

# Procesamiento de Datos de Sensores Móviles mediante Phyphox, MATLAB y Python

Trabajo realizado por Helmuth Jefferson Rodríguez Rincón presentado al Ing Jose De Jesus Rugeles Uribe para la materia de Comunicaciones Digitales

**Resumen** - Este informe presenta el desarrollo de dos prácticas experimentales realizadas mediante la aplicación Phyphox, orientadas a la adquisición y análisis de datos provenientes de sensores integrados en dispositivos móviles. La primera práctica consistió en el estudio de la aceleración lineal del dispositivo durante el movimiento usando el sensor LSM6DSOTR, mientras que la segunda abordó la medición de la intensidad de luz utilizando el sensor TCS3701. En ambos casos, los datos fueron exportados en formato .csv y procesados con los lenguajes de programación Python y MATLAB para generar gráficas que permitieran visualizar el comportamiento de las variables físicas involucradas. El análisis incluyó la identificación automática de valores máximos y mínimos, así como la validación de la funcionalidad de los sensores en condiciones controladas. Los resultados obtenidos demuestran la viabilidad del uso de herramientas accesibles como Phyphox junto con entornos de programación para la realización de prácticas de laboratorio portátiles, precisas y didáctica.

## 1. INTRODUCCION

En esta práctica de laboratorio se utilizó la aplicación Phyphox en un dispositivo móvil Samsung Galaxy A53 (modelo: SM-A536E/DS) para la realización de dos experimentos. En el primero, se registró la aceleración en los tres ejes (X, Y, Z) utilizando el acelerómetro integrado. En el segundo, se midió la intensidad lumínica mediante el sensor de luz ambiental del dispositivo. Posteriormente, los datos obtenidos fueron exportados en formato .csv para su procesamiento y visualización mediante los entornos de programación MATLAB y Python. Antes de abordar el análisis de los resultados, se presenta una breve descripción de los sensores disponibles en el dispositivo móvil utilizado.

| Sensor                   | Nombre/modelo                       | Fabricante               | Rango                    | Resolucion              | Frecuencia | Disponibilidad |
|--------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------|------------|----------------|
| Acelerómetro             | LSM6DSOTR Accelerometer             | STMicroelectronics (STM) | $\pm 78 \text{ m/s}^2$   | $0.0024 \text{ m/s}^2$  | 467.7      | si             |
| Aceleración sin gravedad | Samsung Linear Acceleration Sensor  | Samsung Electronics      | $\pm 78 \text{ m/s}^2$   | $0.0024 \text{ m/s}^2$  | 125.9      | si             |
| Giroscopio               | LSM6DSOTR Gyroscope                 | STMicroelectronics (STM) | $\pm 17.5 \text{ rad/s}$ | $0.00061 \text{ rad/s}$ | 467.7      | si             |
| Magnetómetro             | AK09918C Magnetometer               | Asahi Kasei Microdevices | $\pm 4900 \mu\text{T}$   | $0.060 \mu\text{T}$     | 124.8      | si             |
| Sensor de luz            | TCS3701 Light                       | AMS Inc.                 | 0 – 60,000 lx            | 1.0 lx                  | -----      | si             |
| Proximidad               | Ear Hover Proximity Sensor (ProToF) | Samsung Electronics      | -----                    | -----                   | -----      | si             |
| Presión                  | -----                               | -----                    | -----                    | -----                   | -----      | no             |
| Temperatura              | -----                               | -----                    | -----                    | -----                   | -----      | no             |
| Humedad                  | -----                               | -----                    | -----                    | -----                   | -----      | no             |

Tabla 1 Phyphox Sensor Database[1]

En el desarrollo de esta práctica de laboratorio se emplearon dos sensores específicos del dispositivo Samsung Galaxy A53 (modelo **SM-A536E**), seleccionados según el tipo de experimento a realizar. Estos sensores permitieron registrar datos físicos relacionados con el movimiento y la luminosidad ambiental, los cuales fueron procesados posteriormente para su análisis gráfico.

### 1. Sensor de acelerómetro (LSM6DSOTR)

Para el experimento de aceleración, se utilizó el acelerómetro integrado, modelo **LSM6DSOTR**, fabricado por STMicroelectronics. Este sensor es capaz de medir la aceleración total del dispositivo en los tres ejes espaciales (X, Y, Z), incluyendo el efecto de la gravedad.

Sus principales características técnicas incluyen

- Rango:  $\pm 78 \text{ m/s}$
- Resolución:  $0.0024 \text{ m/s}$
- Frecuencia de muestreo: 467.7 H

Gracias a su alta frecuencia de muestreo y sensibilidad, este sensor es adecuado para registrar variaciones rápidas en el movimiento del dispositivo, permitiendo analizar la dinámica de este durante desplazamientos o vibraciones.

### 2. Sensor de luz ambiental (TCS3701)

Para el segundo experimento, orientado al análisis de la intensidad lumínica, se empleó el sensor de luz TCS3701, desarrollado por AMS Inc. Este sensor mide la cantidad de luz presente en el entorno, expresada en lux (lx), y es comúnmente utilizado para ajustar automáticamente el brillo de la pantalla en condiciones de iluminación cambiantes.

Sus especificaciones clave son:

- Rango: hasta 60,000 lx
- Resolución: 1.0 lx

Este sensor permitió registrar cómo varía la intensidad de luz en diferentes entornos o momentos del día, proporcionando datos útiles para evaluar condiciones de iluminación en espacios interiores o exteriores.

## 2. DESARROLLO

En esta sección se describen las herramientas utilizadas, la configuración experimental y el método seguido para la adquisición y procesamiento de los datos.

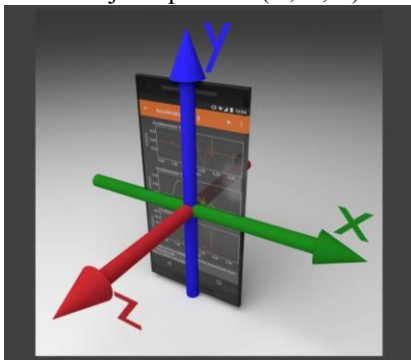
### 2.1 SOFTWARE Y HERRAMIENTAS EMPLEADAS

- **Phyphox:** Aplicación desarrollada por la Universidad RWTH Aachen, que permite acceder a los sensores del dispositivo móvil, registrar sus lecturas en tiempo real y exportarlas en distintos formatos, como .csv.[2]
- **Python (con librerías Matplotlib y Pandas):** Utilizado para el procesamiento de datos, generación de gráficas y análisis de tendencias.[3]
- **MATLAB:** Plataforma de cálculo numérico que permite importar datos, realizar operaciones estadísticas y representar gráficamente los resultados de forma detallada.[4]

### 2.2 PROCEDIMIENTO

#### Primera practica

Para el desarrollo del primer experimento, se utilizó la aplicación Phyphox instalada en el dispositivo móvil Samsung Galaxy A53. Se seleccionó el módulo correspondiente al acelerómetro, con el fin de registrar la aceleración total del dispositivo en los tres ejes espaciales (X, Y, Z)



**Ilustración 1: Orientación de los ejes X, Y y Z en el smartphone**

Una vez iniciada la interfaz del experimento en la aplicación, se procedió a colocar el dispositivo sobre una superficie plana para

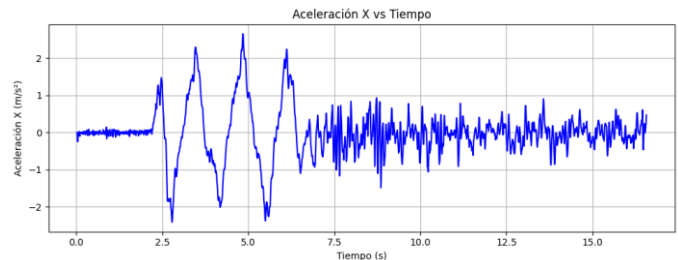
realizar una primera medición en reposo, lo que permitió verificar la calibración del sensor, pero también poder visualizar un valor de referencia que podría considerarse “estático” para que luego fuera más apreciable en el momento en el que en diferentes valores de tiempo el celular se movió en cada uno de los ejes

Posteriormente, se inició la recolección de datos mientras el dispositivo era sometido a distintos tipos de movimiento, tales como desplazamientos lineales e inclinado. Estas acciones permitieron registrar variaciones dinámicas en los valores de aceleración a lo largo del tiempo.

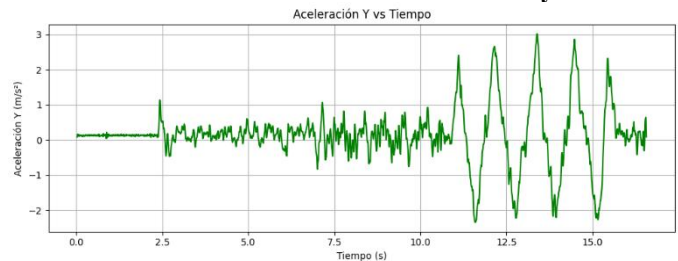
Durante el registro, el sensor **LSM6DSOTR**, con una frecuencia de muestreo de aproximadamente **467.7 Hz** y una resolución de **0.0024 m/s<sup>2</sup>**, El experimento se llevó a cabo durante un intervalo suficiente para garantizar la obtención de una muestra representativa de los cambios de aceleración en condiciones reales de movimiento.

Al finalizar la toma de datos, la aplicación permitió exportar los resultados en formato **.csv**, los cuales fueron posteriormente analizados mediante las plataformas **MATLAB** y **Python**, utilizando herramientas de visualización y procesamiento numérico para generar gráficas, calcular promedios y estudiar las variaciones en los tres ejes.

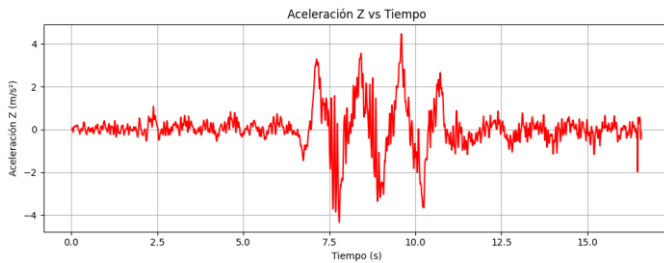
A continuación, se presentan las gráficas generadas mediante el script desarrollado en Python. En este caso, las Ilustraciones 2, 3 y 4 corresponden a aceleraciones lineales registradas en un solo eje, mientras que las Ilustraciones 5, 6 y 7 muestran las aceleraciones obtenidas con una inclinación del dispositivo, lo que resultó en una aceleración significativamente distinta a la lineal.



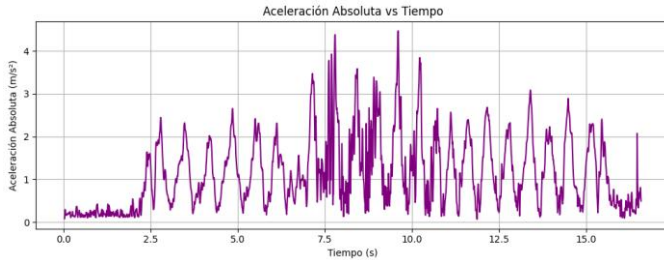
**Ilustración 2 Aceleración lineales en X Python**



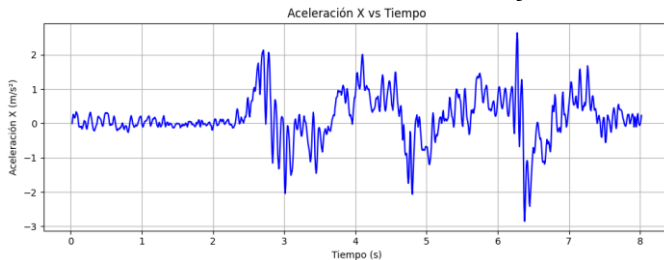
**Ilustración 3 Aceleración lineales en Y Python**



**Ilustración 4 Aceleración lineales en Z Python**



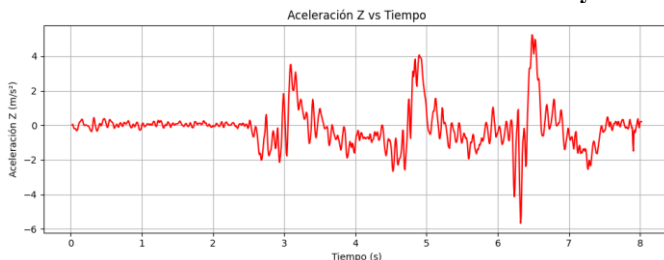
**Ilustración 5 Aceleración Absoluta Python**



**Ilustración 6 Aceleración con inclinación en X Python**



**Ilustración 7 Aceleración con inclinación en Y Python**



**Ilustración 8 Aceleración con inclinación en Z Python**



**Ilustración 9 Aceleración Absoluta con inclinación Python**

El script desarrollado en Python es el siguiente:

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

# Leer el archivo con separador de tabulación
datos = pd.read_csv("Raw Data.csv", sep='\t')

# Verificar las primeras filas
print("Primeras filas del archivo:")
print(datos.head())

# Extraer columnas
tiempo = datos["Time (s)"]
acc_x = datos["Linear Acceleration x (m/s^2)"]
acc_y = datos["Linear Acceleration y (m/s^2)"]
acc_z = datos["Linear Acceleration z (m/s^2)"]
acc_abs = datos["Absolute acceleration (m/s^2)"]

# --- Gráfica 1: Aceleración X ---
plt.figure(figsize=(10, 4))
plt.plot(tiempo, acc_x, color='blue')
plt.title('Aceleración X vs Tiempo')
plt.xlabel('Tiempo (s)')
plt.ylabel('Aceleración X (m/s²)')
plt.grid(True)
plt.tight_layout()
plt.show()

# --- Gráfica 2: Aceleración Y ---
plt.figure(figsize=(10, 4))
plt.plot(tiempo, acc_y, color='green')
plt.title('Aceleración Y vs Tiempo')
plt.xlabel('Tiempo (s)')
plt.ylabel('Aceleración Y (m/s²)')
plt.grid(True)
plt.tight_layout()
plt.show()

# --- Gráfica 3: Aceleración Z ---
plt.figure(figsize=(10, 4))
plt.plot(tiempo, acc_z, color='red')
plt.title('Aceleración Z vs Tiempo')
plt.xlabel('Tiempo (s)')
plt.ylabel('Aceleración Z (m/s²)')
plt.grid(True)
plt.tight_layout()
plt.show()
```

**Ilustración 10 script Python**

Para el análisis de los datos de aceleración obtenidos con el sensor, se empleó el lenguaje de programación Python en el entorno Google Colab. A continuación, se describen los comandos más relevantes utilizados en el procesamiento y visualización de la información:

**Importación de bibliotecas:** Se utilizaron las bibliotecas *pandas* para la manipulación de datos y *matplotlib.pyplot* para la generación de gráficas.

**Lectura del archivo .csv:** Se empleó el comando `pd.read_csv("Raw Data.csv", sep='\t')` para leer el archivo de datos separado por tabulaciones, que contenía información sobre el tiempo y las aceleraciones en los ejes X, Y, Z, así como la aceleración absoluta.

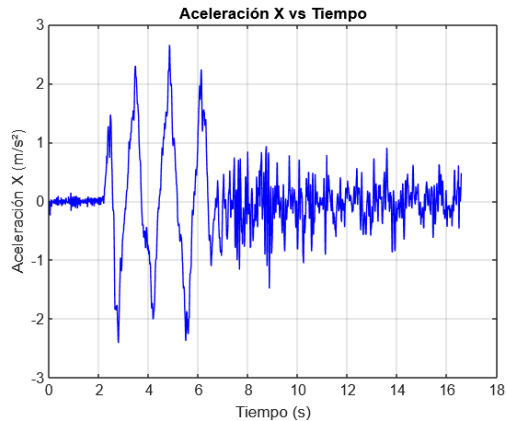
**Extracción de columnas específicas:** Se accedió a las columnas de interés mediante sus nombres, por ejemplo: `datos["Linear Acceleration x (m/s^2)"]` para la aceleración en el eje X.

**Visualización individual de las señales:** Para representar gráficamente cada componente de la aceleración de forma separada, se utilizó la función `plt.plot()` junto con etiquetas de tiempo y unidades correspondientes. A cada gráfico se le añadió un título, etiquetas en los ejes y una cuadrícula para facilitar la

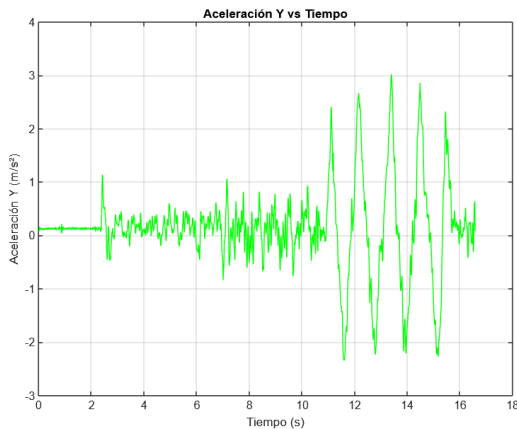
interpretación visual.

**Configuración de las figuras:** Se definió el tamaño de cada figura con `plt.figure(figsize=(10, 4))` y se aplicó `plt.tight_layout()` para mejorar la distribución del contenido en pantalla.

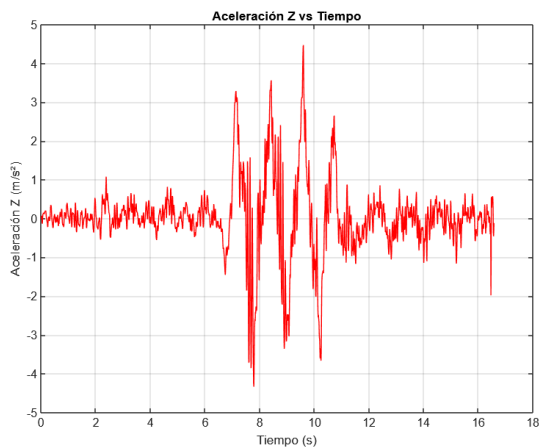
Posteriormente, se repitió este mismo procedimiento utilizando el programa MATLAB.



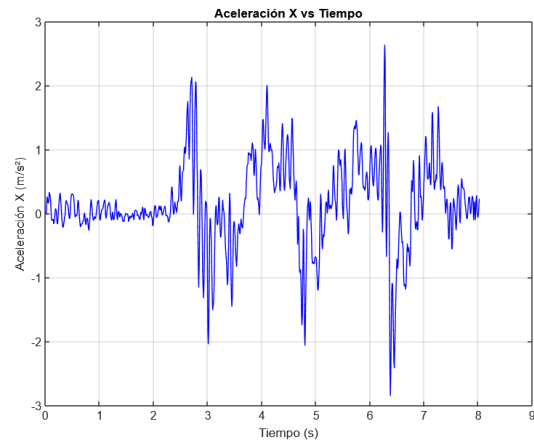
**Ilustración 11 Aceleración lineales en X Matlab**



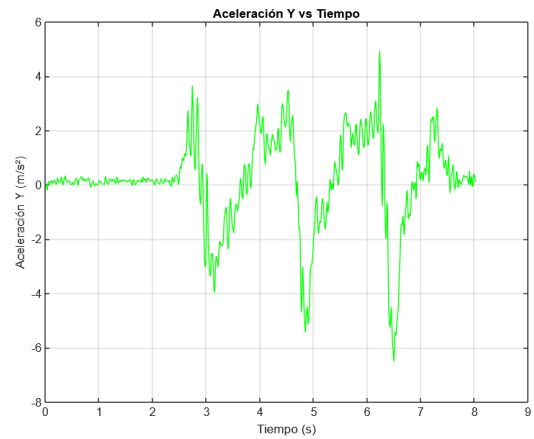
**Ilustración 12 Aceleración lineales en Y Matlab**



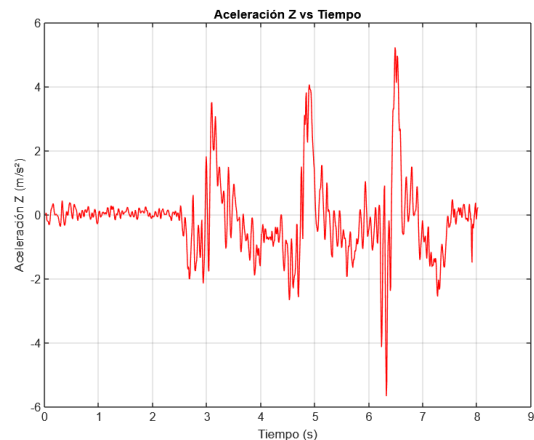
**Ilustración 13 Aceleración lineales en Z Matlab**



**Ilustración 14 Aceleración con inclinación en X Matlab**



**Ilustración 15 Aceleración con inclinación en Y Matlab**



**Ilustración 16 Aceleración con inclinación en Z Matlab**

El script desarrollado en Matlab es el siguiente:

```

>> % Leer el archivo CSV conservando los nombres originales de las columnas
filename = 'Raw Data.csv';
opts = detectImportOptions(filename, 'Delimiter', '\t', 'VariableNamingRule', 'preserve');
data = readtable(filename, opts);

% Extraer columnas por sus nombres exactos
t = data("Time (s)");
ax = data("Linear Acceleration x (m/s^2)");
ay = data("Linear Acceleration y (m/s^2)");
az = data("Linear Acceleration z (m/s^2)");
abs_acc = data("Absolute acceleration (m/s^2)");

% --- Gráfica 1: Aceleración X ---
figure;
plot(t, ax, 'b');
title('Aceleración X vs Tiempo');
xlabel('Tiempo (s)');
ylabel('Aceleración X (m/s^2)');
grid on;

% --- Gráfica 2: Aceleración Y ---
figure;
plot(t, ay, 'g');
title('Aceleración Y vs Tiempo');
xlabel('Tiempo (s)');
ylabel('Aceleración Y (m/s^2)');
grid on;

% --- Gráfica 3: Aceleración Z ---
figure;
plot(t, az, 'r');
title('Aceleración Z vs Tiempo');
xlabel('Tiempo (s)');
ylabel('Aceleración Z (m/s^2)');
grid on;

% --- Gráfica 4: Aceleración Absoluta ---
figure;
plot(t, abs_acc, 'm');
title('Aceleración Absoluta vs Tiempo');
xlabel('Tiempo (s)');
ylabel('Aceleración Absoluta (m/s^2)');
grid on;

```

**Ilustración 17 script Matlab**

Para el procesamiento y análisis de los datos de aceleración en MATLAB, se desarrolló un código que permitió leer un archivo .csv con separador por tabulaciones, extraer la información relevante y visualizarla de forma clara mediante gráficas. A continuación, se describen los comandos más importantes utilizados:

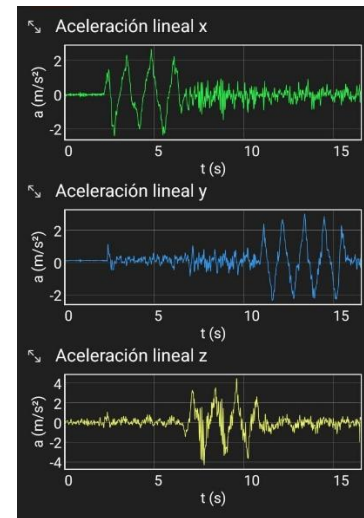
En primer lugar, se empleó el comando *detectImportOptions* con los parámetros '*Delimiter*', '*t*' y '*VariableNamingRule*', '*preserve*' para configurar correctamente la lectura del archivo *Raw Data.csv*, respetando los nombres originales de las columnas. Posteriormente, el comando *readtable* permitió importar los datos como una tabla de MATLAB.

Una vez cargada la información, se extrajeron las columnas correspondientes al tiempo y a las componentes de aceleración lineal en los ejes X, Y, Z, así como la aceleración absoluta. Esta extracción se realizó accediendo directamente a los nombres originales de las columnas, utilizando la sintaxis *data("Nombre de la columna")*.

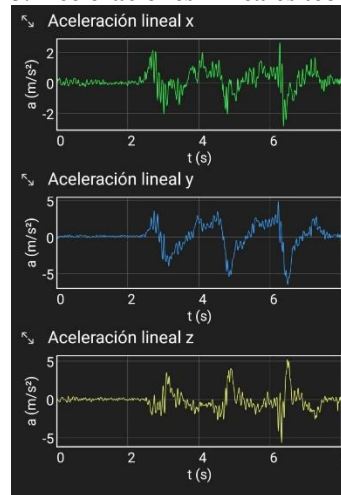
Finalmente, se generaron cuatro gráficas independientes, una por cada componente de aceleración. Para ello, se utilizó el comando *figure* para abrir una nueva ventana de gráfico, seguido de *plot* para trazar cada curva con un color distintivo. Se complementaron las gráficas con títulos (*title*), etiquetas en los ejes (*xlabel*, *ylabel*) y una cuadrícula (*grid on*) que facilitó la interpretación visual de los datos.

Este procedimiento permitió visualizar de forma clara el comportamiento dinámico de cada eje de aceleración a lo largo del tiempo, facilitando el análisis del movimiento registrado por el sensor.

Por último, las gráficas teóricas que mostraba la aplicación en el momento de la realización del experimento fueron las siguientes:



**Ilustración 18: Aceleraciones Lineales teóricas Phyphox**



**Ilustración 19 Aceleraciones con inclinación Teóricas Phyphox**

### ANÁLISIS DE RESULTADOS DE LA PRACTICA

En las gráficas obtenidas del primer experimento, en el cual el dispositivo móvil fue desplazado de forma lineal, se puede identificar claramente el momento inicial en el que el celular comenzó a registrar los datos. Al inicio del registro, se observa que el dispositivo se encontraba en reposo, tanto en los valores de aceleración lineal como en la aceleración con inclinación, lo cual valida la calibración inicial del sensor.

En el caso de la aceleración lineal, se evidencia que, a medida que se desplaza el dispositivo a lo largo de cada eje de forma independiente (X, Y o Z), el sensor registra dicho movimiento de manera adecuada, lo cual demuestra su correcto funcionamiento. Es importante destacar que, durante el desplazamiento en un eje específico, se observan pequeñas perturbaciones en los otros dos ejes. Por ejemplo, al mover únicamente en el eje X, se registran ligeras variaciones en los ejes Y y Z. Esto indica que, aunque se intente generar un movimiento puramente unidimensional, siempre existirán componentes mínimas en los otros ejes, probablemente causadas por vibraciones involuntarias o leves inclinaciones, lo que representa un margen de error de carácter experimental. Adicionalmente, en la gráfica de aceleración absoluta se puede observar la combinación vectorial de las tres aceleraciones



individuales (X, Y y Z), representando así una medida global del movimiento del dispositivo. Esta aceleración absoluta varía a lo largo del tiempo, reflejando los cambios de dirección y magnitud al cambiar de eje de desplazamiento.

En una segunda parte del experimento, una vez validado que el sensor funcionaba correctamente, se realizó un movimiento inclinado (no puramente lineal), es decir, con una trayectoria que implicaba simultáneamente más de un eje. En este caso, se observó cómo la aplicación es capaz de registrar la descomposición vectorial del movimiento a lo largo de los tres ejes. Aunque el ángulo de inclinación era desconocido, se pudo apreciar cómo, al salir del reposo, el dispositivo presentaba aceleraciones distintas en cada eje según la dirección del movimiento.

Tal como se muestra en la Ilustración 15, se identifica un valor significativo de aceleración en el eje Y, evidenciado por una mayor amplitud en los picos de la gráfica. Esto indica que durante ese intervalo, la aceleración fue más pronunciada en dicho eje en comparación con los demás. Por otro lado, la Ilustración 14 muestra que el eje X también presentó una aceleración mayor que el eje Z, lo cual se deduce al comparar la amplitud de sus respectivas gráficas.

Se logra concluir que los datos registrados por la aplicación muestran una clara correspondencia con el comportamiento real del movimiento aplicado al dispositivo, ya que se buscó realizar el desplazamiento de forma intencionada en la dirección de los ejes X e Y, más que en el eje Z. Esto confirma que el sensor realiza lecturas fiables y precisas. El análisis de las gráficas permite identificar la dirección dominante del movimiento en cada caso, validando tanto la funcionalidad del sensor como la utilidad de la aplicación como herramienta efectiva para el desarrollo de experimentos y el procesamiento de datos en contextos académicos y prácticos.

Adicionalmente, se demuestra que programas como MATLAB y Python son capaces de procesar adecuadamente archivos en formato .csv, generando representaciones gráficas que reflejan fielmente los datos capturados por la aplicación, tal como se evidenció en la comparación con los gráficos originales obtenidos durante el experimento.

### Segunda practica

En esta segunda práctica de laboratorio, se midió el nivel de intensidad de luz utilizando el sensor de luminosidad TCS3701 (AMS, Inc.) integrado en el dispositivo Samsung SM-A536E. El sensor opera con un rango de medición de hasta 60 000 lx y una resolución de 1 lx, garantizando una captura precisa de variaciones lumínicas. Para la adquisición y visualización de los datos se empleó la aplicación Phyphox, manteniendo la misma estructura de recolección empleada en la práctica anterior con el acelerómetro.

Se reutilizó íntegramente el código de procesamiento y representación gráfica desarrollado en Python y MATLAB; únicamente se actualizaron los títulos y etiquetas de los ejes para adaptarlos a las magnitudes de luz medidas. Con el fin de comprobar la correcta calibración y respuesta del sensor, se registraron dos condiciones de referencia: alta intensidad lumínica (luz ambiental directa) y baja intensidad lumínica (oscuridad controlada). A continuación, el dispositivo fue expuesto a variaciones graduales de iluminación mediante un objeto que proyectaba sombra progresiva sobre el sensor,

permitiendo observar de forma clara y continua cómo cambiaba la lectura en función de la luz incidente.

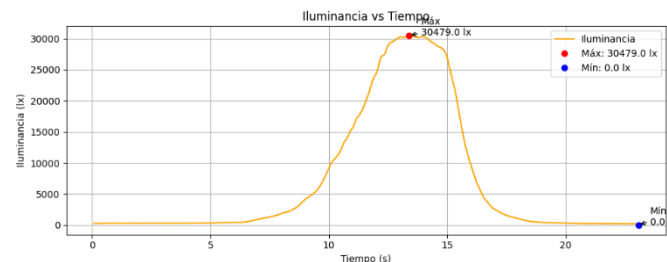


Ilustración 20 Nivel de luminosidad Python

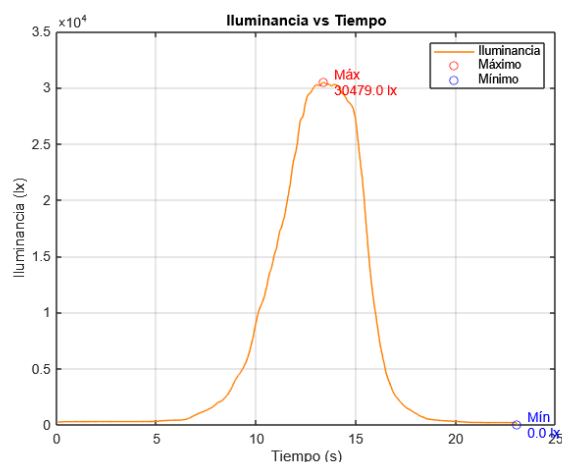


Ilustración 21 Nivel de luminosidad Matlab

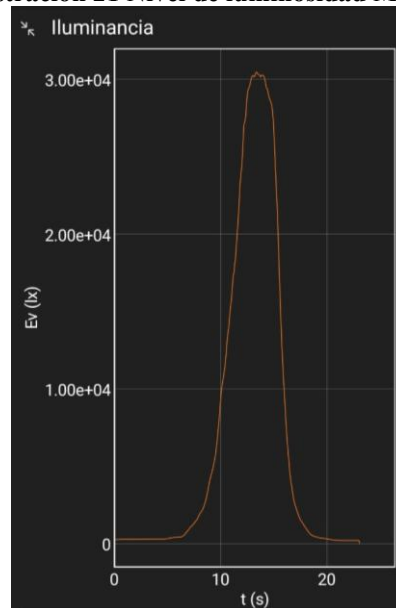


Ilustración 22 Intensidad luminica Teórica

En el código utilizado anteriormente se realizaron algunas modificaciones con el fin de obtener el valor máximo y mínimo de la intensidad de iluminación. Para ello, se emplearon los siguientes comandos específicos que permiten identificar tanto el valor como el instante de tiempo en el que estos ocurren.

#### En python:

- **lux.max()** El número más alto de la columna
- **lux.min()** El número más bajo de la columna

- **lux.idxmax()** devuelve el índice (posición en la tabla) donde se encuentra el valor máximo, que luego se usa para extraer el valor de tiempo correspondiente (**t\_max**).
- **lux.idxmin()** hace lo mismo pero con el mínimo.

#### En Matlab:

- La función **max(lux)** devuelve dos resultados: el valor máximo (**lux\_max**) y el índice donde ocurre (**idx\_max**).
- Igualmente, **min(lux)** devuelve el mínimo (**lux\_min**) y su índice (**idx\_min**).
- Luego, se usa el índice para obtener el valor de tiempo correspondiente con **t(idx\_max)** y **t(idx\_min)**.

#### ANÁLISIS DE RESULTADOS DE LA PRACTICA

En esta práctica se pudo evidenciar, a partir de los valores máximos obtenidos de los datos del archivo CSV, que al acercarse el celular a una fuente de alta intensidad de luz no se logró alcanzar ni superar el valor máximo de referencia de 60000 lx. El valor registrado experimentalmente fue de aproximadamente 30479 lx, lo cual concuerda con la intensidad luminosa comercial de lámparas similares, que suele estar en torno a los 30000 lx.

Esta ligera discrepancia entre el valor experimental y el valor comercial puede atribuirse a distintos factores, como la precisión del sensor del teléfono o las tolerancias de fabricación de la propia lámpara. Aun así, el resultado es coherente dentro del margen de error aceptable.

En las gráficas obtenidas se observa claramente cómo la intensidad luminosa aumenta de forma exponencial al acercarse el dispositivo a la fuente de luz, y decrece de manera similar al alejarlo. Este comportamiento refleja el funcionamiento esperado del sensor, el cual también es utilizado en los dispositivos móviles para ajustar automáticamente el brillo de la pantalla al pasar de interiores a exteriores o viceversa.

Además, este experimento permitió cuantificar de forma aproximada la intensidad luminosa de la lámpara utilizada, demostrando así la utilidad del sensor del celular como herramienta de medición en contextos experimentales.

### 3. CONCLUSIONES

Estas prácticas permitieron evidenciar la funcionalidad y versatilidad de la aplicación Phyphox para la adquisición de datos experimentales mediante los sensores integrados en dispositivos móviles. Así mismo, se demostró la capacidad de los lenguajes de programación Python y MATLAB para el procesamiento, análisis y visualización de los datos recolectados en formato .csv.

En relación con la práctica de aceleración lineal, se logró observar el comportamiento dinámico del dispositivo en movimiento a través de los ejes X, Y y Z, así como la magnitud absoluta de la aceleración. La interpretación gráfica permitió identificar picos máximos y mínimos durante la trayectoria, validando la precisión del acelerómetro. Tanto MATLAB como Python facilitaron la representación clara del fenómeno físico, incluyendo la detección automática de los valores extremos, lo cual respalda el rigor del análisis cuantitativo.

Respecto a la medición de intensidad luminosa, se evidenció que al acercarse el dispositivo móvil a una fuente de luz artificial,

los valores registrados no alcanzaron el máximo teórico de 60000 lx, pero sí se aproximaron al valor comercial típico de lámparas de 30000 lx. El comportamiento creciente y decreciente de la curva de iluminancia, registrado de forma exponencial durante el acercamiento y alejamiento del sensor a la fuente de luz, concuerda con el funcionamiento esperado de sensores ópticos en dispositivos móviles. La diferencia entre el valor experimental (30 479 lx) y el comercial podría atribuirse a tolerancias de fabricación del sensor o a condiciones particulares de la fuente de luz.

En general, se concluye que la integración de herramientas como Phyphox con entornos de programación proporciona una solución eficaz para desarrollar laboratorios portátiles y de bajo costo. La capacidad para capturar, exportar y analizar datos con precisión convierte esta metodología en una alternativa válida para complementar prácticas experimentales en áreas como física, ingeniería y análisis de señales.

#### REFERENCES

- [1] Phyphox, “phyphox Sensor Database”. Consultado: el 27 de julio de 2025. [En línea]. Disponible en: <https://phyphox.org/sensordb/>
- [2] “phyphox – Physical Phone Experiments”. Consultado: el 27 de julio de 2025. [En línea]. Disponible en: <https://phyphox.org/>
- [3] “¿Qué es Python? | Oracle América Latina”. Consultado: el 27 de julio de 2025. [En línea]. Disponible en: <https://www.oracle.com/latam/developer/what-is-python-for-developers/>
- [4] “MATLAB”. Consultado: el 27 de julio de 2025. [En línea]. Disponible en: <https://www.mathworks.com/products/matlab.html>