

MEDICIÓN DE ACELERACIÓN SIN G Y GPS CON PHYPHOX

CARABALLO CARDENAS ANDRES FELIPE, UNIVERSIDAD MILITAR NUEVA GRANADA
est.andres.caraballo@unimilitar.edu.co

Resumen La presente práctica de laboratorio tuvo como objetivo evaluar los resultados obtenidos en los experimentos realizados con la aplicación Phyphox, instalada en un dispositivo Xiaomi Redmi 10 cuyo modelo es 2106111AL. Durante la práctica se emplearon el acelerómetro ST LIS2HH12, para medir la aceleración sin la influencia de la gravedad, y el módulo de ubicación GPS, con el fin de registrar parámetros de posición y velocidad en una trayectoria. Cabe destacar que el dispositivo utilizado también cuenta con otros sensores, como magnetómetro, sensor de luz ambiental y sensor de proximidad. Los datos obtenidos en cada escenario fueron exportados y analizados utilizando librerías de Python, así como en MATLAB, mostrando resultados consistentes y de fácil interpretación por medio de gráficas obtenidas en cada medida en función del tiempo.

I. INTRODUCCIÓN

Para el desarrollo de los experimentos se utilizó la aplicación móvil Phyphox, la cual permite realizar diversas mediciones aprovechando los sensores integrados en el dispositivo. En este caso, se emplearon el acelerómetro ST LIS2HH12, para registrar la aceleración sin la influencia de la gravedad y el módulo de ubicación GPS, con el fin de obtener datos de posición y velocidad en una trayectoria, para su análisis posteriormente.

II. PROCEDIMIENTO

A. Ficha Técnica del dispositivo

En esta sección se presentan las características del dispositivo, tanto a nivel de datos generales como de los sensores integrados, los cuales se detallan a continuación.

2106111AL	
Device	
Manufacturer	Xiaomi
Brand	Redmi
Sample size	3
Variants	1
Accelerometer	
Available	✓
Name	lis2hh12
Vendor	ST
Range	78 m/s ²
Resolution	0.0012 m/s ²
Rate	396.3 Hz
Average	9.963 m/s ²
Standard deviation	0.051 m/s ²
Acceleration (without g)	
Available	✓
Name	LINEARACCEL
Vendor	MTK
Range	39 m/s ²
Resolution	0.0012 m/s ²
Rate	396.3 Hz
Standard deviation	0.043 m/s ²
Gyroscope	
Available	✓
Name	virtual_gyro
Vendor	ST
Range	34.9 rad/s
Resolution	0.0011 rad/s
Rate	396.3 Hz
Standard deviation	0.037 rad/s

Figura 1. Información técnica de los sensores Acelerómetro, Acelerómetro con g y Giroscopio

Magnetometer	Pressure
Available	✓
Name	akm09918
Vendor	AKM
Range	4912 µT
Resolution	0.15 µT
Rate	50.0 Hz
Standard deviation	0.25 µT
Temperature	Pressure
Available	✗
Humidity	Pressure
Available	✗
Light	Proximity
Available	✓
Name	tmd2755_l
Vendor	AMS
Range	65535 lx
Resolution	1.0 lx
Proximity	Proximity
Available	✓
Name	tmd2755_p
Vendor	AMS

Figura 2. Información técnica de los sensores Magnetómetro, Luz y Proximidad

Como se observa en las figuras 1 y 2 se detallan las especificaciones técnicas de los sensores integrados en el dispositivo. Uno de los sensores usados es el acelerómetro (sin gravedad) ST LS2HH12, que cuenta con un rango de medición de 78 m/s² lo que indica un límite máximo de aceleración que el sensor puede registrar de forma precisa, así mismo la resolución de 0.0012 m/s² indica que la precisión es mínima, lo que referencia que el sensor solo puede distinguir cambios extremadamente pequeños en la aceleración.

Por otra parte, el sensor GPS del dispositivo utilizado cuenta con soporte GNSS, lo que permite conectarse a sistemas múltiples de posicionamiento satelital, mejorando la precisión y confiabilidad en las mediciones de posición y velocidad.

B. Instalación de la Aplicación Phophox

En segundo apartado, se instala el aplicativo móvil el cual se encuentra en la tienda de aplicaciones del teléfono, en este caso Play Store como se muestra en la figura adjunta.

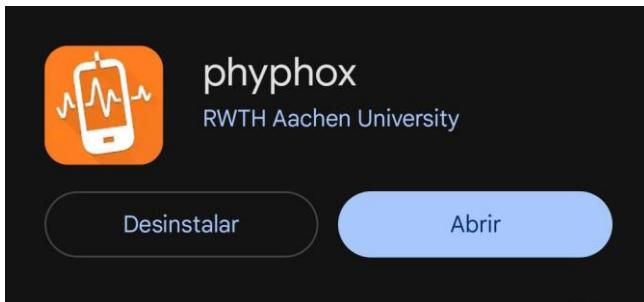


Figura 3. Visualización de instalación de Phophox

C. Interfaz de inicio de Phophox



Figura 4. Interfaz de inicio de Phophox

En la imagen anterior, se muestra la interfaz de inicio de la aplicación la cual esta dividida en categorías, en donde la primera corresponde a los sensores, que son aquellos en los que se pueden realizar mediciones físicas como:

- Aceleración (sin g) y aceleración (con g): Captura los datos del acelerómetro en dos versiones diferentes las cuales son: una que excluye el valor de la gravedad y la otra la tiene en cuenta el valor que es (9,8m/s).
- El giroscopio (velocidad de rotación): Es la que se encarga de medir la rotación del dispositivo en el espacio.
- Luz: Mide la intensidad de luz del ambiente con el sensor de luminosidad.
- Magnetómetro: Registra el campo magnético del entorno.
- Presión: Detecta la presión atmosférica
- Ubicación (GPS): Registra coordenadas geográficas utilizando el sistema de posicionamiento global.

Así mismo, la aplicación Phophox incluye otras categorías, como la acústica que permite emplear el uso del micrófono y altavoz del teléfono para realizar experimentos con sonido. Esta y otras funciones hacen que este aplicativo móvil se convierta en una herramienta de fácil manejo y accesible para interactuar con diferentes fenómenos físicos.

D. Experimento de Aceleración sin g (Sin gravedad)

En este segmento, se presenta uno de los experimentos realizados en el laboratorio, el cual fue la aceleración sin g. Este experimento tuvo finalidad en registrar y analizar los diferentes movimientos del dispositivo móvil en las tres direcciones del espacio X, Y y Z, el propósito de este experimento es obtener los datos únicamente del movimiento del teléfono, excluyendo de manera constante el efecto de la gravedad.

A continuación, se describen los pasos seguidos para la recolección de datos de esta medición

- Tabla de Mediciones

En este primer apartado, la tabla de datos se obtuvo a partir de un archivo .CSV que fue guardado usando tabulación como separador de columnas y punto como separador decimal. Lo que significa que los números decimales se escriben con punto y cada valor de las variables se organiza en columnas usando tabulaciones.

Una vez generado el archivo con las mediciones, se carga a Matlab y se da en la opción importación de datos como se muestra a continuación en la figura con cuadro rojo:

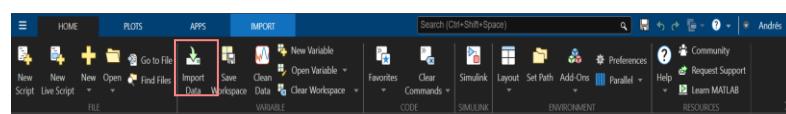


Figura 5. Ubicación del ícono importación de datos

Después de hacer clic en la opción de importar datos y cargar el archivo .CSV, se ajustan algunas configuraciones básicas. Primero, se selecciona que las columnas están delimitadas por tabulación, luego se seleccionan todas las columnas de la tabla, se le asigna un nombre, y por último se confirma la importación. Estas opciones aparecen resaltadas con recuadros de colores, como se muestra en la figura a continuación.

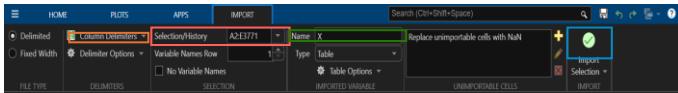


Figura 6. Opciones de modificación en la pestaña de importación

Una vez finalizado el paso anterior, se muestra la tabla con los cambios generados a continuación.

	B	C	D	E
	LinearAcce...	LinearAcce...	LinearAcce...	AbsoluteA...
	Number	Number	Number	Number
Time (s)	Linear Acce...	Linear Acce...	Linear Acce...	Absolute ac...
0.066264	-0.12354	0.27347	0.20101	0.36118
0.071191	-0.12918	0.30667	0.24208	0.41151
0.076119	-0.12088	0.31593	0.2607	0.42707
0.081047	-0.11556	0.33657	0.29113	0.45978
0.085974	-0.10949	0.35243	0.31422	0.4847
0.090901	-0.10949	0.35243	0.31422	0.4847
0.095828	-0.10949	0.35243	0.31422	0.4847
0.10075	2.2645	-2.0805	1.0625	3.2535
0.10568	2.2645	-2.0805	1.0625	3.2535
0.11061	2.2645	-2.0805	1.0625	3.2535
0.11554	2.2645	-2.0805	1.0625	3.2535
0.12046	2.077	-1.9582	0.65033	2.9277
0.12539	2.077	-1.9582	0.65033	2.9277
0.13032	2.077	-1.9582	0.65033	2.9277
0.13524	2.077	-1.9582	0.65033	2.9277
0.14017	2.3888	-1.8607	0.34751	3.0478

Tabla 1. Datos de aceleración sin g registrados en Phyxox

- Código de Matlab para graficar la aceleración sin g en los ejes x, y, z y aceleración Absoluta

En esta sección se adjunta el código el cual se utilizó para graficar cada variable en función del tiempo, como se muestra a continuación

```
figure;
t=X.Time_s_;
subplot(4,1,1);
x1=X.LinearAccelerationX_m_s_2_;
plot(t,x1,'Color','red');
title('Tiempo (S) VS ACELERACION EN X')
xlabel('tiempo')
ylabel('Ace x')
```

```
y1=X.LinearAccelerationY_m_s_2_;
subplot(4,1,2);
plot(t,y1,'Color','green');
title('Tiempo (S) VS ACELERACION EN Y')
xlabel('tiempo')
ylabel('Ace y')
```

```
z1=X.LinearAccelerationZ_m_s_2_;
subplot(4,1,3);
plot(t,z1,'Color','black');
title('Tiempo (S) VS ACELERACION EN Z')
xlabel('tiempo')
ylabel('Ace z')
```

```
a1=X.AbsoluteAcceleration_m_s_2_;
subplot(4,1,4);
plot(t,a1,'Color','blue');
title('Tiempo (S) VS ACELERACION EN ABS')
xlabel('tiempo')
ylabel('Ace en ABS')
```

Listing 1. Código en MATLAB para graficar aceleraciones en los ejes X, Y, Z y la aceleración absoluta.

Este código empieza con la creación de una variable de tiempo general, tomando la columna correspondiente al tiempo (en segundos) desde la tabla de datos llamada X. Luego, se crean variables para cada componente de la aceleración sin g, es decir, para los ejes X, Y, Z y la aceleración absoluta. Cada una de estas variables toma los datos directamente de su columna dentro de la tabla.

Una vez comentada la parte anterior del código, se ejecuta la simulación, la cual genera como resultado las gráficas generadas por medio de línea de código.

- Graficas

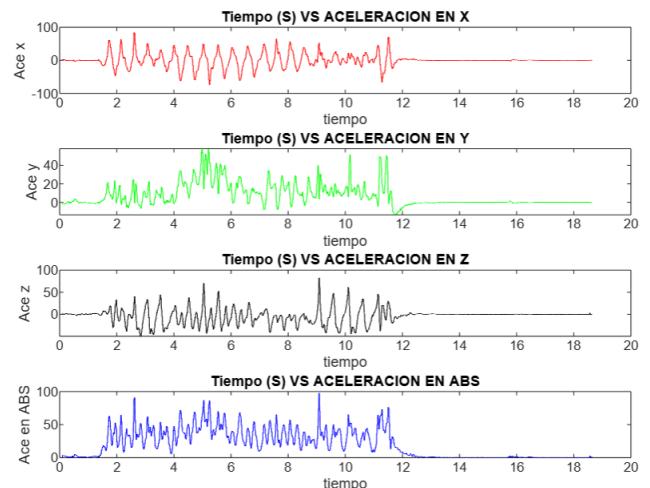


Figura 7. Graficas de Acceleraciones en cada eje de Matlab

El análisis que se tiene de las gráficas es:

- Grafica roja (Tiempo vs Aceleración en X)

En esta sección se observa que las aceleraciones presentan oscilaciones marcadas desde el primer segundo hasta aproximadamente los 11.8 segundos. Durante ese intervalo, se destacan varios picos de aceleración, siendo uno de los más relevantes el que ocurre alrededor de los 2.6 segundos, con un valor cercano a 81.1 m/s^2 , lo que indica un movimiento fuerte o una sacudida importante del dispositivo en ese instante.

Por otro lado, también se aprecia un pico negativo hacia los 10.9 segundos, con un valor aproximado de -3.81 m/s^2 , que refleja un cambio de dirección o una desaceleración más leve en comparación con los picos anteriores.

Finalmente, a partir de los 12 segundos, la señal se estabiliza alrededor de cero, lo que sugiere que el dispositivo dejó de moverse o fue colocado en reposo, sin registrar aceleraciones significativas en ese eje.

- Grafica verde (Tiempo Vs Aceleración en Y)

- Aceleración mayormente positiva y con varios picos entre 5 y 12 segundos.

- También se estabiliza después del segundo 12.

- Llama la atención un pico brusco alrededor del segundo 10, lo que puede ser un movimiento más fuerte.

- Grafica negra (Tiempo Vs Aceleración en Z)

- Mayor variabilidad desde el segundo 2 hasta el 10.
- Picos notables en valores positivos y negativos, lo que sugiere movimientos hacia arriba y hacia abajo del dispositivo.
- Despues del segundo 11, también se estabiliza.

- Grafica Azul (Tiempo VS aceleración Absoluta)

- Esta gráfica combina la aceleración de los tres ejes y muestra la intensidad total del movimiento.
- Se ve una actividad intensa entre los segundos 2 y 11, con muchos picos por encima de 60 m/s^2 .
- Luego decae hasta casi cero, lo que indica que el teléfono dejó de moverse.

- Código de Python para graficar la aceleración sin g en los ejes x, y, z y aceleración Absoluta

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

```
L = pd.read_csv('Raw Data.csv', sep='\t')
```

```
L.columns = L.columns.str.strip().str.replace(" ", "", regex=False)
```

```
print("Columnas detectadas:", L.columns.tolist())
```

```
t = L['Time (s)']
x1 = L['Linear Acceleration x (m/s^2)']
y1 = L['Linear Acceleration y (m/s^2)']
z1 = L['Linear Acceleration z (m/s^2)']
a1 = L['Absolute acceleration (m/s^2)']
```

```
plt.figure(figsize=(10, 8))
```

```
plt.subplot(4, 1, 1)
plt.plot(t, x1, color='red')
plt.title('TIEMPO (S) VS ACCELERACIÓN EN CADA CASO')
plt.xlabel('Tiempo (s)')
plt.ylabel('Acel en X')
```

```
plt.subplot(4, 1, 2)
plt.plot(t, y1, color='green')
plt.xlabel('Tiempo (s)')
plt.ylabel('Acel en Y')
```

```
plt.subplot(4, 1, 3)
plt.plot(t, z1, color='black')
plt.xlabel('Tiempo (s)')
plt.ylabel('Acel en Z')
```

```
plt.subplot(4, 1, 4)
plt.plot(t, a1, color='blue')
plt.xlabel('Tiempo (s)')
plt.ylabel('Acel ABS')
```

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

Listing 2. Código en Python para graficar aceleraciones en los ejes X, Y, Z y la aceleración absoluta.

- Graficas de Python

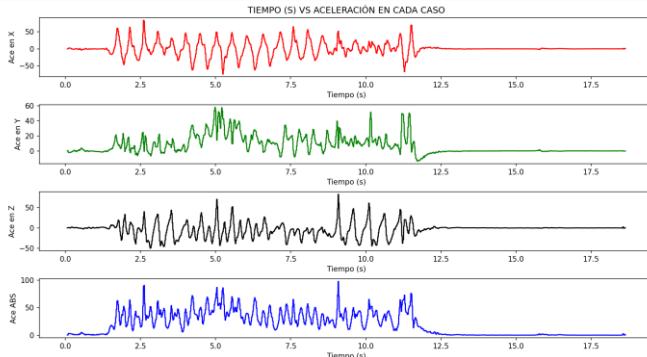


Figura 8. Graficas de Aceleraciones en cada eje de Python

E. Experimento de Ubicación (GPS)

En esta parte del trabajo se presenta uno de los experimentos realizados de forma individual: el experimento de ubicación por GPS. El objetivo principal fue registrar y analizar cómo se comportan distintas variables relacionadas con la posición y el movimiento del dispositivo durante un recorrido al aire libre. Entre las variables observadas se encuentran:

- **Altitud:** indica la altura a la que se encuentra el dispositivo respecto al nivel del mar.
- **Latitud y longitud:** definen la ubicación geográfica exacta del dispositivo sobre la superficie terrestre.
- **Velocidad:** mide qué tan rápido se desplaza el dispositivo a lo largo del trayecto.
- **Dirección:** señala hacia dónde se está moviendo el dispositivo, generalmente en grados (por ejemplo, norte, sur, etc.).
- **Distancia:** es la longitud total recorrida durante el experimento.
- **Precisión horizontal (horizontal accuracy):** se refiere al margen de error en la ubicación medida en el plano horizontal (latitud y longitud).
- **Precisión vertical (vertical accuracy):** indica el margen de error en la medición de la altitud.
- **Satélites:** representan la cantidad de satélites GPS que están conectados y permiten calcular la ubicación con mayor o menor precisión.

A continuación, se describen los pasos seguidos para la recolección de datos de esta medición.

- Tabla de Mediciones

Se repite el proceso anterior y se obtiene la siguiente la siguiente tabla la cual están todas las variables mencionadas anteriormente en función del tiempo.

A	B	C	D	E	F	G	H	I	J	K
Time_s	Latitude	Longitude	Altitude_m	AltitudeW...	Speed_m_s	Direction	Distance_k...	Horizontal...	VerticalAcc...	Satellites
Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
1	0.58175	4.6331	-74.1645	2545.4219	2566.7002	NaN	NaN	0	40.481	1
2	1.8012	4.6331	-74.1645	2545.4219	2566.7002	NaN	NaN	0.039883	49.014	2.70866
3	3.8628	4.6334	-74.1643	2569.622	2590.9	0	0	0.048133	72	213.4
4	9.3628	4.6334	-74.1642	2562.2221	2583.5	0	0	0.055722	61.9	151.4
5	10.3628	4.6335	-74.1642	2564.0221	2585.9	5.2738	45.85	0.061465	7.8	143.8
6	11.3628	4.6335	-74.1641	2564.6222	2585.9	0	45.85	0.066698	7.5	141.5
7	12.3628	4.6335	-74.1641	2563.5222	2584.8	5.1235	48.23	0.07236	7.2	135
8	13.3628	4.6335	-74.164	2564.0223	2585.3	5.0988	49.05	0.078259	7.1	133.9
9	14.3628	4.6336	-74.164	2563.0223	2584.3	4.6201	49.79	0.081812	5.8	115.5
10	15.3628	4.6336	-74.164	2559.2224	2580.5	4.4266	49.33	0.084541	4.5	105
11	16.3628	4.6336	-74.1639	2557.1224	2578.4	3.285	52.1	0.088056	4.3	96.7
12	17.3628	4.6336	-74.1639	2564.4224	2587.7	0	52.1	0.089813	3.8	94.5
13	18.3628	4.6336	-74.1639	2564.4224	2585.7	1.103	43.34	0.091656	3.5	93.9
14	19.3628	4.6336	-74.1639	2561.3224	2582.6	0.67532	37.47	0.092888	3.4	93.8
15	20.3628	4.6336	-74.1639	2559.5224	2580.8	0.23214	49.81	0.094362	3.2	93.8
16	21.3628	4.6336	-74.1639	2558.5224	2579.8	0.31604	59.45	0.094624	3.2	77.1
17										13

Tabla 2. Datos de la ubicación (GPS) registrados en Phyphox

- Código de Matlab

En esta sección se adjunta el código el cual se utilizó para graficar cada variable en función del tiempo, como se muestra a continuación

```

figure;
t=P.Time_s_;
subplot(2,1,1);
l1=P.Latitude__;
plot(t,l1,'Color','blue')
title('LATITUD VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Latitud(°)')
grid on

long=P.Longitude__;
subplot(2,1,2);
plot(t,long,'Color','black')
title('LONGITUD VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Longitud(°)')
grid on

figure;
alitim=P.Altitude_m__;
subplot(2,1,1);
plot(t,alitim,'Color','red')
title('ALTITUD VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Altitud(m)')
grid on

altiw=P.AltitudeWGS84_m__;
subplot(2,1,2);
plot(t,altiw,'Color','green')
title('ALTITUD VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Altitud(w)')
grid on

```

```

figure;
velocidad=P.Speed_m_s_;
subplot(2,1,1);
plot(t,velocidad, 'Color','blue')
title('VELOCIDAD VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Velocidad(m/s)')
grid on

```

```

direccion=P.Direction__;
subplot(2,1,2);
plot(t,direccion, 'Color','red')
title('DIRECCION VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Direccion(°)')
grid on

```

```

figure;
distancia=P.Distance_km_;
subplot(2,1,1);
plot(t,distancia, 'Color','magenta')
title('DISTANCIA VS TIEMPO')
xlabel('Tiempo')
ylabel('Distancia (KM)')
grid on

```

```

horizontal=P.HorizontalAccuracy_m_;
subplot(2,1,2);
plot(t,horizontal, 'Color','cyan')
title('Horizontal VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Horizontal(m)')
grid on

```

```

figure;
vertical=P.VerticalAccuracy_m_;
subplot(2,1,1);
plot(t,vertical, 'Color', 'blue')
title('VERTICAL VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Vertical(m)')
grid on

```

```

satelites=P.Satellites;
subplot(2,1,2);
plot(t,satelites, 'Color', 'blue')
title('SATELITES VS TIEMPO')
xlabel('Tiempo(s)')
ylabel('Satelites')
grid on

```

Listing 3. Código de Matlab para cada variable del GPS

- Graficas de Matlab

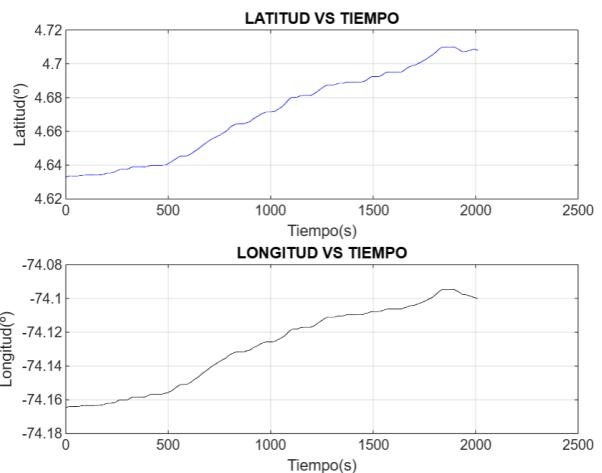


Figura 9. Graficas de Latitud (°) y Longitud (°)

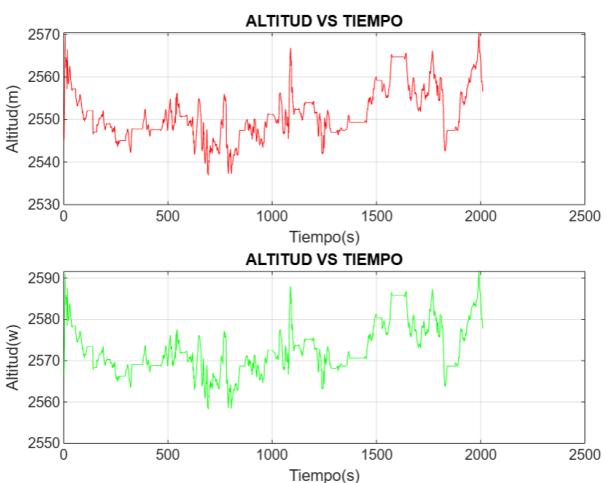


Figura 10. Graficas de Altitud (m) y Altitud(W)

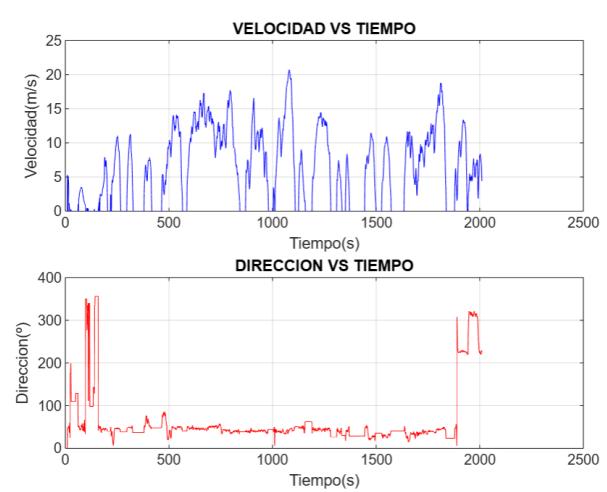


Figura 11. Graficas de Velocidad (m/s) y Dirección (°)

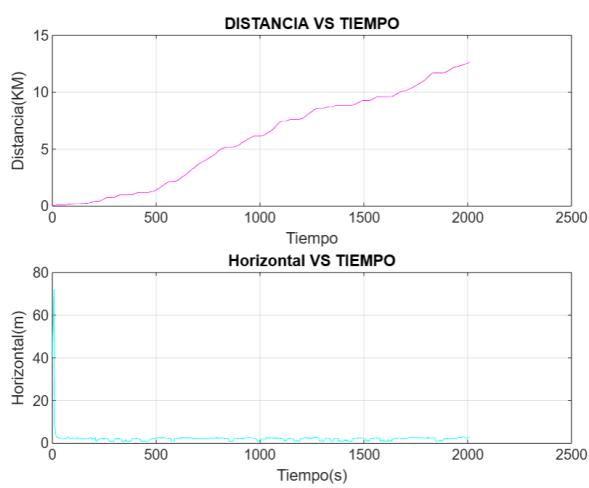


Figura 12. Graficas de Distancia (k/m) y Horizontal(m)

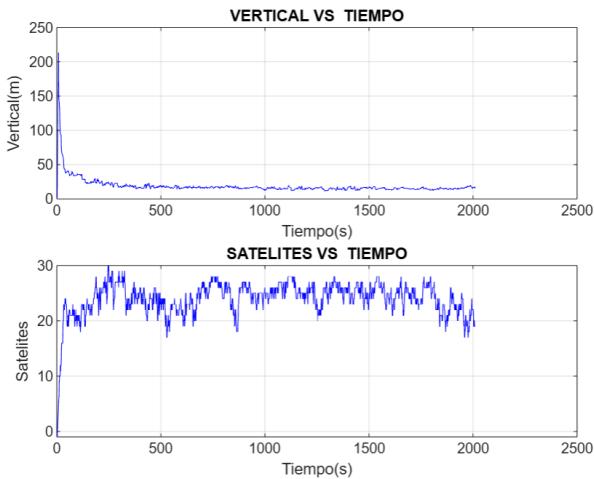


Figura 13. Graficas de Vertical(m) y Satelites

- Código de Python

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

P = pd.read_csv('Raw Data-1.csv', sep='\t')
P.columns      = P.columns.str.strip().str.replace(" ", "")
P.replace("", "", regex=False)

t = P['Time (s)']

# 1. LATITUD y LONGITUD
plt.figure()
plt.subplot(2,1,1)
plt.plot(t, P['Latitude (°)'], color='blue')
plt.title('LATITUD VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Latitud(°)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()

plt.subplot(2,1,2)
plt.plot(t, P['Longitude (°)'], color='black')
plt.title('LONGITUD VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Longitud(°)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()
```

```
plt.tight_layout()
```

```
plt.subplot(2,1,2)
plt.plot(t, P['Longitude (°)'], color='black')
plt.title('LONGITUD VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Longitud(°)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()
```

2. ALTITUD y ALTITUD WGS84

```
plt.figure()
plt.subplot(2,1,1)
plt.plot(t, P['Altitude (m)'], color='red')
plt.title('ALTITUD VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Altitud(m)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()
```

```
plt.subplot(2,1,2)
plt.plot(t, P['Altitude WGS84 (m)'], color='green')
plt.title('ALTITUD VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Altitud(w)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()
```

3. VELOCIDAD y DIRECCIÓN

```
plt.figure()
plt.subplot(2,1,1)
plt.plot(t, P['Speed (m/s)'], color='blue')
plt.title('VELOCIDAD VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Velocidad(m/s)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()
```

```
plt.subplot(2,1,2)
plt.plot(t, P['Direction (°)'], color='red')
plt.title('DIRECCION VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Direccion(°)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()
```

4. DISTANCIA y HORIZONTAL ACCURACY

```
plt.figure()
plt.subplot(2,1,1)
plt.plot(t, P['Distance (km)'], color='magenta')
plt.title('DISTANCIA VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Distancia(KM)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()

plt.subplot(2,1,2)
plt.plot(t, P['Horizontal Accuracy (m)'], color='cyan')
plt.title('Horizontal VS TIEMPO')
plt.xlabel('Tiempo(s)')
```

```

plt.ylabel('Horizontal(m)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()

# 5. VERTICAL ACCURACY y SATÉLITES
plt.figure()
plt.subplot(2,1,1)
plt.plot(t, P['Vertical Accuracy (m)'], color='blue')
plt.title('VERTICAL VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Vertical(m)')
plt.grid(True, linewidth=1.5)
plt.tight_layout()

plt.subplot(2,1,2)
plt.plot(t, P['Satellites'], color='blue')
plt.title('SATÉLITES VS TIEMPO')
plt.xlabel('Tiempo(s)')
plt.ylabel('Satélites')
plt.grid(True, linewidth=1.5)
plt.tight_layout()

plt.show()

```

Listing 4. Código de Python para cada variable del GPS

- Graficas de Python

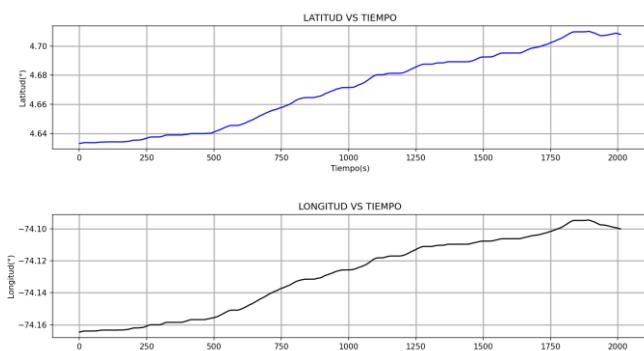


Figura 14. Graficas de Latitud ($^{\circ}$) y Longitud ($^{\circ}$)

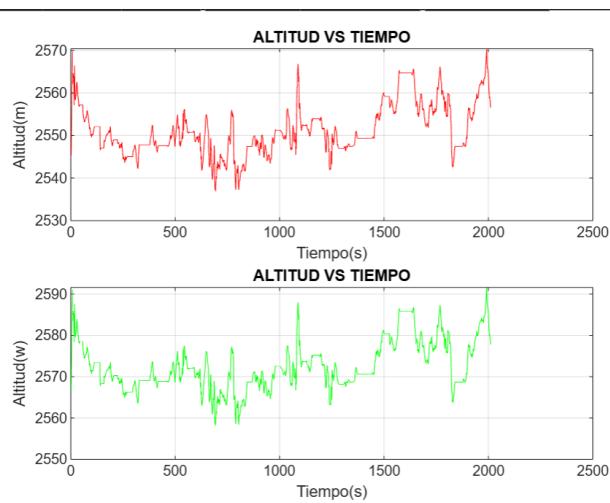


Figura 15. Graficas de Altitud (m) y Altitud(W)

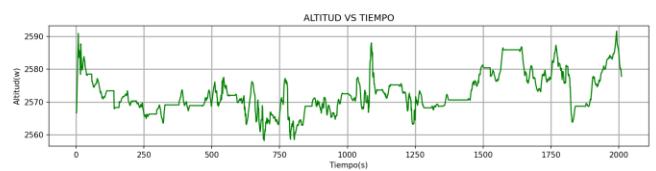
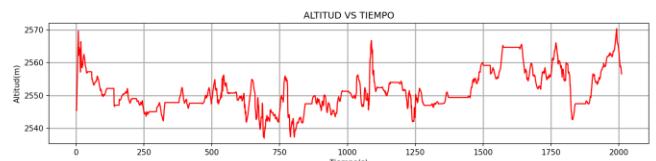


Figura 15. Graficas de Altitud (m) y Altitud(W)

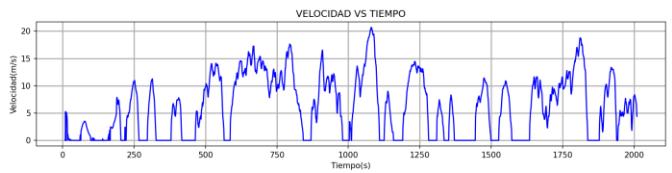


Figura 16. Graficas de Velocidad (m/s) y Dirección ($^{\circ}$)

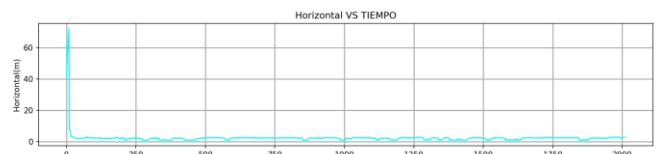
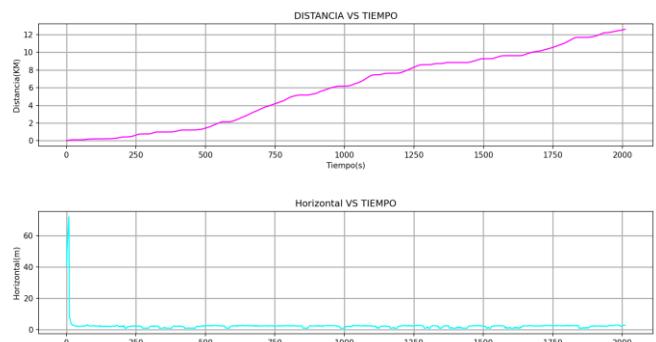


Figura 17. Graficas de Distancia (k/m) y Horizontal(m)

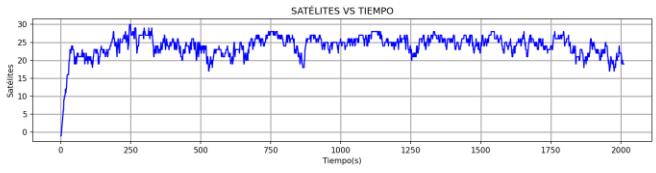
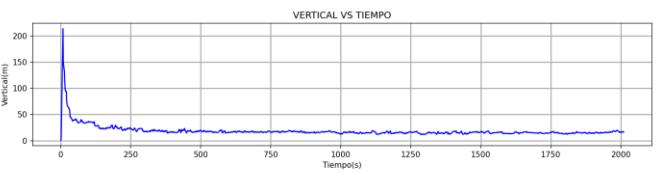


Figura 18. Graficas de Vertical(m) y Satelites

- Código adicional de python para el mapa de recorrido

```

import pandas as pd
import folium
import webbrowser

df = pd.read_csv("Raw Data-1.csv", sep="\t", quotechar="")

df = df.dropna(subset=['Latitude (°)', 'Longitude (°)'])

start_lat = df['Latitude (°)'].iloc[0]
start_lon = df['Longitude (°)'].iloc[0]

mapa = folium.Map(location=[start_lat, start_lon],
                  zoom_start=18)

coordenadas = list(zip(df['Latitude (°)'], df['Longitude (°)']))

folium.PolyLine(coordenadas, color='blue').add_to(mapa)

for lat, lon in coordenadas:
    folium.CircleMarker(location=[lat, lon], radius=3,
                        color='red').add_to(mapa)

mapa_path = 'ruta_gps_phypox.html'
mapa.save(mapa_path)
print(f"Mapa guardado como {mapa_path}")

webbrowser.open(mapa_path)

```

- Grafica del mapa

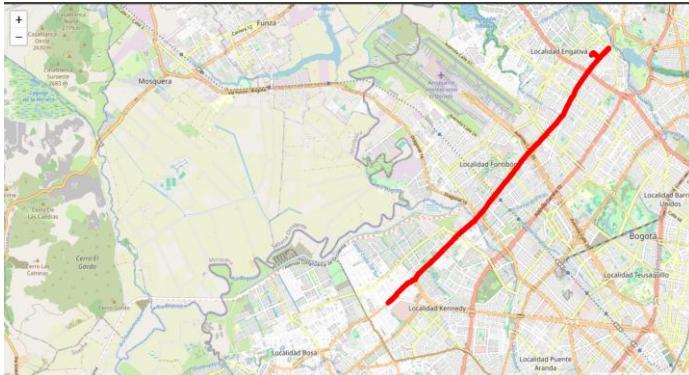


Figura 19. Mapa del recorrido marcado en rojo

III. CONCLUSIONES

- El experimento permitió comprobar que un dispositivo móvil puede registrar con bastante precisión su ubicación en tiempo real mediante el uso del sistema GPS.
- Se evidenció cómo variables como latitud, longitud y altitud cambian continuamente durante un recorrido, reflejando con fidelidad la trayectoria del usuario.
- La cantidad de satélites conectados influye directamente en la calidad de los datos obtenidos: a

mayor número de satélites, más precisa es la ubicación.

- El experimento de aceleración sin g permitió aislar el efecto de la gravedad para centrarse únicamente en los movimientos reales del dispositivo móvil.
- El uso de esta función sin g (gravedad) permite obtener datos más limpios y representativos del movimiento real, ideales para estudios de dinámica, caídas o trayectorias.
- Al registrar la aceleración en los tres ejes (X, Y, Z), se pudo observar cómo responde el sensor a distintas acciones físicas como sacudidas, desplazamientos o cambios de dirección.

REFERENCIAS

- [1] PHYPHOX. "PHYPHOX SENSOR DATABASE". PHYPHOX – PHYSICAL PHONE EXPERIMENTS. [EN LINEA]. DISPONIBLE: [HTTPS://PHYPHOX.ORG/SENSORDB/](https://phypox.org/sensordb/)
- [2] INNOVANDO ANDO. (2020, 15 DE OCTUBRE). *CÓMO USAR PHYPHOX: INTRODUCCIÓN BÁSICA* [VIDEO]. YOUTUBE. [HTTPS://WWW.YOUTUBE.COM/WATCH?V=YMHYLEX5X50](https://www.youtube.com/watch?v=YMHYLEX5X50)
- [3] MATHWORKS. (2024). *MATLAB (VERSIÓN R2024A)* [SOFTWARE]. THE MATHWORKS, INC. [HTTPS://WWW.MATHWORKS.COM/PRODUCTS/MATLAB.HTML](https://www.mathworks.com/products/matlab.html)
- [4] AIVAR ANNAMAA. (2024). *THONNY (VERSIÓN 4.1.4)* [SOFTWARE]. [HTTPS://THONNY.ORG/](https://thonny.org/)