

# LABORATORIO DE PHYPHOX

Vallentina Diaz Valbuena

[est.vallentina.diaz@unimilitar.edu.co](mailto:est.vallentina.diaz@unimilitar.edu.co)

Docente: José de Jesús Rugeles

## I. RESUMEN

Este informe presenta el desarrollo de dos experimentos prácticos utilizando la aplicación Phyphox en un dispositivo móvil, con el objetivo de analizar datos de sensores físicos y procesarlos mediante herramientas computacionales como MatLab y Python. El primer experimento consistió en medir la aceleración lineal sin gravedad mediante movimientos aleatorios del dispositivo, generando cuatro gráficas que muestran las componentes de aceleración en los ejes x, y, z y su magnitud absoluta. El segundo experimento, denominado "Rodar", consistió en hacer girar el dispositivo sobre un tubo de 4 cm de radio, para analizar la relación entre la velocidad angular (medida por el giroscopio) y la velocidad tangencial.

## II. ABSTRACT

This report presents the development of two practical experiments using the Phyphox application on a mobile device, with the aim of analyzing data from physical sensors and processing them using computational tools such as MatLab and Python. The first experiment consisted of measuring the linear acceleration without gravity through random movements of the device, generating four graphs showing the acceleration components on the x, y, and z axes and their absolute magnitude. The second experiment, called "Rolling", consisted of rotating the device on a 4 cm radius tube, to analyze the relationship between the angular velocity (measured by the gyroscope) and the tangential velocity.

## III. DESARROLLO EXPERIMENTAL

Antes de iniciar los experimentos, se verificó que el dispositivo móvil utilizado contará con los sensores necesarios para el funcionamiento de la aplicación Phyphox. Para ello, se descargó la aplicación en un iPhone 13, el cual será el dispositivo empleado en esta práctica. Según la base de datos oficial de sensores proporcionada por Phyphox (<https://phyphox.org/sensordb/>), este modelo incluye sensores de acelerómetro, giroscopio, presión atmosférica, aceleración sin gravedad, magnetómetro y proximidad. No obstante,

carece de sensores dedicados a medir humedad, temperatura ambiental y luz ambiental, este último, aunque presente físicamente en el dispositivo, no es accesible para aplicaciones externas como Phyphox debido a las restricciones de la plataforma iOS. Para facilitar la comprensión de los sensores disponibles, en la Imagen 1 se muestra un resumen gráfico.

iPhone 13 (iPhone14,5)			
<b>Device</b> Manufacturer: Apple Model: iPhone14,5 Sample size: 410 Variants: 1	<b>Accelerometer</b> Available: <input checked="" type="checkbox"/> Name: Accelerometer Vendor: Apple Range: 16g Resolution: 100.0 Hz Rate: 0.013 m/s <sup>2</sup> Standard deviation: 0.027 m/s <sup>2</sup>	<b>Acceleration (without g)</b> Available: <input checked="" type="checkbox"/> Name: Acceleration (without g) Vendor: Apple Range: 16g Resolution: 100.0 Hz Rate: 0.013 m/s <sup>2</sup> Standard deviation: 0.027 m/s <sup>2</sup>	<b>Gyroscope</b> Available: <input checked="" type="checkbox"/> Name: Gyroscope Vendor: Apple Range: 100.0 Hz Resolution: 100.0 Hz Rate: 0.0079 rad/s Standard deviation: 0.0079 rad/s
<b>Magnetometer</b> Available: <input checked="" type="checkbox"/> Name: Magnetometer Vendor: Apple Range: 100.0 Hz Resolution: 100.0 Hz Rate: 0.15 µT Standard deviation: 0.15 µT	<b>Pressure</b> Available: <input checked="" type="checkbox"/> Name: Pressure Vendor: Apple Range: 1.0 Hz Resolution: 1.0 Hz Rate: 0.0016 kPa Standard deviation: 0.0016 kPa	<b>Temperature</b> Available: <input checked="" type="checkbox"/>	<b>Humidity</b> Available: <input checked="" type="checkbox"/>
<b>Light</b> Available: <input checked="" type="checkbox"/>	<b>Proximity</b> Available: <input checked="" type="checkbox"/> Name: Proximity Vendor: Apple		

Imagen 1. Tabla de sensores del dispositivo

Ahora bien, teniendo en cuenta lo anterior, se escogen dos tipos distintos de experimentos a realizar, en este caso se va a manejar uno de sensor y otro de mecánica. El propósito de este laboratorio es lograr entender el funcionamiento de los experimentos, descargar los datos obtenidos y generar sus respectivas gráficas en MatLab y Python.

El primer experimento a trabajar es la aceleración (sin gravedad), consiste en obtener los datos que proporciona la aceleración real sin la aceleración gravitacional, por medio de algunos movimientos, desplazamientos o giros aleatorios generados en el dispositivo a través de los ejes x, y, z, aproximadamente durante 40 s, teniendo en cuenta:

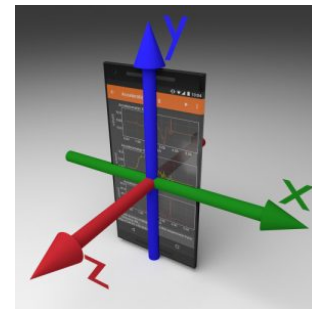


Imagen 2. x, y, z del dispositivo

A partir de la Imagen 2, se tiene una idea de las coordenadas del dispositivo y su dirección. Ya obtenidos los datos se van a exportar en formato CSV (Tabulator, decimal point). Estos se van a descargar en el pc, para el uso de MatLab, estos datos se van a subir a MatLab Drive y se realizan las siguientes acciones:

Name

V

Type

Table

Imagen 3. Variable importada

Al seleccionar todos los datos, se le asigna un nombre corto a la tabla que los contiene, como se observa en la Imagen 3, esto con el propósito de poder identificar fácilmente la tabla de los datos a trabajar.

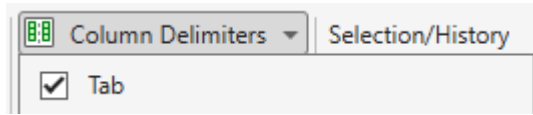


Imagen 4. Delimitadores

Por defecto, MatLab ubica los delimitadores de columna por comas, esto se modifica a la tabulación únicamente, como se observa en la Imagen 4, esto con la intención de que las variables sean separadas y no se reconozcan como una sola. Para finalizar, se selecciona la importación a realizar y se realiza una importación de datos, como se ilustra en la Imagen 5 a continuación.

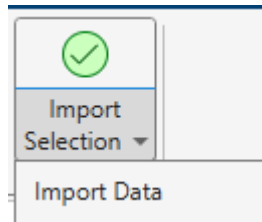


Imagen 5 Importar

Ya con estas modificaciones realizadas, se debe observar la tabla de los datos como se ilustra en la imagen 6, cada columna con su variable correspondiente.

Raw Data.csv ×					
	A	B	C	D	E
	Time (s)	Acceleration x...	Acceleration y...	Acceleration z...	Absolute acc...
	Number ▾	Number ▾	Number ▾	Number ▾	Number ▾
1	Time (s)	Acceleration x ...	Acceleration y (...)	Acceleration z ...	Absolute acc...
2	0.0039376	-4.7402	3.6658	6.6497	8.9513
3	0.0050058	-4.7856	3.3547	6.8734	9.0222
4	0.006073	-5.1924	6.7741	3.88	9.3757

Imagen 6. Tabla de aceleración (sin gravedad)

Ahora bien, se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar las 4 gráficas que se generan a partir del experimento en Phyphox, por medio de sub gráficas, con sus respectivos ejes y títulos.

```
t=V.Time_s_;
x=V.AccelerationX_m_s_2_;
y= V.AccelerationY_m_s_2_;
```

```
z=V.AccelerationZ_m_s_2_;
a=V.AbsoluteAcceleration_m_s_2_;

subplot(4,1,1);

plot(t,x,Color='m')

xlabel("t(s)")

ylabel("a(m/s^2)")

title("Aceleración lineal
x", 'Color', 'm')

subplot(4,1,2);

plot(t,y,Color='g')

xlabel("t(s)")

ylabel("a(m/s^2)")

title("Aceleración lineal
y", 'Color', 'g')

subplot(4,1,3);

plot(t,z,Color='b')

xlabel("t(s)")

ylabel("a(m/s^2)")

title("Aceleración lineal
z", 'Color', 'b')

subplot(4,1,4);

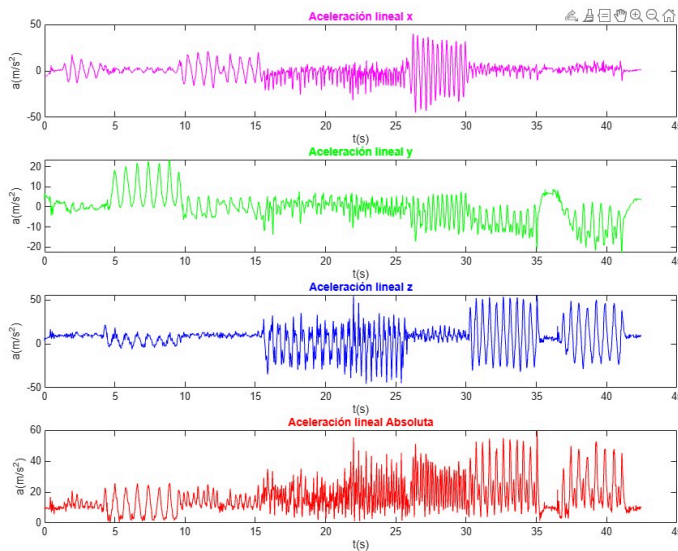
plot(t,a,Color='r')

xlabel("t(s)")

ylabel("a(m/s^2)")

title("Aceleración lineal
Absoluta", 'Color', 'r')
```

A continuación, en la Gráfica 1, se logran visualizar tanto los movimientos específicos en cada eje como la aceleración resultante en el dispositivo.



Gráfica 1. Aceleración (sin g) en MatLab

Los picos y oscilaciones identificados en las señales están asociados a acciones específicas realizadas durante el experimento, tales como desplazamientos, giros o vibraciones controladas, como se mencionó anteriormente.

Al mismo tiempo, se implementa el siguiente código para poder asignar las variables de la tabla de datos a variables más simples y lograr visualizar las 4 gráficas que se generan a partir del experimento en Phyphox, por medio de sub gráficas, con sus respectivos ejes y títulos, en el entorno de desarrollo integrado Thonny con Python.

```
import pandas as pd

import matplotlib.pyplot as plt

ruta_csv =
r'C:\Users\Estudiantel\Downloads\Raw
Data.csv'

df = pd.read_csv(ruta_csv, sep='\t')

df.columns =
df.columns.str.strip().str.replace(' ',
'')

print("Columnas detectadas:",
df.columns.tolist())

t = df['Time (s)']

x = df['Acceleration x (m/s^2)']

y = df['Acceleration y (m/s^2)']

z = df['Acceleration z (m/s^2)']

a = df['Absolute acceleration (m/s^2)']
```

```
plt.figure(figsize=(10, 10))

plt.subplot(4, 1, 1)

plt.plot(t, x, color='m')

plt.title("Aceleración lineal x",
color='m')

plt.ylabel("a (m/s²)")

plt.subplot(4, 1, 2)

plt.plot(t, y, color='g')

plt.title("Aceleración lineal y",
color='g')

plt.ylabel("a (m/s²)")

plt.subplot(4, 1, 3)

plt.plot(t, z, color='b')

plt.title("Aceleración lineal z",
color='b')

plt.ylabel("a (m/s²)")

plt.subplot(4, 1, 4)

plt.plot(t, a, color='r')

plt.title("Aceleración lineal
Absoluta", color='r')

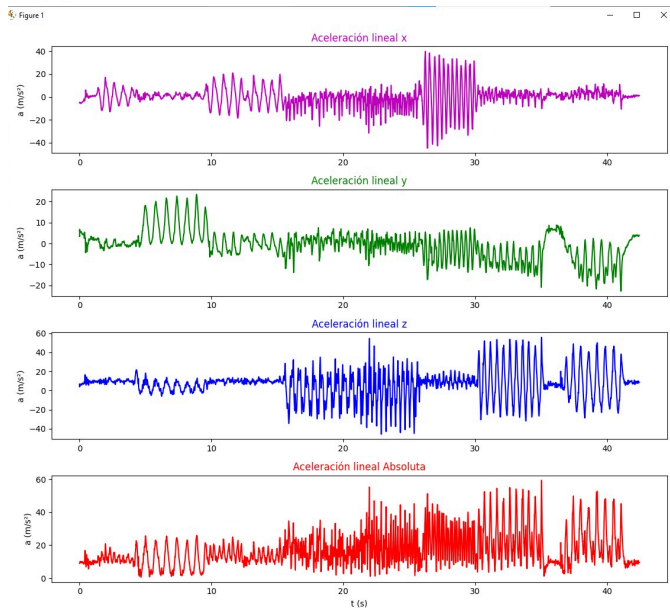
plt.xlabel("t (s)")

plt.ylabel("a (m/s²)")

plt.tight_layout()

plt.show()
```

Se importaron dos librerías, se leyó el archivo usando tabulaciones como separador, se limpiaron nombres de columnas (eliminando comillas y espacios) y se verificaron los nombres reales en el código, para poder visualizar tanto los movimientos específicos en cada eje como la aceleración resultante en el dispositivo, en la Gráfica 2.



Gráfica 2. Aceleración (sin g) en Thonny

La consistencia entre las componentes individuales y la magnitud absoluta permite validar la integridad de los datos y confirmar la correcta operación del sensor de acelerómetro en el dispositivo.

El segundo experimento a trabajar es el rodar, este consiste en colocar el dispositivo en un tubo para que gire sobre su lado más largo. Este experimento luego toma la velocidad angular medida por el giroscopio del dispositivo, así como el radio del tubo, el cual en esta ocasión es de 4 cm, para calcular la magnitud de la velocidad tangencial, es decir la velocidad a la que viaja el tubo.

Ya obtenidos los datos se van a exportar en formato CSV (Tabulator, decimal point). Estos se van a descargar en el pc, para el uso de MatLab y se realizan las mismas acciones que en el primer experimento.

Velocity.csv x

The following variable was imported: J (267x3 table)

	A	B	C
	Time_s_	Velocity_m_s_	GyroscopeY_rad_s_
	Number	Number	Number
1	Time (s)	Velocity (m/s)	Gyroscope y (rad/s)
2	0.026285	-0.0055373	-0.13843
3	0.065979	-0.030805	-0.77013
4	0.10567	-0.045006	-1.1251

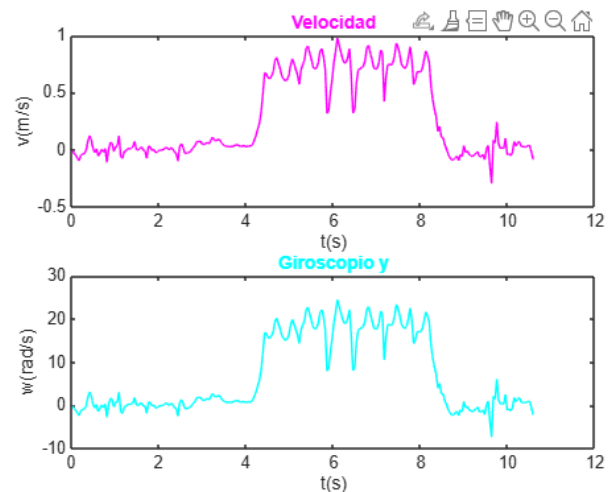
Imagen 7. Tabla de rodar

Ya con las modificaciones realizadas, se debe observar la tabla de los datos como se ilustra en la imagen 7, cada columna con su variable correspondiente. Ahora bien, se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar las 4 gráficas que se

generan a partir del experimento en Phypox, por medio de sub gráficas, con sus respectivos ejes y títulos.

```
t=J.Time_s_;
v=J.Velocity_m_s_;
g=J.GyroscopeY_rad_s_;
subplot(2,1,1);
plot(t,v,Color='m')
xlabel("t(s)")
ylabel("v(m/s)")
title("Velocidad", 'Color', 'm')
subplot(2,1,2);
plot(t,g,Color='c')
xlabel("t(s)")
ylabel("w(rad/s)")
title("Giroscopio y", 'Color', 'c')
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del sensor de velocidad lineal y del giroscopio integrados en el dispositivo utilizado durante el experimento. Las gráficas muestran la evolución temporal de estas magnitudes, expresadas en unidades de metros por segundo (m/s) para la velocidad y radianes por segundo (rad/s) para el giroscopio.



Gráfica 3. Rodar en MatLab

Estas gráficas permiten analizar tanto el movimiento lineal como la rotación del dispositivo en función del tiempo. La coincidencia entre los eventos de mayor actividad en ambas

señales (velocidad y giroscopio) sugiere que los movimientos realizados durante el experimento involucraron tanto traslación como rotación.

Al mismo tiempo, se implementa el siguiente código para poder asignar las variables de la tabla de datos a variables más simples y lograr visualizar las 4 gráficas que se generan a partir del experimento en Phyphox, por medio de sub gráficas, con sus respectivos ejes y títulos, en el entorno de programación en la nube, Google Colab con Python.

```
import pandas as pd

import matplotlib.pyplot as plt

from google.colab import files

print("Sube tu archivo Velocity.csv")

uploaded = files.upload()

data = pd.read_csv('Velocity.csv',
sep='\t')

data.columns =
data.columns.str.strip().str.replace(' ', '')

print("Columnas detectadas:",
data.columns.tolist())

t = data['Time (s)']

v = data['Velocity (m/s)']

g = data['Gyroscope y (rad/s)']

fig, (ax1, ax2) = plt.subplots(2, 1,
figsize=(10, 7))

ax1.plot(t, v, color='m')

ax1.set_xlabel("t (s)")

ax1.set_ylabel("v (m/s)")

ax1.set_title("Velocidad", color='m')

ax1.grid(True)

ax2.plot(t, g, color='c')

ax2.set_xlabel("t (s)")

ax2.set_ylabel("w (rad/s)")

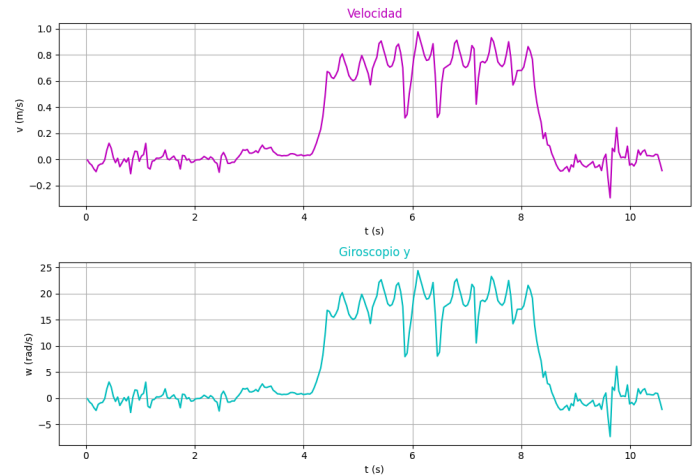
ax2.set_title("Giroscopio y",
color='c')

ax2.grid(True)
```

```
plt.tight_layout()
```

```
plt.show()
```

Se importaron tres librerías, se subió y leyó el archivo, se limpiaron nombres de columnas (eliminando comillas y espacios) y se verificaron los nombres reales en el código, para poder visualizar los datos obtenidos del sensor de velocidad lineal y del giroscopio, en la Gráfica 4.



Gráfica 4. Rodar en Google Colab

Los picos y oscilaciones identificados en las señales pueden estar asociados a acciones específicas realizadas durante el experimento, tales como desplazamientos, giros o vibraciones controladas. La consistencia entre las dos gráficas permite validar la integridad de los datos y confirmar la correcta operación de los sensores de velocidad y giroscopio.

#### IV. CONCLUSIÓN

Aunque el iPhone 13 cuenta con una variedad de sensores físicos, aplicaciones como Phyphox tienen acceso limitado a algunos de ellos debido a las restricciones de privacidad y seguridad del sistema operativo. Ahora bien, los sensores de acelerómetro y giroscopio proporcionaron datos coherentes y de buena calidad, lo que permite su uso confiable en los experimentos. Y se demostró la capacidad de importar, limpiar y visualizar datos de sensores en entornos como MatLab y Python, realizando buen análisis y flexible de datos físicos.

#### V. REFERENCIAS

1. - YouTube. (n.d.-a). Youtu.Be. Retrieved July 30, 2025, from <https://youtu.be/gPq4Le9kXWE?si=gDjL8G9Y6ZNQqM6y>

2. - YouTube. (n.d.-b). Youtu.Be. Retrieved July 30, 2025, from  
<https://youtu.be/YMhyleX5X50?si=ZMHike0L5C4plZYU>
3. Staacks, S. (n.d.). Supported sensors. Phyphox.  
Retrieved July 30, 2025, from  
<https://phyphox.org/sensors/>