

CURSO PROGRAMACIÓN WEB FULL-STACK

NIVEL 2

MÓDULO 5 – BackEnd

JSON

JSON (acrónimo de JavaScript Object Notation, 'notación de objeto de JavaScript') es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

Los tipos de datos disponibles con JSON son:

- **Números:** Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos. Ejemplo: 123.456
- **Cadenas:** Representan secuencias de cero o más caracteres. Se ponen entre doble comilla y se permiten cadenas de escape. Ejemplo: "Hola"
- **Booleanos:** Representan valores booleanos y pueden tener dos valores: true y false
- **null:** Representan el valor nulo.
- **Array:** Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchetes. Ejemplo ["juan","pedro","jacinto"]
- **Objetos:** Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena entre comillas dobles. El valor puede ser de cualquier tipo. Ejemplo:

```
{
  "departamento": 8,
  "nombredepto": "Ventas",
  "director": "Juan Rodríguez",
  "empleados": [
    {
      "nombre": "Pedro",
      "apellido": "Fernández"
    },
    {
      "nombre": "Jacinto",
      "apellido": "Benavente"
    }
  ]
}
```

Ejemplo JSON

En una planta de envasado de líquidos, disponen de 3 tipos de envases, 2 tipos de contenedores para empaquetar grupos de envases y 5 tipos de líquidos que pueden ser almacenados en los envases. Sin embargo, hay ciertas restricciones en los líquidos que pueden ser almacenados en cada envase y los tamaños de los envases hacen que no se pueda almacenar todos los envases en todos los contenedores.

- Tenemos envases:
 - Tipo1, apto para alimentos
 - Tipo2, resistente a sustancias corrosivas
- Tenemos contenedores:
 - Grande, 5 litros o mas
 - Mediano, de 1 a 4.99 litros
 - Pequeño, menos de 1 litro
- Tenemos líquidos:
 - Aceite, Agua, Aguarrás, Alcohol, Jabón, Leche, Lejía, Vinagre

Para estos líquidos, envases y contenedores, se puede representar la información en un archivo JSON, quedando de la siguiente manera:

```
var envasado = {
  "contenedores": [
    {
      "tipo": "Grande",
      "envases": [
        {
          "tipo": "Tipo1",
          "liquidos": [
            {
              "tipo": "alimento",
              "text": "Aceite"
            },
            {
              "tipo": "alimento",
              "text": "Agua"
            },
            {
              "tipo": "alimento",
              "text": "Leche"
            },
            {
              "tipo": "alimento",
              "text": "Vinagre"
            }
          ]
        },
        {
          "tipo": "Tipo2",
          "liquidos": [
            {
              "tipo": "corrosivo",
              "text": "Aguarrás"
            },
            {
              "tipo": "corrosivo",
              "text": "Alcohol"
            },
            {
              "tipo": "corrosivo",
              "text": "Jabón"
            }
          ]
        }
      ]
    }
  ]
}
```

```

        {
            "tipo": "corrosivo",
            "text": "Lejía"
        }
    ]
}
],
},
{
    "tipo": "Mediano",
    "envases": [
        {
            "tipo": "Tipo1",
            "liquidos": [
                {
                    "tipo": "alimento",
                    "text": "Aceite"
                },
                {
                    "tipo": "alimento",
                    "text": "Agua"
                },
                {
                    "tipo": "alimento",
                    "text": "Leche"
                },
                {
                    "tipo": "alimento",
                    "text": "Vinagre"
                }
            ]
        },
    ],
},
{
    "tipo": "Tipo2",
    "liquidos": [
        {
            "tipo": "corrosivo",
            "text": "Aguarrás"
        },
        {
            "tipo": "corrosivo",
            "text": "Alcohol"
        },
        {
            "tipo": "corrosivo",
            "text": "Jabón"
        },
        {
            "tipo": "corrosivo",
            "text": "Lejía"
        }
    ]
}
],
},
{

```

```

"tipo": "Pequeño",
"envases": [
  {
    "tipo": "Tipo1",
    "liquidos": [
      {
        "tipo": "alimento",
        "text": "Aceite"
      },
      {
        "tipo": "alimento",
        "text": "Agua"
      },
      {
        "tipo": "alimento",
        "text": "Leche"
      },
      {
        "tipo": "alimento",
        "text": "Vinagre"
      }
    ]
  },
  {
    "tipo": "Tipo2",
    "liquidos": [
      {
        "tipo": "corrosivo",
        "text": "Aguarrás"
      },
      {
        "tipo": "corrosivo",
        "text": "Alcohol"
      },
      {
        "tipo": "corrosivo",
        "text": "Jabón"
      },
      {
        "tipo": "corrosivo",
        "text": "Lejía"
      }
    ]
  }
]
}

```

Para dar formato y validar se pueden emplear herramientas online:

<https://jsonformatter.org/>

Ejercicio 1:

Crear un archivo JSON llamado personas.json que contenga los siguientes datos:

Nombre	Apellido	DNI	Telefono	Localidad
Pedro	Guzman	22773450	3755121314	L. N. Alem
Maria	Nieves	35928532	3764131517	Posadas
Roberto Carlos	Guerra	40726384	3758171819	Apóstoles
Julieta	Lopez	29384710	3751222324	Eldorado
Ramon	Rodriguez	31749223	3765334455	Posadas

Ejercicio 2:

Crear un archivo JSON llamado “departamentos.json” que contenga la siguiente información:

Contabilidad:

id_empleado: 10; nombre: Juan Segovia

id_empleado: 15; nombre: Ramiro Romero

Tecnología:

id_empleado: 12; nombre: Pablo Richmon ; puesto: PM

id_empleado: 23; nombre: Marta Fernandez; puesto: TechLead

id_empleado: 24; nombre: Pedro Mendez ; puesto: FullStack JS Producción:

id_empleado: 7; nombre: Gabriela Lescano

Comercial:

id_empleado: 2; nombre: Fabiana Martines

Ejercicio 3:

Crear un archivo JSON llamado productos.json que contenga la siguiente información:

- id_producto: 283; nombre: Gafas de sol; Marca: Acme; disponible: 20; precio: 9800; temporada: Verano 2023;
 - características: color: negro; marco: metal;
- id_producto: 191; nombre: Reloj deportivo; Marca: Timer; disponible: 2; precio: 21400;
 - características: color: plateado; material: goma y plástico;
- id_producto: 148; nombre: Juego de medias; Marca: Piecito; disponible: 38; precio: 500;
 - características: color: varios; material: algodón;
- id_producto: 82; nombre: Camiseta selección Argentina; Marca: Argentó; disponible: 11; precio: 6800; talle: L
- id_producto: 81; nombre: Camiseta selección Argentina; Marca: Argentó; disponible: 15; precio: 6800; talle: M

Node.JS

A partir de este punto es necesario tener instalado en nuestro entorno node y npm.
<https://kinsta.com/es/blog/como-instalar-node-js/>

Ejercicio 4:

- a) Crear una carpeta llamada “nodejs” en algún lugar de preferencia en su PC.
- b) Abrir dicha carpeta con el Visual Studio Code y en su interior crear el archivo llamado index.js con el siguiente contenido:

```
console.log('Hola soy un programa')
```

- c) Posicionados sobre la carpeta “nodejs”, ejecutar el comando “npm init” en la terminal. Seguir los pasos por defecto. (Esto creará un archivo package.json)
- d) Luego, dentro del archivo “package.json”, editar el objeto “scripts” y agregar la línea: **"start": "node index.js"** quedando:

```
{ // ...  
  "scripts": {  
    "start": "node index.js",  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  // ... }
```

- e) Utilizando la terminal, ejecutar: **node index.js**
- f) Luego ejecutar **npm start** (esto debería ejecutar el archivo de una manera similar a la anterior)

Ejercicio 5:

- a) Dentro de la carpeta nodejs, crear un archivo llamado “programa.js” y declarar la siguiente función:

```
// Declaro la función sumar y paso como parametro 2 numeros  
function sumar(num1, num2) {  
  resultado = num1 + num2;  
  console.log("La suma es: "+resultado);  
}  
  
// Declaro la explotación de la función  
module.exports = {  
  "sumar": sumar  
}
```

- b) En el archivo “index.js”, agregar el siguiente contenido:

```
const operacion = require("./programa")  
// Llamado al objeto operacion  
operacion.sumar(4, 3);
```

- c) Ejecutar **npm start**