

# INTRODUCCIÓN A JSON

LIMA UD3 – Tema 1

IES Plurilingüe Antón Losada Diéguez

Adrián Fernández González



**Tabla de contenido**

1. Introducción.....2

2. Sintaxis.....2

    2.1. Datos primitivos .....2

    2.2. Arrays.....2

    2.3. Objetos .....3

# Introducción a JSON

---

## 1. Introducción

JSON (JavaScript Object Notation) es un formato ligero y sencillo para el intercambio de datos.

Aunque inicialmente nació dentro del estándar de JavaScript, se independizó como formato propio.

Fue diseñado para ser fácil de leer y escribir por humanos y por máquinas.

Es una alternativa a XML que está eclipsando al lenguaje de marcas debido a su traducción casi inmediata y la facilidad para crear analizadores (*parser*) en cualquier lenguaje. Además, de estar presente de base en los navegadores que dispongan de JavaScript.

## 2. Sintaxis

JSON se basa en tres elementos fundamentales, los datos primitivos, los arrays y los objetos.

```
{
  "base": {
    "nombre": "",
    "apellidos": "",
    "cumple": ""
  },
  "datos de vida": {
    "madre": "",
    "padre": "",
    "infancia": [
      ""
    ],
    "adolescencia": [
      ""
    ],
    "edad adulta": [
      ""
    ]
  },
  "curriculum": {
    "estudios": [
      ""
    ],
    "trabajo": [
      ""
    ],
    "idiomas": [
      ""
    ],
    "licencias": [
      ""
    ],
    "otros": [
      ""
    ]
  }
}
```

### 2.1. Datos primitivos

Se representan como una pareja de clave-valor, por ejemplo "nombre": "Pepe"

```
"nombre": "",
"apellidos": "",
"cumple": ""
```

### 2.2. Arrays

Se representan como una lista de elementos entre corchetes [] y separados por comas.

Cualquier tipo de elemento puede ser un *array*, siempre que carezca de clave y esté entre corchetes.

```
"estudios": [  
    "",  
],
```

En este ejemplo, estudios es un *array* en el que se incluirá un título por línea.

### 2.3. Objetos

Los objetos están representados por llaves que engloban elementos tipo clave-valor. Cualquier tipo de elemento puede formar parte de un objeto, siempre que respete el formato de clave-valor.

```
"base": {  
    "nombre": "",  
    "apellidos": "",  
    "cumple": ""  
},
```

Como puede verse en el ejemplo, si el objeto forma parte de otro objeto tendrá un nombre "nombre": {} siguiendo la estructura clave-valor, siendo el valor el propio objeto.