

# Introducción:

JSON (acrónimo de JavaScript Object Notation) es un formato de texto ligero para el intercambio de datos, este se considera un lenguaje independiente.

Gracias a la función eval () es mucho más sencillo escribir un analizador sintáctico respecto a XML, gracias a esto es operativo en la mayoría de navegadores web. JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia, también se suele implementar JSON junto a XML.

## Sintaxis

**Los tipos de datos disponibles con JSON son:**

**Números:** permite números negativos y opcionalmente parte fraccional separados por puntos.

**Cadenas:** cadenas de cero o más caracteres y se escriben entre dobles comillas. ( " ")

**Booleanos:** valores booleanos cuales valores pueden ser true o false.

**Null:** valor nulo.

**Array:** lista ordenada de cero o más valores que pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchete. ( [ " ", " " ] )

**Objetos:** colecciones ordenadas de pares de la siguiente forma: <nombre>:<valor>, separados por comas y puestas entre llaves. El nombre tiene que ser una cadena y el valor puede ser de cualquier tipo. { "nombre": "valor", "nombre": "valor" }

## Modelos de Procesamiento:

Al ser JSON un formato muy extendido para el intercambio de datos, se han desarrollado API para distintos lenguajes que permiten analizar sintácticamente, generar, transformar y procesar este tipo de dato.

**-Modelo de objeto:** El JSON completo es almacenado en memoria en un formato de árbol. Este árbol es navegado, analizado y modificado con las API apropiadas. Este modelo necesita emplear muchos recursos debido a su estructura de trabajo.

**-Modelo de flujo:** Los datos son leídos o escritos en bloques. El analizador genera eventos apropiados para indicar el tipo de bloque de que se trata. El cliente puede procesar el contenido escuchando los eventos apropiados.

**-Convirtiendo los objetos JSON en objetos del lenguaje:** En java esto es realizado por ejemplo por las bibliotecas Jackson y Gson.

## USO de JSON:

Lo que en este apartado nos explica es que es muy importante analizar JSON con JavaScript y que para ello existe una función encargada de ello (*JSON.parse()*), pero debemos tener en cuenta que esta función utiliza *eval()* pues esta ve expresiones seguras. Después nos pone un ejemplo de cómo funcionaría *JSON.parse()* por dentro.