

## GUIA DE ENTREGA 3

### REGISTRO, GRAFICACIÓN Y CONTROL DE TOMA DE DATOS CON ARDUINO + PYTHON

#### Objetivo general

Integrar todos los conocimientos de lectura de sensores, comunicación serial y análisis de datos mediante la implementación de un sistema que:

1. Lea múltiples sensores conectados al Arduino.
2. Envíe los datos hacia Python para graficarlos y almacenarlos.
3. Permita controlar el tiempo o la detención de la toma de datos mediante un botón físico.

Cada estudiante o grupo deberá adaptar esta entrega a su proyecto particular (por ejemplo, monitoreo de invernadero, sistema de riego o control de cultivo).

1. Basado en tu conocimiento y la guía del taller 3 vas a intentar hacer una implementación donde se tomen datos por 1h continua, y luego responde:
  - ¿Cómo y donde podrías ajustar el tiempo total de medición en tu código de Python o Arduino si quisieras medir por 2, 3 o 24 horas?
  - ¿Qué parámetro o variable controlarías para definir ese tiempo?
  - ¿Qué pasa con la frecuencia de muestreo (cada cuántos segundos tomas datos)?  
¿Cómo influye eso en el tamaño del archivo y en la calidad de la gráfica?
2. Archivo .csv con el registro completo
  - Guarda todos los datos capturados durante la hora en un archivo .csv.
  - El archivo debe tener encabezados para cada sensor (por ejemplo: tiempo, temperatura, humedad, luz).
  - Al finalizar la toma de datos, el sistema debe guardar automáticamente el archivo.

Después de realizar la implementación responde:

- ¿Qué línea del código en Python se encarga de abrir y escribir el archivo .csv?
  - ¿Podrías hacer que el nombre del archivo se genere automáticamente usando la hora del sistema (datetime)?
  - ¿Cómo verificarías si los datos realmente se guardaron correctamente en el archivo?
3. Toma de datos hasta que se oprima un botón físico  
El sistema deberá permanecer tomando datos hasta que el usuario oprima un botón físico conectado al Arduino.  
Al presionar el botón:
    - Se detiene la toma de datos.
    - Se guarda automáticamente el archivo .csv con los datos recolectados.
    - Se muestra el gráfico final con las curvas de los sensores.
    - El sistema debe quedar listo para iniciar nuevamente desde el tiempo 0.

Luego de hacer la implementación responde:

- ¿Qué tipo de lectura debe tener el pin del botón en Arduino (digital o analógica)?
- ¿Cómo se podría usar una interrupción externa para detectar inmediatamente cuando el usuario oprime el botón?
- En Python, ¿qué condición deberías usar para romper el ciclo de toma de datos cuando llega la señal de parada?
- ¿Qué ventajas tiene usar un botón físico frente a detener el código manualmente desde el teclado?

### **Reto de decisión final**

Al terminar la actividad, cada grupo debe elegir uno de los dos caminos para su proyecto:

#### **1. Modo de tiempo definido por el usuario:**

- El programa pregunta al inicio:  
"¿Cuántas horas desea registrar datos?"
- El sistema toma datos durante ese tiempo y al finalizar guarda el archivo y muestra el gráfico.

#### **2. Modo infinito controlado por botón:**

- El sistema toma datos continuamente hasta que el usuario oprima el botón.
- Al presionarlo, se guarda el archivo .csv, se genera el gráfico, y el sistema se reinicia automáticamente.

***SOLO ES TOMAR SU DECISION AQUÍ, NADA MAS***

### **Recomendaciones técnicas**

- Puedes usar Tinkercad si no tienes el hardware físico.
- Apóyate en las notas de laboratorio anteriores y en los códigos de referencia del repositorio oficial del curso.
- Puedes consultar y apoyarte en IA (como ChatGPT, Gemini o Claude) para interpretar errores o adaptar funciones, pero el código final debe reflejar tu comprensión del proceso.
- Recuerda cerrar el Monitor Serial en Arduino IDE antes de correr el script en Python.
- Verifica siempre que el SERIAL\_PORT en Python corresponda al que estás usando en Arduino.

## **Entrega final**

### **Tu entrega debe incluir:**

1. Informe corto (1 a 2 páginas) describiendo:
  - Proyecto (qué mide y por qué es importante).
  - Sensores utilizados y conexiones.
  - Explicación de cómo ajustaste tiempo, muestreo y parada.
  - Obtener una gráfica continua que muestre el comportamiento de todos los sensores del sistema durante un periodo de 1 hora.
2. Código Arduino y Python, correctamente comentado.
3. Archivo .csv generado automáticamente.
4. Un video corto mostrando la toma de datos hasta que se hunda el botón (mostrar también la gráfica y el archivo .csv generado)