

JSON

(JavaScript Object Notation)

- Es un formato ligero de intercambio de datos.
- Es independiente del lenguaje.
- Es, en general, menos pesado que XML.
- Es un formato compuesto por texto plano, por tanto fácil de leer y de escribir.
- Su sintaxis es simple.
- Posee una estructura jerárquica.
- Es un subconjunto de JavaScript.

- Permite el intercambio de datos complejos, debido a su característica de estructura jerárquica.
- No es necesario construir parsers complejos, como sucede en XML.
- Su uso es básico en aquellos entornos donde el flujo de información entre proceso cliente y servidor está limitado, y es un requerimiento importante.
- NO es un lenguaje de marcas.
- NO define funciones.
- NO tiene espacios de nombres.
- Existe un lenguaje de validación (JSONSchema).

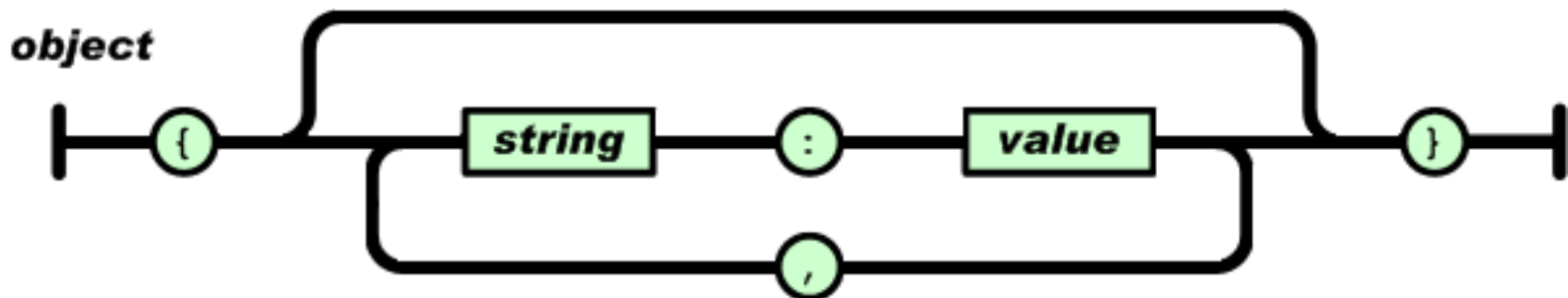
- Algunos lenguajes que lo soportan son:
 - C / C++
 - .NET (C#, VB.NET...)
 - Delphi
 - **Java**
 - JavaScript
 - Perl
 - PHP
 - Python
 - Ruby
 - ...
- Existen diferentes API para cada uno de los lenguajes.

- RFC7159 available in <https://www.ietf.org/rfc/rfc7159.txt>
- Recommended website of JSON: <http://www.json.org/>

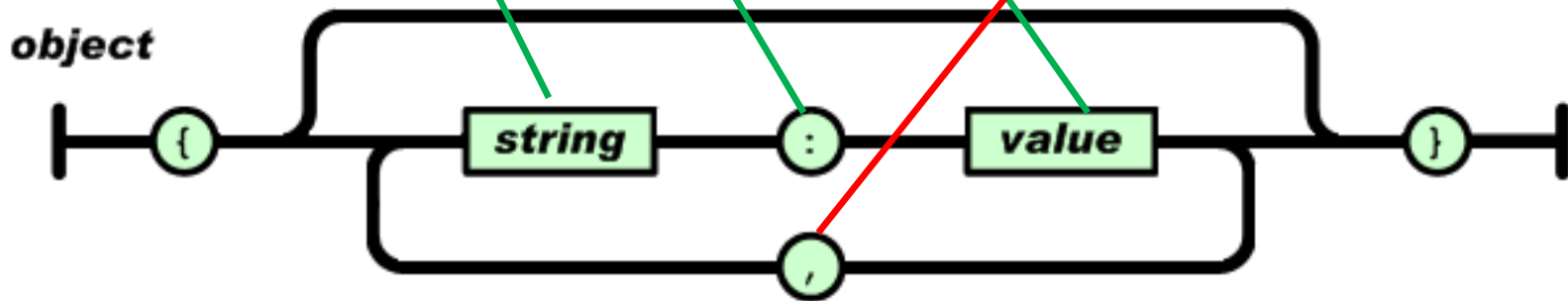
- JSON está constituido por dos estructuras:
 - Una colección de pares *nombre* ↔ *valor*: se tratará como un OBJETO.
 - Una lista ordenada de valores: se tratará como un ARRAY.

<http://www.json.org/>

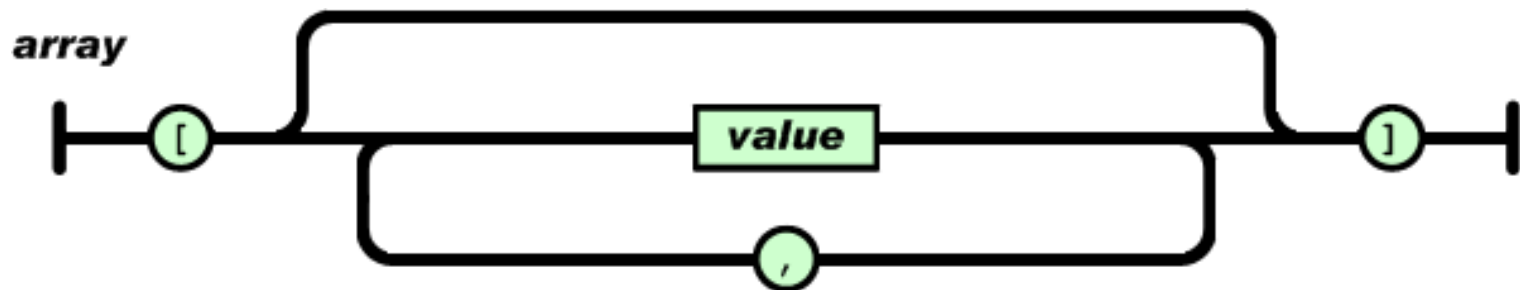
- Un Objeto es un conjunto no ordenado de pares *nombre* ↔ *valor* (propiedad).
- Notación:
 - Comienza con una llave de apertura {
 - Acaba con una llave de cierre }
 - Sus propiedades:
 - Se separan con comas ,
 - El nombre y el valor están separados por dos puntos :

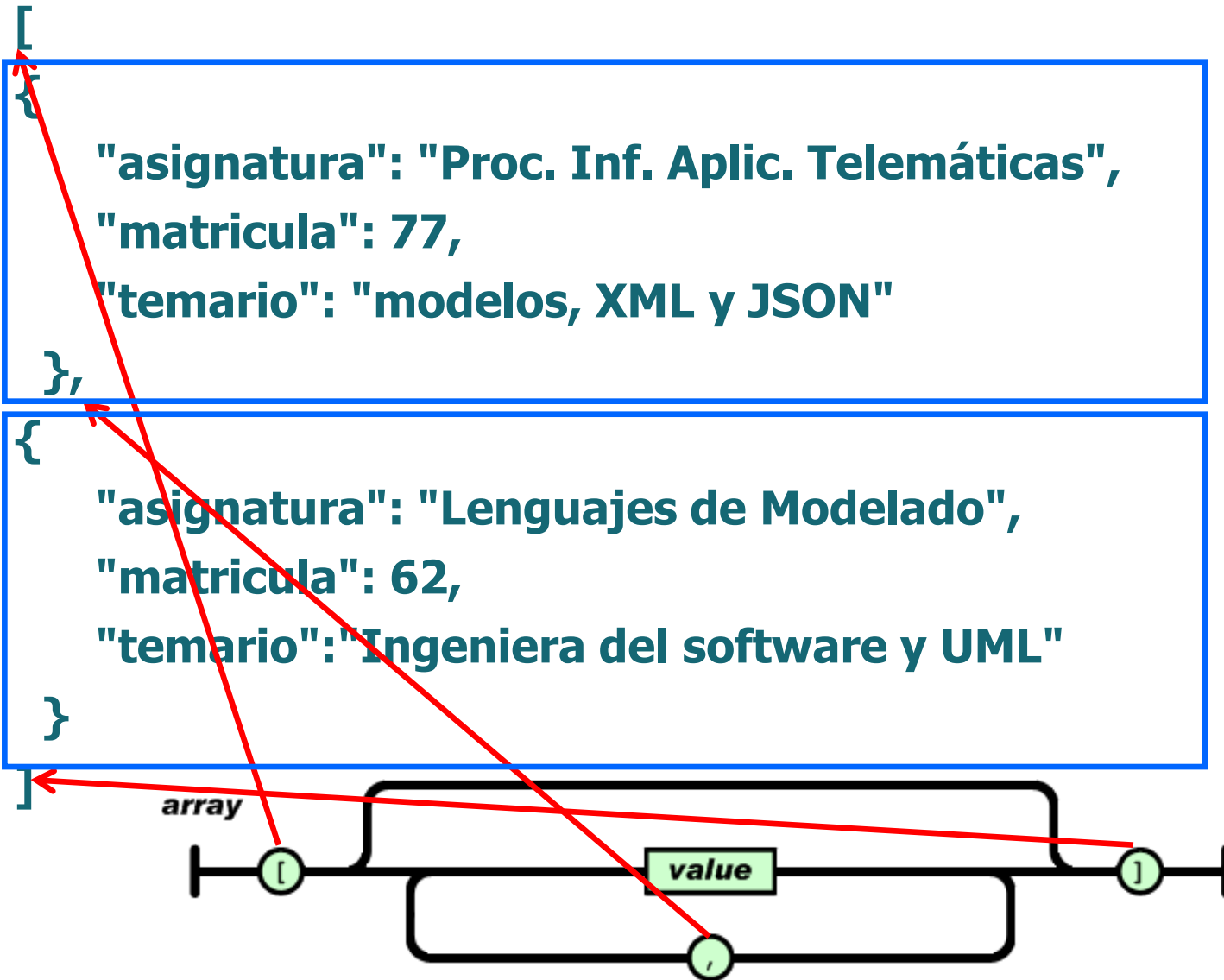


```
{  
  "asignatura": "Aplicaciones Telematicas",  
  "matricula": 77,  
  "temario": "modelos, XML y JSON"  
}
```



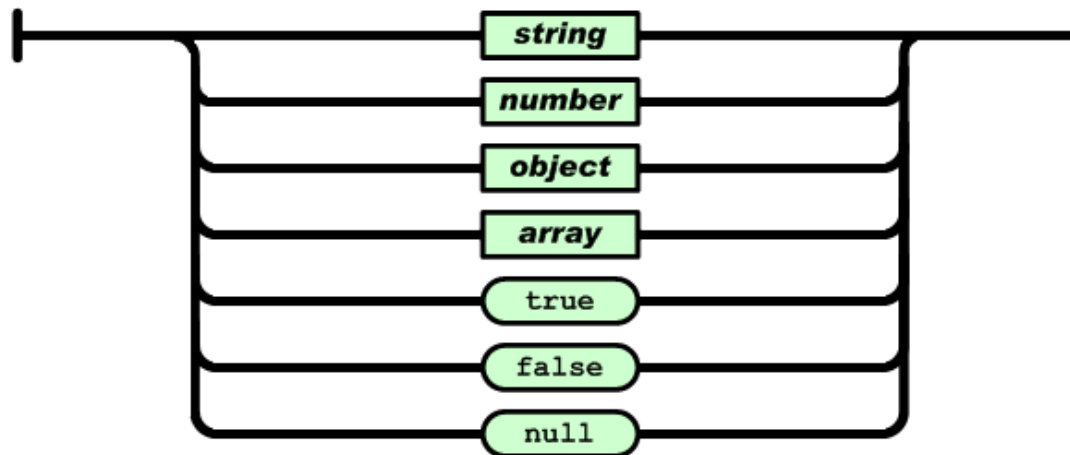
- Un Array es una colección ordenada de valores o de objetos
- Notación
 - Comienza con un corchete izquierdo [
 - Acaba con un corchete derecho]
 - Los valores se separan con una coma ,





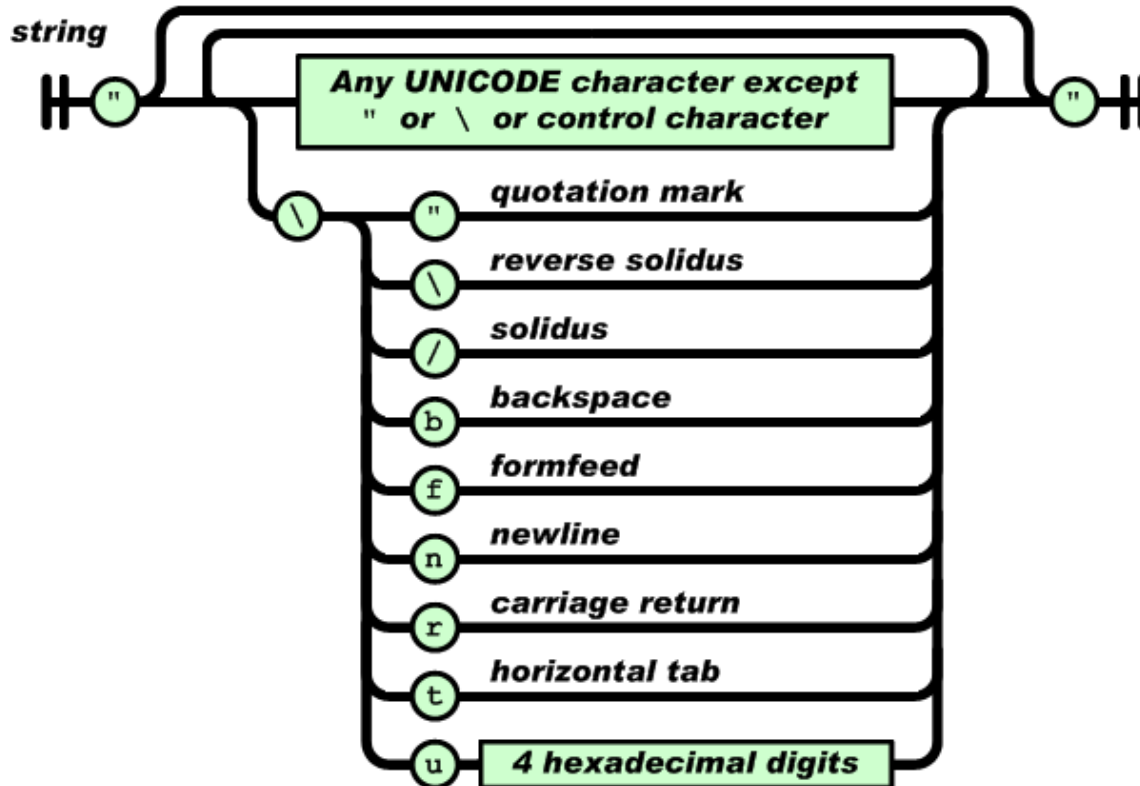
- Puede ser
 - Una cadena de caracteres con comillas dobles
 - Un objeto
 - Un array
 - Un número
 - Uno de estos: true, false, null

value



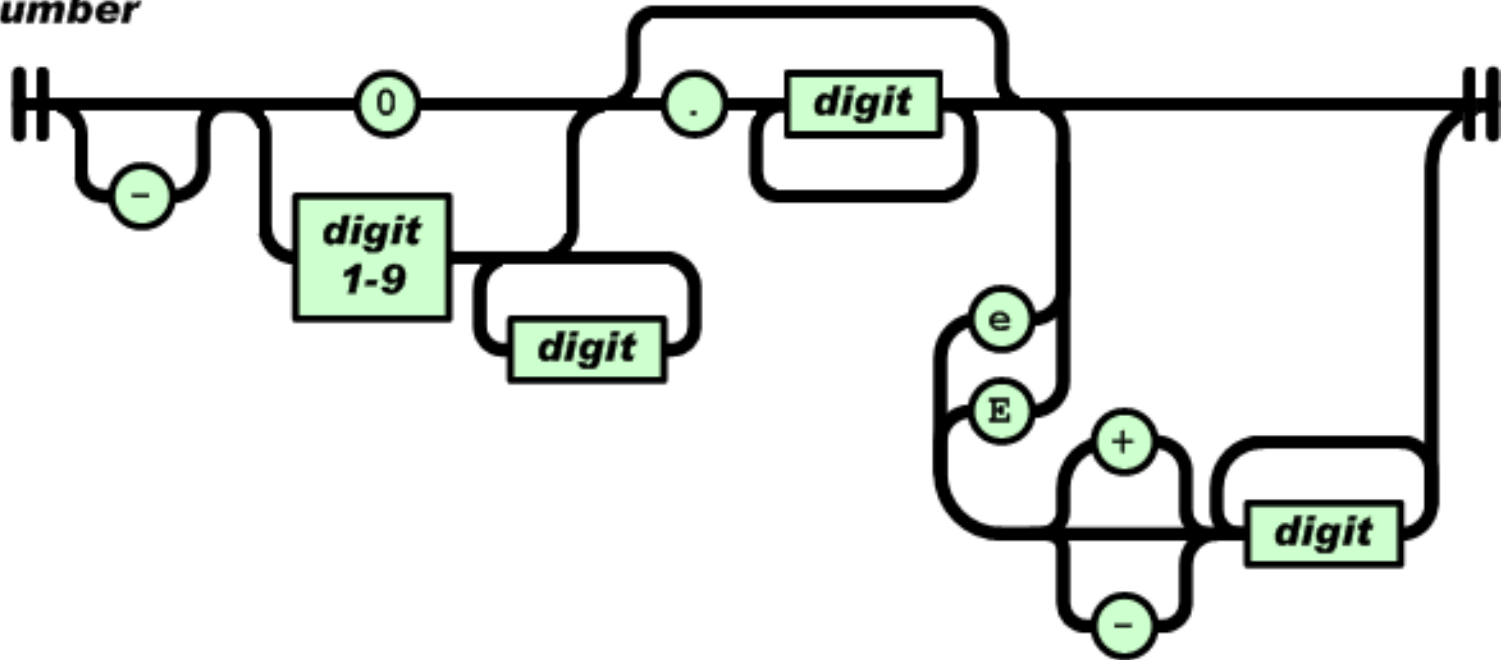
- Estas estructuras pueden anidarse.

- Un String es una colección de cero a mas caracteres Unicode encerrados entre comillas dobles.
- Los caracteres de escape se prefijan con la barra invertida.
- Es similar a una cadena de caracteres en C o en Java.



- Similar a los números de C o Java.
- Puede representar: Integer, Real o Scientific.
- No usa formato octal o hexadecimal.
- No puede ser **NaN** o **Infinity**, en su lugar se usa **null**.

number



- Son legibles por el ser humano, aunque esa no es su principal finalidad.
- Tienen una sintaxis muy simple
- Ambos son jerárquicos en lo referente a la representación de la información.
- Son independientes del lenguaje de programación
- Son fáciles de crear y manipular.

- Su sintaxis es muy diferente
- JSON es mas compacto que XML lo que facilita su uso en entornos con recursos escasos (capacidad de procesamiento limitada, ancho de banda escaso ...).
- JSON permite incluir arrays.
- XML puede ser parseado mediante un modelo que define un conjunto de reglas, aunque JSON Schema ya está en el proceso avanzado.
- JSON se interpreta en el lado del cliente.

```
<menu id="file" value="File" >
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()"/>
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {
          "value": "New",
          "onclick": "CreateNewDoc()"
        },
        {
          "value": "Open",
          "onclick": "OpenDoc()"
        },
        {
          "value": "Close",
          "onclick": "CloseDoc()"
        }
      ]
    }
  }
}
```


- **JSON Schema is a JSON media type for defining the structure of JSON data.**
- JSON Schema is intended to define validation, documentation, hyperlink navigation, and interaction control of JSON data.
- This specification defines JSON Schema core terminology and mechanisms, including pointing to another JSON Schema by reference, dereferencing a JSON Schema reference, and specifying the vocabulary being used.

<https://json-schema.org/specification.html>

- Current versión: Draft-08 (<https://json-schema.org/specification.html>)
- Last draft in decembre 2020.
- Four documents:
 - JSON Schema Core: Basic foundations os JSON Schema.
<https://tools.ietf.org/html/draft-handrews-json-schema-02>
 - **JSON Schema Validation: validation keywords of JSON Schema.**
<https://tools.ietf.org/html/draft-handrews-json-schema-validation-02>
 - JSON Hyper-Schema: hyper-media keywords of JSON Schema.
<https://tools.ietf.org/html/draft-handrews-json-schema-hyperschema-02>
 - Relative JSON Pointer: extends the JSON Pointer syntax for relative pointers.
<https://tools.ietf.org/html/draft-handrews-relative-json-pointer-02>

JSON

(JavaScript Object Notation)