

# Anexos

Análisis y tratamiento en tiempo real de la señal  
de un electroencefalograma con fines  
educativos

Streaming analysis and processing of an  
encephalogram signal for educational purposes

Autora

Claudia Gonzalo Gimeno

Directores

Isidro Urriza Parroqué

Jorge Luis Falcó Boudet

# Índice

1	Bibliotecas y paquetes.....	1
2	Configuración comando ‘Empezar una prueba en tiempo real’ con diferentes frecuencias .	3
2.1	20 Hz .....	3
2.2	50 Hz .....	5
2.3	100 Hz .....	6
2.4	250 Hz .....	8
2.5	500 Hz .....	9
3	Prueba FFT .....	11
4	Diagramas de flujo .....	13
4.1	Diagramas del modo serie .....	13
4.2	Diagramas del modo archivo.....	16
5	Manual de instalación.....	18
6	Manual de usuario .....	20
6.1	Registro de archivo.....	21
6.2	Nuevo registro .....	25
7	Datasheet manual EEG.....	28
8	Bibliografía .....	38
9	Índice de ilustraciones.....	38
10	Índice de tablas.....	39
11	Índice de ecuaciones.....	39

# 1 Bibliotecas y paquetes

En esta sección se detallan las bibliotecas y paquetes que se han empleado para realizar el presente proyecto, se hace una breve descripción de ellos y se menciona cual ha sido su utilidad a la hora de implementar la aplicación.

- **PyQtGraph:** es una biblioteca de GUI (interfaces gráficas de usuario) y gráficos de Python que está diseñado para su uso en aplicaciones matemáticas y científicas de ingeniería. A pesar de estar escrita completamente en Python, la biblioteca es muy rápida debido a su gran aprovechamiento de la biblioteca numpy para procesar números y por el marco de trabajo GraphicsView de Qt para la visualización. Se distribuye bajo la licencia de código abierto [1]. Esta biblioteca se ha empleado para generar el gráfico 2D en ambos modos de funcionamiento, tanto usando un registro de archivo como tomando y representando los datos en tiempo real. Se le permite al usuario desplazarse por el gráfico arrastrando el dibujo y, en la parte inferior izquierda del gráfico, se habilita un botón denotado por una 'A' que le permite al usuario poder visualizar todos los datos acumulados en la misma ventana. En el modo archivo se pueden representar todos los canales que se quieran, generando hasta cuatro subfiguras por fila ya que los datos se toman todos de vez, en cambio para el modo serie, se toman los datos de los dos electrodos del lóbulo frontal del alumno en tiempo real.
- **MNE:** Es un paquete Python de código abierto para explorar, visualizar y analizar datos neurofisiológicos humanos: MEG, EEG, sEEG, ECoG, NIRS y más. Tiene excelentes habilidades de trazado interactivo para explorar datos, y hay algunos comandos de trazado interactivos similares a GUI. Es de código abierto [2]. Se ha empleado para crear la cartografía cerebral, es decir un mapa topológico con los 32 canales de un registro de archivo del EEG. Se implementa con un slider para poder actualizar los datos respecto al tiempo e incorpora una leyenda que indica a que rango de valores le corresponde cada color del gráfico.
- **Serial:** Este módulo encapsula el acceso para el puerto serie, se publica bajo una licencia de software libre. Accede a la configuración del puerto a través de las propiedades de Python y posee soporte para diferentes tamaños de bytes, bits de parada, paridad y control de flujo [3]. Se ha empleado para establecer la comunicación entre el EEG y el ordenador vía serie mediante un puerto USB. Se ha implementado el código de forma que se revisan todos los puertos del ordenador y se hace una lista con los detectados indicando su puerto, su descripción y su hardware ID. Cuando se encuentra el puerto requerido, se actualiza como puerto de comunicación y se le asigna el valor del baud rate necesario para implementar de forma correcta la comunicación.
- **Threading:** Este módulo construye interfaces de subprocesamiento de nivel superior sobre el threadmódulo de nivel inferior, es decir genera paralelismo con hilos [4]. Se utiliza para crear y manejar hilos dentro de la aplicación, de esta forma se ejecutan los archivos .py que contienen lo relativo a los gráficos. Gracias a esto el usuario puede ver la aplicación a la vez que el gráfico emergente. También se emplea en el modo serie junto con una pantalla emergente en la opción 'Guardado de datos' para que el usuario pueda determinar cuando quiere finalizar el recibimiento de datos por parte del EEG.
- **Numpy:** Es el paquete fundamental para la informática científica con Python, ofrece funciones matemáticas completas, generadores de números aleatorios, rutinas de álgebra lineal, transformadas de Fourier y más. Es una fuente abierta, distribuida bajo una licencia liberal, la sintaxis de alto nivel de NumPy lo hace accesible y productivo para programadores [5]. Esta biblioteca, aparte de emplearla como recurso matemático

genérico, se ha empleado para obtener la transformada de Fourier de los datos obtenidos y así poder dividirlo en las diferentes bandas de frecuencia.

- **Matplotlib:** Es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python. Permite crear gráficos de calidad de presentación y figuras interactivas (zoom, desplazamientos, actualizaciones...), es de código abierto [6]. Gracias a esta biblioteca se han representado el diagrama de sectores y el gráfico de barras de las diferentes bandas de frecuencias de los datos del EEG. Esto se ha implementado tanto en su versión en tiempo real, donde el gráfico se actualiza sobre sí mismo de forma automática conforme recibe datos del EEG, como su versión de registro de archivo donde se cargan todos los datos una vez realizada su FFT, se dividen en figuras y se van actualizando mediante la inclusión de un slider. Se incluye un título con el nombre del canal y cada representación indica las bandas de frecuencia que se corresponden con cada barra o sección del gráfico.
- **Flask:** Es un microframework que permite crear aplicaciones web funcionales con Python. Al principio, con la instalación, no se tienen todas las funcionalidades, estas se pueden extender por medio de plugins. Emplear el patrón MVC (modelo vista controlador) es una manera o una forma de trabajar que permite diferenciar y separar lo que es el modelo de datos, la vista y el controlador. Incluye un servidor web de desarrollo, mediante el cual se puede ir actualizando la página para ver los resultados que se van obteniendo [7]. Es por este servidor web por lo que se elige flask para realizar la aplicación, los detalles sobre esta los encontramos en la memoria.
- **Tkinter:** El paquete tkinter es la interfaz estándar de Python para el kit de herramientas Tk GUI. Está disponible en la mayoría de las plataformas Unix, así como en los sistemas Windows [8]. Se ha empleado para crear la ventana emergente en el modo serie que le permite determinar al usuario en la opción 'Guardado de datos' cuando quiere finalizar dicho proceso.

Hay que mencionar que se han descrito las bibliotecas más destacadas, en el proyecto se han empleado otras básicas tales como: sys, time, math, random... que son más comunes, por lo que no se detallan.

## 2 Configuración comando ‘Empezar una prueba en tiempo real’ con diferentes frecuencias

En esta sección se explica la configuración el comando ‘Empezar una nueva prueba en tiempo real’ (0x11) para los 32 canales de la máquina Bitmed eXea Ultra con las diferentes frecuencias de: 10, 50, 100, 250 o 500 Hz. Se explica como se determinan los valores y el orden correspondiente para que la petición sea correcta.

En la memoria se detalla las partes que conforman la configuración de este parámetro, los primeros valores de todas ellas coinciden ya que en todos los casos empleamos los 32 canales, estos son:

**17:** Se corresponde con el comando 0x11 en decimal, ‘Empezar una nueva prueba en tiempo real’.

**126:** Refleja el número de bytes, su fórmula se corresponde con la Ecuación I.

*Ecuación I*

$$N = \frac{n^{\circ}\text{canales AC}}{4} + 2 * n^{\circ}\text{canales} + n^{\circ}\text{canales} + 2 + 2$$

En la Ecuación II encontramos el resultado final.

*Ecuación II*

$$126 = \frac{32}{4} + (2 * 32 + 6 * 2) + (32 + 6) + 2 + 2$$

Los canales AC (nº canales AC) son 32, mientras que todos los canales (nº canales) son los 6 canales de frecuencia fija y los 32 AC.

**0:** Configuración de canales, se envían ‘N’ bytes con el valor 0 siguiendo la Ecuación III.

*Ecuación III*

$$N = \frac{n^{\circ}\text{canales AC}}{4} = \frac{32}{4} = 8$$

Por lo que se envía el valor 0 ocho veces separados por comas.

El resto de los valores varían según la frecuencia elegida, los cuales se detallan a continuación.

### 2.1 20 Hz

En la Tabla 1 se observa la configuración del comando para una frecuencia de 20 Hz.

Configuración 32 canales, 20 Hz
<b>comnd_ComDirecta=[</b> 17, 126, 0, 0, 0, 0, 0, 0, 0, 0, 20, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 25, 50, 50, 50, 50, 50, 50, 2, 0, 140, 0 <b>]</b>

Tabla 1: Comando 'Empezar una prueba en tiempo real', 20Hz, 32 canales

A continuación, se explican los valores propios de esta frecuencia.

**20, 0 y 10, 0:** Tasa de muestreo, 2 bytes por canal

- **20, 0**-> Valor para los 32 canales AC a 20Hz.
- **10, 0**-> Valor para los canales restantes, frecuencia fija a 10 Hz.

**1 y 50:** Módulos del canal, le corresponde 1 byte por canal basándose en la Ecuación IV.

*Ecuación IV*

$$N = \frac{500}{\text{Tasa de muestreo}}$$

- **1**-> Para los canales AC, hay 32 bytes con el resultado de la Ecuación V separado por comas.

*Ecuación V*

$$N = \frac{500}{\text{Tasa de muestreo}} = \frac{500}{20} = 25$$

- **50** -> Para el resto de los canales, hay 6 bytes con el resultado de la Ecuación VI separado por comas.

*Ecuación VI*

$$N = \frac{500}{\text{Tasa de muestreo}} = \frac{500}{10} = 50$$

**50, 0:** Índice de relación, le corresponden 2 bytes en formato 'little endian', se calculan con la Ecuación VII.

*Ecuación VII*

$$N = \frac{\text{Máxima tasa de muestreo}}{\text{Mínima tasa de muestreo}} = \frac{20}{10} = 2$$

**140, 0:** Tamaño del paquete, le corresponden 2 bytes en formato 'little endian', lo cual se calcula con la Ecuación VIII.

$$N = 2 + a + 10$$

Siendo el valor de 'a' el correspondiente con la Ecuación IX.

$$a = 2 * \frac{\sum(\text{tasa de muestreo de cada canal})}{10}$$

Tras varias pruebas se concreta que el sumatorio se refiere solo a los canales AC por lo que el resultado se encuentra en la Ecuación X.

$$N = 2 + 2 * \frac{20 * 32}{10} + 10 = 140$$

Cada paquete formado por 140 bytes en total, los cuales están formados por la siguiente secuencia:

- 2 bytes de preámbulo
- 10 bytes de los canales de frecuencia fija (2 bytes para los 4 primeros y 1 byte para los dos últimos)
- 128 bytes de los 32 canales AC. Cada canal aporta 4 bytes que se corresponden con 2 muestras ya que son de 2 bytes.

## 2.2 50 Hz

En la Tabla 2 se observa la configuración del comando para una frecuencia de 50 Hz.

Configuración 32 canales, 50 Hz
<b>comnd_ComDirecta=[</b> <b>17, 126, 0, 0, 0, 0, 0, 0, 0, 0,</b> <b>50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0,</b> <b>50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0,</b> <b>50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0,</b> <b>50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0, 50, 0,</b> <b>10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0,</b> <b>10, 10, 10, 10, 10, 10, 10, 10,</b> <b>10, 10, 10, 10, 10, 10, 10, 10,</b> <b>10, 10, 10, 10, 10, 10, 10, 10,</b> <b>10, 10, 10, 10, 10, 10, 10, 10,</b> <b>50, 50, 50, 50, 50, 50,</b> <b>5, 0,</b> <b>76, 1</b> <b>]</b>

Tabla 2: Comando 'Empezar una prueba en tiempo real', 50Hz, 32 canales

A continuación, se explican los valores propios de esta frecuencia.

**50, 0 y 10, 0:** Tasa de muestreo, 2 bytes por canal

- **50, 0**-> Valor para los 32 canales AC a 50Hz
- **10, 0**-> Valor para los canales restantes, frecuencia fija a 10 Hz

**10 y 50:** Módulos del canal, le corresponde 1 byte por canal, se calcula con la Ecuación IV.

- **1->** Para los canales AC, hay 32 bytes con el con el resultado de la Ecuación XI separado por comas.

*Ecuación XI*

$$N = \frac{500}{Tasa\ de\ muestreo} = \frac{500}{50} = 10$$

- **50 ->** Para el resto de los canales, hay 6 bytes con el con el resultado de la Ecuación XII separado por comas.

*Ecuación XII*

$$N = \frac{500}{Tasa\ de\ muestreo} = \frac{500}{10} = 50$$

**50,0:** Índice de relación, le corresponden 2 bytes en formato ‘little endian’, que se calculan con la Ecuación XIII.

*Ecuación XIII*

$$N = \frac{Máxima\ tasa\ de\ muestreo}{Minima\ tasa\ de\ muestreo} = \frac{50}{10} = 5$$

**76, 1:** Tamaño del paquete, le corresponden 2 bytes en formato ‘little endian’ siguiendo la Ecuación VIII y la Ecuación IX. Por lo que el resultado de este parámetro se encuentra en la Ecuación XIV.

*Ecuación XIV*

$$N = 2 + 2 * \frac{50 * 32}{10} + 10 = 332$$

El número decimal 332, se corresponde con el 0x14C en hexadecimal, la transmisión comienza por el byte menos significativo, quedando en decimal de la siguiente forma: 76,1.

Cada paquete formado por 332 bytes en total, los cuales están formados por la siguiente secuencia:

- 2 bytes de preámbulo
- 10 bytes de los canales de frecuencia fija (2 bytes para los 4 primeros y 1 byte para los dos últimos)
- 320 bytes de los 32 canales AC. Cada canal aporta 10 bytes que se corresponden con 5 muestras ya que son de 2 bytes.

## 2.3 100 Hz

En la Tabla 3 se observa la configuración del comando para una frecuencia de 100 Hz.



Configuración 32 canales, 100 Hz
<b>comnd_ComDirecta=[</b> <b>17, 126, 0, 0, 0, 0, 0, 0, 0, 0,</b> <b>100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0,</b> <b>100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0,</b> <b>100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0,</b> <b>100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0, 100, 0,</b> <b>10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0,</b> <b>5, 5, 5, 5, 5, 5, 5, 5,</b> <b>5, 5, 5, 5, 5, 5, 5, 5,</b> <b>5, 5, 5, 5, 5, 5, 5, 5,</b> <b>5, 5, 5, 5, 5, 5, 5, 5,</b> <b>50, 50, 50, 50, 50, 50,</b> <b>10, 0,</b> <b>140, 2</b> <b>]</b>

Tabla 3: Comando 'Empezar una prueba en tiempo real', 100Hz, 32 canales

A continuación, se explican los valores propios de esta frecuencia.

**100, 0 y 10, 0:** Tasa de muestreo, 2 bytes por canal

- **100, 0**-> Valor para los 32 canales AC a 100Hz
- **10,0**-> Valor para los canales restantes, frecuencia fija a 10 Hz

**5 y 50:** Módulos del canal, le corresponde 1 byte por canal, se calcula con la Ecuación IV.

- **1**-> Para los canales AC, hay 32 bytes con el resultado de la Ecuación XV separado por comas.

*Ecuación XV*

$$N = \frac{500}{\text{Tasa de muestreo}} = \frac{500}{100} = 5$$

- **50** -> Para el resto de los canales, hay 6 bytes con el resultado de la Ecuación XVI Ecuación V separado por comas.

*Ecuación XVI*

$$N = \frac{500}{\text{Tasa de muestreo}} = \frac{500}{10} = 50$$

**10,0:** Índice de relación, le corresponden 2 bytes en formato 'little endian', que se calculan con la Ecuación VII.

*Ecuación XVII*

$$N = \frac{\text{Máxima tasa de muestreo}}{\text{Minima tasa de muestreo}} = \frac{100}{10} = 10$$

**140, 2:** Tamaño del paquete, le corresponden 2 bytes en formato 'little endian', siguiendo la Ecuación VIII y la Ecuación IX. Por lo que el resultado de este parámetro se encuentra en la Ecuación XVIII.

$$N = 2 + 2 * \frac{100 * 32}{10} + 10 = 652$$

El número decimal 652, se corresponde con el 0x28C en hexadecimal, la transmisión comienza por el byte menos significativo, quedando en decimal de la siguiente forma: 140, 2.

Cada paquete formado por 652 bytes en total, los cuales están formados por la siguiente secuencia:

- 2 bytes de preámbulo
- 10 bytes de los canales de frecuencia fija (2 bytes para los 4 primeros y 1 byte para los dos últimos)
- 640 bytes de los 32 canales AC. Cada canal aporta 20 bytes que se corresponden con 10 muestras ya que son de 2 bytes.

## 2.4 250 Hz

En la Tabla 4 se observa la configuración del comando para una frecuencia de 250 Hz.

Configuración 32 canales, 250 Hz
<b>comnd_ComDirecta=[</b> <b>17, 126, 0, 0, 0, 0, 0, 0, 0, 0,</b> <b>250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0,</b> <b>250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0,</b> <b>250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0,</b> <b>250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0, 250, 0,</b> <b>10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0,</b> <b>2, 2, 2, 2, 2, 2, 2, 2,</b> <b>2, 2, 2, 2, 2, 2, 2, 2,</b> <b>2, 2, 2, 2, 2, 2, 2, 2,</b> <b>2, 2, 2, 2, 2, 2, 2, 2,</b> <b>50, 50, 50, 50, 50, 50,</b> <b>25, 0,</b> <b>76, 6</b> <b>]</b>

Tabla 4: Comando 'Empezar una prueba en tiempo real', 250 Hz, 32 canales

A continuación, se explican los valores propios de esta frecuencia.

**250, 0 y 10, 0:** Tasa de muestreo, 2 bytes por canal

- **250, 0**-> Valor para los 32 canales AC a 250Hz
- **10,0**-> Valor para los canales restantes, frecuencia fija a 10 Hz

**2 y 50:** Módulos del canal, le corresponde 1 byte por canal, se calcula con la Ecuación IV.

- **1**-> Para los canales AC, hay 32 bytes con el resultado de la Ecuación XIX separado por comas.

$$N = \frac{500}{Tasa\ de\ muestreo} = \frac{500}{250} = 2$$

- **50** -> Para el resto de los canales, hay 6 bytes, con el resultado de la Ecuación XX separado por comas.

$$N = \frac{500}{\text{Tasa de muestreo}} = \frac{500}{10} = 50$$

**25,0:** Índice de relación, le corresponden 2 bytes en formato ‘little endian’, que se calculan con la Ecuación XXI.

$$N = \frac{\text{Máxima tasa de muestreo}}{\text{Minima tasa de muestreo}} = \frac{250}{10} = 25$$

**76, 6:** Tamaño del paquete, le corresponden 2 bytes en formato ‘little endian’ siguiendo la Ecuación VIII y la Ecuación IX. Por lo que el resultado de este parámetro se encuentra en la Ecuación XXII.

$$N = 2 + 2 * \frac{250 * 32}{10} + 10 = 1612$$

El número decimal 1612, se corresponde con el 0x64C en hexadecimal, la transmisión comienza por el byte menos significativo, quedando en decimal de la siguiente forma: 76, 6.

Cada paquete formado por 1612 bytes en total, los cuales están formados por la siguiente secuencia:

- 2 bytes de preámbulo
- 10 bytes de los canales de frecuencia fija (2 bytes para los 4 primeros y 1 byte para los dos últimos)
- 1600 bytes de los 32 canales AC. Cada canal aporta 50 bytes que se corresponden con 25 muestras ya que son de 2 bytes.

## 2.5 500 Hz

En la Tabla 5 se observa la configuración del comando para una frecuencia de 500 Hz.

Configuración 32 canales, 500 Hz
<pre>comnd_ComDirect=[   17, 126, 0, 0, 0, 0, 0, 0, 0, 0,   244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1,   244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1,   244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1,   244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1, 244, 1,   10, 0, 10, 0, 10, 0, 10, 0, 10, 0, 10, 0,   1, 1, 1, 1, 1, 1, 1, 1,   1, 1, 1, 1, 1, 1, 1, 1,   1, 1, 1, 1, 1, 1, 1, 1,   1, 1, 1, 1, 1, 1, 1, 1,   50, 50, 50, 50, 50, 50,   50, 0,   140, 12 ]</pre>

Tabla 5: Comando 'Empezar una prueba en tiempo real', 500 Hz, 32 canales

A continuación, se explican los valores propios de esta frecuencia.

**244, 1 y 10, 0:** Tasa de muestreo, 2 bytes por canal

- **250, 0**-> Valor para los 32 canales AC a 500Hz
- **10,0**-> Valor para los canales restantes, frecuencia fija a 10 Hz

**1 y 50:** Módulos del canal, le corresponde 1 byte por canal, se calcula con la Ecuación IV.

- **1**-> Para los canales AC, hay 32 bytes con el resultado de la Ecuación XXIII separado por comas.

*Ecuación XXIII*

$$N = \frac{500}{Tasa\ de\ muestreo} = \frac{500}{500} = 1$$

- **50** -> Para el resto de los canales, hay 6 bytes con el resultado de la Ecuación XXIV separado por comas.

*Ecuación XXIV*

$$N = \frac{500}{Tasa\ de\ muestreo} = \frac{500}{10} = 50$$

**50,0:** Índice de relación, le corresponden 2 bytes en formato ‘little endian’, que se calculan con la Ecuación XXV.

*Ecuación XXV*

$$N = \frac{Máxima\ tasa\ de\ muestreo}{Mínima\ tasa\ de\ muestreo} = \frac{500}{10} = 50$$

**140, 12:** Tamaño del paquete, le corresponden 2 bytes en formato ‘little endian’, siguiendo la Ecuación VIII y la Ecuación IX. Por lo que el resultado de este parámetro se encuentra en la Ecuación XXVI.

*Ecuación XXVI*

$$N = 2 + 2 * \frac{500 * 32}{10} + 10 = 3212$$

El número decimal 3212, se corresponde con el 0xC8C en hexadecimal, la transmisión comienza por el byte menos significativo, quedando en decimal de la siguiente forma: 140, 12.

Cada paquete formado por 1612 bytes en total, los cuales están formados por la siguiente secuencia:

- 2 bytes de preámbulo
- 10 bytes de los canales de frecuencia fija (2 bytes para los 4 primeros y 1 byte para los dos últimos)
- 3200 bytes de los 32 canales AC. Cada canal aporta 100 bytes que se corresponden con 50 muestras ya que son de 2 bytes.

### 3 Prueba FFT

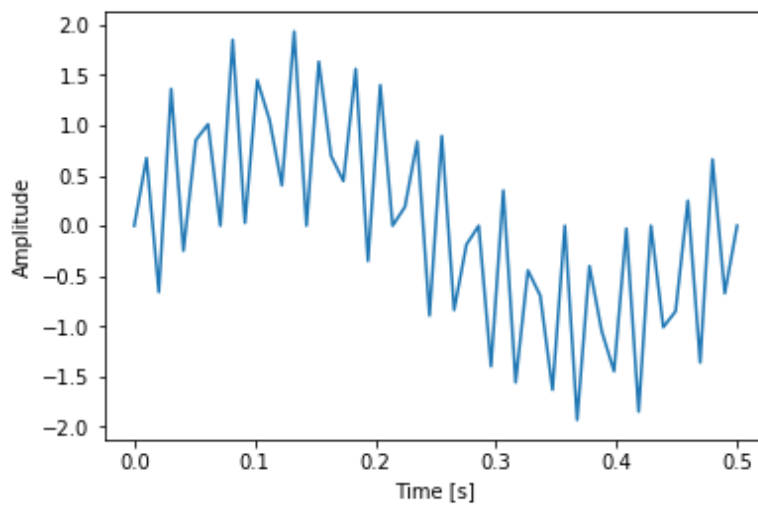
En esta sección se explica cuál ha sido el proceso de verificación del código implementado respecto a la FFT de los datos. En este código se incluye la FFT de los datos recogidos del EEG y su distribución en las cinco bandas de frecuencia.

Para esta verificación, se genera una onda sinusoidal de frecuencia 2 y 40 Hz tal y como podemos ver en la Ecuación XXVII.

*Ecuación XXVII*

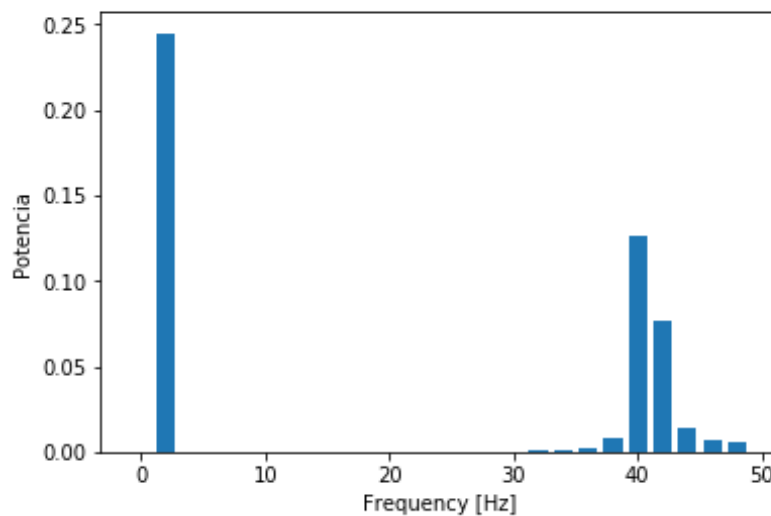
$$valores = \sin(2.0 * \pi * 2 * t) + \sin(2.0 * \pi * 40 * t)$$

Donde la ‘t’ se corresponde con el tiempo. El resultado de la onda generada para estos valores de frecuencia se puede observar en Ilustración 1.



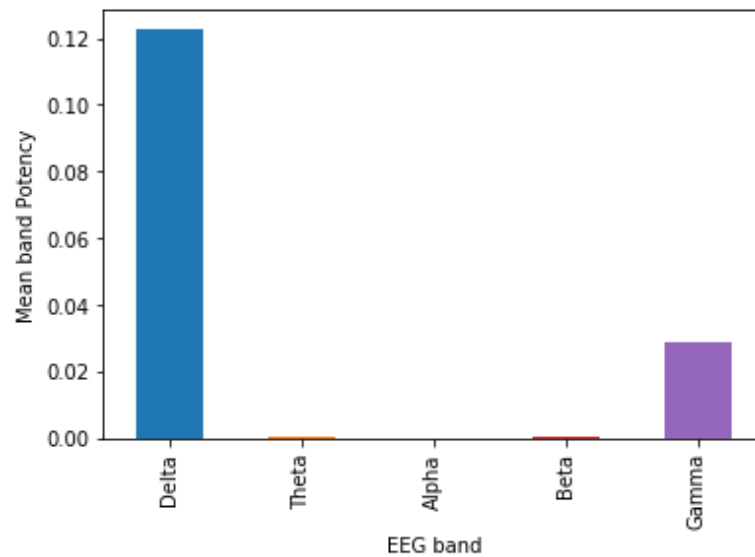
*Ilustración 1: Onda resultante*

A los datos de esta onda se les aplica el código implementado para obtener la potencia espectral, quedando visualmente esta distribución como en la Ilustración 2.



*Ilustración 2: Distribución de frecuencias*

Se observa como los picos se corresponden con los valores de frecuencia de 2 y de 40 Hz. Una vez obtenido este gráfico, gracias a un diccionario, se dividen estos valores en los diferentes rangos de frecuencias ya comentados en la memoria. El resultado de los valores de frecuencia en las diferentes bandas se observa en la Ilustración 3.



*Ilustración 3: Distribución en las diferentes bandas de frecuencia*

Se puede observar como el valor de frecuencia de 2Hz se encuentra dentro de la banda Delta y 40Hz dentro de la banda Ganma. La altura es diferente por la distribucción en la FFT que genera cada frecuencia y por el número de frecuencias que hay dentro de cada banda.

De esta forma queda verificado el correcto funcionamiento del código implementado para realizar la FFT de los datos recibidos del EEG y su correcta distribución en las distintas bandas de frecuencia. Así se asegura que los resultados obtenidos que se van a amostrar en la aplicación son útiles para el experto.

## 4 Diagramas de flujo

En esta sección se muestran y explican los diagramas de flujo acordes a los diferentes gráficos de ambos modos de funcionamiento de la aplicación. Estos diagramas se corresponden con los procesos ‘Lanzar gráfico’ mencionados en el diagrama de la estructura de la aplicación. Se dividen según el modo de funcionamiento de la aplicación: en los diagramas del modo serie y del modo archivo.

Como se ha mencionado en la memoria, el código se encuentra modulado según los diferentes tipos de gráficos (cada gráfico se corresponde con un archivo .py diferente) por ello hay diferentes diagramas que representan su comportamiento.

### 4.1 Diagramas del modo serie

Se explican los diagramas de flujo relativos al modo serie, que se corresponden con los gráficos disponibles cuando el usuario decide realizar un nuevo registro, en este modo los gráficos se actualizan conforme la aplicación recibe datos del EEG en tiempo real.

- **Guardado de datos:** Se corresponde con el archivo más simple de todos. En primer lugar, se establece la comunicación por el puerto serie entre el ordenador y el EEG. Dependiendo del valor introducido por el usuario, capturado por la aplicación, se enviará una configuración u otra con el comando ‘Empezar una nueva prueba en tiempo real’ (0x11), lo cual se corresponde con un nuevo registro. El valor capturado es la frecuencia, no se captura el valor de los canales ya que se guardan los datos de los 32 canales. Cuando la aplicación comienza a recibir datos del EEG, se abre una pequeña ventana que recogerá la información del proceso guardado de datos con un botón para finalizarlo. Mientras esa ventana esté abierta y no se presione ese botón el ordenador recogerá los datos y los ordenará por canales. Si el usuario cierra dicha ventana, el proceso de recogida terminará y se procederá a guardar los datos en una carpeta de la aplicación. El diagrama se encuentra en la Ilustración 4.

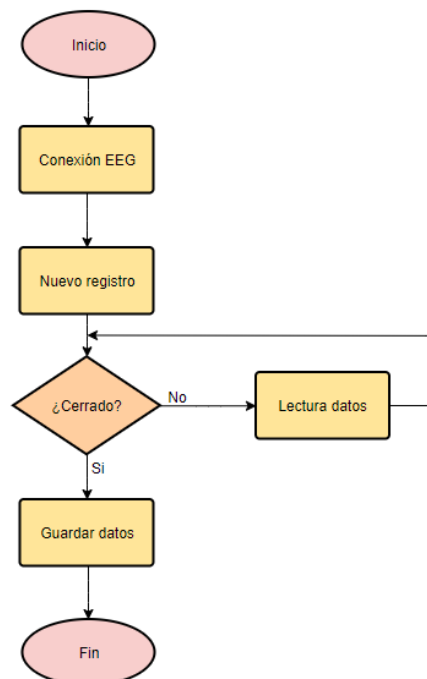


Ilustración 4: Diagrama de flujo del modo serie, guardado de datos

- **Gráfico de barras y diagramas de sectores:** En primer lugar, se establece la comunicación por el puerto serie entre el ordenador y el EEG. Dependiendo de los valores capturados por la aplicación, se enviará una configuración u otra con el comando ‘Empezar una nueva prueba en tiempo real’ (0x11), lo cual se corresponde con un nuevo registro, y abrirá una ventana en la que se irá posteriormente actualizando el gráfico en cuestión. Los parámetros capturados son la frecuencia, los canales y si se quiere guardar el registro.

Mientras el usuario no cierre la ventana del gráfico, la aplicación recibirá datos del EEG y los acumulará hasta llegar a un número representativo de muestras. Una vez llegado al mínimo número de datos necesarios para hacer una actualización, se realiza la FFT de los datos y se dividen en las cinco bandas de frecuencias. Finalmente se actualiza el gráfico con dichos valores.

En el caso de que el usuario cierre la ventana del gráfico, el programa guardará los datos su el usuario así lo requiere. El diagrama se encuentra en la Ilustración 5.

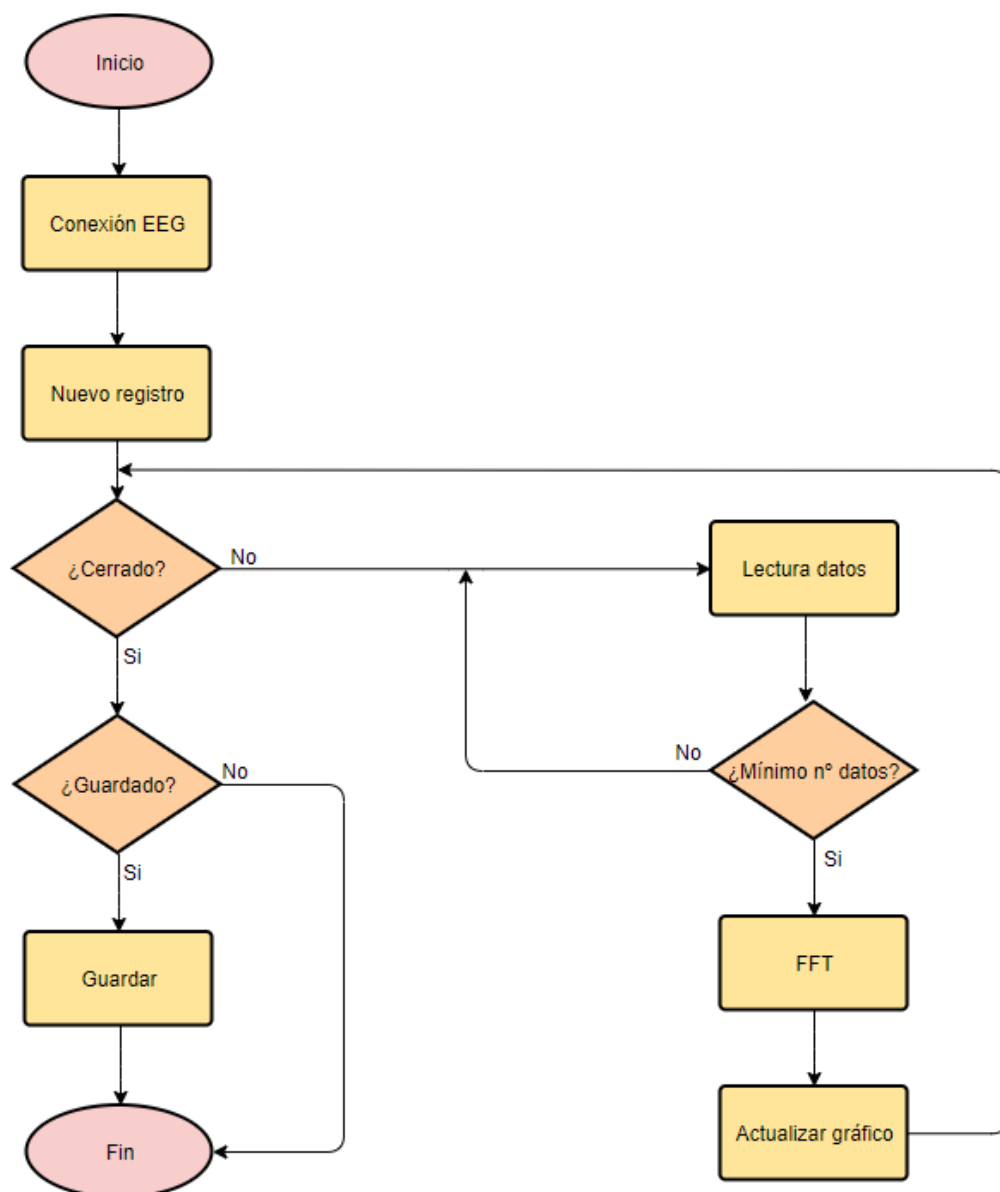


Ilustración 5: Diagrama de flujo del modo serie, gráfico de barras y diagrama de sectores



- **Visualización 2D:** Se establece la comunicación con el EEG, como se ha mencionado anteriormente teniendo en cuenta los parámetros requeridos por el usuario para empezar un nuevo registro en tiempo real y se abre la ventana del gráfico, en este caso el 2D. Mientras dicha ventana no se cierre, la aplicación recibirá los datos y estos se actualizarán uno a uno en el gráfico, desplazando el conjunto de datos hacia la izquierda. Si el usuario decide guardar los datos, una vez que se cierre la ventana se procederá al guardado de ellos en una carpeta de la aplicación.

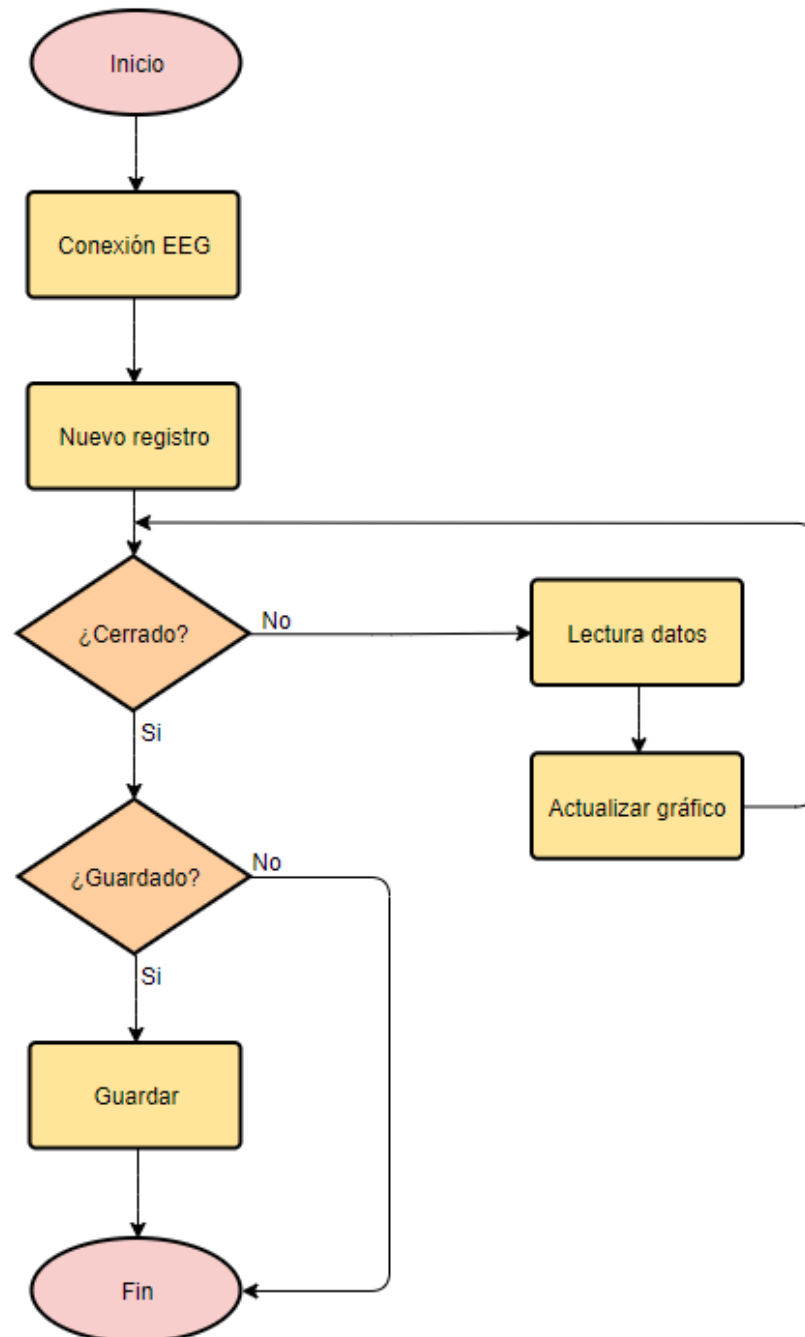


Ilustración 6: Diagrama de flujo del modo serie, gráfico visualización 2D

## 4.2 Diagramas del modo archivo

Este modo se corresponde con la decisión del usuario de querer realizar un registro de archivo, es decir, representando los datos de un registro guardado anteriormente. Los gráficos se disponen acumulados, es decir, se guardan todas las actualizaciones y mediante un 'slider' el usuario estudia la evolución de los datos.

- **Gráfico de barras y diagramas de sectores:** Teniendo en cuenta los parámetros introducidos por el usuario, se cargan los datos necesarios en la aplicación y se abre una pestaña inicial. A los datos cargados, se les realiza la FFT con su correspondiente distribución en frecuencias y se van añadiendo nuevas actualizaciones del gráfico una tras otra. Una vez se han representado todos los datos disponibles, se mantiene la ventana del gráfico abierta para que el usuario pueda analizar con detenimiento lo que necesite. Cuando se cierre la ventana se deja de visualizar el gráfico y su interacción. El diagrama que representa este comportamiento se encuentra en la Ilustración 7.

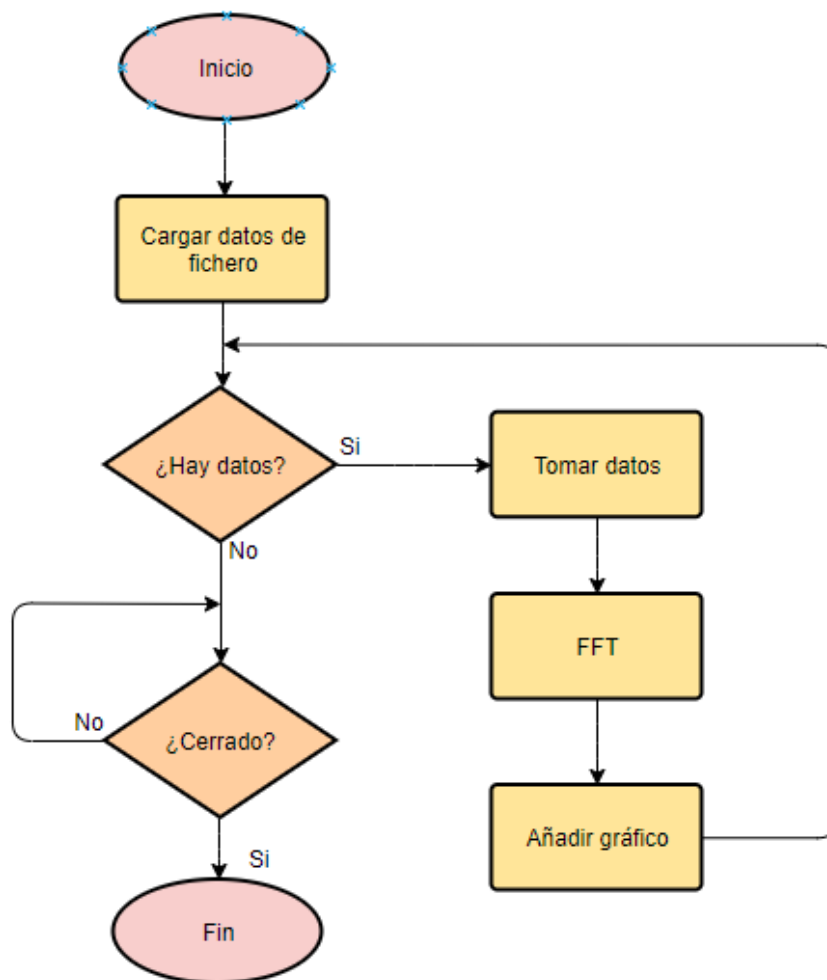


Ilustración 7: Diagrama de flujo del modo archivo, gráfico de barras y diagrama de sectores

- **Cartografía cerebral y visualización 2D:** En este caso, si se ha seleccionado la cartografía cerebral se cargan los ficheros de todos los canales para una determinada frecuencia establecida por el usuario. Se coge el primer valor de todos los canales y se realiza una actualización del gráfico (se añade un gráfico), se repite esto para todos los datos de los archivos. Una vez se han representado todos los datos disponibles en el formato requerido, se deja la ventana abierta para que el usuario pueda analizar con detenimiento lo que necesite. Incorpora un slider para moverse entre las diferentes actualizaciones del gráfico.

En el caso de la visualización 2D se representan los canales introducidos por el usuario de forma que se representan los datos uno tras otro en diferentes subfiguras (añadir gráfico) acorde al número de canales. El gráfico permite que el usuario desplace manualmente los datos con el ratón.

En ambos casos cuando el usuario cierre la ventana se finaliza el gráfico y su interacción, el diagrama de flujo que muestra el comportamiento del programa se puede observar en la Ilustración 8.

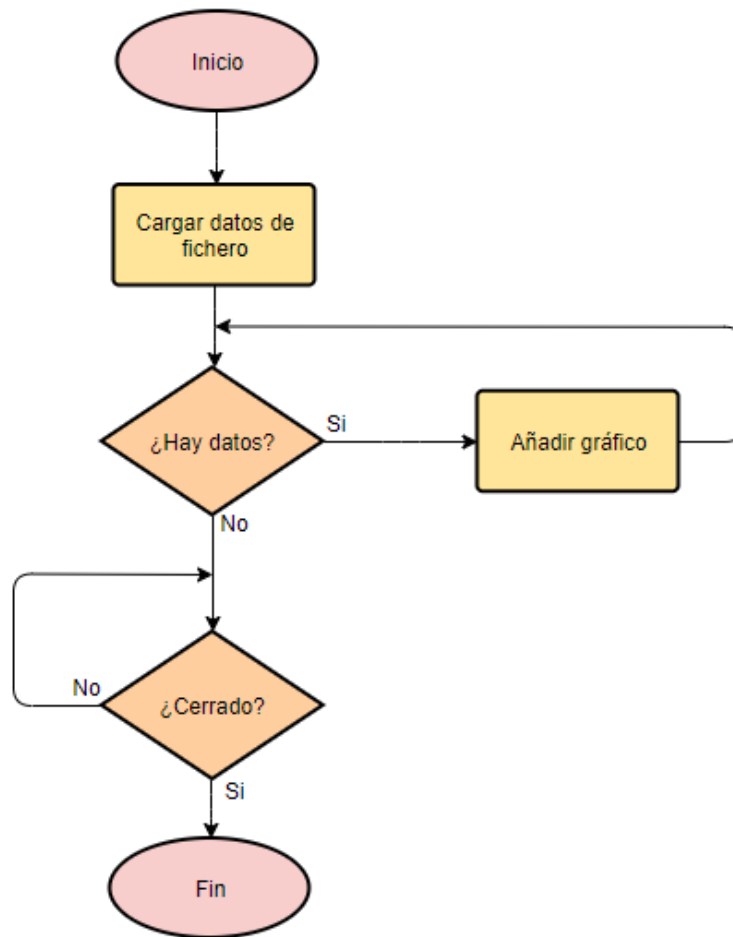


Ilustración 8: Diagrama de flujo del modo archivo, cartografía cerebral y visualización 2D

## 5 Manual de instalación

En esta sección se explica, en primer lugar, la instalación del EEG para que sea reconocido por el ordenador, en segundo lugar, como se lanza la aplicación. Se muestran los requisitos necesarios para que se ejecute correctamente. Se recomienda la lectura de este manual de instalación y del [manual de usuario](#) antes de ejecutar la aplicación por primera vez.

El sistema operativo sobre el que se ha desarrollado la conexión serie con el EEG y la aplicación es Windows 10, se recomienda por lo tanto emplear este sistema operativo para utilizar la aplicación ya que no se ha probado sobre otros. Hay que tener en cuenta que el lenguaje de programación, python, es multiplataforma pero el EEG no hay constancia de su funcionamiento sobre otro sistema operativo.

Antes de ejecutar la aplicación por primera vez, hay que instalar los drivers que se encuentran en la carpeta de la aplicación. Es muy importante instalarlos en el orden correcto tal y como se muestra en la Tabla 6. En caso de realizarlo en orden inverso el EEG no será reconocido por el ordenador y no se podrá ejecutar la aplicación. Estos drivers se encuentran en el repositorio.

DRIVER	ORDEN
ftdiport.inf	Primero
ftdibus.inf	Segundo

Tabla 6: Orden de instalación de los drivers

Para poder trabajar con la aplicación es requisito necesario tener instalado Python en el ordenador, ya que es el lenguaje de programación que emplea esta, al igual que las bibliotecas mencionadas en el [Anexo 1](#). La versión de python debe ser la empleada para el desarrollo de la aplicación, Python 3.8.5, o una compatible con esta.

En el repositorio se encuentra la carpeta propia de la aplicación que incluye un fichero bat encargado de ejecutar la aplicación, en Windows 10 basta con hacer doble click con el botón izquierdo del ratón.

Una vez se ejecuta este fichero, al usuario le aparece directamente la terminal. En esta aparece el siguiente mensaje: ‘Comenzando con la instalación de la aplicación web’ donde el usuario debe contestar a la pregunta: ‘Dispone de todas las bibliotecas instaladas?’ con una S (si) o con una N (no). Además, se le informa de que si pulsa cualquier letra se asumirá que dispone de todas las bibliotecas instaladas.

Las bibliotecas que se instalan automáticamente son: numpy, flask, qytGraph, mne, serial y tkinter.

En el caso de que seleccione si, se procederá a la instalación de las bibliotecas utilizadas por la aplicación, la terminal indica cuales posee ya en su ordenador y cuales está instalando. En cambio, si selecciona no, se carga la aplicación y se abre en el navegador Crome para que el usuario pueda visualizar la aplicación ya con todas sus utilidades.

La aplicación, como se acaba de comentar, se lanza sobre el navegador Crome, por lo que es requisito tenerlo instalado en el ordenador. Esta se abre en la ruta 127.0.0.1:5000, se ejecuta en local. Hay en ocasiones donde la aplicación no se carga tan rápido como se abre el navegador, esto ocurre generalmente si se ejecuta la aplicación nada más encender el ordenador, en caso de que esto ocurra basta con recargar la página de forma manual o pulsando la tecla F5.

En la Tabla 7 encontramos un resumen de los requisitos necesarios para poder ejecutar de forma correcta la aplicación tras la instalación de los drivers.

RECURSO	OBLIGATORIO
Lenguaje python	Si
Bibliotecas	No
Navegador Crome	Si

*Tabla 7: Requisitos de la aplicación*

## 6 Manual de usuario

En esta sección se encuentra detallado el funcionamiento de la aplicación, acompañado con capturas de esta, para servir de guía al usuario. Se detalla la interfaz de usuario de los dos modos de funcionamiento y sus correspondientes gráficos.

Al lanzarse la aplicación, esta se abre en la ruta 127.0.0.1:5000 del navegador Chrome donde se encuentra la vista inicial, en el directorio raíz, tal y como se observa en la Ilustración 9

En esta vista se encuentran los dos modos de funcionamiento de la aplicación, con una explicación de cada uno de ellos, finalmente se encuentra un botón para seleccionar el modo que el usuario desee.



*Ilustración 9: Página principal de la aplicación*

Independientemente del modo elegido la siguiente vista se corresponde con la selección de gráficos, es la observada en la Ilustración 10. Para cada modo se muestran los gráficos y opcionalidades propios. Hay un botón arriba a la izquierda que le permite al usuario regresar a la página anterior, la que se corresponde con la Ilustración 9.



*Ilustración 10: Página selección de la aplicación*

A continuación, se detalla lo correspondiente con los dos modos de funcionamiento de la aplicación: nuevo registro y registro de archivo.

## 6.1 Registro de archivo

En este modo no se toman datos directamente del EEG sino que se cargan registros anteriores realizados por este, de esta forma se puede analizar con detenimiento un evento o suceso que ocurra en el alumno y se visualice en los distintos gráficos que proporciona la aplicación.

Para que el usuario seleccione este modo ha de seleccionarse en la página principal: 'Registro de archivo' y hacer click en el botón de continuar tal y como vemos en la Ilustración 11.

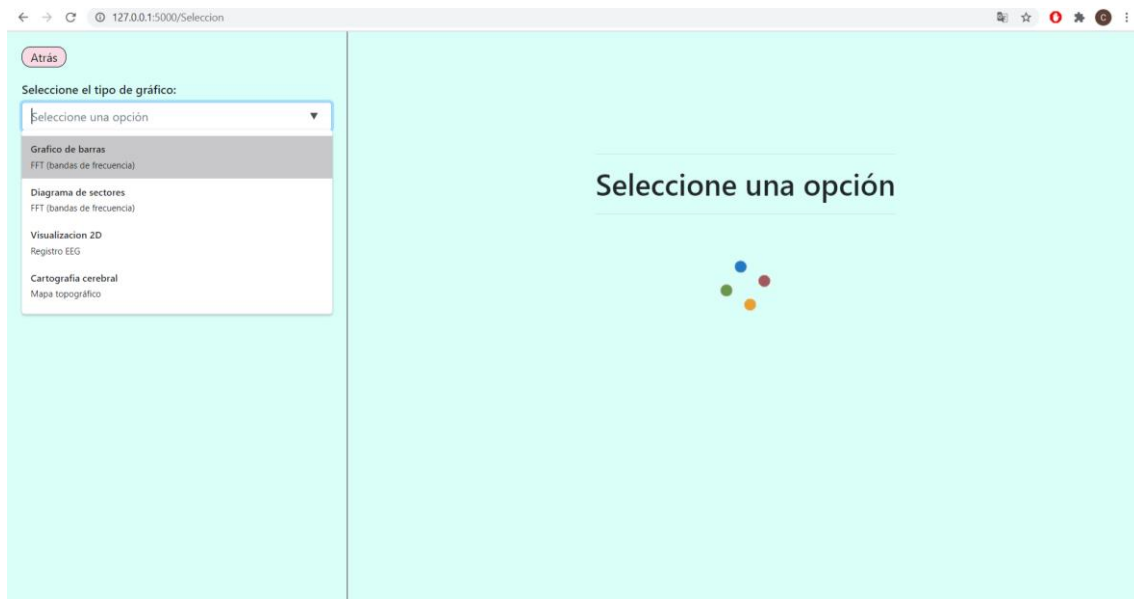


*Ilustración 11: Selección del modo 'Registro de archivo'*

A continuación, en el mando de la izquierda el usuario puede seleccionar el tipo de gráfico que desee, tal y como se observa en la Ilustración 12. Puede escribirlo de forma manual o seleccionar

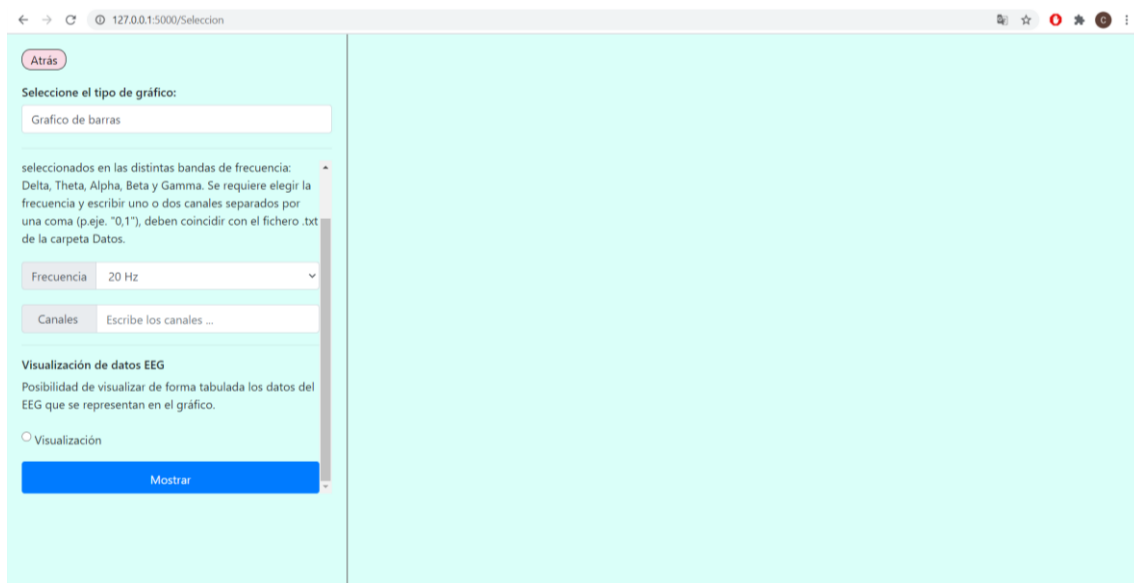
una de las opciones del desplegable. Cuando se sitúa sobre un tipo de gráfico este se sombrea y tras hacer click con el botón izquierdo del ratón se accede a la configuración de dicho gráfico.

Cada tipo de gráfico viene con una pequeña explicación de sus datos o formato.



*Ilustración 12: Página modo archivo*

Así pues, si el usuario selecciona, por ejemplo, el gráfico de barras como en Ilustración 13, se encuentra con una breve descripción del gráfico y los parámetros para configurar.



*Ilustración 13: Configuración del gráfico de barras, modo archivo*

Estos parámetros son la frecuencia y los canales del registro que se quiere cargar para ver la representación de sus datos, en ese caso en un gráfico de barras. La selección de la frecuencia se realiza con un desplegable y la de los canales se introduce de forma manual separando los números por comas. En la Ilustración 14 se encuentra un ejemplo de la configuración de los canales 12 y 13 con una frecuencia de 100Hz.



Frecuencia

20 Hz

20 Hz

50 Hz

100 Hz

250 Hz

500 Hz

Canales

100 Hz

Visualización

Frecuencia

100 Hz

Canales

12,13

Ilustración 14: Parámetros modo archivo. (a) Frecuencia. (b) Canales.

Tras seleccionar los parámetros, el usuario tiene la posibilidad de visualizar los datos de forma tabulada en la parte derecha de la pantalla paralelamente al gráfico en cuestión. Hay un botón de visualización con una breve descripción tal y como se observa en la Ilustración 15.

Visualización de datos EEG

Posibilidad de visualizar de forma tabulada los datos del EEG que se representan en el gráfico.

☒ Visualización

Mostrar

Ilustración 15: Visualización de datos, modo archivo

Los datos quedan representados de forma horizontal tal y como se ve en la Ilustración 16, se dispone de un slider para poder ver todos los datos.

Canales	Dato 1	Dato 2	Dato 3	Dato 4	Dato 5	Dato 6	Dato 7	Dato 8	Dato 9	Dato 10	Dato 11	Dato 12
Canal 12	-25977	-11142	13688	22912	4496	-8835	-30850	-23904	9345	19584	-18553	24512
Canal 13	-2945	-26241	-1665	11904	12928	-19841	-3676	-7560	-25717	-15747	-10625	-13125

Ilustración 16: Datos en forma tabulada, modo archivo

A la vez que al usuario le aparecen los datos de forma tabulada se lanza el gráfico de barras, su formato se observa en la Ilustración 17. Se indica el nombre de los canales representados encima de su correspondiente gráfico y cada barra representa una de las bandas de frecuencia establecidas, cuyos nombres se indican en la parte inferior de estas.

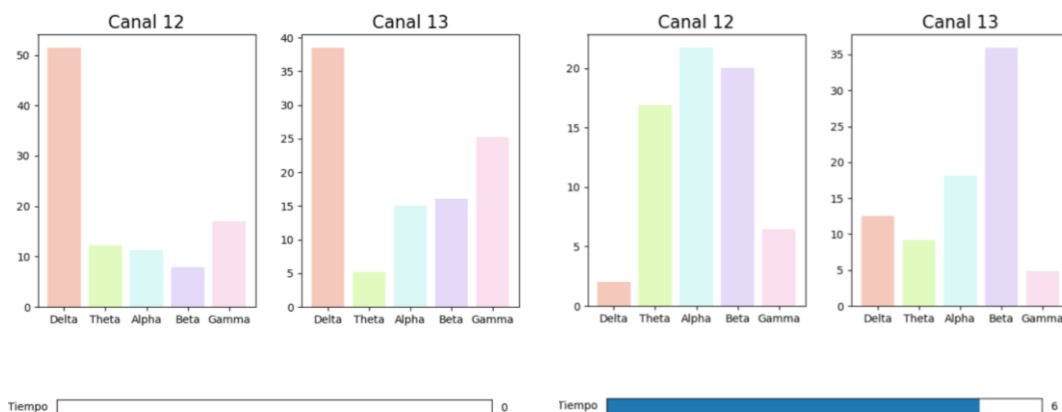


Ilustración 17: Gráfico de barras, modo archivo

En la parte inferior del gráfico encontramos un slider, este le permite al usuario moverse por las distintas actualizaciones del gráfico, a cada una de ellas le corresponden 100 datos. Hay un número al final que indica el número de muestra o actualización que el usuario está viendo. Para poder cambiar la actualización el usuario ha de pulsar y desplazar el ratón por el slider.

El formato para seleccionar la frecuencia y canales es igual para el resto de los gráficos menos para el de cartografía cerebral cuya selección está bloqueada, tal y como se puede ver en la Ilustración 18, ya que se emplean todos los canales.

Ilustración 18: Parámetro canales en cartografía cerebral, modo archivo

Como se ha mencionado en la memoria, en los gráficos relativos a la FFT que son el gráfico de barras y el diagrama de sectores solo se pueden introducir dos canales debido al uso de la aplicación en el ámbito de la educación especial. Mientras que en el gráfico de la visualización 2D se pueden introducir los que se desee.

A continuación, se muestra un ejemplo del resto de los gráficos implementados en este modo.

- **Diagrama de sectores:** Se observa un ejemplo para frecuencia 100 Hz de los canales 22 y 23 en la Ilustración 19. El formato del gráfico es el mismo que el gráfico de barras, con su correspondiente slider, título y leyenda.

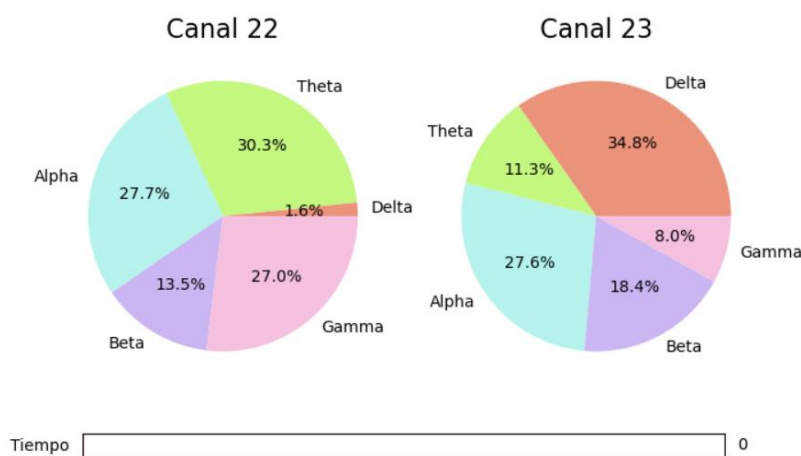


Ilustración 19: Diagrama de sectores modo archivo

- **Visualización 2D:** Se observa un ejemplo de los canales 0,1,2,3,4,5,6,7,8,9,10,11 a la frecuencia de 100Hz en la Ilustración 20. El usuario puede pulsar sobre el contenido de uno de los canales y desplazarse por dentro de los datos.

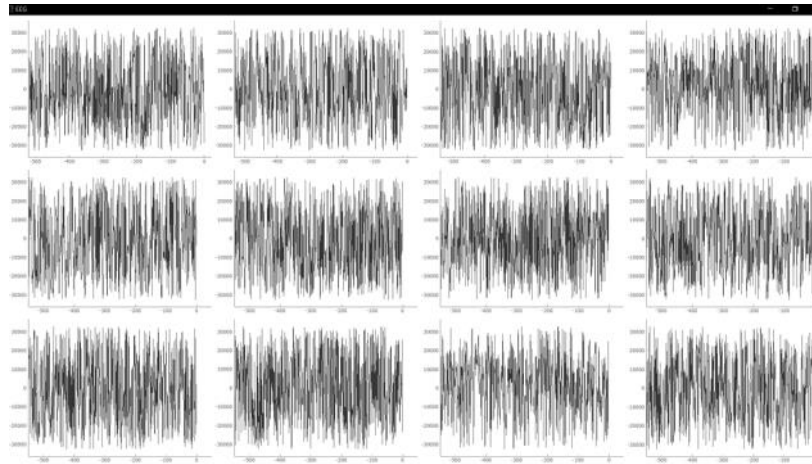


Ilustración 20: Visualización 2D, modo archivo

- **Cartografía cerebral:** Se observa un ejemplo para una frecuencia de 250 Hz de todos los canales en la Ilustración 21. Se observa como en la parte inferior hay un slider que le permite al usuario modificar el tiempo y a la derecha una leyenda donde se muestra que rango de valores corresponde para los valores de los datos. La cartografía se realiza sobre un dibujo de una cabeza para facilitar el análisis al experto.

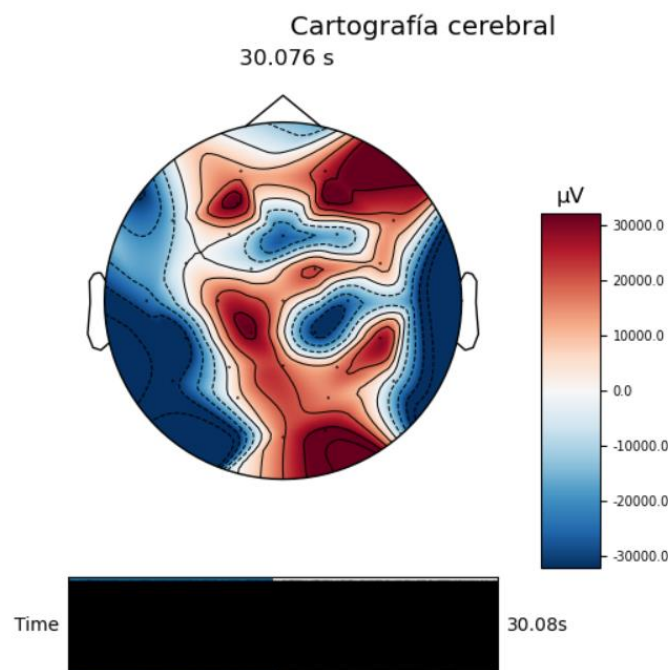


Ilustración 21: Cartografía cerebral, modo archivo

En el modo archivo, se le permite al usuario abrir varios gráficos de forma paralela para un análisis completo de los datos.

## 6.2 Nuevo registro

Este modo se basa en realizar un registro de los datos del EEG en tiempo real, para ello ha de estar correctamente conectado al puerto serie del ordenador, con batería suficiente (o conectado a la red) y encendido. El ordenador detectará automáticamente con que puerto establecer la conexión.

La vista es la misma que la Ilustración 10, con las opciones que se pueden observar en la Ilustración 22. En comparación con el otro modo descrito anteriormente, en ese no se puede realizar una cartografía cerebral, pero añade una opción de guardado de datos.

Seleccione el tipo de gráfico:

Seleccione una opción ▼

- Guardado de datos**  
Guardar los datos en un fichero
- Grafico de barras**  
FFT (bandas de frecuencia)
- Diagrama de sectores**  
FFT (bandas de frecuencia)
- Visualizacion 2D**  
Registro EEG

Ilustración 22: Gráficos disponibles, modo serie

La opción de guardado de datos permite realizar un registro sin visualizar ningún tipo de gráfico, solo hay que seleccionar la frecuencia deseada ya que se guardan todos los canales, el parámetro canales está inhabilitado. Al pulsar sobre mostrar, aparece una ventana emergente en la cual se indica al usuario que para finalizar el registro debe hacer click. Una vez se pulsa, internamente el código está implementado para cerrar el puerto serie de forma segura. Los parámetros y la ventana emergente se pueden observar en la Ilustración 23.

**(a) Parámetros:**

Atrás

Seleccione el tipo de gráfico:

Guardado de datos

**Información y parámetros requeridos**

Permite guardar el registro del EEG, hay que indicar la frecuencia deseada. Guarda todos los canales, cada uno en un fichero con su frecuencia y el número de canal. Para finalizar la captura de datos se debe pulsar el botón.

Frecuencia: 20 Hz

Canales: Escribe los canales ...

Mostrar

**(b) Ventana emergente:**

Ventana para delimitar el tiempo de guardado

Click para finalizar el registro del EEG en tiempo real

Ilustración 23: Guardado de datos, modo serie. (a) Parámetros. (b) Ventana emergente.

La forma de introducir los parámetros de canales y frecuencias para el resto de los gráficos es la misma que en el modo anterior con la diferencia de que en vez de visualizar datos se da la opción de guardarlos, tal y como se muestra en la Ilustración 24.

**Guardado de datos EEG:**

Posibilidad de guardar de forma tabulada el registro del EEG que se representa en el gráfico. Se guardan todos los canales, cada uno en un fichero.

☒ Guardar

Mostrar

Ilustración 24: Opción de guardado de datos en los gráficos, modo serie

Si el usuario selecciona guardar los datos, al acabar el registro, los ficheros de datos correspondientes a una frecuencia serán actualizados. Hay que mencionar que, aunque, la visualización de un gráfico, debido al empleo de esta herramienta en el ámbito educativo especial, sea de dos canales, la opción de guardar los datos guarda los 32 canales para que, si el usuario lo desea, en el modo archivo pueda analizar otros canales u obtener el mapa topográfico con todos ellos.

En este modo, los gráficos se actualizan automáticamente conforme se reciben los datos por parte del EEG, respecto al gráfico de barras y al diagrama de sectores en primera instancia se lanza el gráfico, pero este no tiene valores suficientes por lo que permanece en blanco, tal y como se puede ver en la Ilustración 25, hasta que se actualiza con la primera muestra de 100 datos.

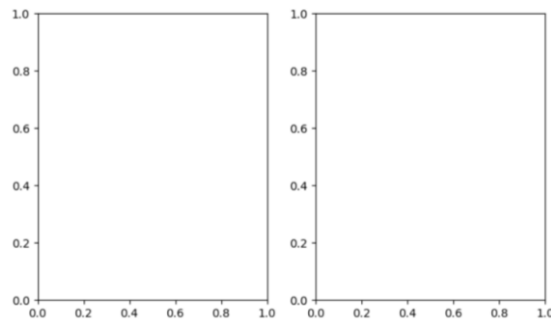


Ilustración 25: Gráfico de barras y diagrama de sectores en blanco, modo serie

A continuación, se muestran los gráficos implementados para este modo.

- **Gráfico de barras y diagrama de sectores:** Para el diagrama de barras, con una frecuencia de 500Hz y los canales 14 y 20 se encuentra el ejemplo de la Ilustración 26 . Para el diagrama de sectores en la Ilustración 27 se encuentra un ejemplo de la representación del canal 0 y 1 a una frecuencia de 100Hz. Estos dos gráficos se actualizan automáticamente conforme se reciben los datos del EEG.

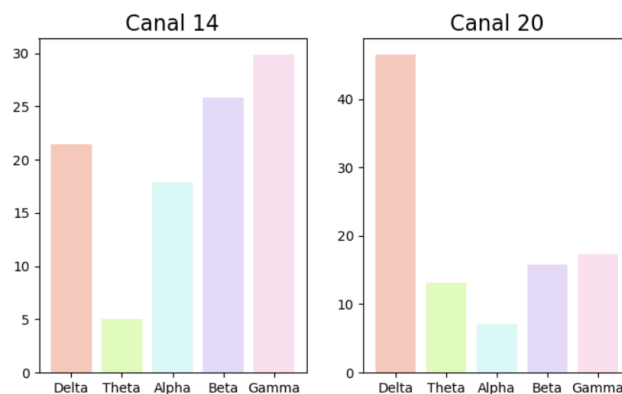


Ilustración 26: Gráfico de barras, modo serie

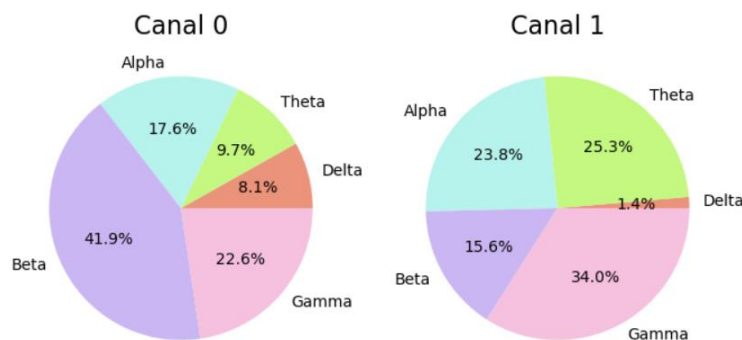


Ilustración 27: Diagrama de sectores, modo serie

- **Visualización 2D:** Se muestra un ejemplo del canal 2 y 3 a una frecuencia de 250Hz en la Ilustración 28. Los datos se actualizan uno a uno en el 0 y se desplazan de izquierda a derecha, el usuario arrastrando los datos puede desplazarlos.

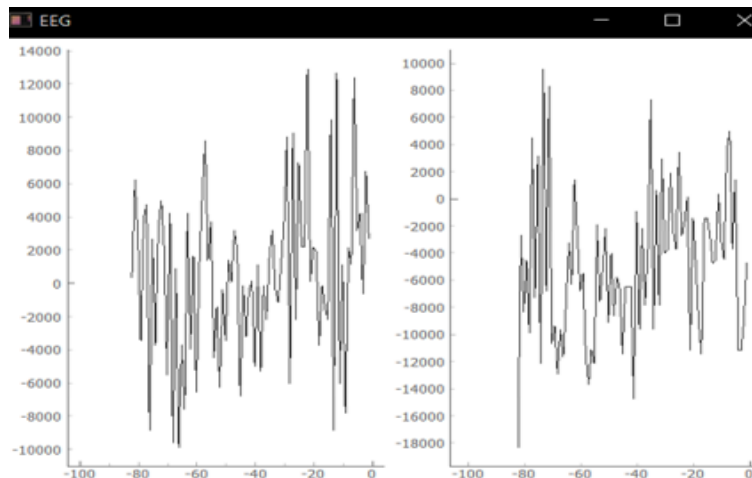


Ilustración 28: Visualización 2D, modo serie

En este modo, no se puede iniciar otro gráfico hasta no haber cerrado el anterior y que, por tanto, el puerto esté cerrado de forma segura.

Para ambos modos, en caso de que el usuario introduzca, más canales de los permitidos, de forma incorrecta o con unos valores no existentes, la aplicación muestra la vista de la Ilustración 29. En esta vista se le permite volver atrás para realizar las modificaciones oportunas.

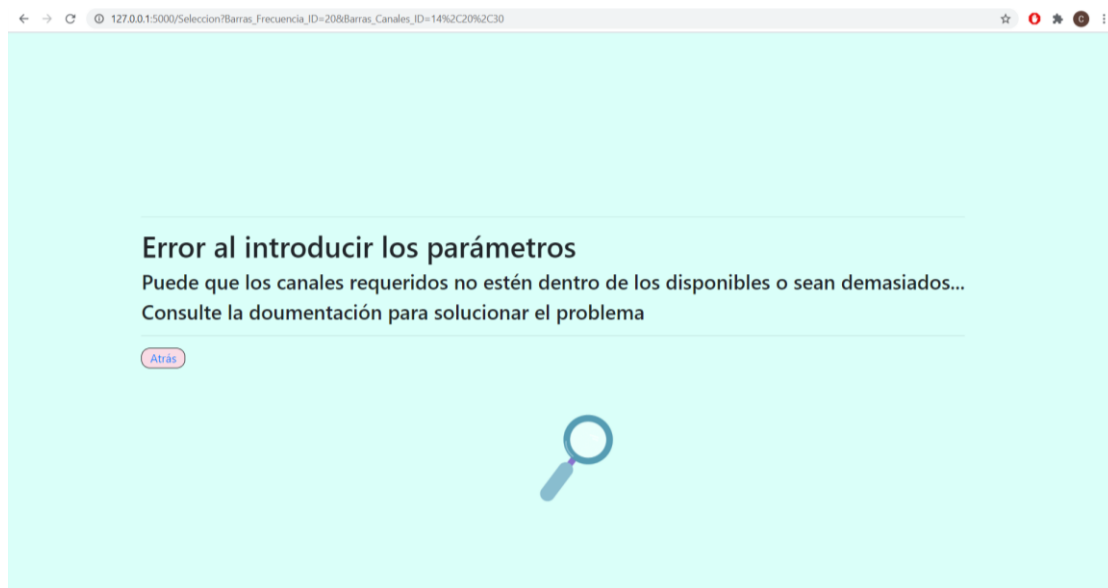


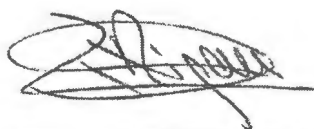

Ilustración 29: Vista error

## 7 Datasheet manual EEG

En ese apartado se adjunta el manual proporcionado por el fabricante sobre las propiedades del EEG y su comunicación.

**COMMUNICATIONS PROTOCOL BETWEEN THE EXIM/EXEA DEVICES AND THE  
 XGPLAB SOFTWARE**

Date	Version	Description	Author
2011/05/05	1.0	Initial Version	Ignacio Del Corral

Revised	Approved
	
Innovation Director	RDI Director
Date: 2011/05/17	Date: 2011/05/17

## The XGPCComm DLL

The communication with the eXim/eXea devices is performed through an USB connection. The interchange of data is made by means of a DLL named XGPCComm.dll, which hides the low level management of the USB driver. This DLL is programmed in C++ and offers high level functions declared in the *CommUSB.h* file, which are used to communicate with the device.

Include the *CommUSB.h* file in your code to access the functions detailed below. The following files should also be added to the include path: *ftd2xx.h*, *timer.h*, *BufferCircular.h*, *CommParalelo.h*, *CommSerie.h*, *CommTriggerExterno.h*, *Comm.h* and *defines.h*.

## Functions exported by the XGPCComm DLL

The DLL supports four different classes. Each one manages a different type of communication.

Class	Description	File
CCommUSB	Used for communicating with USB devices	CCommUSB.h
CCommParalelo	Used for communicating through parallel port	CCommParalelo.h
CCommSerie	Used for communicating through serial port	CCommSerie.h
CCommTriggerExterno	Used for communicating with external devices through serial port	CCommTriggerExterno.h

However, the USB communication is the only one described in this document, since the eXim/eXea devices only support USB ports.

The CCommUSB class exports the following functions:

- CargardriverUSB
- ListarDispositivosConectados
- FIFOVacía
- p\_LeeNibble
- p\_LeeByteDirec
- p\_LeeByte
- MandarInterrup



- InicializarPuerto
- InicializarPuertoPIC
- Conectado
- Desconectar

### **Sending commands to the eXim/ eXea device**

---

All commands are sent to the eXim/eXea thanks to the *CCommUSB::MandarInterrupcion* function, with the following format:

```
int CCommUSB::MandarInterrup(BYTE byTipoInterrupcion, int iNumBytes = -1, BYTE* pbyBytes = NULL, BYTE* pbyBytesEnviados = NULL)
```

- Parameters:
  - o *byTipoInterrupcion*: code for the command to be sent to the device.
  - o *iNumBytes*: number of bytes allocated by the *pbyBytes* parameter. This parameter is optional.
  - o *pbyBytes*: pointer to an array of bytes needed for certain commands. This parameter is optional.
  - o *pbyBytesEnviados*: pointer to an array of bytes with the bytes sent to the device. This parameter is optional.
- Return value: If all bytes were sent to the device. Otherwise the value is 0.

An ACK is sent to the PC for each command received. This ACK is the command code. The device sends additional information for some commands. This information is described below for each command.

The general procedure to send a command to the eXim/eXea device is as follows:

- Create an instance of CCommUSB class.
- Call the CCommUSB::InicializarPuerto function.
- Send the command by means of the CCommUSB::MandarInterrup function.

- Receive and check the ACK and any other additional data sent by the device.

## Available commands

The following commands are available to perform actions with the device. Not all of them are described in this document.

Action	Command
Start a new real time test	0x11
Stop a real time test	0x17
Delete the device's memory	0x22
Start downloading the device's memory	0x33
Configure the device for holter mode	0x67
Get the configuration for holter mode	0x69
Get available memory	0x47
Get total memory	0x41
Start impedance measurement	0x77
Start calibration	0x88

## Data packet structure in Real time tests

In real time tests the eXim/eXea devices send information to the PC in the form of data packets, at a rate of 10 packets per second. The size of these packages varies, depending on the sampling frequency set for each channel, but there is at least one sample of every channel in each packet. Once the sampling frequencies have been set, all packets are the same size and the channel's data arrive always in the same position. Finding data from any channel is thus easy.

The available sampling frequencies for each eXim/eXea are described in the following table:

Model	Available sampling frequencies
-------	--------------------------------

eXim Apnea, eXim Pro	<ul style="list-style-type: none"> <li>▪ Channel 1 to 8 (AC): 20, 50, 100, 250, 500 Hz</li> <li>▪ DC channel 1: 10 Hz</li> <li>▪ DC channel 2: 10 Hz</li> <li>▪ Pulse rate: 10 Hz</li> <li>▪ Oxygen saturation: 10 Hz</li> <li>▪ Light sensor: 10 Hz</li> <li>▪ Event marker: 10 Hz</li> </ul>
eXea PSG 3 Series	<ul style="list-style-type: none"> <li>▪ Channel 1 to 12 (AC): 20, 50, 100, 250, 500 Hz</li> <li>▪ DC channel 1: 10 Hz</li> <li>▪ DC channel 2: 10 Hz</li> <li>▪ Pulse rate: 10 Hz</li> <li>▪ Oxygen saturation: 10 Hz</li> <li>▪ Light sensor: 10 Hz</li> <li>▪ Event marker: 10 Hz</li> </ul>
eXea PSG 4 Series	<ul style="list-style-type: none"> <li>▪ Channel 1 to 16 (AC): 20, 50, 100, 250, 500 Hz</li> <li>▪ DC channel 1: 10 Hz</li> <li>▪ DC channel 2: 10 Hz</li> <li>▪ Pulse rate: 10 Hz</li> <li>▪ Oxygen saturation: 10 Hz</li> <li>▪ Light sensor: 10 Hz</li> <li>▪ Event marker: 10 Hz</li> </ul>
eXea PSG 5 Series	<ul style="list-style-type: none"> <li>▪ Channel 1 to 20 (AC): 20, 50, 100, 250, 500 Hz</li> <li>▪ DC channel 1: 10 Hz</li> <li>▪ DC channel 2: 10 Hz</li> <li>▪ Pulse rate: 10 Hz</li> <li>▪ Oxygen saturation: 10 Hz</li> <li>▪ Light sensor: 10 Hz</li> <li>▪ Event marker: 10 Hz</li> </ul>
eXea Ultra	<ul style="list-style-type: none"> <li>▪ Channel 1 to 32 (AC): 20, 50, 100, 250, 500 Hz</li> <li>▪ DC channel 1: 10 Hz</li> <li>▪ DC channel 2: 10 Hz</li> <li>▪ Pulse rate: 10 Hz</li> <li>▪ Oxygen saturation: 10 Hz</li> <li>▪ Light sensor: 10 Hz</li> <li>▪ Event marker: 10 Hz</li> </ul>

The size of samples coming from the event marker and the light sensor is 1 byte, while those coming from any other channel are 2 byte wide.

The data packet follows this schema:

Position	Value
0	0xFD (header)
1	0x03 (header)
2	Event marker (not used) (1 byte per sample)
3	Light sensor (1 byte per sample)
4	DC channel 1 (2 bytes per sample)
6	DC channel 2 (2 bytes per sample)
8	Pulse rate (2 bytes per sample)
10	Oxygen saturation (2 bytes per sample)
12 - ...	AC channels (2 bytes per sample) (see how data is organized below)

Since 10 packets per second are sent to the PC, each packet stores a number of samples for each channel equal to the channel sampling frequency divided by 10. The package is split in as many divisions as samples are present in the channel with the highest sampling rate (referred to as *reference channel*). The number of divisions that must be skipped to find the next sample of any channel is calculated by dividing the number of samples of the channel by the number of samples of the reference channel.

For example, if we have an exim Pro (8 AC channels) in which the first channel is set to 500 Hz and all other channels are set to 100 Hz we would have:

Channel	Samples per packet	# divisions until next sample
1	50	1
2	10	5
3	10	5

Channel	Samples per packet	# divisions until next sample
4	10	5
5	10	5
6	10	5
7	10	5
8	10	5

Now, we have to figure out the order in which the samples come from the device. First, a sample of each channel is sent. Next, we check (in the natural order of the channels) if a sample of the channel should arrive or not for each division. In the case of extio Pro, the sequence of samples of the 8 AC channels would be as follows (the first channel is 0):

0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 0,  
0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5,  
6, 7, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 0, 0, 0, 1, 2,  
3, 4, 5, 6, 7, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 0, 0, 0, 0

If the 8 AC channels were all set to 100 Hz the order would be easier:

0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0,  
1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1,  
2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7

## Starting a new real time test

The command 0x11 instructs the device to start sending data to the PC.

- Command: 0x11
- Number of bytes:  $(\text{number of AC channels} / 4) + (2 * \text{number of channels}) + \text{number of channels} + 2 + 2$
- Additional bytes: the following bytes should be sent in the specified order:
  - o Channel configurations: N bytes set to 0, where

$N = \text{number of AC channels} / 4$

o Sampling rate: 2 bytes each in big endian format, in this order:

- AC channels
- Pulse rate
- Oxygen saturation
- DC channel 1
- DC channel 2
- Light sensor
- Event marker

o Channel modules (1 byte for each channel):

Channel module = 500 / sampling rate.

o Ratio index (2 bytes in big endian format).

Ration index = maximum sampling rate / minimum sampling rate

o Packet size (2 bytes in big endian format)

Packet size = 2 + a + 10

where

$a = 2 * \sum(\text{sampling rate of each channel} / 10)$

For instance, the configuration bytes for an exim Pro where all AC channels are set to 100 Hz, would be:

Channel configurations	0x00, 0x00
Sampling frequencies	0x6400, 0x6400, 0x6400, 0x6400, 0x6400, 0x6400, 0x6400, 0x6400, 0x0A00, 0x0A00, 0x0A00, 0x0A00, 0x0A00, 0x0A00
Channel modules	0x05, 0x05, 0x05, 0x05, 0x05, 0x05, 0x05, 0x05, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32
Ratio index	0x0A00
Packet size	0xAC00

If we change set first AC channel to 500 Hz, then the configuration bytes would be:

Channel configurations	0x00, 0x00
Sampling frequencies	0xF401, 0x6400, 0x6400, 0x6400, 0x6400, 0x6400, 0x6400, 0x6400, 0x0A00, 0x0A00, 0x0A00, 0x0A00, 0x0A00, 0x0A00
Channel modules	0x01, 0x05, 0x05, 0x05, 0x05, 0x05, 0x05, 0x05, 0x32, 0x32, 0x32, 0x32, 0x32, 0x32
Ratio index	0x3200
Packet size	0xFC00

After sending this command, the device returns three bytes: an ACK (the command previously sent: 0x11), 0xFD and another byte with variable value (0x02 for eXim Apnea and eXim Pro, 0x03 for eXea PSG 3 Series, 0x04 for eXea PSG 4 Series, 0x05 for eXea PSG 5 series and 0x08 for eXea Ultra).

### Example

```
int iRes = 0;

// Create an instance of CCommUSB class
CCommUSB *pComm;
pComm = new CCommUSB;

// Call the CCommUSB::InicializarPuerto function, passing the USB device number (usually 0),
// the maximum number of errors before read/write functions fails, and the device's serial
// number.
pComm->InicializarPuerto(
    0, // USB device number (0 if only one eXim/eXea device is connected to
    // the PC)
    550, // do not change this value
    "345-011"); // device's serial number, depends on each particular device
                // (check the device's label)

// Create the structure with all the bytes needed to setup the test.
BYTE abyParametrosComunicacion[MAX_LONG_COMANDO_USB];
// Populate the array with the configuration data as described previously
// ...

// Send the command to the device
if (pComm->MandarInterrupto(
    0x11, // command
    iNumBytes, // number of bytes sent along with the command
    abyParametrosComunicacion // bytes sent along with the command
) != 0)
{
    // Look for the ACK
    ...
}

// Now start processing data received by successive calls pComm->p_LeeByteDirec
// ...
```

### Receiving data from the device

After the "com\_directa" command has been sent to the device and three ACK bytes are read, the eXim/eXea will start sending data packets to the PC. The PC does not send any ACK to the device. Instead, it should read all data received or the device will stop sending data after a certain time-out (the device's FIFO buffer cannot hold any more data and the device's status LEDs turns to red color).

Reception of data via USB is managed by the XGPCComm DLL and stored in an internal FIFO buffer. The next byte can be read by calling to the *CCommUSB::p\_LeeByteDirec* function. This function should be called repeatedly in order to retrieve data sent by the device:

```
unsigned char CCommUSB::p_LeeByteDirec(unsigned char &ucDato);
```

- Parameters:
  - o *ucDato*: the variable where the byte will be stored.
- Return value: 1 if any byte could we read, 0 otherwise.

Existence of data can be checked by calling the *CCommUSB::FIFOVacia* function:

```
BOOL FIFOVacia();
```

- Return value: TRUE if any data is available in the buffer, FALSE otherwise.

### **Ending a real time test**

---

Send the 0x17 command with no arguments to instruct the device to stop sending data to the PC. After receiving this command the device sends 0x17 as ACK to the PC. However, after sending the command some data might still be available in the input queue, so make sure you read first those bytes before searching for the ACK.



## 8 Bibliografía

- [1] «Pyqtgraph,» [En línea]. Available: <http://www.pyqtgraph.org/>. [Último acceso: 28 Julio 2020].
- [2] «MNE,» [En línea]. Available: <https://mne.tools/stable/index.html>. [Último acceso: 15 Agosto 2020].
- [3] «Pyserial,» [En línea]. Available: <https://pyserial.readthedocs.io/en/latest/index.html>. [Último acceso: 13 Abril 2020].
- [4] «Threading,» [En línea]. Available: <https://docs.python.org/3/library/threading.html>. [Último acceso: 24 Septiembre 2020].
- [5] «Numpy,» [En línea]. Available: <https://numpy.org/>. [Último acceso: 12 Mayo 2020].
- [6] «Matplotlib,» [En línea]. Available: <https://matplotlib.org/>. [Último acceso: 12 Junio 2020].
- [7] «Flask,» [En línea]. Available: <https://openwebinars.net/blog/que-es-flask/>. [Último acceso: Agosto 25 2020].
- [8] «Tkinter,» [En línea]. Available: <https://docs.python.org/3/library/tkinter.html>. [Último acceso: 23 Septiembre 2020].

## 9 Índice de ilustraciones

Ilustración 1: Onda resultante .....	11
Ilustración 2: Distribución de frecuencias.....	11
Ilustración 3: Distribución en las diferentes bandas de frecuencia .....	12
Ilustración 4: Diagrama de flujo del modo serie, guardado de datos .....	13
Ilustración 5: Diagrama de flujo del modo serie, gráfico de barras y diagrama de sectores .....	14
Ilustración 6: Diagrama de flujo del modo serie, gráfico visualización 2D.....	15
Ilustración 7: Diagrama de flujo del modo archivo, gráfico de barras y diagrama de sectores...	16
Ilustración 8: Diagrama de flujo del modo archivo, cartografía cerebral y visualización 2D.....	17
Ilustración 9: Página principal de la aplicación.....	20
Ilustración 10: Página selección de la aplicación.....	21
Ilustración 11: Selección del modo 'Registro de archivo' .....	21
Ilustración 12: Página modo archivo.....	22
Ilustración 13: Configuración del gráfico de barras, modo archivo.....	22
Ilustración 14: Parámetros modo archivo. (a) Frecuencia. (b) Canales. ....	23
Ilustración 15: Visualización de datos, modo archivo .....	23
Ilustración 16: Datos en forma tabulada, modo archivo .....	23
Ilustración 17: Gráfico de barras, modo archivo.....	23
Ilustración 18: Parámetro canales en cartografía cerebral, modo archivo.....	24
Ilustración 19: Diagrama de sectores modo archivo .....	24
Ilustración 20: Visualización 2D, modo archivo.....	25
Ilustración 21: Cartografía cerebral, modo archivo.....	25

Ilustración 22: Gráficos disponibles, modo serie .....	26
Ilustración 23: Guardado de datos, modo serie. (a) Parámetros. (b) Ventana emergente. ....	26
Ilustración 24: Opción de guardado de datos en los gráficos, modo serie .....	26
Ilustración 25: Gráfico de barras y diagrama de sectores en blanco, modo serie.....	27
Ilustración 26: Gráfico de barras, modo serie .....	27
Ilustración 27: Diagrama de sectores, modo serie.....	27
Ilustración 28: Visualización 2D, modo serie .....	28
Ilustración 29: Vista error .....	28

## 10 Índice de tablas

Tabla 1: Comando 'Empezar una prueba en tiempo real', 20Hz,32 canales.....	4
Tabla 2: Comando 'Empezar una prueba en tiempo real', 50Hz,32 canales.....	5
Tabla 3: Comando 'Empezar una prueba en tiempo real', 100Hz,32 canales.....	7
Tabla 4: Comando 'Empezar una prueba en tiempo real', 250 Hz,32 canales.....	8
Tabla 5: Comando 'Empezar una prueba en tiempo real', 500 Hz,32 canales.....	9
Tabla 6: Orden de instalación de los drivers .....	18
Tabla 7: Requisitos de la aplicación.....	19

## 11 Índice de ecuaciones

Ecuación I.....	3
Ecuación II .....	3
Ecuación III.....	3
Ecuación IV.....	4
Ecuación V .....	4
Ecuación VI.....	4
Ecuación VII .....	4
Ecuación VIII .....	5
Ecuación IX.....	5
Ecuación X.....	5
Ecuación XI.....	6
Ecuación XII .....	6
Ecuación XIII.....	6
Ecuación XIV .....	6
Ecuación XV .....	7
Ecuación XVI.....	7
Ecuación XVII .....	7
Ecuación XVIII .....	8
Ecuación XIX.....	8
Ecuación XX.....	9
Ecuación XXI.....	9
Ecuación XXII .....	9
Ecuación XXIII .....	10
Ecuación XXIV .....	10
Ecuación XXV .....	10
Ecuación XXVI.....	10
Ecuación XXVII.....	11