Solución IoT basada en HTTP

Introducción

Objetivos y competencias a reforzar

- El estudiante conocerá cómo es un web service en la práctica real, el cual no se trata solo de código sino de recursos y estrategias serias para lograr que las personas que lo usen no resulten frustradas con los servicios que ofrece.
- El estudiantes aprenderá que el éxito de una solución depende de poder plantear un problema mayor en forma de sprints. También que en cada sprint es necesario alcanzar el dominio necesario de las herramientas usadas

Prerrequisitos

El estudiante debe:

- Ser hábil en el uso de la IDE de Google Apps Script,
- Ser hábil en el uso de Google Colab o cualquier otro IDE de programación en Python
- Conocer lo que significa un webservice, incluyendo aspectos como:
    - el uso del método GET
    - el uso de método POST
    - el uso de una web app usando la URL o endpoint
- Tener habilidades de comunicación efectiva con GPT para obtener código.
- Tener facilidades para aprender por sus medios sobre el uso de una ESP

Conocimientos previos

Nuevos términos

- Registro o record: Cuando se habla de bases de datos, un registro (record) se refiere generalmente a los datos que aparecen en una fila de datos.
- CRUD: Significa Create, Read, Update, Delete. Es una sigla muy usada en bases de datos para referirse a las facilidades que se tienen para crear nuevos registros de datos, leerlos, actualizarlos o borrarlos.
- Desarrollador: se refiere al programador o a las herramientas que el programador usa para disfrutar de este CRUD.

- Clase: Es el código que se escribe para modelar una realidad. Podemos compararlo con los planos de una casa. A partir de esos planos se pueden crear muchas casas, las cuales se pueden diferenciar si se varían parámetros.
- Método: es el término usado para referirse a las funciones cuando estas son parte de un clase.
- Instanciar: es el proceso de crear un objeto a partir de una clase
- Constructor: es el primer método, que corre por defecto cuando se instancia una clase

Recursos a usar

- Web Service para soportar CRUD sobre Google Sheet: Documentación del Web Service

El reto general

Implementar un sistema IoT básico que consiste en dispositivos remotos censando información y enviándola a una base de datos en la nube.
Se cuenta con un Web Service propio (hecho por el semillero IoT de la UIS) que permite realizar registros de información en una base de datos basada en Google Sheet, desde cualquier lugar remoto.

Vamos a dividir el reto en sprints que sean fáciles de alcanzar pero que nos conduzca claramente a la solución final:

Sprint 1. Solución completa pero simulando el sensado y usando Python en el llamado remoto

Paso 1. Reconocimiento, a partir de la documentación, del web service creado para operaciones CRUD

Enlace a la documentación del web service CRUD

Tabla para entregar evidencias del paso 2
Siguiendo las instrucciones del capítulo “Información General”, escriba aquí, un ejemplo para un objeto de datos que aplique para realizar un registro de datos. Puede tomar el ejemplo 1 y adecuarlo a un caso propio, con datos propios, diferentes a los del ejemplo y a los que el profesor o amigos han dado. Si no conoce algunos parámetros puede usar valores hipotéticos.

```
datos = {
            "ordentipo": "tipo_de_orden",
            "url": "url_de_la_google_sheet",
            "numeroHoja": número _de_la_hoja,
            "filaencabezados": numero_de_la_fila encabezados,
            "columnaId": numero_de_columna_para id,
            "id": "valor_unico_para_el_registro", // Solo necesario para leer, actualizar y borrar
            "datos": "datos_a_registrar", // Sólo necesario para crear o actualizar
            "encabezado": "encabezado_para_busqueda", // Solo necesario para buscar en columna
            "valor": "valor_para_busqueda", // Solo necesario para buscar en columna
            "nombreHoja": "nombre_de_la_nueva_hoja" // Solo necesario para agregar una hoja
        }
```

Revisado todo el material correspondiente al webservice que ofrece el profesor para este taller, qué elementos cree usted que debe tener todo webservice hecho profesionalmente, para ser puesto al servicio de un amplio público.

1. Documentación Detallada

- **Descripción del Servicio**: Explicar de manera clara qué hace el web service, incluyendo ejemplos prácticos. Tu archivo PDF contiene ejemplos que cumplen con este criterio(Documentación.CRUD como…).
- **Definición de Endpoints y Métodos HTTP**: Documentar claramente todos los endpoints, los métodos HTTP soportados (GET, POST, PUT, DELETE), y cuándo se deben utilizar.
- **Parámetros de Solicitud y Respuesta**: Explicar todos los parámetros necesarios en las solicitudes y los valores esperados en las respuestas, incluyendo el manejo de errores.

2. Seguridad

- **Autenticación y Autorización**: Implementar mecanismos de autenticación, como OAuth, API keys o JWT, y definir claramente los permisos.
- **Cifrado**: Asegurar que todas las comunicaciones se realicen a través de HTTPS para proteger los datos en tránsito.
- **Validación de Datos**: Validar tanto los datos de entrada como los de salida para evitar ataques como la inyección de código.

3. Escalabilidad y Rendimiento

- **Optimización del Rendimiento**: Minimizar el tiempo de respuesta optimizando la base de datos y el código. Implementar caché cuando sea necesario.
- **Manejo de Cargas Elevadas**: Preparar el web service para manejar múltiples solicitudes concurrentes y grandes volúmenes de datos sin degradar el rendimiento.
- **Control de versiones**: Soporta múltiples versiones del API para permitir actualizaciones sin interrumpir el servicio a los usuarios existentes.

4. Manejo de Errores

- **Mensajes de Error Claros**: Proporcionar mensajes de error detallados y claros que indiquen qué salió mal y cómo corregirlo.
- **Códigos de Estado HTTP**: Utilizar correctamente los códigos de estado HTTP (e.g., 200 OK, 400 Bad Request, 500 Internal Server Error).

5. Mantenimiento y Monitoreo

- **Monitoreo**: Implementar soluciones para monitorear el estado del servicio en tiempo real y recibir alertas ante fallas o degradaciones.
- **Registro de Logs**: Mantener registros detallados de todas las solicitudes y errores para facilitar la resolución de problemas.
- **Soporte y Actualizaciones**: Proveer soporte técnico y un ciclo de actualizaciones regulares para corregir errores y mejorar el servicio.

6. Usabilidad

- **Facilidad de Integración**: Hacer que la integración con el servicio sea sencilla, proporcionando SDKs o bibliotecas en diferentes lenguajes de programación.
- **Pruebas Automatizadas**: Asegurarse de que el servicio esté completamente cubierto por pruebas automatizadas que verifiquen la funcionalidad y la estabilidad.

7. Compatibilidad y Estándares

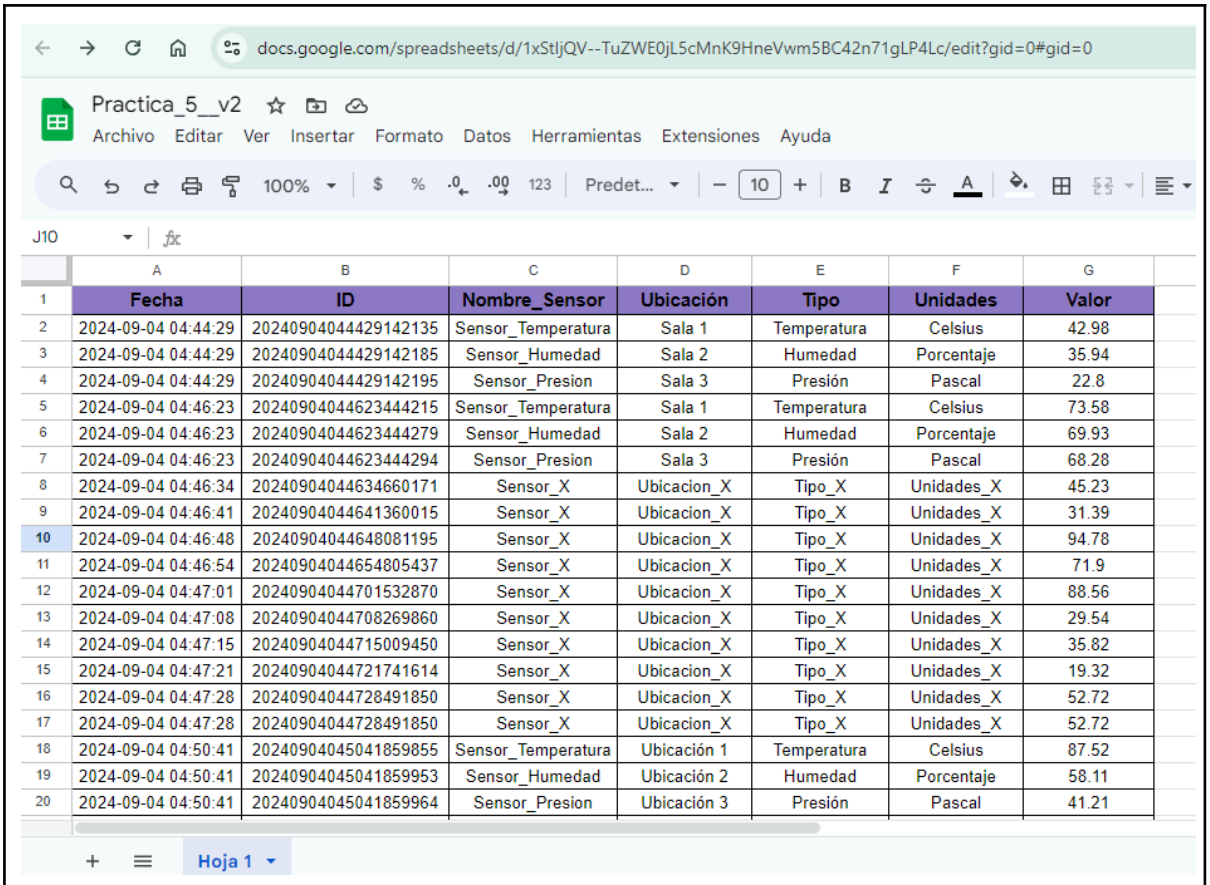
- **Adherencia a Estándares**: Cumplir con los estándares web y de API como RESTful, SOAP, o GraphQL, dependiendo del caso de uso.
- **Compatibilidad con Diferentes Clientes**: Asegurarse de que el servicio sea compatible con diferentes tipos de clientes (e.g., navegadores, aplicaciones móviles, otros servicios web).

Paso 2. Creación de una base de datos basada en Google Sheet
No es necesario comprender en detalle todo el código del webservice. Imagina que tu quieres usar la API de GPT. Para ello no necesitas entrar a mirar el código detrás de esa solución, solo necesitas saber como usarla. Pero por razones pedagógicas, en este paso se busca es que el estudiante conozca el código para que constate que es un desarrollo sencillo, pero sobre todo bien documentado, para brindarle seriedad a los usuarios que quieran usar este web service, para consumirlo para crear algo nuevo a partir del mismo. Si de pronto hay algo que te inquieta del código, puedes pasarselo a Chat GPT y preguntarle más detalles.

La tarea a realizar consiste en usar la documentación para:

- Conocer los servicios que brinda y complementar revisando los ejemplos
- Abrir el código para conocerlo

Tabla para entregar evidencias del paso 3
Escriba aquí la URL de la Google Sheet creada para la base de datos
https://docs.google.com/spreadsheets/d/1xStIjQV--TuZWE0jL5cMnK9HneVwm5BC42n71gLP4Lc/edit?usp=sharing
Agrega aquí una captura de pantalla que muestre los encabezados para tu base de datos (Nota: debes ser original, no usar los mismos del profesor u otros compañeros)



docs.google.com/spreadsheets/d/1xStljQV--TuZWE0jL5cMnK9HneVwm5BC42n71gLP4Lc/edit?gid=0#gid=0

Practica_5__v2

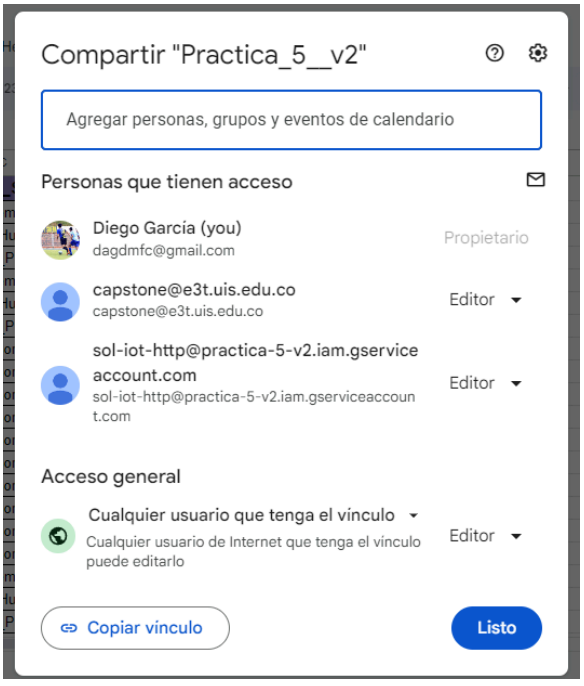
Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda

J10

	A	B	C	D	E	F	G
1	Fecha	ID	Nombre_Sensor	Ubicación	Tipo	Unidades	Valor
2	2024-09-04 04:44:29	20240904044429142135	Sensor_Temperatura	Sala 1	Temperatura	Celsius	42.98
3	2024-09-04 04:44:29	20240904044429142185	Sensor_Humedad	Sala 2	Humedad	Porcentaje	35.94
4	2024-09-04 04:44:29	20240904044429142195	Sensor_Presion	Sala 3	Presión	Pascal	22.8
5	2024-09-04 04:46:23	20240904044623444215	Sensor_Temperatura	Sala 1	Temperatura	Celsius	73.58
6	2024-09-04 04:46:23	20240904044623444279	Sensor_Humedad	Sala 2	Humedad	Porcentaje	69.93
7	2024-09-04 04:46:23	20240904044623444294	Sensor_Presion	Sala 3	Presión	Pascal	68.28
8	2024-09-04 04:46:34	20240904044634660171	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	45.23
9	2024-09-04 04:46:41	20240904044641360015	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	31.39
10	2024-09-04 04:46:48	20240904044648081195	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	94.78
11	2024-09-04 04:46:54	20240904044654805437	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	71.9
12	2024-09-04 04:47:01	20240904044701532870	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	88.56
13	2024-09-04 04:47:08	20240904044708269860	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	29.54
14	2024-09-04 04:47:15	20240904044715009450	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	35.82
15	2024-09-04 04:47:21	20240904044721741614	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	19.32
16	2024-09-04 04:47:28	20240904044728491850	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	52.72
17	2024-09-04 04:47:28	20240904044728491850	Sensor_X	Ubicacion_X	Tipo_X	Unidades_X	52.72
18	2024-09-04 04:50:41	20240904045041859855	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	87.52
19	2024-09-04 04:50:41	20240904045041859953	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	58.11
20	2024-09-04 04:50:41	20240904045041859964	Sensor_Presion	Ubicación 3	Presión	Pascal	41.21

Hoja 1

Agrega una captura de pantalla donde muestra que le has otorgado permisos de edición a capstone@e3t.uis.edu.co que es el propietario del web service.



Paso 3. Implementación de la solución en Python.

Tabla para entregar evidencias del paso 4
Escriba aquí la información de entrenamiento preparada para GPT
Tengo un webservice que podriamos llamar "webcrud". Esta es la documentación de ese web service. [Aquí adjunté el documento CRUD suministrado por el profesor] memorizar: gracias a la documentación, tu ya conoces: 1) la URL del web service 2) sabes que sirve para hacer registros en una base de datos. 3) que para consumir el webservice, los datos deben ser previamente organizados en un objeto de datos (en un diccionario, en términos de python) 4) el siguiente es un ejemplo sobre la manera en que se organizan los datos: { "ordentipo": "crear", "url": "https://docs.google.com/spreadsheets/d/1aZ02LgYHxfcpkfMDu17bzfcCieAcYIJqJRguP8pZGM", "numeroHoja": 0, "filaencabezados": 1, "columnaId": 1, "datos": "[\"2024-08-07T12:00:00Z\",\"Laboratorio de Física\",\"Datos aleatorios de prueba\",16,\"0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2,1.3,1.4,1.5,1.6\"]" } **Memorizar:** Los datos de mi caso particular son: 1) Mi base de datos es una google sheet que tiene la siguiente URL: " https://docs.google.com/spreadsheets/d/1rwOnspT85k0knE56ZCb4ul8XgJonbEd5XQg0THVIiMc/edit?usp=sharing". 2) Usaré la hoja cero. 3) Los encabezados que tengo son estos: Date; ID; SensorName; Location; Unidades; MeditionValue. 4) Uso la primera fila para los encabezados y la segunda columna para los ID Lo que quiero es un código en python, para Google Colab que cumpa lo siguiente: 1) Tendremos una función llamada "simuladorSensor" que genera datos de un sensado hipotético, por ejemplo de temperatura. 2) El código principal consumirá esa función las función anterior cada 5 segundos para leer la temperatura. 3) Organizará el objeto de datos (diccionario) para consumir el web service para "crear" un nuevo registro en la base de datos en cada caso de medición
Escriba aquí la información de entrenamiento para explicarle a GPT cual es la información particular para tu caso
Memorizar: Los datos de mi caso particular son: 1) Mi base de datos es una google sheet que tiene la siguiente URL: " https://docs.google.com/spreadsheets/d/1xStIjQV--TuZWE0jL5cMnK9HneVwm5BC42n71gLP4Lc/edit?usp=sharing". 2) Usaré la hoja 1. 3) Los encabezados que tengo son estos: Fecha, ID, Nombre_Sensor, Ubicación, Tipo, Unidades y Valor. 4) Uso la primera fila para los encabezados y la segunda columna para los ID Lo que quiero es un código en python, para Google Colab que cumpa lo siguiente: 1) Tendremos una función llamada "simular_datos_sensor" que genera datos de un sensado hipotético, por ejemplo de temperatura, humedad y presión. 2) El código principal consumirá esa función las función anterior cada 5 segundos para leer la temperatura. 3) Organizará el objeto de datos (diccionario) para consumir el web service para "crear" un nuevo registro en la base de datos en cada caso de medición
Escriba aquí el prompt usado para solicitar a GPT que genere el código para la solución
Quisiera que me generaras un código en Python para simular la recepción de diferentes

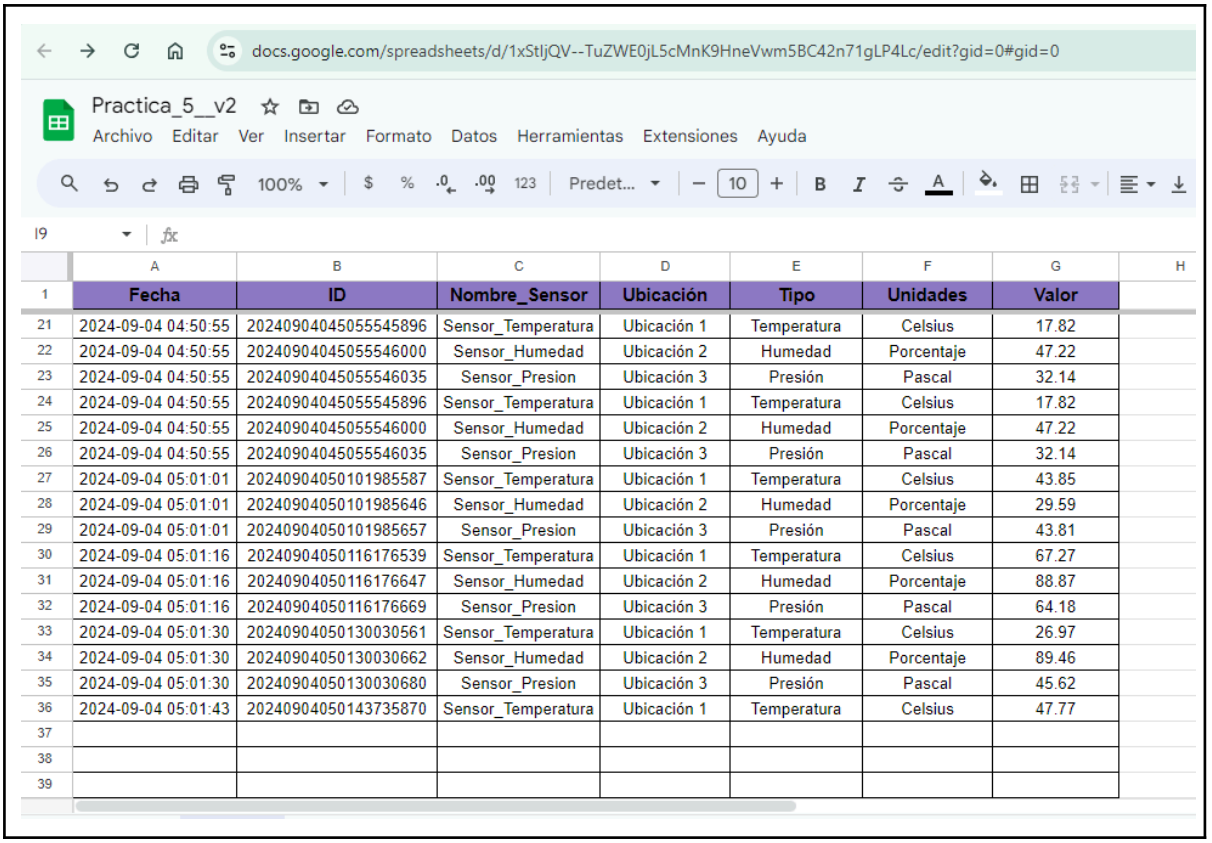
datos de sensores que se reciben a traves del código en Python y posteriormente se visualizan en un archivo de Google Sheets, ya teniendo un archivo de Google Sheets cuya URL es la siguiente: https://docs.google.com/spreadsheets/d/1xStIjQV--TuZWE0jL5cMnK9HneVwm5BC42n71gLP4Lc/edit?usp=sharing. Quiero que en ese Google Sheets, en una tabla que tiene los siguientes encabezados: Fecha, ID, Nombre_Sensor, Ubicación, Tipo, Unidades_Medida y Valor, contenida en la Hoja1 del archivo, se almanecen esos datos simulados de sensores de Temperatura, Humedad y Presión desde la fila 2 en adelante ya que la fila 1 contiene los encabezados de la tabla. Finalmente al ejecutar el código, por ejemplo en Google Colab, se muestra el resultado de la tabla con los valores simulados ya agregados y también se observan esos datos agregados en el Google Sheet.
1) Tendremos una función llamada "simular_datos_sensor" que genera datos de un sensado hipotético de Temperatura, Humedad y Presión.
2) El código principal consumirá esa función las función anterior cada 3 segundos para leer los sensores.
3) Organizará el objeto de datos (diccionario) para consumir el web service para "crear" un nuevo registro en la base de datos en cada caso de medición

Código obtenido

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials
import random
from datetime import datetime
from google.colab import drive
import time  # Para retrasos en la simulación

# Montar Google Drive
drive.mount('/content/drive')

# Ruta al archivo JSON en Google Drive
json_keyfile_name = '/practica-5-v2-d2d3049d3f0a.json'

# Configuración de la autenticación para acceder a Google Sheets
scope = ["https://spreadsheets.google.com/feeds",
"https://www.googleapis.com/auth/drive"]
creds =
ServiceAccountCredentials.from_json_keyfile_name(json_keyfile_name,
scope)
client = gspread.authorize(creds)

# Accede a la hoja de cálculo
spreadsheet_url =
"https://docs.google.com/spreadsheets/d/1xStIjQV--TuZWE0jL5cMnK9HneVw
m5BC42n71gLP4Lc/edit?usp=sharing"
sheet = client.open_by_url(spreadsheet_url).get_worksheet(0)  # Hoja
1 (índice 0)

# Función para generar un ID único
def unico_ID():
    return datetime.now().strftime("%Y%m%d%H%M%S%f")  # Basado en la
fecha y hora actual

# Función para simular los datos del sensor
def simular_dato_sensor(nombre_sensor, ubicacion, tipo,
unidades_medida):
```

```
    id_unico = unico_ID()
    valor = round(random.uniform(10.0, 100.0), 2)  # Valor aleatorio
entre 10.0 y 100.0
    fecha = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    return [fecha, id_unico, nombre_sensor, ubicacion, tipo,
unidades_medida, valor]

# Función para enviar datos simulados al Google Sheets
def enviar_datos_al_sheet(datos_sensores):
    for dato in datos_sensores:
        sheet.append_row(dato)
        print(f"Datos simulados agregados: {dato}")
        time.sleep(3)  # Esperar 5 segundos antes de agregar el
siguiente dato

# Enviar los datos simulados a la hoja de cálculo
enviar_datos_al_sheet(datos_sensores)

try:
    while True:
        # Call the function with the required arguments
        datos_sensores = [
          simular_dato_sensor("Sensor_Temperatura", "Ubicación 1",
"Temperatura", "Celsius"),
          simular_dato_sensor("Sensor_Humedad", "Ubicación 2",
"Humedad", "Porcentaje"),
          simular_dato_sensor("Sensor_Presion", "Ubicación 3",
"Presión", "Pascal")]
        enviar_datos_al_sheet(datos_sensores)
        time.sleep(3)
except KeyboardInterrupt:
    print("Simulación finalizada.")

# Mostrar los datos actuales en la hoja de cálculo
datos_en_sheet = sheet.get_all_records()
print("\nDatos actuales en la hoja de cálculo:")
for fila in datos_en_sheet:
    print(fila)
```

Evidencias del resultado obtenido con el código



docs.google.com/spreadsheets/d/1xStljQV--TuZWE0jL5cMnK9HneVwm5BC42n71gLP4Lc/edit?gid=0#gid=0

Practica_5__v2

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda

	A	B	C	D	E	F	G	H
1	Fecha	ID	Nombre_Sensor	Ubicación	Tipo	Unidades	Valor	
21	2024-09-04 04:50:55	20240904045055545896	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	17.82	
22	2024-09-04 04:50:55	20240904045055546000	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	47.22	
23	2024-09-04 04:50:55	20240904045055546035	Sensor_Presion	Ubicación 3	Presión	Pascal	32.14	
24	2024-09-04 04:50:55	20240904045055545896	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	17.82	
25	2024-09-04 04:50:55	20240904045055546000	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	47.22	
26	2024-09-04 04:50:55	20240904045055546035	Sensor_Presion	Ubicación 3	Presión	Pascal	32.14	
27	2024-09-04 05:01:01	20240904050101985587	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	43.85	
28	2024-09-04 05:01:01	20240904050101985646	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	29.59	
29	2024-09-04 05:01:01	20240904050101985657	Sensor_Presion	Ubicación 3	Presión	Pascal	43.81	
30	2024-09-04 05:01:16	20240904050116176539	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	67.27	
31	2024-09-04 05:01:16	20240904050116176647	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	88.87	
32	2024-09-04 05:01:16	20240904050116176669	Sensor_Presion	Ubicación 3	Presión	Pascal	64.18	
33	2024-09-04 05:01:30	20240904050130030561	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	26.97	
34	2024-09-04 05:01:30	20240904050130030662	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	89.46	
35	2024-09-04 05:01:30	20240904050130030680	Sensor_Presion	Ubicación 3	Presión	Pascal	45.62	
36	2024-09-04 05:01:43	20240904050143735870	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	47.77	
37								
38								
39								

Nota: los siguientes sprints se realizarán como parte de un nuevo taller futuro. En todo caso, se trata de los siguientes:

- **Sprint 2.** Solución completa pero simulando el sensado y usando una ESP-32 y Arduino
- **Sprint 3.** Solución completa usando una ESP-32, Arduino y sensores disponibles