



TECNICATURA SUPERIOR EN TELECOMUNICACIONES

ARQUITECTURA Y CONECTIVIDAD

Módulo II: Familia de Protocolos IoT - II

TRABAJO PRÁCTICO N°: 4

Profesor: Ing. Jorge Elías Morales.

Alumnos:

- Huk, Romina vanesa - GitHub: RoHu17
- Roldán, Patricio Leandro - GitHub: pleroldan
- Pantoja, Paola Natalia Alejandra - GitHub: PaolaaPantoja
- Paez, Tiziano Adrian - GitHub: tpaez
- Gutiérrez, Emma - GitHub: Emygut



- 1) ¿Qué es una Comunicación REST?, ¿Para qué se usan? Ejemplifique.
- 2) ¿Qué es un Formato de datos JASON?, ¿Para qué se usan? Ejemplifique
- 3) Implementar un pequeño código JASON, donde un cliente solicita una pizza. (por lo menos 4 o 5 tipos de pizza) (2 tamaños) (precios).
- 4) Implementar un código JASON, para comunicar un sensor de temperatura y humedad con un ESP32, Arduino, simulando los mismos en WOKWI, Proteus, LabView; etc. ¿Cuáles serían los campos mínimos para hacer la implementación?

1) ¿Qué es una Comunicación REST?, ¿Para qué se usan? Ejemplifique.

Una comunicación REST (Representational State Transfer) es un estilo arquitectónico para sistemas distribuidos, permitiendo la comunicación entre diferentes aplicaciones web utilizando el protocolo HTTP.

Con respecto a la comunicación, las aplicaciones web interactúan a través de una API (interfaz de Programación de Aplicaciones) que utiliza un conjunto de verbos HTTP (GET, POST, PUT, DELETE, etc.) para enviar y recibir en formato JSON (JavaScript Object Notation) o XML (Extensible Markup Language).

Se utilizan mayormente en desarrollo de aplicaciones web, móviles, IoT, entre otras.

Ejemplo:

Sistema de sensores en una granja. Un ESP32 mide temperatura y humedad, y cada cierto tiempo envía esos datos a un servidor usando REST.

Cliente (ESP32 o aplicación):

POST /datos HTTP/1.1



Host: midominio.com

Content-Type: application/json

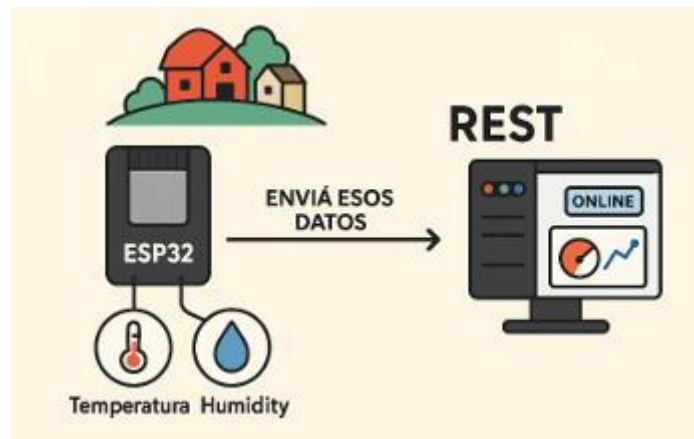
{

"temperatura": 24.5, "humedad": 60

}

Servidor (API REST):

Recibe esos datos y los guarda en una base de datos para visualizar después en una aplicación o tablero





2) ¿Qué es un Formato de datos JASON?, ¿Para qué se usan? Ejemplifica



JSON, que significa **JavaScript Object Notation**, es un formato de datos ligero y fácil de leer, utilizado para almacenar y transmitir información entre diferentes sistemas, especialmente en aplicaciones web. Se usa para transferir datos entre un servidor y un cliente web, permitiendo que las aplicaciones web interactúen con datos de forma eficiente.

¿Para qué se usan?

Transmisión de datos en aplicaciones web:

JSON es fundamental para enviar y recibir datos entre un servidor y un navegador web. Por ejemplo, cuando se envía un formulario en una página web, los datos se pueden enviar al servidor en formato JSON, y el servidor puede devolver una respuesta también en formato JSON.

Configuración de aplicaciones:

JSON se utiliza para almacenar la configuración de una aplicación, como las preferencias del usuario, los ajustes de la interfaz, etc.

Almacenamiento de datos:

Aunque no es un formato de base de datos, JSON se puede usar para almacenar datos estructurados en archivos. Intercambio de información entre diferentes sistemas: JSON es independiente de cualquier lenguaje de programación, lo que facilita el intercambio de datos entre sistemas que utilizan diferentes lenguajes.



Ejemplos:

Datos de un usuario.

Código

```
{  
  "nombre": "Juan Pérez",  
  "edad": 30,  
  "ciudad": "Córdoba"  
}
```

Lista de productos.

Código

```
[  
  {  
    "id": 1,  
    "nombre": "Laptop",  
    "precio": 1000  
  },  
  {  
    "id": 2,  
    "nombre": "Mouse",  
    "precio": 25  
  }  
]
```

Configuración de una aplicación.

Código

```
{  
  "idioma": "es",  
  "tema": "oscuro",  
  "fuente": "Arial"  
}
```



Características clave de JSON:

Facilidad de lectura: La estructura de clave-valor de JSON lo hace fácil de leer y escribir tanto para humanos como para máquinas.

Tamaño reducido: JSON es un formato ligero, lo que permite una transmisión más rápida de datos.

Compatibilidad: JSON es compatible con la mayoría de los lenguajes de programación. Trabajando con JSON - Aprende desarrollo web | MDN

¿Qué es JSON? | Oracle Argentina

JSON no es un archivo ni un código. Es un formato simple utilizado para almacenar y transportar datos.

Oracle

¿Qué es JSON? | Oracle México

Explicación de JSON JSON es un formato de datos que suelen utilizar los desarrolladores web para transferir datos.

JSON no es un archivo ni un código. Es un formato simple utilizado para almacenar y transportar datos. Es un formato de texto sin formato, que permite un fácil intercambio de datos entre diferentes lenguajes de programación. Se utiliza a menudo para enviar datos entre aplicaciones web y servidores.



3. Implementar un pequeño código JASON, donde un cliente solicita una pizza. (por lo menos 4 o 5 tipos de pizza) (2 tamaños) (precios).

```
{
  "cliente": {
    "nombre": "Pablo Castillo",
    "teléfono": "123456789",
    "dirección": "Avenida siempre viva 1234"
  },
  "pedido": {
    "pizzas_disponibles": [
      {
        "tipo": "Muzzarella",
        "tamaños": {
          "mediana": 8.00,
          "grande": 15.00
        }
      },
      {
        "tipo": "Calabresa",
        "tamaños": {
          "mediana": 8.00,
          "grande": 15.00
        }
      },
      {
        "tipo": "Cuatro Quesos",
        "tamaños": {
          "mediana": 8.00,
          "grande": 15.00
        }
      },
      {
        "tipo": "Hawaiana",
        "tamaños": {
          "mediana": 8.00,
          "grande": 15.00
        }
      },
      {
        "tipo": "Vegetariana",
        "tamaños": {
          "mediana": 8.00,
          "grande": 15.00
        }
      }
    ],
    "orden_solicitada": {
      "tipo": "Cuatro Quesos",
      "tamaño": "grande",
      "precio": 15.00
    }
  }
}
```



EXPLICACION DE LAS PARTES DEL ARCHIVO JSON

Información del Cliente

```
{
  "cliente": {
    "nombre": "Pablo Castillo",
    "teléfono": "123456789",
    "dirección": "Avenida siempre viva 1234"
  },
}
```

"cliente": Es un objeto que contiene la información de la persona que está haciendo el pedido.

Información del Pedido

```
"pedido": {
  "pizzas_disponibles": [
    {
      "tipo": "Muzzarella",
      "tamaños": {
        "mediana": 8.00,
        "grande": 15.00
      }
    }
  ]
}
```

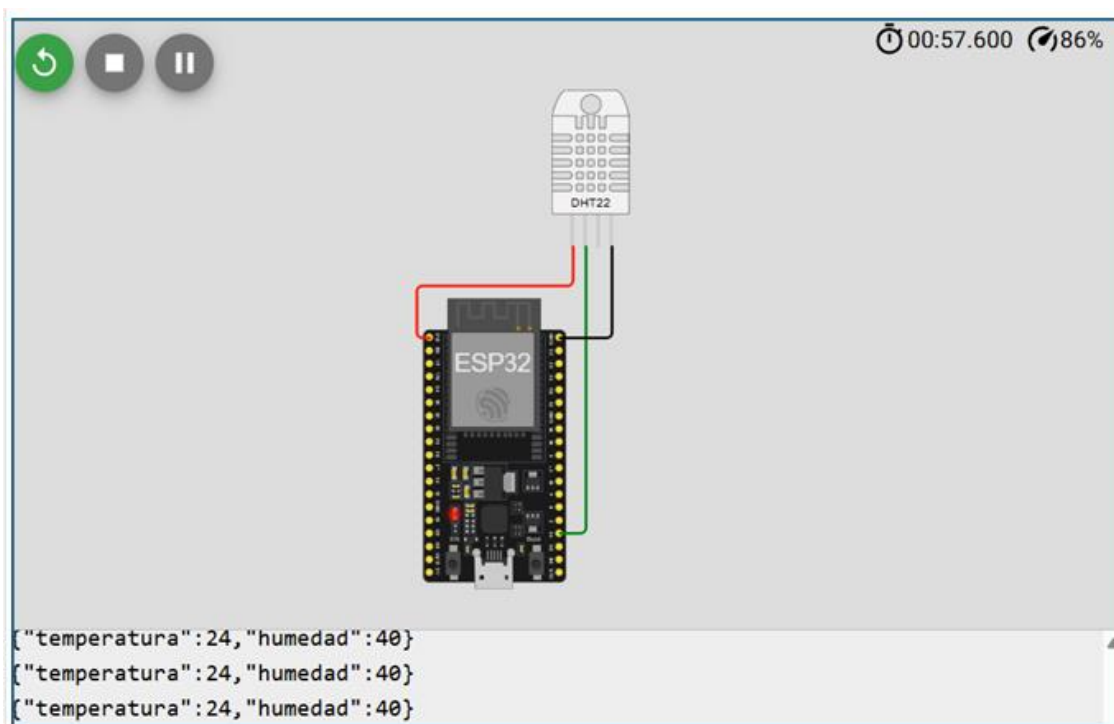
```
  "orden_solicitada": {
    "tipo": "Cuatro Quesos",
    "tamaño": "grande",
    "precio": 15.00
  }
}
```

"pedido": Detalles del pedido, que incluyen:

- **Pizzas disponibles**: Una lista de tipos de pizza con los precios de los diferentes tamaños.
- **Orden solicitada**: El tipo de pizza, tamaño y precio que el cliente ha seleccionado para su pedido.



4) Implementar un código JASON, para comunicar un sensor de temperatura y humedad con un ESP32, Arduino, simulando los mismos en WOKWI, Proteus, LabView; etc. ¿Cuáles serían los campos mínimos para hacer la implementación?



Link: <https://wokwi.com/projects/428993783396750337>

Objetivo

El objetivo de este programa es leer datos de temperatura y humedad desde un sensor **DHT22** conectado a un ESP32, y mostrar los datos en formato **JSON** a través del **Monitor Serie**. Esta implementación permite una comunicación clara y estructurada que puede ser útil para integrar con otras plataformas, como aplicaciones móviles, sistemas IoT o bases de datos.



Creación y Envío del JSON

```
StaticJsonDocument<128> json;  
json["temperatura"] = temp;  
json["humedad"] = hum;  
  
serializeJson(json, Serial);  
Serial.println();
```

Ejemplo de salida esperada –

Estos son los campos mínimos en un código JSON para comunicar un sensor de temperatura y humedad con un ESP32.

```
{"temperatura": 24.3, "humedad": 56.8}
```

Código completo

```
#include <DHT.h>  
#include <ArduinoJson.h>  
  
#define DHTPIN 15  
#define DHTTYPE DHT22  
  
DHT dht(DHTPIN, DHTTYPE);  
  
void setup() {  
  Serial.begin(115200);  
  dht.begin();  
}  
  
void loop() {  
  delay(2000); // espera 2 segundos entre lecturas  
  
  float temp = dht.readTemperature();  
  float hum = dht.readHumidity();  
  
  if (isnan(temp) || isnan(hum)) {  
    Serial.println("Error al leer el sensor");  
    return;  
  }  
  
  StaticJsonDocument<128> json;  
  json["temperatura"] = temp;  
  json["humedad"] = hum;  
  
  serializeJson(json, Serial);  
  Serial.println();  
}
```