

UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y
SISTEMAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS
(Año del Bicentenario, de la consolidación de nuestra Independencia, y de la
conmemoración de la heroica batalla de Junín y Ayacucho)



TITULO: SIMULADOR PENDULO DE ONDAS

CURSO: FISICA I

DOCENTE: CARCAUSTO QUISPE CARLOS

NOMBRE: COAQUIRA IDME TAYLOR YAMPIER

SEMESTRE: II **GRUPO:** "A"

FECHA: 14/ 12/2024

PUNO – PERÚ

2024

PENDULO DE ONDAS

Codigo en Python colab :

```
import math
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable

def main():
    g = 9.76
    l = float(input("Ingrese la longitud del último péndulo en metros: "))
    t1 = float(input("Ingrese el periodo para el primer péndulo: "))
    p = int(input("Ingrese la cantidad de péndulos: "))
    t = 2 * math.pi * math.sqrt(l / g)
    print(f"El periodo del último péndulo es: {t:.2f} s")
    l1 = (t1 / (2 * math.pi)) ** 2 * g
    print(f"La longitud del primer péndulo es: {l1 * 100:.2f} cm")
    it = (t - t1) / (p - 1) if p > 1 else 0
    table_periodos = PrettyTable()
    table_periodos.field_names = ["Péndulo", "Periodo (s)", "Longitud (cm)"]
    table_frecuencias = PrettyTable()
    table_frecuencias.field_names = ["Péndulo", "Frecuencia (Hz)"]

    frecuencias = []
    # Mostrar datos de cada péndulo
    for i in range(p):
        longitud_cm = l1 * 100
        table_periodos.add_row([i + 1, f"{t1:.4f}", f"{longitud_cm:.4f}"])
        frecuencia = 1 / t1
        frecuencias.append(frecuencia)
        table_frecuencias.add_row([i + 1, f"{frecuencia:.6f}"])
        t1 += it
```

```

    l1 = (t1 / (2 * math.pi)) ** 2 * g
print()
print("Tabla de Periodos y Longitudes:")
print(table_periodos)
print()
print("Tabla de Frecuencias:")
print(table_frecuencias)
T = 0.01 # Paso de tiempo para la gráfica
tiempo = np.arange(0, 2, T) # Rango de tiempo
plt.figure(figsize=(10, 6))
for i, frecuencia in enumerate(frecuencias):
    # Señal para cada péndulo
    señal = np.sin(2 * np.pi * frecuencia * tiempo)
    plt.plot(tiempo, señal, label=f'Péndulo {i + 1} (f={frecuencia:.2f} Hz)')
plt.title('Señales de los Péndulos')
plt.xlabel('Tiempo (s)')
plt.ylabel('Amplitud')
plt.legend()
plt.grid()
plt.xlim(0, 2) # Ajustar el eje x
plt.ylim(-1.5, 1.5) # Ajustar el eje y
plt.show()

if __name__ == "__main__":
    main()

```