



# JSON, JQUERY y AJAX

-Adrian MIRANDA

-José SANDOVAL



# JSON

Historia

Principio de los 90's → PROBLEMA

XML

JSON → principios de los 2000 por Douglas Crockford (gurú de JavaScript), arquitecto senior de Yahoo JavaScript

Popularidad → Estándar





# ¿Qué es JSON?

JSON es un formato de datos basado en texto que sigue la sintaxis de objeto de JavaScript. (Ligero)

**JSON** es el acrónimo para *JavaScript Object Notation*, y aunque su nombre lo diga, no es necesariamente parte de JavaScript, sigue la sintaxis de objeto de JavaScript.

Estándar basado en texto plano para la organización de datos y el intercambio de información, por lo que se usa en muchos sistemas que requieren mostrar o enviar información para ser interpretada por otros sistemas,

La ventaja de JSON al ser un formato que es independiente de cualquier lenguaje de programación, es que los servicios que comparten información por éste método, no necesitan hablar el mismo idioma

# Lenguajes con soporte JSON

**ABAP:** EPO Connector.

**ActionScript:** ActionScript3 , JSONConnector.

**Ada:** GNATCOLL, JSON.

**AdvPL:** JSON-ADVPL.

**ASP:** JSON for ASP , JSON ASP utility class.

**AWK:** JSON , awk , rhawk.

**Bash:** Jshon , JSON , sh.

**BlitzMax:** bmx-rjson.

**C:** JSON\_checker , YAJL , jsOn , LibU , json-c , json-parser , jsonsl , WJElement , M's JSON parser , cJSON , Jansson , jsmn , cson , parson , ujson4c , nxjson , frozen , microjson.

**C++:** JSONKit , jsonme-- , ThorsSerializer , JsonBox , jvar , rapidjson , jsoncons , json , JSON Support in Qt , QJson , qmjson , jsoncpp , zoolib , JOST , CAJUN , libjson , nosjob , JSON++ , SuperEasyJSON , Casablanca , JSON library for IoT , minijson.

**C#:** fastJSON , JSON\_checker , Jayrock , Json , NET - LINQ to JSON , LitJSON , JSON for , NET , JsonFx , JSON@CodeTitans , How do I write my own parser? , JSONSharp , JsonExSerializer , fluent-json , Manatee Json , FastJsonParser

**Ciao:** Ciao JSON encoder and decoder.

**Clojure:** data , json.

**Cobol:** XML Thunder.

**ColdFusion:** SerializeJSON , toJSON.

**D:** Cashew , Libdjson

**Dart:** json library.

**Delphi:** Delphi Web Utils , JSON Delphi Library , tiny-json.

**E:** JSON in TermL.

**Erlang:** mochijson2.

**Fantom:** Json.

**FileMaker:** JSON.

**Fortran:** json-fortran , YAJL-Fort.

**Go:** package json.

**Groovy:** groovy-jo

**Haskell:** RJson package , json package

**Java:** JSON-java , JSONUtil , Jackson JSON Processor , jsonp , Json-lib , JSON Tools , Stringtree , SOJO , Jettison , json-taglib , XStream , Flexjson , JON tools , Argo , jsonij , fastjson , mjson , jjson , json-simple , json-io , JsonMarshaller , google-gson , Json-smart , FOSS Nova JSON , Com CONVERTER , Apache johnzon

**JavaScript:** JSON , json2 , js , clarinet , Oboe , js.

**LabVIEW:** i3-JSON , LAVA JSON ,

**Lisp:** Common Lisp JSON , Yason , Emacs Lisp.

**LiveCode:** mergJSON ,

**LotusScript:** JSON LS.

**LPC:** Grimoire: LPC JSON.

**Lua:** JSON Modules.

**M:** DataBallet.

**Matlab:** JSONlab , 20565 , 23393.

**Net.Data:** netdata-json.

**Objective C:** NSJSONSerialization , json-framework , JSONKit , yajl-objc , TouchJSON , ObjFW.

**OCaml:** Yojson , jsonm.

**OpenLaszlo:** JSON.

**PascalScript:** JsonParser.

**Perl:** CPAN , perl-JSON-SL.

**Photoshop:** JSON Photoshop Scripting

**PHP:** PHP 5 , 2 , json , Services\_JSON , Zend\_JSON , Comparison of php json libraries.

**PicoLisp:** picolisp-json.

**Pike:** Public , Parser , JSON , Public , Parser , JSON2.

**PL/SQL:** pljson: Librairie-JSON.

**PowerShell:** PowerShell.

**Prolog:** SWI-Prolog HTTP support.

**Puredata:** PuRestJson

**Python:** The Python Standard Library , simplejson , pyson , Yajl-Py , ultrajson , metamac , json

**R:** rjson , jsonlite.

**Racket:** json-parsing.

**Rebol:** json , r.

**RPG:** JSON Utilities

**Ruby:** json , yajl-ruby , json-stream , yajl-ffi.

**Scala:** package json.

**Scheme:** MZScheme , PLT Scheme.

**Squeak:** Squeak.

**Symbian:** s60-json-library.

**Tcl:** JSON.

**Visual Basic:** VB-JSON , PW , JSON

**Visual FoxPro:** fwJSON , JSON , vfpjson.



# Datos soportados

**Numeros** → 1; 1.5; -1

**String** → “comillas dobles”

**Boolean** → true; false

**Objetos** → {“nombre”: “Juan”}

**Null**

**Array** → []; [1,2]; [“hola”]; [{“nombre”:”Juan”}]



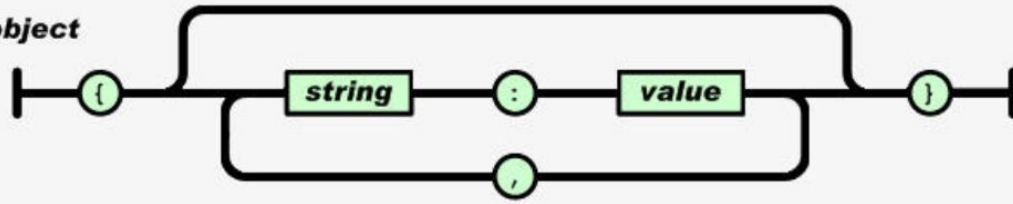
# Estructura

JSON puede representar **cuatro tipos primitivos**(cadenas, números, booleanos, valores nulos) y **dos tipos estructurados**(objetos y arreglos).

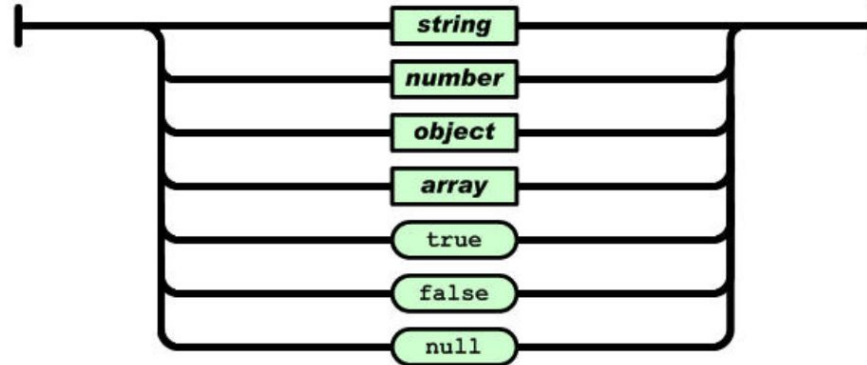
Hay dos elementos centrales en un objeto JSON: claves (Keys) y valores (Values).

- Las **Keys** deben ser cadenas de caracteres (strings). Como su nombre en español lo indica, estas contienen una secuencia de caracteres rodeados de comillas.
- Los **Values** son un tipo de datos JSON válido. Puede tener la forma de un arreglo (array), objeto, cadena (string), booleano, número o nulo.

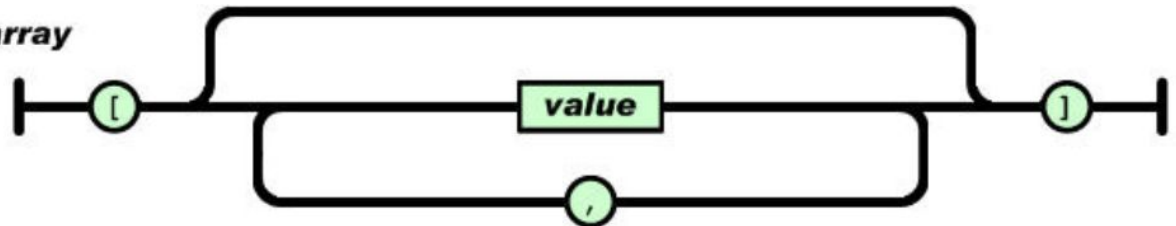
**object**



**value**



**array**



# JSON Y XML

Formato JSON	Formato XML
<pre>{   "nombre": "Juan",   "apellidos": "Suárez Iglesias",   "edad": 25,   "direccion": {     "calle_y_num": "Acacias, nº34",     "ciudad": "Valencia",     "codigo_postal": "43005",     "pais": "Colombia",   },   "telefono": [     {       "tipo": "fijo",       "numero": "0034966432134"     },     {       "tipo": "móvil",       "numero": "0034677493826"     }   ] }</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;nombre&gt;Juan&lt;/nombre&gt; &lt;apellidos&gt;Suárez Iglesias&lt;/apellidos&gt; &lt;edad&gt;25&lt;/edad&gt; &lt;direccion&gt;   &lt;calle_y_num&gt;Acacias, nº34&lt;/calle_y_num&gt;   &lt;ciudad&gt;Valencia&lt;/ciudad&gt;   &lt;codigo_postal&gt;43005&lt;/codigo_postal&gt;   &lt;pais&gt;Colombia&lt;/pais&gt; &lt;/direccion&gt; &lt;telefono&gt;   &lt;tipo&gt;fijo&lt;/tipo&gt;   &lt;numero&gt;0034966432134&lt;/numero&gt; &lt;/telefono&gt;   &lt;tipo&gt;móvil&lt;/tipo&gt;   &lt;numero&gt;0034677493826&lt;/numero&gt; &lt;/telefono&gt;</pre>





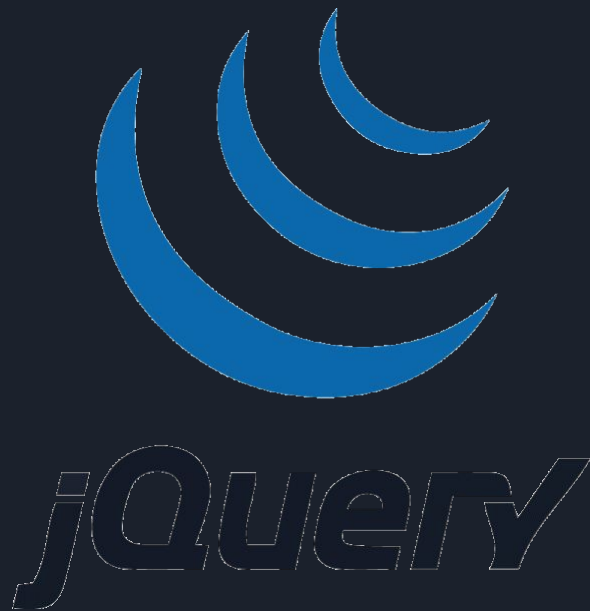
```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {
          "value": "New", "onclick": "CreateNewDoc()"
        }, {
          "value": "Open", "onclick": "OpenDoc()"
        }, {
          "value": "Close", "onclick": "CloseDoc()"
        }
      ]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

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



# JQUERY

Creada por John Resig (14 de enero de 2006)




```
var elementoPorId = document.getElementById('mi-id');
var elementoPorClase = document.getElementsByClassName('mi-clase')[0];
var elementoPorTag = document.getElementsByTagName('mi-tag')[0];
var elementoPorNombre = document.getElementsByName('mi-nombre')[0];
```

```
$('#mi-elemento').on('click', function() {
    // Esta función se ejecuta cuando
    // se hace click sobre mi-elemento
});
```

```
var elementoPorId = $('#mi-id');
var elementoPorClase = $('.mi-clase');
var elementoPorTag = $('mi-tag');
var elementoPorNombre = $('[name="mi-nombre"]');
```

```
$.ajax({
    url: "http://mi-api.com/endpoint",
    success: function (response) {
        if(response.status === "success") {
            // ¡Tengo datos!
        } else if(response.status === "error") {
            // Error
        }
    }
});
```



Es una biblioteca multiplataforma de JavaScript rápida y concisa que simplifica la manera de interactuar con el documento HTML, manejo de eventos, animación y las interacciones AJAX para el desarrollo web.

No es un lenguaje de programación sino una serie de funciones y métodos de Javascript.

Ofrece una serie de funcionalidades basadas en javascript que de otra manera requerirían de mucho mas código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempos y espacio.

Existen varios plugins (gratis y de pago) creados por desarrolladores de todo el mundo.

Cada plugin tiene un sitio web desde donde se pueden descargar sus archivos (demos, instrucciones, implementación, configuración, licencias) .

Luego está la manipulación **DOM (Modelo de Objetos del Documento)**, que se puede hacer por varios métodos. Pero para ponerlo en términos sencillos, te permite insertar y/o eliminar elementos DOM en una página HTML, así como agrupar líneas más fácilmente.



# Funcionalidades

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones `$()` o `jQuery()`.

```
<script type="text/javascript" src="jquery.js">
```

Ejemplo de una función

```
$(document).ready(function() {  
    //escribimos el código  
});
```

```
$("#tablaAlumnos"); // Devolverá el elemento con id="tablaAlumnos"  
$(".activo"); // Devolverá una matriz de elementos con class="activo"
```



# AJAX

AJAX es el acrónimo de Asynchronous Javascript and XML, JAVASCRIPT ASÍNCRONO Y XML

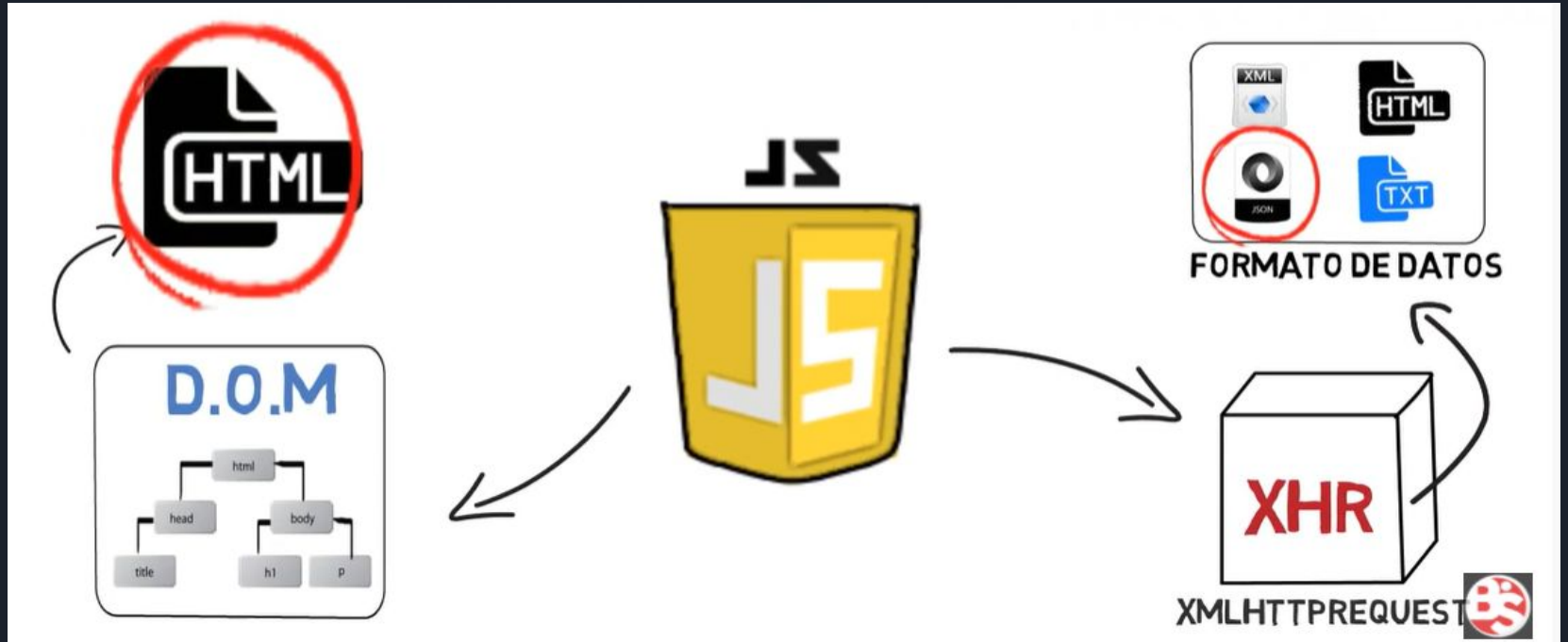
No es un lenguaje de programación.

Este acrónimo fue utilizado por primera vez por Jesse James Garret en 2005, en su publicación Ajax: a New Approach to Web Applications

conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de forma asíncrona. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

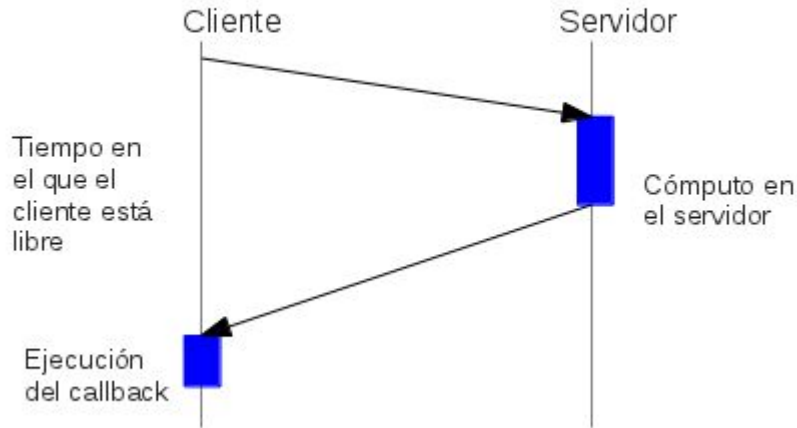


# Combinación de tecnologías

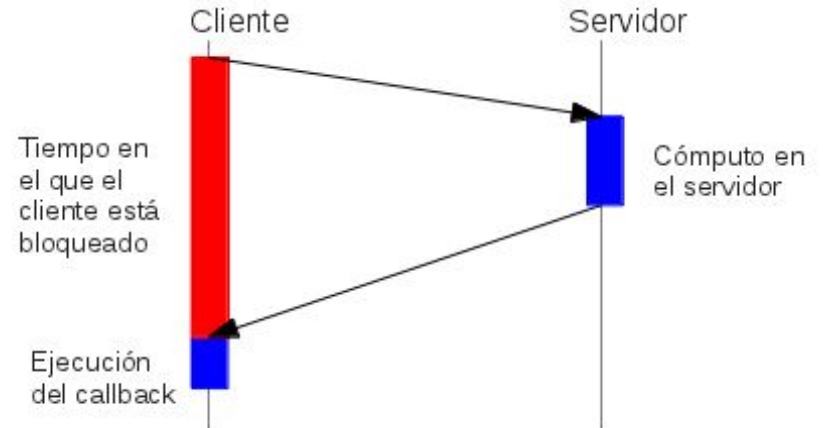


# Diferencia entre peticiones de cliente

## Petición Asíncrona

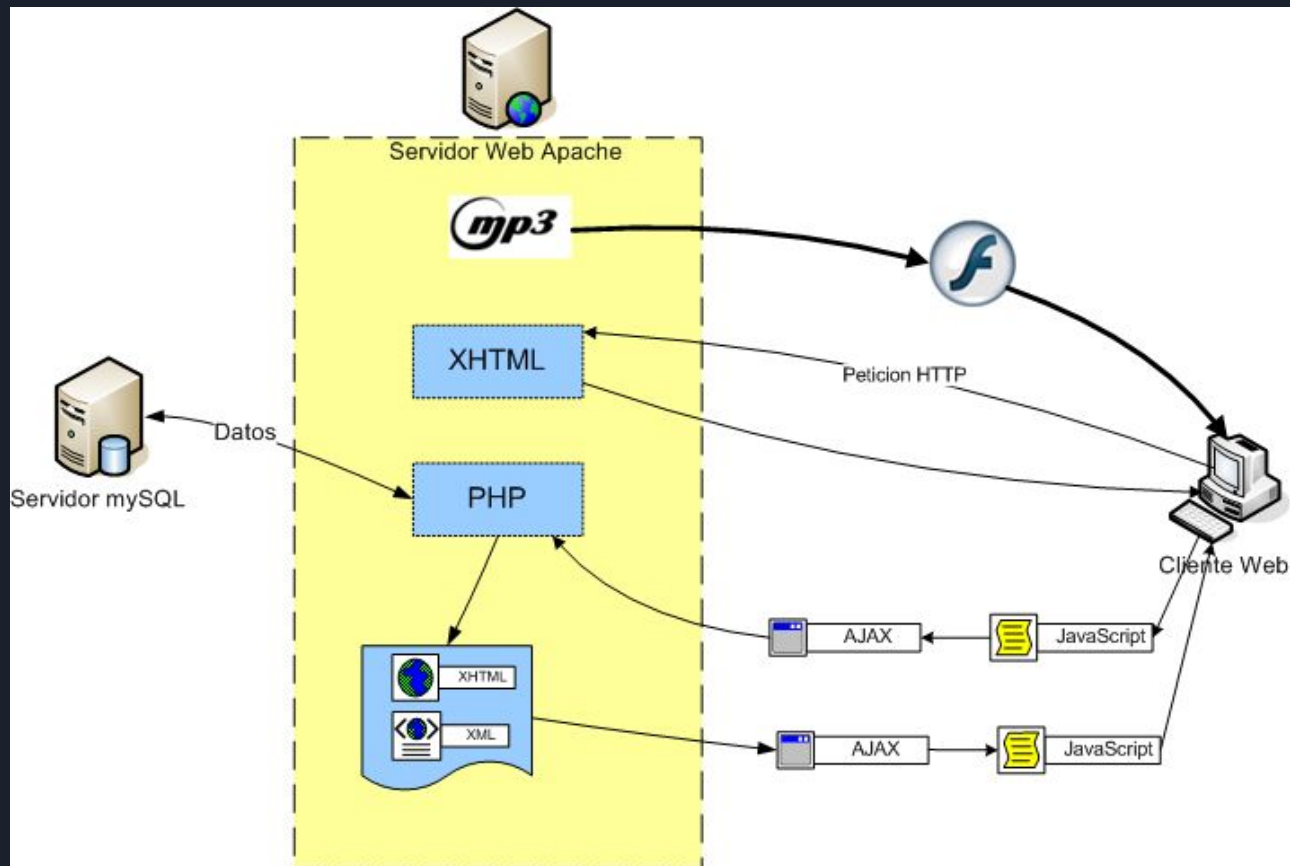


## Petición Síncrona





# Velocidad, interactividad y usabilidad





# Desventajas

- Avance y retroceso del navegador
- El desarrollo de aplicaciones web se puede volver más complejo.
- Existen problemas y restricciones de seguridad relacionados con el uso de Ajax. Hay que tener en cuenta que por motivos de seguridad no todos los procesos se pueden realizar del lado del cliente (que por su propia naturaleza es “manipulable”).
- La indexación para los motores de búsqueda se ve dificultada, con lo cual nuestros sitios web pueden perder visibilidad en los buscadores. No es lo mismo un contenido “constante” o aproximadamente estático, fácilmente rastreable para un buscador, que un contenido “cambiante” en función de la ejecución de JavaScript, difícilmente rastreable para un buscador.



*EJEMPLO APLICATIVO* →