


# Developing a data processing pipeline for brain mapping with Curry 7 and Python

Meglys Pérez-Bernal, MSc<sup>1</sup>, José Ávila-Paz, MSc<sup>1</sup>, Lauren Porta-Bastidas, UG<sup>1</sup>, Pedro Puentes-Rozo, PhD<sup>1</sup>





<sup>1</sup>Universidad Simón Bolívar, Colombia, [meglys.perez@unisimon.edu.co](mailto:meglys.perez@unisimon.edu.co), [jose.avila@unisimon.edu.co](mailto:jose.avila@unisimon.edu.co),

[lauren.porta@unisimon.edu.co](mailto:lauren.porta@unisimon.edu.co), [pedro.puentes@unisimon.edu.co](mailto:pedro.puentes@unisimon.edu.co)

**Abstract**– EEG (Electroencephalographic) signal processing and brain mapping face significant challenges due to low signal-to-noise ratio, inter-subject variability, and high data dimensionality. These limitations hinder the identification of neurophysiological patterns and reduce their clinical utility, especially in contexts requiring high diagnostic precision. Addressing these issues requires advanced preprocessing techniques, artifact removal, and spatial localization of brain activity, which adds complexity to the analysis. This study presents an automated pipeline developed in Python to process multichannel EEG signals (64 channels), acquired using the Neuroscan system and Curry Neuroimaging Suite 7 software in resting-state patients with eyes open. The tool enables data loading, visualization, cleaning, analysis, and export, facilitating its use by researchers and clinical professionals. Its main contribution lies in improving EEG data quality through automated procedures that reduce noise without compromising the original signal, thus supporting more accurate and reliable interpretation in clinical and neuroscientific applications.

**Keywords**-- Brain mapping, signal processing, filters, brainwave bands, Python.

# Desarrollo de un pipeline de procesamiento de datos para mapeo cerebral con Curry 7 y Python

Meglys Pérez-Bernal, MSc<sup>1</sup>, José Ávila-Paz, MSc<sup>1</sup>, Lauren Porta-Bastidas, UG<sup>1</sup>, Pedro Puentes-Rozo, PhD<sup>1</sup>

<sup>1</sup>Universidad Simón Bolívar, Colombia, [meglys.perez@unisimon.edu.co](mailto:meglys.perez@unisimon.edu.co), [jose.avila@unisimon.edu.co](mailto:jose.avila@unisimon.edu.co),

[lauren.porta@unisimon.edu.co](mailto:lauren.porta@unisimon.edu.co), [pedro.puentes@unisimon.edu.co](mailto:pedro.puentes@unisimon.edu.co)

**Resumen—** El procesamiento de señales EEG (Electroencefalográficas) y el mapeo cerebral enfrentan desafíos importantes debido a la baja relación señal-ruido, la variabilidad entre individuos y la alta dimensionalidad de los datos. Estas limitaciones dificultan la identificación de patrones neurofisiológicos y reducen su utilidad clínica, especialmente en contextos que requieren alta precisión diagnóstica. Para abordar estos retos, se requieren técnicas avanzadas de preprocesamiento, eliminación de artefactos y localización espacial de la actividad cerebral, lo que incrementa la complejidad del análisis. En este trabajo se presenta un pipeline automatizado desarrollado en Python para procesar señales EEG multicanal (64 canales), adquiridas con el sistema Neuroscan y el software Curry Neuroimaging Suite 7 en pacientes en reposo con los ojos abiertos. La herramienta permite cargar, visualizar, limpiar, analizar y exportar los datos, facilitando su manejo por parte de investigadores y profesionales clínicos. Su principal contribución es la mejora de la calidad de los registros EEG mediante procedimientos automáticos que reducen el ruido sin afectar la señal original, lo cual favorece una interpretación más precisa y confiable en aplicaciones clínicas y neurocientíficas.

**Palabras clave—** Mapeo cerebral, procesamiento de señales, filtros, bandas cerebrales, Python.

## I. INTRODUCCIÓN

La electroencefalografía (EEG) es una técnica no invasiva y ampliamente utilizada en las neurociencias, la cual es fundamental para la investigación de la actividad eléctrica cerebral y el diagnóstico clínico [1]. Su alta resolución temporal, permite capturar eventos neuronales en milisegundos, y su relativa asequibilidad la convierten en una herramienta indispensable para el estudio de una amplia gama de trastornos neurológicos. Entre ellos, el EEG es particularmente valioso en la caracterización y seguimiento de trastornos como la epilepsia, el deterioro cognitivo leve (DCL), y enfermedades neurodegenerativas progresivas como el Alzheimer y el Parkinson, donde la detección temprana de anomalías en la actividad cerebral puede influir significativamente en el manejo clínico y la investigación [2, 3].

Históricamente, el análisis de los datos de EEG ha dependido en gran medida de procesos manuales. Si bien la

experiencia humana es irremplazable en la interpretación final, el procesamiento manual de grandes volúmenes de datos brutos es naturalmente tedioso, susceptible a errores humanos y fundamentalmente difícil de replicar, lo que compromete la estandarización de los resultados y la validez entre estudios [4]. Esta limitación se agrava a medida que la cantidad de datos recolectados aumenta, debido al uso de tecnologías más modernas y estudios más prolongados.

En los últimos años, la combinación de los avances en la computación y la creciente disponibilidad de plataformas de código abierto ha revolucionado el procesamiento de señales EEG. El desarrollo de algoritmos computacionales implementados en lenguajes como Python ha permitido la automatización y estandarización de tareas que antes eran exclusivamente manuales. Herramientas y librerías especializadas como MNE-Python [5], EEGLAB (basado en MATLAB) [6] y módulos de procesamiento de señales como SciPy [7] han facilitado el acceso al manejo avanzado de análisis. Estas plataformas facilitan procesos complejos como el filtrado de ruido, la eliminación de artefactos (oculares, musculares, cardíacos), el análisis espectral (Transformada Rápida de Fourier - FFT), y la aplicación de técnicas de Transformada Wavelet para el análisis tiempo-frecuencia. Esta automatización no solo optimiza la eficiencia, sino que es un pilar fundamental para la reproducibilidad científica y la comparación de resultados entre diferentes centros de investigación [8].

Las tendencias actuales en el campo del procesamiento de EEG se centran en la integración de técnicas avanzadas para extraer información cada vez más completa y relevante. La aplicación de aprendizaje automático (Machine Learning) y, más recientemente, el aprendizaje profundo (Deep Learning), está transformando la capacidad de identificar patrones sutiles en los datos EEG para la clasificación de estados cerebrales, la detección temprana de patologías y la predicción de respuestas a tratamientos [9]. Por ejemplo, las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN) se utilizan para la detección automatizada de espigas epilépticas o la clasificación de etapas del sueño. Otra área en crecimiento es el análisis de la conectividad cerebral funcional, que busca entender cómo diferentes regiones del cerebro interactúan entre sí, proporcionando una visión más completa de la dinámica cerebral más allá del análisis de

frecuencias localizadas [10-12]. Asimismo, la necesidad de manejar grandes volúmenes de datos (Big Data) generados por estudios con miles de participantes o registros prolongados, ha impulsado el desarrollo de pipelines robustos y escalables que puedan procesar eficientemente esta información.

Proyectos recientes se han enfocado en automatizar este flujo de trabajo para grandes volúmenes de datos, especialmente en estudios multicanal. Por ejemplo, MNE-Python ha demostrado ser eficaz para trabajar con configuraciones estándar de 64 y 128 canales, permitiendo visualizaciones interactivas, análisis por bandas cerebrales y modelado de fuentes. Sin embargo, estos proyectos suelen requerir estructuras complejas de datos (por ejemplo, archivos.fif) y conocimientos avanzados de programación, lo cual limita su adopción en entornos clínicos donde se requiere simplicidad y velocidad en la generación de resultados [13].

A pesar de la existencia de estas potentes herramientas de código abierto, persisten desafíos significativos. Uno de ellos radica en la adaptación de estos recursos a los formatos de salida específicos de ciertos softwares de adquisición clínica. Equipos hospitalarios o de investigación a menudo utilizan software propietario, como Curry Neuroimaging Suite 7, que genera archivos en formatos binarios no estándar (como .dat) difíciles de integrar directamente con las librerías comunes, lo que crea un cuello de botella en el flujo de trabajo de análisis y la interoperabilidad de los datos [14].

Esta problemática se agudiza en el contexto específico del Grupo de Neurociencias del Caribe, donde los estudios de mapeo cerebral se realizan sistemáticamente utilizando el software de adquisición Curry Neuroimaging Suite 7. En el presente trabajo se realizaron mapeos cerebrales en pacientes adultos durante un periodo de cinco minutos, bajo condiciones estandarizadas que incluyeron un estado de vigilia plena, en posición sentada y con los ojos abiertos. Esta configuración experimental permitió garantizar la homogeneidad en la adquisición de las señales y reducir la influencia de variables externas, asegurando así la validez y confiabilidad de los registros obtenidos para su posterior análisis.

Este software de adquisición genera archivos en formato.dat, los cuales contienen datos de EEG multicanal (específicamente 64 canales). Si bien estos archivos tienen un valor clínico y científico incalculable para el diagnóstico y la investigación de trastornos neurológicos en la región Caribe de Colombia, su formato binario y la ausencia de herramientas automatizadas para su procesamiento masivo han representado una limitación crítica.

Esta situación genera un conjunto de problemas interconectados. Primero, la manipulación manual de estos archivos para su carga, visualización y preprocesamiento es excesivamente lenta y laboriosa, resultando en retrasos significativos en el procesamiento de datos. Esto, a su vez, dificulta la capacidad del grupo para analizar cohortes amplias de pacientes o realizar estudios longitudinales en tiempos razonables. Segundo, la dependencia de procesos manuales aumenta considerablemente el margen de error humano en

etapas críticas como la limpieza de artefactos y la segmentación de la señal, comprometiendo la calidad y fiabilidad de los datos finales. Finalmente, la dificultad para procesar masivamente y estandarizar los datos obstaculiza la identificación eficiente de patrones neurofisiológicos relevantes y biomarcadores asociados a las patologías estudiadas, lo que impacta directamente la toma de decisiones clínicas y el progreso de la investigación.

Ante este panorama, surge la necesidad de diseñar e implementar un pipeline automatizado en Python para el procesamiento integral de señales EEG multicanal obtenidas a través del software Curry 7. Este pipeline busca permitir la carga eficiente de datos, la limpieza de artefactos, la segmentación precisa por bandas cerebrales, el análisis espectral avanzado y la generación automatizada de informes clínicos detallados, todo ello con el fin de optimizar drásticamente el tiempo de análisis y facilitar la interpretación clínica de los resultados.

Para abordar el problema planteado, este trabajo se centró en la implementación de un pipeline modular y automatizado desarrollado íntegramente en Python. La elección de Python se justifica por su amplio ecosistema de librerías científicas de código abierto (NumPy, SciPy, MNE-Python, etc.), su flexibilidad para la manipulación de datos complejos y su capacidad para integrarse con diversas plataformas [15].

## II. METODOLOGÍA

Para el desarrollo del presente trabajo se utilizó un enfoque computacional aplicado al análisis de señales electroencefalográficas (EEG) multicanal. La metodología fue diseñada para ser completamente reproducible y modular, permitiendo la limpieza, segmentación, visualización y reporte automatizado de los datos. Fue ejecutado utilizando el lenguaje de programación Python, por su robustez en el procesamiento de señales biomédicas, su amplio ecosistema de librerías científicas, y su compatibilidad con formatos binarios como los generados por Curry 7.

### A. Herramientas y entorno de desarrollo

- Lenguaje de programación: Python 3.12
- Librerías utilizadas: numpy, scipy, matplotlib, pywt, pandas, sklearn, fpdf
- Hardware de procesamiento: PC con procesador Intel i7, 16 GB RAM
- Software de adquisición: Curry Neuroimaging Suite 7
- Formato de datos: .dat (float32, 64 canales)

### B. Fases

Este trabajo fue estructurado en once fases secuenciales y lógicas, diseñadas para cubrir el ciclo completo de procesamiento de EEG, desde la adquisición inicial hasta la generación de informes interpretables básicos. Para optimizar el análisis visual y los recursos computacionales, el pipeline

segmenta las señales en ventanas de 2 segundos. La estructura del código fue optimizada para el procesamiento eficiente de grandes volúmenes de datos, priorizando la vectorización de operaciones y la minimización del uso de bucles y memoria.

#### 1) Carga de datos

Los archivos .dat exportados desde Curry 7 fueron leídos en Python utilizando la estructura binaria float32, estableciendo 64 canales y una frecuencia de muestreo de 500 Hz. Se validó la integridad del archivo para asegurar que el número total de muestras sea múltiplo del número de canales.

#### 2) Visualización cruda

Se realizaron visualizaciones multicanal de la señal EEG sin procesar, segmentada en ventanas de 2 segundos, para facilitar la inspección visual canal por canal.

#### 3) Análisis frecuencial (FFT)

Se aplicó la transformada rápida de Fourier (FFT) para detectar artefactos o ruidos de alta energía en frecuencias específicas, especialmente en 60 Hz (interferencia de red).

#### 4) Filtrado

Se aplicó un filtro pasa banda (1–30 Hz) para eliminar frecuencias irrelevantes o no fisiológicas y posteriormente un filtro notch de 60 Hz para eliminar la interferencia eléctrica.

#### 5) Eliminación de artefactos (ICA)

Se utilizó análisis de componentes independientes (ICA) por ventanas para separar y eliminar componentes relacionados con artefactos fisiológicos o ambientales como parpadeos, movimientos musculares o ruido.

#### 6) Segmentación por bandas cerebrales

La señal fue filtrada en cinco rangos clásicos: Delta (0.2–4 Hz), Theta (4–8 Hz), Alfa (8–12 Hz), Beta (12–30 Hz) y Gamma (30–90 Hz), permitiendo evaluar la energía relativa por banda en cada canal.

#### 7) Análisis avanzado (Wavelet y espectrograma)

Se utilizó la transformada Wavelet discreta para la detección de eventos transitorios, y se generaron espectrogramas canal por canal, así como un espectrograma promedio para evaluar la evolución temporal de las frecuencias.

#### 8) Exportación de resultados

Todos los datos procesados fueron organizados y exportados en formato .csv, incluyendo la señal filtrada, la reconstruida por ICA, la señal limpia por Wavelet y las bandas cerebrales.

#### 9) Generación de informe clínico automático

Se incorpora una visualización tipo semáforo y un sistema de interpretación automatizada de la actividad por banda, integrando medidas de energía relativa y referencias clínicas preestablecidas para una evaluación rápida.

#### 10) Visualización y síntesis de resultados

Se generaron gráficas por canal y banda, así como un gráfico global con colores de advertencia, con el fin de facilitar una interpretación rápida del estado de la señal EEG.

#### 11) Generación de informe en PDF

Se automatizó la generación de un documento PDF que incluye gráficos, resumen cuantitativo por banda, semáforo de actividad y conclusiones clínicas automatizadas.

### III. RESULTADOS

El desarrollo completo del pipeline permitió demostrar que cada etapa contribuye al cumplimiento gradual del objetivo general, garantizando la calidad, organización y utilidad clínica de los datos procesados.

Se logró cargar correctamente archivos .dat, como se observa en la Fig. 1, obtenidos de registros realizados en Curry 7, en formato float32, utilizando numpy. Se desarrolló un método robusto que permite identificar y corregir discrepancias en el número de muestras (ajustándolo para que sea múltiplo del número de canales). Los datos fueron estructurados en una matriz de 64 canales por N muestras, con generación automática del vector de tiempo. La visualización inicial de la señal cruda en ventanas de 2 segundos por canal permitió una inspección clara de la señal (ver Fig. 2), identificando segmentos con artefactos visuales o ruido.

```
Archivo cargado correctamente. Total de muestras: 11171100
Advertencia: El número de muestras no es múltiplo del número de canales.
Duración total: 349.10 s
Formato de datos cargados: (64, 174548)
```

Fig. 1 Carga de archivo .dat.

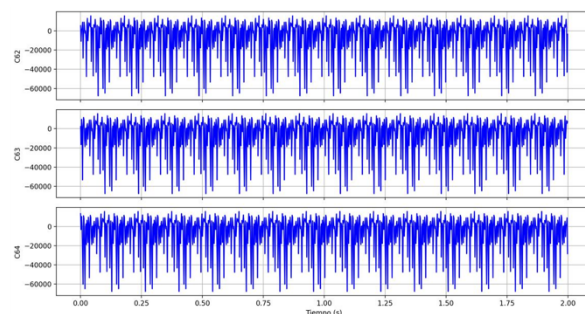


Fig. 2 Visualización de la señal cruda.

Posteriormente, se aplicó la transformada rápida de Fourier (FFT) para detectar artefactos o ruidos de alta energía en frecuencias específicas, especialmente en 60 Hz (interferencia de red) (Fig. 3).

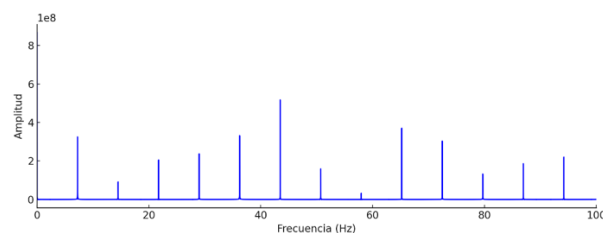


Fig. 3 FFT de la señal.

A continuación, se implementaron filtros notch (60 Hz) y pasa banda (1–30 Hz) (Fig. 4) mediante scipy.signal,

eliminando componentes de baja frecuencia y ruido de red eléctrica. Posteriormente, se aplicó Independent Component Analysis (ICA) (Fig. 5) con sklearn, por ventanas de 10 segundos, para identificar y eliminar artefactos comunes como parpadeos y movimientos.

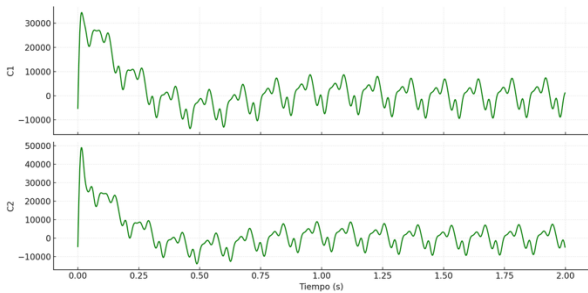


Fig. 4 Aplicación de filtros notch y pasa banda.

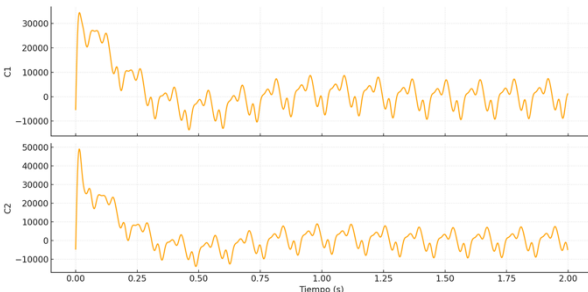


Fig. 5 Implementación de ICA.

Para el refinamiento, se utilizó transformada wavelet discreta (pywt) (Fig. 6), filtrando picos no deseados en la señal. Como resultado, se observó una mejora significativa en la forma de onda, reducción de ruido de alta frecuencia y eliminación de interferencias visibles. Esto puede evidenciarse en las gráficas comparativas entre señal cruda, filtrada e ICA, y en los gráficos individuales por canal (Fig. 7 y 8).

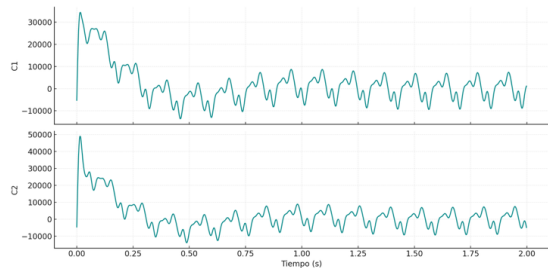


Fig. 6 Visualización de la transformada wavelet.

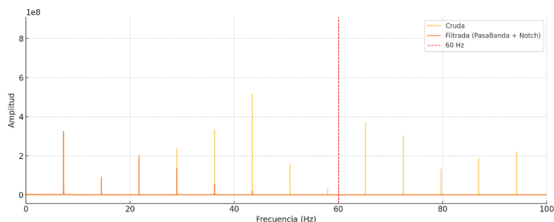


Fig. 7 Comparación de FFT antes y después del filtrado.

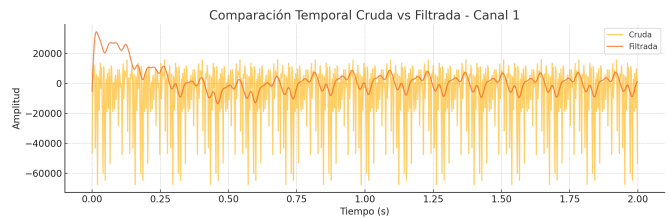


Fig 8 Comparación de la señal cruda vs filtrada.

Se definieron cinco bandas cerebrales clásicas: Delta (0.2–4 Hz), Theta (4–8 Hz), Alfa (8–12 Hz), Beta (12–30 Hz) y Gamma (30–90 Hz). Para cada una se aplicó filtrado pasa banda individual, extrayendo la actividad por banda de cada canal (Fig. 9 y 10). Posteriormente, se calcularon los porcentajes de energía relativa por banda y canal, lo que permitió establecer una distribución energética de la actividad neuronal. Esta información fue representada mediante gráficos individuales por banda/canal, gráficos generales de los 64 canales y un gráfico tipo “semáforo” que clasifica la actividad como normal, elevada o reducida, según rangos establecidos en la literatura (Fig. 11). La distribución espacial se infiere de la disposición de los canales (según el sistema 10-20), mientras que la dimensión temporal se observa en las visualizaciones secuenciales por ventanas de tiempo.

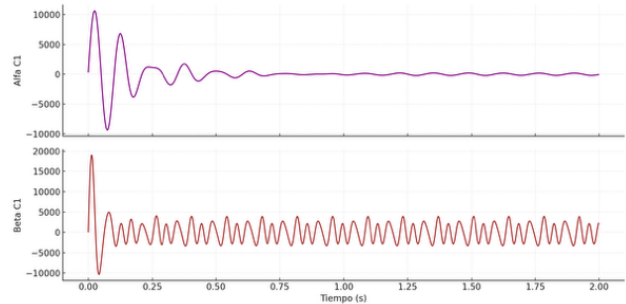


Fig. 9 Visualización individual de las bandas alfa y beta.

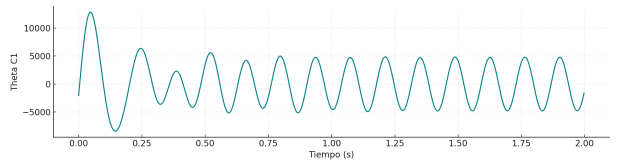


Fig. 10 Visualización de la banda theta.

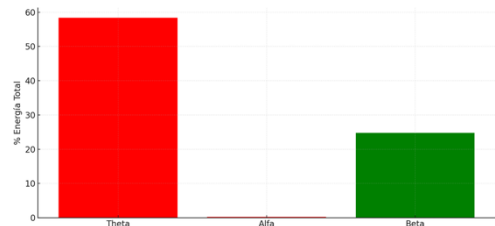


Fig. 11 Gráfico tipo semáforo.

Se automatizó la generación de gráficos de alta resolución de la señal EEG para cada etapa del proceso: cruda, filtrada,



ICA, wavelet y por bandas. Además, se desarrolló un informe clínico automatizado en formato .txt y .pdf, (Fig. 12 y 13) que incluye: datos técnicos del registro, porcentaje de energía por banda y canal., comentarios clínicos sugeridos, gráficos tipo semáforo, interpretaciones automáticas generadas por el sistema. Los informes se almacenan en carpetas organizadas por tipo de análisis, permitiendo su posterior revisión por el personal clínico.

```

== INTERPRETACIÓN CLÍNICA AUTOMÁTICA ==
Delta: Actividad Delta elevada (nan%)
Theta: Actividad Theta elevada (58.38%)
Alfa: Actividad Alfa reducida (0.25%)
Beta: Actividad Beta normal (24.80%)

```

Fig. 12 Interpretación con porcentajes de actividad.

```

== INFORME CLÍNICO AUTOMÁTICO DE EEG ==
Archivo analizado: DCL 003.dat
Frecuencia de muestreo: 500 Hz

Banda Delta - Canal 1: nan% - Actividad Delta elevada puede indicar sueño profundo o anomalías si está presente en vigilia.
Banda Delta - Canal 2: nan% - Actividad Delta elevada puede indicar sueño profundo o anomalías si está presente en vigilia.
Banda Theta - Canal 1: 58.48% - Actividad Theta alta puede estar relacionada con somnolencia o falta de atención.
Banda Theta - Canal 2: 58.28% - Actividad Theta alta puede estar relacionada con somnolencia o falta de atención.
Banda Alfa - Canal 1: 0.22% - Ondas Alfa predominantes suelen indicar un estado de relajación.
Banda Alfa - Canal 2: 0.28% - Ondas Alfa predominantes suelen indicar un estado de relajación.
Banda Beta - Canal 1: 24.79% - Ondas Beta elevadas pueden sugerir ansiedad, concentración o estrés.
Banda Beta - Canal 2: 24.81% - Ondas Beta elevadas pueden sugerir ansiedad, concentración o estrés.

```

Fig. 13 Informe clínico.

#### IV. DISCUSIÓN

El diseño modular por pasos implementado en el pipeline demostró ser una estrategia eficaz para garantizar escalabilidad y eficiencia en el procesamiento de grandes volúmenes de datos EEG. La optimización del código mediante el uso de vectores y matrices de NumPy redujo significativamente la dependencia de bucles iterativos, lo que permitió un manejo más ágil y estable de archivos extensos. Este enfoque se complementó con la aplicación de ICA segmentada por ventanas, que no solo mejoró la estabilidad del algoritmo, sino que también evitó bloqueos del sistema, mostrando ventajas frente a aproximaciones convencionales de análisis global.

El empleo de herramientas de código abierto como Python, NumPy, Matplotlib, SciPy, scikit-learn y FPDF refuerza la replicabilidad y portabilidad del trabajo, garantizando que la metodología pueda ser adaptada por otros grupos de investigación sin restricciones de licenciamiento. Además, la ausencia de datos clínicos identificables permitió cumplir con estándares éticos y de confidencialidad, aspecto fundamental en estudios que se proyectan hacia la práctica clínica.

En términos de aportes científicos, la integración de técnicas de filtrado, análisis espectral avanzado y segmentación en ventanas de dos segundos proporcionó un marco robusto para la interpretación de señales cerebrales, mejorando la capacidad de identificar patrones temporales sin

sacrificar la visión global de la señal. Sin embargo, persistieron limitaciones relacionadas con la calidad heterogénea de los registros y la necesidad de un prefiltrado más riguroso antes de aplicar ICA, lo cual evidencia la complejidad inherente al análisis de señales biológicas.

Finalmente, el pipeline plantea un potencial significativo de expansión al integrar en el futuro algoritmos de inteligencia artificial para la detección automática de anomalías, el desarrollo de interfaces gráficas que faciliten su adopción en entornos clínicos y la incorporación de bases de datos que sistematicen los resultados por paciente. Estas proyecciones permiten vislumbrar la consolidación de una herramienta no solo investigativa, sino también aplicable en la práctica hospitalaria, con un impacto positivo en el campo del neurodiagnóstico computacional.

#### V. CONCLUSIONES

El pipeline automatizado desarrollado e implementado en este estudio demostró ser una herramienta eficiente, estructurada y replicable para el análisis clínico de señales EEG multicanal, aportando soluciones innovadoras a retos clásicos como la limpieza, segmentación, análisis espectral y generación de informes automatizados. Su diseño modular y optimizado garantiza un desempeño ágil en el procesamiento de grandes volúmenes de datos, sin comprometer la estabilidad del sistema.

Este trabajo no solo responde a necesidades inmediatas en el análisis clínico de mapeos cerebrales, sino que establece una base sólida para el desarrollo de sistemas avanzados de neurodiagnóstico, capaces de integrar inteligencia artificial y gestión automatizada de resultados. En consecuencia, se concluye que la propuesta constituye un aporte relevante para la bioingeniería aplicada a la neurociencia clínica y abre nuevas perspectivas para la innovación tecnológica en el área del procesamiento de señales biomédicas.

#### AGRADECIMIENTOS

A la Universidad Simón Bolívar y al Grupo de Neurociencias del Caribe.

#### REFERENCIAS

- [1] F. Ramos-Argüelles, G. Morales, S. Egozcue, R. M. Pabón, and M. T. Alonso, "Técnicas básicas de electroencefalografía: principios y aplicaciones clínicas," [https://scielo.isciii.es/scielo.php?script=sci\\_arttext&pid=S1137-66272009000600006](https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1137-66272009000600006)
- [2] S. User, "1.4 Metodos diagnosticos en epilepsia: EEG y video-EEG," <https://guiaepilepsia.sen.es/index.php/1-4-metodos-diagnosticos-en-epilepsia-eeeg-y-video-eeeg>
- [3] S. Villazana, G. Montilla, A. Eblen, and C. Maldonado, "Epileptic signal detection using quilted synchrosqueezing transform based convolutional neural networks," *Computación Y Sistemas*, vol. 25, no. 2, May 2021, doi: 10.13053/cys-25-2-3461.
- [4] "NTP 620: Fiabilidad humana: evaluación simplificada del error humano (II) - PDF -Portal INSST - INSST," Portal INSST. <https://www.insst.es/documentacion/coleccion-tecnicas/ntp-notas->

- tecnicas-de-prevencion/18-serie-ntp-numeros-611-a-645-ano-2003/ntp-620-fiabilidad-humana-evaluacion-simplificada-del-error-humano-ii-
- [5] “Overview of MEG/EEG analysis with MNE-Python — MNE 1.9.0 documentation,” Feb. 26, 2025. <https://mne.tools/stable/autotutorials/intro/10overview.html>
  - [6] “EEGLAB,” EEGLAB - File Exchange - MATLAB Central File Exchange — MATLAB Central, Apr. 03, 2025. <https://la.mathworks.com/matlabcentral/fileexchange/56415-eeglab/>
  - [7] AryaKoureshi, “GitHub - AryaKoureshi/Signal-Processing-and-Analysis-of-EEG-Data-Using-Python: This project demonstrates various signal processing techniques, such as signal generation, window functions, filtering, downsampling, zero-padding, and the application of time-frequency analysis using the Short-Time Fourier Transform (STFT).,” GitHub. <https://github.com/AryaKoureshi/Signal-Processing-and-Analysis-of-EEG-Data-Using-Python>
  - [8] R. M. Mujica-Sequera, “AUTOMATIZACIÓN, PREDICCIÓN y NUEVOS PARADIGMAS EN LA CIENCIA| DOCENTES 2.0,” Docentes 2.0, Apr. 18, 2025. <https://blog.docentes20.com/2025/04/%E2%9C%8Dautomatizacion-prediccion-y-nuevos-paradigmas-en-la-ciencia-docentes-2-0/>
  - [9] E. E. De Bitbrain, “Cómo el Deep Learning está Revolucionando el Procesamiento de Señal de EEG,” Bitbrain, Apr. 25, 2025. <https://www.bitbrain.com/es/blog/deep-learning-y-senal-eeg>
  - [10] S. Villazana, G. Montilla, A. Eblen, and C. Maldonado, “Epileptic signal detection using quilted synchrosqueezing transform based convolutional neural networks,” Computación Y Sistemas, vol. 25, no. 2, May 2021, doi: 10.13053/cys-25-2-3461.
  - [11] J. A. Castillo, “Electroencephalographic analysis of the functional connectivity in habituation by graphics theory,” 2016. doi: 10.17488/rmib.37.3.3.
  - [12] Scen, “GitHub-sccn/eeg\_pipelines,” GitHub. [https://github.com/sccn/eeg\\_pipelines](https://github.com/sccn/eeg_pipelines)
  - [13] A. L. Torres, “Biomarcadores de neuroimagen para el diagnóstico temprano de la enfermedad de Alzheimer. Un abordaje desde las redes neuronales,” DELETED, vol. 26, no. 3, pp. 105–112, Sep. 2020, doi: 10.4067/s0717-93082020000300105.
  - [14] “Curry 7 PDF documentation — Compumedics Neuroscan” <https://compumedicsneuroscan.com/curry-7-pdf-documentation/>
  - [15] A. S. Guzmán and A. C. Ramírez, “Introduciendo SigProU, una interfaz gráfica de usuario para el análisis y procesamiento de señales EEG,” Ciencia Latina Revista Científica Multidisciplinar, vol. 5, no. 6, pp. 13060–13074, Dec. 2021, doi: 10.37811/clrcm.v5i6.1307.