

MP_0373.
**Lenguajes de marcas y sistemas de
gestión de información**

**UF3. Lenguajes de Almacenamiento y
transmisión de información**

3.3. JSON

Índice

☰	Objetivos	3
☰	¿Qué es JSON?	4
☰	Sintaxis JSON	6
☰	Tipos de datos	9
☰	JSON frente a XML	10
☰	Resumen	12

Objetivos

Conocer la estructura y características de los documentos JSON.

¿Qué es JSON?

JSON es una notación que se basa en la forma que tiene JavaScript de guardar los datos en los objetos. Nos permite crear estructuras de datos de forma fácil y rápida. Y lo que es más importante, de forma óptima.

Sintaxis

Para crear un documento JSON debemos seguir las siguientes reglas:

- Los datos deben estar separados por comas.
- Los datos se pondrán con la estructura "*clave:valor*".
- Tanto las *claves* como los *valores* irán entre comillas.
- Los *objetos* JSON se definen por llaves "{}".
- Los *arrays* van con corchetes [].
- Tanto los *objetos* como los *arrays* pueden contener otros *objetos* o *arrays*.

```
{  
    "nombre": "Jose",  
    "edad": "20",  
    "genero": "masculino",  
    "email": "correodejose@dominio.com",  
    "localidad": "Madrid",  
    "telefono": "91000000",  
    "mando": "true",  
    "marco": "<iframe width=560 height=315 src=https://www.youtube.com/embed/QCEU-vba8mw  
frameborder=0 allowfullscreen></iframe>"  
}
```

Sintaxis JSON

La sintaxis JSON es igual a la de JavaScript. Los datos se ponen en pares, como *clave* y *valor*.

```
"nombre":"valor"
```

1

Para definir un objeto utilizamos las llaves {}.

```
{
  "nombre":"valor"
}
```

2

Para definir varios elementos los separaremos por comas ",".

```
{
  "nombre":"Juan",
  "apellidos":"Gonzalez"
}
```

3

También podemos poner varios objetos, igualmente los sepáramos por comas ", ".

```
{  
  "nombre":"Juan",  
  "apellidos":"Gonzalez"  
},  
{  
  "nombre":"María",  
  "apellidos":"Alonso"  
},  
{  
  "nombre":"José",  
  "apellidos":"amohedo"  
}
```

4

JSON permite la definición de *arrays* con el uso de los corchetes [].

```
{  
  "nombre":"juan",  
  "edad":30,  
  "coches":[ "Ford", "BMW", "Fiat"]  
}
```

5

Los *arrays* pueden contener tanto valores como objetos.



Recuerda, un *array* es un conjunto de elementos organizados en memoria en posiciones contiguas. Todos los elementos son del mismo tipo y se puede acceder a ellos a partir de subíndices.

```
{  
  "nombre":"juan",  
  "edad":30,  
  "coches": [  
    { "nombre":"Ford", "modelo":[ "Fiesta", "Focus", "Mustang" ] },  
    { "nombre":"BMW", "modelo":[ "320", "X3", "X5" ] },  
    { "nombre":"Fiat", "modelo":[ "500", "Panda" ] }  
  ]  
}
```

Tipos de datos

JSON deriva de JavaScript, pero solo en el modelo de datos, por lo que elementos como funciones o datos no definidos no están contemplados.

Los tipos de datos soportados por JSON son:

- *string*: cadenas de texto.
- *number*: datos numéricos.
- *object*: objetos JSON.
- *arrays*: arrays, vectores, etc.
- *boolean*: datos booleanos (true, false).
- *null*: valores nulos.

JSON frente a XML

Tanto JSON como XML se usan para intercambiar información entre ordenadores o programas, además de almacenar datos, como configuraciones de un servidor o aplicación, etc.

Inicialmente podemos pensar que JSON, al ser más breve a la hora de definirlo, es mejor que XML. Pero los dos están profundamente implantados en todos los lenguajes y sistemas informáticos.



JSON tiene una dependencia profunda de los lenguajes de programación, dado que necesita ser tratado para obtener la información, mientras que con XML podemos definir reglas y está más abierto a los sistemas.

Veamos un ejemplo con la misma información en los dos sistemas:

```
{"employees": [  
    { "firstName":"John", "lastName":"Doe" },  
    { "firstName":"Anna", "lastName":"Smith" },  
    { "firstName":"Peter", "lastName":"Jones" }  
]
```

Ejemplo JSON.

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

Ejemplo XML.

Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado.

En esta unidad hemos aprendido:

- 1 Estructura y sintaxis de los datos en JSON.
- 2 Tipos de datos soportados por JSON.
- 3 Algunos ejemplos de aplicación.



PROEDUCA