**🌱 Introducción general**

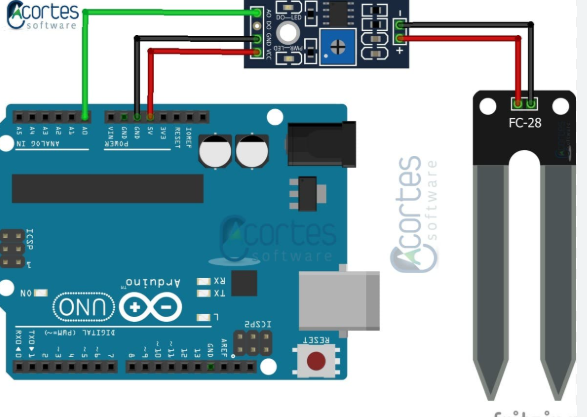
La ingeniería agronómica moderna depende cada vez más del uso de sensores y tecnologías de automatización para garantizar cultivos sostenibles, trazables y altamente productivos. Desde el control de temperatura en invernaderos hasta el monitoreo de parámetros en sistemas de acuaponía, los ingenieros deben ser capaces de capturar, procesar y analizar datos en tiempo real.

En esta guía básica aprenderemos a trabajar con Arduino y a integrar sus lecturas con Python, desarrollando un flujo completo: desde la obtención de datos de sensores, su trazabilidad con un reloj externo, hasta el almacenamiento seguro en una tarjeta MicroSD. Estos ejercicios no solo fortalecen la lógica de programación y el pensamiento computacional, sino que también representan el primer paso para crear sistemas de monitoreo confiables en entornos agrícolas.

Para acompañar el proceso, los estudiantes contarán con el repositorio oficial del curso:

<https://github.com/yejaramilm/PensamientoComputacional2/tree/main/Clases>

1. **recopilación de Datos y graficación**
2. Tome su sensor de temperatura y haga la conexión al Arduino usando el modulo adicional de conexión. Conecte la salida a la entrada analógica 0



1. Descargue los códigos de implementacion de:  
   <https://github.com/yejaramilm/PensamientoComputacional2/tree/main/Clases/Semana-11/02_AdquirirDatosyGuardar>
2. Para este ejercio puede descargar todos los archivos contenidos en la carpeta, para descargar archivos solo de la ubicación de una carpeta use:  
   <https://download-directory.github.io/>
3. Ubique todos los archivos en usa misma carpeta de forma tal que le sea cómodo trabajar solo en esa carpeta

A screenshot of a computer

AI-generated content may be incorrect.

1. Recuerde que usted trabaja en 2 entornos, uno es visualStudio Code, y el otro es Arduino IDE. En el Arduino IDE abra el archivo 01\_AdquirirDatosBasico.ino  
   A screenshot of a computer

   AI-generated content may be incorrect.
2. Analice el código detalladamente, apoyese en la IA y en los comentarios que le dejo el profesor para entender que esta pasando en cada línea.
3. Corra el algoritmo y verifique la lectura en el monitor serial de Arduino, puede sumergir el sensor en algún liquido de ser necesarion

A screenshot of a computer

AI-generated content may be incorrect.

1. Una vez verificada la lectura, **cierre el serial monitor de Arduino**, ya que no lo puede tener abierto para enviar datos a Python.

*Vamos ahora a Python*

1. Vaya a visualStudio code y abra el archivo **sensor\_lecture\_and\_plot.py**  
   A screenshot of a computer

   AI-generated content may be incorrect.
2. De nuevo, apoyese en la IA y en los comentarios que le dejo el profesor para intentar entender cada línea de código y lo que hacen en el sistema
3. Verifique que el SERIAL\_PORT sea el mismo que el que usted tiene definido en Arduino IDE, si no es así, cámbielo y ponga el suyo.

A screenshot of a computer

AI-generated content may be incorrect.

1. Despues de verificar y cambiar esto (en caso de que haya sido necesario), puede ejecutar el código de Python en VisualStudio Code.  
   Deberia empezar a ver las mediciones y finalmente el grafico

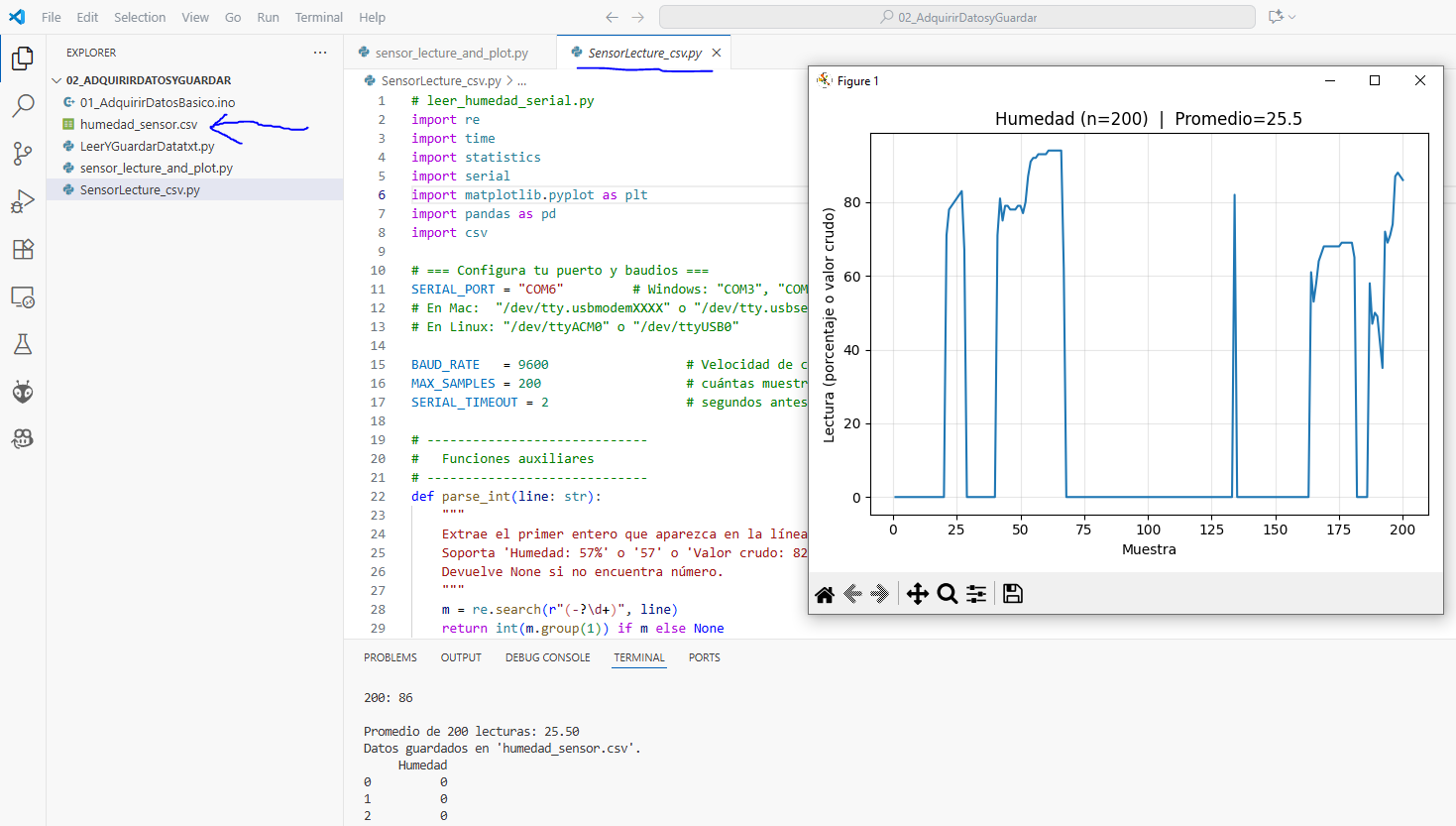
A screenshot of a computer

AI-generated content may be incorrect.

A graph with numbers and lines

AI-generated content may be incorrect.

1. Ahora intente hacer cambios que mejoren el código. Piense esto como un punto de partida, pero realmente hay mucho que puedes hacer, por ejemplo, tomar mas o menos datos según tu necesidad, mejorar la gráfica, o incluso guardar los datos en el computador para verlos después. Empieza por algo simple, **intenta tomar mas datos, y que sea con mayor frecuencia.**
2. Si quieres, puedes guardar los datos en un archivo .txt o .csv, cada uno de los otros script the Python que ves en VisualStudio es justamente para esos casos. Puedes correr SensorLecture\_csv.py para correr el mismo sistema, pero ahora guardando los datos en un csv



Nota que el cambio se da solo agregando como función la siguiente composición:

   with open(nombre\_archivo, 'w', newline='', encoding='utf-8') as f:

        writer = csv.writer(f)

        writer.writerow(["Humedad"])   # encabezado

        for d in datos:

            writer.writerow([d])

He invocandola en el código principal.

**Conclusiones**:

*Despues de realizar este proceso de forma juiciosa, es muy importante que consolides tus conclusiones o notas importantes aquí, con el fin de que sea algo que puedas recordad fácilmente, y te ayude en la implementación de tu final.*Conclusion 1:

Conclusion 2:

Conclusion 3:

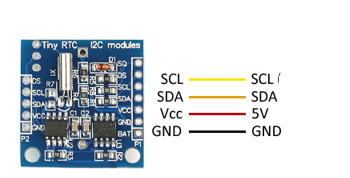
*Lineas de código que vale la pena resaltar por su importancia:*

|  |  |  |  |
| --- | --- | --- | --- |
| *Arduino IDE* | *Porque es importante?* | *Python* | *Porque es importante?* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Lectura de datos con sensores y uso de un clock externo para asegurar la trazabilidad**

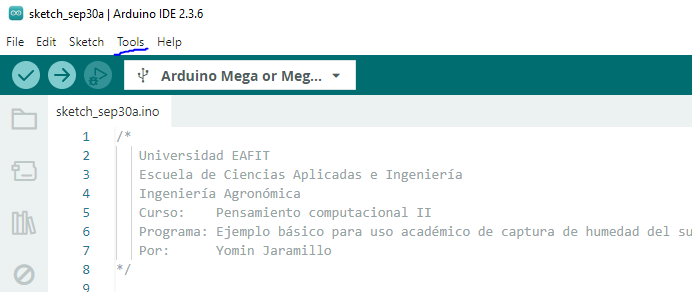
La segunda etapa se centra en incorporar un reloj externo (RTC – Real Time Clock), lo que nos permitirá asociar cada dato capturado a una fecha y hora exacta. Esto es clave en proyectos agrícolas, donde la trazabilidad garantiza confianza: por ejemplo, saber qué día y a qué hora un cultivo alcanzó determinada humedad o temperatura. Sin esta referencia temporal, los datos pierden valor para el análisis histórico y la toma de decisiones.

Conecta el RTC basándote en las siguientes referencias:

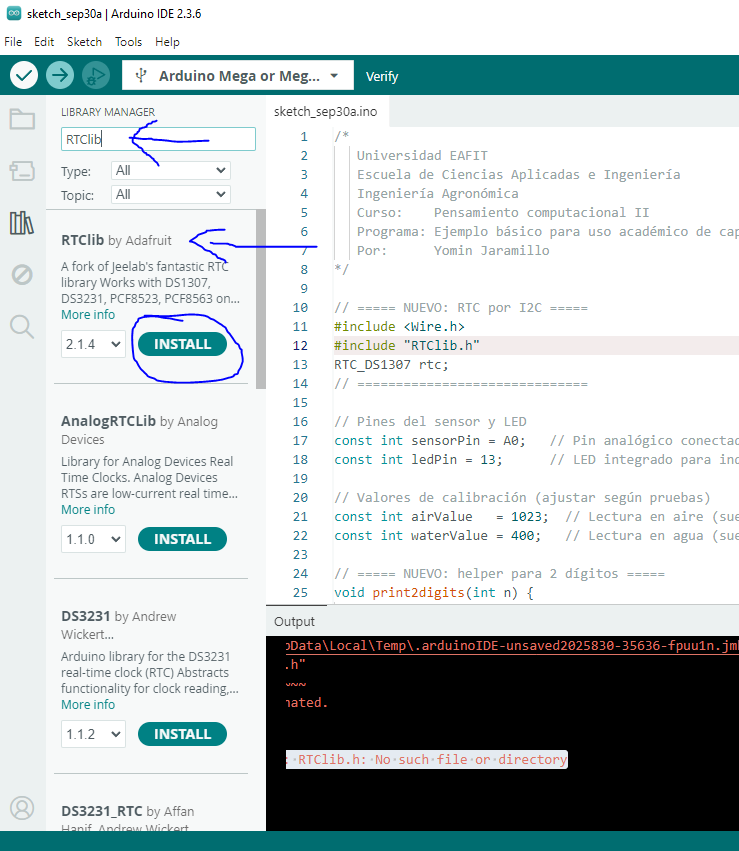


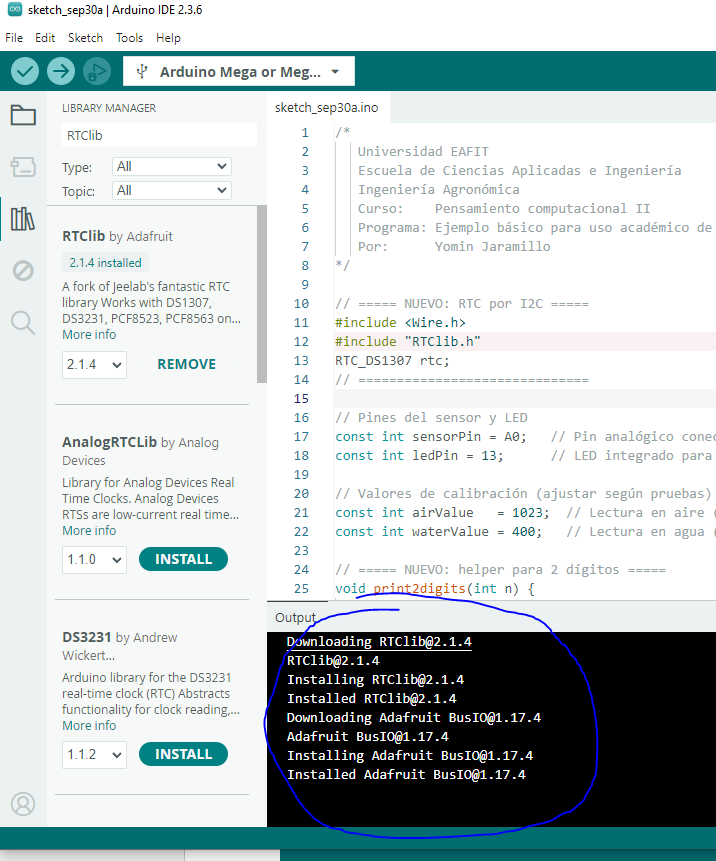


1. Primero vamos a tener que instalar la librería para poder usar el RTC, ve a tu Arduino IDE, ve al menú🡪herramientas🡪Administrar librerías

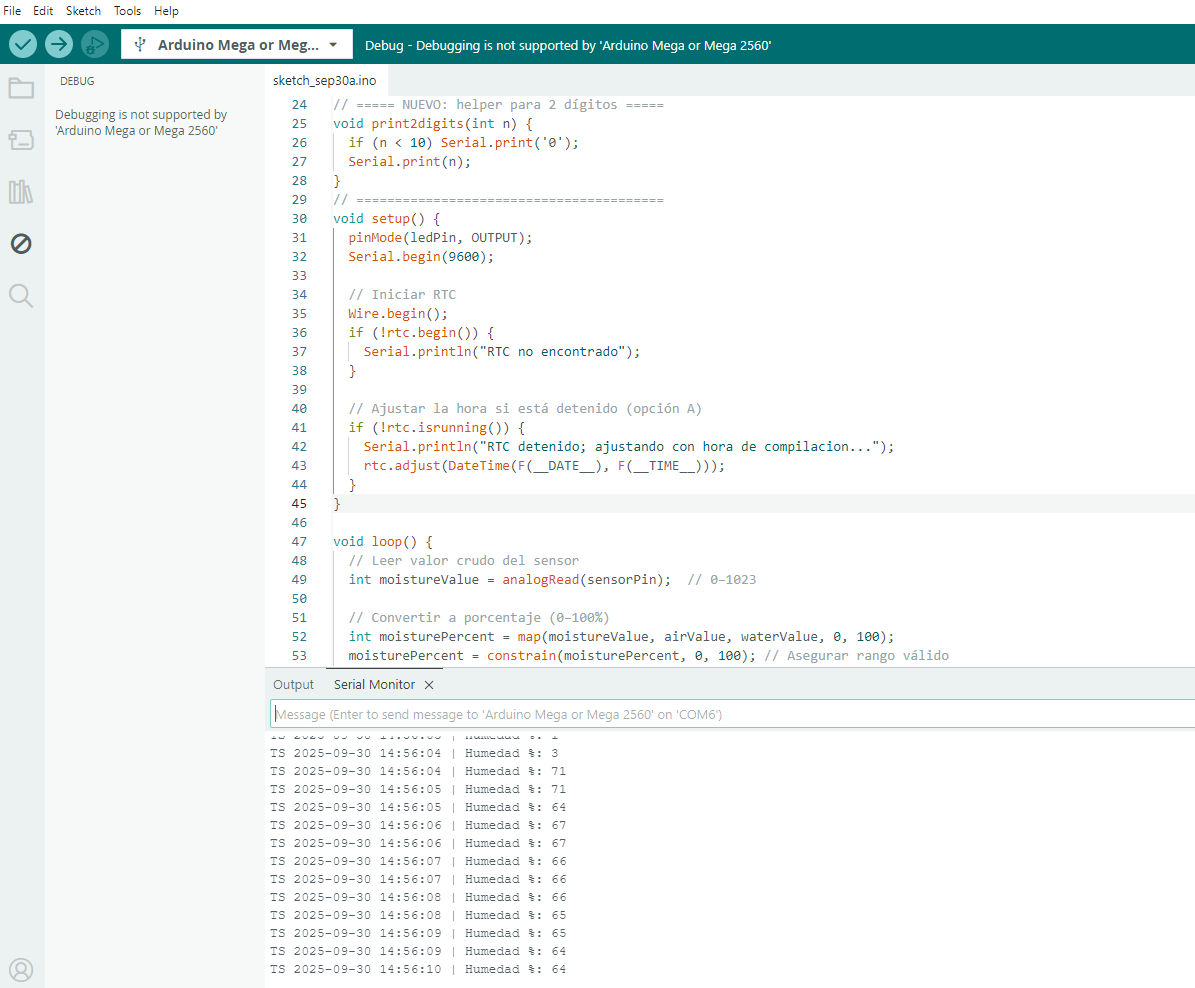


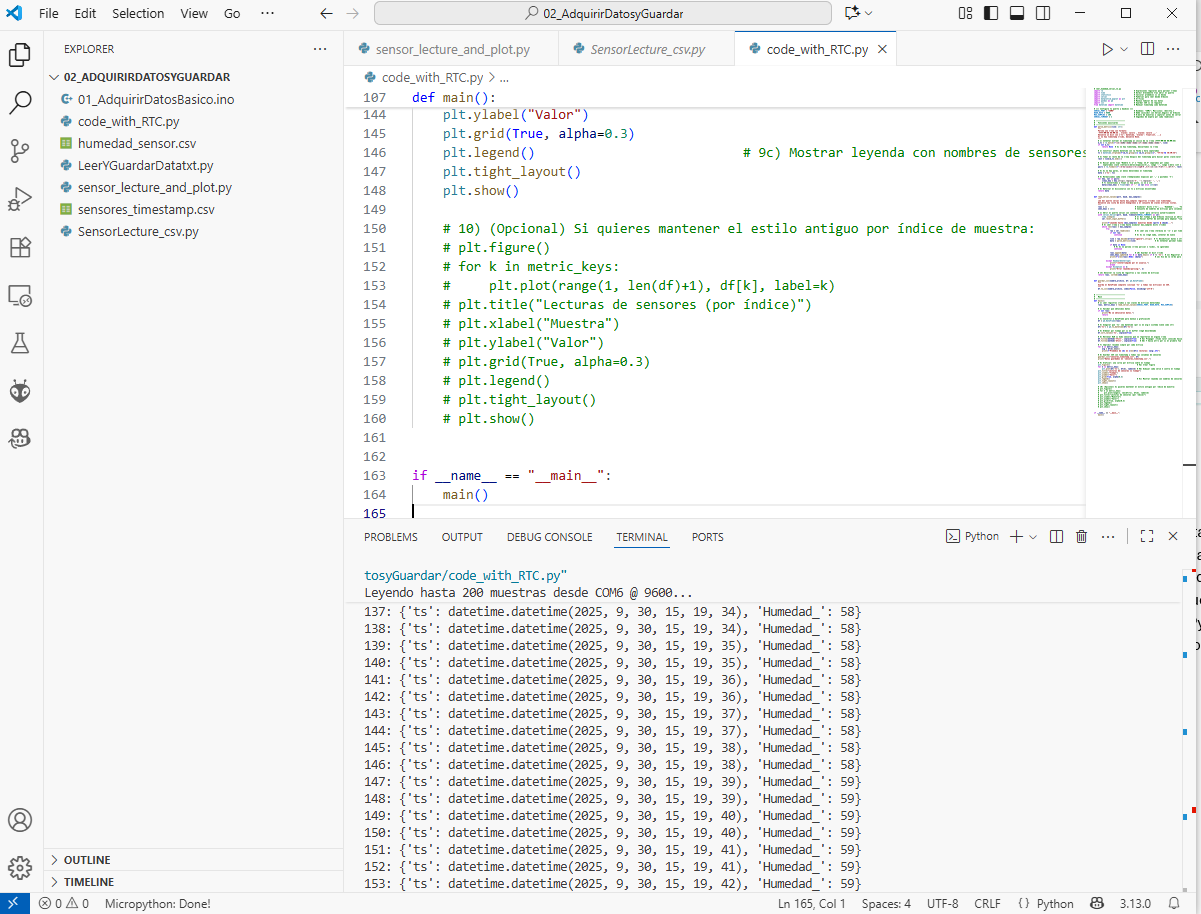
1. En la barra de búsqueda escribe RTClib 🡪 Selecciona RTClib by Adafruit (es la oficial y más usada) 🡪 Dale click en instalar 🡪 Instalar todo





1. Despues de esto, puedes correr el archivo sensor\_with\_RTC.ino, y veras que la medición del sensor esta acompañada de la hora exacta en que dicha medición fue tomada.



1. De nuevo, tomate el tiempo, con ayuda de la IA y de los comentarios del profesor de entender cada una de las líneas y cual es su importancia en el código, para que más adelante puedas adaptarlo a tu necesidad.
2. Si consideras que lo haz entendido bien, prosigue. Recuerda que debes cerrar el Serial Monitor en Arduino IDE antes de ir a Python. Una vez en Python puedes correr el archivo code\_with\_RTC.py, y vera como Python toma los datos desde Arduino y le da formato en forma de diccionario. Este formato hacer generalizable el código, es decir, le permite escalarlo a mas sensores cuando este midiendo muchas variables.  
     
   

Se tomarán 200 datos y nuevamente se graficará. Adicionalmente, el código guarda el registro del tiempo y la medición en un archivo sensores\_timestamp.csv

1. De nuevo, tomate el tiempo, con ayuda de la IA y de los comentarios del profesor de entender cada una de las líneas y cuál es su importancia en el código, para que más adelante puedas adaptarlo a tu necesidad.
2. Verifique el contenido del archivo .csv

**Conclusiones**:

*Despues de realizar este proceso de forma juiciosa, es muy importante que consolides tus conclusiones o notas importantes aquí, con el fin de que sea algo que puedas recordad fácilmente, y te ayude en la implementación de tu final.*Conclusion 1:

Conclusion 2:

Conclusion 3:

*Lineas de código que vale la pena resaltar por su importancia:*

|  |  |  |  |
| --- | --- | --- | --- |
| *Arduino IDE* | *Porque es importante?* | *Python* | *Porque es importante?* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Hacer medición de más de un sensor para el registro de datos**

Como dijimos anteriormente, ahora tenemos un código generalizable, por lo que vamos a trabajar con mas sensores, la idea es que hagamos este mismo proceso pero con todos los sensores que tenga nuestro sistema.

En este caso yo lo hare con el sensor MQ135 pero tu lo puedes hacer con los que quieras, apoyate en la IA.  
a. Conectar el sensor MQ135, en particular yo lo conecte en A1, por lo que si lo conectas en algún punto diferente necesitaras cambiar el código indicando el pin.  
A diagram of a circuit board

AI-generated content may be incorrect.

b. Para este punto debería tener conectados: el sensor de humedad a A0, el RTC y el sensor de gases MQ135 en A1. Descargue los códigos del repositorio publico:

c. En este caso el sensor de gases es un sensor analogico por lo que no es necesario instalar ninguna librería, solo debemos correr multisensor\_RTC.py para observar su funcionamiento.

A screenshot of a computer

AI-generated content may be incorrect.

d. Revisa donde y como se incorporo el nuevo sensor. De igual forma repasa el código para mejorar tu entendimiento del mismo. Recuerda apoyarte de la IA y de los comentarios que te he dejado. **LO MAS IMPORTANTE ES QUE ENTIENDAS COMO SE HACE LA LECTURA Y SE MANDAN POR EL PUERTO SERIAL**  
e. Después de que ya haz comprendido esto, recuerda cerrar el serial Monitor antes de ir a Python

f. En este caso, el mismo código que utilizaste en el punto pasado te puede servir “code\_with\_RTC.py”, ya que recuerda que lo hicimos generalizable a la cantidad de datos que requieras leer, por lo tanto solo debes correrlo.  
A screenshot of a computer

AI-generated content may be incorrect.

**Conclusiones**:

*Despues de realizar este proceso de forma juiciosa, es muy importante que consolides tus conclusiones o notas importantes aquí, con el fin de que sea algo que puedas recordad fácilmente, y te ayude en la implementación de tu final.*Conclusion 1:

Conclusion 2:

Conclusion 3:

*Lineas de código que vale la pena resaltar por su importancia:*

|  |  |  |  |
| --- | --- | --- | --- |
| *Arduino IDE* | *Porque es importante?* | *Python* | *Porque es importante?* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TAREAS PARA VACACIONES**

Con la información que tienens ya puedes montar la lectura de todos los sensores de tu sistema y hacerles seguimiento en el tiempo, la idea es que realices esta actividad con el fin de avanzar en tu entrega final.  
Tu resultado debe ser:

* Un gráfico de todos tus sensores en el tiempo, toma 1 h de datos para la gráfica.
* Un archivo .csv que guarde el registro de los datos por dicha hora
* Puedes hacer que tu sistema tome datos hasta que tu lo interrumpas hundiendo un botón?