



¿Qué es un formato de datos JSON?

¿Para qué se usan? Ejemplifique

¿Qué es JSON?

JSON es un formato de texto que forma parte del sistema de JavaScript y que se deriva de su sintaxis, pero no tiene como objetivo la creación de programas, sino el acceso, almacenamiento e intercambio de datos. Usualmente es conocido como una alternativa al lenguaje XML.

JSON son las siglas de JavaScript Object Notation, o sea, Notación de Objeto Javascript.

La sintaxis de JSON funciona de modo similar a JavaScript. Por ejemplo:

- El arreglo de información se hace mediante claves (keys).
- La información se separa por comas.
- Las llaves agrupan objetos.
- Los corchetes agrupan arreglos de datos.

Sin embargo, JSON se distingue de JavaScript porque sus claves tienen que ser strings (o secuencias de caracteres), mientras que sus valores deben ser strings, números, objetos, arreglos, booleans o null.

¿Para qué sirve un archivo JSON?

Un archivo JSON es un documento digital creado en este lenguaje que almacena información organizada, con el fin de hacer más simple su búsqueda y acceso. La ventaja de este formato es que permite obtener código legible para las personas con nombres y valores que funcionan como indicadores de la información que contienen.

¿Dónde se utiliza JSON?

Como hemos visto, los archivos JSON permiten obtener un código legible de almacenamiento y también son útiles para manipular la información de un programa a la hora de crear un software.

Originalmente, JSON era utilizado únicamente bajo la notación basada en objetos de JavaScript. Actualmente, debido a su popularidad, muchos lenguajes de programación son compatibles con el formato JSON.

Los archivos en este formato suelen tener la terminación *.json* y son especialmente provechosos para intercambiar o transferir información a lo largo de diferentes tipos de dispositivos digitales.

El caso más común de uso de JSON está en el diseño de sitios web. Al crear páginas en línea queremos asegurarnos de que el sitio lea correctamente la información contenida en el servidor y que la muestre de forma óptima. Además, es deseable que el programador pueda modificar el código durante la marcha para corregir errores.

Pero esta no es su única función. Por ejemplo, podemos emplear JSON para la creación de aplicaciones móviles y programas computacionales o incluso para la transferencia de documentos. Esta herramienta es tan versátil que podríamos asegurar que está prácticamente en todos lados.

Ejemplo de código JSON

Aquí podemos ver un ejemplo de un código en formato JSON.

```
{
  "nombre":"Jonh Doe",
  "profesion":"Programador",
  "edad":25,
  "lenguajes":["PHP","Javascript","Dart"],
  "disponibilidadParaViajar":true,
  "rangoProfesional": {
    "aniosDeExperiencia": 12,
    "nivel": "Senior"
  }
}
```

Los archivos JSON pueden tener cualquier nivel de anidación que sea necesario, ya que podemos colocar unos objetos dentro de otros, creando un árbol de datos de cualquier profundidad.

Reglas de sintaxis en JSON

Los archivos en formato JSON tienen las mismas reglas de sintaxis del lenguaje Javascript. Pero además necesitan cumplir algunas normas adicionales, que no son necesarias en los literales de objeto Javascript:

Estructuras disponibles

En JSON podemos estructurar los datos de dos maneras distintas, como arrays y objetos. Todos los archivos de JSON representan una de estas dos estructuras:

1. On objeto, es decir, una lista de pares clave / valor.
2. Una colección de elementos, lo que se conoce como array o arreglo.

Esto quiere decir que no puedes poner un valor primitivo (números, cadenas, booleanos) tal cual como un JSON válido, sino que tendrá que estar dentro de una estructura permitida, como un objeto o un array.

Llaves para representar objetos

Los archivos JSON que representan objetos comienzan siempre con una llave de inicio { y acaban con la llave de cierre }.

```
{  
  "id": 44,  
  "game": "Colonos de Catan",  
  "author": "Klaus Teuber",  
  "ages": "10+"  
}
```

Corchetes para representar los arrays

Los archivos JSON que representan una lista de valores, es decir, un arreglo, comienzan por [y terminan por].

```
[ 1, 2, 8 ]
```

El anterior array es un JSON válido, pero es más común el uso de valores complejos en las casillas del array. De hecho, dentro de una casilla podríamos tener cualquier otro valor JSON válido, incluso otros objetos complejos.

```
[  
  {  
    "id": 56431  
    "name": "El Padrino",  
    "year": 1972  
  },  
  {  
    "id": 7553  
    "name": "El Padrino 2",  
    "year": 1974  
  },  
  {  
    "id": 19563
```

```
"name": "El Padrino III",  
  "year": 1990  
},  
]
```

Valores representables con JSON

Todos los valores que podemos encontrar en JSON representan una de estas posibilidades:

1. Un valor primitivo, que pueden ser de tipos diversos de Javascript, numérico, cadena de caracteres, y booleanos (true / false).
2. Arrays, es decir, una secuencia de valores separados por comas. En los arrays utilizamos los corchetes para delimitar el principio y fin de los valores que contienen.
3. Objetos, es decir, otros elementos JSON con pares clave / valor.
4. Además, también es válido el valor null.

Uso de comillas dobles

Se deben usar siempre comillas dobles a la hora de encerrar cadenas y nombres de los atributos del objeto.

Este punto es importante, porque la sintaxis de Javascript para los literales de objetos permite colocar los nombres de las propiedades con y sin comillas, mientras que en JSON estamos obligados a colocar los nombres de las propiedades siempre con comillas dobles.

```
{  
  "id": 1,  
  "name": "Patatas nuevas",  
  "description": "Patatas cosechadas este año",  
  "available": true  
}
```

En el código anterior, los nombres de las propiedades "id", "name", etc. se deben colocar con comillas dobles. Todos los nombres de los atributos del objeto deben tener comillas necesariamente. Además, las cadenas también se deben colocar con comillas dobles.

Comas para separar los pares clave / valor

Cada elemento del objeto JSON se separa del siguiente con una coma (,). Pero no debe haber una coma después del último elemento en ningún caso (Javascript sí que la permitiría).

UTF-8 obligatorio

Todos los archivos con código JSON deben estar codificados con UTF-8 de manera obligatoria. El archivo transferido en un JSON debe ser necesariamente codificado como UTF8 para que no haya problemas de visualización de los caracteres contenidos en él.

¿Cómo abrir un archivo JSON?

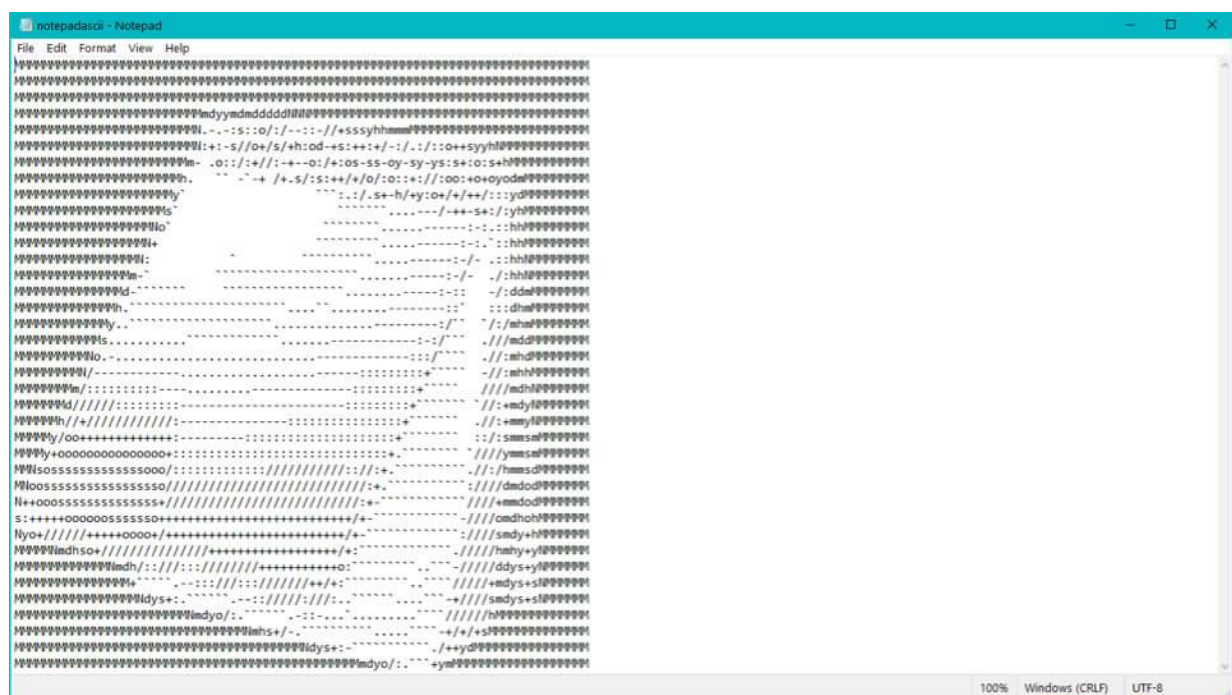
Hasta ahora hemos descrito qué es un archivo JSON y cómo luce en diferentes situaciones, pero no hemos hablado de la forma en que un sistema puede leer esa información y transferirla a un nuevo equipo. En realidad es muy simple.

Debido a que JSON está estructurado básicamente como un formato de texto (motivo por el que solo puedes trabajar con strings y números en él) únicamente necesitas un programa capaz de leer estos datos.

¿Estos programas son software especializado? Claro que no. Programas tan simples como el bloc de notas o un editor de textos son ideales para leer estos archivos. Veamos algunos de los más populares.

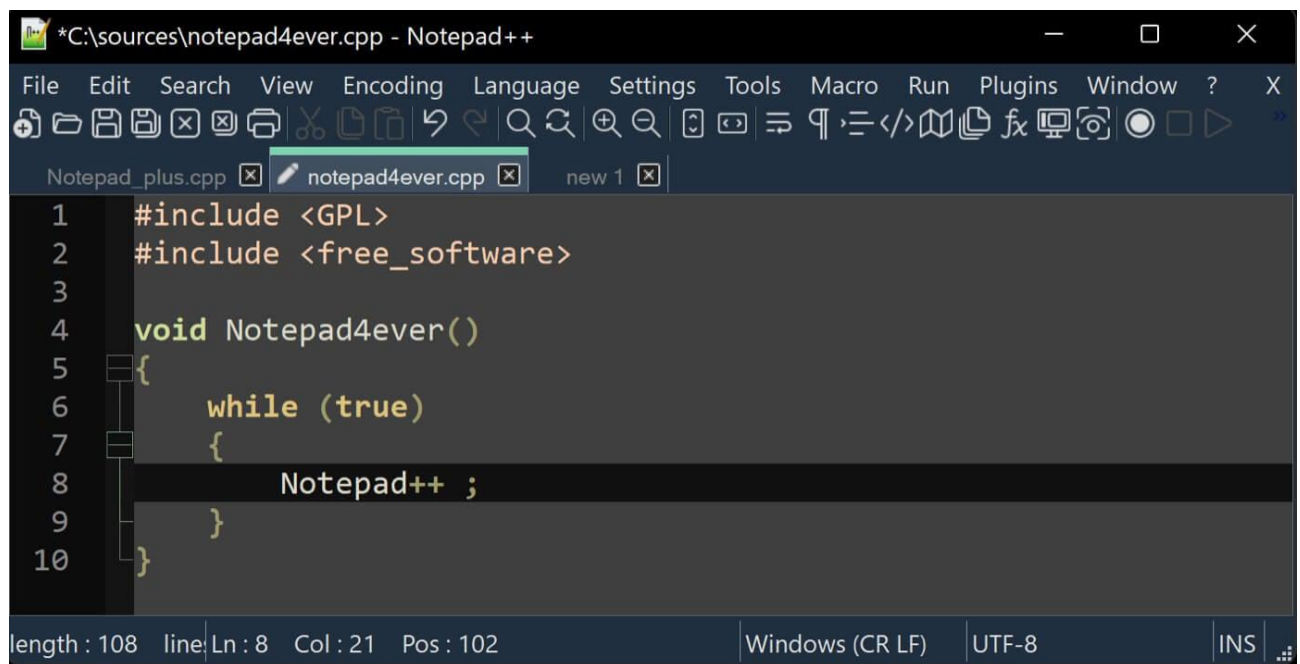
1. Windows Notepad

Es un editor de texto muy simple creado por Windows especialmente para su propio software. Pero debido a la gran simplicidad de los archivos JSON, el Notepad de Windows no tendrá ningún problema para abrirlos ni editarlos.



2. Notepad++

Otro programa realmente simple para visualizar y alterar código fuente es Notepad++. Esta herramienta se distingue del Windows Notepad porque es mucho más flexible para la edición. Debido a que el programa está optimizado para C++, puede procesar archivos JSON sin mayor complicación.

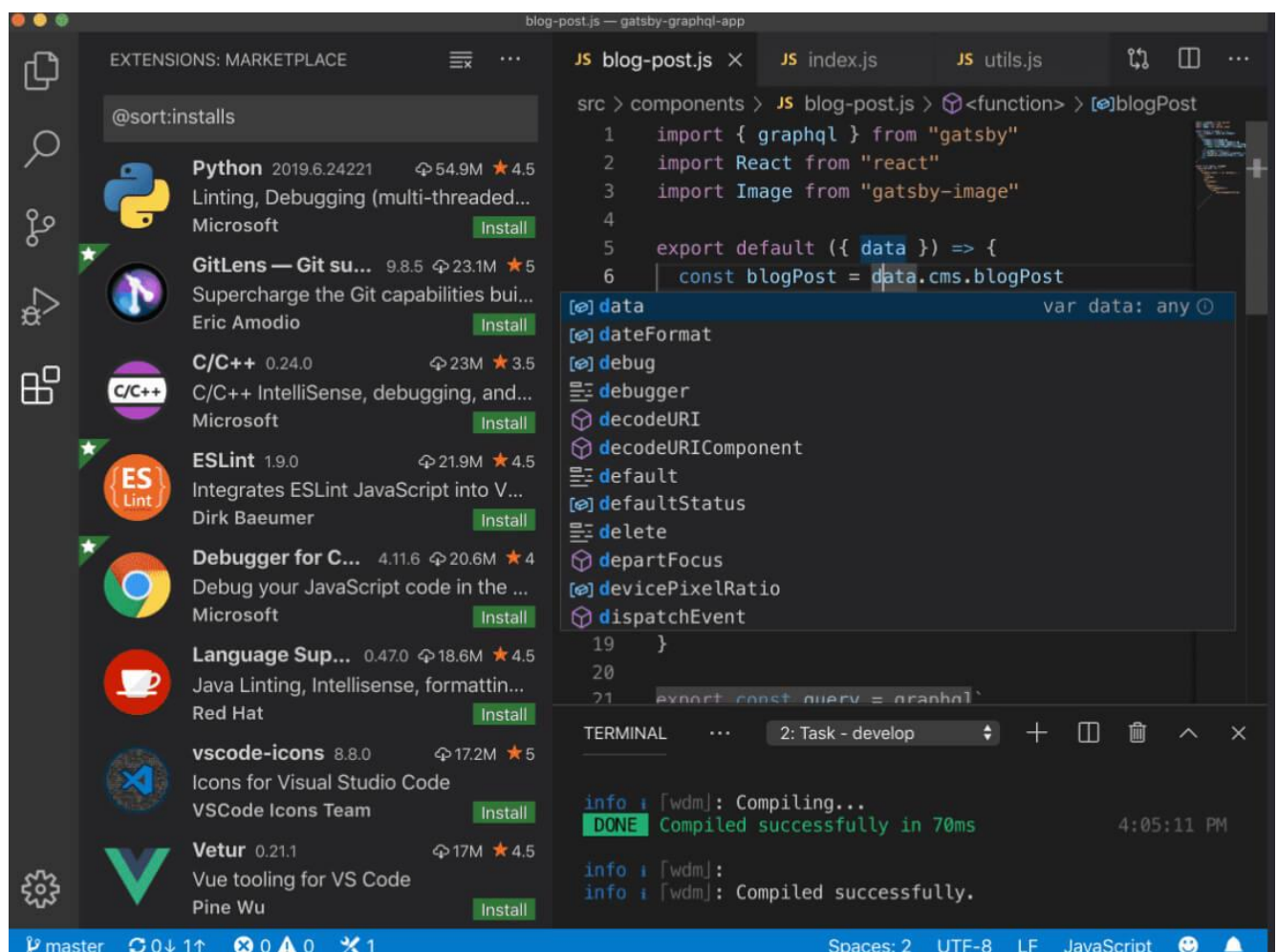


```
1 #include <GPL>
2 #include <free_software>
3
4 void Notepad4ever()
5 {
6     while (true)
7     {
8         Notepad++ ;
9     }
10 }
```

length : 108 line: 8 Col : 21 Pos : 102 Windows (CR LF) UTF-8 INS

3. [Visual Studio Code](#)

Una herramienta mucho más potente que las anteriores es Visual Studio Code. Este programa es un complejo editor de texto que interactúa con diferentes tipos de información y lenguajes de programación. Por ello es tan útil a la hora de abrir y editar archivos JSON.



Al considerar el uso de archivos JSON debes considerar que son realmente simples. Esto representa una gran ventaja, ya que optimiza la compatibilidad de la información a través de diferentes dispositivos y mejora el rendimiento de los programas.

En resumen, JSON ofrece las siguientes ventajas:

- Simplicidad: la sintaxis de JSON es muy simple de usar y leer.
- Rendimiento: el software que usa JSON para manejar datos tiene un mejor rendimiento, ya que la información se puede procesar rápidamente.
- Funcionalidad: a pesar de que tiene sus limitaciones, JSON es lo suficientemente versátil como para cubrir muchas aplicaciones y necesidades.
- Accesibilidad: al ser un formato de texto, es fácil de leer tanto para humanos como para máquinas, por lo que no tendrás problema para acceder a tus datos o para manipularlos.