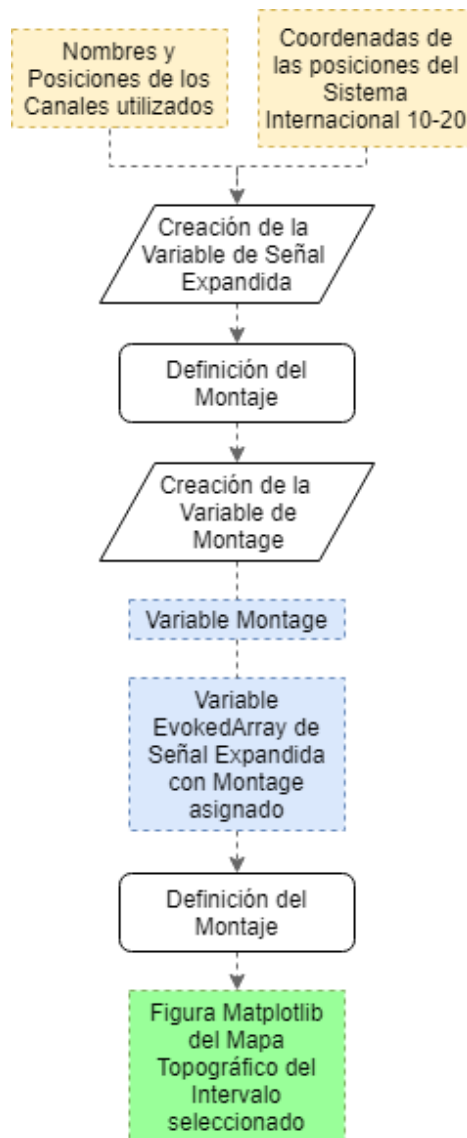


Figura 3.27: Diagrama: Mapas Topográficos

En la cuarta iteración se generan mapas topográficos como modelos gráficos de la cabeza. En estos mapas se visualizan las posiciones de los electrodos según un sistema de posicionamiento junto con el detalle de qué zonas del cerebro se activan en cada instante de tiempo. La herramienta implementa el sistema de posicionamiento 10-20 porque es el estándar utilizado y genera los mapas utilizando la función *plot_topo* de la librería MNE.

Definición de Montaje

Entradas:

1. Nombres y Posiciones de los canales utilizados.
2. Coordenadas de las posiciones del Sistema Internacional 10-20.

Librerías Utilizadas:

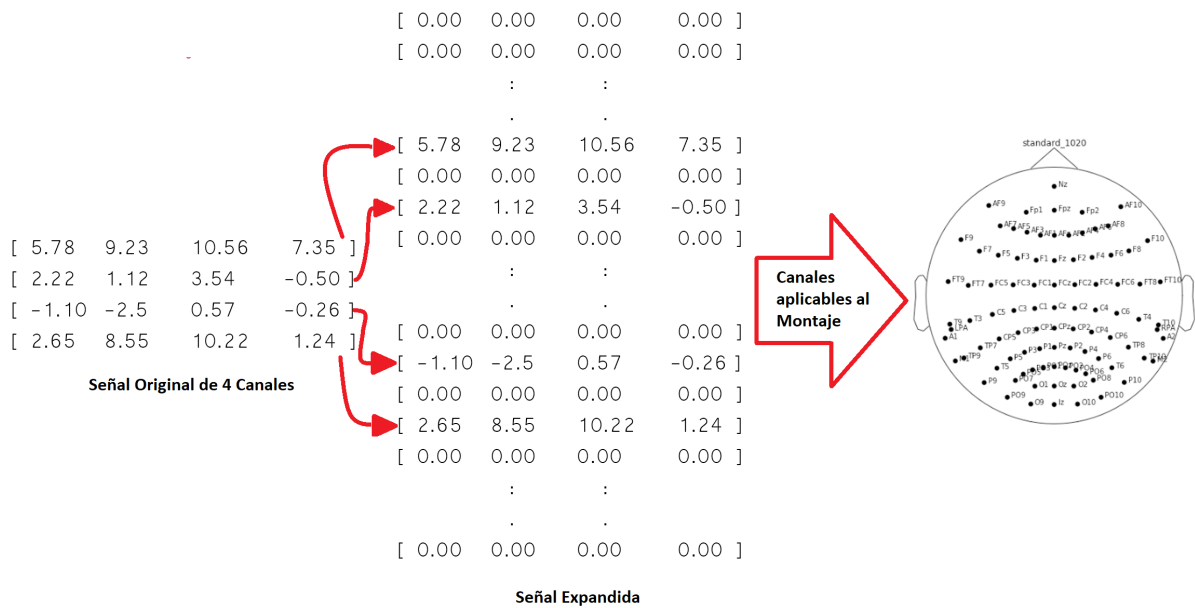


Figura 3.29: Proceso de creación de la Variable de Señal Expandida

Salidas:

1. Variable de Señal Expandida.
2. Montaje del Sistema Internacional 10-20 (Incluido como Info en la Variable de Señal Expandida).

Generación de Mapas Topográficos

Entradas:

1. Variable MNE de la Señal Procesada en Tiempo Expandida.
2. Montaje del Sistema Internacional 10-20.
3. Intervalo a Graficar.

Librerías Utilizadas:

1. MNE (Sección 3.1.4.6)

Desarrollo del Método:

En la Figura 3.30 se observa un mapa topográfico en colores según la amplitud de la señal en cada instante de tiempo (o en un promedio de instantes de tiempo). Las posiciones en color dependen de aquellas que fueron asignadas a cada canal en la iteración 1. Las demás posiciones no se colorean debido a que se asignaron a dichas posiciones arreglos de ceros. Cada posición toma un color con mayor o menor intensidad según algún patrón. El patrón más común es el rojo para mayores amplitudes, y colores azules para amplitudes de menor valor.

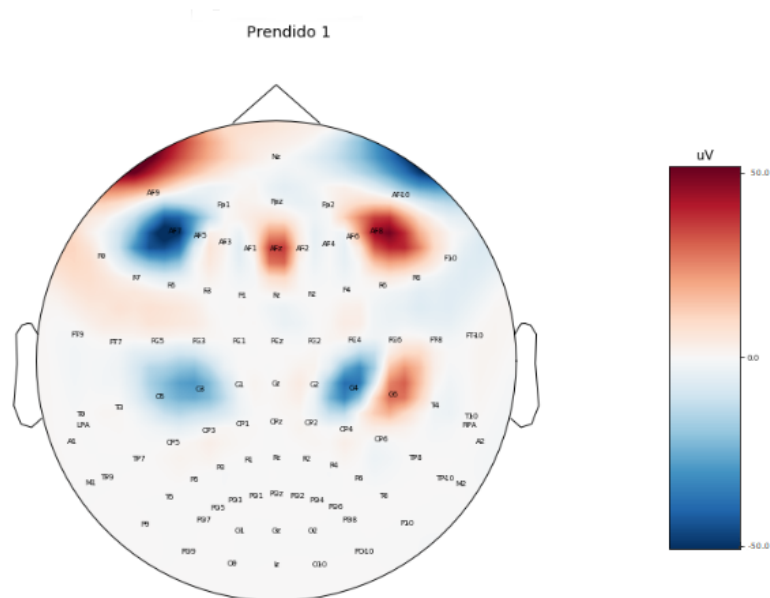


Figura 3.30: Mapa Topográfico que representa los valores de señal en un determinado instante de tiempo según escala de colores. A la derecha se observa la escala de colores para cada valor de amplitud de señal, y dentro del modelo se observan las líneas que MNE utiliza para proveer posibles áreas funcionales del cerebro.

La librería MNE ofrece una funcionalidad para la creación de mapas topográficos a partir de una variable `RawArray`. Muchos parámetros pueden modificarse en esta función para lograr visualizar o remarcar algún aspecto de la señal:

- **Escala de Valores:** Determina que nivel de amplitud se asigna al color de mayor y menor valor. Esta escala puede ocultarse para comodidad del usuario, y puede asignarse la unidad de medición para la misma.
- **Cantidad de Zonas Delimitadas:** La funcionalidad de MNE intenta delimitar zonas de procesamiento en el cerebro para cada instante. La zonas que genera MNE se pueden delimitar, o incluso eliminar. Debido a que en algunos casos son pocos los canales seleccionados se pueden delimitar zonas por demás y generar así errores de interpretación.
- **Cantidad de Mapas Topográficos por Intervalo:** Determina la cantidad de gráficos generados en un intervalo de tiempo. La herramienta se encuentra predefinida a 4 gráficos por intervalo, con la posibilidad de ampliar esta cantidad.
- **Cantidad de Instantes de Tiempo a Promediar por Mapa Topográfico:** Permite generar un único Mapa con el promedio de valores del intervalo total o generar Mapas con el promedio de valores por cada intervalo de tiempo seleccionado.

Esta función a diferencia de la creación de transformadas en frecuencia y espectrogramas, no genera una matriz de objetos con los mapas topográficos de todos los intervalos, sino que la misma genera en ese instante los mapas de los intervalos seleccionados en la

Ventana de selección, por lo que la salida de esta etapa son gráficos, y no una matriz de objetos.

Salidas:

1. Variable de tipo figura con el mapa topográfico seleccionado.

3.2.3.6. Iteración 5: Análisis en Frecuencia

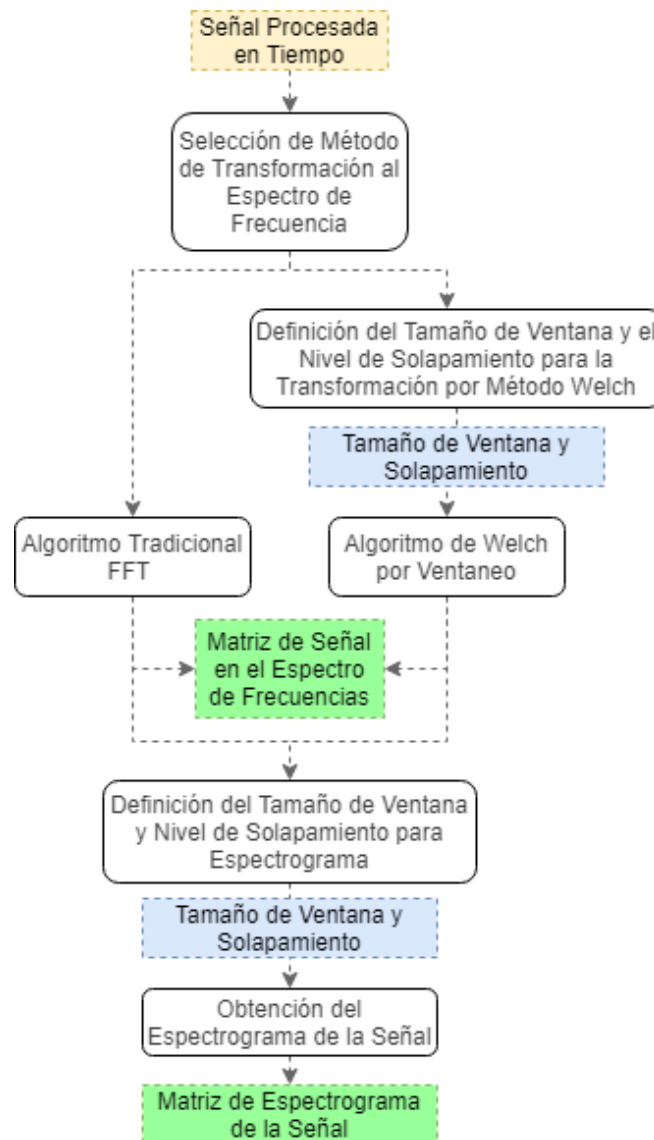


Figura 3.31: Diagrama: Análisis en Frecuencia

En la quinta iteración se incluyen aquellas funciones que transforman la señal al espectro de frecuencia y permiten conocer como está conformada. Ésta información es útil al momento de analizar las respuestas generadas a estímulos, o el nivel de ruido presente en la señal durante un determinado intervalo de tiempo. Por ejemplo, los ritmos cerebrales son detectados en frecuencia y no en tiempo.

Existen una gran cantidad de métodos y algoritmos para transformar la señal al espectro de frecuencia aunque en este proyecto se seleccionaron aquellos que son utilizados

con mayor frecuencia en el análisis de señales: Dos que consisten en la transformación al espectro de frecuencia, denominados FFT y de Welch, y un tercero denominado Espectrograma que combina el espectro de frecuencia y tiempo.

Algoritmo FFT

Entradas:

1. Variable MNE de la señal procesada en tiempo.

Librerías Utilizadas:

1. Numpy (Sección 3.1.4.1)

Desarrollo del Método:

Para transformar la señal al espectro de frecuencia se utiliza el algoritmo FFT de la librería Numpy. Ésta recibe como entrada un arreglo con muestras de la señal y la distancia de tiempo entre cada muestra. A partir de estos datos, el algoritmo lleva a cabo la transformación y devuelve a la salida un arreglo de la señal transformada junto con los residuos de frecuencia. Éstos residuos son los puntos de frecuencia sobre los que puede medirse la transformada.

Por cada intervalo definido se obtiene un arreglo con los residuos de frecuencia y la transformada al espectro para cada canal, y con los datos de todos los intervalos se genera una matriz que contiene todos los productos de esta etapa.

Salidas:

1. Matriz de vectores de transformación al espectro de frecuencia mediante algoritmo de FFT, por intervalo y canal.
2. Vector de residuos de frecuencia por cada Intervalo.

Algoritmo Welch

Entradas:

1. Variable MNE de la señal Procesada en Tiempo.

Librerías Utilizadas:

1. Scipy (Sección 3.1.4.2)

Desarrollo del Método:

El algoritmo Welch se diferencia de la FFT en la generación de ventanas que recorren la señal llevando a cabo transformadas en pequeñas porciones de la misma, pudiendo estar o no superpuestas, y promediando finalmente los resultados obtenidos (Sección 2.3.3).

Esto suele mejorar la relación Señal-Ruido obtenida en la transformación, especialmente disminuyendo el ruido aleatorio presente.

Para lograr esta transformada se implementa la función Welch de la librería Scipy y la misma recibe como entrada los siguientes datos:

- El Arreglo de la Señal.
- La Frecuencia de Muestreo (o de Submuestro si es aplicado), a diferencia del algoritmo FFT que requiere la distancia entre muestras.
- El tamaño de la ventana que se utilizará para la transformación, en cantidad de muestras.
- La cantidad de muestras a solapar entre una Ventana y la siguiente.

Por defecto, el tipo de ventana utilizada por el algoritmo es la Hanning (o Hann) cuya forma puede observarse en la Figura 3.32.

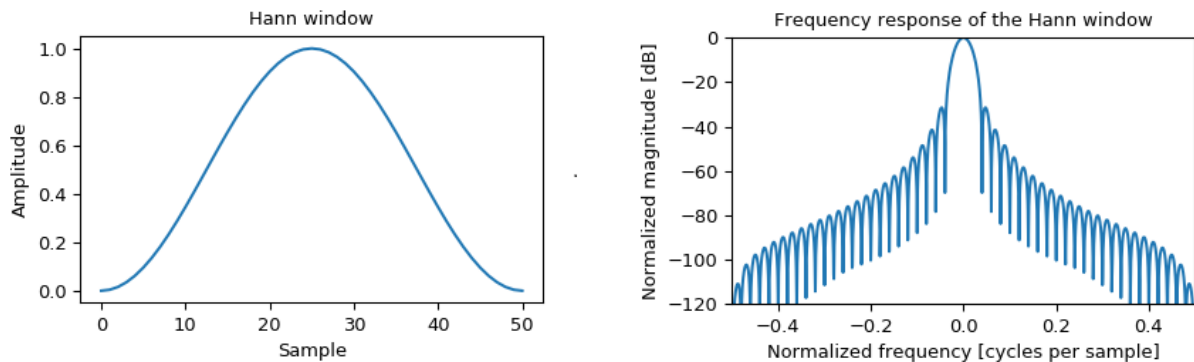


Figura 3.32: Forma en Tiempo y Respuesta en Frecuencia de la ventana Hanning. Se observa que la misma permite destacar las muestras intermedias de la ventana, atenuando las más externas.

Como resultado de la etapa se genera una matriz similar a la mencionada en la FFT, conformada por arreglos con transformadas y residuos de frecuencia para cada intervalo.

Salidas:

1. Matriz de Vectores de transformación al espectro de frecuencia generado mediante algoritmo Welch, por cada intervalo y canal.
2. Vector de residuos de frecuencias por cada intervalo.

Espectrograma

Entradas:

1. Variable MNE de la señal Procesada en Tiempo.

Librerías Utilizadas:

1. Scipy (Sección 3.1.4.2)

Desarrollo del Método:

El último paso de esta etapa consiste en generar espectrogramas de la señal por cada intervalo y canal. Para lograrlo se implementa la función `spectrogram` de la librería Scipy y la misma recibe como entrada los mismos datos utilizados por el algoritmo de Transformada Welch: El arreglo de valores de la señal, la frecuencia de muestreo o submuestreo, tamaño de la ventana y solapamiento de la misma en cantidad de muestras.

Sin embargo, a diferencia del algoritmo de Welch la salida de la función produce tres elementos:

- Los valores de espectrogramas, que no se ubican en arreglos (vectores unidimensionales) sino en matrices (dos dimensiones) donde cada punto representa el valor de la energía a una cierta frecuencia en un determinado instante de tiempo.
- Arreglo de residuos de frecuencia que representan los valores de frecuencia sobre los que se puede calcular el espectrograma.
- Arreglo de residuos de tiempo que representan los instantes de tiempo sobre los que se calculan cada uno de los espectros de energía en frecuencia.

Salidas:

1. Matriz de valores de espectrogramas por cada intervalo y canal.
2. Vector de residuos de frecuencias y vector de residuos de tiempo por cada intervalo.

3.2.3.7. Iteración 6: Visualización

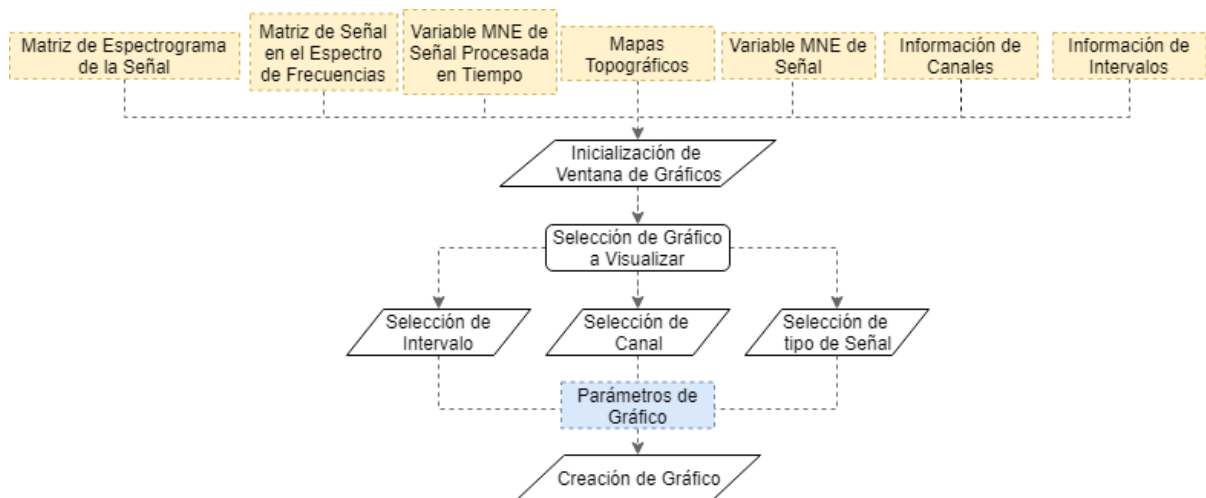


Figura 3.33: Diagrama: Visualización

Durante la sexta iteración se desarrolla la etapa de visualización e incluye aquellas funciones que representan mediante gráficos los distintos aspectos de la señal. Ésta etapa pretende facilitar al usuario el análisis de datos mediante gráficos, teniendo en cuenta las dificultades que se presentan al momento de analizarlos en una lista de valores.

Entradas:

1. Variable MNE de la señal cruda (sin procesar).
2. Variable MNE de la señal procesada en tiempo.
3. Matriz de transformadas al espectro de frecuencia por canal e intervalo.
4. Matriz de espectrogramas por canal e intervalo.

Librerías Utilizadas:

1. Plotly (Sección 3.1.4.5)
2. Tkinter (Sección 3.1.4.8)

Desarrollo del Método:

En las etapas anteriores se somete la señal a transformaciones y procesamientos con el objetivo de realzar la información relevante y atenuar todo lo posible el ruido presente en la misma. Por lo tanto, la etapa de visualización adquiere mucha importancia dentro de la herramienta debido a que permite al usuario visualizar los datos de la señal producidos en estas distintas etapas.

La ventana de control principal para la visualización se observa en la Figura 3.34. En ella pueden observarse tres variables de selección junto con tres botones en la parte inferior y sus funciones se describen a continuación:

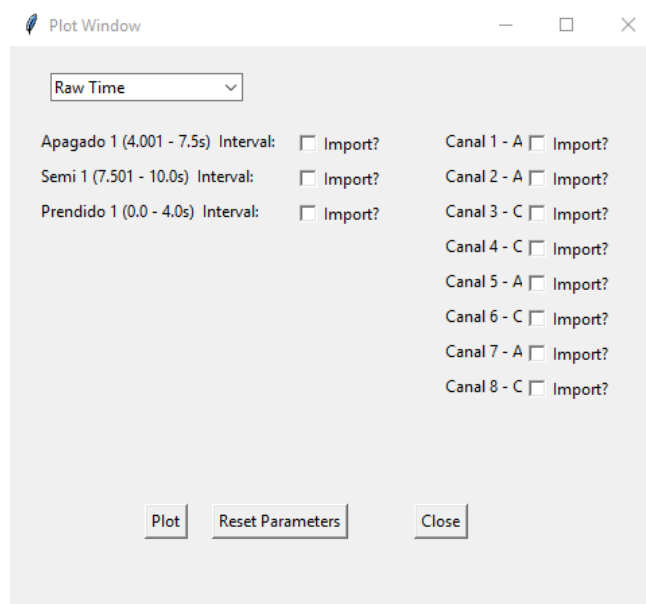


Figura 3.34: Ventana de control para visualización de gráficos

- El selector permite elegir en qué forma o sobre qué espectro se desea visualizar la señal, es decir: en el espectro de tiempo (Señal Cruda y Procesada), en el espectro de frecuencia, en formato de espectrograma o en forma de mapas topográficos.
- La lista de valores que se encuentra por debajo del anterior permite seleccionar que Intervalo de los definidos previamente (Sección 3.2.3.2) se quiere visualizar. En la figura se observa que estos se enumeran por nombre, instante de inicio y fin (en segundos).
- La lista de valores a la derecha contiene los canales presentes en la señal de EEG, presentados a través del nombre y la posición asignada previamente. Éstos permiten seleccionar cuál de ellos quiere visualizarse.
- El botón **Plot** ejecuta la función de gráfico, generando tantos ploteos como sean necesarios para mostrar los datos en el formato y según los canales e intervalos seleccionados en ambas listas correspondientes.
- El botón **Reset** restaura todas las modificaciones de procesamiento y acondicionamiento aplicadas a la señal sin eliminar los gráficos generados, permitiendo al usuario el reingreso de parámetros y la generación de nuevos gráficos. De esta manera es posible comparar los resultados de distintos parámetros y seleccionar los mejores métodos.
- El botón **Close** cierra la ventana y finaliza la etapa de Visualización pasando a la próxima. Cabe aclarar que los gráficos generados no se eliminan, permitiendo al usuario cerrarlos manualmente cuando lo desee.

A continuación se listan algunos comentarios adicionales con respecto a la implementación de esta etapa:

- En el selector se muestran dos formatos de señal en el espectro de tiempo, una opción para datos crudos (sin procesar ni acondicionar) y otra para datos procesados y acondicionados. Ésta característica no se repite en el resto de los espectros ya que no resulta necesario para el análisis y se evita además sobrecargar al programa con opciones.
- En los gráficos de espectro en tiempo y frecuencia se genera un nuevo gráfico por cada intervalo seleccionado y en el mismo se visualizan superpuestos las señales.
- En la visualización de los espectrogramas, la superposición de señales generaría una dificultad en el análisis y comprensión de los datos. Por esto, se genera un nuevo gráfico por cada intervalo y canal seleccionado.
- En la visualización de mapas topográficos, la lista de canales no tiene verdadera utilidad ya que en ellos se visualizan todos los canales de la señal, sin importar cuáles hayan sido seleccionados. Entonces se genera un gráfico por cada intervalo seleccionado.

- Los Mapas Topográficos están configurados de manera predeterminada para mostrar el promedio de valores en todo el intervalo en un único gráfico.

Salidas:

- Gráfico de la señal en distintos espectros o formatos, sobre los canales e intervalos seleccionados.

3.2.3.8. Iteración 7: Correlación y Coherencia

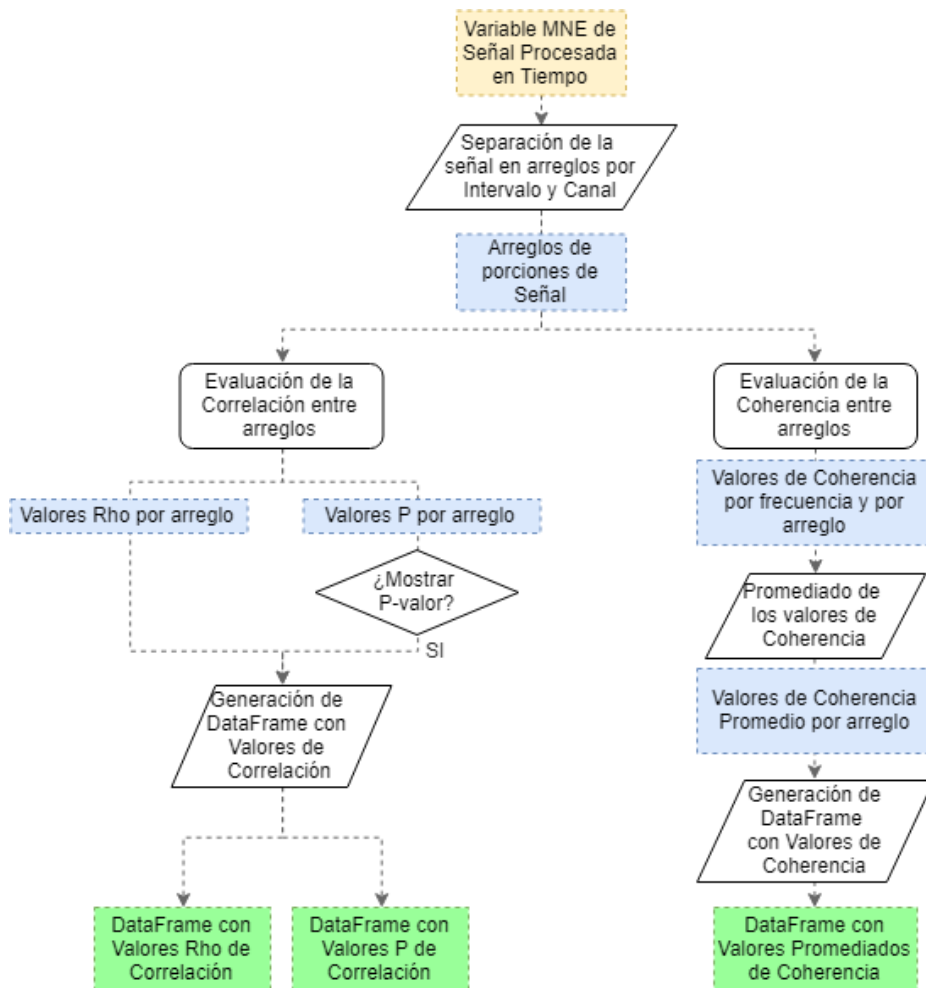


Figura 3.35: Diagrama: Correlación y Coherencia

En el presente bloque no se implementan los métodos de correlación y coherencia como mediciones utilizadas para cuantificar la similitud entre dos señales. En cambio, se implementan para comparar una señal medida por distintos electrodos en la misma instancia de adquisición y asegurar la sincronización temporal de ambas señales. Para lograrlo, se evaluaron cada una de estas medidas de a pares de canales y por cada uno de los intervalos, ubicando los datos obtenidos en tablas de variable DataFrame de Pandas como se describe a continuación:

Correlación

Entradas:

1. Variable MNE de la señal Procesada en Tiempo.

Librerías Utilizadas:

1. Scipy (Sección 3.1.4.2)
2. Pandas (Sección 3.1.4.3)

Desarrollo del Método:

Debido a que las señales de EEG se encuentran contaminadas con ruido y secuencias aleatorias, estas pueden clasificarse como señales aleatorias o estocásticas. Para el estudio de correlación entre canales de la señal se implementa la función Spearmanr de la librería Scipy. Ésta Función aplica un métodos para el calculo de correlación denominado Spearmanr, que se diferencia de otros métodos, como el método Pearson, porque asume que los datos no poseen una distribución probabilística normal.

La función Spearmanr calcula el coeficiente de Correlación Rho cuyos posibles resultados se encuentran entre -1 y 1, donde 1 representa señales totalmente correlacionadas, 0 indica que no existe correlación y -1 indica señales opuestamente correlacionadas (opuestas en fase).

Además, esta función calcula el P-valor o valor P de la correlación que expresa cuál es la probabilidad de que ambas señales no estén correlacionadas. Sin embargo, se utiliza una ventana (Figura 3.36) para advertir al usuario que los P-valores no son completamente confiables, especialmente si la cantidad de datos es menor a 500, ofreciendo al usuario la opción de calcular únicamente el coeficiente Rho.

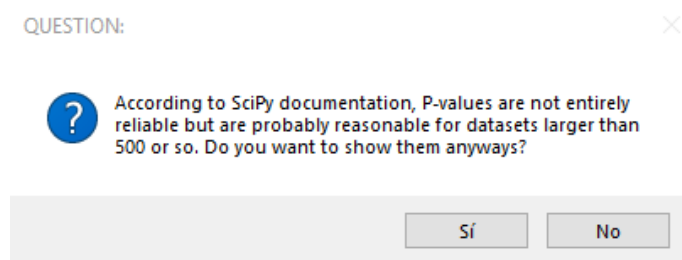


Figura 3.36: Advertencia de uso del P-Valor en Coherencia

La Correlación se calcula por cada intervalo tomando canales de a pares hasta completar todas las combinaciones posibles (incluso la de cada canal con sí mismo), a la vez que todos los resultados se ubican en una tabla DataFrame de Pandas como la que se detalla en la Figura 3.37.

	Sin Defase 1	Desfase 45° 1	Desfase 90° 1	Desfase de 180° 1	Desfase de 270° 1	Desfase de 315° 1
Ch1 vs Ch1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Ch1 vs Ch2	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch1 vs Ch3	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch2 vs Ch1	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch2 vs Ch2	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Ch2 vs Ch3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Ch3 vs Ch1	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch3 vs Ch2	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Ch3 vs Ch3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figura 3.37: DataFrame Pandas con resultados de Correlación. Cada fila corresponde a la correlación de un canal con otro (incluso un canal con sí mismo) y cada columna representa un intervalo.

Salidas:

1. Tabla DataFrame de pandas con los valores Rho de correlación entre pares de canales de la señal por intervalo.
2. Tabla DataFrame de pandas con los valores P de correlación entre pares de canales de la señal por intervalo (opcional).

Coherencia

Entradas:

1. Variable MNE de la señal procesada en tiempo.

Librerías Utilizadas:

1. Scipy (Sección 3.1.4.2)
2. Pandas (Sección 3.1.4.3)

Desarrollo del Método:

En esta etapa se evalúa la Coherencia entre señales medidas en cada canal, y esto se logra utilizando la función coherence de Scipy. El mismo genera un arreglo de niveles de Coherencia en función de la frecuencia.

Si se visualizan los resultados de todos los arreglos obtenidos por combinación de pares de canales y por intervalos, la información se vuelve muy confusa y poco clara, demandando además la generación de muchos gráficos que permitan ver cada arreglo con claridad. Éstos arreglos de coherencia presentan demasiadas variaciones y no aportan mucha información por si mismos.

Para brindar una solución a este problema, se generan promedios de cada uno de los arreglos obtenidos como resultado y se crea una Tabla DataFrame con los mismos, similar al estudio de Correlación (Figura 3.38).

	Sin Desfase 1	Desfase 45° 1	Desfase 90° 1	Desfase 180° 1	Desfase 270° 1	Desfase 315° 1
Ch1 vs Ch1	1.0	1.000000	1.000000	1.000000	1.000000	1.000000
Ch1 vs Ch2	1.0	0.309153	0.827498	0.036905	0.038873	0.144892
Ch1 vs Ch3	1.0	0.309153	0.827498	0.036905	0.038873	0.144892
Ch2 vs Ch1	1.0	0.309153	0.827498	0.036905	0.038873	0.144892
Ch2 vs Ch2	1.0	1.000000	1.000000	1.000000	1.000000	1.000000
Ch2 vs Ch3	1.0	1.000000	1.000000	1.000000	1.000000	1.000000
Ch3 vs Ch1	1.0	0.309153	0.827498	0.036905	0.038873	0.144892
Ch3 vs Ch2	1.0	1.000000	1.000000	1.000000	1.000000	1.000000
Ch3 vs Ch3	1.0	1.000000	1.000000	1.000000	1.000000	1.000000

Figura 3.38: DataFrame Pandas con Resultados de Coherencia. Cada Fila representa un producto de coherencia entre un par de canales, y cada columna un intervalo.

Salidas:

1. Tabla DataFrame de Pandas con los valores Coherencia Promedio entre pares de canales de la señal por cada intervalo.

3.2.3.9. Iteración 8: Exportación de Datos

En la última etapa del proyecto, se exportan y guardan en una carpeta aquellos datos que son registrados durante el procesamiento de las señales, estos datos incluyen:

- Parámetros de configuración de cada corrida, para que puedan ser cargados nuevamente y agilizar las corridas posteriores.
- Gráficos y Tablas de resultados de cada corrida, para que puedan ser analizados posteriormente.

La exportación se realiza mediante la creación de una carpeta con el nombre de la señal ingresado en la primer instancia en la misma ubicación donde se encuentra el Jupyter Notebook del programa. En el caso que dicha carpeta exista, se agrega al nombre la palabra *Run* (corrida) y un número correspondiente (en una primera instancia Run 2, en segunda instancia Run 3 y así sucesivamente). Se informa al usuario mediante un output del programa cuál fue el nombre resultante de la carpeta creada.

La información almacenada en dicha carpeta es la siguiente:

1. Una copia del archivo de señal utilizado en la corrida.
2. Un archivo Excel con los datos de los intervalos definidos (opcional): Éste archivo se crea de cero si los intervalos se definen manualmente o con etiquetas (Tags), utilizando una columna para el nombre, otra para el instante de inicio en segundos, y otra para el instante de finalización en segundos. En el caso que se utilice otro archivo excel para la carga de intervalos, el mismo es copiada en la carpeta de exportación.

- Un archivo de tipo **HTML** por cada gráfico generado en la etapa de visualización junto con nombre del intervalo y el canal seleccionado. Cuando se exporta en formato HTML, se mantienen las mismas propiedades que son generados en la corrida (escalar o transportar ejes, activar o desactivar canales, zoom, etc.)

Un ejemplo de exportación puede visualizarse en la Figura 3.39.

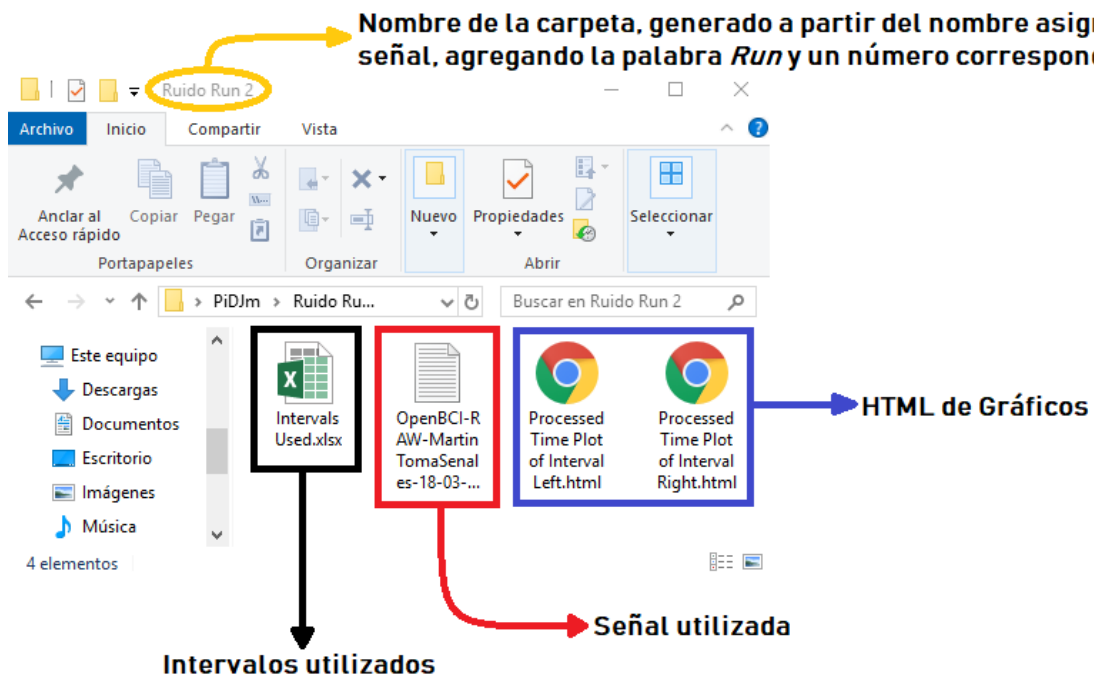


Figura 3.39: Exportación de la corrida de una señal a la cuál se asignó el nombre Ruido junto con todos los elementos mencionados previamente en la sección. Debido a que se trata de la segunda corrida de la misma señal, se agrega *Run 2* al nombre de la carpeta.

3.2.4. Revisión de Requisitos iniciales

Como conclusión del capítulo, se presentan mediante el Cuadro 3.2 los impactos resultantes luego de implementar cada uno de los métodos mencionados.

En la misma puede observarse como solo en cinco de los 18 requisitos iniciales no se pudo eliminar el Impacto de Fallo. A continuación se detallan las razones por las cuales estos requisitos no pudieron ser completamente resueltos:

- **Requisito 2:** abarca la definición de los tipos y nombres de los canales del archivo de señal. El mismo no puede satisfacerse completamente debido a que no se implementa un método que evite el ingreso de la misma posición para dos canales diferentes. Esto genera luego errores en la creación del archivo montaje de los mapas topográficos.
- **Requisitos 7:** Rereferenciado Digital. En este caso no se logra implementar el método de filtros laplacianos, que generan una nueva referencia a partir de la suma ponderada de canales utilizando valores que se definen a partir de la distancia que separa a dichos canales. En cambio, sólo se implementa una variación de la suma, utilizando ponderaciones que el usuario debe definir según lo considere.

- **Requisito 13:** Visualización. Este requisito se cumple casi en su totalidad, pudiendo generar gráficos en cualquiera de los espectros trabajados seleccionando cualquier combinación de canales. Sin embargo, el requisito inicial requería el uso de dos ventanas que permitan modificar los parámetros de filtrado y transformación y generar gráficos de dichos cambios de manera simultánea. Se propone como solución, en la ventana de generación de gráficos (Figura 3.34), la posibilidad de reiniciar y redefinir los parámetros de Acondicionamiento, Análisis en Tiempo y en Frecuencia, aplicando los cambios deseados, para poder luego visualizarlos.
- **Requisito 14:** referido a escalas de gráficos modificables interactivamente. El impacto resultó en *Bajo* ya que, si bien las escalas modificables se pudieron implementar en todos los gráficos, las escalas de colores de los espectrogramas, sólo pueden ser modificadas una vez finalizada la ejecución de una corrida.
- **Requisito 17:** Exportación de los parámetros de Acondicionamiento y Análisis. Debido a que no se encontró un formato de archivo adecuado para la exportación y posterior importación de los parámetros de análisis utilizados durante cada corrida, no se implementó ningún método de guardado de los mismos. Aún así, el impacto resulta *Bajo* debido a que este era el nivel asignado al requisito en la instancia inicial.

Cuadro 3.2: Revisión de Requisitos

No.	Referencia	Impacto Resultante	Iteración que lo satisface	Test
Req 1	Formato y carga	Nulo	Iter 1	4.2.1
Req 2	Canales	Bajo	Iter 1	4.2.10
Req 3	Tasa de Muestreo	Nulo	Iter 1	4.2.2
Req 4	Intervalos	Nulo	Iter 1	4.2.3
Req 5	Filtros	Nulo	Iter 2	4.2.4
Req 6	Submuestreo	Nulo	Iter 2	4.2.5
Req 7	Rereferenciado	Bajo	Iter 3	4.2.7
Req 8	ICA	Nulo	Iter 3	4.2.6
Req 9	Mapas Topográficos	Nulo	Iter 4	4.2.10
Req 10	FFT y Welch	Nulo	Iter 5	4.2.8
Req 11	Parámetros Welch	Nulo	Iter 5	4.2.8
Req 12	Espectrograma	Nulo	Iter 5	4.2.9
Req 13	Visualización	Bajo	Iter 6	4.2.8
				4.2.9
				4.2.10
Req 14	Escalas modificables	Bajo	Iter 6	-
Req 15	Correlación y Coherencia	Nulo	Iter 7	4.2.11
				4.2.12
Req 16	Exportación de Carga	Nulo	Iter 8	-
Req 17	Exportación de Análisis	Bajo	Iter 8	-
Req 18	Exportación de Gráficos	Nulo	Iter 8	-

Se concluye entonces que los objetivos planteados ha sido resueltos de forma satisfactoria y que solo un 27% de ellos no se han podido resolver de manera completa. Sin embargo, los impactos resultantes no implican una modificación importante en el funcionamiento de la herramienta.

Capítulo 4

Verificación

La herramienta desarrollada durante el proyecto implementa varios métodos con el objetivo de procesar señales electroencefalográficas. Estos métodos utilizan funciones seleccionadas de librerías disponibles en Python y adaptadas a las señales de interés, algunos fueron diseñados por los autores del proyecto y otros surgen como una combinación entre funciones disponibles y métodos adaptados a las mismas.

Los métodos utilizados en una herramienta desarrollada en software deben someterse a un proceso de verificación. El objetivo de la verificación es el de asegurar que la herramienta es robusta, que la información de salida es confiable, y que los métodos propuestos realizan las operaciones y transformaciones de manera correcta.

La verificación de este proyecto se realiza utilizando señales conocidas como entradas del sistema, para evaluar si los resultados obtenidos a la salida se corresponden con los esperados para dichas entradas.

4.1. Señales utilizadas para la verificación:

En esta sección se presentan las señales utilizadas para el proceso de verificación. Estas señales son referidas en cada método según el nombre designado en esta sección, y una misma señal puede ser utilizada para validar uno o varios de ellos.

Archivos con datos aleatorios en distintos formatos:

Con el objetivo de verificar que la herramienta es capaz de leer cada uno de los formatos establecidos (.txt, .xls o .xlsx), se implementan señales formadas por datos aleatorios cargados en diferentes tipos de archivos:

- **Datos aleatorios separados por espacios:**

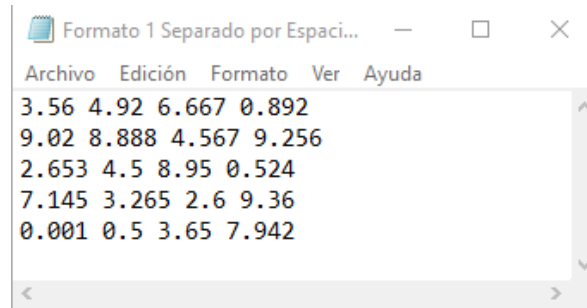


Figura 4.1: Señal ejemplo con datos aleatorios separados por espacios.

- **Datos aleatorios separados por punto y coma:**

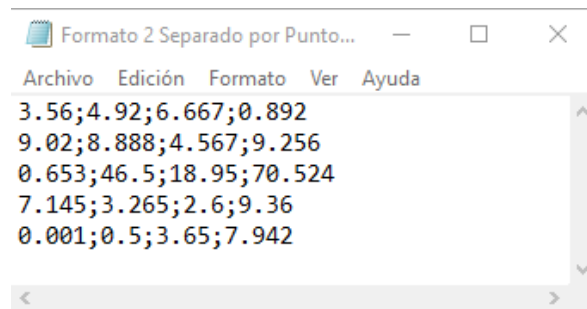


Figura 4.2: Señal ejemplo con datos aleatorios separados por punto y coma.

- **Datos aleatorios en tabla de Excel:**

	A	B	C	D
1				
2	3,560	4,920	6,667	0,892
3	9,200	8,888	4,567	9,256
4	2,653	4,500	8,950	0,524
5	4,145	3,265	2,600	9,360
6	0,001	0,500	3,650	7,942

Figura 4.3: Señal ejemplo con datos aleatorios en Excel

Señal senoidal a distintas frecuencias:

La señal es generada en Excel y está formada por ondas senoidales a distintas frecuencias. Cada onda está representada en una columna y se incluyen además columnas con datos de tiempo y etiquetas para definir intervalos.

En la Figura 4.4 puede observarse además de las columnas mencionadas, bloques de configuración para las frecuencias utilizadas en cada seno. Los datos temporales en la columna **Time 1** están muestreados a 1000 Hz. En cambio la columna de tiempo **Time 2** contiene datos temporales a 100 Hz y la misma se agrega al archivo con el objeto de

evaluar la capacidad de la herramienta para detectar la presencia de dos columnas con datos temporales y permitir al usuario elegir una de ellas.

Time 1	Seno 1	Seno 2	Seno 3	Seno 4	Seno 5	Seno 6	Seno 7	Seno 8	Tag	Time 2				Frecuencia 1	10
0	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	2	0,00			Frecuencia 2	15
0,001	0,063	0,094	0,125	0,187	0,249	0,309	0,368	0,536		2	0,01			Frecuencia 3	20
0,002	0,125	0,187	0,249	0,368	0,482	0,588	0,685	0,905		2	0,02			Frecuencia 4	30
0,003	0,187	0,279	0,368	0,536	0,685	0,809	0,905	0,992		2	0,03			Frecuencia 5	40
0,004	0,249	0,368	0,482	0,685	0,844	0,951	0,998	0,771		2	0,04			Frecuencia 6	50
0,005	0,309	0,454	0,588	0,809	0,951	1,000	0,951	0,309		2	0,05			Frecuencia 7	60
0,006	0,368	0,536	0,685	0,905	0,998	0,951	0,771	-0,249		2	0,06			Frecuencia 8	90
0,007	0,426	0,613	0,771	0,969	0,982	0,809	0,482	-0,729		2	0,07				
0,008	0,482	0,685	0,844	0,998	0,905	0,588	0,125	-0,982		2	0,08				
0,009	0,536	0,750	0,905	0,992	0,771	0,309	-0,249	-0,930		2	0,09				
0,01	0,588	0,809	0,951	0,951	0,588	0,000	-0,588	-0,588		2	0,10				

Figura 4.4: Archivo Excel con señales senoidales a Distintas frecuencias. De izquierda a derecha cada columna representa: Columna de Tiempo Principal (Time 1), 8 Columnas con señales senoidales a distintas Frecuencias, Columna con Etiquetas para definir Intervalos y una Columna de Tiempo Secundario (Time 2).

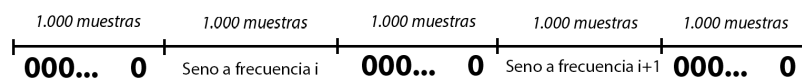
Señales senoidales de frecuencia creciente:

La señal es generada en Excel y está formada por una columna de ondas senoidales, una de datos temporales (muestreada a 1000 Hz) y dos de etiquetas.

Según puede observarse en la Figura 4.5b, la señal se encuentra dividida en intervalos de 2.000 muestras, dentro de los cuales las primeras 1.000 muestras son nulas y las restantes contienen ondas senoidales a distintas frecuencias. La frecuencia inicial es de 10 Hz incrementándose 10 Hz por cada intervalo.

Time 1	Seno 1	Tag 1	Tag 2	Frecuencia Inicial:	0
0	0,000	0	0	Frecuencia Salto:	10
0,001	0,000	0	0		
0,002	0,000	0	0		
0,003	0,000	0	0		
0,004	0,000	0	0		
0,005	0,000	0	0		
0,006	0,000	0	0		
0,007	0,000	0	0		
0,008	0,000	0	0		
0,009	0,000	0	0		
0,01	0,000	0	0		

(a)



(b)

Figura 4.5: Señales senoidales de frecuencia creciente. En la Figura (a) se presenta un Archivo Excel utilizado para contener la señal, posee una columna de Tiempo, una columna de señales senoidales y dos columnas con Etiquetas. En la Figura (b) se describe la secuencia utilizada para producir el escalonamiento.

Como puede observarse en la Figura 4.5a, el archivo Excel posee dos columnas con etiquetas para la definición de intervalos. La primera, **Tag 1**, contiene etiquetas utilizadas tanto para la selección de intervalos como para definir la frecuencia de cada onda senoidal.

Estas etiquetas toman valores de 1 a 5 para los segmentos de intervalos de ondas senoidales y un valor igual a 0 para aquellos segmentos con datos nulos. En cambio, **Tag 2** contiene etiquetas que permiten distinguir entre intervalos con datos nulos (con etiqueta 0) y no nulos (con etiqueta 1).

Si bien la verificación se realiza utilizando la columna **Tag 1**, se agrega la columna **Tag 2** para evaluar si la herramienta es capaz de distinguir correctamente dos columnas de etiquetas, y luego solicitar al usuario cuál utilizar.

Señales senoidales desfasadas y con amplitud variada:

Señal generada en Archivo Excel que contiene tres ondas senoidales junto con una columna de datos temporales y una columna de etiquetas. Cada una de las etiquetas determina no solo un intervalo de muestras sino también un valor de desfase entre las señales. Las 3 ondas senoidales corresponden a:

- Una señal senoidal a una frecuencia y amplitud constante.
- Una señal senoidal con la misma frecuencia que la primera aunque desfasada un determinado valor en cada intervalo.
- Una señal senoidal con la misma frecuencia pero distinta amplitud y desfasada un determinado valor con respecto a la primer señal.

Un ejemplo de las mismas puede visualizarse en la Figura 4.6.

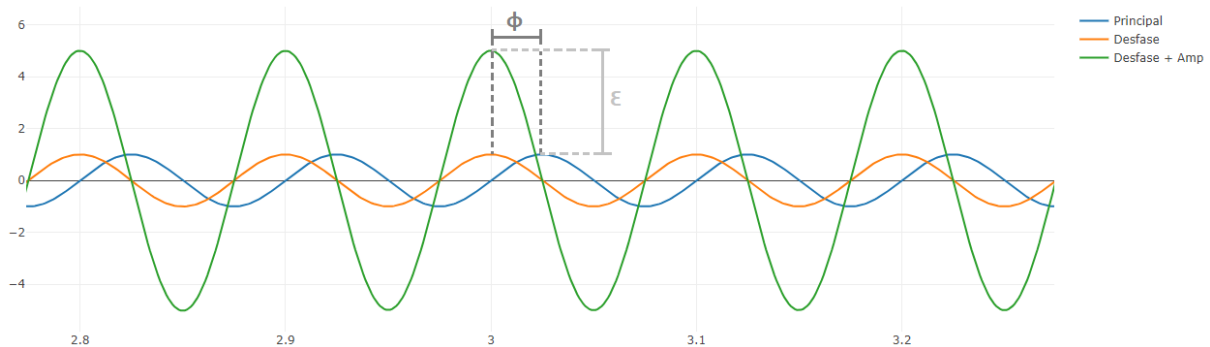


Figura 4.6: Señales senoidales desfasadas y escaladas en amplitud. Se observa como la señal en naranja se encuentra desfasada de la señal principal (azul) una distancia ϕ , mientras que la señal en verde además del desfase, se diferencia en un escalamiento ϵ de la amplitud.

Ondas Cuadradas:

Señal formada por 3 canales con ondas periódicas de forma cuadrada. Todas las ondas se encuentran a una frecuencia de 5 Hz, pero cada una de ellas posee amplitudes de 1, 3 y 5 respectivamente. Un gráfico de la señal puede verse en la Figura 4.7.

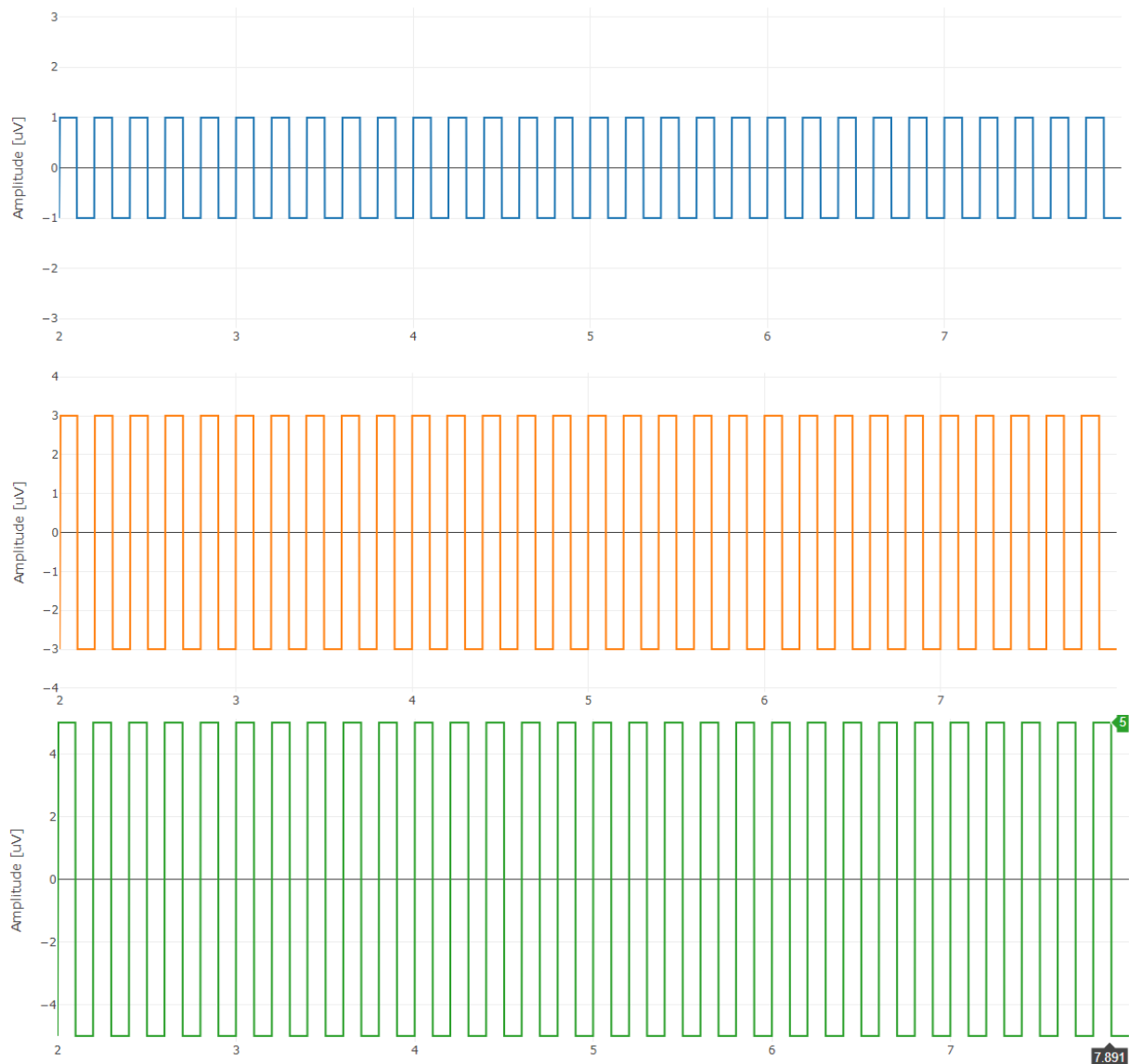


Figura 4.7: Señal Ejemplo de Ondas Cuadradas.

Señal "Dummy" de Ondas Alfa contaminada con Pestañeo y ruido aleatorio:

Se trata de una señal artificial que emula la forma de una señal electroencefalográfica real formada por ondas alfa contaminada por ruido de pestañeo y ruido aleatorio (ruido blanco). La señal es generada mediante un código de Python siguiendo los lineamientos planteados en¹ y exportada a un Archivo de Excel, donde se agrega una columna para los datos temporales. La misma está formada por de 3 canales con ondas cerebrales, capaces de simular Ritmos Alfa contaminados por Ruido de Pestañeo y Ruido Aleatorio. Este tipo de señales conocidas como "Dummy" son ampliamente utilizadas para la verificación de etapas en el procesamiento de señales de EEG.

¹Fuente: <https://stackoverflow.com/questions/53071709/remove-eye-blinks-from-eeg-signal-with-ica?fbclid=IwAR3l2Cm-h6pJcS1KDNsI9mcCRQPmq8NJKBb8-fIDInPJr9mzdnfhQyLQ8Z8>. Última visita: 08-2019.

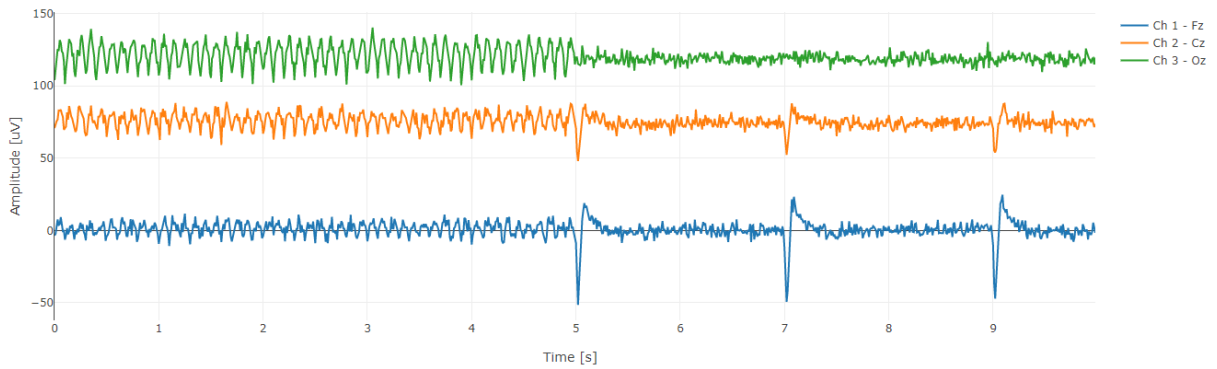


Figura 4.8: Señal Dummy de Ondas Alfa contaminada con ruido de Pestañeo y aleatorio

Señal Dummy de Ondas Cerebrales a distintas frecuencias contaminada con Pestañeo:

Similar al ejemplo anterior, la señal es generada mediante un código en Python y luego exportada a un archivo Excel. La Dummy generada contiene ondas similares a los Ritmos Alfa con frecuencias de 10, 15 y 20Hz respectivamente y se encuentran contaminadas solamente con ruido de Pestañeo, sin la presencia de ruido aleatorio(Figura 4.9).

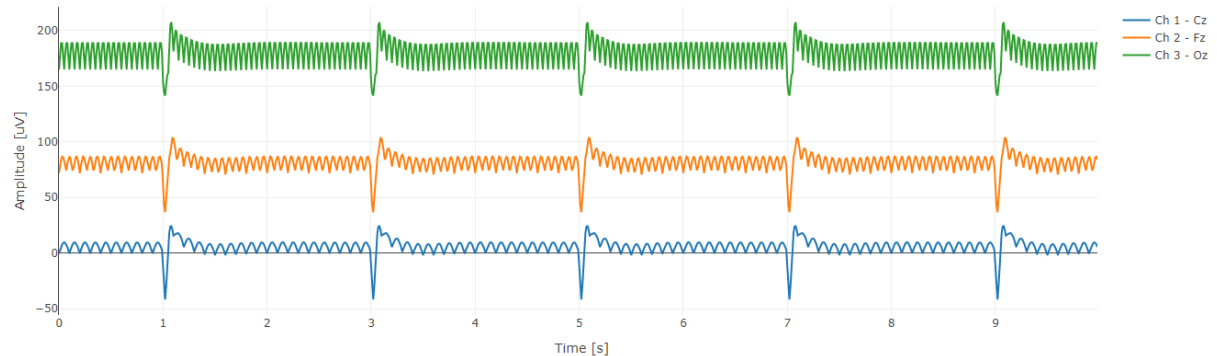


Figura 4.9: Señal Dummy de Ondas Cerebrales a distintas frecuencias con ruido de Pestañeo

4.2. Verificaciones:

4.2.1. Carga de archivos en distintos formatos:

Señal de Ejemplo Utilizada:

- Archivos con Datos Aleatorios en Distintos formatos

Desarrollo:

En esta etapa se verifica la capacidad del programa para detectar correctamente los distintos formatos de señal. Para ello se cargan señales con datos aleatorios en distintos

formatos usando la ventana de Parámetros de Carga (Figura 3.11). En la Figura 4.10 se observa que en cada ejemplo, la señal fue cargada correctamente en un DataFrame de Pandas.

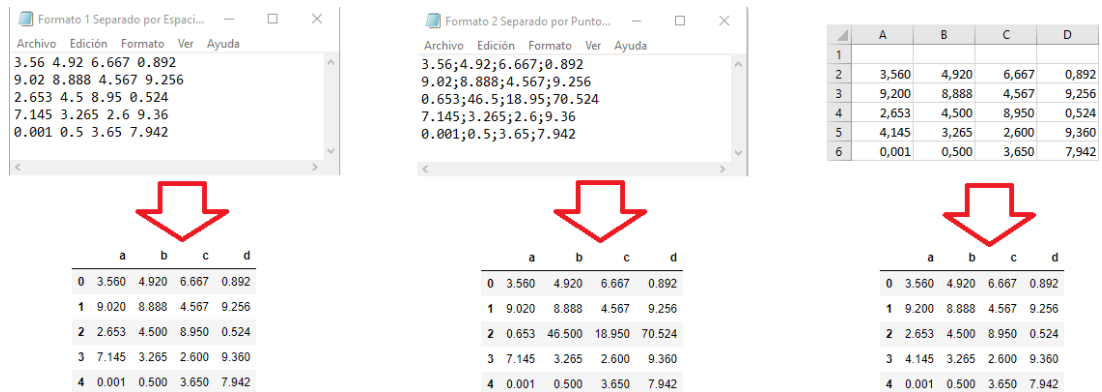


Figura 4.10: Resultados de la verificación de carga de Señal con Distintos Formatos.

4.2.2. Definición de la Tasa de Muestreo:

Señales de Ejemplo Utilizadas:

- Señales senoidales a Distintas Frecuencias

Desarrollo:

La Tasa de Muestreo puede definirse a través del ingreso manual o detectarse de forma automática mediante columnas de tipo *Time*.

En esta etapa se verifican dos operaciones: La Detección Automática de la Tasa de Muestreo y la capacidad de la herramienta para detectar varias columnas de tiempo.

Los resultados de la verificación se observan en las Figuras 4.11 y 4.12.

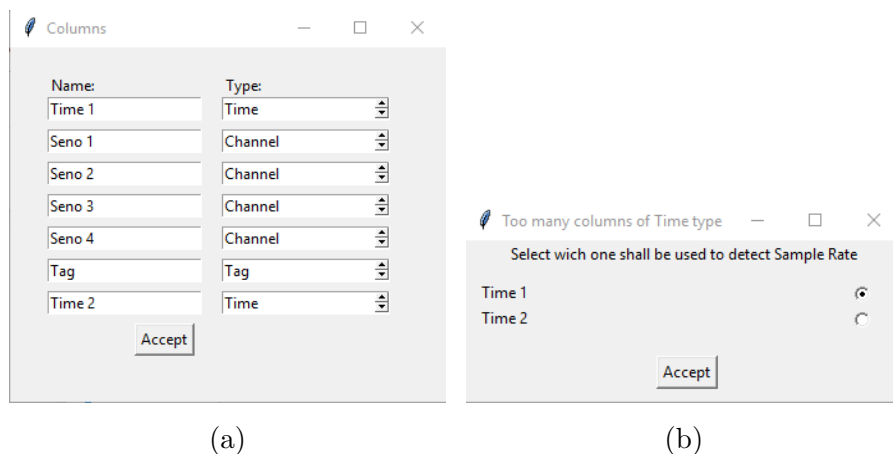


Figura 4.11: En la **Figura a** se visualizan los parámetros de carga de la señal. La misma posee dos columnas de datos temporales, denominadas Time 1 y Time 2. Al detectar ambas columnas de tiempo, se ejecuta una ventana (**Figura b**) para solicitar al usuario cuál utilizar en la definición la Tasa de Muestreo.

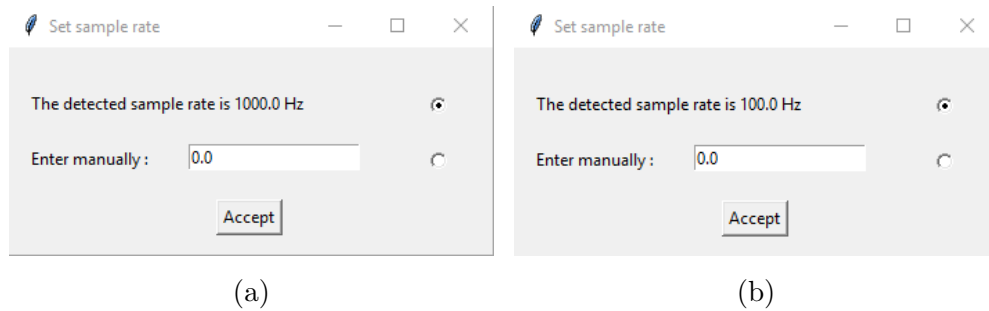


Figura 4.12: Ventanas de detección automática de Tasa de Muestreo. A la izquierda la Tasa detectada para la columna Time 1 (1000 Hz), y a la derecha la Tasa detectada para la columna Time 2 (100 Hz).

Se concluye que las Tasas de Muestreo detectadas de manera automática coinciden con los datos temporales de las columnas Time 1 y Time 2.

4.2.3. Intervalos definidos por etiquetas:

Señales de Ejemplo Utilizadas:

- Señales senoidales de frecuencia creciente

Desarrollo:

Según lo descrito en la Sección 3.2.3.2, los Intervalos pueden definirse a través de tres métodos: ingreso manual, mediante un archivo Excel y mediante una Columna con etiquetas (Tags).

El objeto de esta etapa es verificar si la herramienta define intervalos a través de columnas con etiquetas, y si permite al usuario seleccionar una de ellas en el caso que se presenten más de una columna.

Para la verificación se implementa una señal formada por señales senoidales de frecuencia creciente, de manera que al utilizar la columna de etiquetas queden definidos intervalos según su frecuencia, como puede observarse en Figura 4.13.

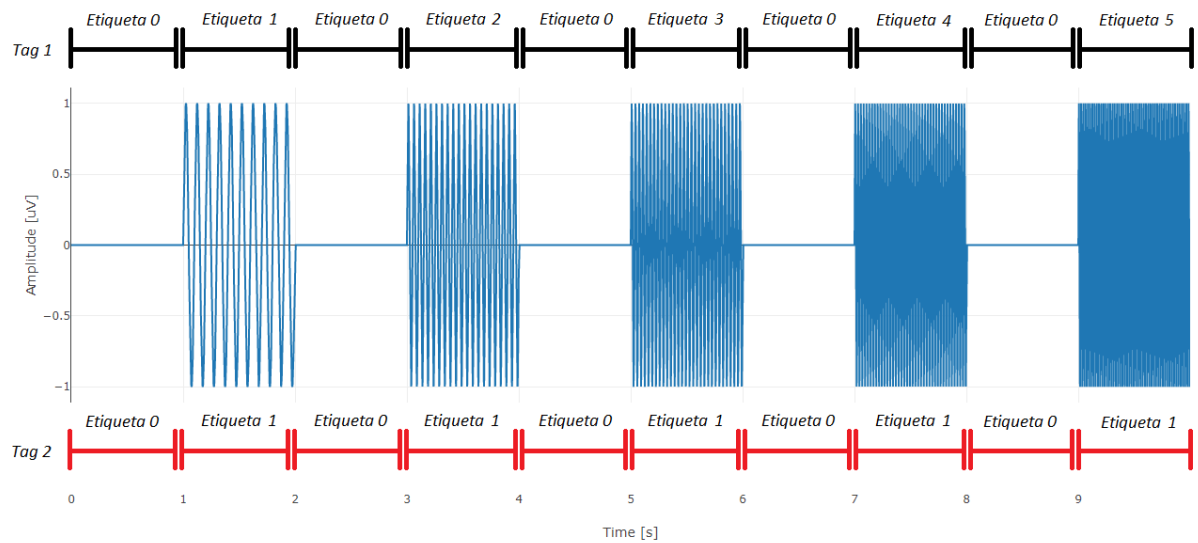
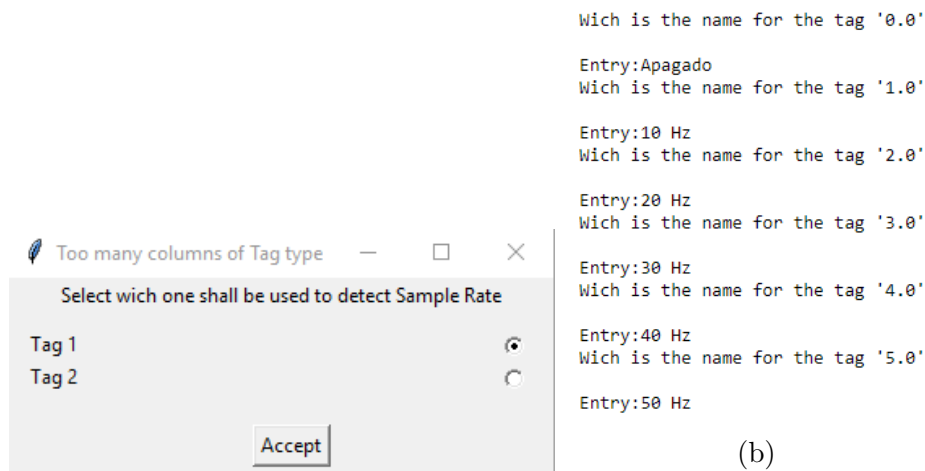


Figura 4.13: Señal de señales senoidales Escalonados, con etiquetas presentes en cada intervalo según columna de Tags (Tag 1 y Tag 2).

Una vez cargada la señal, la herramienta detecta dos columnas de etiquetas y solicita al usuario cuál utilizar para definir Intervalos (Figura 4.14a). Una vez seleccionada una de ellas (Tag 1), se solicita al usuario el ingreso de los nombres de cada intervalo, como puede observarse en la Figura 4.14.



(a)

(b)

Figura 4.14: En la **Figura a** se observa la ventana desplegada al detectar dos columnas de tipo *Tags*, con los nombres respectivos de cada columna. En la **Figura b** se representa el proceso de definición de intervalos, donde la herramienta detecta las etiquetas disponibles y solicita al usuario un nombre para cada uno de los intervalos a definir a partir de las mismas.

Para la verificación se consideran solamente las etiquetas de la columna Tags 1, relacionadas a las distintas frecuencias de la señal. Luego para verificar la definición de intervalos, se selecciona el intervalo con etiqueta 1 correspondiente a la señal con frecuencia de 10 Hz .

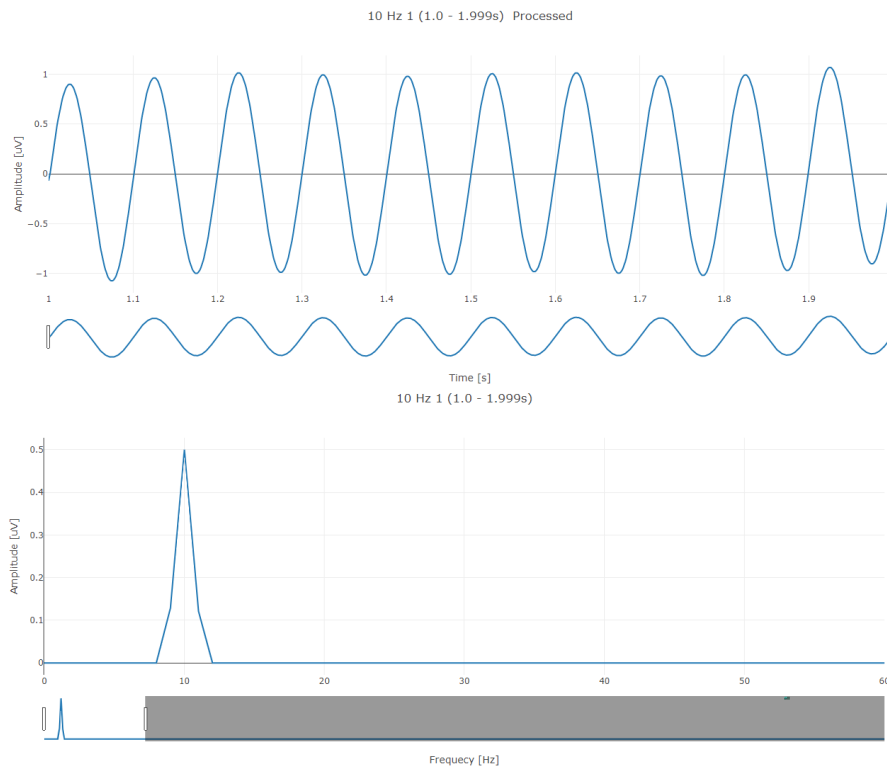


Figura 4.15: Ejemplo de Intervalo generado por etiquetas, donde se observa tanto en el espectro de tiempo como el de frecuencias, como la señal en dicho intervalo posee la frecuencia deseada de 10 Hz.

Como se observa en la Figura 4.15, el intervalo de la señal seleccionado corresponde con la frecuencia deseada, verificando de esta manera el método. Resultados equivalentes se obtuvieron para el resto de los intervalos definidos.

4.2.4. Filtrado:

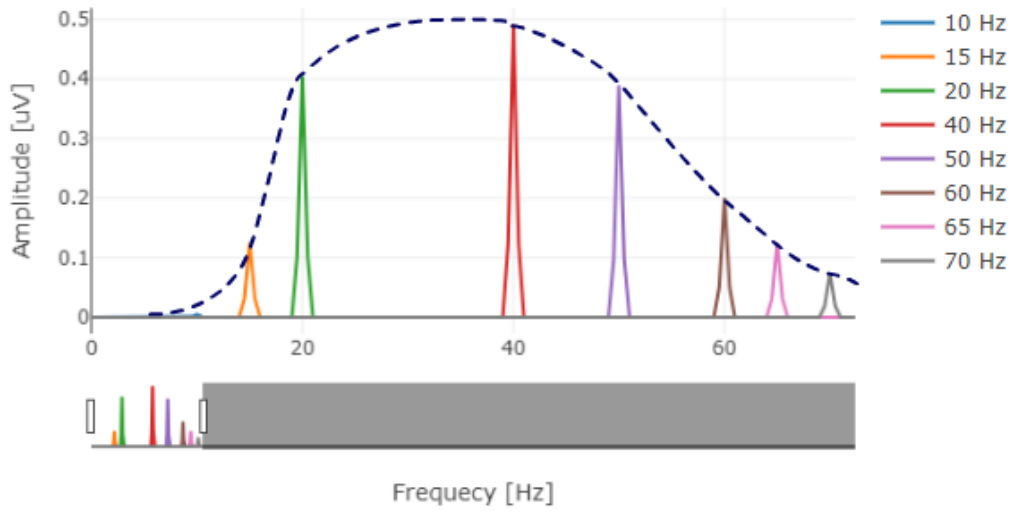
Señales de Ejemplo Utilizadas:

- Señales senoidales a distintas frecuencias

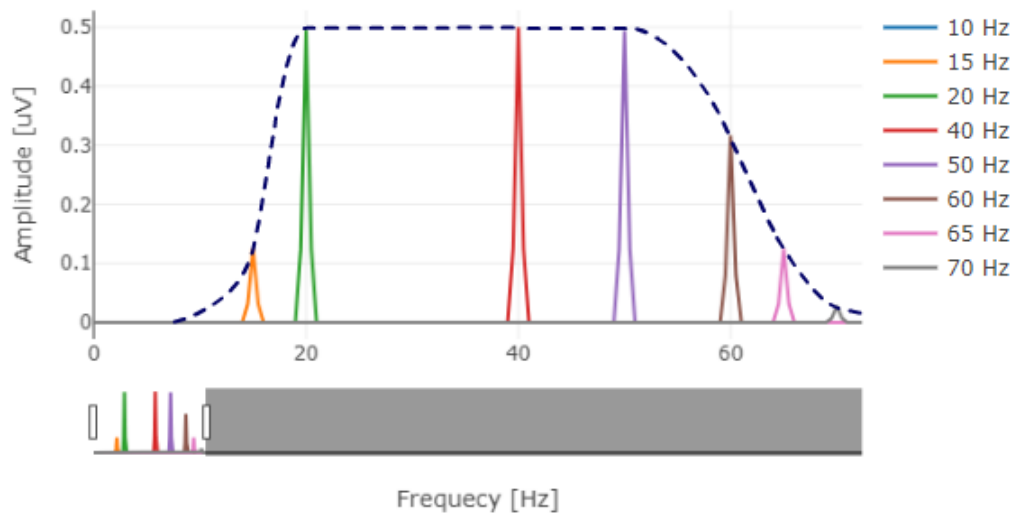
Desarrollo:

Para la verificación del método de Filtrado, se implementa una señal compuesta por ondas senoidales a distintas frecuencias y se modifican los parámetros de filtrado (Figura 3.19), verificando luego si la Respuesta en Frecuencia de cada Filtro aplicado es la esperada.

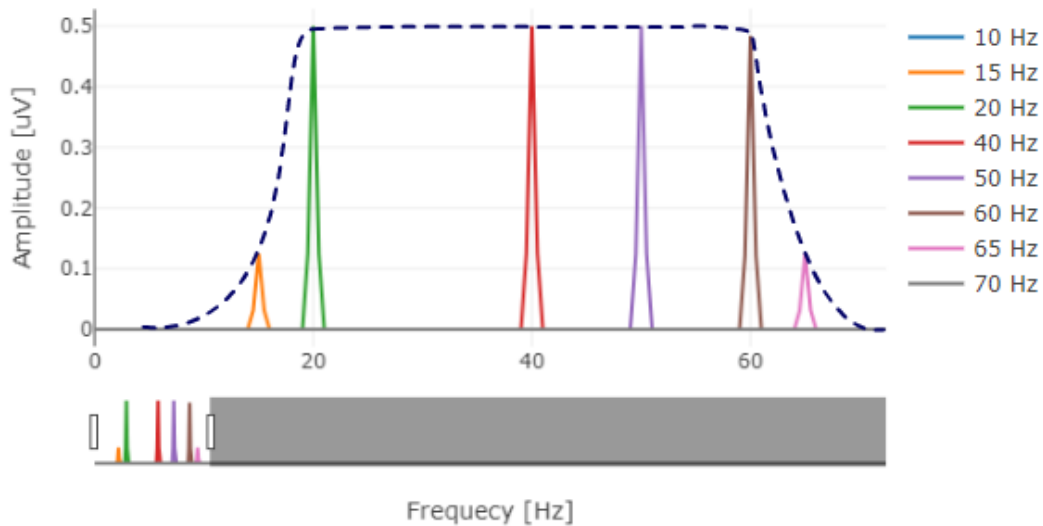
A continuación se visualizan los Espectros de Frecuencia resultantes en cada instancia de filtrado, junto con una aproximación de la Respuesta en Frecuencia de cada Filtro en línea punteada azul:



(a) Filtro Butterworth de Orden 2 entre 15 y 65 Hz, con Notch en 2 Hz.

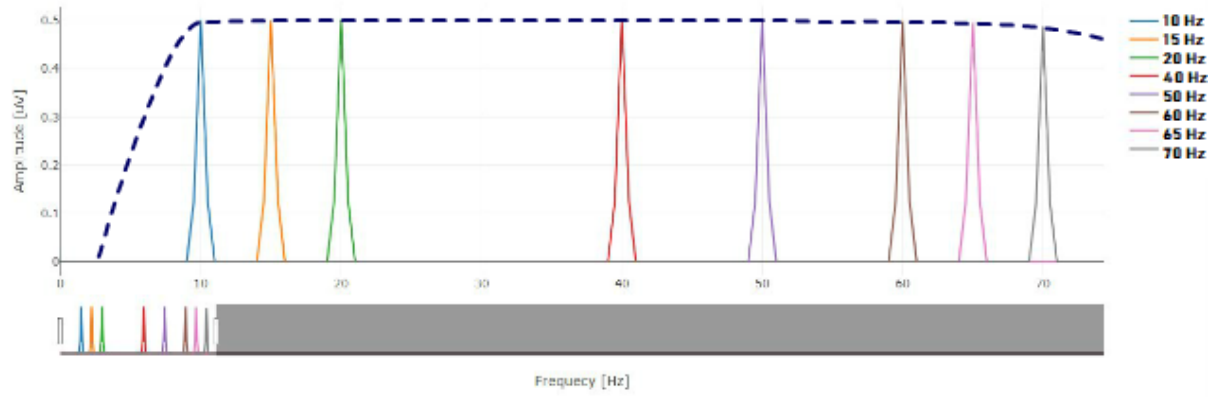


(b) Filtro Butterworth de Orden 5 entre 15 y 65 Hz, con Notch en 2 Hz.

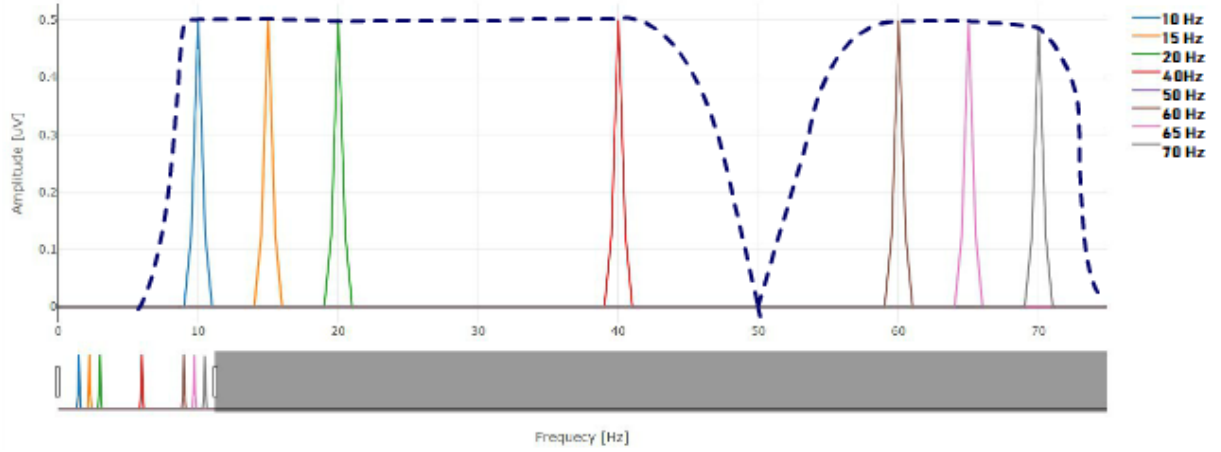


(c) Filtro Butterworth de Orden 15 entre 15 y 65 Hz, con Notch en 2 Hz.

Figura 4.16: Comparación del Espectro de Frecuencia al variar el Orden en Filtros Butterworth.

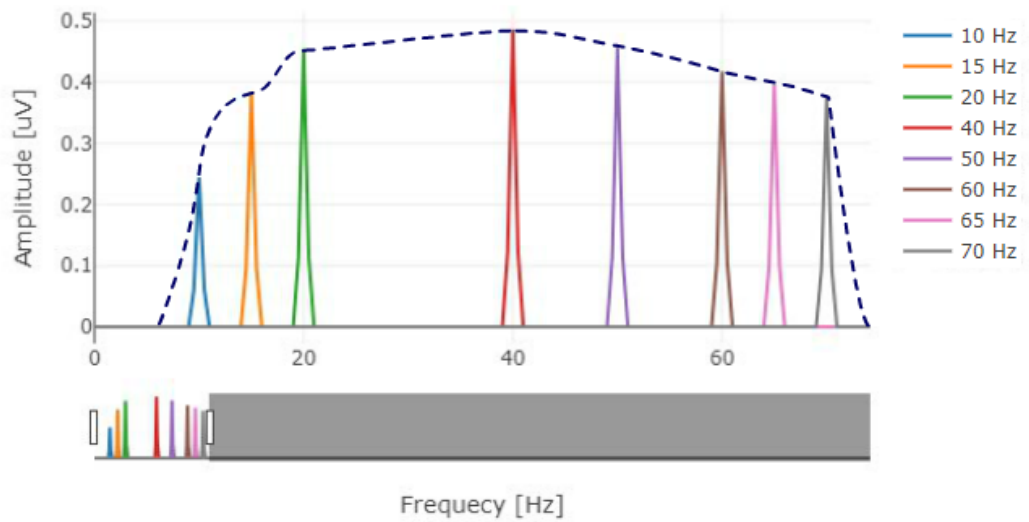


(a) Filtro Butterworth de Orden 5 entre 5 y 100 Hz, sin Filtro Notch.

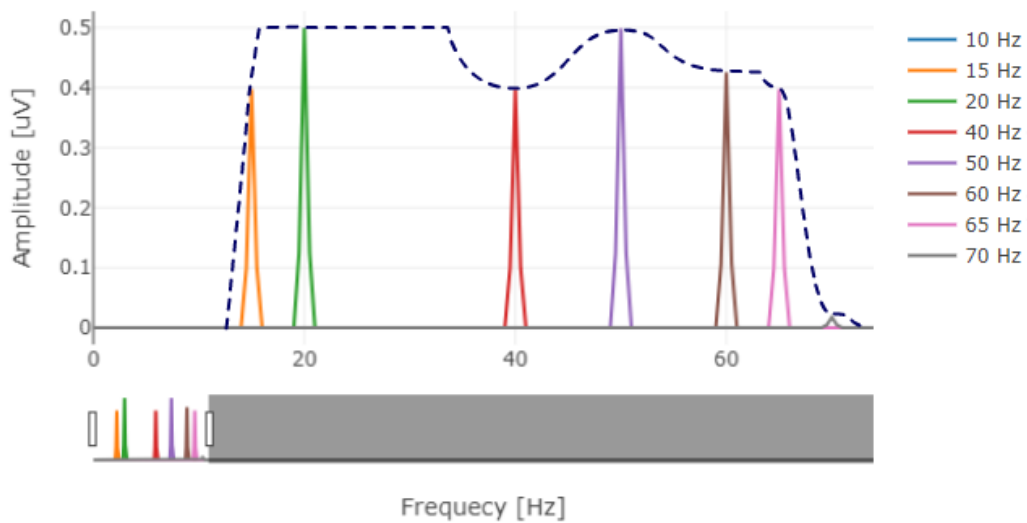


(b) Filtro Butterworth de Orden 5 entre 5 y 100 Hz, con Notch en 50 Hz.

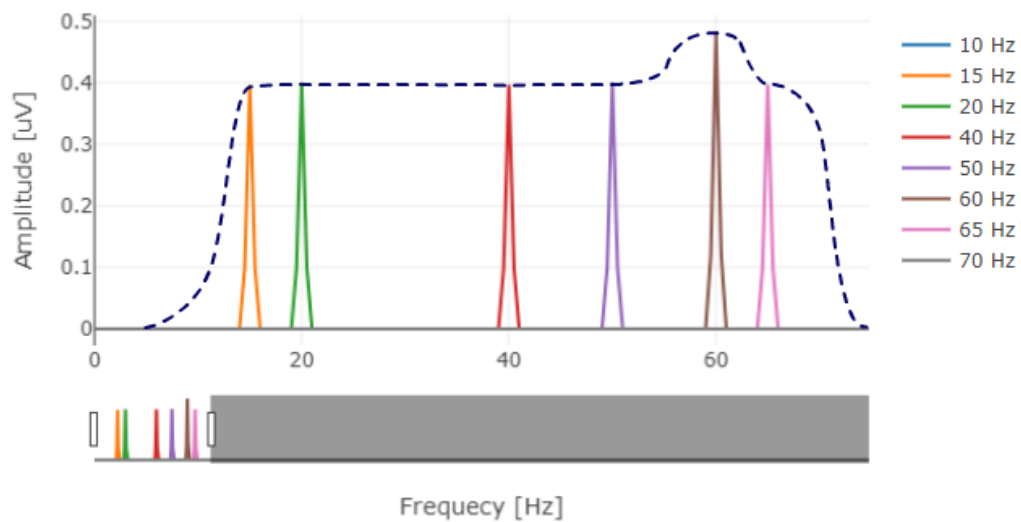
Figura 4.17: Comparación del efecto de un Filtro Notch aplicado a 50 Hz.



(a) Filtro Chebyshev de Orden 1 entre 15 y 65 Hz, con ripple de 0,5.

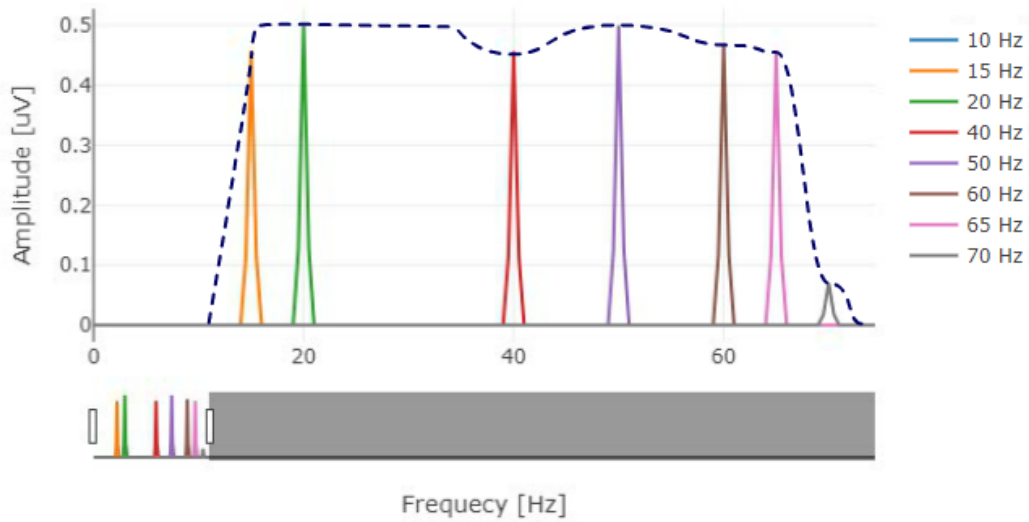


(b) Filtro Chebyshev de Orden 5 entre 15 y 65 Hz, con ripple de 0,5.

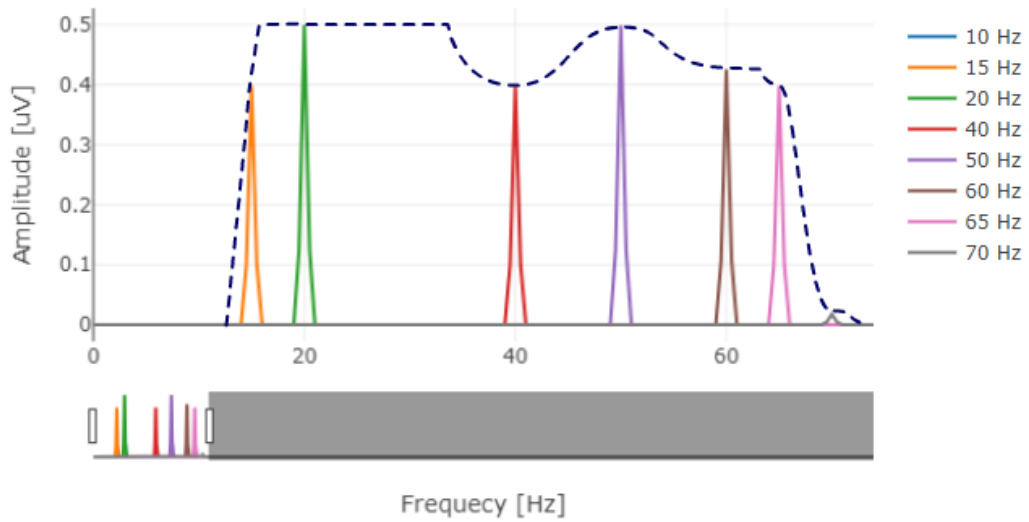


(c) Filtro Chebyshev de Orden 15 entre 15 y 65 Hz, con ripple de 0,5.

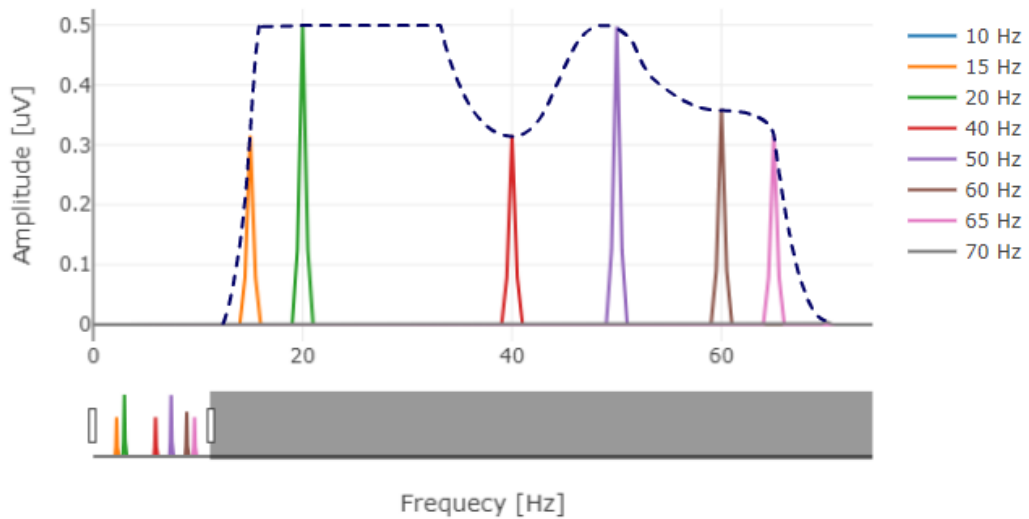
Figura 4.18: Comparación del Espectro de Frecuencias al variar el Orden de los Filtros Chebyshev. En comparación al filtro Butterworth de la Figura 4.16, el efecto del ripple logra una bajada más directa en la frecuencia de corte, aunque introduce ruido en la señal.



(a) Filtro Chebyshev de Orden 5 entre 15 y 65 Hz, con ripple de 0,2.



(b) Filtro Chebyshev de Orden 5 entre 15 y 65 Hz, con ripple de 0,5.



(c) Filtro Chebyshev de Orden 5 entre 15 y 65 Hz, con ripple de 1.

Figura 4.19: Comparación del Espectro de Frecuencia al variar el ripple en filtros Chebyshev. Se observa como un mayor ripple logra los mismos efectos con un menor orden, pero afectando algunas frecuencias.

Aunque la envolvente generada (en línea azul discontinua) para el análisis de la respuesta en frecuencia sólo incluye ondas senoidales en frecuencias específicas de interés y no un barrido completo del espectro de frecuencias, son evidentes muchas de las características de filtrado descritas previamente en la Sección 2.1, entre ellas:

- El orden de un filtro mejora la definición de la frecuencia de corte, dándole una bajada mucho más pronunciada.
- Los filtros tipo Chebyshev, a diferencia de los tipo Butterworth, generan un determinado nivel de riple sobre el espectro de frecuencias de la señal filtrada, contaminándola. Sin embargo, a mayor nivel de riple, se logra una bajada más pronunciada y definida con el mismo o menor orden.
- Los filtros Notch permiten eliminar efectivamente información de la señal de una frecuencia en particular, sin afectar componentes de frecuencia que se encuentren a una distancia mayor a 1 Hz de la frecuencia eliminada.

4.2.5. Submuestreo y Nyquist:

Señales de Ejemplo Utilizadas:

- Señales senoidales a Distintas Frecuencias

Desarrollo:

En esta etapa se evalúa la capacidad de la herramienta para reducir de manera correcta la cantidad de muestras de una señal mediante la función de submuestreo. Además, se verifica según el Teorema de Nyquist, como las señales se deforman a medida que disminuye la cantidad de muestras por instante de tiempo. Para tal verificación se implementa una Señal formada por señales senoidales a Distintas Frecuencias, de manera que al aplicar submuestreo, aquellas ondas cuyas frecuencias se encuentren por encima de la mitad de la frecuencia de submuestreo no pueden ser analizadas. Al momento de realizar la verificación, se debe tener en cuenta que:

- Si una onda senoidal (representada en el Espectro de Frecuencia como una componente) deja de satisfacer el Teorema Nyquist, no desaparece sino que se deforma hasta el punto de no poder ser comprendida, como se visualiza en la Figura 4.22.
- Si una onda senoidal satisface el Teorema de Nyquist pero su frecuencia se encuentra cercana al límite de pérdida, sufre una cierta deformación y aunque pueda no parecer una onda senoidal en tiempo, mantiene sus propiedades en frecuencia como puede visualizarse en la Figura 4.21.

El submuestreo aplicado a la señal durante esta etapa es de 100 Hz, por lo que el límite de frecuencias es de 50 Hz.

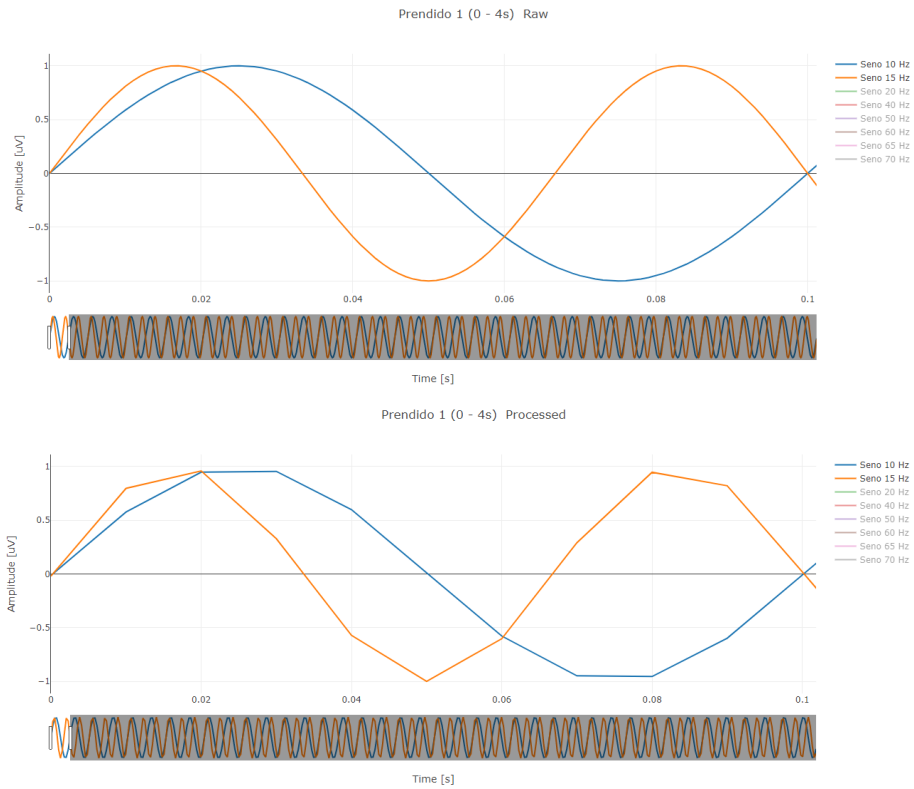


Figura 4.20: Señales senoidales de Frecuencias de 10 y 15 Hz, sin submuestreo (arriba) y con submuestreo a 100 Hz (abajo).

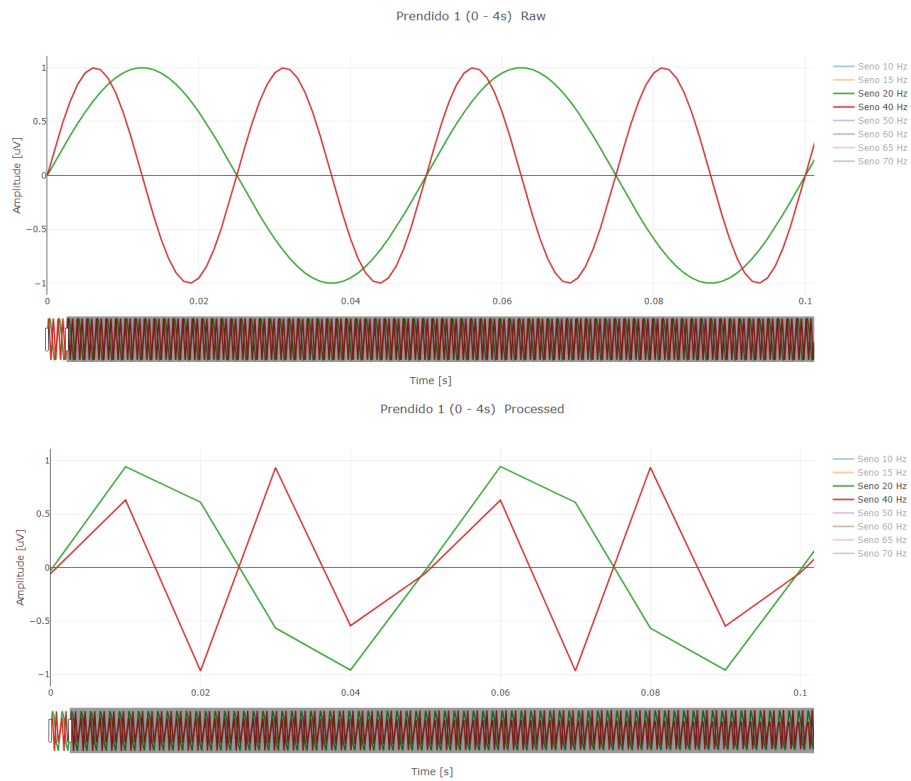


Figura 4.21: Señales senoidales de Frecuencias de 20 y 40 Hz, sin submuestreo (arriba) y con submuestreo a 100 Hz (abajo). Se observa como al acercarse al límite de frecuencias (50 Hz) las señales se empiezan a deformar.

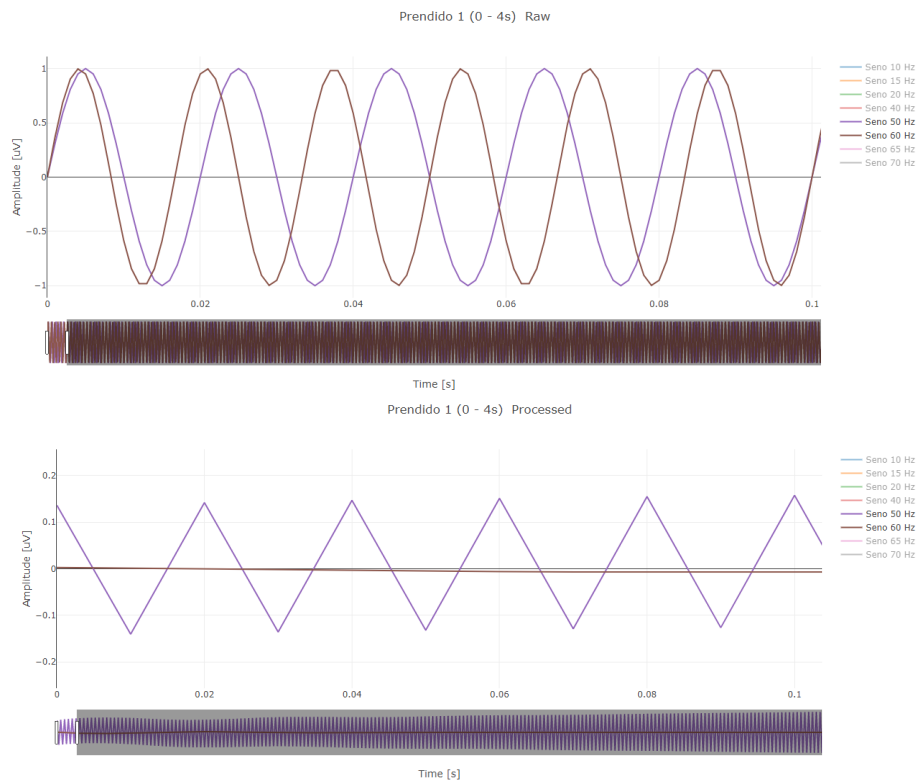


Figura 4.22: Señales senoidales de Frecuencias de 50 y 60 Hz, sin submuestreo (arriba) y con submuestreo a 100 Hz (abajo). Se observa como la señal senoidal de 50 Hz apenas conserva su forma, mientras que el de 60 Hz no posee una forma capaz de ser analizada.

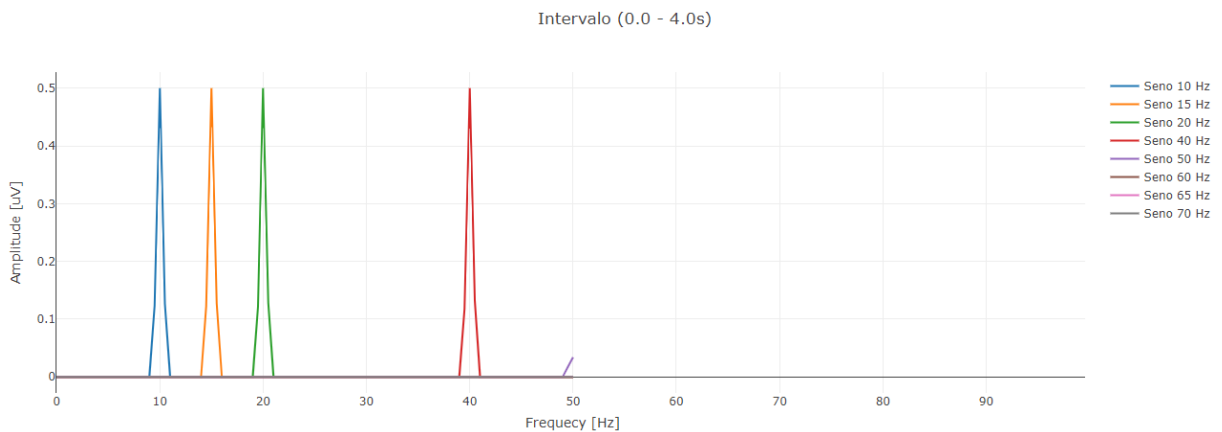


Figura 4.23: Espectro de frecuencia de señales senoidales con submuestreo a 100 Hz. Se observa como las señales a frecuencias cercanas a los 50 Hz mantienen su espectro aunque se encuentran disminuidas.

4.2.6. Descomposición ICA:

Señales de Ejemplo Utilizadas:

- Dummy de ondas alfa, contaminada con pestañeo y ruido aleatorio

Desarrollo:

La señal utilizada para validar el método de Descomposición o Análisis en Componentes Independientes (ICA) es una Dummy de ondas alfa contaminada con pestañeo y ruido aleatorio (Figura 4.24). El objeto de la verificación es separar estos tres canales en componentes independientes para poder eliminar luego aquellos que son fuente de contaminación y obtener como resultado una señal de ondas alfa sin ruido.

En la Figura 4.25 se observa un gráfico de los componentes independientes separados. Entre ellos se decide eliminar el tercero (verde) debido a que contiene la mayor proporción de ruido. En dicha imagen se aprecia que la primer componente (azul) también posee una proporción de ruido considerable, pero se decide conservarla para que la señal resultante no pierda información relevante.

En la Figura 4.26 se observa el gráfico de dos de los canales después de procesarlos por el método de ICA.

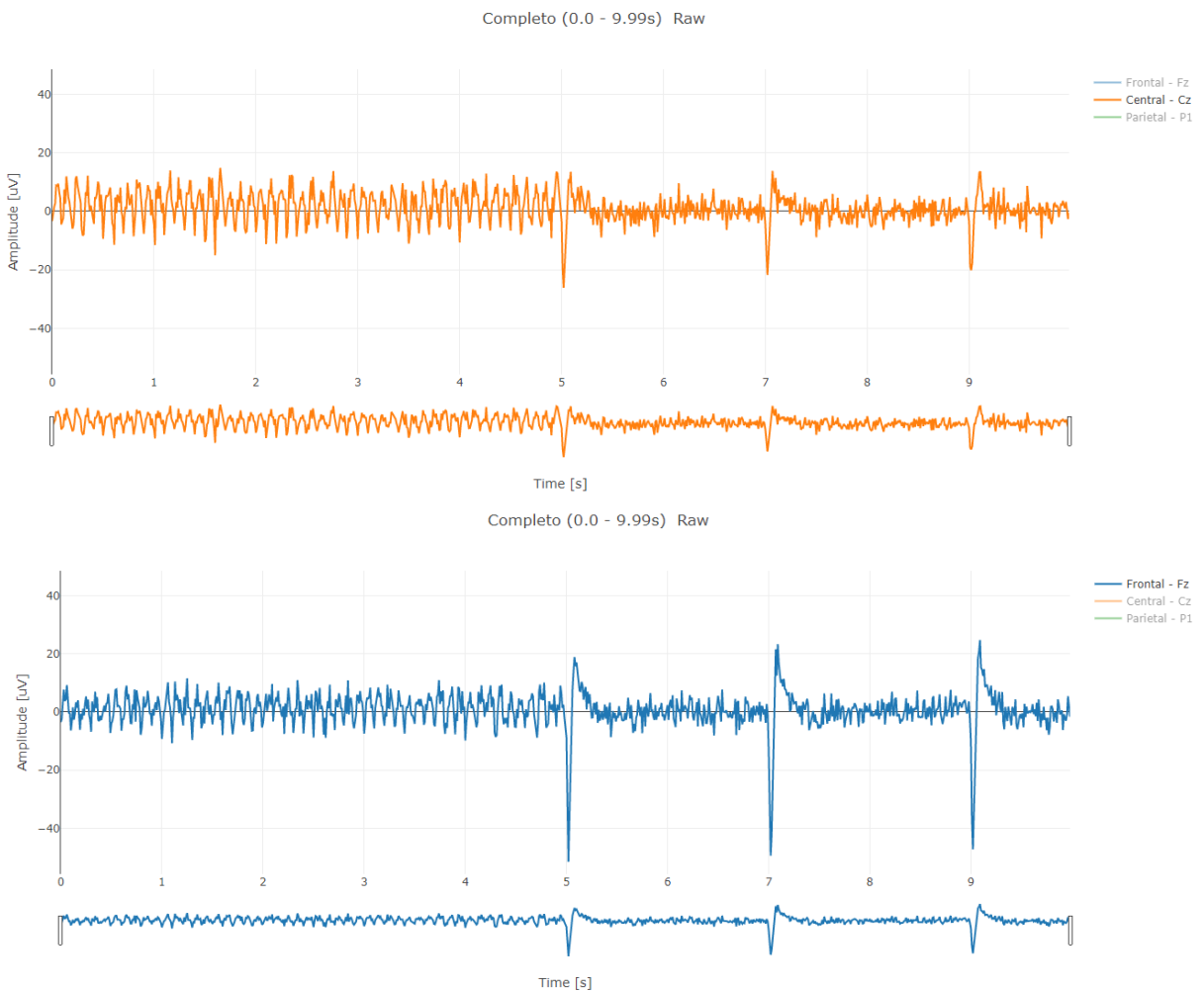


Figura 4.24: Señales originales sin aplicación el método ICA. Estas señales no poseen ningún acondicionamiento previo.

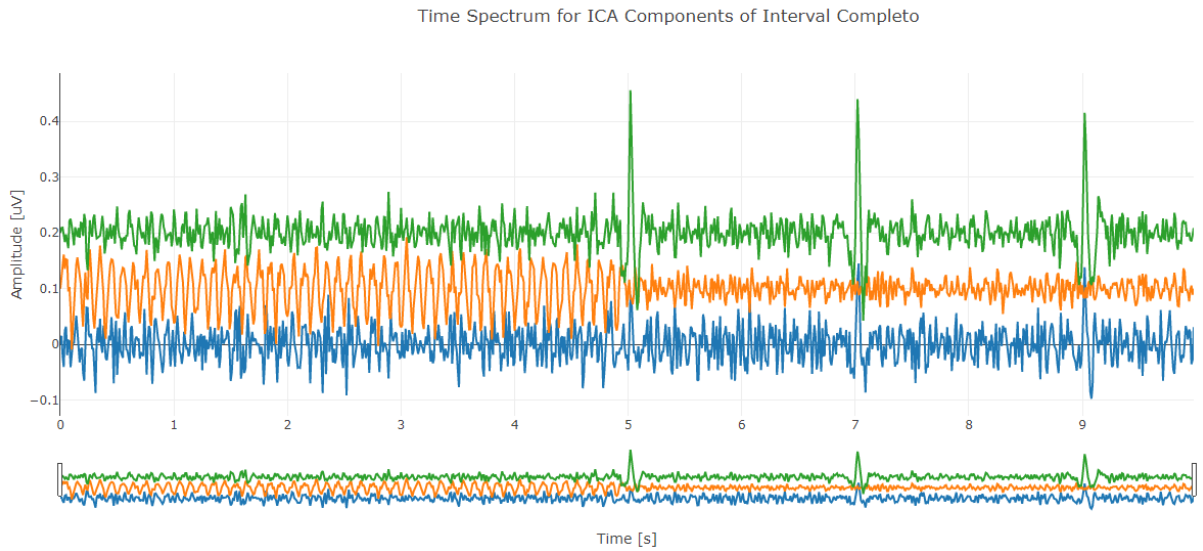


Figura 4.25: Componentes independientes de la señal obtenidos mediante el método ICA. De los mismos se decidió eliminar el tercero (en verde) y reconstruir la señal con los dos restantes.

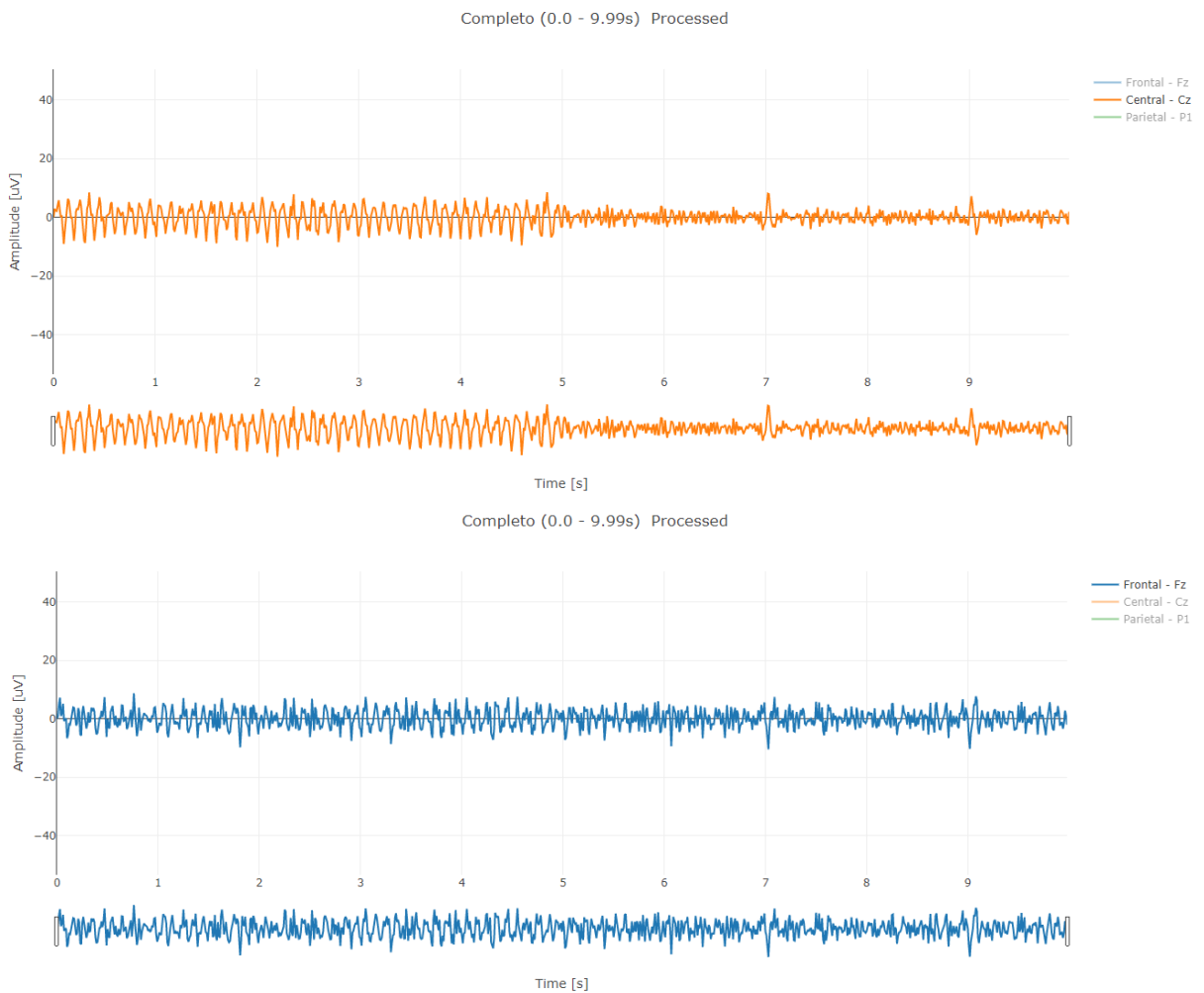


Figura 4.26: Señales reconstruidas luego de ser procesadas según método ICA.

Aclarando que ICA es un método basado en la probabilidad, es posible que en una

aplicación del mismo los componentes no se separen correctamente. Esto no invalida el método, sino que el mismo debe reiterarse varias veces hasta obtener una separación que resulte satisfactoria. La decisión de cuáles componentes eliminar se realiza a través de un análisis cualitativo de los componentes generados, por lo que este método requiere de un usuario experimentado. Además, el mismo puede ser automatizado mediante la implementación de algoritmos de identificación de patrones adicionales.

4.2.7. CAR:

Señales de Ejemplo Utilizadas:

- Dummy de Ondas Cerebrales a distintas frecuencias contaminada con Pestañeo
- Ondas Senoidales Cuadradas

Desarrollo:

Para validar matemáticamente el método CAR se utilizan señales cuadradas. De esta manera, con la aplicación del método la señal resultante debería presentar en cada canal su amplitud original menos el promedio de las tres amplitudes. En la Figura 4.27, se visualizan los efectos de haber aplicado el método. Teniendo en cuenta que el promedio de las amplitudes resulta ser 3, la onda de amplitud 1 se invierte y se reduce a -2 y la onda de amplitud 3 se anula, mientras que la onda de amplitud 5 se reduce a 2.

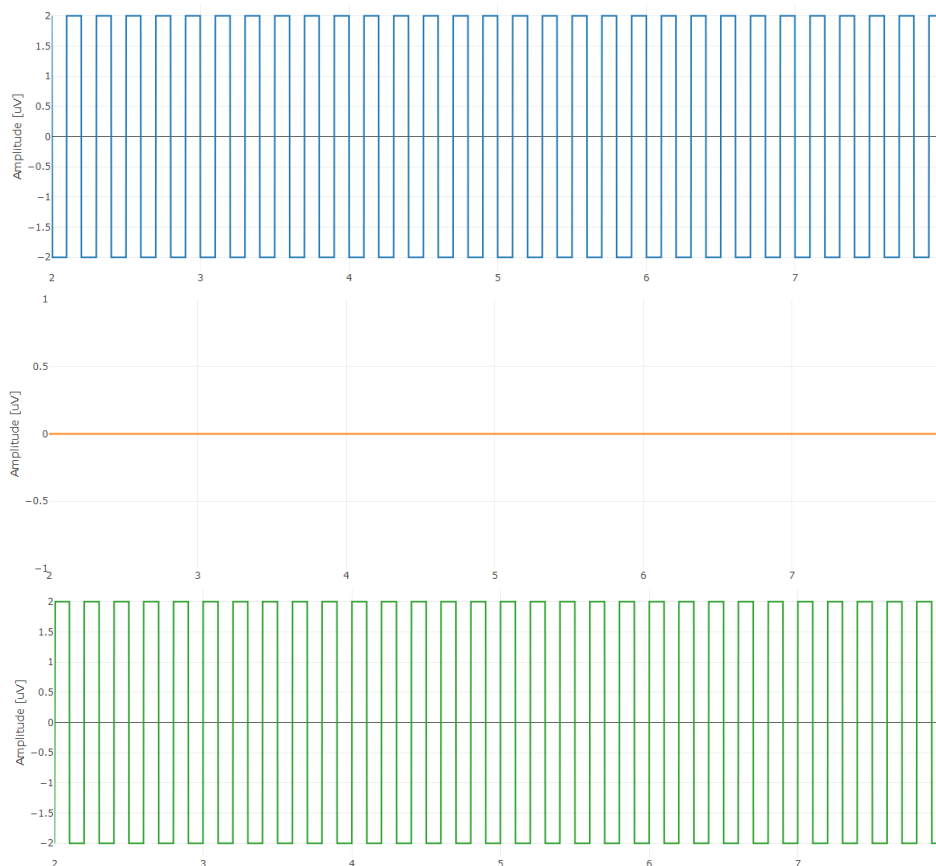


Figura 4.27: Señal de Ondas Cuadradas resultantes, obtenidas mediante aplicación del método CAR.

Como complemento de verificación se compara el método CAR con el método ICA, ambos utilizados para la eliminación de Ruido. Para ello se utiliza una Señal Dummy de Ondas Cerebrales a distintas frecuencias contaminadas con Pestañeo (Figura 4.28 a).

Emplear el método CAR para procesar la señal resulta efectivo en la eliminación del ruido de pestañeo, debido a que éste es común a las tres señales. Sin embargo, presenta la desventaja de que, al restar el promedio de todos los canales, una parte de la información de cada canal se transmite a cada uno de los otros, y se obtiene como resultado canales con información mixta. En la Figura 4.28, estos efectos pueden observarse tanto en el espectro de tiempo, donde desaparecen las formas que simulaban el ruido de pestañeo pero cada canal ya no presenta una onda senoidal fundamental; como en el espectro de frecuencia, donde desaparecen las componentes de menor frecuencia correspondientes al ruido pero aparecen en cada canal frecuencias que corresponden a otros canales.

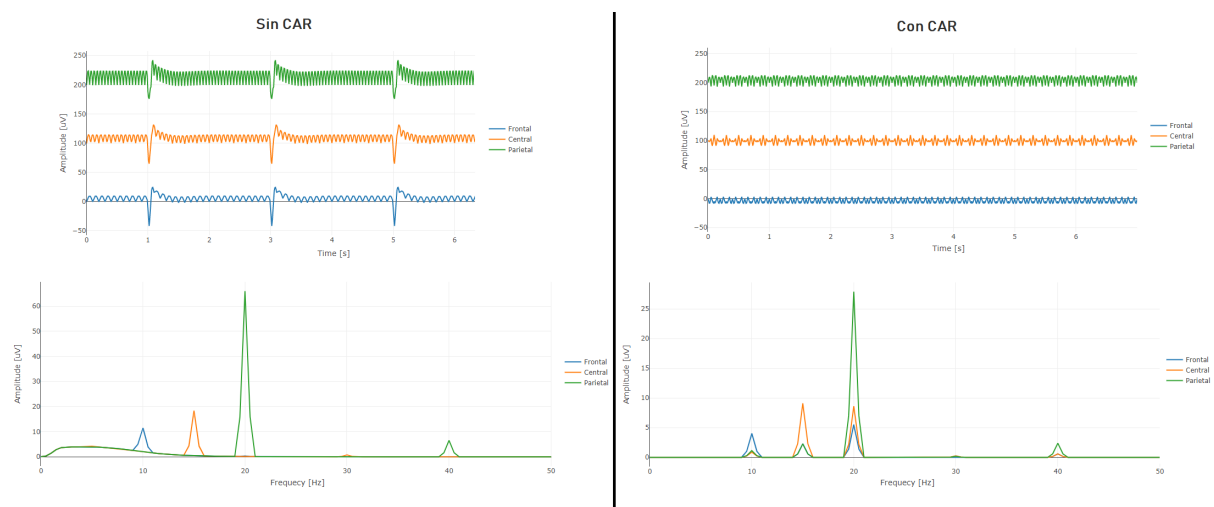


Figura 4.28: Comparación de la señal ejemplo de Dummy de Ondas Cerebrales y Pestañeo previo a aplicar el método CAR y luego de haberlo aplicado.

Por lo tanto, el método CAR es útil para la eliminación de ruido cuando el mismo es común a todos los canales, pero además cuando la información relevante de cada canal es notable respecto al resto, para que no se pierda durante el proceso.

ICA en cambio, requiere un mayor conocimiento de la señal y de los artefactos que se desean eliminar, siendo mayor la cantidad de casos donde puede ser aplicado efectivamente. ICA requiere un costo computacional mucho mayor en comparación al esfuerzo requerido por el método CAR. Esto tiene consideraciones muy importantes a la otra de procesar señales de manera *online*, debido a que un mayor esfuerzo computacional conlleva un mayor tiempo de procesamiento.

En la Sección 5.1 y la Sección 5.3, se compara el efecto de los métodos ICA y CAR al ser aplicados a una misma señal.

4.2.8. Transformada en Frecuencia: FFT y Welch:

Señales de Ejemplo Utilizadas:

- Señales senoidales a Distintas Frecuencias

Desarrollo:

En esta etapa se verifica si los métodos FFT y Welch detectan de manera correcta las frecuencias involucradas en una Señal de Ejemplo. Para la verificación se utiliza una Señal formada por señales senoidales a distintas frecuencias, con amplitud 1.

El espectro de frecuencias de las señales senoidales debe resultar en un pico en la frecuencia fundamental de las mismas, con la mitad de su amplitud. Teniendo en cuenta que la señal implementada posee amplitud 1 y que el método considera valores absolutos de transformada, se debe verificar que ambos métodos obtengan un Espectro en Frecuencia con un pico de 0,5 sobre la frecuencia de cada onda.

En la Figura 4.29 se observa como la función FFT de la librería NumPy detecta las frecuencias involucradas con una energía igual a 0,5 en cada una de las mismas, validando de esta manera el método implementado.

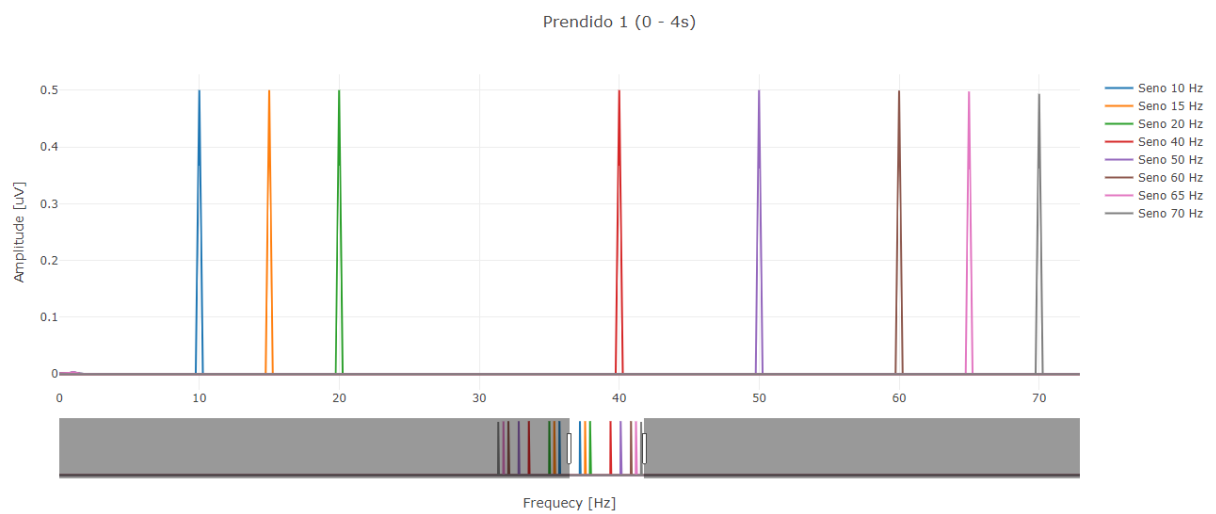


Figura 4.29: Espectro en Frecuencias obtenido mediante método FFT de una señal compuesta por señales senoidales de diferentes frecuencias y amplitud 1.

Por otra parte, como puede observarse en la Figura 4.30, el algoritmo Welch de la librería MNE detecta las frecuencias involucradas en la señal con una energía igual a 0,5 en cada una de las mismas, validando de esta manera el método implementado.

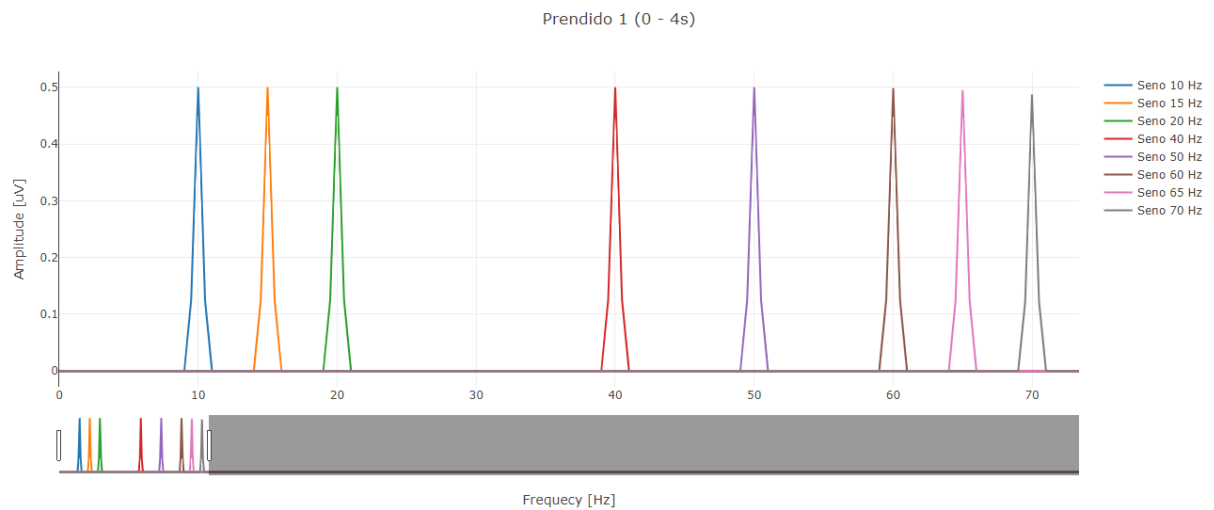


Figura 4.30: Espectro en Frecuencias obtenido mediante método Welch de una señal compuesta por señales senoidales a diferentes frecuencias y amplitud 1 .

4.2.9. Espectrograma:

Señales de Ejemplo Utilizadas:

- Señales senoidales de frecuencia creciente

Desarrollo:

Para validar la generación de espectrogramas se implementa una señal formada por ondas senoidales de frecuencia creciente. Los resultados obtenidos durante la verificación se pueden observar en la siguiente Figura 4.31. Se observa el cambio de la Frecuencia Principal (en amarillo) cada 2 segundos, separados por intervalos de energía nula de 1 segundo. Estos resultados pueden compararse con la descripción de esta señal en la Figura 4.5a.

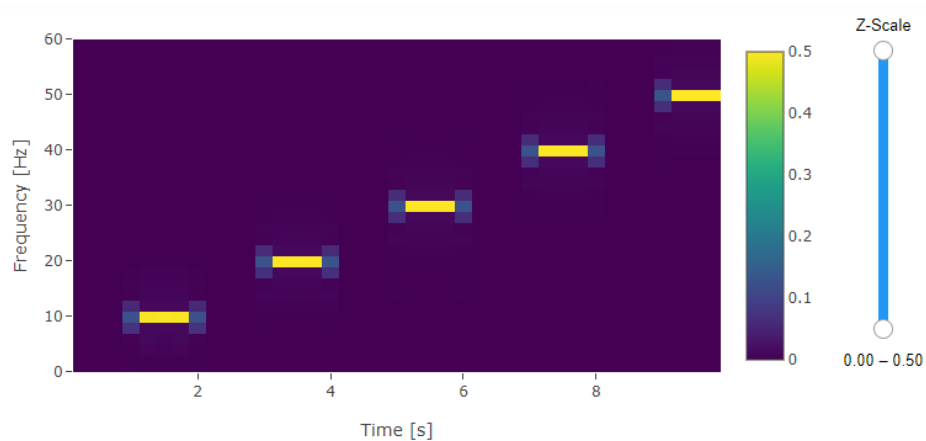


Figura 4.31: Espectrograma de una Señal formada por señales senoidales de frecuencia creciente.

4.2.10. Mapas Topográficos:

Señales de Ejemplo Utilizadas:

- Señales senoidales Desfasados y con Amplitud variada

Desarrollo:

Para generar los mapas topográficos, la herramienta implementa una función específica de la librería MNE denominada (`plot_topo`). La misma permite generar el mapa topográfico con los valores de la señal en un instante de tiempo, o el promedio de valores durante un intervalo.

Durante el desarrollo de la herramienta se decide generar mapas topográficos a través del promedio de valores en un intervalo. En cambio durante esta etapa, se verifica el correcto funcionamiento de ambos métodos. Para esto se utiliza la señal formada por ondas senoidales desfasadas a 180° . De esta manera, eligiendo correctamente el instante de tiempo donde la señal principal se encuentra en un valle y las otras dos señales en un pico, se puede apreciar claramente la diferencia de energía entre los canales.

A su vez, implementando el promedio de las señales se deberían obtener valores nulos (o casi nulos), ya que al promediar la señal los valores positivos se cancelan con los negativos.

En primer lugar como puede observarse en la Figura 4.32, se deben seleccionar las posiciones para cada uno de los canales. Las posiciones seleccionadas fueron en la zona central de la cabeza según el Sistema 10-20 (Cz, C5 y C6).

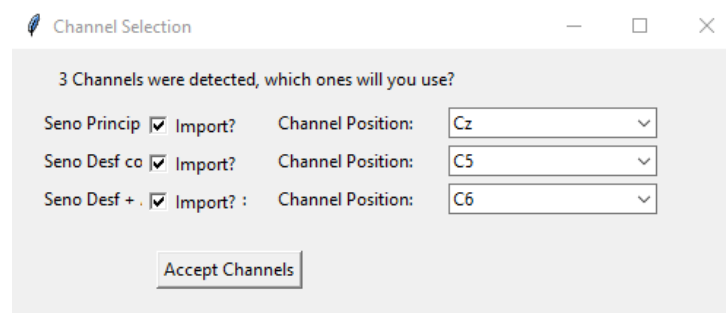


Figura 4.32: Posicionamiento de los canales utilizados: la señal senoidal Principal en Cz (central); la señal senoidal Desfasado en C5 (a la izquierda) y la señal senoidal Desfasado y amplificado en C6 (a la derecha).

En la Figura 4.33 se comparan los valores de la señal en el instante de tiempo igual a 4,675 segundos junto con el mapa topográfico en dicho instante. Se observa como los mapas generados coinciden con los resultados esperados, donde: la onda senoidal principal, ubicado en el centro, se colorea azul suave y posee un valor de -1, la onda senoidal desfasada se colorea de rojo suave y posee un valor de 1 mientras que la onda senoidal desfasada y amplificada se colorea de rojo intenso debido a que tiene valor igual a 5 en ese instante de tiempo.

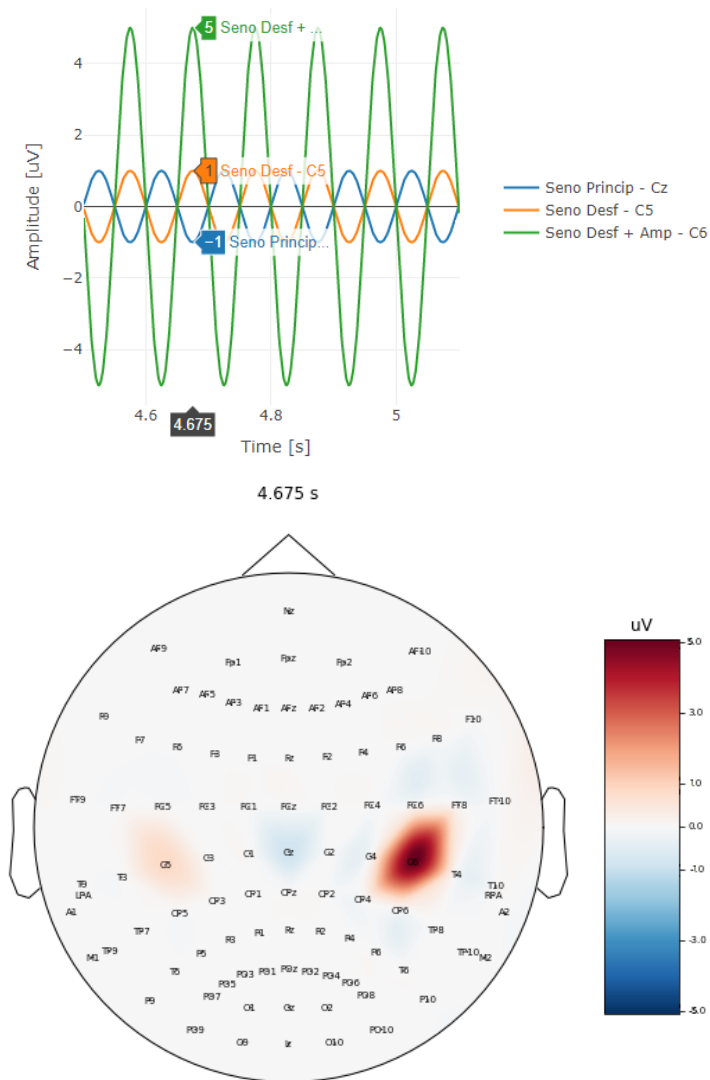


Figura 4.33: Comparación de valores de la Señal en el instante 4,675 segundos, y el mapa topográfico generado en dicho instante. Se observan los colores opuestos entre las señales de la posición central e izquierda, ya que se encuentran desfasados, y la mayor intensidad en la posición de la derecha, correspondiente a la señal de mayor amplitud.

En la Figura 4.34, se muestra un mapa topográfico generado al realizar un promediado de valores durante un intervalo. El instante de tiempo medio está ubicado en los 6.5 segundos, con 2 segundos de promediado.

Para estas 3 señales los resultados esperados luego del promediado son valores nulos debido a que se cancelan los valores positivos con los negativos. En el gráfico no se obtienen valores completamente nulos debido a que los filtros aplicados distorsionan la señal en los extremos de la misma, modificando levemente el valor obtenido en el promedio de las señales. Los valores obtenidos pueden considerarse prácticamente nulos ya que los mismos se encuentren a rangos de escalas despreciables.

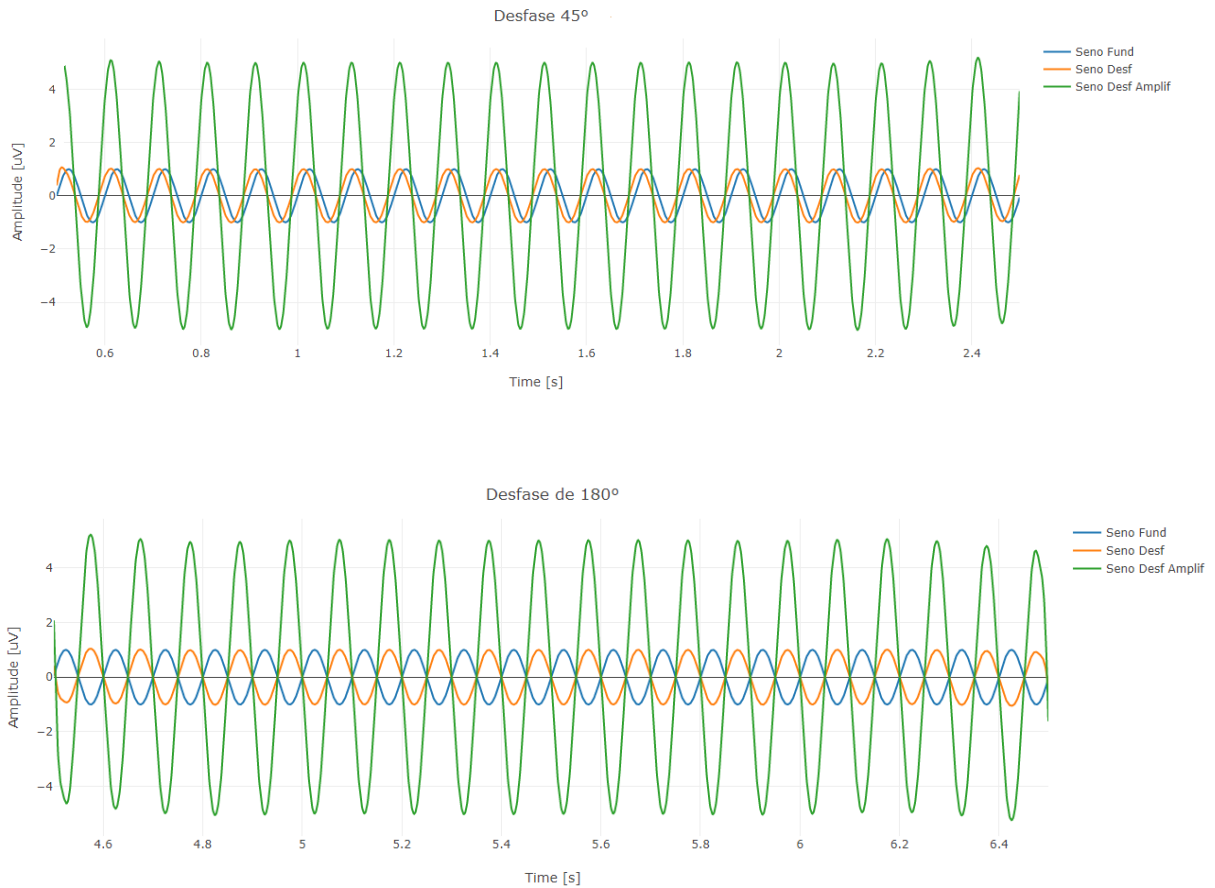


Figura 4.35: Dos de los ejemplos estudiados durante la verificación de Correlación con desfases de 45° y 180° respectivamente. Se observa como las onda naranja y verde se encuentran en fase, ambas desfasadas de la azul, aunque la verde tiene una amplitud 5 veces mayor.

	Sin Defase 1	Desfase 45° 1	Desfase 90° 1	Desfase de 180° 1	Desfase de 270° 1	Desfase de 315° 1
Ch1 vs Ch1	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000
Ch1 vs Ch2	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch1 vs Ch3	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch2 vs Ch1	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch2 vs Ch2	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000
Ch2 vs Ch3	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000
Ch3 vs Ch1	0.999261	0.68674	-0.000409	-0.997878	-0.000068	0.675345
Ch3 vs Ch2	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000
Ch3 vs Ch3	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000

Figura 4.36: Valores obtenidos durante la verificación del método de Correlación para distintos desfases de la señal. En esta tabla, Ch1 es la señal senoidal principal, Ch2 es la señal senoidal desfasada de la principal 0° , 45° , 90° , 180° , 270° y 315° por cada columna de la tabla, y Ch3 es la señal senoidal con un desfase idéntico a Ch2 pero amplificado 5 veces.

Como resultado se obtiene que la correlación entre Ch2 y Ch3 resulta siempre 1 ya que se encuentran en fase, mientras que la correlación entre estas y Ch1 varía según el desfase, siendo notable el resultado -1 para el desfase de 180° .

4.2.12. Coherencia:

Señales de Ejemplo Utilizadas:

- Señales senoidales a Distintas Frecuencias

Desarrollo:

Para verificar la Coherencia se utilizan señales senoidales a distintas frecuencias. Al analizar la similitud entre estas señales, el valor de Coherencia deberá ser mayor entre ondas de frecuencias armónicas. En la Figura 4.38 se observan, como ejemplo notable, los resultados obtenidos al estudiar la coherencia de cada una de las ondas respecto a la onda de 10 Hz. El valor de Coherencia 1 corresponde al caso de la onda de 10 Hz con sí misma, a la vez que los valores cercanos a 1 corresponden al análisis de la onda de 10 Hz con cada uno de sus armónicos. El único caso que resulta en un valor cercano a 0 es el de la Coherencia de la onda de 10 Hz respecto a la de 15 Hz (segunda fila), ya que estas no son frecuencias armónicas.

Cabe destacar que cada intervalo abarca una etapa distinta de los períodos de las ondas senoidales, ya que los instantes de inicio y fin difieren entre sí. En consecuencia, los resultados no se mantienen constantes a lo largo de cada uno de ellos. Este efecto explica, además, por qué la Coherencia entre las ondas de 10 y 90 Hz resulta cercana a 0 para los primeros dos intervalos pero más cercana a 1 para el tercer intervalo. Esta consideración resulta de suma importancia al momento de analizar la similitud de señales utilizando el método de Coherencia, ya que su análisis en un único intervalo podría conducir a conclusiones erróneas.

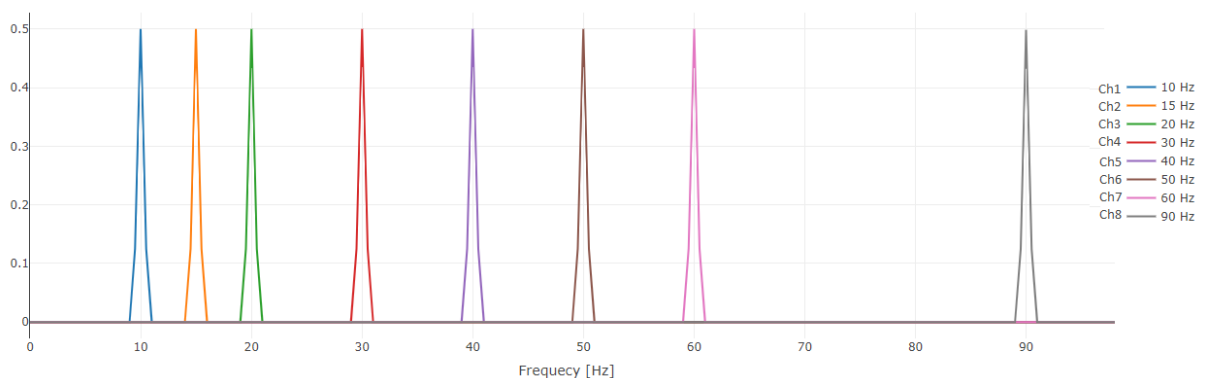


Figura 4.37: Espectro en Frecuencia de la Señal Ejemplo utilizada, para usar como guía al analizar las tablas siguientes.

	Intervalo 1	Intervalo 2	Intervalo 3
10 Hz vs 10 Hz	1.000000	1.000000	1.000000
10 Hz vs 15 Hz	0.018862	0.052375	0.034045
10 Hz vs 20 Hz	0.776172	0.858830	0.821562
10 Hz vs 30 Hz	0.914958	0.942389	0.955179
10 Hz vs 40 Hz	0.935864	0.956144	0.971881
10 Hz vs 50 Hz	0.913569	0.941115	0.947777
10 Hz vs 60 Hz	0.924881	0.948499	0.965410
10 Hz vs 90 Hz	0.106488	0.286372	0.860143

Figura 4.38: Tabla de resultados de Coherencia obtenidos durante la Verificación. Se presenta sólo la coherencia de la onda de 10 Hz respecto al resto.

Capítulo 5

Casos de Aplicación

La validación de los métodos implementados asegura la correcta utilización de los mismos, dentro de los parámetros y situaciones previamente establecidas. Luego, se vuelve factible el uso de la herramienta para el análisis de señales electroencefalográficas reales. A modo de ejemplo, se presentan a continuación tres:

5.1. Análisis de Patrones Fisiológicos en Frecuencia

Objetivo: Detectar patrones fisiológicos en una señal de EEG utilizando el análisis en el espectro de frecuencia, estos son:

5.1.1. Ondas Alfa

Ritmos cerebrales contenidos en la banda de frecuencia entre los 7 y 13 Hz y generados al momento en que el sujeto cierra los ojos.

Desarrollo:

El estudio de los patrones mencionados se realiza sobre una señal de cuatro canales cuyos electrodos están posicionados en PO7, PO8, Pz, POz (respectivamente). La señal fue registrada utilizando el hardware Ganglion de OpenBCI a una tasa de muestreo de 200 Hz sobre un sujeto sano de 22 años. Las ondas alfa deberían visualizarse en el espectro de frecuencia en la banda comprendida entre 7 y 13 Hz. En contraste, se analiza un intervalo con el sujeto con ojos abiertos sin estimulación de luz parpadeante para su comparación con los intervalos de ojos cerrados y estimulación. Todos los resultados se observan en las Figuras 5.1, 5.2 y 5.3.

Previo a transformar la señal al espectro de frecuencia, se procesó la señal aplicando los siguientes parámetros:

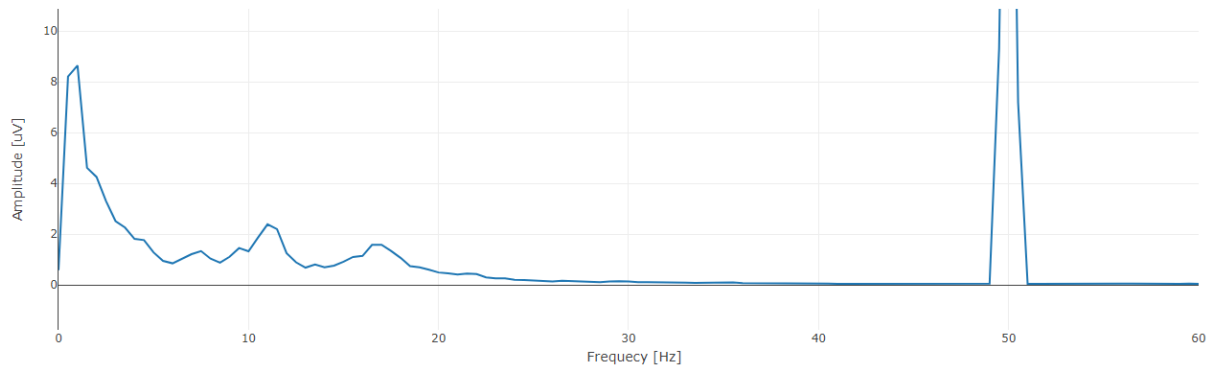
- Filtro Pasa-Banda de Orden 5 entre las frecuencias 5 y 60 Hz, para restringir la banda de frecuencias posibles.
- Filtro Notch en 50 Hz con un Ancho de Banda de 1 Hz (de 49 a 51 Hz), para eliminar el ruido de la tensión de línea.

Resultados:

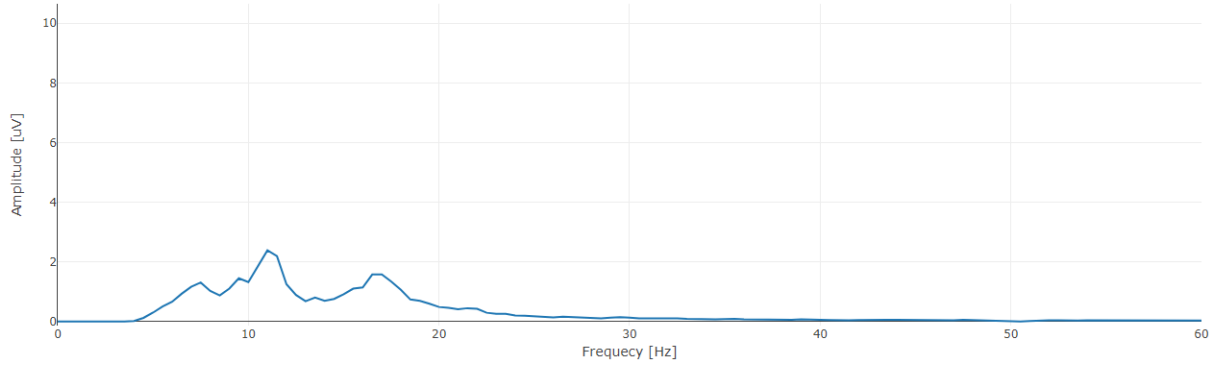
Además del acondicionamiento de la señal, se utilizó tanto el método CAR de rereferenciado como el método ICA para reducir el ruido, obteniendo los siguientes resultados:

El Rereferenciado con CAR logró eliminar gran parte del ruido, pero debido a que la información útil también se encontraba presente en los cuatro canales, su energía también fué reducida, como se visualiza en los incisos (c) de las Figuras.

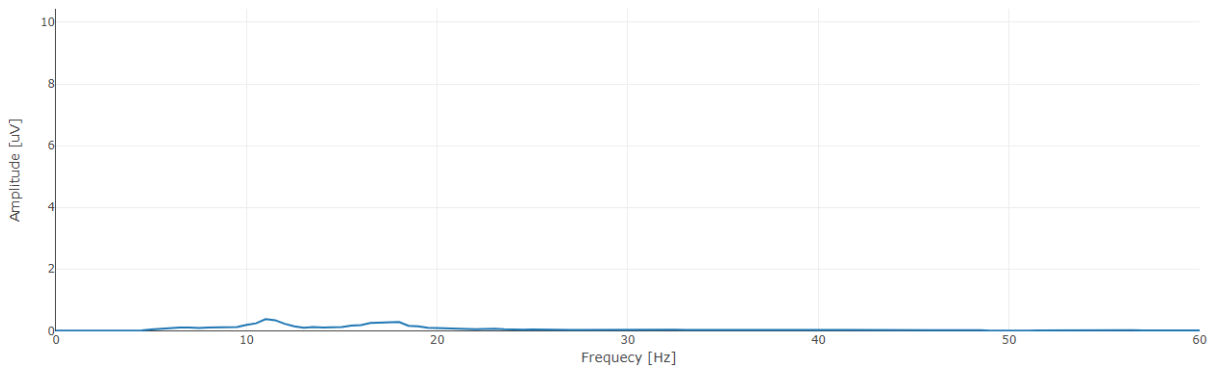
Por otra parte, el método ICA logró limpiar mucho mejor la señal, dejando muy en claro cada uno de los patrones buscados. Sin embargo, este requiere de un trabajo computacional mayor y que el usuario conozca el ruido que desea eliminar. Además, debido a que el método tiene bases probabilísticas, no puede ser automatizado fácilmente ya que cada vez que se aplica se obtienen resultados distintos. Los resultados de la aplicación de ICA pueden visualizarse en los incisos (d) de las Figuras 5.1 y 5.2.



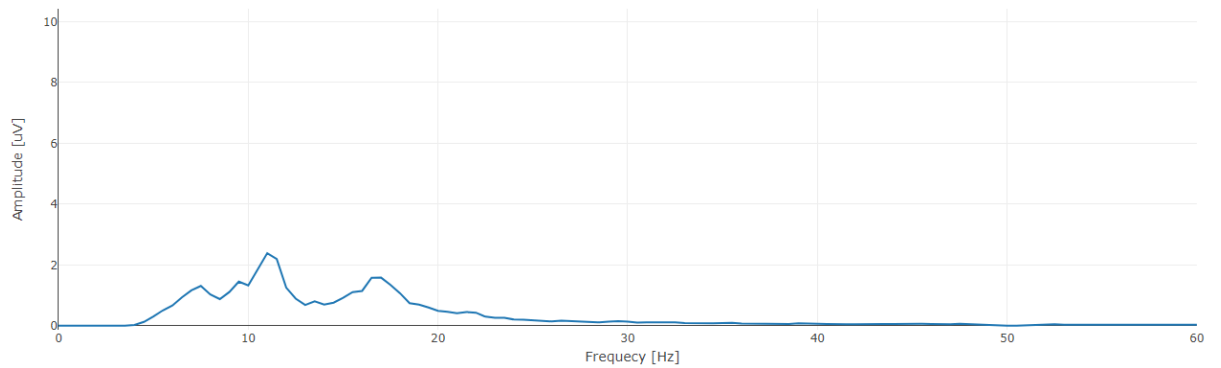
(a)



(b)

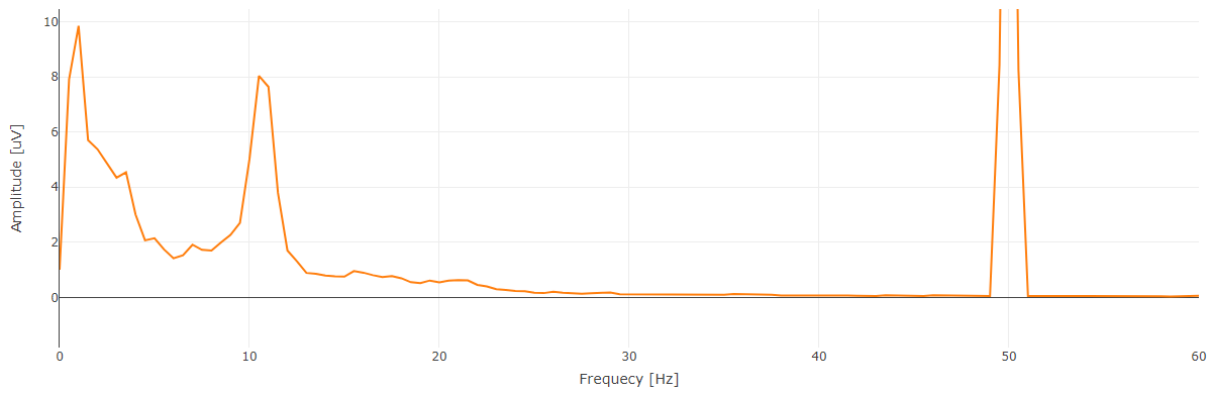


(c)

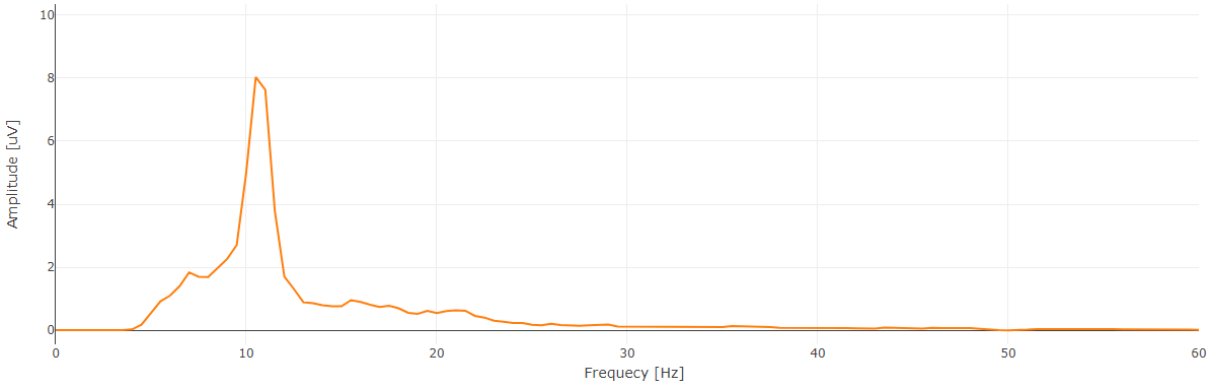


(d)

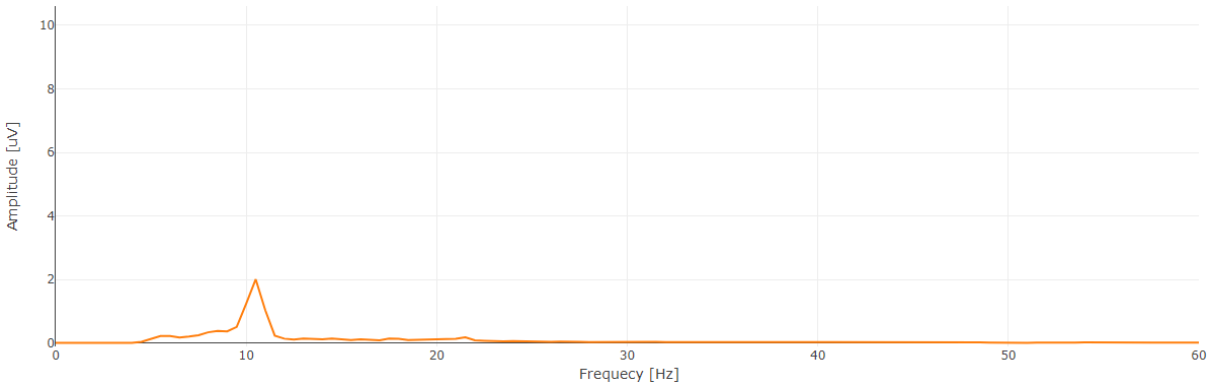
Figura 5.1: Espectro en Frecuencia de la señal en el intervalo en que el sujeto tenía los ojos abiertos, sin Procesamiento ni Acondicionamiento (a), con Acondicionamiento (b), con Acondicionamiento y CAR (c) y con Acondicionamiento e ICA (d).



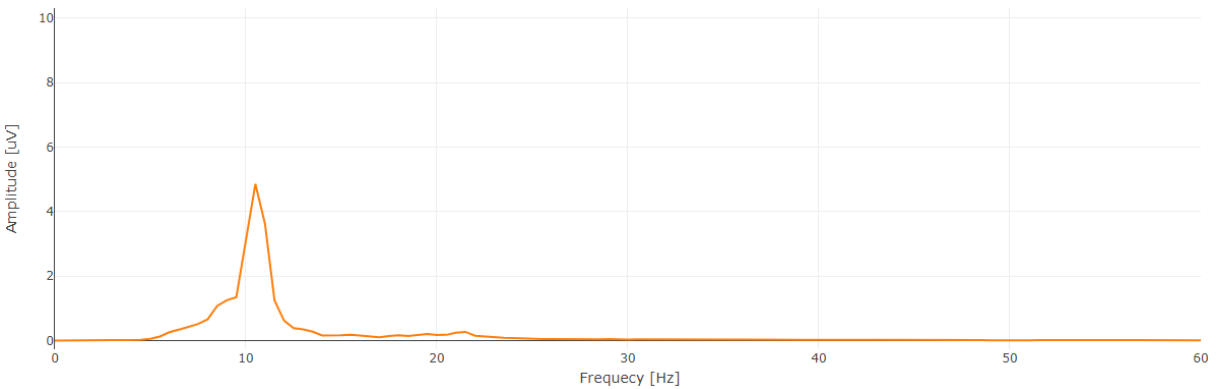
(a)



(b)



(c)



(d)

Figura 5.2: Espectro en Frecuencia de la señal en el intervalo en que el sujeto tenía los ojos cerrados, sin Procesamiento ni Acondicionamiento (a), con Acondicionamiento (b), con Acondicionamiento y CAR (c) y con Acondicionamiento e ICA (d).

5.1.2. Potencial Evocado Visual de Estado Estacionario (SS-VEP)

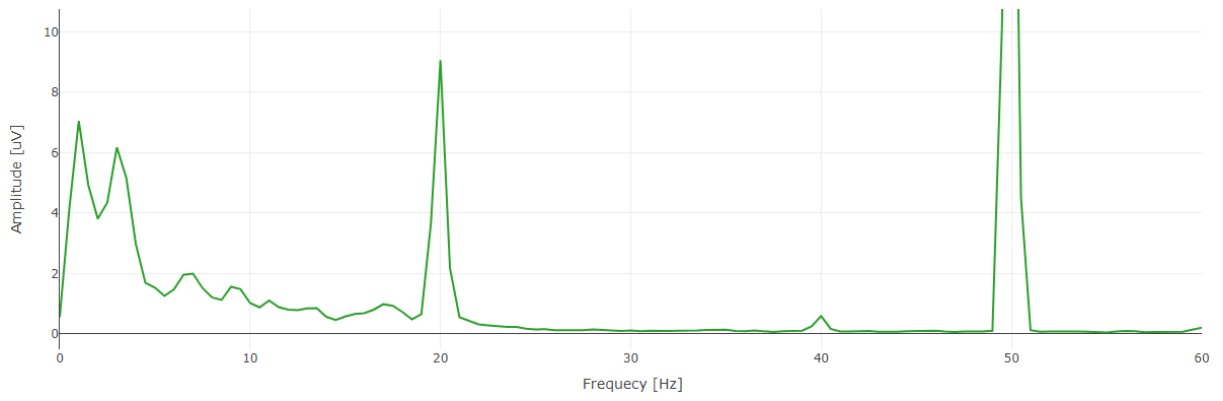
Ritmo generado en respuesta al estímulo visual de una luz parpadeante que puede observarse en el registro electroencefalográfico como una onda a una frecuencia igual a la del parpadeo de la luz.

Desarrollo:

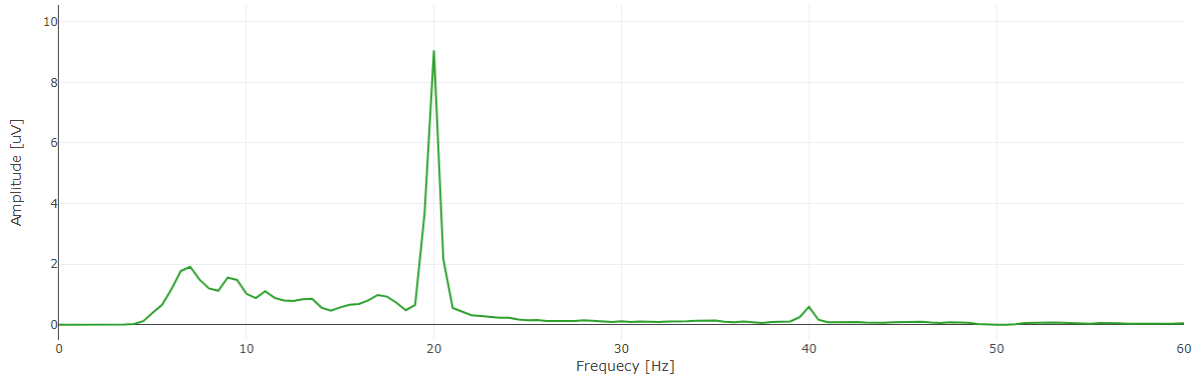
Para el estudio de los patrones SSVEP, se utilizó una señal registrada en las mismas condiciones que las utilizadas para el estudio de Ondas Alfa, y procesada bajo los mismos parámetros. Para generar el potencial SSVEP, la estimulación con luz se realiza a una frecuencia de 20 Hz por lo que esta misma debería observarse en el registro.

Resultados:

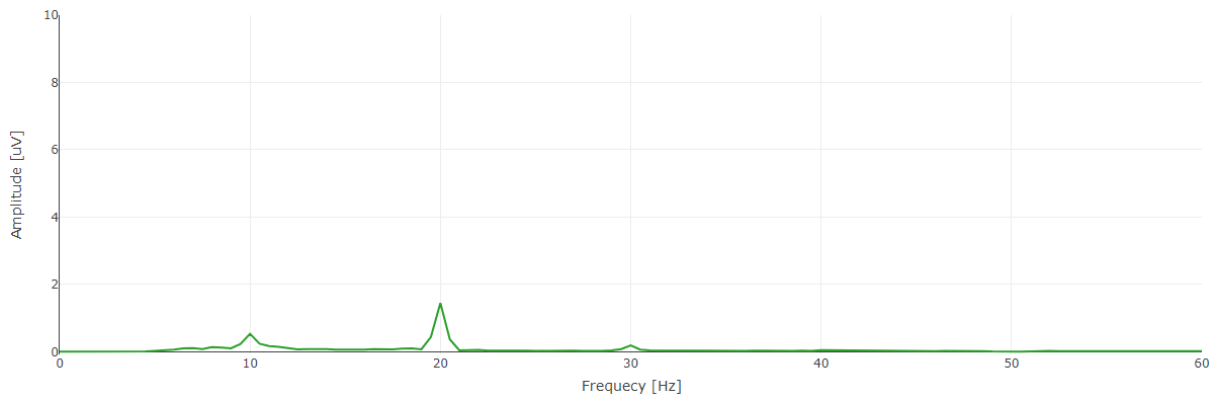
Al igual que en el caso de Ondas Alfa, el inciso (a) de la Figura 5.3 muestra la señal previa al procesamiento, en el inciso (b) se observa la señal luego de ser filtrada y acondicionada, mientras que los incisos (c) y (d) corresponden a la aplicación de los métodos CAR e ICA respectivamente. Los resultados fueron similares también a los obtenidos en el estudio de Ondas Alfa, donde el método CAR logra una buena reducción de ruido, pero mucho inferior a la lograda con el método ICA.



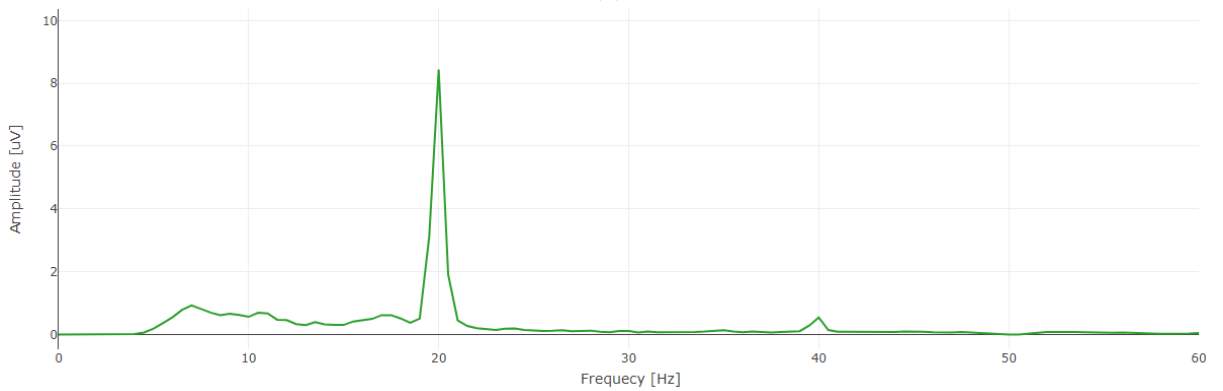
(a)



(b)



(c)



(d)

Figura 5.3: Espectro en Frecuencia de la señal en el intervalo en que el sujeto era estimulado por una luz a 20 Hz, sin Procesamiento ni Acondicionamiento (a), con Acondicionamiento (b), con Acondicionamiento y CAR (c) y con Acondicionamiento e ICA (d).

5.1.3. Conclusiones

Como conclusiones principales de este caso de aplicación:

- El principal método para reducir el ruido es el filtrado de la señal en las frecuencias de interés, y es crucial para el análisis.
- Ambos, métodos CAR e ICA, ayudan a disminuir el ruido restante. Sin embargo ICA, aunque más complejo, es más efectivo y resigna menos energía de señal útil.

5.2. Análisis Potencial Evocado P300

Objetivo: Análisis de la onda P300. La onda P300 es un potencial evocado que consiste en una deflexión positiva del registro electroencefalográfico generado por el paradigma de estimulación *odd-ball*, el cual plantea generar dicho potencial utilizando un estímulo de sorpresa. Luego de aplicado el estímulo, la deflexión se genera entre 200 y 500 milisegundos más tarde. En la Figura 5.4, se observa un diagrama explicativo del paradigma *odd-ball*, en el cual luego de varias instancias sin estimulación (o bajo una estimulación estándar), S, se genera un estímulo sorpresa, T; junto con el registro esperado para una onda P300, en el cual el origen de tiempo representa alguna instancia de estimulación T.

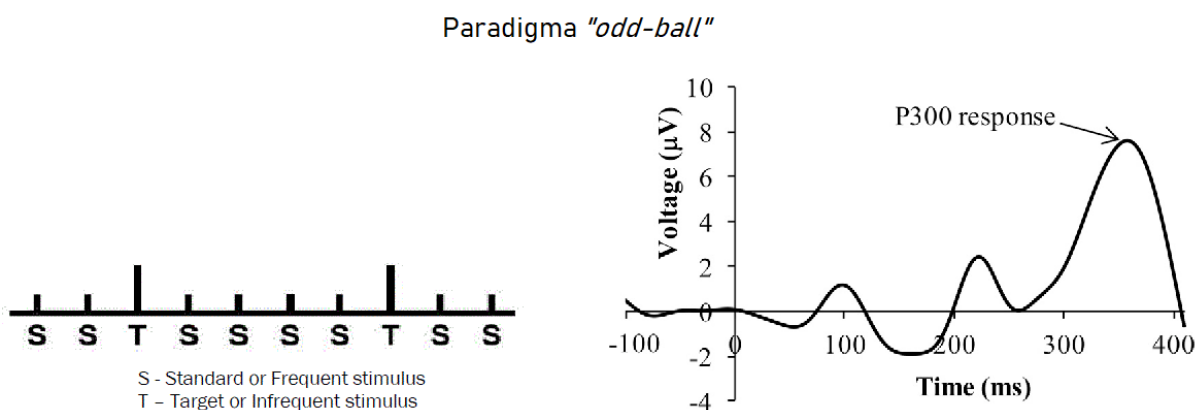


Figura 5.4: A la izquierda se visualiza el paradigma de estimulación *odd-ball*, donde las marcas S representan instancias de no estimulación o de estimulación estándar, mientras que las marcas T representan instancias de estimulación sorpresiva, no frecuente, las cuales provocan respuestas de onda P300. A la derecha se presenta un ejemplo de onda P300. En la misma, el instante de tiempo cero, origen de coordenadas de tiempo, representa alguna de las instancias de estimulación T del paradigma *odd-ball*.

Desarrollo: La señal utilizada para este caso proviene de los datasets creados para la plataforma BCI Competition, la cual consiste en un proyecto alemán para fomentar las contribuciones científicas en el ámbito de las BCI, y buscar la validación y el progreso del estado del arte de las mismas. La señal utilizada corresponde al Dataset IIb, el cual fue aportado por la comunidad científica Wadsworth Center, para el entrenamiento de inteligencias artificiales de BCI que utilicen el paradigma P300. La señal consta de 64

canales medidos a 240 Hz, utilizando la plataforma BCI2000.[15] Para reducir la cantidad de datos a procesar, se redujo la cantidad de canales a 16 de los 64 del estudio original, seleccionando aquellos en los que más probablemente se detectase la onda P300: aquellos que cubren las zonas parietal, temporal y occipital. Los canales finalmente elegidos se observan en la Figura 5.5.

La herramienta permite detectar el potencial P300 en una única época, es decir, un único intervalo que contenga el potencial, simulando el análisis en tiempo real. Sin embargo, para el estudio offline, en general se utilizan técnicas de promediado de épocas que evidencien mejor este potencial, debido a que es un potencial transitorio y que varía ampliamente de persona en persona.

21 Channels were detected, which ones will you use?

Ch45 column	<input checked="" type="checkbox"/> Import?	Channel Position:	TP7
Ch46 column	<input checked="" type="checkbox"/> Import?	Channel Position:	TP8
Ch47 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P7
Ch48 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P5
Ch49 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P3
Ch50 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P1
Ch51 column	<input checked="" type="checkbox"/> Import?	Channel Position:	Pz
Ch52 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P2
Ch53 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P4
Ch54 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P6
Ch55 column	<input checked="" type="checkbox"/> Import?	Channel Position:	P8
Ch56 column	<input checked="" type="checkbox"/> Import?	Channel Position:	PO7
Ch57 column	<input checked="" type="checkbox"/> Import?	Channel Position:	PO3
Ch58 column	<input checked="" type="checkbox"/> Import?	Channel Position:	POz
Ch59 column	<input checked="" type="checkbox"/> Import?	Channel Position:	PO4
Ch60 column	<input checked="" type="checkbox"/> Import?	Channel Position:	PO8
Ch61 column	<input checked="" type="checkbox"/> Import?	Channel Position:	O1
Ch62 column	<input checked="" type="checkbox"/> Import?	Channel Position:	Oz
Ch63 column	<input checked="" type="checkbox"/> Import?	Channel Position:	O2
Ch64 column	<input checked="" type="checkbox"/> Import?	Channel Position:	Iz
Stim column	<input checked="" type="checkbox"/> Import?	Channel Position:	RPA

Accept Channels

Figura 5.5: Lista de canales utilizados en el análisis y las posiciones que les fueron asignadas.

Una vez seleccionados los canales a utilizar se definieron los intervalos manualmente, ya que para ello se estudió la onda correspondiente al encendido del LED usado para la estimulación, para definir intervalos de 800 milisegundos posteriores a cada estimulación (asegurando de esta manera que se pueda encontrar la onda P300 tanto 200 como 500 milisegundos pos-estímulo). El arreglo de impulsos de estímulos de luz se visualiza en la Figura 5.6.

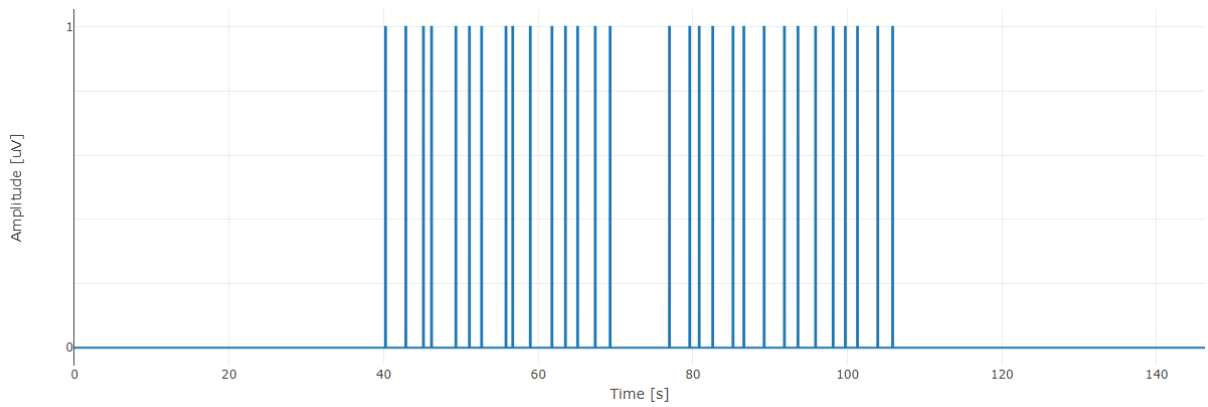


Figura 5.6: Señal completa de las estimulaciones llevadas a cabo para generar la onda P300. Las separaciones desiguales se deben a que la estimulación debe resultar sorprendente y no predecible para el usuario.

Además, para lograr una visualización más clara, se aplicó a la señal un acondicionamiento de:

- Filtro Pasa-Banda de Orden 5 entre las frecuencias 2 y 40 Hz, para restringir la banda de frecuencias posibles (la deflexión de la onda P300 es más lenta, por lo que los límites de la banda son inferiores).
- Filtro Notch no fue necesario ya que al filtrar por debajo de 40 Hz, la tensión de la línea se excluyó.

Resultados:

Una vez acondicionada la señal y definidos los intervalos donde se genera la estimulación, se analiza cada uno de ellos por separado en busca de la deflexión de la onda P300. En la Figura 5.7 se muestra uno de los resultados más notables. Éste corresponde a la tercera iteración de la sesión 010 del dataset con estimulación correspondiente a los 40,2 segundos. En el gráfico se observa el impulso de la estimulación de luz junto con el registro electroencefalográfico correspondiente, donde se visualiza la deflexión positiva alrededor de 200 milisegundos posterior al estímulo.

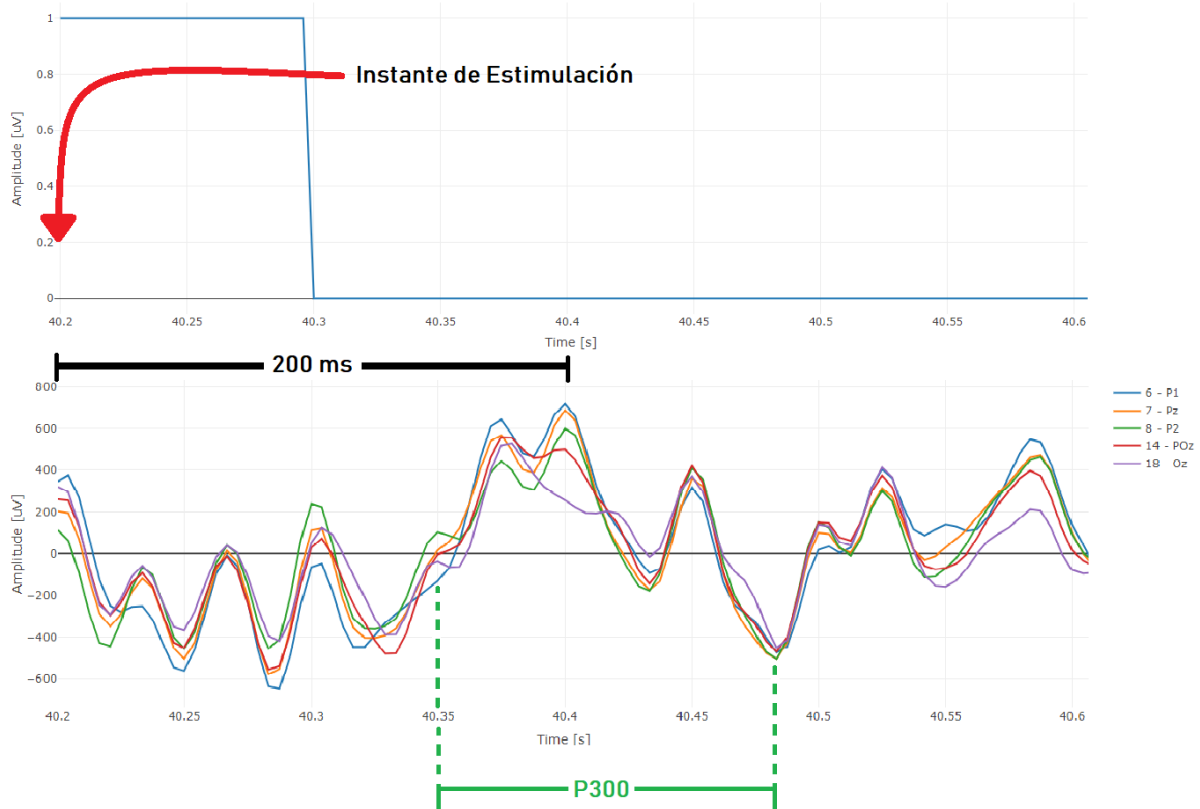


Figura 5.7: Arriba (a), la señal de encendido de la luz de estimulación. Abajo (b), el registro electroencefalográfico con la deflexión positiva centrada alrededor de los 200 milisegundos posteriores (40,4 segundos).

5.3. Eliminación de ruido

Objetivo: Eliminar o reducir el ruido proveniente del pestañeo del sujeto.
(ver Sección 1.4)

Desarrollo:

Para un tercer caso de uso, se seleccionó una señal contaminada por ruido de pestañeo y se propuso como objetivo la eliminación o reducción del mismo utilizando los métodos disponibles en la herramienta.

La señal fue registrada posicionando electrodos en Fp2, F7, Fp1 y F8 (en ese orden) sobre un sujeto sano, con la intención de generar artefactos de pestañeo sobre el registro de EEG e intentar eliminarlos en esta instancia. El ruido generado tiene una forma de onda que podría confundirse fácilmente con los artefactos generados por la despolarización de los músculos cardíacos (ECG), aunque observando la frecuencia de aparición cada 2 o 3 segundos (alrededor de 0,4 Hz), se verifica verificar que el mismo es generado por pestañeo.

Resultados:

Mediante la aplicación de los métodos CAR e ICA se intentó eliminar el ruido de pestañeo presente en todos los canales del registro. En la Figura 5.8 se observa un intervalo de la señal con artefacto de pestañeo, y para una mejor visualización se realiza

la ampliación de la imagen utilizando la funcionalidad zoom provista por los gráficos de Plotly. En las Figuras 5.9 y 5.10 se observa el mismo intervalo de la señal aunque procesada con los métodos CAR e ICA respectivamente.

Como resultado de la etapa, se observa que el método CAR reduce significativamente la energía total de las señales, y si bien el ruido fue atenuado en los canales verde y naranja, el mismo no se reduce de manera considerable en el canal azul. En cambio, el método ICA elimina correctamente el ruido del canal azul y atenúa el mismo en los canales verde y naranja, sin reducir la energía de la señal.

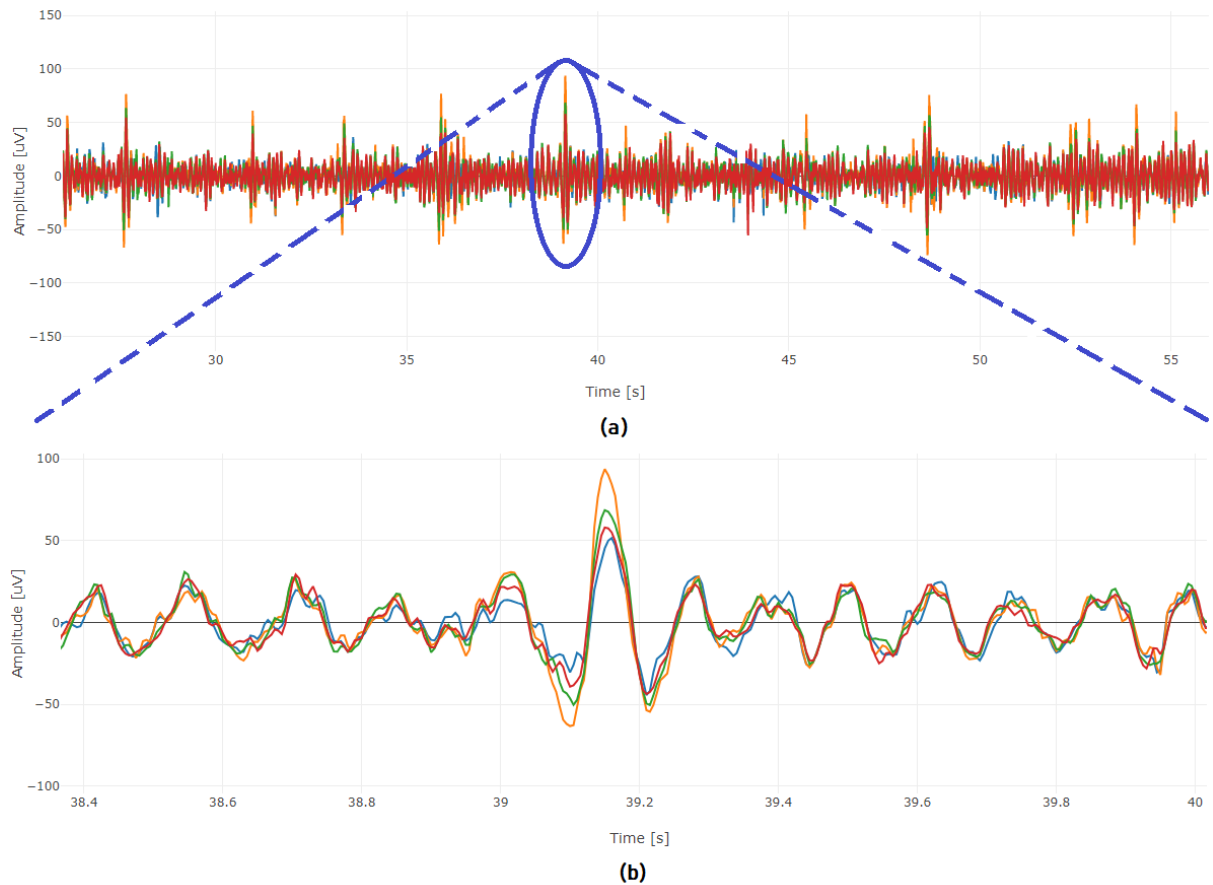


Figura 5.8: Artefacto de Pestañeo presente en la señal estudiada, visualizado en todo el intervalo (a), y con zoom (b).

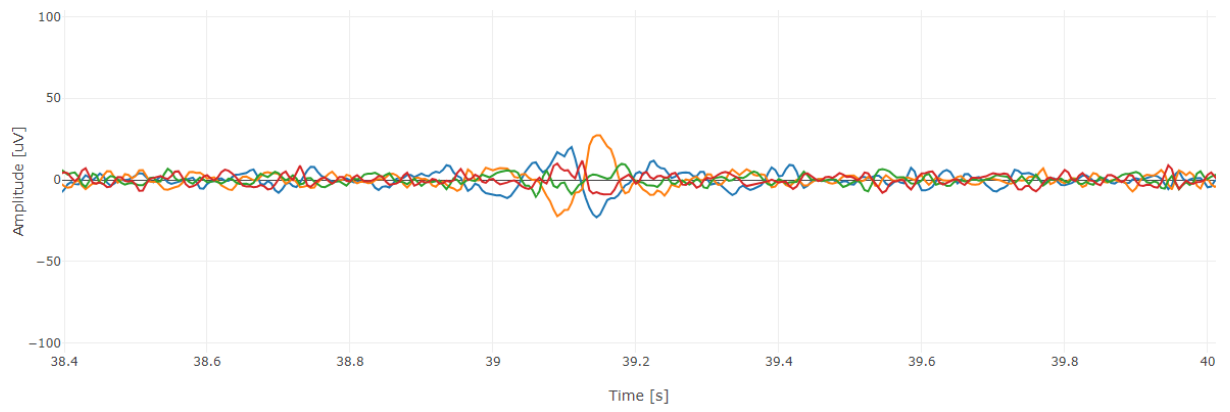


Figura 5.9: Artefacto de Pestaño luego de aplicar el método CAR de rereferenciado, con zoom.

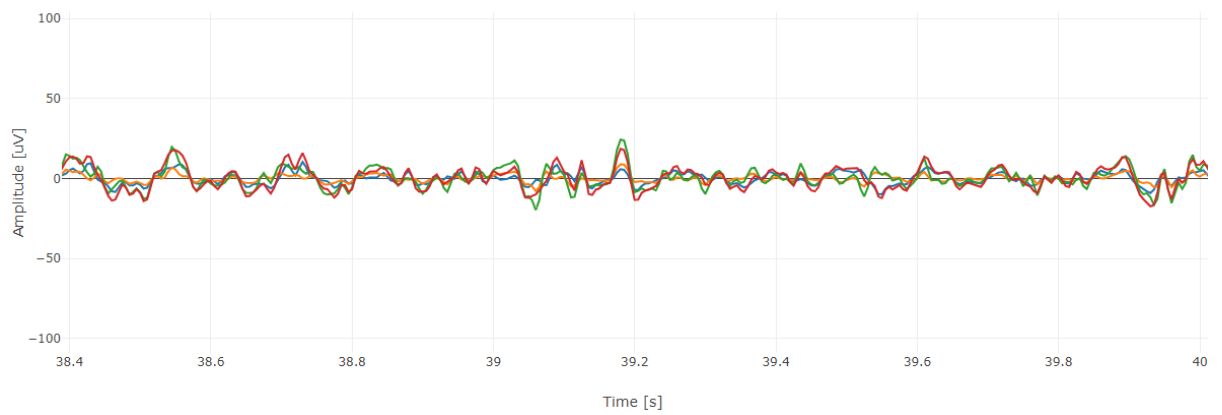


Figura 5.10: Artefacto de Pestaño luego de aplicar el método ICA, con zoom.

Conclusión

5.4. Conclusión

Con la realización del presente proyecto se obtuvo una herramienta para el procesamiento y análisis de registros electroencefalográficos de manera offline. Para llevar adelante este desarrollo se realizó un estudio del estado del arte de los métodos implementados con mayor frecuencia en el procesamiento de las señales de EEG. Esto permitió no sólo definir los métodos que mejor se adaptan a los requisitos propuestos, sino también establecer cuáles son los parámetros de diseño de cada uno.

Ante los requisitos propuestos, se dividieron los mismos en categorías según el área de procesamiento que abarcaban y se decidió desarrollar el proyecto a través de iteraciones. En cada una de ellas se agregaron nuevas funcionalidades que mitigaran los riesgos y cumplieran los requisitos. Esto resultó beneficioso en la gestión del proyecto ya que permitió tener en claro los objetivos de cada etapa y abordar específicamente cada área del procesamiento.

La herramienta desarrollada implementa dos funcionalidades principales: el acondicionamiento de la señal para su posterior análisis y la conversión de los registros en datos de calidad mediante tablas y gráficos que resaltan la información relevante de manera oportuna, es decir, presentando selectivamente los datos de interés para cada aplicación.

La herramienta prueba ser eficaz en el procesamiento y análisis de señales de electroencefalografía. Los métodos implementados, como por ejemplo el análisis ICA o la transformada Welch son algunos de los más utilizados actualmente en el estudio de bioseñales y más específicamente de señales electroencefalográficas. Éstos fueron verificados de manera correcta con siete datasets sintéticos y se validaron satisfactoriamente en tres casos de aplicación reales en ambientes controlados. Además, mediante el desarrollo de una interfaz gráfica, la herramienta puede ser fácilmente empleada incluso por personas con mínimos conocimientos informáticos.

Sin embargo se presentaron diversas problemáticas en gran parte debido al importante componente de codificación de software presente en la herramienta, cuestión que no era totalmente competente a los autores del proyecto. Dentro de las más notables se pueden mencionar: la integración de una interfaz gráfica para el control de funcionalidades, la localización de las posiciones de los electrodos en el modelado de mapas topográficos y la personalización de gráficos. A nivel hardware uno de los principales inconvenientes se presentó al momento de definir la tasa de muestreo de las señales obtenidas con la placa Ganglion de OpenBCI, debido a que la misma no provee información correcta sobre los tiempos de adquisición.

Para la resolución de las problemáticas se plantearon distintas soluciones que abarcan desde la adaptación de funciones hasta la modificación del formato de la señal, las cuales permitieron mitigar la totalidad de los riesgos asociados a los requerimientos obligatorios y lograr el funcionamiento esperado de la herramienta.

Por todo esto, se concluye que los objetivos planteados al inicio del proyecto fueron cumplidos de forma satisfactoria.

El desarrollo del Proyecto Integrador resultó muy enriquecedor en el aspecto formativo, tanto en la temática específica del procesamiento de señales y su aplicación a señales de EEG, como también en el área de programación de software.

5.5. Trabajo a futuro

Durante este proceso, se identificaron nuevos objetivos y propuestas a futuro que permiten dar continuidad al trabajo realizado, entre ellos cabe mencionar:

- Diseño de la herramienta según el paradigma de programación orientado a objetos.
- Diseño de una interfaz gráfica que permita al usuario el análisis de los registros desde una única ventana interactiva.
- Selección de los métodos y parámetros que mejor se adapten al estudio a llevar a cabo (Potenciales evocados, Detección de ritmos fisiológicos, eliminación de artefactos, etc.).
- Inclusión de mapas topográficos interactivos que permitan la visualización de los sectores activados en cada instante de tiempo.
- Incorporación de métodos y funciones basados en aprendizaje automático e inteligencia artificial, que asistan al usuario en la detección de potenciales y eventualidades del registro.

La implementación de estos puntos implican una mejora importante de las funcionalidades y usabilidad de la herramienta.

Referencias

- [1] Peter D. Welch. “The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over short, Modified Periodograms”. En: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967). DOI: 10.1109/TAU.1967.1161901.
- [2] Maria Corsi-Cabrera Miguel A. Guevara. “EEG Coherence or EEG Correlation?”. En: *International Journal of Psychophysiology* 23.3 (1996), págs. 145-153. DOI: 10.1016/S0167-8760(96)00038.
- [3] Dennis J. McFarland. “Spatial filter selection for EEG-based communication”. En: *Electroencephalography and clinical Neurophysiology* 103.3 (1997), págs. 386-394. DOI: 10.1016/S0013-4694(97)00022-2.
- [4] Steven W. Smith. *The Scientist and Engineer’s Guide to Digital Signal Processing*. International series of monographs on physics. California Technical Publishing, 1999. ISBN: 0966017668.
- [5] Scott Makeig Tzyy-Ping Jung. “Removing electroencephalographic artifacts by blind source separation”. En: *Psychophysiology* 37 (2000), págs. 136-178.
- [6] Jaime Parra Antonio Gil-Nagel. *Manual de Electroencefalografía*. S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, 2001.
- [7] John E. Hall Arthur C. Guyton. *Tratado de Fisiología Médica*. S.A. ELSEVIER ESPAÑA, 2006.
- [8] Pablo Cetta. *Filtros Digitales*. FACM - UCA, 2009.
- [9] Agnieszka Kitlas Golińska. “Coherence function in biomedical signal processing: a short review of applications in Neurology, Cardiology and Gynecology”. En: *Studies in Logic, Grammar and Rhetoric* 25.38 (2011), págs. 73-82.
- [10] Larson E. Gramfort A. Luessi M. “MNE software for processing MEG and EEG data.” En: *NeuroImage, PubMed*. 86 (2014), págs. 446-460.
- [11] Bastian Venthur. “Wym: A Brain-Computer Interface Toolbox in Python.” En: *Neuroinform* 13 (2015), págs. 471-486. DOI: 10.1007/s12021-015-9271-8.
- [12] Harriet Dashnow Juan Nunez-Iglesias Stéfan van der Walt. *Elegant Scipy: The Art of Scientific Python*. O’Reilly Media, Inc., 2017.
- [13] Andrea Noelia Bermúdez Cicchino. “Técnicas de procesamiento de EEG para detección de eventos.” Tesis de mtría.

- [14] Martinos Center for Biomedical Imaging MNE Development Team. *MNE Documentation*. Extraído: 06-2019. URL: <https://martinos.org/mne/stable/documentation.html>.
- [15] Gerwin Schalk. *2nd Wadsworth BCI Dataset (P300 Evoked Potentials)*. Extraído: 08-2019. URL: http://www.bbci.de/competition/ii/albany_desc/albany_desc_ii.pdf.
- [16] Ana Lucía Schmidt. *FFT: Transformada Rápida de Fourier*. Extraído: 04-2019. URL: <http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-Schmidt%5C%20Ana%5C%20Luc%5C%C3%5C%ADa.pdf>.
- [17] Anaconda Development Team. *Anaconda Distribution Documentation*. Extraído: 06-2019. URL: <https://docs.anaconda.com/anaconda/>.
- [18] Jupyter Development Team. *Jupyter Documentation*. Extraído: 06-2019. URL: <https://jupyter.readthedocs.io/>.
- [19] José Unpingco. *Python fo Signal Processing. Featuring IPython Notebooks*. EXTRA Materials. Springer (Science+Business Media). ISBN: 9783319013411.