JSON – Estructura de Datos

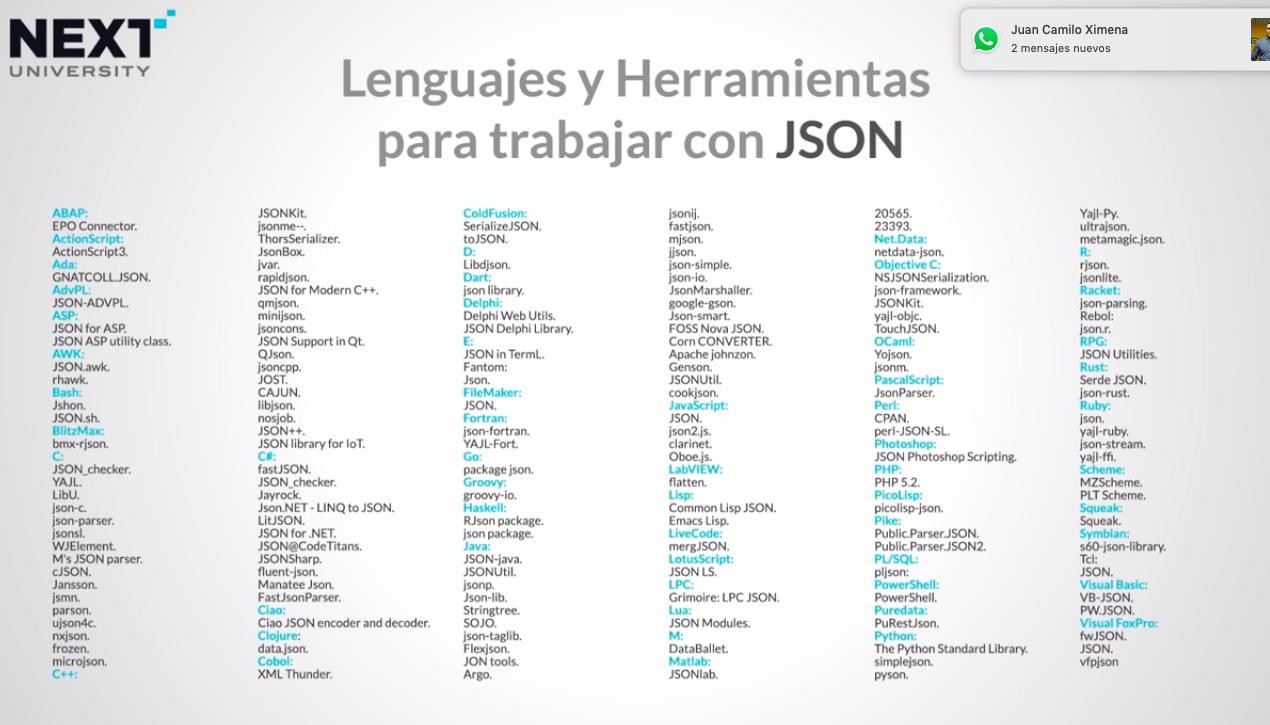
JavasScript Objet Notation.

Derivado de objetos literales de JS, usado para transporte y uso de datos.

Servicios WEB ECMAScript3 estructuras mas usadas.

Se define como estructura {ker, value}

Se define como vector.



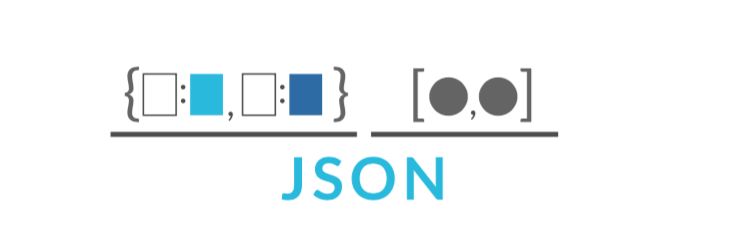
# XML vs JSON

Durante mucho tiempo XML fue la única opción que existía para compartir datos entre servidores y sistemas. XML era la solución a todos los problemas de intercambio de datos.   
Uno de los mayores problemas de XML es que es muy poco legible para el usuario, ya que una estructura XML puede almacenar cualquier tipo de dato incluyendo imágenes, audio, video y mucho más, formando un código demasiado extenso. Esto puede ser peligroso a nivel de seguridad ya que se podría incluir código malicioso dentro de la estructura XML. Otra desventaja de los archivos XML es que son difíciles de tratar, además de que la sintaxis entre marcas, algo parecido al HTML, hace que sean muy pesados cuando tienen grandes cantidades de información.  
  
JSON es una estructura que está limitada a almacenar datos clásicos y comunes, como números y textos, permitiendo que sea un formato ligero ideal para el transporte de datos. En la actualidad es el formato estándar usado para la comunicación de datos, superando a XML.

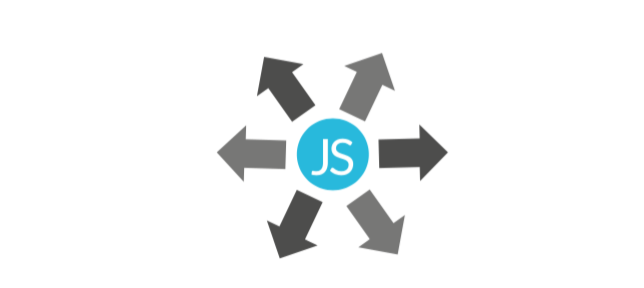
Ejemplo de la misma estructura en Formato XML y en JSON:



# Ventajas de usa**r JSON**



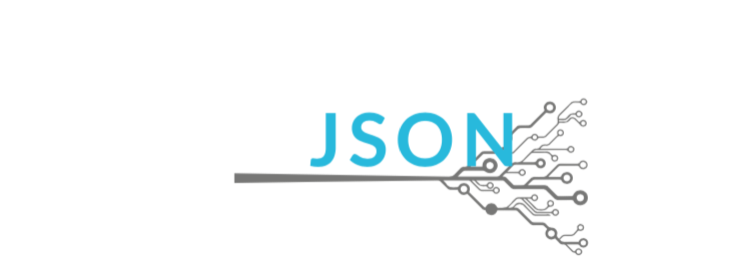
* JSON soporta dos tipos de estructuras: un set de pares llave-valor contenidos por un objeto y la otra es un array de valores.



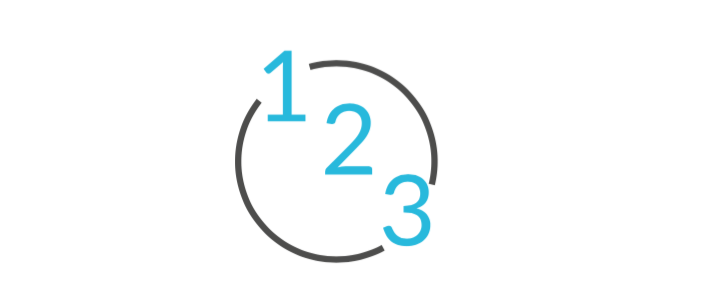
* JSON es una estructura totalmente independiente, no necesita de tecnologías alternas.



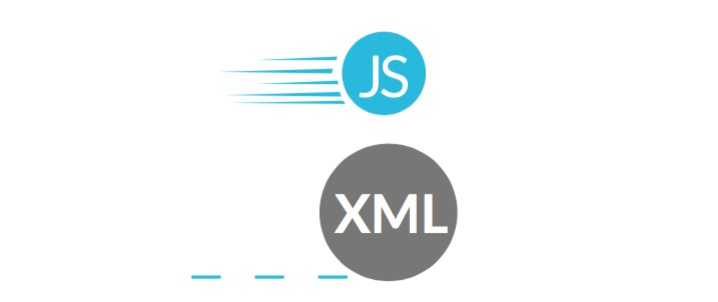
* JSON es totalmente flexible y para extenderlo solo hace falta agregar una nueva propiedad con los valores deseados.



 JSON permite definir estructuras de datos muy complejas.



 JSON es un formato muy intuitivo y fácil de comprender.

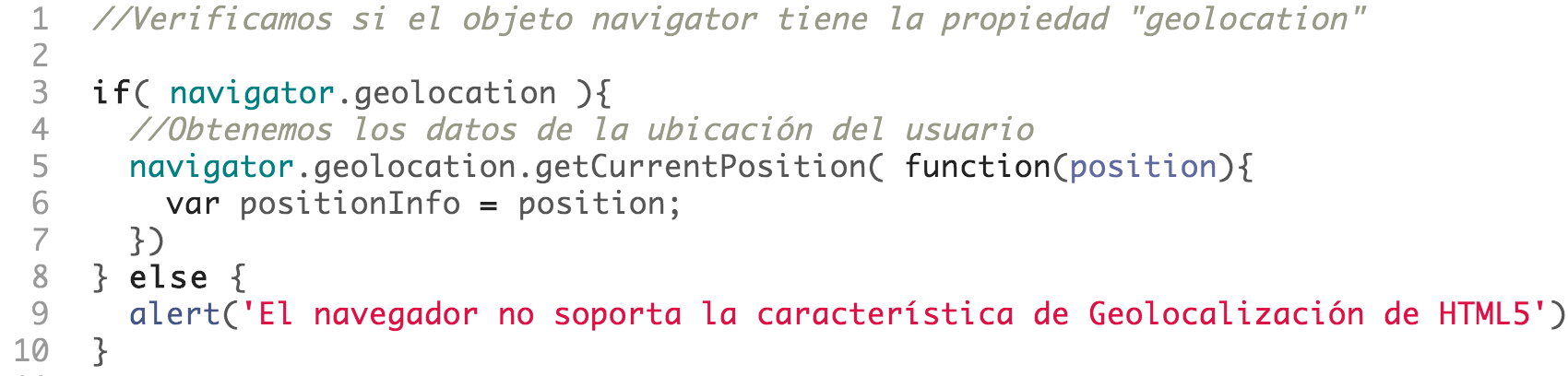


 JSON Tiene una alta velocidad de procesamiento y un menor tamaño con respecto a XML.

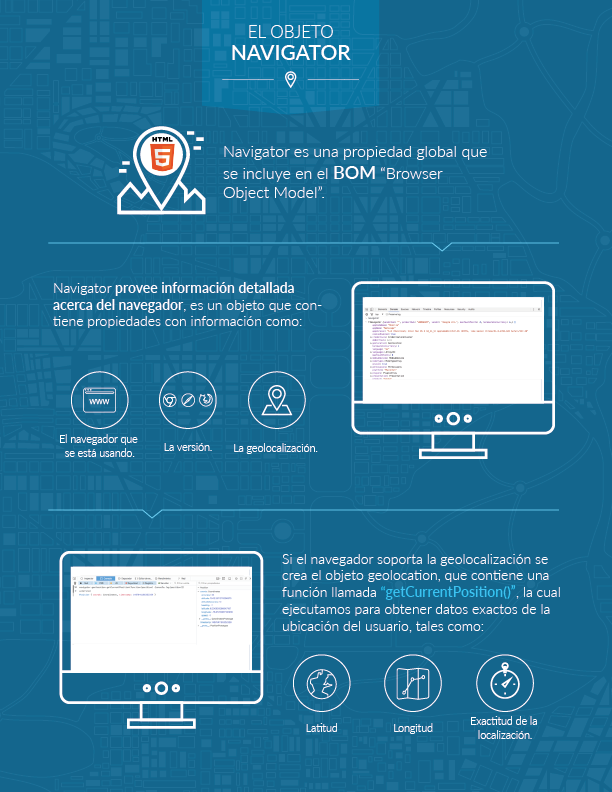
En conclusión, **JSON** es usado con frecuencia en ambientes web en los que fluye mucha información entre el cliente y el servidor; y también donde los tiempos de respuesta son vitales para el rendimiento.

# Geolocalización en**HTML5**

Entre muchas de las nuevas funcionalidades que se incluyeron en la especificación 5 de HTML, la geolocalización fue, sin duda, una de las mejores.  
La geolocalización fue creada para obtener facilmente la ubicación de los usuarios y ofrecer una mejor experiencia, además de proveer información acerca de la zona en la que se encuen- tran, lo cual resulta muy útil para aplicaciones que prestan servicios en esta área.  
En el caso de la Web HTML5 utiliza el navegador para acceder a la ubicación a través de la IP o la Red WIFI, en el caso de los móviles se utiliza el GPS nativo del celular.  
La Geolocalización está disponible a través del objeto Navigator, el cual estudiaremos detalladamente en el siguiente tema, este objeto se encarga de verificar si el navegador desde el cual se está accediendo a la web tiene soporte para la caracteristica de geolocalización deHTML, de ser así, el objeto navigator crea una propiedad llamada **geolocation** y es allí donde encontramos información detallada acerca de la ubicación del usuario.  
En el siguiente cuadro podemos ver cómo verificar si el navegador soporta la geolocalización y la forma de acceder a la información del usuario, si no lo soporta, lanzamos una alerta indican- do que no se pudo obtener la información.



# El objeto Navegador



Notas actividades:

map.setZoom(15)

Sirve para aumentar un poco el zoom de nuestro mapa.

infoWindow.setPosition(pos)

Sirve para que la ventana de información se sitúe en la posición del usuario.

infoWindow.setContent(‘Ubicacion Encontrada’)

Sirve para agregar el título de ubicación encontrado.

map.setCenter(pos)

Sirve para que el mapa se ubique en las posiciones que capturamos del usuario.

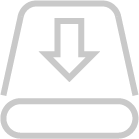
# Almacenamiento Local en **HTML5**



Continuamos explorando las nuevas funcionalidades de HTML5 y sin duda otra de las mejores herramientas que se implementaron en esta versión Web Storage o Almacenamiento local. Esta herramienta guarda datos del usuario o de la aplicación directamente en la maquina del usuario y no en un servidor de bases de datos. Así, su funcionalidad facilita la vida de los desarrolladores a la hora de manipular datos que tienen que persistir constantemente. Para hacer uso de esta caracteristica solo es necesario Javascript, por lo que es otra ventaja grande frente a otros métodos de almacenamiento. Además, con esta funcionalidad podemos guardar hasta aproximadamente 5MB de datos en el disco duro del usuario. Existe otra forma de guardar datos localmente que se conoce como cookies pero estas tienen graves problemas de seguridad y solo soportan unos 4KB. Como en todas las nuevas características el factor Compatibilidad es esencial y debe ser tenido en cuenta a la hora de implementar estas funciones. La página de caniuse nos puede ayudar con eso.

## Tipos de ****Almacenamiento Local****

Existen dos tipos de almacenamiento local, LocalStorage & SessionStorage. Ambos son objetos globales asignados a window, por lo tanto también se puede acceder a ellos usando window.localStorage o window.sessionStorage. Estos objetos están en capacidad de almacenar datos llave-valor (key/value) y la única diferencia que existe entre ellos es que sessionStorage almacena los datos mientras el usuario se encuentre activo y no salga del navegador; una vez cerrado los datos se pierden.



Por el contrario los datos almacenados en localStorage permanecen en el tiempo y el usuario puede cerrar el navegador sin ningun problema. La única manera de eliminarlos es borrar la caché del navegador.   
Para usar datos con el almacenamiento local lo indicado es usar formatos JSON. Por tanto usaremos el objeto nativo JSON para serializar y deserializar este tipo de formatos, empleando las funciones **"JSON.stringify"** y **"JSON.parse"**. LocalStorage y sessionStorage ofrecen las siguientes funciones para la manipulación de los datos:





## Que son los Web Workers

Modelo de emisión de eventos.

Var worker = new worker(“taeas.js”)

// Iniciar la ejecución

Worker.postMessage()

//Escuchar los eventos

Worker.addEventListener(“message”, function(e) {

Console.log(“El worker envío:· + e.data)

}, false)

//Enviar mensajes

Worker.postMessage(“Hola Worker”)

## CONLUSIÓN

**JavaScript** es el lenguaje de programación más popular y usado entre los desarrolladores de todo el mundo, según los últimos reportes de **GitHub**, y se encuentra en continua evolución: librerías y frameworks creados para la Web son puestos a disposición de los usuarios con frecuencia.

**Hasta los años 2004 y 2005** aproximadamente, **Javascript era un simple lenguaje de scripting** que se utilizaba para hacer más dinámicas las páginas web y agregarles interactividad asociada a eventos, como pulsar un botón o hacer clic en algún link para realizar una acción. Hoy en día JavaScript está en todas partes y, además de ser el lenguaje por defecto para el desarrollo de código del lado del cliente en páginas y aplicaciones web, también se encuentra presente en sistemas operativos, bases de datos, desarrollo para móviles y servidores. Sin duda alguna, saber JavaScript te abrirá muchísimas puertas como desarrollador.

**Hoy en día JavaScript está en todas partes** y, además de ser el lenguaje por defecto para el desarrollo de código de páginas y aplicaciones web, también es usado en los sistemas operativos, las bases de datos y el desarrollo para móviles y servidores.

**Sin duda alguna, saber JavaScript te abrirá muchísimas puertas como desarrollador.**

Gracias a este curso y lo que con esfuerzo y dedicación practicaste con los ejercicios, estás en capacidad de:

1. **Usar las herramientas de los navegadores.**

Gracias a los videos conociste los navegadores más completos para desarrollar en la Web, punto principal y primordial para luego, con el conocimiento previo, usar herramientas como: **la consola, la depuración del código y los puntos de debug**para el análisis de errores. Estas herramientas son fundamentales para empezar a construir páginas y aplicaciones web con Javascript.

1. **Construir algoritmos complejos y solucionar problemas a través de la programación, usando JavaScript.**

Descubriste las estructuras de datos, **las estructuras de control condicionales y cíclicas** e identificaste cómo implementarlas para construir algoritmos capaces de resolver problemas y construir lógicas complejas, acordes a la necesidad de cada proyecto. Probaste el funcionamiento de las variables y de las operaciones matemáticas y, además, tomaste decisiones con base a las entradas del usuario.

1. **Implementar las interacciones del DOM.**

Identificaste las interacciones del DOM, usando JavaScript. Descubriste conceptos como: **selectores, modificaciones de contenidos, estilos y estados de los elementos**, además, aprendiste a detectar los principales eventos que puede desencadenar un usuario en una página o aplicación web, como: un clic, una tecla, hacer scroll, entre otros.

1. **Usar las API’s de HTML5.**

Conociste las API’s más relevantes de la versión 5 de HTML, por las cuales es conocida como la versión más importante. Descubriste el formato **JSON** para el intercambio y manejo de datos. Aprendiste cómo es posible implementar geolocalización para ubicar a los usuarios de una manera fácil y **Web Workers** para ejecutar procesos en paralelo en JavaScript y resolver tareas complejas, algo que no era posible en versiones anteriores.