

# MANUAL PARA LA CONFIGURACIÓN Y USO DEL SISTEMA DE ADQUISICIÓN DE SEÑALES EEG

Equipo amplificador de señales biomédicas

G.USBamp RESEARCH



**g<sup>®</sup>·USBamp**  
USB BIOSIGNAL AMPLIFIER



**AUTORES:**

Luis Ricardo Ochoa Ochoa

Juan Andrés Morocho Aguila

## Índice de contenido

<b>Índice de contenido.....</b>	<b>ii</b>
Índice de tablas: .....	4
Índice de figuras: .....	5
<b>1. Introducción a g.USBamp RESEARCH .....</b>	<b>7</b>
<b>2. Características generales del amplificador .....</b>	<b>8</b>
<b>3. Hardware del equipo g.USBamp.....</b>	<b>9</b>
3.1. Componentes del Equipo g.USBamp .....	9
3.2. Características físicas del amplificador g.USBamp .....	10
3.2.1. Grupo De Conectores De Electrodo (Vista Frontal) .....	10
3.2.2. Conectores De Alimentación y Conexión USB (Vista Trasera) .....	10
<b>4. Conexiones del equipo g.USBamp.....</b>	<b>11</b>
4.1. Sincronización de equipos g.USBamp para 16 canales .....	12
4.2. Apagado y almacenamiento de g.USBamp .....	13
<b>5. Software del equipo g.USBamp.....</b>	<b>14</b>
5.1. g.tec Suite 2020.....	14
5.1.1. Instalación y Activación del producto. ....	14
5.1.2. Descripción de las aplicaciones .....	19
<b>6. Ejecutado g.RECORDER .....</b>	<b>20</b>
6.1. Visualización de señal .....	21
6.2. Configuración de filtros .....	22
6.3. Elementos de g.RECORDER .....	24
6.3.1. Barra de estado .....	24
6.3.2. Barra de herramientas .....	24
<b>7. Matlab API .....</b>	<b>26</b>
7.1. Instalación de Matlab.....	26
7.1.1. Pasos Para Instalar MATLAB.....	27
7.2. g.NEEDAccess Matlab API.....	29
7.3. Pruebas de funcionamiento del equipo g.USBamp con MATLAB.....	29
7.4. Código de Adquisición de datos para 32 canales-Guía del programador.....	33
7.4.1. Configuración de la ventana de visualización Scope .....	33
7.4.2. Función gtecDeviceInterface .....	34
7.4.3. Configuración de IP y los puertos para el host y el cliente .....	34
7.4.4. Dispositivos conectados .....	35

7.4.5. Canales disponibles .....	36
7.4.6. Configuración de canales, tierra, referencia y parámetros de grabación.....	36
7.4.7. Inicio de Adquisición de datos.....	39
7.4.8. Función para detener la grabación.....	40
7.4.9. Filtrado de datos .....	41
7.5. Funciones adicionales para g.USBamp .....	42
7.6. Código completo.....	44
<b>8. Advertencias .....</b>	<b>49</b>
<b>9. Seguridad y Mantenimiento .....</b>	<b>50</b>
<b>10. Especificaciones Técnicas .....</b>	<b>51</b>
<b>11. Referencias .....</b>	<b>52</b>

## **Índice de tablas:**

<b>Tabla 1.</b> Requerimientos de Instalación.....	14
<b>Tabla 2.</b> Características de la Barra de herramientas de g.RECORDER .....	24
<b>Tabla 3.</b> Requerimientos mínimos y recomendados para instalación y ejecución de MATLAB. 26	
<b>Tabla 4.</b> Funciones específicas de Matlab para adquirir datos con el equipo g.USBamp.....	43
<b>Tabla 5.</b> Advertencias sobre el equipo g.USBamp.....	49
<b>Tabla 6.</b> Información para la seguridad y mantenimiento del equipo.....	50

## Índice de figuras:

<b>Figura 1.</b> Componentes del amplificador de señales g.USBamp de g.tec.....	9
<b>Figura 2.</b> Grupo de canales para conectores de electrodos.....	10
<b>Figura 3.</b> Conectores de alimentación, sincronización y conexiones USB ( vista trasera) .....	11
<b>Figura 4.</b> Conexión de dos dispositivos g.USBamp (32 canales) .....	13
<b>Figura 5.</b> Instalación finalizada del software g.tec Suite 2020.....	16
<b>Figura 6.</b> Activación de licencia g.tec Suite 2020.....	16
<b>Figura 7.</b> Interfaz de aplicaciones g.tec Suite 2020 .....	17
<b>Figura 8.</b> Opción para actualizaciones de software g.tec Suite 2020.....	17
<b>Figura 9.</b> Interfaz del Software completo una vez terminada la instalación en el computador...	18
<b>Figura 10.</b> Verificación del estado de instalación de los requerimientos necesarios System Diagnosis Tool. ....	19
<b>Figura 11.</b> Ventana principal al iniciar la aplicación g.RECORDER .....	20
<b>Figura 12.</b> Selección del equipo disponible g.USBamp .....	21
<b>Figura 13.</b> Ejecutando g.RECORDER con un dispositivo de 16 canales .....	22
<b>Figura 14.</b> Ventana de configuración de canales del equipo y configuración de filtros para los 16 canales .....	23
<b>Figura 15.</b> Visualización de las señales con aplicación de filtros .....	23
<b>Figura 16.</b> Elementos de la barra de estado de g.RECORDER .....	24
<b>Figura 17.</b> Barra superior de herramientas de g.RECORDER .....	24
<b>Figura 18.</b> Sustitución del nombre Polyspace por MATLAB previo a la instalación del programa .....	27
<b>Figura 19.</b> Desactivar la función MATLAB SERVER previo a la instalación de MATLAB.....	28
<b>Figura 20.</b> Carpeta de MATLAB en archivos del programa.....	28
<b>Figura 21.</b> Aplicación gNEEDAccess Matlab API. ....	29
<b>Figura 22.</b> Ejemplos incluidos en paquete de g.tec dentro de g.NEEDaccessMATLABAPI .....	30
<b>Figura 23.</b> Set Path en Matlab.....	30
<b>Figura 24.</b> Opción Add with Subfolders para agregar las funciones a la carpeta de MATLAB .	31
<b>Figura 25.</b> Carpeta con las funciones necesarias para ejecutar en MATLAB .....	31
<b>Figura 26.</b> Añadir ruta: C:\Usuarios\<nombre_de_usuario>\Documentos\gtec .....	32
<b>Figura 27.</b> Carpetas agregadas en MATLAB .....	32
<b>Figura 28.</b> Conexión de tierra y referencia comunes para los 16 canales.....	37
<b>Figura 29.</b> Filtros disponibles para utilizarlos en la configuración de adquisición de señales. a) Filtro pasa banda, b) Filtro notch.....	38

**Figura 30.** Especificaciones Técnicas del equipo amplificador de señales g.USBamp.....51

## **1. Introducción a g.USBamp RESEARCH**

El equipo g.USBamp de la compañía austríaca g.tec (2023) (medical engineering) es un amplificador de bioseñales de alta gamma que tiene la capacidad de adquirir, medir, recopilar y amplificar señales de tipo EEG, ECG y EMG con una alta precisión y rendimiento utilizado para la investigación neuropsicológica y neurofisiológica, las ciencias de la vida y la biorretroalimentación, la neuro retroalimentación y la investigación de la interfaz cerebro-computadora (BCI).

Al utilizar el g.USBamp Research, los investigadores pueden obtener datos detallados y precisos sobre las actividades eléctricas del cuerpo humano en diversos contextos y situaciones lo que brinda una oportunidad única para el estudio y la comprensión de la actividad cerebral, muscular y cardíaca en áreas como la neurociencia, la psicología, la medicina y la rehabilitación.

Además, el equipo ofrece herramientas de hardware y software para manipular y configurar el dispositivo. Por un lado, se encuentra la aplicación exclusiva desarrollada por g.tec, la cual proporciona una interfaz intuitiva y funcional para visualizar y analizar las señales adquiridas y que ofrece diversas funciones y herramientas diseñadas específicamente para maximizar la utilidad del equipo y facilitar el trabajo de los investigadores.

En las secciones siguientes de este manual, se presentarán las instrucciones paso a paso para la configuración inicial, el uso básico y las funciones avanzadas del g.USBamp Research. También se abordarán aspectos importantes como la seguridad, el mantenimiento y la solución de problemas comunes.

## **2. Características generales del amplificador**

Las características de los amplificadores se describen a continuación:

- Grabación de señales a través de USB
- 16 entradas analógicas con 24 Bits y frecuencia de muestreo de hasta 38.400 Hz por canal
- Filtrado digital de los datos de bioseñal
- Conexión directa a un PC o portátil
- Muestreo simultáneo de 16 canales
- Conexión directa de electrodos con conector de seguridad estándar
- Capacidad de sincronizar múltiples dispositivos
- ID Equipo 1: UR-2020.04.01
- ID Equipo 2: UR-2020.04.02



### 3. Hardware del equipo g.USBamp

El equipo amplificador de señales físico se lo evidencia Figura 1, el cual se encuentra dentro del laboratorio I2TEC y se cuenta con dos ejemplares de 16 canales cada uno, para las investigaciones presentes y futuras.

#### 3.1. Componentes del Equipo g.USBamp

En la **Figura 1** se muestran los componentes necesarios para el funcionamiento equipo de adquisición de señales, los cuales, reposan en el centro de investigación I2TEC de la Facultad de la Energía las Industrias y los Recursos Naturales no Renovables (FEIRNNR).

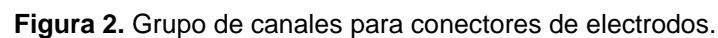
- g.USBamp – Amplificador de bioseñal USB
- GlobTek GTM21097-3005: unidad de fuente de alimentación médica
- Cable de alimentación
- Cable USB
- Conectores Push-Pull
- Cable SYNC para sincronización de equipos



**Figura 1.** Componentes del amplificador de señales g.USBamp de g.tec.

El amplificador consta de dos secciones. En la parte frontal se localizan las entradas para los grupos de electrodos de adquisición de señales, que constan de un total de 16 canales, junto con las conexiones para las tomas de tierra y de referencia. Por otro lado, la sección trasera está dedicada a las entradas necesarias para la alimentación y la conexión al ordenador para el registro de datos.

El amplificador tiene 4 grupos (GRUPOS A, B, C, D) de 4 canales de entrada cada uno, lo que resulta un total de 16 canales. Cada grupo está formado por un conector de tierra y de referencia, mismos que se los puede utilizar para cada grupo de electrodos de manera independiente, sin embargo, también se puede configurar una sola referencia y tierra para los 4 grupos. De la misma manera, se cuenta con una entrada de conector push-pull que es otra alternativa de conectar electrodos utilizando un casco de electrodos. Los conectores de electrodos, puesta a tierra y de referencia están diferenciados son un color específico dentro del grupo de conectores de la forma en que se lo evidencia en la **Figura 2**.



### 3.2.2. Conectores De Alimentación y Conexión USB (Vista Trasera)

- POWER: Botón encendido/apagado
- POWER SUPPLY: Conector cable de alimentación
- FUSE: Localización de fusible
- SYNC IN/OUT: Para la sincronización con otro g.USBamp
- DIG I/O 1-2: Entradas y salidas digitales

- SC: Para desconectar los electrodos de entrada de las unidades de amplificación
- USB: Para conectar al computador



**Figura 3.** Conectores de alimentación, sincronización y conexiones USB ( vista trasera)

#### 4. Conexiones del equipo g.USBamp

Para empezar la utilización del equipo es necesario seguir los pasos descritos a continuación:

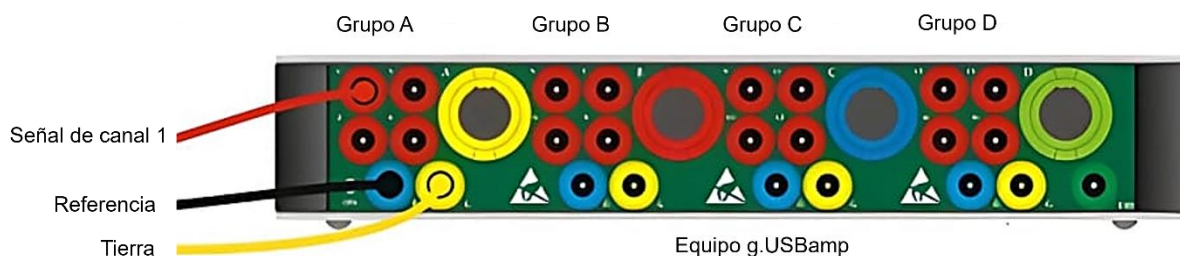
1. Conectar el equipo a su alimentación (110 V) tanto al cable de alimentación como al GlobTek GTM21097-3005.



2. Se conecta el cable USB en el puerto del equipo g.USBamp y en el puerto de la computadora.



3. Se debe colocar los electrodos en los canales 1-16 de cada grupo correspondiente con sus respectivas referencias y tierras.



4. Encender el equipo con el interruptor ON/OFF (El led verde indica si el equipo está correctamente encendido).
5. Por último, verificar que la computadora haya detectado el dispositivo USB desde la opción Administrador de dispositivos>USB.



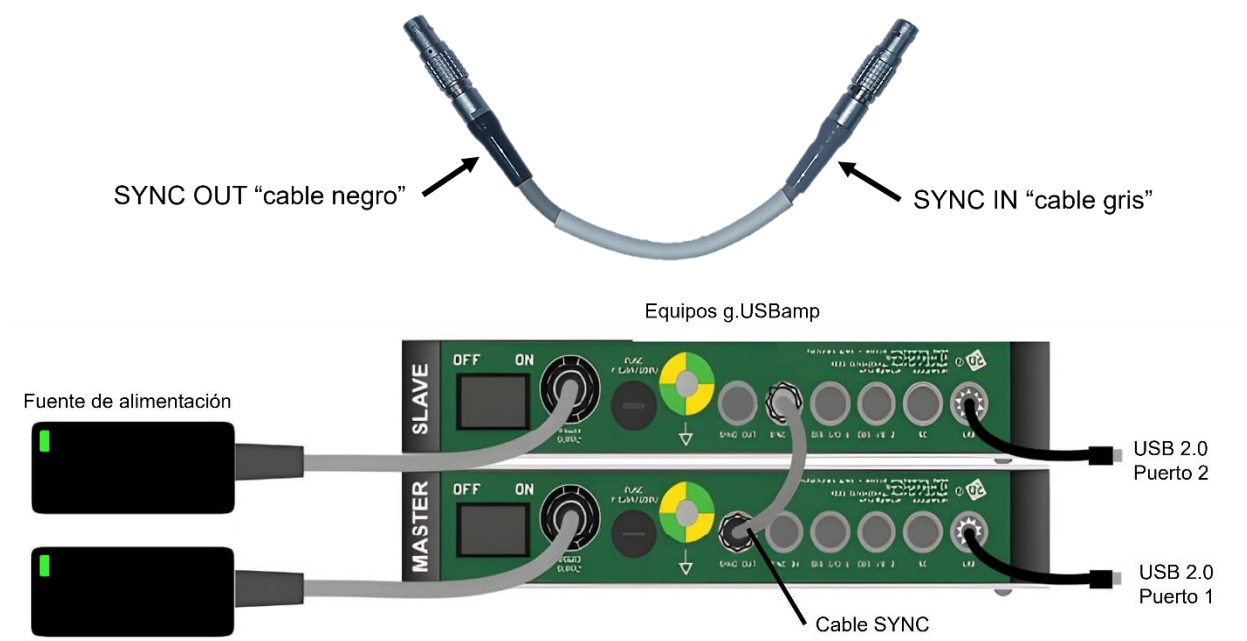
Asegúrese que se realizaron todas las conexiones necesarias para el registro antes de encender el dispositivo y evite apagar el equipo durante la grabación de señales. Una interrupción de grabación representa un error de adquisición.

#### 4.1. Sincronización de equipos g.USBamp para 16 canales

El g.USBamp cuenta con la impresionante capacidad de sincronizarse con hasta 4 dispositivos simultáneamente, lo que permite manejar 16, 32, 48 o incluso 64 canales de adquisición. Esta sincronización se logra estableciendo un dispositivo MAESTRO y conectando los demás dispositivos como ESCLAVOS mediante un cable de sincronización que la propia compañía proporciona en el paquete de compra. Esta funcionalidad brinda a los usuarios una poderosa solución para ampliar y mejorar la captura de datos de bioseñales,

facilitando investigaciones y aplicaciones que requieran un mayor número de canales para obtener resultados más completos y precisos.

Para sincronizar dos equipos g.tec, es esencial comenzar por determinar qué equipo actuará como maestro y cuál será el esclavo, ya que, esto influye en la conexión física de los dos dispositivos. En la **Figura 4** se presenta el diagrama de conexión de los dos dispositivos siendo el equipo inferior el MASTER y el superior el ESCLAVO. El cable de sincronización se conecta a la entrada (SYNC OUT) para establecer el dispositivo (MASTER), mientras que la entrada (SYNC IN) establece el segundo dispositivo como (SLAVE).



**Figura 4.** Conexión de dos dispositivos g.USBamp (32 canales)

El ejemplo gUSBampAPISynchDemo.m permite realizar pruebas con dos dispositivos a la vez, obteniendo un total de canales con un tiempo de grabación de 10 segundos.

#### 4.2. Apagado y almacenamiento de g.USBamp

Siga los siguientes pasos para apagar y almacenar g.USBamp correctamente:

1. Apague el dispositivo con el interruptor de la parte trasera (posición del interruptor OFF). El LED verde está apagado.
2. Desconecte todos los electrodos.
3. Desconecte el cable de alimentación.
4. Desconecte el cable USB.
5. Desconecte el cable de alimentación de la fuente de alimentación médica.

## 5. Software del equipo g.USBamp

### 5.1. g.tec Suite 2020

G.tec Suite 2020 contiene la biblioteca de software para adquisición de datos y procesamiento en tiempo real con g.USBamp, g.Hlamp y g.Nautilus. Consta de los paquetes g.Hlsys, g.Recorder, g.NEEDaccess y g.BSanalyze. Además, permite controlar el g.Estim FES y g.Estim PRO.

La activación y sincronización de la aplicación requiere una licencia que se incluye en el paquete de adquisición física del equipo, donde se otorgan las credenciales para hacer uso de la aplicación. La instalación está basada en tres fases comunes las cuales son: Descarga,

#### 5.1.1. Instalación y Activación del producto.

1. Verificar Requerimientos De Instalación: Para el correcto funcionamiento del software es necesario tener en cuenta los requerimientos mencionados de la **Tabla 1**.

**Tabla 1.**  
*Requerimientos de Instalación.*

Características	Requerimientos
CPU	Procesador de 2,6 GHz o más rápido
Resolución de pantalla	1920 x 1080 píxeles (Recomendado)
RAM	> = 8 GB
Puerto USB 2.0	Un puerto USB libre para cada dispositivo que se conectará
Disco Duro	> =100GB
Versión Windows	Windows 10, Windows 11
Microsoft Visual Studio	Actualización 3 de Visual Studio 2015
MATLAB	Lanzamiento R2020a
Simulink	Lanzamiento R2020a



---

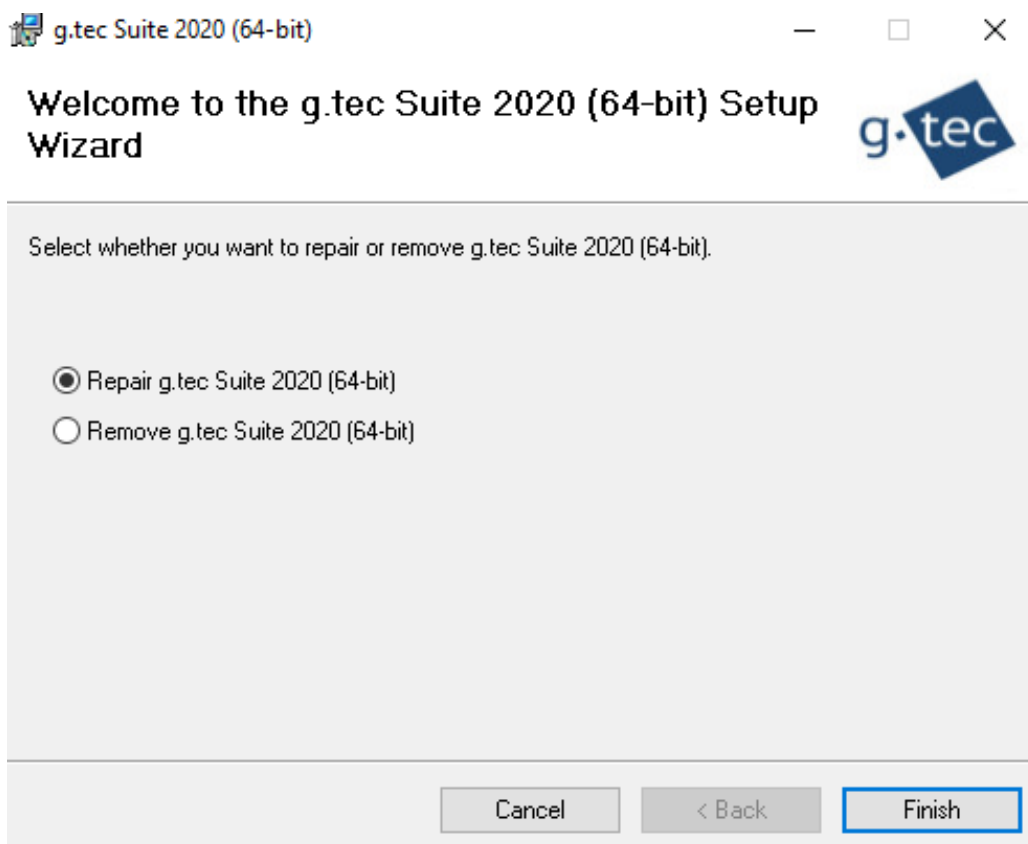
Python	$\geq 3.6$
Numpy	$\geq 1.13.3$
Matplotlib	$\geq 2.1.0$
Cffi	$\geq 1.11.2$
Pywin32	$\geq 223$

---

Nota: Estos son los requerimientos recomendados para la ejecución del software, considerando que el equipo amplificador almacena una gran cantidad de muestras

---

2. **Descarga g.tec Suite2020:** El archivo .exe para su descarga está incluido en el paquete de compra con el equipo g.USBamp, sumado a las credenciales de activación del software, sin embargo, también se la puede encontrar en la página principal de g.tec en dentro de su lista de productos que ofrece la compañía.
3. **Instalación:** El proceso de instalación es sencillo, ya que la aplicación, en primera instancia, no consume muchos recursos. Cuando se descargue el archivo, se ejecutará como administrador en la cual se deberá aceptar los términos y condiciones y seguir las instrucciones para completar el proceso. Una vez finalizada la instalación, se mostrará una confirmación, similar a lo que se muestra en la **Figura 5**.

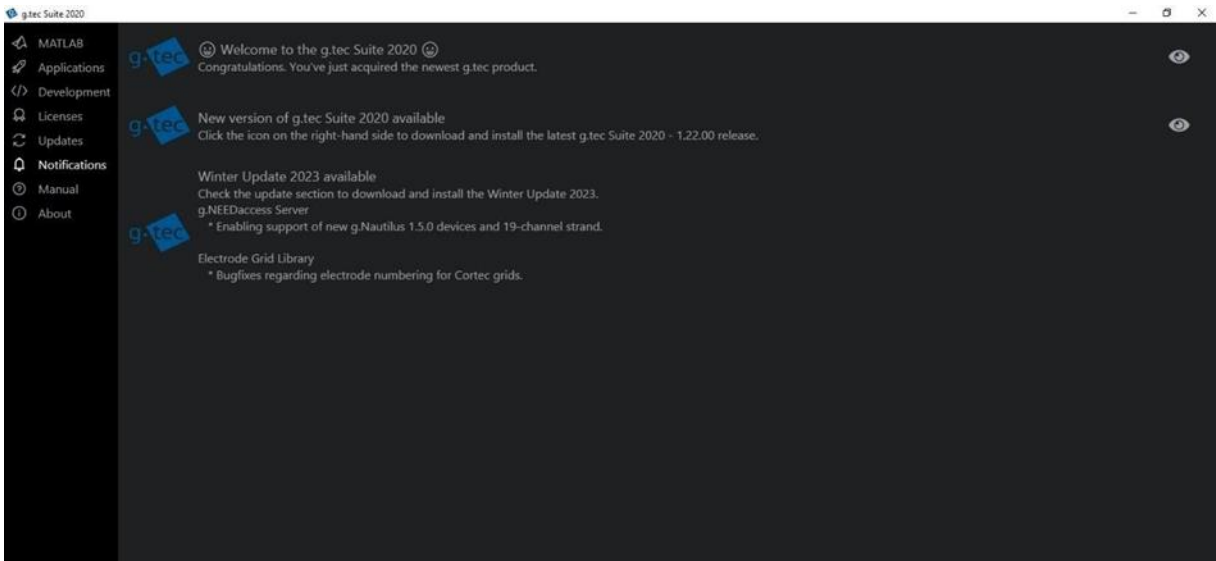


**Figura 5.** Instalación finalizada del software g.tec Suite 2020

- 4. Activación De Licencia:** Se deberá incluir las credenciales que se otorgaron al momento de la adquisición del producto en los espacios de la **Figura 6** y una vez completado la activación, la interfaz de la aplicación se muestra como en la **Figura 7**.

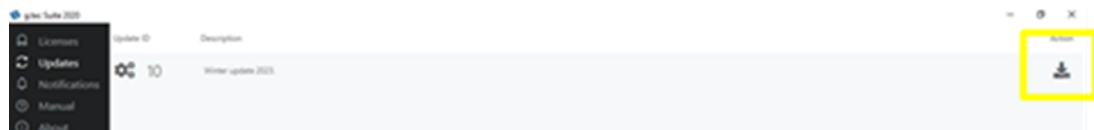
**Figura 6.** Activación de licencia g.tec Suite 2020





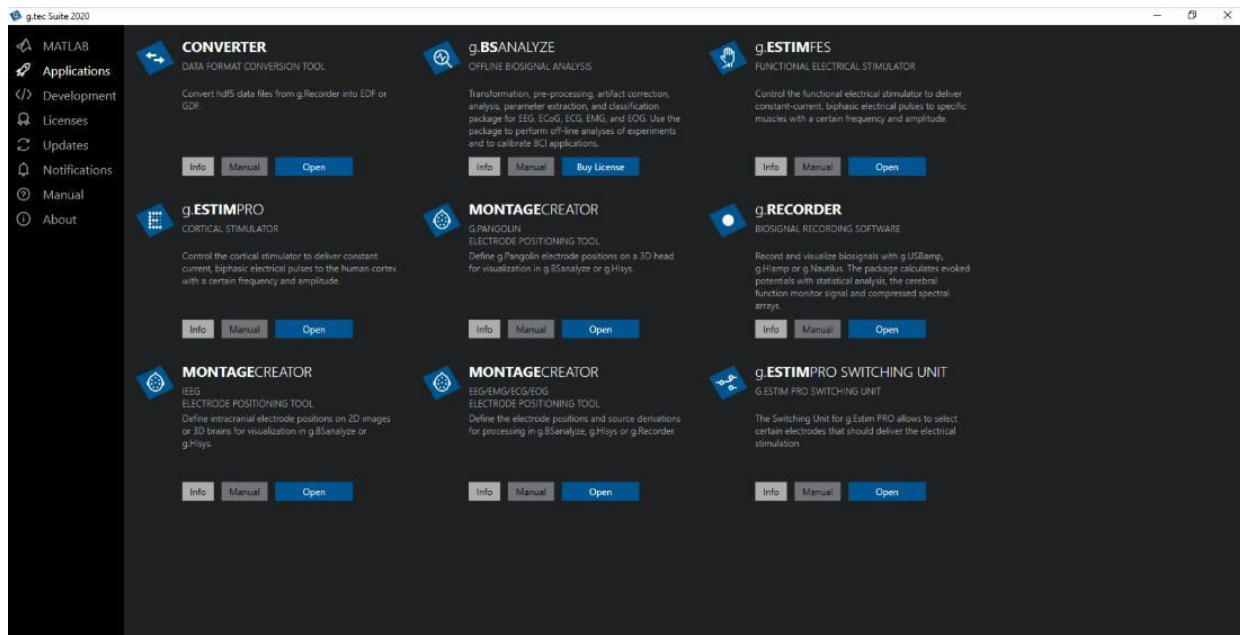
**Figura 7.** Interfaz de aplicaciones g.tec Suite 2020

5. Descarga e Instalación del paquete completo contenido en g.tec Suite 2020: Para tener acceso a todos los paquetes disponibles en la aplicación, se debe descargar la última actualización a través del icono de la pestaña "UPDATES" (ver **Figura 8**) y luego ejecutar e instalar dicha actualización. Así, poder aprovechar todas las características y mejoras que ofrece.



**Figura 8.** Opción para actualizaciones de software g.tec Suite 2020

El software completo se muestra en la **Figura 9**, donde se pueden observar las diversas opciones disponibles para ejecutar con el amplificador de señales. Estas aplicaciones incluyen una guía que ayudará en el uso y configuración del dispositivo, considerando que cada una está destinada a realizar una tarea específica.

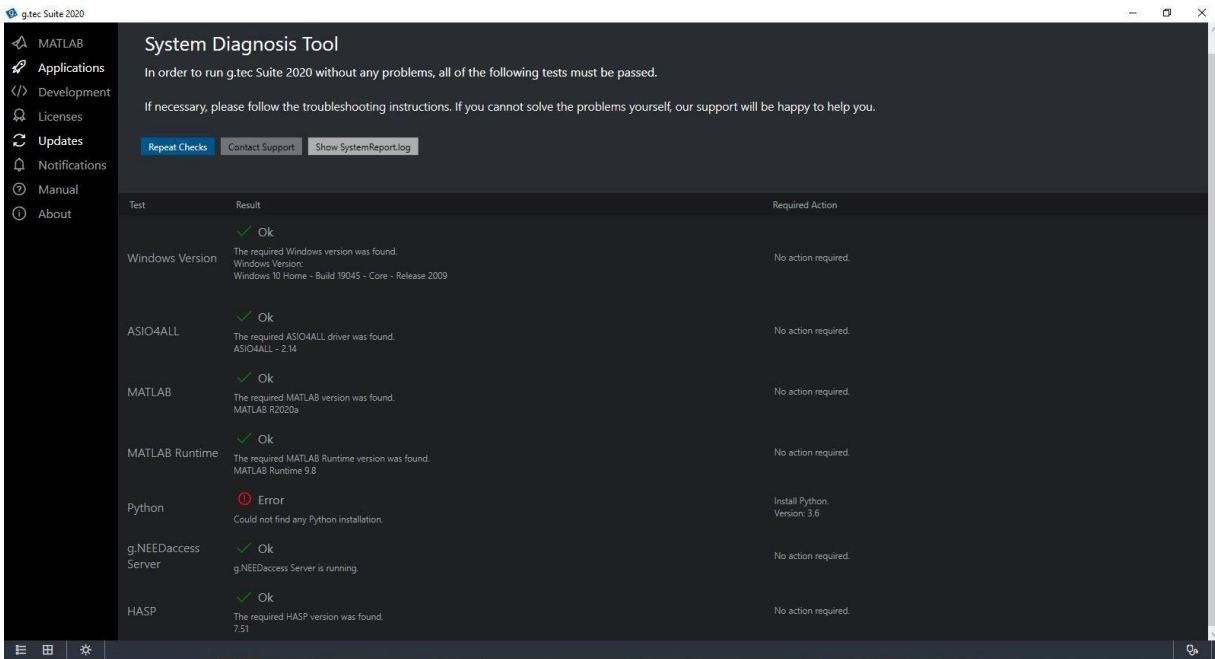


**Figura 9.** Interfaz del Software completo una vez terminada la instalación en el computador

**6. Instalar Drivers y Complementos:** Para usar el equipo correctamente, hay que instalar los controladores y aplicaciones complementarias incluidos en el paquete de instalación del software proporcionado por el fabricante, esenciales para garantizar su funcionamiento. Estos drivers se mencionan a continuación:

- ASIO4ALL\_2\_14\_English.exe
- HASPUserSetup.exe
- MATLAB\_Runtime\_R2020a\_Update\_4\_win64

La correcta instalación de cada uno de los requerimientos necesarios se lo puede verificar en la pestaña Sistem Diagnosis Tool como se lo evidencia en la **Figura 10** donde los drivers y aplicaciones están instalados de forma correcta.



**Figura 10.** Verificación del estado de instalación de los requerimientos necesarios System Diagnosis Tool.  
**Nota:** En este caso, debido a que se eligió la herramienta de MATLAB como lenguaje de programación para adquirir señales EEG y aplicar el algoritmo de presentación de estímulos visuales, no se ha procedido a instalar Python.

### 5.1.2. Descripción de las aplicaciones

**g.BSANALYZE:** Se trata de un software de procesamiento de señales biológicas que se utiliza fuera de línea y se puede ejecutar tanto en MATLAB como de forma independiente. Este software ofrece funciones esenciales como transformación, preprocesamiento, eliminación de artefactos, análisis y clasificación, a las que se accede mediante interfaces gráficas de usuario o mediante la línea de comandos de MATLAB.

**g.ESTIMFES:** Es un estimulador eléctrico funcional programable (FES por sus siglas en inglés) que permite una estimulación bipolar (ánodo y cátodo) mediante electrodos. El dispositivo es controlado por una computadora vía USB.

**g.ESTIMPRO:** Es un estimulador de CC bifásico diseñado para estimular las neuronas durante el mapeo cerebral activo (USB) antes de la resección cortical adyacente a la corteza inferior.

**g.Pangolin MONTAGECREATOR:** Permite simular y definir posiciones en 3D de los electrodos EEG para su posterior visualización en cualquiera de las aplicaciones antes mencionadas.

**g.RECORDER:** Es una herramienta de visualización, almacenamiento y registro de señales compatible con todos los equipos amplificadores (g.tec). Es una interfaz que permite una fácil configuración y puesta en marcha una vez conectado al computador.



De manera particular, para iniciar validando el funcionamiento del equipo se puede hacer uso de la aplicación *g.RECORDER* debido a la facilidad de configuración y ejecución del software.

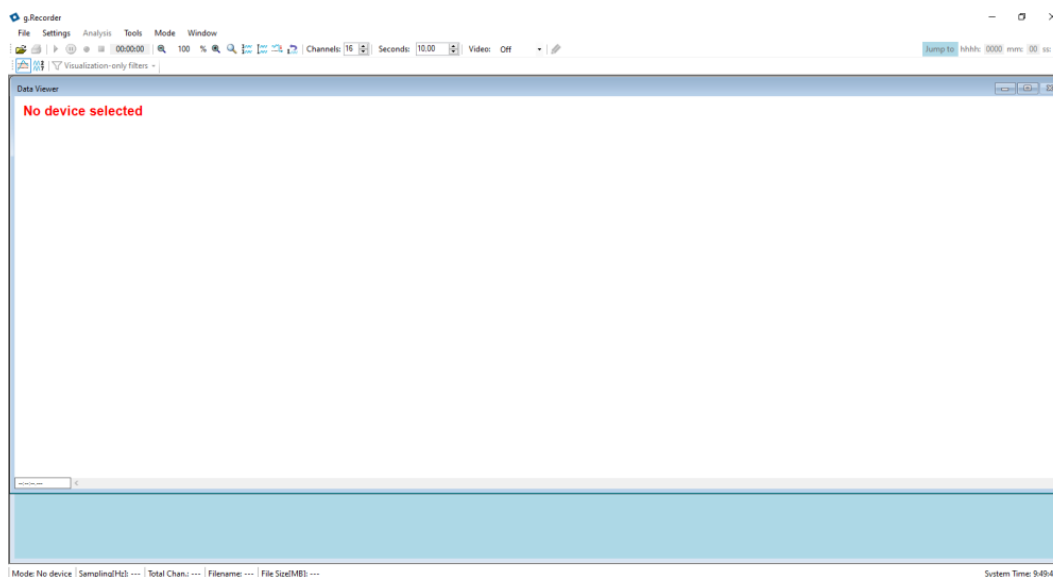
---

## 6. Ejecutado g.RECORDER

Para la utilización de la herramienta seguir los pasos que se enlistan a continuación:

### 1. Abrir g.RECORDER

Ingresar al ícono de g.RECORDER en el listado de aplicaciones en g.tec Suite2020, la misma que se inicia en modo de usuario (**Figura 11**). Este modo es el modo estándar y normalmente g.tec proporciona una versión demo de la aplicación, lo que es posible visualizar las señales, pero no es posible grabar desde este medio ya que, para hacerlo, es necesario la versión de paga.



**Figura 11.** Ventana principal al iniciar la aplicación g.RECORDER

### 2. Encender el dispositivo

Proceder a encender g.USBamp, conectar directamente el equipo vía USB para verificar el estado del equipo. En este punto, una vez conectado al computador, la aplicación identifica la presencia de un dispositivo, que deberá seleccionarse y ponerse en marcha.

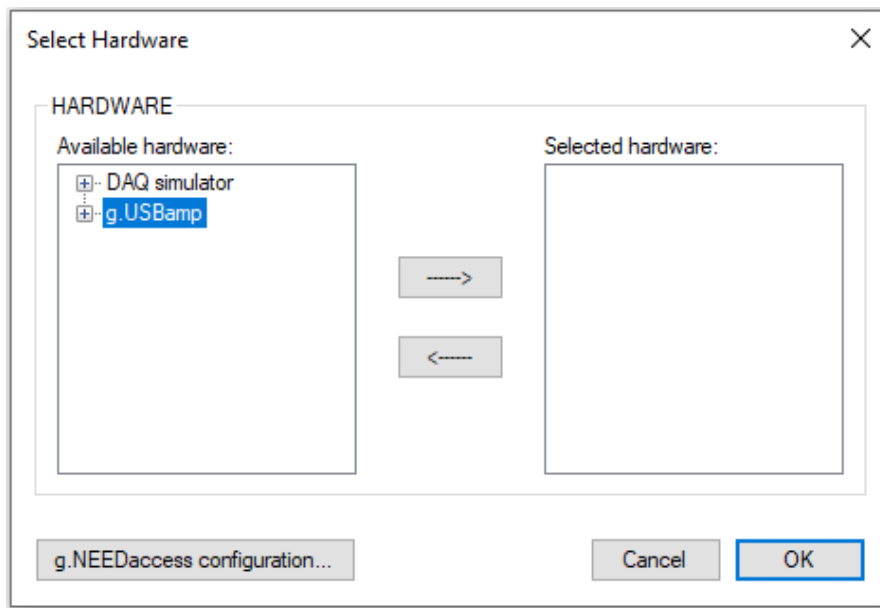
### 3. Seleccionar dispositivo

Para seleccionar el dispositivo ir a Settings> Select Hardware donde se procederá a escanear los dispositivos disponibles que hayan sido conectados al computador.



Nota: En caso de que no se identifique ningún dispositivo verificar el administrador de dispositivos del PC, verificar la actualización de los drivers; si el problema persiste, verificar el estado del equipo amplificador.

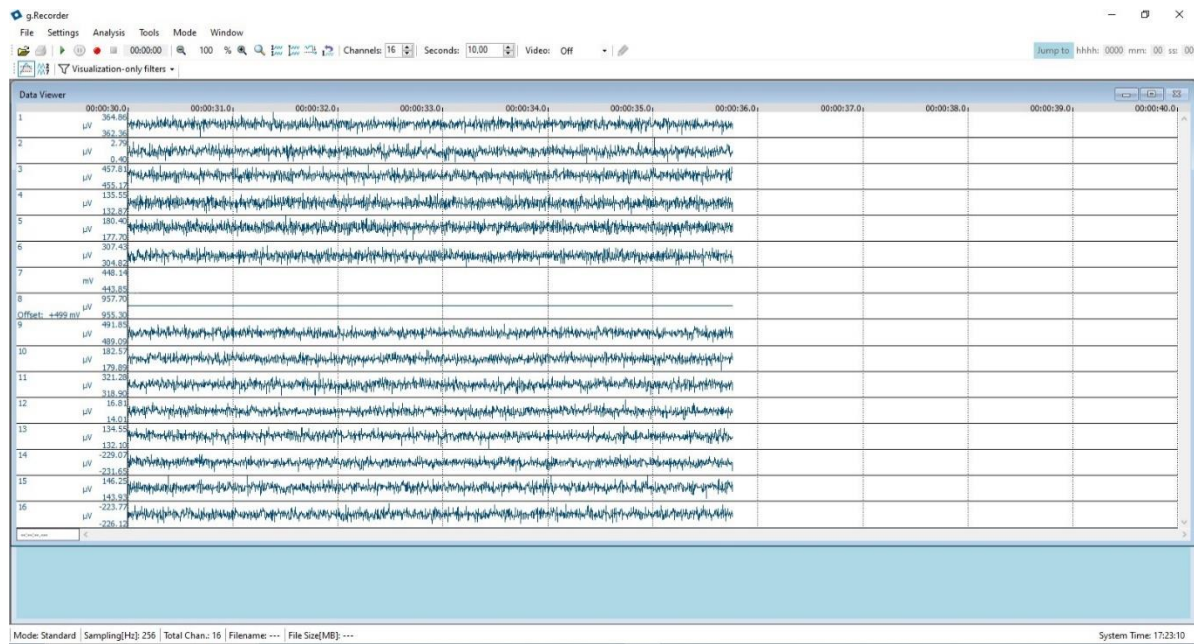
4. Seleccionar el dispositivo de la izquierda, moverlo a la parte derecha de la ventana () y clic en OK (ver Figura 12).



**Figura 12.** Selección del equipo disponible g.USBamp

### 6.1. Visualización de señal

La opción Start data viewing permite iniciar la visualización en tiempo real de las señales amplificadas por el dispositivo. En la **Figura 13** se visualizan las señales obtenidas del amplificador. Debido a que g.USBamp es un equipo amplificador y con alta sensibilidad, se recomienda trabajar en un entorno libre de ruidos o situaciones que puedan causar algún tipo de interferencia.



**Figura 13.** Ejecutando g.RECORDER con un dispositivo de 16 canales

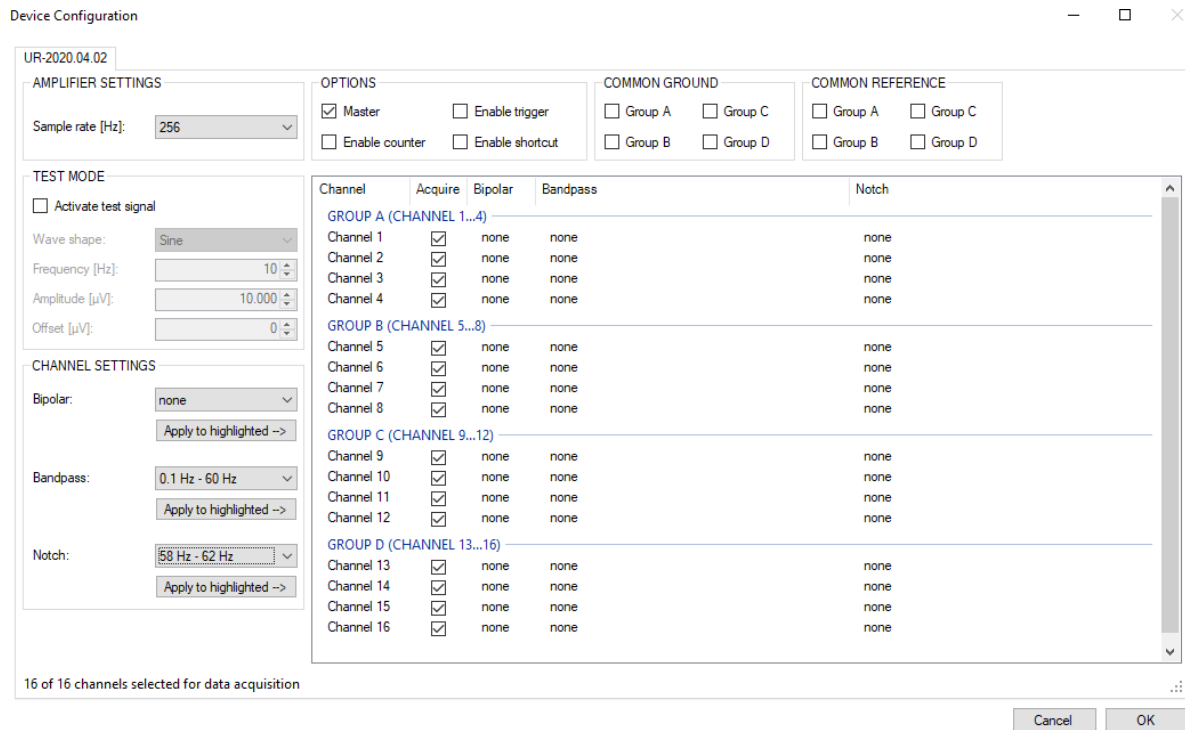
## 6.2. Configuración de filtros

La configuración de filtros para cada canal se lo lleva a cabo mediante la ventana de configuración de canal. Para esto, dar clic en Settings>channels. Se abrirá una ventana en la cual, se visualiza, en la parte izquierda las opciones de configuración y en la parte derecha los canales del amplificador.

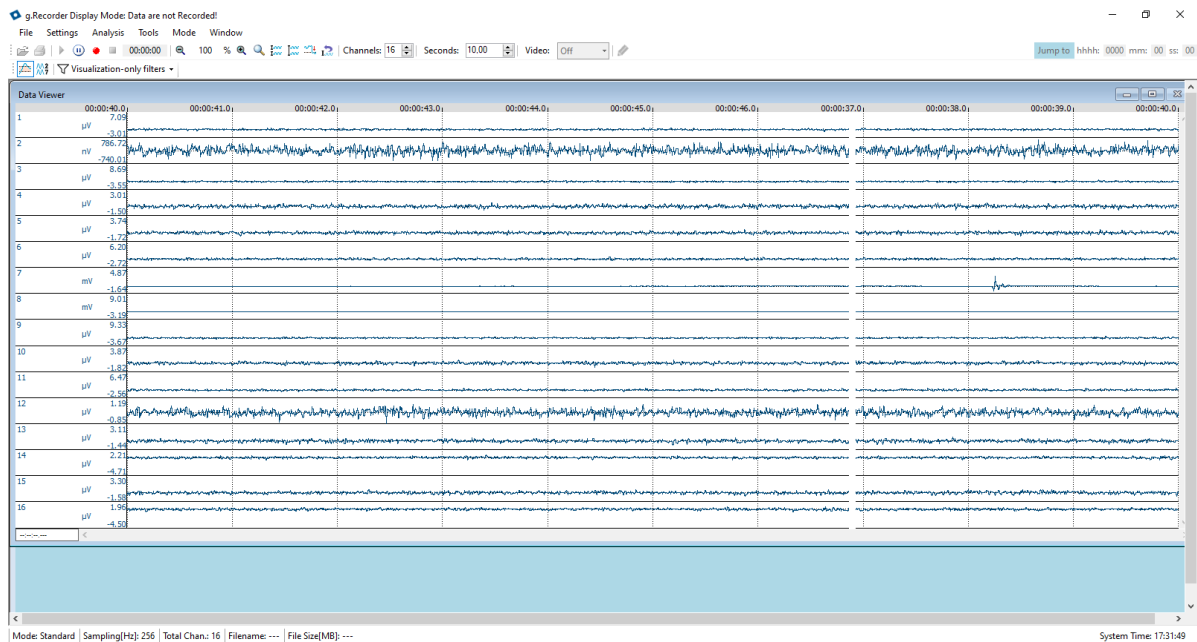
En esta ventana es necesario:

- Insertar el número de muestras en Hz (por defecto está configurado con 256 muestras)
- Seleccionar el orden del filtro bandpass
- Seleccionar el orden del filtro notch

En la **Figura 14** se visualiza con los filtros ya seleccionados y aplicados a los 16 canales y en la **Figura 15** se visualiza las señales una vez aplicado los filtros.



**Figura 14.** Ventana de configuración de canales del equipo y configuración de filtros para los 16 canales



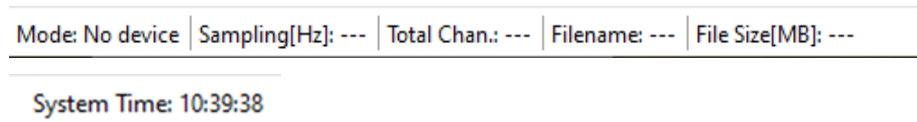
**Figura 15.** Visualización de las señales con aplicación de filtros

### 6.3. Elementos de g.RECORDER

#### 6.3.1. Barra de estado

La barra de estado de g.RECORDER de la **Figura 16** se ubica en la parte inferior de la ventana principal. Contiene configuración de los siguientes parámetros:

- Frecuencia de muestreo por canal en hercios [Hz],
- Número total de canales,
- Nombre de archivo seleccionado
- Tamaño real del archivo en megabytes [MB]
- Hora real del sistema en el formato hh:mm:ss AM/PM.



**Figura 16.** Elementos de la barra de estado de g.RECORDER

#### 6.3.2. Barra de herramientas



La barra de herramientas de se localiza en la parte superior de la ventana del menú e incluye funciones para abrir archivos, imprimir, iniciar/detener la adquisición de datos y la captura de vídeo, establecer marcadores y navegar hasta un instante determinado de los datos. En la **Figura 17** se muestra los elementos dentro de la interfaz de la aplicación mientras que la **Tabla 2** presenta los elementos y funciones de la barra de herramientas.






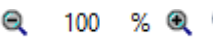
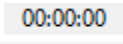







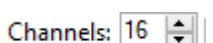
**Figura 17.** Barra superior de herramientas de g.RECORDER

**Tabla 2.**

*Características de la Barra de herramientas de g.RECORDER*

Herramienta	Función
Open	Abrir/cargar archivo
	Imprimir
	Iniciar visualización de datos



	Pausar visualización de datos
	Iniciar la adquisición de datos en el disco duro
	Detener la adquisición de datos
	Alejar y acercar
	Tiempo de grabación actual
	<p>Escala automática</p> <p>Si la escala automática está activada, aparece un pequeño icono en el área de información del canal, a la izquierda de la ventana del Visor de datos, y los datos se reescalan para cada nueva página de datos.</p>
	Auto escala máxima individual.
	Auto escala todos los canales individualmente a su valor mínimo y máximo de la secuencia de datos visualizada (más 10%). Esta escala sólo se aplica una vez por clic.
	Auto escala máxima global Auto escala todos los canales al valor global mínimo y máximo de la secuencia de datos mostrada (más 10%). Esta escala se aplica sólo una vez por clic.
	Escala automática de desplazamiento Auto escala cada canal a su offset individual, preservando su rango de amplitud previamente configurado. Esta escala aplica una vez por clic.
	Restaurar la escala definida por el usuario
	Recorte de señales
	Mostrar/ocultar la visualización del valor numérico de la señal.
	Selección de número de canales de visualización.

---

Seconds: 10.00

Tiempo en segundos que se mostrarán en la vista de datos.

Video: Off

Encender/Apagar la cámara web

Jump to hhhh: 0000

Saltar a un punto específico en el tiempo de grabación

---

## 7. Matlab API

La empresa g.tec ofrece la posibilidad de utilizar el equipo amplificador de señales a través de la aplicación de MATLAB/SIMULINK, lo cual brinda a los programadores una gran oportunidad de generar su propio medio de adquisición. Con la aplicación de MATLAB, los programadores pueden acceder a varias funciones y herramientas para procesar y analizar señales capturadas, dando una solución completa y potente para las necesidades, ya que esta API está integrada en el paquete de herramientas de MATLAB (MathWorks, 2024).

### 7.1. Instalación de Matlab

El equipo amplificador de señales es compatible con la versión R2020a de MATLAB disponible en la página oficial de MathWorks para su descarga. La **Tabla 3** especifica los requerimientos mínimos del sistema para poder ejecutar MATLAB en la aplicación.

**Tabla 3.**  
*Requerimientos mínimos y recomendados para instalación y ejecución de MATLAB.*

Mínimo		Recomendado
Procesador	Intel o AMD x86-64	Intel o AMD x86-64 con cuatronúcleos lógicos y compatibilidad con el conjunto de instrucciones AVX2
Disco	5-8 GB	Se recomienda un SSD
RAM	4 GB	8 GB
Gráficos	No se requiere una tarjeta gráfica específica	Se recomienda una tarjetagráfica acelerada por hardware compatible con OpenGL 3.3 con memoria GPU de 1 GB

### 7.1.1. Pasos Para Instalar MATLAB

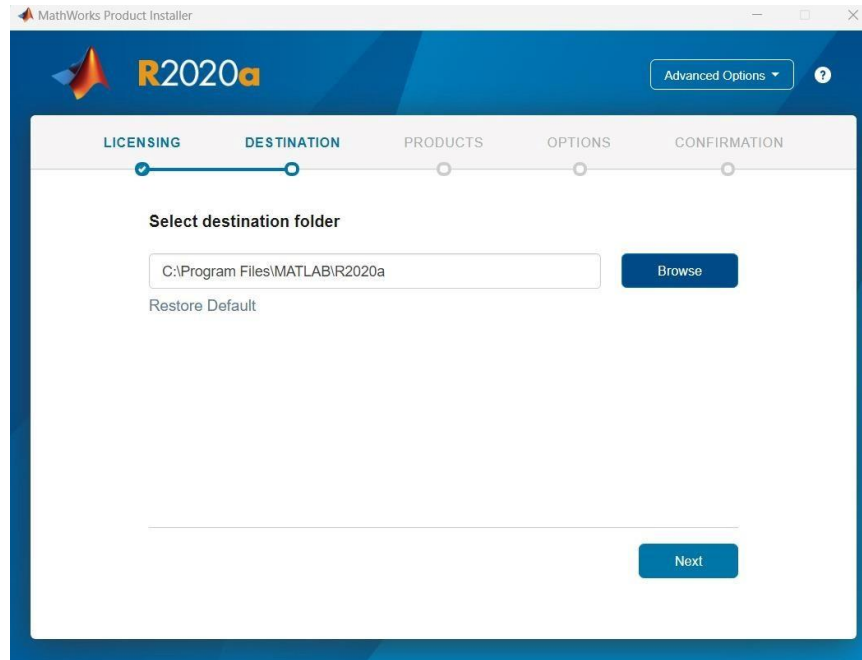
Una vez descargado el archivo .exe en su versión R2020a proceder con ejecutar los siguientes pasos:

1. Ejecutar como administrador el archivo .exe descargado
2. Insertar la clave de activación de producto
3. Aceptar términos y condiciones
4. Elegir la ruta de ubicación de instalación de la aplicación (ver **Figura 18**).



**IMPORTANTE**

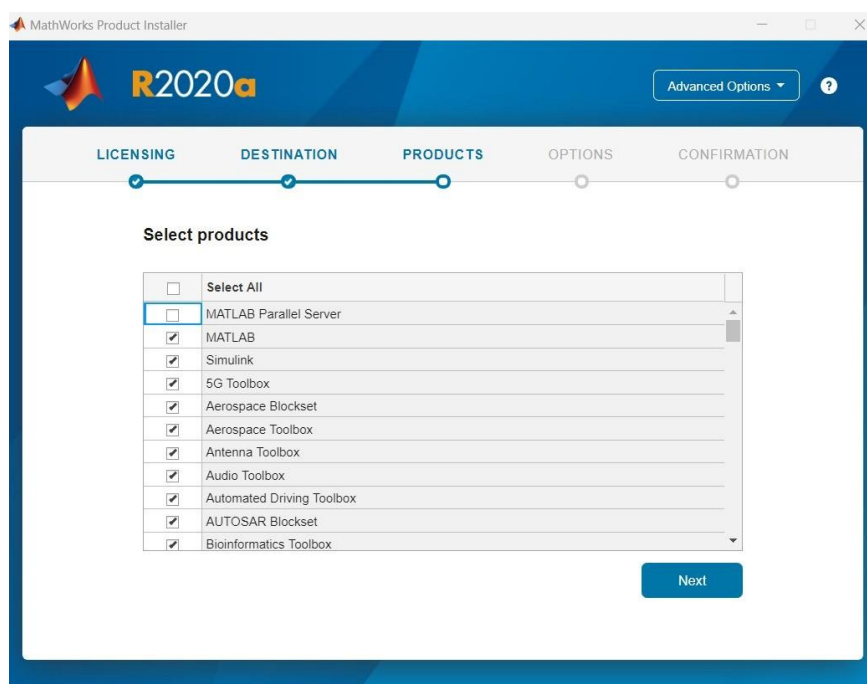
Generalmente, al instalar MATLAB se crea, por defecto, la carpeta *Polyspace* en el disco C, dentro de los cuales se almacenan los elementos fundamentales para el funcionamiento de la aplicación. Sin embargo, una vez finalizada la instalación, *g.tec Suite2020* no va a reconocer la presencia de la aplicación con este nombre (*Polyspace*), por lo que al momento de instalar se recomienda cambiar el nombre de la carpeta de destino por **MATLAB** (**Figura 17**), lo que lo debería identificar dentro del sistema del computador.



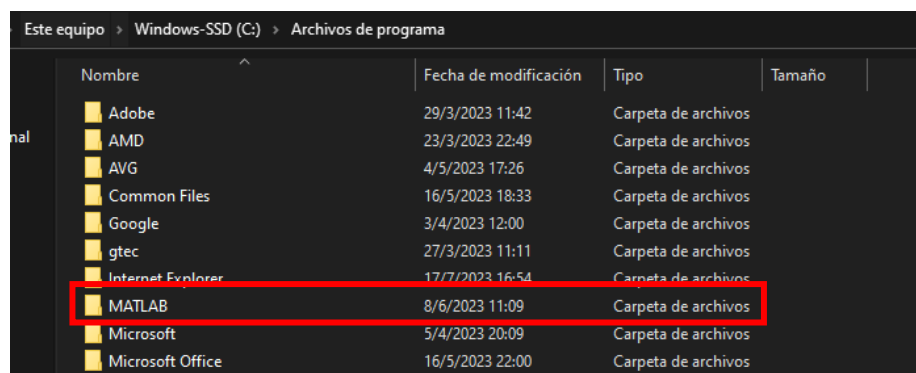
**Figura 18.** Sustitución del nombre Polyspace por MATLAB previo a la instalación del programa

5. Deshabilitar la función de MATLAB SERVER

Es fundamental deshabilitar la función de MATLAB SERVER para permitir su ejecución en conjunto con la aplicación previamente instalada en el paquete g.tec Suite2020 (gNEEDaccess - MATLAB API), como se ilustra en la **Figura 19**. Opcionalmente, también se puede optar por deshabilitar algunas funciones o toolboxes innecesarios para la adquisición de señales, con el objetivo de asegurar que el ordenador pueda escanear un gran número de muestras por segundo sin ralentizar el programa. No obstante, tener precaución al tomar esta decisión y evitar deshabilitar funciones esenciales para el entorno de programación. La **Figura 20** evidencia la carpeta con nombre MATLAB en la ruta C>>Archivos del programa.



**Figura 19.** Desactivar la función MATLAB SERVER previo a la instalación de MATLAB



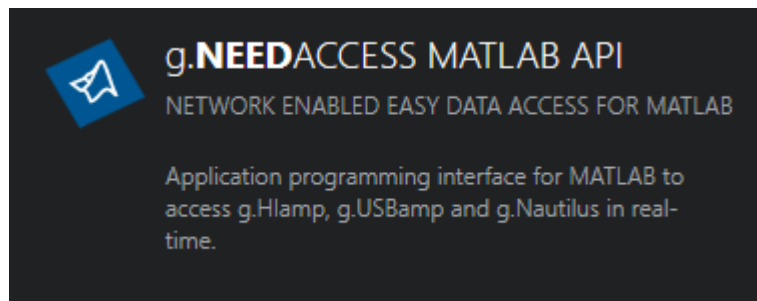
**Figura 20.** Carpeta de MATLAB en archivos del programa

6. Verificar si el programa de MATLAB se ha instalado correctamente y si g.tec Suite2020 lo reconoce en su estado. Para eso verificar la pestaña Sistem Diagnosis

Tool, ubicado en la parte inferior de la ventana principal. Una correcta instalación se lo puede evidenciar en la **Figura 10**.

## 7.2. g.NEEDAccess Matlab API

G.tec ha desarrollado dispositivos que se integran perfectamente con la caja de herramientas de MATLAB. Específicamente, la API de g.NEEDAccess para MATLAB (ver **Figura 21**) ofrece la funcionalidad de conectarse con diversos controladores de estos dispositivos, lo que posibilita la lectura de datos de bioseñales como EEG, ECoG, EMG, EOG y ECG directamente dentro del entorno de MATLAB.



**Figura 21.** Aplicación gNEEDAccess Matlab API.








## 7.3. Pruebas de funcionamiento del equipo g.USBamp con MATLAB

Para proceder a realizar pruebas con el equipo mediante líneas de código, proceder a realizar los pasos de conexión física mencionados en la Sección 4, y a continuación, ejecutar MATLAB con sus funciones de adquisición.

g.NEEDAccess pone a disposición un conjunto de códigos de prueba (ver

**Figura 22**) que permiten utilizarlos y que representan un punto de partida para empezar a adquirir datos en base a líneas de código. Estos códigos de prueba son muy útiles porque proporcionan una excelente guía al empezar a trabajar con MATLAB, y otorga una práctica de cómo interactuar con el software y con las funciones de adquisición instaladas. Los demos que tiene por defecto la aplicación se los pueden encontrar en la ruta

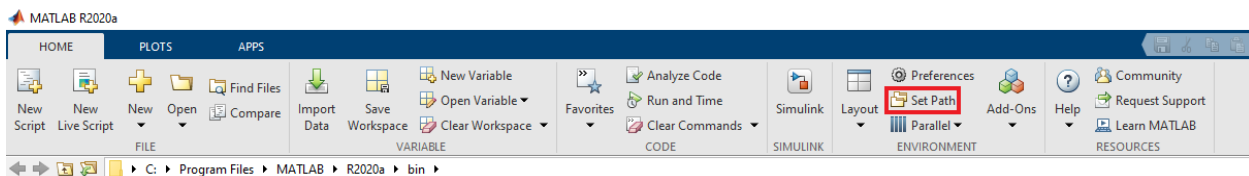
C:\Usuarios<nombre\_de\_usuario>\Documentos\gtec\gNEEDaccessMATLABAPI\Examples\gUSBamp evidenciados en la Figura mismos que se permiten hacer pruebas de inicio de adquisición.

Nombre	Fecha de modificación	Tipo	Tamaño
 gUSBampAPIDemo1	12/5/2023 9:38	MATLAB Code	4 KB
 gUSBampAPIDemo2	4/10/2017 14:03	MATLAB Code	4 KB
 gUSBampAPIDIDemo	6/7/2023 16:56	MATLAB Code	4 KB
 gUSBampAPIGetScalingDemo	23/11/2017 15:37	MATLAB Code	3 KB
 gUSBampAPIImpedanceDemo	23/11/2017 15:37	MATLAB Code	3 KB
 gUSBampAPIScope	2/5/2023 11:15	MATLAB Code	4 KB
 gUSBampAPISynchDemo	6/7/2023 15:17	MATLAB Code	5 KB

**Figura 22.** Ejemplos incluidos en paquete de g.tec dentro de g.NEEDaccessMATLABAPI

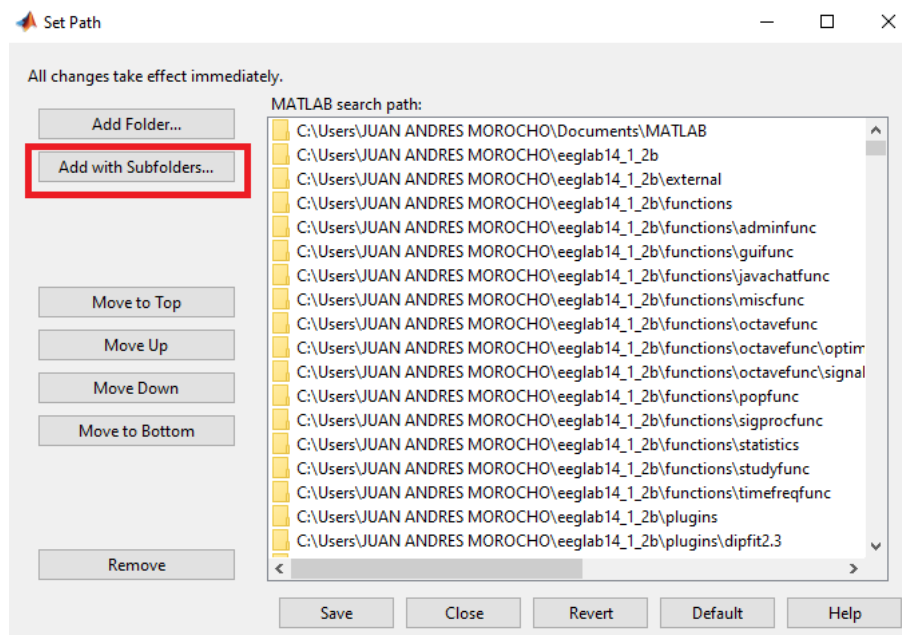
Sin embargo, para ejecutar cualquiera de los demos .m en el entorno de MATLAB, es necesario agregar la ruta donde se encuentran los mismos. Para ello, considerar los siguientes pasos:

1. Dar clic en la opción Set Path (ver **Figura 23**) de la ventana principal de MATLAB



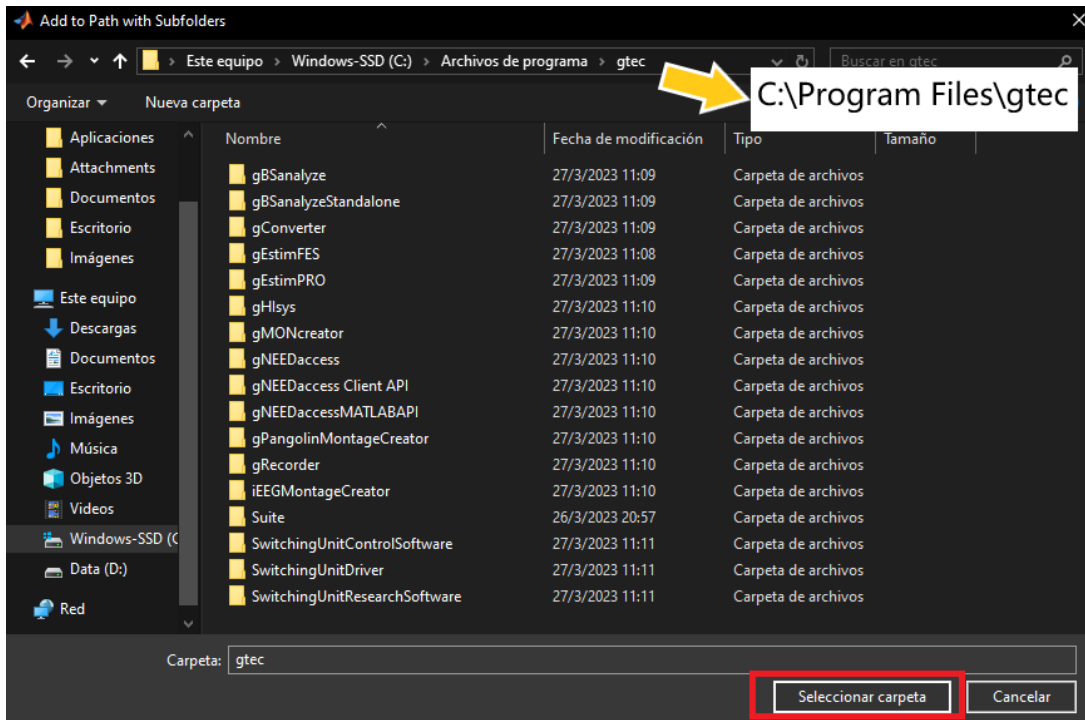
**Figura 23.** Set Path en Matlab

2. Seleccionar la opción Add with subfolders (ver **Figura 21**)



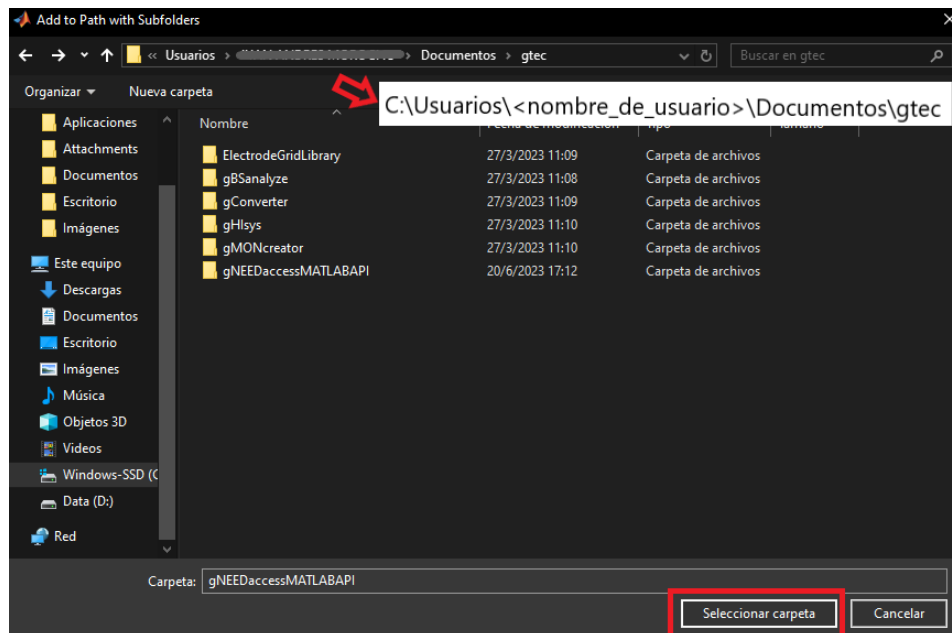
**Figura 24.** Opción Add with Subfolders para agregar las funciones a la carpeta de MATLAB

3. Seleccionar y guardar la carpeta con la ruta C:\Program Files\gttec (FIGURA).



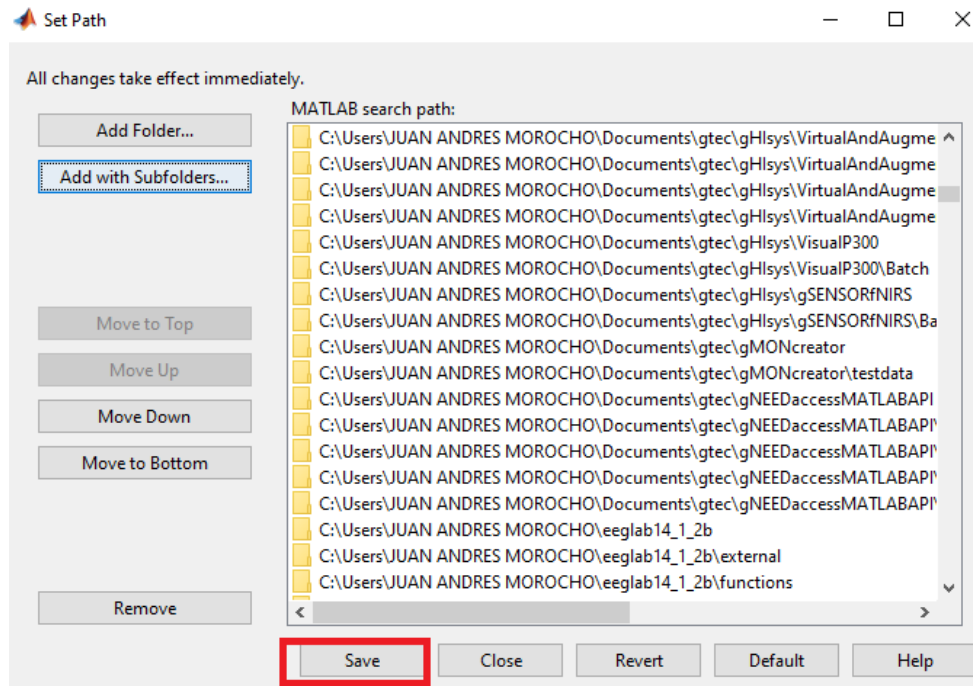
**Figura 25.** Carpeta con las funciones necesarias para ejecutar en MATLAB

4. Opcionalmente se puede añadir otra carpeta donde están la documentación necesaria incluido los demos de prueba del paquete. Esto siguiendo el paso 2 y ubicándose en la ruta: C:\Usuarios\



**Figura 26.** Añadir ruta: C:\Usuarios\<nombre\_de\_usuario>\Documentos\gtec

5. Verificar las subcarpetas agregadas y guardar cambios (ver **Figura 27**).



**Figura 27.** Carpetas agregadas en MATLAB

Como punto de partida se recomienda ejecutar el gUSBampAPIDemo1.m el cual realiza una adquisición de datos de dos canales de entrada analógicos durante 10 segundos a una frecuencia de muestreo de 256 Hz. El ejemplo genera una señal interna tipo sinusoidal que puede configurarse en frecuencia y amplitud, a la vez, es adquirida por el equipo g.USBamp y se muestra en un osciloscopio (scope). Por lo general, los ejemplos en Matlab incluyen una señal interna de prueba muy útil antes de la adquisición. Sin embargo, si se desea adquirir señales exclusivamente de los electrodos sin generar una señal de prueba, es necesario suprimir las siguientes líneas de código:

```
gusbamp_siggen=gUSBampInternalSignalGenerator();
gusbamp_siggen.Enabled = true;
gusbamp_siggen.Frequency = 10;
gusbamp_siggen.WaveShape = 3;
gusbamp_siggen.Amplitude = 200;
gusbamp_siggen.Offset = 0;
gusbamp_config.InternalSignalGenerator= gusbamp_siggen;
```



#### 7.4. Código de Adquisición de datos para 32 canales-Guía del programador

La adquisición de señales se realizó utilizando el demo integrado gUSBampAPISynchDemo, tal como se describe en la Figura, y se ajustó a las condiciones específicas del experimento del Proyecto de Integración Curricular. Este código permite la grabación de 32 canales de manera indefinida, de los cuales 16 contienen señales EEG, 1 canal para marcadores, y los canales restantes no son relevantes para la experimentación. Finalmente, se diseña un filtro para eliminar ruido de los 16 primeros canales. En el siguiente apartado se describen las funciones y líneas de cabo para adquirir señales que pueden emplear los programadores.

##### 7.4.1. Configuración de la ventana de visualización Scope

Crear una ventana de visualización con 5 canales de entrada, una frecuencia de muestreo de 256 Hz y una longitud de búfer de 10 segundos (2560 muestras por canal). Se crea un osciloscopio de tiempo utilizado para visualizar señales en el dominio del tiempo. Luego, se establecen ciertos parámetros adicionales del osciloscopio utilizando la función set.

```
scope_handle = dsp.TimeScope(5,256, 'BufferLength', 2560,...  
    'YLimits', [-5 5], 'TimeSpan', 10, 'LayoutDimensions', [5,1],...  
    'ReduceUpdates',true, 'YLabel','Amplitude [uV]');
```

- 5: Número de canales que se visualizarán en el osciloscopio. Este valor puede cambiar (1-32), sin embargo, mientras mayor sea el número de canales presentados, menor será la calidad de visualización.
- 256: Frecuencia de muestreo de las señales que se visualizarán.
- BufferLength', 2560: Longitud del búfer interno del osciloscopio, es decir, el número máximo de muestras que se pueden almacenar antes de que comience a descartar datos antiguos. En este caso, se establece en 2560 muestras.
- 'YLimits', [-5 5]: Límites del eje Y del osciloscopio, lo que define el rango de valores que se mostrarán en el eje Y. En este caso, se establece el rango de -5 a 5 uV. Este valor puede ser sustituido.
- 'TimeSpan', 10: Tiempo de visualización del osciloscopio, que define la ventana de tiempo que se mostrará en el eje X. En este caso, se establece en 10 segundos, sin embargo, también puede ser reemplazado por otro valor en segundos.

- 'LayoutDimensions', [5,1]: Dimensiones del diseño del osciloscopio en términos de filas y columnas. En este caso, se muestra como una columna con 5 canales. Este vector puede tomar más dimensiones.
- 'ReduceUpdates', true: Esta configuración le indica al osciloscopio que reduzca la cantidad de actualizaciones en la pantalla para mejorar el rendimiento, especialmente útil cuando se manejan grandes cantidades de datos.
- 'YLabel', 'Amplitud [ $\mu$ V]': Etiqueta del eje Y del osciloscopio, indicando la unidad de la amplitud de las señales (en este caso, microvoltios).

```
set(scope_handle, 'ActiveDisplay',5, 'YLimits', [-5 5], 'YLabel','Amplitud [ $\mu$ V]');
```

Se utilizan los parámetros ActiveDisplay, YLimits y YLabel para configurar el osciloscopio creado anteriormente.

- 'ActiveDisplay', 5: Establece el canal activo del osciloscopio en el canal número 5. Esto significa que el canal 5 será el que se muestre de manera predeterminada cuando se ejecute el osciloscopio.
- 'YLimits', [-5 5]: Establece los límites del eje Y del osciloscopio en el rango de -5 a 5 unidades (posiblemente microvoltios).
- 'YLabel', 'Amplitud [ $\mu$ V]': Establece la etiqueta del eje Y del osciloscopio como "Amplitud [ $\mu$ V]", indicando la unidad de la amplitud de las señales (en este caso, microvoltios).

#### **7.4.2. Función gtecDeviceInterface**

```
gds_interface = gtecDeviceInterface();
```

La línea de código gds\_interface llama a un objeto de una clase de nombre gtecDeviceInterface que es una biblioteca externa. Esta instancia representa una interfaz para interactuar con un dispositivo de hardware proporcionado por la empresa g.tec, que puede ser utilizado para adquirir datos.

#### **7.4.3. Configuración de IP y los puertos para el host y el cliente**

```
gds_interface.IPAddressHost = '127.0.0.1';
```

```
gds_interface.IPAddressLocal = '127.0.0.1';  
gds_interface.LocalPort = 50224;  
gds_interface.HostPort = 50223;
```

- gds\_interface.IPAddressHost = '127.0.0.1': Establece la dirección IP del host al que se conectará la interfaz de dispositivo. La dirección IP '127.0.0.1' es la dirección local, lo que significa que el software se conectará al mismo dispositivo en el que se está ejecutando.
- gds\_interface.IPAddressLocal = '127.0.0.1': Establece la dirección IP local que utilizará la interfaz de dispositivo. Al igual que en la línea anterior, se utiliza '127.0.0.1', que es la dirección local de bucle de retorno.
- gds\_interface.LocalPort = 50224: Establece el número de puerto local que utilizará la interfaz de dispositivo para la comunicación. En este caso, se establece en el puerto 50224.
- gds\_interface.HostPort = 50223: Establece el número de puerto del host al que se conectará la interfaz de dispositivo. En este caso, se establece en el puerto 50223.

Se sugiere no variar estos parámetros que g.tec incorpora por defecto en su configuración.

#### **7.4.4. Dispositivos conectados**

```
connected_devices = gds_interface.GetConnectedDevices();  
gusbamp_configs(1,1:2) = gUSBampDeviceConfiguration();  
gusbamp_configs(1,1).Name = connected_devices(1,1).Name;  
gusbamp_configs(1,2).Name = connected_devices(1,2).Name;  
gds_interface.DeviceConfigurations = gusbamp_configs;
```

- connected\_devices = gds\_interface.GetConnectedDevices(): Con esta función crea una lista de dispositivos conectados y devuelve esta información en forma de una estructura de datos (connected\_devices), que contiene detalles sobre los dispositivos conectados, como nombres, identificadores, etc.
- gusbamp\_configs(1,1:2) = gUSBampDeviceConfiguration(): Se está inicializando una matriz gusbamp\_configs de tamaño 1x2. Así se están configurando dos dispositivos con una función que proporciona una configuración predefinida para los dispositivos gUSBamp.

- `gusbamp_configs(1,1).Name = connected_devices(1,1).Name`: Se está asignando el nombre del primer dispositivo conectado (`connected_devices(1,1).Name`) al campo `Name` del primer elemento de `gusbamp_configs`.
- `gusbamp_configs(1,2).Name = connected_devices(1,2).Name`: Se está asignando el nombre del segundo dispositivo conectado (`connected_devices(1,2).Name`) al campo `Name` del segundo elemento de `gusbamp_configs`.
- `gds_interface.DeviceConfigurations = gusbamp_configs`: Se está asignando la matriz `gusbamp_configs`, que contiene la configuración de los dispositivos gUSBamp, al campo `DeviceConfigurations` de la interfaz de dispositivo (`gds_interface`).

#### **7.4.5. Canales disponibles**

```
available_channels_master = gds_interface.GetAvailableChannels(connected_devices(1,1).Name);
available_channels_slave = gds_interface.GetAvailableChannels(connected_devices(1,2).Name);
```

Estas líneas de código están obteniendo los canales disponibles para dos dispositivos específicos conectados a través de la interfaz de dispositivo `gds_interface`.

- Obtiene los canales disponibles equipo 1 y lo guarda en la variable `available_channels_master`.
- Obtiene los canales disponibles equipo 2 y lo guarda en la variable `available_channels_slave`.

#### **7.4.6. Configuración de canales, tierra, referencia y parámetros de grabación**

```
for i=1:size(gusbamp_configs,2)
    gusbamp_configs(1,i).SamplingRate = 256;
    gusbamp_configs(1,i).NumberOfScans = 8;
    gusbamp_configs(1,i).CommonGround = true(1,4);
    gusbamp_configs(1,i).CommonReference = true(1,4);
    for j=1:size(gusbamp_configs(1,i).Channels,2)
        if (available_channels_master(1,j))
            gusbamp_configs(1,i).Channels(1,j).Available = true;
            gusbamp_configs(1,i).Channels(1,j).Acquire = true;
            gusbamp_configs(1,i).Channels(1,j).BandpassFilterIndex = -1;
            gusbamp_configs(1,i).Channels(1,j).NotchFilterIndex = -1;
        end
    end
end
```

```

end
end
gds_interface.DeviceConfigurations = gusbamp_configs;
gds_interface.SetConfiguration();

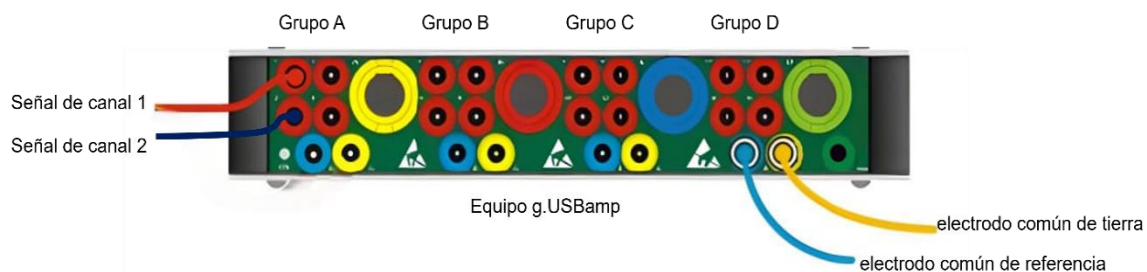
```

- $i$ =número de equipos conectados al pc (en este caso 2)
- $j$ = número de canales de cada equipo (16 canales por equipo)
- $i=1:\text{size}(\text{gusbamp\_configs},2)$ : iteración sobre un rango de valores de  $i$  desde 1 hasta el tamaño de la segunda dimensión de la matriz `gusbamp_configs`
- `gusbamp_configs(1,i).SamplingRate = 256`: Frecuencia de muestreo
- `gusbamp_configs(1,i).NumberOfScans = 8`: Número de escaneos.

### Tierra y referencia

- `gusbamp_configs(1,i).CommonGround = true(1,4)`: Tierra común para los 16 electrodos.
- `gusbamp_configs(1,i).CommonReference = true(1,4)`; Referencia común para los 16 electrodos

Los 4 grupos de canales pueden ser utilizados por sus entradas de tierra y referencia respectivamente, pero también pueden funcionar con una tierra y referencia comunes para los 16 canales, en este sentido, el equipo permite reducir el número de electrodos activos a la hora de realizar las mediciones. La conexión de tierra y referencia se lo evidencia en la **Figura 28**.



**Figura 28.** Conexión de tierra y referencia comunes para los 16 canales

- $j=1:\text{size}(\text{gusbamp\_configs}(1,i).\text{Channels},2)$ : Esto establece la variable de control  $j$  para el bucle. En este caso,  $j$  comienza en 1 y se incrementa en 1 en cada iteración hasta que alcance el tamaño de la segunda dimensión de la matriz `gusbamp_configs(1,i)`. La función `size(gusbamp_configs(1,i).Channels,2)` devuelve el número de columnas de la matriz Channels, lo que determina cuántas iteraciones tendrá el bucle, en este caso, 16 para el equipo 1 + 16 del equipo 2.

- `if (available_channels_master(1,j))`: Esta línea verifica si el canal `j` está disponible según el valor almacenado en `available_channels_master(1,j)`. Si el valor es distinto de cero (lo que se interpreta como verdadero en MATLAB), se ejecuta el bloque de código dentro del `if`.
- `gusbamp_configs(1,i).Channels(1,j).Available = true;`: Si el canal `j` está disponible, se establece la propiedad `Available` del canal correspondiente en la estructura `gusbamp_configs(1,i)` como `true`. Esto indica que el canal está disponible para ser utilizado.
- `gusbamp_configs(1,i).Channels(1,j).Acquire = true;`: Se establece la propiedad `Acquire` del canal correspondiente en la estructura `gusbamp_configs(1,i)` como `true`. Esto indica que el canal debe ser adquirido durante la operación de adquisición de datos.

### Configuración De Filtros

- `gusbamp_configs(1,i).Channels(1,j).BandpassFilterIndex = -1`; Con `-1` no se aplica filtro bandpass. El valor de `-1` puede ser sustituido por valores de la figura 22a dependiendo de los requerimientos.
- `gusbamp_configs(1,i).Channels(1,j).NotchFilterIndex = -1`; Con `-1` no se aplica filtro notch. El valor de `-1` puede sustituirse por valores de la figura 22b según los requerimientos.

En esta etapa, g.tec ofrece filtros del tipo NOTCH y PASOBANDA, los cuales pueden ser configurados según la frecuencia de muestreo utilizada. En la **Figura 29** se presentan los filtros disponibles para el equipo, indicando su orden, índice y frecuencia de muestreo correspondiente.

Fields	FilterIndex	SamplingRate	Order	LowerCutoffFrequency	UpperCutoffFrequency	TypeId
1	1	32	8	0.1000	0	'Butterworth'
2	1	32	8	1	0	'Butterworth'
3	2	32	8	2	0	'Butterworth'
4	3	32	8	3	0	'Butterworth'
5	4	32	8	0	15	'Butterworth'
6	5	32	8	0.0100	15	'Butterworth'
7	6	32	8	0.1000	15	'Butterworth'
8	7	32	8	0.3000	15	'Butterworth'
9	8	32	8	2	15	'Butterworth'
10	9	64	8	0.1000	0	'Butterworth'
11	10	64	8	1	0	'Butterworth'
12	11	64	8	2	0	'Butterworth'
13	12	64	8	3	0	'Butterworth'
14	13	64	8	0	30	'Butterworth'
15	14	64	8	0.0100	30	'Butterworth'
16	15	64	8	0.1000	30	'Butterworth'
17	16	64	8	0.3000	30	'Butterworth'
18	17	64	8	2	30	'Butterworth'
19	18	128	8	0.1000	0	'Butterworth'
20	19	128	8	1	0	'Butterworth'
21	20	128	8	2	0	'Butterworth'
22	21	128	8	3	0	'Butterworth'
23	22	128	8	0	15	'Butterworth'

Fields	FilterIndex	SamplingRate	Order	LowerCutoffFrequency	UpperCutoffFrequency	TypeId
1	1	128	4	48	52	'Butterworth'
2	1	128	4	58	62	'Butterworth'
3	2	256	4	48	52	'Butterworth'
4	3	256	4	58	62	'Butterworth'
5	4	512	4	48	52	'Butterworth'
6	5	512	4	58	62	'Butterworth'
7	6	800	4	48	52	'Butterworth'
8	7	800	4	58	62	'Butterworth'
9	8	1200	4	48	52	'Butterworth'
10	9	1200	4	58	62	'Butterworth'
11	10	2400	4	48	52	'Butterworth'
12	11	2400	4	58	62	'Butterworth'
13	12	4800	4	48	52	'Butterworth'
14	13	4800	4	58	62	'Butterworth'
15	14	9600	4	48	52	'Butterworth'
16	15	9600	4	58	62	'Butterworth'
17	16	19200	4	48	52	'Butterworth'
18	17	19200	4	58	62	'Butterworth'
19	18	38400	4	48	52	'Butterworth'
20	19	38400	4	58	62	'Butterworth'
21						
22						
23						

**Figura 29.** Filtros disponibles para utilizarlos en la configuración de adquisición de señales. a) Filtro pasa banda, b) Filtro notch.

- `gds_interface.DeviceConfigurations = gusbamp_configs;;` Esto significa que los datos de configuración de los dispositivos gUSBamp almacenados en la matriz `gusbamp_configs` se utilizan para configurar la interfaz de dispositivo `gds_interface`. Esto incluiría la configuración de las frecuencias de muestreo, el número de escaneos, la disponibilidad de canales, etc.
- `gds_interface.SetConfiguration();` Aplica la configuración establecida previamente.

#### 7.4.7. Inicio de Adquisición de datos

La adquisición de datos se le ejecuta mediante la línea:

```
gds_interface.StartDataAcquisition();
recorded_data = []
samples_acquired = 0;
```

- `gds_interface.StartDataAcquisition();` Función principal de adquisición de datos.
- `recorded_data = [];` Se crea una matriz `recorded_data`.
- `samples_acquired = 0;` Contador de muestras con valor inicial 0.

La grabación continuará por tiempo indefinido. Para detenerla se ha creado un botón STOP.

```
stop_button_fig = figure;
set(stop_button_fig, 'KeyPressFcn', @keypress_callback);
uicontrol(stop_button_fig, 'Style', 'pushbutton', 'String', 'Stop', 'Position', [20, 20, 60, 20],
'Callback', @stop_callback);
```

#### Bucle de grabación

```
while ~stop_flag && isvalid(scope_handle)
    try
        [scans_received, data] = gds_interface.GetData(8);
    catch ME
        disp(ME.message);
        break;
    end
    recorded_data = [recorded_data; data];
    samples_acquired = samples_acquired + scans_received;
    step(scope_handle, data(:,1),data(:,5) ,data(:,9) ,data(:,13),data(:,17));
    drawnow;
```

```
end
```

- while ~stop\_flag && isvalid(scope\_handle): la grabación se va a realizar mientras stop\_flag=true (botón presionado) o hasta la ventana de visualización sea invalida (error de lectura, o cerrar la ventana de visualización).
- [scans\_received, data] = gds\_interface.GetData(8); Función para adquirir los escaneos.
- break; Se detiene la grabación en caso de un error de adquisición

Las variables recorded\_data y samples\_acquired se van actualizando a valores de adquisición:

- recorded\_data = [recorded\_data; data];
- samples\_acquired = samples\_acquired + scans\_received;
- step(scope\_handle, data(:,1),data(:,5),data(:,9),data(:,13),data(:,17)); Visualizar proceso de grabación para canales 1-5-9-13-17.
- drawnow; Actualizar la ventana de visualización.

#### **7.4.8. Función para detener la grabación**

Una vez presionado el botón STOP (variable stop\_button\_fig), el código va a llamar la función gds\_interface.StopDataAcquisition().

```
gds_interface.StopDataAcquisition()
recorded_data = transpose(recorded_data);
delete(gds_interface)
clear gds_interface gusbamp_config scope_handle;
```

- recorded\_data = transpose(recorded\_data): Esta línea transpone la matriz recorded\_data. La transposición de una matriz intercambia sus filas y columnas, cambiando la orientación de los datos almacenados en recorded\_data.
- delete(gds\_interface): Esta línea llama a la función delete() para eliminar el objeto gds\_interface y liberar los recursos asociados con él. Esto se hace para limpiar y liberar la memoria utilizada por la interfaz de dispositivo después de que ya no se necesite.
- clear gds\_interface gusbamp\_config scope\_handle: Esta línea utiliza la función clear para eliminar las variables gds\_interface, gusbamp\_config y scope\_handle del espacio de trabajo de MATLAB.



#### 7.4.9. Filtrado de datos

```
ch=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16];
ch2=[17];
a=recorded_data(ch,:);
a17=recorded_data(ch2,:);
b=double(a);
aa17=double(a17');
bpFilt = designfilt('bandpassiir','FilterOrder',14, ...
'HalfPowerFrequency1',0.5,'HalfPowerFrequency2',45, ...
'SampleRate',256);
c=(filtfilt(bpFilt,b'));
s=[c aa17];
figure
e1=subplot(3,1,1);
plot(a')
xlabel('Tiempo')
ylabel('Amplitud (uV)')
title('Canales')
grid on
%
e2=subplot(3,1,2);
plot(c)
xlabel('Tiempo')
ylabel('Amplitud (uV)')
title('Canales Filtrados')
grid on
%
e3=subplot(3,1,3);
plot(s)
xlabel('Tiempo')
ylabel('Amplitud (uV)')
title('Canales con marcadores')
grid on
axis([1,length(s),-800,800])
```

```
linkaxes([e1 e2 e3], 'x')  
ylim([-500 800])
```

- `ch=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16];` Se define el vector `ch` que contiene los números de canal que se van a seleccionar de los datos grabados.
- `ch2=[17];` Define el vector `ch2` que contiene el canal de marcadores
- `a=recorded_data(ch,:);` Se seleccionan las filas correspondientes a los 16 canales especificados en el vector `ch` de la matriz `recorded_data`. Esto crea una nueva matriz `a` que contiene solo los datos de los canales seleccionados.
- `a17=recorded_data(ch2,:);` Variable donde se almacena el canal 17 de los marcadores
- `b=double(a);` La matriz `a` se convierte a tipo de datos `double` y se almacena en la matriz `b`, necesario para operaciones de filtrado que requieren datos de tipo `double`.
- `aa17=double(a17'); Convierte los datos del canal adicional a tipo double y transpone la matriz.`
- `bpFilt=designfilt('bandpassiir','FilterOrder',14,...'HalfPowerFrequency1',0.5,'HalfPowerFrequency2',45, ... 'SampleRate',256);` Se diseña un filtro pasa banda IIR utilizando la función `designfilt()`. Este filtro tiene un orden de 14 y frecuencias de corte a -3 dB de 0.5 Hz y 45 Hz. La frecuencia de muestreo se establece en 256 Hz. Los valores de orden, frecuencia de muestreo y de corte pueden ser sustituidos por otros valores.
- `c=(filtfilt(bpFilt,b'));` Los datos en la matriz `b` se filtran utilizando el filtro diseñado `bpFilt` utilizando la función `filtfilt()`. La salida filtrada se almacena en la matriz `c`.
- `s=[c aa17];` Concatena los datos filtrados con los del canal de marcadores
- Visualización de los datos:
  - Se crea una figura y se generan tres subgráficos para mostrar:
  - Los datos originales de los canales.
  - Los datos filtrados.
  - Los datos de los canales con el canal adicional.
- `linkaxes([e1 e2 e3], 'x');` Vincula los ejes `x` de los subgráficos.
- `ylim([-500 800])` : Establece el rango de valores en el eje `y` para los subgráficos.

## 7.5. Funciones adicionales para g.USBamp

Mientras que en `g.RECORDER` se ofrece una interfaz con botones que facilitan la realización de funciones específicas, en el entorno de MATLAB se presenta una dinámica similar. No obstante, la diferencia radica en que todas las acciones de visualización y adquisición se

llevan a cabo mediante líneas de código y funciones desarrolladas especialmente para los equipos g.tec lo permite una mayor flexibilidad y control personalizado sobre las operaciones, otorgando a los usuarios la capacidad de adaptar el funcionamiento del equipo según sus necesidades específicas. La **Tabla 4** describe la lista de funciones que son necesarias para el funcionamiento del dispositivo.

**Tabla 4.**

*Funciones específicas de Matlab para adquirir datos con el equipo g.USBamp*

<b>Función</b>	<b>Uso</b>
<i>gds_interface.GetConnectedDevices()</i>	Escanear dispositivos conectados al computador
<i>gds_interface.GetImpedance(gusbamp_config.Name,channels_selected)</i>	Obtener impedancia
<i>gds_interface.GetAvailableChannels(gusbamp_config.Name)</i>	Obtener canales disponibles
<i>gds_interface.GetAvailableFilters(gusbamp_config.Name,sampling_rate)</i>	Obtener filtros disponibles
<i>gds_interface.GetSupportedSamplingRates(gusbamp_config.Name)</i>	Obtener frecuencias de muestreo aceptadas
<i>gds_interface.GetDeviceInfo(gusbamp_config.Name)</i>	Obtener el nombre del dispositivo
<i>gds_interface.GetScaling(gusbamp_config.Name)</i>	Obtienen una compensación para cada canal
<i>gds_interface.GetAvailableChannels(gusbamp_config.Name)</i>	Obtener canales disponibles
Nota: Algunas funciones pueden o no ejecutarse, mientras que otras son estrictamente necesarias para su funcionamiento	

## 7.6. Código completo

```
% crear una ventana de visualización con 5 canales de entrada, una frecuencia de muestreo
% de 256 Hz y una longitud de búfer de 10 segundos (2560 muestras por canal)
scope_handle = dsp.TimeScope(5,256, 'BufferLength', 2560,...
    'YLimits', [-5 5], 'TimeSpan', 10, 'LayoutDimensions', [5,1],...
    'ReduceUpdates',true, 'YLabel','Amplitude [uV]');
% cambiar al objeto del segundo eje para cambiar el límite y la etiqueta
set(scope_handle, 'ActiveDisplay',5, 'YLimits', [-5 5], 'YLabel','Amplitude [uV]');
drawnow
%%
% llamar la función gtecDeviceInterface
gds_interface = gtecDeviceInterface();
% Buscar dispositivos gtec conectados
connected_devices = gds_interface.GetConnectedDevices();

%% crear array de los dispositivos g.USBamp conectados
gusbamp_configs(1,1:2) = gUSBampDeviceConfiguration(); %Crea una única matriz con dos
dispositivos, los cuales van a ser ocupados por ambos dispositivos
gusbamp_configs(1,1).Name = connected_devices(1,1).Name; % master. Equipo 1, arroja el
nombre y ocupa la posición (1,1)
gusbamp_configs(1,2).Name = connected_devices(1,2).Name;% esclavo. Equipo 2, arroja el
nombre y ocupa la posición (1,2)
%% la función gds_interface.DeviceConfigurations asume las configuraciones anteriores
gds_interface.DeviceConfigurations = gusbamp_configs;
%% obtener los canales disponibles
available_channels_master =
gds_interface.GetAvailableChannels(connected_devices(1,1).Name); % canales disponibles
equipo 1 y lo guarda en la variable available_channels_master
available_channels_slave =
gds_interface.GetAvailableChannels(connected_devices(1,2).Name);% canales disponibles
equipo 2 y lo guarda en la variable available_channels_slave
%% Filtros disponibles
available_filters = gds_interface.GetAvailableFilters((connected_devices(1,1).Name));
%Filtros disponibles almacenados en la variable available_filters
```

```

%% Configuración de canales y parámetros de grabación

% i=número de equipos conectados al pc (en este caso 2)
% j= número de canales de cada equipo (16 canales por equipo)

for i=1:size(gusbamp_configs,2)% iteración sobre un rango de valores de i desde 1 hasta el
tamaño de la segunda dimensión de la matriz gusbamp_configs
    gusbamp_configs(1,i).SamplingRate = 256; %Frecuencia de muestreo
    gusbamp_configs(1,i).NumberOfScans = 8;%Numero de escaneos
    gusbamp_configs(1,i).CommonGround = true(1,4); %tierra común para los 16 electrodos.
    gusbamp_configs(1,i).CommonReference = true(1,4);%referencia común para los 16
electrodos.
    % gusbamp_configs(1,i).ShortCutEnabled = false;
    % gusbamp_configs(1,i).CounterEnabled = false;
    % gusbamp_configs(1,i).TriggerEnabled = false;

    for j=1:size(gusbamp_configs(1,i).Channels,2) %bucle para los 16 canales del equipo 1+los
16 canales del equipo 2
        if (available_channels_master(1,j)) %condición para verificar si existen canales
disponibles
            gusbamp_configs(1,i).Channels(1,j).Available = true; %se empieza a leer los canales
desde 1 hasta j
            gusbamp_configs(1,i).Channels(1,j).Acquire = true;%se empiezan a adquirir las
muestras de los canales 1 hasta j
        end
    end
end
end
%% Funciones necesarias de adquisición
% Aplicar la configuración a la interfaz GDS
gds_interface.DeviceConfigurations = gusbamp_configs;
gds_interface.SetConfiguration();
%% Iniciar la adquisición de datos
gds_interface.StartDataAcquisition(); %Función que permite adquirir las muestras
%%

```

```

%% La grabación va a continuar por tiempo indefinido. Para detenerla se ha creado un botón
STOP
%%
stop_button_fig = figure;
set(stop_button_fig, 'KeyPressFcn', @keypress_callback);
uicontrol(stop_button_fig, 'Style', 'pushbutton', 'String', 'Stop', 'Position', [20, 20, 60, 20],
'Callback', @stop_callback);
%%

global stop_flag; %Esta variable se va a utilizar para detener el proceso de adquisición de
datos cuando se presiona el botón
stop_flag = false;

%% VARIABLE DE GRABACION
recorded_data = []; %se crea una matriz vacía recorded_data
samples_acquired = 0; %contador de muestras con valor inicial 0

    while ~stop_flag && isvalid(scope_handle)%la grabación se va a realizar mientras
stop_flag=true (botón presionado)
        %o hasta la ventana de visualización sea invalida (error de lectura, o cerrar la ventana de
visualización)
        try
            [scans_received, data] = gds_interface.GetData(8);%función para adquirir los
escaneos
        catch ME
            disp(ME.message);
            break; %se detiene la grabación en caso de un error de adquisición
        end

        % las variables recorded_data y samples_acquired se van
        % actualizando a valores de adquisición
        recorded_data = [recorded_data; data];
        samples_acquired = samples_acquired + scans_received;

```

```

%% Visualizar proceso de grabación para canales 1-5-9-13-17
step(scope_handle, data(:,1),data(:,5) ,data(:,9) ,data(:,13),data(:,17));
%     samples_acquired = samples_acquired + scans_received;
drawnow; % Actualizar la ventana de visualización
end

%% Función para detener la adquisición
gds_interface.StopDataAcquisition();
%%
recorded_data = transpose(recorded_data);

%% Limpiar interfaz GDS para una futura grabación

delete(gds_interface)
clear gds_interface gusbamp_config scope_handle;

%% APLICAR FILTRO Y PRESENTACIÓN DE DATOS

ch=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]; %canales a filtrar
ch2=[17]; % canal de marcadores

a=recorded_data(ch,:); %variable donde se almacenan los canales 1-16
a17=recorded_data(ch2,:); %variable donde se almacena el canal 17 marcadores
b=double(a); % La matriz a se convierte a tipo de datos double y se almacena en la matriz b,
necesario para operaciones de filtrado que requieren datos de tipo double.
aa17=double(a17'); %convierte los datos del canal adicional a tipo double y transpone la
matriz.

%% DISEÑO DEL FILTRO PASA BANDA
% define un filtro pasa banda de orden 14 con una frecuencia de corte de 0.5 a 45 Hz
bpFilt = designfilt('bandpassiir','FilterOrder',14, ...
'HalfPowerFrequency1',0.5,'HalfPowerFrequency2',45, ...
'SampleRate',256);
c=(filtfilt(bpFilt,b')); % filtra los datos de los canales utilizando el filtro diseñado.

```

```

s=[c aa17]; % concatena los datos filtrados con los del canal de marcadores

figure % Se crea una figura y se generan tres subgráficos para mostrar:
e1=subplot(3,1,1);
plot(a')
xlabel('Tiempo')
ylabel('Amplitud (uV)')
title('Canales')
grid on
%
e2=subplot(3,1,2);
plot(c)
xlabel('Tiempo')
ylabel('Amplitud (uV)')
title('Canales Filtrados')
grid on
%
e3=subplot(3,1,3);
plot(s)
xlabel('Tiempo')
ylabel('Amplitud (uV)')
title('Canales con marcadores')
grid on
axis([1,length(s),-800,800])
% Se ajustan los ejes, etiquetas y títulos de los gráficos.
linkaxes([e1 e2 e3],'x') % vincula los ejes x de los subgráficos.
ylim([-500 800]) % establece el rango de valores en el eje y para los subgráficos.

close(stop_button_fig); % cerrar la figura del botón detener
% Función de devolución de llamada para el botón de parada
function stop_callback(~, ~) % define la función con dos argumentos de entrada, pero en este
caso, los argumentos no se utilizan, por eso se usan ~ para indicar que se ignoran.
    global stop_flag; % declara la variable stop_flag como global, lo que significa que se puede
acceder y modificar desde cualquier parte del código.

```



```
stop_flag = true; % establece la variable stop_flag en true, que se utiliza para detener el
proceso de adquisición en el código principal cuando se presiona el botón de parada.
end
```

## 8. Advertencias

Las advertencias para la manipulación garantizan un uso seguro y eficaz del equipo. En la **Tabla 5** se menciona las advertencias.

**Tabla 5.**

*Advertencias sobre el equipo g.USBamp*



- El dispositivo no debe utilizarse para la monitorización de pacientes.
- El dispositivo no debe utilizarse para determinar la muerte cerebral.
- El dispositivo no debe utilizarse junto con otro dispositivo médico de alta frecuencia. El uso de un dispositivo de alta frecuencia junto con g.USBamp puede provocar quemaduras debajo de los electrodos y podría dañar el amplificador de bioseñales.
- El aparato no debe utilizarse directamente sobre el corazón.
- No utilice el aparato cerca de un sistema de calefacción ni directamente al sol. La temperatura máxima del ambiente no debe ser superior a 40 °C ni inferior a 5 °C.
- No utilizar otras fuentes de alimentación que no sean la fuente de alimentación técnica médica original

## 9. Seguridad y Mantenimiento

Las recomendaciones para la seguridad y mantenimiento del equipo se mencionan en la **Tabla 6.**

**Tabla 6.**

*Información para la seguridad y mantenimiento del equipo.*

---



- El dispositivo no debe utilizarse para la monitorización de pacientes. Los equipos accesorios conectados a las interfaces analógicas y digitales deben estar certificados según las IEC (por ejemplo, IEC 60950 para equipos de procesamiento de datos e IEC 60601-1 para equipos médicos). médicos). Además, todas las configuraciones deberán cumplir la norma del sistema IEC 60601-1-1.
  - Asegurarse de utilizar cables y conectores adecuados y en buen estado para evitar cortocircuitos y daños al equipo.
  - Si g.USBamp está conectado a otros dispositivos (excepto la fuente de alimentación suministrada con g.USBamp) como un PC, deben comprobarse las siguientes corrientes de fuga.
    - Corriente de fuga a tierra
    - Corriente de fuga del recinto
    - ☐ Corriente de fuga del paciente
  - Ubicar el equipo en un lugar limpio, seco y bien ventilado para evitar la acumulación de polvo y la exposición a la humedad. El aparato no debe utilizarse en condiciones peligrosas, como espacios húmedos o entornos explosivos. La humedad relativa debe estar comprendida entre el 25 % y el 95 %.
  - Realizar una limpieza periódica y cuidadosa del dispositivo utilizando alcohol medicinal, pero es muy importante evitar que el líquido entre en contacto con el g.USBamp.
  - Al transportar o manipular el equipo, asegurarse de hacerlo con cuidado para evitar golpes o caídas que puedan dañar componentes internos.
  - Cuando no esté en uso, guardar el equipo en su estuche o en un lugar seguro para protegerlo de posibles daños
-

## 10. Especificaciones Técnicas

g.USBamp cumple con las siguientes especificaciones mostradas en la **Figura 30**.

### **g.USBamp**

Model	g.USBamp
Type	USB biosignal amplifier
Rated power consumption	7 VA
Rated DC voltage	5 V
Rated current of fuse	1.6 A, quick acting fuse, type 20 mm
Rated voltage of fuse	250 V≈
Produced	see serial number of g.USBamp
Producer	g.tec medical engineering GmbH Sierningstrasse 14 4521 Schiedlberg Austria <a href="http://www.gtec.at">http://www.gtec.at</a>

### Maximum voltages at the following sockets

USB	5 V DC
SC	5 V DC
DIGITAL I/O	5 V DC
SYNC IN	5 V DC
SYNC OUT	5 V DC
POWER SUPPLY	5 V DC
8 pin socket D (DRL, calibration)	± 250 mV AC

### **Amplifier Settings**

#### Channels 1 to 16

Sensitivity:	± 250 mV
Highpass:	0 Hz
Lowpass:	6.6 kHz
Input Impedance:	$>10^{10} \Omega$

#### Analog-digital converter (ADC)

ADC resolution	24 Bit
Sampling frequency	38.400 Hz per channel
Number of ADCs	16

**Figura 30.** Especificaciones Técnicas del equipo amplificador de señales g.USBamp.

## 11. Referencias

- g.tec. (2023). *g.USBAMP RESEARCH | EEG/Biosignal Amplifier | g.tec medical engineering GmbH medical engineering*. <https://www.gtec.at/product/gusbamp-research/>
- MathWorks. (2024). *g.NEEDaccess – MATLAB API - Network-enabled, platform independent data access to g.tec devices from MATLAB - Third-Party Products & Services - MATLAB & Simulink*. [https://la.mathworks.com/products/connections/product\\_detail/g-needaccess.html](https://la.mathworks.com/products/connections/product_detail/g-needaccess.html)