

# Experimento Phyphox

Brayan Steven Mendivelso Pérez

[est.brayan.mendive@unimilitar.edu.co](mailto:est.brayan.mendive@unimilitar.edu.co)

Docente: José de Jesús Rúgeles

**Resumen—** El presente informe describirá cómo opera Phyphox, una app para teléfonos que permite llevar a cabo experimentos, recopilar información y apoyar su análisis, fomentando una mejor enseñanza científica., teniendo en cuenta esto se procederá a implementar un informe sobre dos experimentos de Phyphox aceleración sin g y aceleración centrípeta.

**Abstract—** This report will describe how Phyphox operates, an app for phones that allows conducting experiments, collecting information, and supporting its analysis, fostering better science teaching. Taking this into account, we will proceed to implement a report on two Phyphox experiments: acceleration without g and centripetal acceleration.

## I. PRIMER EXPERIMENTO.

La fase inicial del experimento consistirá en verificar que el teléfono móvil disponible (iPhone 13) cumpla con los requisitos sensoriales mínimos de la aplicación Phyphox. De acuerdo con la base de datos sensorial oficial, este dispositivo incluye acelerómetro, giroscopio, sensor de presión, aceleración (sin g), magnetómetro y sensor de proximidad, pero no dispone de sensores de humedad, temperatura ni luz. Para mayor claridad, los sensores disponibles se presentan en la Ilustración 1.

iPhone 13 (iPhone14,5)			
<b>Device</b> Manufacturer: Apple Brand: Apple Sample size: 410 Variants: 1	<b>Accelerometer</b>		
	Available	✓	
	Name		
	Vendor		
	Range		
<b>Magnetometer</b> Available: ✓ Name: Vendor: Range: Resolution: Rate: 100.0 Hz Standard deviation: 0.15 $\mu$ T	<b>Pressure</b>		
	Available	✓	
	Name		
	Vendor		
	Range		
<b>Light</b> Available: ✗	<b>Temperature</b>		
	Available	✗	
	Name		
	Vendor		
	Range		
<b>Proximity</b> Available: ✓ Name: Vendor: Range: Resolution: Rate: 1.0 Hz Standard deviation: 0.0078 mPa	<b>Humidity</b>		
	Available	✗	
	Name		
	Vendor		
	Range		
<b>Acceleration (without g)</b> Available: ✓ Name: Vendor: Range: Resolution: Rate: 100.0 Hz Standard deviation: 0.019 m/s <sup>2</sup>	<b>Gyroscope</b>		
	Available	✓	
	Name		
	Vendor		
	Range		

Ilustración 1 Tabla de sensores dispositivo celular.

Ahora se procede a instalar la aplicación, en este caso se usará la app store y se aceptamos los permisos requeridos para estos dos experimentos, el primer experimento que vamos a realizar es la aceleración sin g, lo primero será abrir la aplicación que tiene la siguiente interfaz que se ve en la ilustración 2, después seleccionamos la opción de Aceleración (sin g) como se ve en la ilustración 3.



Ilustración 2 Interfaz de Phyphox.



Ilustración 3 Aceleración (sin g).

Ahora nos saldrá el menú de Aceleración (sin g) que son 3 acelerómetros de los ejes x, y y z como se ve en la ilustración 4, para este experimento vamos a lanzar el celular dando giros en la cama para mayor seguridad, ya que este experimento mide los 3 ejes y esta fue la manera mas sencilla de hacer mover todos los ejes.

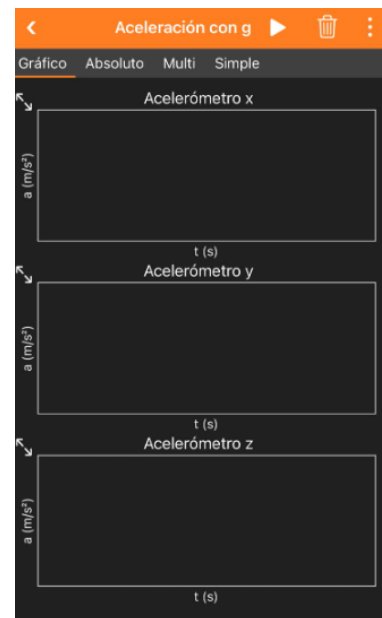


Ilustración 4 Interfaz de aceleración (sin g)

Ahora vamos a ver los resultados de los 3 ejes y la aceleración absoluta como se ve en la ilustración 4 y 5.

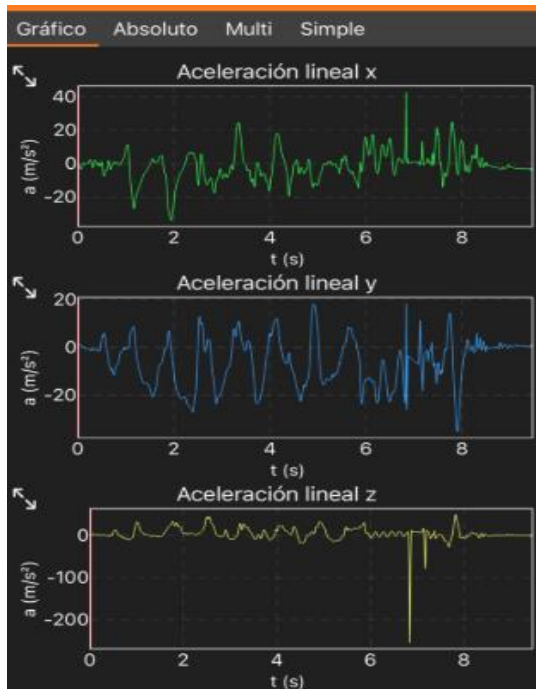


Ilustración 5 Aceleración de los 3 ejes.



Ilustración 6 Aceleración absoluta.

Ahora vamos a exportar los datos para esto vamos a dar clic en los 3 punto que se encuentra en la parte superior derecha, damos en la opción exporta datos y tenemos varias opciones, para nuestro caso vamos a seleccionar la opción CSV(Tabulador,deciaml point) este nos genera una tabla de datos como se ve en la ilustración 7, este archivo lo vamos a usar para generar los gráficos para esto vamos a usar la aplicación Matlab y Thonny, esta va a generar el mismo gráfico, pero es para utilizar diferente método y tener un mejor aprendizaje.

Time (s)	Linear Acceleration x (m/s²)	Linear Acceleration y (m/s²)	Linear Acceleration z (m/s²)	Absolute acceleration (m/s²)
2.073963334E-2	-3.828125310E-1	9.024814832E-1	-2.223589630E-2	4.30065759E0
3.071183334E-2	-6.289551577E-1	7.11283797E-1	-1.333305920E-1	7.65187823E0
4.068383334E-2	-1.560551691E-1	2.73797219E-2	0.17646090E-1	1.024497442E0
5.065483334E-2	-2.526222572E-1	2.20866761E-1	4.21593143E-2	1.45533442E0
6.062683334E-2	-3.270335728E-1	0.51010409E-2	4.495123278E-2	4.246628202E0
7.059883334E-2	-3.630186535E-1	7.40048821E-1	-3.128040519E-2	4.848374721E0
8.057083334E-2	-3.282518109E-1	4.409166518E-1	3.43065544E-2	4.771688340E0
9.054283334E-2	-2.200693996E-2	2.283738247E-1	-3.399600416E-2	4.056118285E0
1.005138333E-1	-1.017655844E-2	-3.603755951E-2	-2.897044510E-2	3.070796154E0
1.104868333E-1	-4.93475769E-2	-1.838650794E-1	-2.152631392E-2	2.161906468E0
1.204578333E-1	-4.55505451E-1	-9.21372028E-2	-1.28106354E-2	1.362779045E0
1.304298333E-1	-6.982978821E-2	-1.934524986E-1	-4.868325289E-1	-6.101304246E-1
1.404008333E-1	-6.913412899E-1	-2.882007610E-1	-2.315678862E-1	-6.889419676E-1
1.503728333E-1	-6.263017005E-1	-2.441124827E-2	-3.977499199E-1	-7.370208474E-1
1.603448333E-1	-1.644704819E-1	-4.962923888E-1	-8.248982871E-1	-9.786339328E-1

Ilustración 7 Archivo CSV.

Una vez tenemos el archivo .csv vamos a abrir Matlab y subir este archivo y nos mostrará el archivo de la siguiente manera como se ve en la ilustración 8.

Ilustración 8 muestra una captura de pantalla de la interfaz de Matlab. Se ve una ventana con el título 'RawData.csv X' que contiene una tabla de datos. La tabla tiene 5 columnas: 'Time (s)', 'Linear Acceleration x (m/s²)', 'Linear Acceleration y (m/s²)', 'Linear Acceleration z (m/s²)' y 'Absolute acceleration (m/s²)'. La tabla contiene 18 filas de datos.

Ilustración 8 archivo en Matlab.

Ahora vamos a organizar esta tabla para esto vamos a ingresar a IMPORT y seleccionar toda la tabla y vamos a darle Column Delimiters y seleccionamos Tab, y se transforma en la siguiente tabla como se ve en la ilustración 9 que ya está mejor organizada.

	A	B	C	D	E
	Time_s_	LinearAcce...	LinearAcce...	LinearAcce...	AbsoluteA...
Number	Number	Number	Number	Number	Number
1	Time (s)	Linear Acce...	Linear Acce...	Linear Acce...	Absolute ac...
2	0.02074	-0.38281	0.90248	-2.2236	2.4301
3	0.030712	-0.62857	0.97113	-1.3333	1.7652
4	0.040684	-1.5606	1.2738	0.20176	2.0245
5	0.050655	-2.5262	1.2209	1.422	3.1455
6	0.060627	-3.2703	1.051	2.4951	4.2455
7	0.070599	-3.6302	0.744	3.128	4.8494
8	0.080571	-3.2825	0.44092	3.4351	4.7717
9	0.090543	-2.2008	0.22837	3.3996	4.0561
10	0.10051	-1.0177	-0.036038	2.897	3.0708
11	0.11049	-0.049348	-0.19386	2.1526	2.1619
12	0.12046	0.45558	-0.092137	1.2811	1.3628
13	0.13043	0.06983	0.15345	0.46683	0.51013
14	0.1404	-0.58134	0.2882	0.23157	0.68894
15	0.15037	-0.6263	-0.024411	0.38775	0.73702
16	0.16034	-0.16447	-0.49629	0.8249	0.97663
17	0.17032	0.088918	-0.83167	1.3338	1.5743
18	0.18029	0.24185	-1.006	1.7208	2.0079

Ilustración 9 datos organizados.

Por último, vamos a volver estas columnas volver variables para poder graficar, para poder graficar vamos a usar los siguientes comandos como se ve la ilustración 10

```
x = RawData.LinearAccelerationX_m_s_2_;
y = RawData.LinearAccelerationY_m_s_2_;
z = RawData.LinearAccelerationZ_m_s_2_;
h = RawData.AbsoluteAcceleration_m_s_2_;
t = RawData.Time_s_;

subplot(2,2,1)
plot(t, x)
grid on
xlabel('t(s)')
ylabel('a(m/s^2)')
title('Aceleración en el eje x')

subplot(2,2,2)
plot(t, y)
grid on
xlabel('t(s)')
ylabel('a(m/s^2)')
title('Aceleración en el eje y')

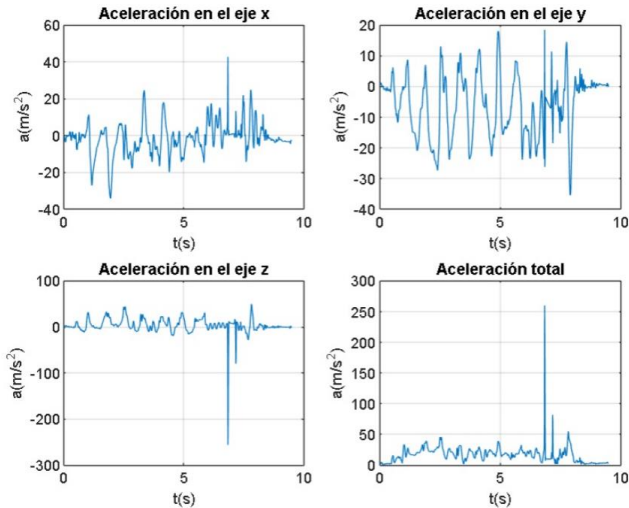
subplot(2,2,3)
plot(t, z)
grid on
xlabel('t(s)')
ylabel('a(m/s^2)')
title('Aceleración en el eje z')

subplot(2,2,4)
plot(t, h)
grid on
xlabel('t(s)')
ylabel('a(m/s^2)')
title('Aceleración total')
```

Ilustración 10 Comando de Matlab.

Ahora vamos a explicar que hace cada comando, el primer comando `x = RawData.LinearAccelerationX_m_s_2_;` crea la variable x y agrega los datos de la aceleración en el eje x, así

con el eje y,z y aceleración total, el comando subplot(2,2,1) es para crear un subconjunto de gráficos el primer 2 es columnas que va generar, el segundo 2 es para las filas, el 1 es para saber a cuál posición vas a poner la información, el comando plot() es para poner las dos variables que vamos a graficar ,el comando grid on es para poner la cuadra del fondo de la graficas, xlabel y ylabel son para poner texto en los ejes por ultimo title es para poner un título a la gráfica.



**Ilustración 11 gráficos generados por Matlab.**

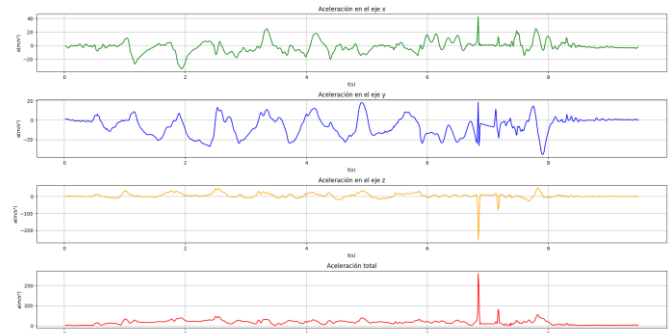
Ahora vamos a usar Thonny para realizar la misma gráfica, lo primero que vamos a hacer es instalar las siguientes librerías pandas, matplotlib.pyplot y numpy, panda se utiliza para la manipulación y análisis de datos estructurados como tablas y series temporales, permitiendo cargar, limpiar y transformar datos de manera eficiente. La librería numpy es fundamental para operación numéricas avanzadas, proporcionando arreglos multidimensionales rápidos y funciones matemáticas esenciales que sirven de base para cálculos científicos. matplotlib.pyplot se emplea para visualización de datos, creando gráficos estáticos o interactivos (líneas, barras, dispersión, etc.) que facilitan la exploración y presentación de resultados.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 ruta_csv = r'D:\Universidad\Semestre 6\Comunicaciones digitales\meta\RawData.csv'
6 a = pd.read_csv(ruta_csv, sep='t')
7 # Limpiar nombres de columnas (eliminar comillas y espacios)
8 a.columns = a.columns.str.strip().str.replace(' ', '')
9 # Verificar los nombres reales de las columnas
10 print("Columnas detectadas:", a.columns.tolist())
11 x = a['Linear Acceleration x (m/s²)']
12 y = a['Linear Acceleration y (m/s²)']
13 z = a['Linear Acceleration z (m/s²)']
14 h = a['Absolute acceleration (m/s²)']
15 t = a['time (s)']
16 plt.figure(figsize=(12, 10))
17 plt.subplot(4, 1, 1)
18 plt.plot(t, x, color='green')
19 plt.grid(True)
20 plt.xlabel('t(s)')
21 plt.ylabel('a(m/s²)')
22 plt.title('Aceleración en el eje x')
23 plt.subplot(4, 1, 2)
24 plt.plot(t, y, color='blue')
25 plt.grid(True)
26 plt.xlabel('t(s)')
27 plt.ylabel('a(m/s²)')
28 plt.title('Aceleración en el eje y')
29 plt.subplot(4, 1, 3)
30 plt.plot(t, z, color='orange')
31 plt.grid(True)
32 plt.xlabel('t(s)')
33 plt.ylabel('a(m/s²)')
34 plt.subplot(4, 1, 4)
35 plt.plot(t, h, color='red')
36 plt.grid(True)
37 plt.xlabel('t(s)')
38 plt.ylabel('a(m/s²)')
39 plt.title('Aceleración total')
40 plt.tight_layout()
41 plt.show()
```

**Ilustración 12 Comandos usando.**

Como se ve en la ilustración 12 creamos la variable de la ruta donde se encuentra en archivo csv, y como en el anterior la

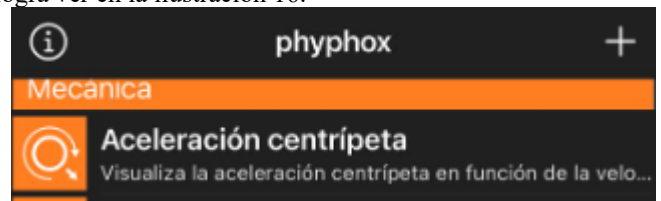
variable de los ejes y variable para imprimir las gráficas, ahora vamos a observar la gráfica que nos arroja esta debería ser idéntica a la de Matlab.



**Ilustración 13 Grafica por Thonny.**

## II. SEGUNDO EXPERIMENTO.

En esta segunda parte vamos a usar la Aceleración centrípeta , para este experimento vamos a usar nuestro celular y una cicla, el celular lo pondremos en la llanta delantera de la cicla , primero vamos a giraremos lentamente la llanta para asegurar que el celular no salga volando , una vez tengamos seguro el celular vamos a hacer la prueba, en la ilustración 14 vemos la opción de aceleración centrípeta vamos a abrirla y nos saldrá la interfaz como se ve en la ilustración 15 por ultimo vamos a mostrar el montaje del celular en la llanta delantera como se logra ver en la ilustración 16.



**Ilustración 14 Aceleración centrípeta.**



**Ilustración 15 interfaz de aceleración centrípeta.**



**Ilustración 16 celular en la llanta.**

Como se logra observa en la Ilustración 15, tenemos la aceleración y la velocidad angular ambas comparadas en función del tiempo, ahora vamos a mostrar los resultados que nos da aplicación Phyphox, como se logra ver en la ilustración 17, como en el experimento anterior vamos a extraer los datos en CSV con la opción (Tabulador,deciaml point) para este experimento no vamos a mostrar el archivo, vamos a ir directamente a Matlab ya que los pasos son los mismos.



**Ilustración 17 Gráfico de experimento dos.**

Una vez subimos y organizamos el archivo en el Matlab nos queda de la siguiente manera.

	A	B	C
	Time_s_	AngularVe...	Accelerati...
	Number	Number	Number
1	Time (s)	Angular vel...	Acceleratio...
2	0.056666	0.13218	0.72077
3	0.55517	1.7835	4.8501
4	1.5522	9.7406	18.2253
5	2.0507	9.5759	17.205
6	2.5492	9.4042	16.4821
7	3.0477	9.28	16.7916
8	3.5462	9.097	16.0542
9	4.0447	8.91	14.6577
10	4.5432	8.76	14.7743
11	5.0417	8.6586	14.5555
12	5.5402	8.4896	13.2637
13	6.0387	8.2981	13.2373

**Ilustración 18 datos en Matlab.**

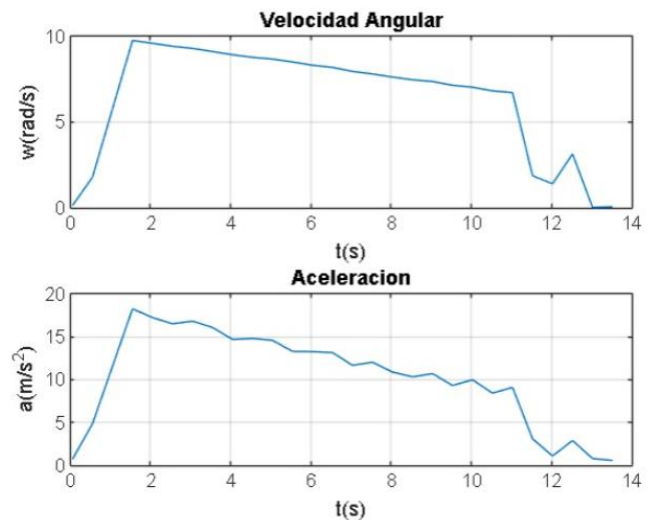
```
t=Dat.Time_s_;
x=Dat.AngularVelocity_rad_s_;
y=Dat.Acceleration_m_s_2_;
```

```
subplot(2,1,1)
plot(t,x)
grid on
xlabel('t(s)')
ylabel('a(m/s^2)')
title("velocidad angular")
```

```
subplot(2,1,2)
plot(t,y)
grid on
xlabel('t(s)')
ylabel('w(rad/s)')
title("Aceleración")
```

**Ilustración 19 comandos para genera gráficos.**

Como en el experimento anteriores las variables t,x,y las creamos para pasar la información de las columnas a estas variables, en la variable t ponemos la información de Time\_s\_, en la variable x se pone la información de la velocidad angular por último la variable y para poner la información de la aceleración, como en el experimento anterior se usan los mismos comandos ya explicados entonces es redundante explicarlo nuevamente, vamos a visualizar la gráfica generada en Matlab.



**Ilustración 20 Gráfica de experimento dos en Matlab.**

Por último, vamos a usar Thonny para generar este gráfico usaremos los comandos que se observan en la ilustración 21, estos son muy parecidos al experimento anterior pero mucho más corto, pero hace la misma función crea variable y las almacena en t, accel y vel. Después las graficamos con plot(), ponemos los nombres a los ejes y un título.



```
import pandas as pd
import matplotlib.pyplot as plt
ruta_csv = r'D:\Universidad\Semestre 6\Comunicaciones d
a = pd.read_csv(ruta_csv, sep='\t')
a.columns = a.columns.str.strip().str.replace(' ','')
print("Columnas detectadas:", a.columns.tolist())
t = a['Time (s)']
acel = a['Acceleration (m/s^2)']
vel=a['Angular velocity (rad/s)']
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plt.plot(t, acel, color='purple')
plt.grid(True)
plt.xlabel('Tiempo (s)')
plt.ylabel('a (m/s^2)')
plt.title('Aceleración ')
plt.subplot(2, 1, 2)
plt.plot(t, vel, color='blue')
plt.grid(True)
plt.xlabel('Tiempo (s)')
plt.ylabel('w (rad/s)')
plt.title('Velocidad Angular')
plt.tight_layout()
plt.show()
```

#### Ilustración 21 Comando para Thonny.

Vamos a ver las graficas que nos genera estos comandos como se ve en la ilustración 22 , tenemos la misma grafica que nos dios Phyphox y Matlab.

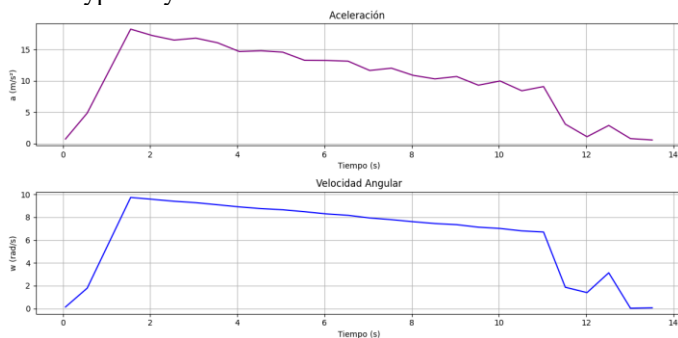


Ilustración 22 grafica en Thonny segundo experimento.

### III. ANÁLISIS

En estos experimentos se utilizó la aplicación Phyphox, una herramienta que permite convertir un teléfono celular en un laboratorio portátil mediante el uso de sus sensores internos, como acelerómetro, giroscopio y magnetómetro. Esta aplicación es útil para realizar experimentos de física de manera sencilla, accesible y sin necesidad de equipos costosos. Entre sus principales ventajas está la posibilidad de exportar los datos en formato .csv lo que permite analizarlos posteriormente en programas como MATLAB o entornos de programación como Thonny (Python). Además, ofrece una interfaz amigable que facilita la visualización de los resultados en tiempo real, lo cual es ideal tanto para estudiantes como para docentes en el campo de la enseñanza STEM.

En el primer experimento, se configuró la app para medir la aceleración lineal (sin incluir la gravedad) y se procedió a lanzar el celular sobre una cama, permitiéndole girar y caer libremente. Al analizar los datos obtenidos, se observan gráficas de aceleración en los tres ejes (X, Y y Z). En los ejes X e Y se presentan oscilaciones con picos que superan los  $\pm 20$  m/s<sup>2</sup>, lo cual indica un comportamiento irregular y rotacional. En el eje Z se registran aceleraciones que llegan incluso por debajo de los -200 m/s<sup>2</sup>, lo que sugiere impactos o cambios bruscos de dirección, y una zona de caída abrupta que podría estar asociada con una fase de microgravedad, cuando el celular se encuentra momentáneamente en el aire. Este

experimento permite analizar fenómenos como la caída libre, la rotación del cuerpo y el impacto, proporcionando un ejemplo claro de cómo varía la aceleración en un sistema no controlado. En el segundo experimento, se colocó el celular en la llanta de una bicicleta y se utilizó la opción de medición de aceleración centrípeta junto con la velocidad angular. Al iniciar la rotación de la rueda, los datos muestran un aumento progresivo tanto en aceleración como en velocidad angular. La aceleración alcanza un pico cercano a los 17 m/s<sup>2</sup> y luego desciende paulatinamente, mientras que la velocidad angular llega a aproximadamente 9 rad/s antes de disminuir. Las gráficas presentan una forma suave y continua, característica de un movimiento circular uniforme y controlado. Este experimento es ideal para observar cómo la aceleración centrípeta depende del cuadrado de la velocidad angular y para comprender la relación entre los movimientos circulares y las fuerzas involucradas.

### IV. CONCLUSIÓN

La aplicación Phyphox permite realizar experimentos físicos de forma práctica y accesible utilizando los sensores del teléfono móvil. Su capacidad para registrar aceleración, velocidad angular y otros parámetros, junto con la posibilidad de exportar los datos para analizarlos en MATLAB o Python, la convierte en una herramienta muy útil tanto en entornos educativos como en trabajos experimentales básicos. Su uso promueve una comprensión más activa y visual de los conceptos físicos.

En el primer experimento, al lanzar el celular sobre una cama, se obtuvo un registro caótico de aceleraciones en los tres ejes, evidenciando movimientos bruscos, impactos y posibles momentos de microgravedad. Este comportamiento permitió observar de forma clara la dinámica de un cuerpo en rotación libre, así como los efectos de la caída y los rebotes, fenómenos que usualmente son difíciles de estudiar sin equipamiento especializado.

En contraste, el segundo experimento, con el celular girando dentro de la llanta de una bicicleta, mostró un movimiento circular más controlado. Las gráficas de aceleración centrípeta y velocidad angular fueron suaves y consistentes, lo que facilitó el análisis del movimiento rotacional. La comparación entre ambos experimentos evidenció cómo distintos tipos de movimiento generan patrones muy diferentes, y cómo herramientas como Phyphox pueden ser clave para observar, medir y entender fenómenos físicos reales con claridad.

### REFERENCIAS

- [1] Phyphox.org.Retrieved July 26,2025, from <https://phyphox.org/sensordb/>
- [2] iPhone 13 - Especificaciones técnicas. (n.d.). Apple Support. Retrieved July 26, 2025, from <https://support.apple.com/es-lamr/111871>
- [3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [4] Recuerdos de Max (n.d) Youtube Retrived July 26 2025 from <https://www.youtube.com/watch?v=cOxzeHyD49U>