

## JSON con ESP32

Se describe un código para un microcontrolador ESP32 que lee datos de temperatura y humedad de un sensor DHT22 y los convierte en formato JSON. Este texto ampliado proporcionará detalles adicionales sobre el código, la biblioteca utilizada y la simulación en Wokwi.

**Link al proyecto:** <https://wokwi.com/projects/396998834866335745>

### Explicación del código:

#### Configuración del ESP32:

- Se incluyen las bibliotecas necesarias: DHT.h para el sensor DHT y ArduinoJson.h para trabajar con JSON.
- Se define el pin al que se conecta el sensor DHT22.
- Se crea una instancia de la clase DHT para interactuar con el sensor.

#### Lectura de datos del sensor DHT22:

- Se llama al método readTemperature() de la instancia DHT para obtener la temperatura en grados Celsius.
- Se llama al método readHumidity() de la instancia DHT para obtener la humedad relativa en porcentaje.

#### Creación del objeto JSON:

- Se crea una instancia de la clase StaticJsonDocument para almacenar los datos JSON.
- Se agregan los valores de temperatura y humedad al objeto JSON como pares clave-valor.

#### Serialización del objeto JSON:

- Se crea una variable String para almacenar la cadena JSON.
- Se llama al método serializeJson() para convertir el objeto JSON en una cadena de texto.

Impresión de la cadena JSON:

- Se imprime la cadena JSON en el puerto serie monitor.

### **Biblioteca ArduinoJson:**

- Es una biblioteca liviana y eficiente para trabajar con JSON en Arduino.
- Proporciona funciones para crear, leer, modificar y serializar objetos JSON.
- Simplifica el manejo de datos en formato JSON para aplicaciones embebidas.

### **Campos mínimos para el proyecto**

Los campos mínimos en el archivo JSON para representar la lectura de temperatura y humedad podrían ser simplemente los valores de temperatura y humedad. Aquí está un ejemplo de cómo se vería un archivo JSON mínimo para este proyecto:

```
{  
  "temperature": 25.5,  
  "humidity": 60.2  
}
```

En este JSON, "temperatura" representa la temperatura medida en grados Celsius y "humedad" representa la humedad medida en porcentaje. Estos son los campos esenciales necesarios para comunicar los datos del sensor de temperatura y humedad al ESP32 y luego serializarlos en formato JSON.

### **Conclusiones:**

El código descrito permite al ESP32 leer datos de temperatura y humedad de un sensorDHT22 y representarlos en formato JSON. La biblioteca ArduinoJson facilita el manejo de datos JSON y la simulación en Wokwi permite probar el código sin necesidad de hardware real.