

# Informe I – Uso de Phyphox e importación de datos a MatLab

Dylan Ferney Vasquez Rojas  
1401597  
Comunicaciones Digitales

**Resumen—** En esta práctica se utilizó la aplicación Phyphox (Physical Phone Experiments) para realizar mediciones físicas a través de los sensores integrados en un teléfono inteligente, tales como acelerómetro, giroscopio y sensor de presión. Se registraron datos experimentales en tiempo real y se exportaron en formato CSV. Posteriormente, estos datos fueron importados y analizados en MATLAB, permitiendo la visualización gráfica, el procesamiento de señales y el cálculo de parámetros relevantes.

**Abstract—** In this practical, the Phyphox (Physical Phone Experiments) application was used to perform physical measurements using smartphone-integrated sensors, such as an accelerometer, gyroscope, and pressure sensor. Experimental data was recorded in real time and exported in CSV format. These data were subsequently imported and analyzed in MATLAB, enabling graphical visualization, signal processing, and the calculation of relevant parameters.

## I. INTRODUCCIÓN

El uso de sensores integrados en dispositivos móviles ha abierto nuevas posibilidades para la enseñanza y el análisis de fenómenos físicos. En este informe se presentan dos prácticas realizadas con la aplicación *Phyphox*, una herramienta que permite aprovechar los sensores del teléfono inteligente para registrar datos físicos de manera accesible y precisa. La primera práctica consistió en medir la aceleración lineal eliminando el efecto de la gravedad (aceleración sin "g"), utilizando el acelerómetro del dispositivo. La segunda práctica empleó el giroscopio para registrar la velocidad de rotación en distintos ejes. Los datos obtenidos en ambas experiencias fueron exportados en formato CSV e importados posteriormente a *MATLAB*, donde se realizó el análisis numérico y gráfico de los resultados. Estas actividades permitieron reforzar conceptos como aceleración, movimiento rotacional y análisis de datos, mediante el uso combinado de tecnologías móviles y herramientas de programación científica.

## DESARROLLO DE LA PRÁCTICA

Antes de iniciar el experimento, se verificó que el dispositivo móvil disponible contara con los sensores mínimos requeridos por la aplicación Phyphox. En este caso, se utilizó un Xiaomi Redmi, el cual, según la base de datos oficial de sensores de la aplicación (<https://phyphox.org/sensordb/>), dispone de acelerómetro, giroscopio, sensor de presión, aceleración sin gravedad (sin "g"), magnetómetro y sensor de proximidad. Sin embargo, no cuenta con sensores de humedad,

temperatura ni luz. Para facilitar la comprensión, en la Ilustración 1 se presenta un resumen visual de los sensores incorporados en el dispositivo.

Fabricante	Modelo	Acelerómetro	Aceleración (sin g)	Giroscopio	Magnetómetro	Presión	Temperatura	Humedad	Luz	Proximidad	Barómetro
Xiaomi	2302RMA00L	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible
Xiaomi	2302RMA00L	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible	Disponible

Ilustración. 1. Tabla de sensores dispositivo celular.

II. Ahora se procede a instalar la aplicación Phyphox. En este caso, se utilizó la Play Store para su descarga e instalación. Durante este proceso, se aceptaron los permisos necesarios para que la aplicación pudiera acceder a los sensores del dispositivo y ejecutarse correctamente. Una vez instalada, se muestra una interfaz como la que se observa en la Ilustración 2.

Posteriormente, se seleccionó el experimento de aceleración sin gravedad (sin "g"), el cual permite registrar únicamente la aceleración producida por el movimiento del dispositivo, excluyendo el efecto de la aceleración gravitacional. Esta función es útil para analizar desplazamientos lineales sin interferencia de la gravedad. Durante la práctica, se colocó el dispositivo sobre una superficie móvil para registrar los datos, los cuales fueron almacenados y exportados para su posterior análisis en MATLAB.



Ilustración. 2. Interfaz inicial de PhyPhox.

Luego se realizó el experimento de aceleración sin gravedad (sin "g"), registrando los valores de aceleración neta en los tres ejes del dispositivo (X, Y y Z), excluyendo el efecto constante de la gravedad. El teléfono se desplazó manualmente sobre una

superficie plana para generar variaciones de aceleración. Al finalizar la toma de datos, los resultados fueron visualizados en la interfaz de Phyphox y posteriormente exportados en formato CSV. En la Ilustración 3 se presentan los datos obtenidos durante esta práctica.

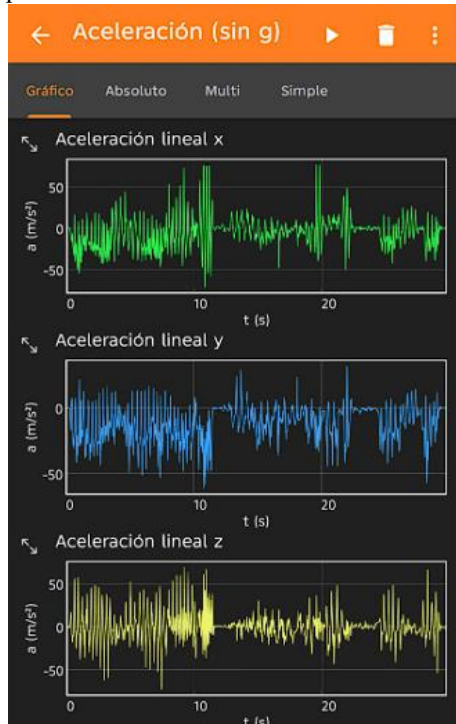


Ilustración 3. Datos obtenidos del experimento

Una vez obtenidos los datos en la aplicación Phyphox, se procedió a exportarlos en formato CSV para su análisis en MATLAB. Este archivo contiene las lecturas de aceleración en los ejes X, Y y Z en función del tiempo. En MATLAB, se importaron los datos utilizando las funciones de lectura de archivos, y posteriormente se graficaron las aceleraciones registradas. Estas gráficas permiten visualizar con mayor claridad los cambios de aceleración durante el experimento, facilitando la interpretación del comportamiento dinámico del dispositivo.

Para importar los archivos a MATLAB Online, primero se sube el archivo Raw Data (generado por Phyphox en formato CSV) a la nube de MATLAB. Esto se realiza desde la interfaz web, utilizando la opción "Upload" en el panel de archivos. Una vez cargado el archivo, se puede acceder a su contenido mediante comandos en el editor o la consola de MATLAB.

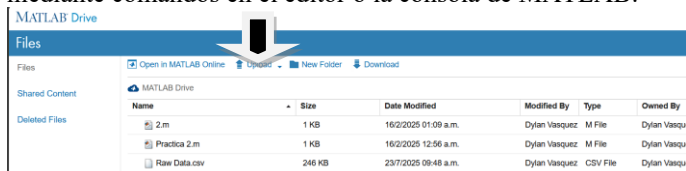


Ilustración 4. Directorio de MATLAB con el archivo Raw Data subido

Después de subir el archivo Raw Data a la nube de MATLAB Online, se procede a importar los datos utilizando la herramienta Import Data. Para ello, se selecciona el archivo previamente cargado (en formato tabulado, ya que fue exportado desde Phyphox con separadores de tabulación). El asistente de importación permite visualizar y seleccionar las

columnas deseadas, facilitando la conversión de los datos en variables que pueden ser utilizadas directamente en el espacio de trabajo de MATLAB para su análisis y graficación.

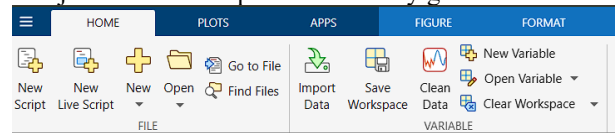


Ilustración 5. Barra de MATLAB para importar el Raw Data

Una vez se ha subido e importado el archivo Raw Data en MATLAB Online mediante la opción Import Data, los datos se visualizan como una tabla estructurada, con columnas correspondientes a las variables medidas (por ejemplo, tiempo y aceleraciones en los ejes X, Y y Z). Esta visualización permite verificar la correcta lectura del archivo y facilita la selección de los datos a utilizar en el análisis. La estructura de la tabla resultante se muestra en la Ilustración 6.

Raw Data.csv X

The following variable was imported: S (3469x5 table)

	A	B	C	D	E
	Time_s	LinearAcce...	LinearAcce...	LinearAcce...	AbsoluteA...
	Number	Number	Number	Number	Number
1	Time (s)	Linear Acce...	Linear Acce...	Linear Acce...	Absolute ac...
2	0.016203	0.15698	0.31276	0.46467	0.5817
3	0.024788	0.48233	0.098744	0.77458	0.9178
4	0.033373	0.43484	-0.23413	0.96522	1.0842
5	0.041959	0.36288	-0.37365	1.0907	1.2087
6	0.050544	0.30779	-0.22038	1.0475	1.1138
7	0.059129	0.26024	-0.14293	0.84685	0.8974
8	0.067714	0.1083	-0.057745	0.72658	0.73687
9	0.076299	-0.013721	0.028676	0.68683	0.68756
10	0.084884	-0.1173	0.028541	0.72013	0.73018
11	0.093469	-0.17427	-0.007315	0.77162	0.79108
12	0.10205	-0.11733	-0.097574	0.76512	0.78019
13	0.11064	0.014222	-0.032644	0.5673	0.56842
14	0.11922	0.1662	0.026471	0.22874	0.28398
15	0.12781	0.3107	-0.12856	0.3284	0.47001

Ilustración 6. Tabla del Raw Data

Una vez importados los datos y generada la tabla en el entorno de trabajo, se procede a ejecutar los siguientes comandos en la ventana de comandos de MATLAB para extraer las variables y graficar las aceleraciones registradas:

```
t=S.Time_s;
x=S.LinearAccelerationX_m_s_2;
y=S.LinearAccelerationY_m_s_2;
z=S.LinearAccelerationZ_m_s_2;
A=S.AbsoluteAcceleration_m_s_2;
subplot(4,1,1);
plot(t,x, Color='r')
title('Aceleracion en X')
subplot(4,1,2);
plot(t,y, Color='g')
title('Aceleracion en Y')
subplot(4,1,3);
plot(t,z)
title('Aceleracion en Z')
subplot(4,1,4);
plot(t,z, Color='k')
title('Aceleracion Absoluta')
```

Después de ejecutar el código anterior, se puede visualizar en

MATLAB el comportamiento de las aceleraciones registradas durante el experimento, representadas en forma de gráficas. Estos resultados permiten comparar y validar los datos previamente observados en la Ilustración 3, ya que corresponden a las mismas mediciones realizadas con la aplicación Phyphox, pero ahora procesadas y representadas de forma más detallada en un entorno de análisis numérico como MATLAB.

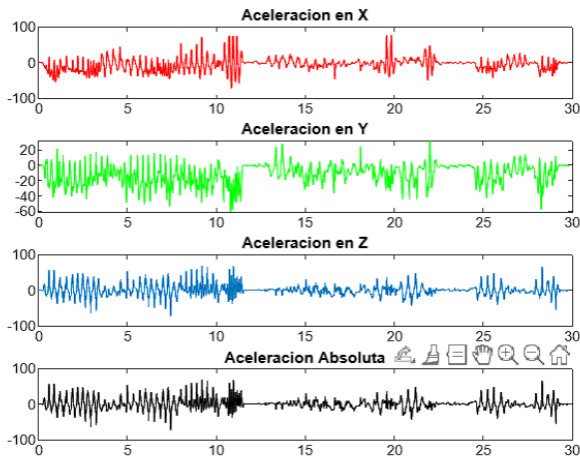


Ilustración 7. Gráficas vistas en MATLAB

Una vez finalizada la práctica de aceleración sin gravedad, se procedió a realizar el segundo experimento utilizando el sensor giroscópico del dispositivo. Este sensor permite medir la velocidad angular en los tres ejes espaciales (X, Y y Z), proporcionando información sobre la rotación del teléfono durante el movimiento. Para ello, se seleccionó en la aplicación Phyphox el experimento correspondiente al giroscopio, y se realizaron movimientos controlados de rotación con el dispositivo, con el fin de registrar las variaciones angulares.

Al igual que en la práctica anterior, los datos obtenidos fueron almacenados y exportados en formato CSV (tabulado) como archivo Raw Data. Este archivo fue posteriormente subido a la nube de MATLAB Online mediante la opción "Upload", y luego importado usando la herramienta Import Data. Como resultado, se generó una tabla estructurada con las variables correspondientes al tiempo y la velocidad angular en los ejes X, Y y Z. Esta visualización se muestra en la Ilustración 8, permitiendo confirmar que la importación se realizó correctamente antes de proceder con el análisis y graficación de los datos.

	A	B	C	D	E
Time s	Gyroscope...	Gyroscope...	Gyroscope...	Gyroscope...	Absolute r...
Number	Number	Number	Number	Number	Number
1	Time (s)	Gyroscope...	Gyroscope...	Gyroscope...	Absolute (r...
2	0.011516	0.084425	-0.00935	-0.011275	0.085686
3	0.013663	0.066138	-0.011825	-0.011275	0.068126
4	0.016809	0.04785	-0.0143	-0.0088	0.05071
5	0.017956	0.0308	-0.013062	-0.0088	0.034593
6	0.020102	0.013612	-0.011825	-0.006325	0.019109
7	0.022249	-0.00715	-0.00825	-0.0050875	0.012044
8	0.024395	-0.022962	-0.0020625	-0.00385	0.023374
9	0.026542	-0.037675	0.00275	-0.00275	0.037875
10	0.028688	-0.054725	0.0064625	-0.0015125	0.055128
11	0.030835	-0.071912	0.012512	-0.0015125	0.073009
12	0.032981	-0.086488	0.014988	-0.00275	0.087777
13	0.035128	-0.1012	0.01375	0.0009625	0.10213
14	0.037274	-0.1122	0.012512	-0.000275	0.1129
15	0.039421	-0.1232	0.0077	-0.00275	0.12347

Ilustración 8. Tabla en MATLAB Del Giroscopio

Ya teniendo los datos correctamente importados y organizados en una tabla (como se muestra en la Ilustración 8), se procedió a generar las gráficas correspondientes a la velocidad angular en MATLAB. Para ello, se utilizó el siguiente bloque de comandos en la ventana de comandos:

```
t=Dy.Time_s_;
x=Dy.GyroscopeX_rad_s_;
y=Dy.GyroscopeY_rad_s_;
z=Dy.GyroscopeZ_rad_s_;
A=Dy.Absolute_rad_s_;
subplot(4,1,1);
plot(t,x, Color='r')
xlabel('time(s)');
ylabel('w(rad/s)');
title('Giroscopio en X')
subplot(4,1,2);
plot(t,y, Color='g')
xlabel('time(s)');
ylabel('w(rad/s)');
title('Giroscopio en Y')
subplot(4,1,3);
plot(t,z)
xlabel('time(s)');
ylabel('w(rad/s)');
title('Giroscopio en Z')
subplot(4,1,4);
plot(t,A, Color='k')
xlabel('time(s)');
ylabel('w(rad/s)');
title('Giroscopio Absoluto')
```

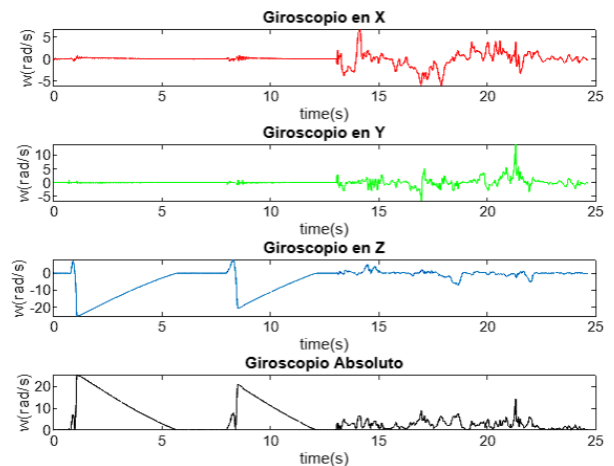


Ilustración 9. Gráficas Del Giroscopio

Para el procesamiento y visualización de los datos obtenidos del sensor de aceleración, se empleó Google Colab como entorno de programación, utilizando el lenguaje Python. A través de la librería Pandas, se cargó el archivo .csv generado por el dispositivo, el cual contenía información de la aceleración lineal en los tres ejes (X, Y, Z) y la aceleración absoluta. Posteriormente, se realizó una limpieza de los nombres de las columnas para facilitar su manipulación, y se renombraron las etiquetas de forma más concisa.

Con ayuda de la librería Matplotlib, se generaron cuatro gráficas individuales utilizando subgráficas (subplots), representando la aceleración en cada eje por separado: X, Y, Z, así como la aceleración absoluta en función del tiempo. Esto permitió una visualización clara y ordenada del comportamiento del sistema en los distintos ejes espaciales. El código utilizado fue el siguiente:

```
import pandas as pd
import matplotlib.pyplot as plt

# Ruta al archivo CSV
ruta = '/content/drive/MyDrive/Raw Data.csv'

# Cargar el archivo
df = pd.read_csv(ruta, sep='\t')

# Limpiar nombres de columna
df.columns = df.columns.str.strip().str.replace(r'\s+', ' ', regex=True)
df.rename(columns={
    'Time (s)': 'Time',
    'Linear Acceleration x (m/s^2)': 'Ax',
    'Linear Acceleration y (m/s^2)': 'Ay',
    'Linear Acceleration z (m/s^2)': 'Az',
    'Absolute acceleration (m/s^2)': 'Abs'
}, inplace=True)

# Crear subplots
fig, axs = plt.subplots(4, 1, figsize=(12, 10), sharex=True)

axs[0].plot(df['Time'], df['Ax'], color='blue')
axs[0].set_ylabel('Ax (m/s^2)')
axs[0].set_title('Aceleración en X')
axs[0].grid(True)

axs[1].plot(df['Time'], df['Ay'], color='green')
axs[1].set_ylabel('Ay (m/s^2)')
axs[1].set_title('Aceleración en Y')
axs[1].grid(True)

axs[2].plot(df['Time'], df['Az'], color='red')
axs[2].set_ylabel('Az (m/s^2)')
axs[2].set_title('Aceleración en Z')
axs[2].grid(True)

axs[3].plot(df['Time'], df['Abs'], color='purple')
axs[3].set_xlabel('Tiempo (s)')
axs[3].set_ylabel('Abs (m/s^2)')
axs[3].set_title('Aceleración Absoluta')
axs[3].grid(True)

plt.tight_layout()
plt.show()
```

A partir de este procedimiento, se obtuvieron las siguientes gráficas:

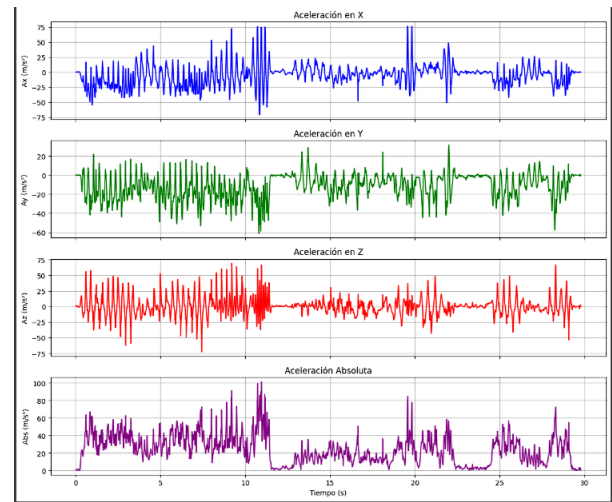


Ilustración.10. Gráficas De La Aceleración Sin g En Python

Al observar las gráficas obtenidas a partir del procesamiento en Python, se puede evidenciar que los resultados visuales son coherentes con los generados por otras plataformas utilizadas durante la práctica, como la aplicación Phyphox y el software MATLAB. En todos los casos, las curvas de aceleración para los tres ejes y la aceleración absoluta presentan una concordancia en forma y comportamiento, lo cual valida la consistencia de los datos capturados por el sensor del dispositivo móvil, independientemente de la herramienta utilizada para su análisis.

Posteriormente, se replicó el mismo procedimiento, pero esta vez enfocado en los datos provenientes del giroscopio. Al igual que, con la aceleración, se cargó el archivo .csv correspondiente en Google Colab, se limpiaron y renombraron las columnas que contenían las velocidades angulares en los ejes X, Y y Z, y finalmente se graficaron para observar el comportamiento rotacional del sistema a lo largo del tiempo. Este enfoque permitió comparar y analizar tanto el movimiento lineal como el rotacional del dispositivo en el mismo entorno de trabajo.

Para analizar los datos registrados por el giroscopio del dispositivo móvil, se utilizó Google Colab como entorno de programación, permitiendo la ejecución de código en Python directamente desde la nube. Se empleó la biblioteca pandas para la lectura y procesamiento de los datos, y matplotlib para su visualización gráfica.

El siguiente código fue implementado para cargar el archivo CSV exportado desde la aplicación Phyphox, limpiar los nombres de las columnas, y graficar la velocidad angular en los tres ejes (X, Y, Z), así como la velocidad angular absoluta:

```
import pandas as pd
import matplotlib.pyplot as plt

# Ruta al archivo CSV
ruta = '/content/drive/MyDrive/Raw D.csv'

# Cargar el archivo
df = pd.read_csv(ruta, sep='\t')
```

Limpiar nombres de columna



```
df.columns = df.columns.str.strip().str.replace(r'\s+', ' ',
regex=True)
df.rename(columns={
'Time (s)': 'Time',
'Gyroscope x (rad/s)': 'Gx',
'Gyroscope y (rad/s)': 'Gy',
'Gyroscope z (rad/s)': 'Gz',
'Absolute (rad/s)': 'Abs'
}, inplace=True)

# Crear subplots
fig, axs = plt.subplots(4, 1, figsize=(12, 10), sharex=True)

axs[0].plot(df['Time'], df['Gx'], color='orange')
axs[0].set_ylabel('Gx (rad/s)')
axs[0].set_title('Velocidad angular en X')
axs[0].grid(True)

axs[1].plot(df['Time'], df['Gy'], color='green')
axs[1].set_ylabel('Gy (rad/s)')
axs[1].set_title('Velocidad angular en Y')
axs[1].grid(True)

axs[2].plot(df['Time'], df['Gz'], color='blue')
axs[2].set_ylabel('Gz (rad/s)')
axs[2].set_title('Velocidad angular en Z')
axs[2].grid(True)

axs[3].plot(df['Time'], df['Abs'], color='black')
axs[3].set_xlabel('Tiempo (s)')
axs[3].set_ylabel('Abs (rad/s)')
axs[3].set_title('Velocidad angular absoluta')
axs[3].grid(True)

plt.tight_layout()
plt.show()
```

Como resultado de este procedimiento, se generaron cuatro gráficas que muestran de forma clara la evolución temporal de la velocidad angular en cada eje, así como la magnitud absoluta del vector de velocidad angular. Estas representaciones visuales son útiles para estudiar el comportamiento dinámico del sistema medido, facilitando su análisis e interpretación.

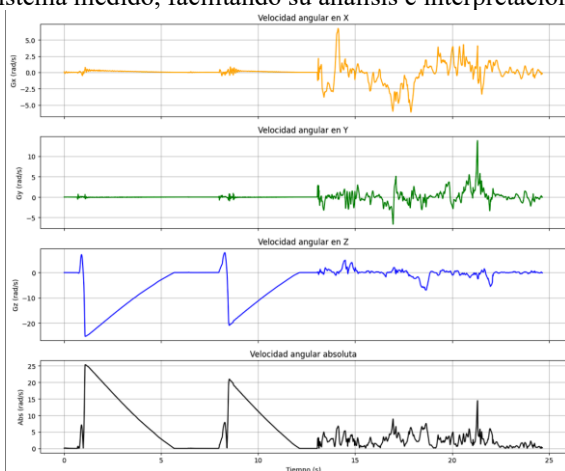


Ilustración.11. Gráficas Del Giroscopio En Python

## CONCLUSIONES

[1] Los datos obtenidos con la app Phyphox permitieron registrar de forma precisa la velocidad angular en los ejes X, Y, Z y su valor absoluto.

[2] Las gráficas obtenidas en Python son coherentes y comparables con las visualizadas en MATLAB y en la misma app Phyphox, lo cual valida la confiabilidad del procedimiento.

[3] Se evidenció que la limpieza y renombramiento de columnas en los datos es clave para evitar errores y facilitar su manipulación.

[4] El análisis de los datos se realizó exitosamente utilizando Python en Google Colab, lo cual ofreció una alternativa eficiente, gratuita y accesible para el procesamiento y visualización de grandes cantidades de datos experimentales. Las bibliotecas pandas y matplotlib facilitaron la manipulación de los datos y la generación de gráficos informativos.

## REFERENCIAS

[1] PhyPhox. “phyphox Sensor Database”. phyphox – Physical Phone Experiments. Accedido el 27 de julio de 2025. [En línea]. Disponible: <https://phyphox.org/sensordb/>

[2] PhyPhox Youtube. “Bevor Sie zu YouTube weitergehen”. Accedido el 27 de julio de 2025. [En línea]. Disponible: [https://www.youtube.com/channel/UC0974nWSlcyJEIXUfYDv9\\_w](https://www.youtube.com/channel/UC0974nWSlcyJEIXUfYDv9_w)

[3] “Google Colab”. Google Colab. Accedido el 27 de julio de 2025. [En línea]. Disponible: <https://colab.research.google.com/>

[4] “MATLAB Online”. MathWorks - Maker of MATLAB and Simulink - MATLAB & Simulink. Accedido el 27 de julio de 2025. [En línea]. Disponible: <https://www.mathworks.com/products/matlab-online/limitations.html>