

XML, JSON y el intercambio de información

XML, JSON and the information exchange

Frank Pompa Sourd

Universidad Agraria de la Habana: Fructuoso
Rodríguez Pérez, Cuba. Carretera Tapaste, km
23 ½, San José de las Lajas, Mayabeque

Autores para correspondencia: mabel.rdguez@infomed.sld.cu

Resumen

La comunicación es un proceso de intercambio de información y en el mundo de la informática este proceso es vital para la relación hombre - máquina, para el funcionamiento de los programas y sistemas informáticos dentro de un mismo ordenador, así como, para el funcionamiento de una red de computadoras. Un ejemplo lo constituye Internet que se ha convertido en un gran medio de comunicación. Para propiciar este intercambio de información, en el ámbito digital, se han creado diferentes lenguajes y estándares computacionales. En el siguiente trabajo se realiza una comparación entre el lenguaje XML y el estándar JSON, debido a que constituyen dos de las opciones más utilizadas para el intercambio de información entre sistemas y aplicaciones en la actualidad.

Palabras claves: XML, JSON, intercambio de información.

Abstract

Communication is a process of information exchange and in the computer world, this process is vital for the human-machine relationship, for the operation of computer programs and systems in the same computer, as well as for the operation of a Computer network. An example is the Internet that has become a great means of communication. In order to enjoy the digital world information flow, to facilitate this exchange of information, in the computer field, different languages and computational standard have been created. In the following work, a comparison is made between the XML language and the JSON standard, because they constitute two of the most used options for the exchange of information between systems and applications today.

Keywords: XML, JSON, exchange of information.

Recibido: 6 de septiembre de 2019

Aprobado: 18 marzo de 2020

Introducción

Desde los albores de la humanidad el hombre se ha enfrentado al problema del intercambio de información para lograr, por un lado, la no tan simple comunicación entre individuos y por otro transmitir el cúmulo de información a ser guardado. El desarrollo de las formas primitivas de comunicación humana trajo como resultado el surgimiento del lenguaje oral, producto de la necesidad del intercambio de las experiencias adquiridas, fundamentalmente, en las relaciones de trabajo. Sin embargo, el hombre primitivo sintió también preocupación por representar sus pensamientos de forma escrita lo que conllevó a la aparición de los jeroglíficos, signos gráficos destinados a evocar ideas. Este sistema ideográfico se transformó posteriormente en escritura silábica y aproximadamente en el año 895 antes de nuestra era, los fenicios, inventaron el alfabeto (Gener y Enrique, 2009). La experiencia acumulada, debía ser legada, transferida, intercambiada, ya fuese de padre a hijo, de maestro a aprendiz, o en general de una generación a otra. Para lograr este objetivo, el hombre, se ha valido, a lo largo de la historia, de diferentes medios y herramientas, naturales o inventadas que le permitieron realizar los primeros dibujos o la representación de sílabas y letras, asentándolos inicialmente en piedras o cavernas. Más tarde, el papiro resultó ser el soporte de sus representaciones ideales, medio utilizado aún después de la invención del papel. Otros medios, ingeniosos y sorprendentes, se han utilizado para transmitir información, desde las señales de humo y quipus, hasta las modernas y sofisticadas transmisiones vía satélites, sin olvidar creaciones tan geniales como el código Morse y el sistema Braille.

El mundo de la informática no ha estado ajeno a esta dificultad ya que el intercambio de información siempre ha sido un serio problema cuando se utilizan lenguajes y sistemas operativos diversos e incompatibles, lo que se agudizó con el surgimiento de Internet.

Desarrollo

Introduction

Since the dawn of humanity, man has faced the problem of information exchange to achieve, on the one hand, not so simple communication between individuals and, on the other, transmit the accumulation of treasured information. The development of primitive forms of human communication resulted in the emergence of oral language, a product of the need to exchange the experiences acquired, fundamentally, in work relationships. However, primitive man was also concerned about representing his thoughts in writing, which led to the appearance of hieroglyphs, graphic signs designed to evoke ideas. This ideographic system was later transformed into syllabic writing and in approximately 895 BC, the Phoenicians invented the alphabet (Gener and Enrique, 2009). The accumulated experience had to be bequeathed, transferred, exchanged, and were from father to son, from teacher to apprentice, or generally from one generation to another. To achieve this objective, man has used, throughout history, different means and tools, natural or invented, that allowed him to make the first drawings or the representation of syllables and letters, settling them before in stones or caves. Later, papyrus turned out to be the support for his ideal representations, a medium used even after the invention of paper. Other ingenious and elegant media have been used to transmit information, from smoke signals and quipus, to modern and sophisticated satellite transmissions, not to mention such great creations as Morse code and Braille.

The world of computing has not been immune to this difficulty since the exchange of information has always been a serious problem when using diverse and incompatible languages and operating systems, which became more acute with the emergence of the Internet.

Development

At the end of the 60s of the last century, specialists from the IBM company, faced with the need to store large volumes of information and documentation on

A finales de la década del 60 del pasado siglo especialistas de la empresa IBM, ante la necesidad de almacenar grandes volúmenes de información y documentación de diferentes temas, además del hecho, de que muchos sistemas de la empresa no podían comunicarse entre sí debido al distinto sistema de descripción que utilizaban, crearon el lenguaje de marcado generalizado GML, por sus siglas en inglés, con el cual pretendían clasificar todo tipo de información y escribir cualquier documento para su posterior procesamiento. También, pretendían, que permitiese entenderse con los diferentes documentos generados por distintos sistemas y plataformas, reuniendo en una misma etiqueta el formato y la descripción del contenido (Montero, 2001).

GML fue diseñado para componer estructuras de datos descriptivas, es decir, estructuras de datos que describen a otras estructuras de datos, o sea, es un meta-lenguaje. En 1986 la Organización Internacional de Estándares (ISO por sus siglas en inglés), normalizó dicho lenguaje y creó el lenguaje de marcado generalizado estándar SGML, por sus siglas en inglés, que fue adoptado como estándar internacional para el intercambio y almacenaje de información, ISO 8879. SGML es un metalenguaje que permite mantener repositorios de documentación estructurada en formato electrónico. SGML es un lenguaje poderoso y extremadamente versátil, pero esas facilidades justamente lo hacen difícil de utilizar. Entrenar personas para usar documentos SGML y crear aplicaciones para leer estos documentos requiere gran cantidad de tiempo y energía. (Sturm, 2000)

En 1989, el investigador de la Organización Europea para la Investigación Nuclear (CERN por sus siglas en francés) Timothy John Berners-Lee, crea el lenguaje de marcación de hipertexto (HTML por sus siglas en inglés) que fue utilizado en el servicio de Internet, World Wide Web (Pompa, 2006). Este lenguaje fue diseñado para estructurar textos y presentarlos en forma de hipertexto, o sea, como un documento digital que se puede leer de manera no secuencial, que es el formato estándar de las páginas Web. Contrariamente a otros lenguajes de programación, el HTML, definido en términos del SGML, utiliza etiquetas, o sea, breves instrucciones

different topics, in addition to the fact that many systems of the company could not communicate with each other due to the different The description system they used, created the generalized markup language GML, for its acronym in English, with which they intended to classify all kinds of information and write any document for further processing. They also intended that it allow understanding with the different documents generated by different systems and platforms, bringing together the format and the content description in the same label (Montero, 2001).

GML was designed to compose descriptive data structures, that is, data structures that describe other data structures, that is, it is a meta-language. In 1986 the International Standards Organization (ISO), standardized this language and created the generalized markup language SGML standard, for its acronym in English, which was adopted as an international standard for the exchange and storage of information, ISO 8879. SGML is a metalanguage that allows to maintain structured documentation repositories in electronic format. SGML is a powerful and extremely versatile language, but those facilities just make it difficult to use. Training people to use SGML documents and creating applications to read these documents requires a great deal of time and energy. (Sturm, 2000)

In 1989, the European Organization for Nuclear Research (CERN) researcher Timothy John Berners-Lee, created the hypertext markup language (HTML) that was used in the Internet service, World Wide Web (Pompa, 2006). This language was designed to structure texts and present them in the form of hypertext, that is, as a digital document that can be read in a non-sequential manner, which is the standard format for Web pages. Contrary to other programming languages, HTML, defined in terms of SGML, uses tags, that is, short instructions or directives, by means of which the way of representing the text, as well as the images and other elements, is

o directivas, mediante las cuales se determina la forma de representar el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador mediante el uso de un navegador (Montero, 2001).

El desarrollo de Internet propició que HTML se convirtiera en uno de los formatos más populares para la construcción de documentos y fue adoptado rápidamente por la comunidad. En 1994 al navegador web Netscape le extendieron el HTML, o sea, le agregaron un juego de etiquetas que no funcionaban correctamente en ningún otro navegador, este fue el comienzo de lo que se llamó "la guerra de los navegadores" (Sturm, 2000) ya que varias organizaciones hicieron lo mismo con los navegadores que desarrollaban, en aras de lograr el visualizador más versátil y por ende el más exitoso comercialmente, así cada visualizador web daba formato al documento según su criterio, (Montero, 2001; Pompa y Rodríguez, 2006) pero esto provocó que algunas de esas extensiones no fueran mostradas adecuadamente por otros visores web lo cual atentaba contra la posibilidad de obtener un estándar del HTML.

Desde el 1996 una entidad llamada World Wide Web Consortium, [www.w3c.org] abreviadamente W3C, organización creada y dirigida por Berners-Lee, que produce estándares para la World Wide Web, trató de poner orden en el HTML y establecer sus reglas y etiquetas para lograr convertirlo en un estándar. Entre 1997 y 1999, el W3C estableció varias recomendaciones, equivalente a un estándar, para el HTML, hasta llegar a la versión 4.01 [www.w3.org/TR/html4/] en diciembre de 1999 (Pompa y Rodríguez, 2006).

Sin embargo, y a pesar de su uso y desarrollo, el HTML no resolvió todos los problemas y expectativas que planteaba la sociedad global de Internet. Entre los puntos en contra del mismo estaba que define más la presentación que el contenido y tiene un fuerte vínculo entre contenido y forma. Por otra parte su interpretación es ambigua, ya que obedece al navegador web utilizado lo que provoca que la presentación del documento en pantalla dependa de dicho navegador. Estas carencias del HTML, respecto al tratamiento de la

determined in the computer screen by using a browser (Montero, 2001).

The development of the Internet led to HTML becoming one of the most popular formats for building documents and it was quickly adopted by the community. In 1994, they extended HTML to the Netscape web browser, that is, they added a set of tags that did not work correctly in any other browser, this was the beginning of what was called "the browser war" (Sturm, 2000). that several organizations did the same with the browsers they developed, in order to achieve the most versatile viewer and therefore the most commercially successful, thus each web viewer formatted the document according to their criteria (Montero, 2001; Pompa and Rodríguez, 2006) but this caused that some of these extensions were not shown properly by other web viewers which undermined the possibility of obtaining an HTML standard.

Since 1996 an entity called the World Wide Web Consortium, [www.w3c.org] for short W3C, an organization created and directed by Berners-Lee, which produces standards for the World Wide Web, tried to put order in HTML and establish its rules and labels to make it a standard. Between 1997 and 1999, the W3C established several recommendations, equivalent to a standard, for HTML, until reaching version 4.01 [www.w3.org/TR/html4/] in December 1999 (Pompa and Rodríguez, 2006).

However, despite its use and development, HTML did not solve all the problems and expectations posed by the global Internet society. Among the points against it was that it defines the presentation more than the content and has a strong link between content and form. On the other hand, its interpretation is ambiguous, since it obeys the web browser used, which causes the presentation of the document on the screen to depend on said browser. These shortcomings of HTML, with respect to the treatment of information, together with the inconveniences of SGML implementation, due to its complexity, to a

información, unidas a los inconvenientes de implementación del SGML, debido a su complejidad, en cierta medida provocaron que el W3C creara, en 1996, un grupo de trabajo, llamado "XML WorkingGroup", originalmente conocido como "Web SGML", liderado por Jon Bosak, que obtuvo, como resultado, la primera versión del lenguaje de marcado extensible XML, por sus siglas en inglés, publicada en 1998 (Sturm, 2000; Montero, 2001).

En aquel tiempo XML constituyó un nuevo estándar, con una funcionalidad similar a la del SGML aunque más simple de usar que este pero con una estructura más rígida que HTML y adaptado a Internet. XML no era una versión mejorada de HTML, ni un lenguaje para hacer mejores páginas Web. Es un estándar internacionalmente reconocido, que no pertenece a ninguna compañía y su uso es libre [www.w3.org/TR/REC-xml].

Un documento XML, al igual que el HTML, se basa en documentos de texto llano en los que se utilizan etiquetas para delimitar los elementos, la idea es que las etiquetas "describan" la información y no como presentarla. Una ventaja de XML es su extensibilidad, o sea, que las etiquetas a usar son definidas por el usuario, ya que no existe un juego de etiquetas predefinido como en HTML y esto permite almacenar datos estructurados de una manera muy sencilla. En otras palabras, XML define las etiquetas en función del tipo de datos que está describiendo y no de la apariencia final que tendrán en pantalla o en la copia impresa. Este lenguaje separa completamente la información o dato de su forma de presentación o formato, o sea, separa el contenido de la forma. Esto, además, facilita el análisis sintáctico y la búsqueda y extracción de información. También es independiente del Sistema Operativo y de la aplicación que lo utilice y puede ser utilizado en cualquier lenguaje o alfabeto. Otro de los objetivos del XML era usar e intercambiar información sin estar atado a los motores de bases de datos relacionales de la época. El mismo se convirtió en un estándar aceptado por la industria pero con su desarrollo posterior fueron apareciendo varias herramientas que, por una parte lo complementaban, pero por otra complicaron el trabajo con este; como ejemplo se pueden mencionar entre otras:

certain extent caused the W3C to create, in 1996, a working group, called "XML WorkingGroup", originally known as "Web SGML", led by Jon Bosak, who obtained, as a result, the first version of the extensible markup language XML, for its acronym in English, published in 1998 (Sturm, 2000; Montero, 2001).

At that time XML constituted a new standard, with a functionality similar to that of SGML, although simpler to use than this, but with a more rigid structure than HTML and adapted to the Internet. XML was not an improved version of HTML, nor was it a language for making better web pages. It is an internationally recognized standard, which does not belong to any company and its use is free [www.w3.org/TR/REC-xml].

An XML document, like HTML, is based on plain text documents in which tags are used to delimit the elements, the idea is that the tags "describe" the information and not how to present it. An advantage of XML is its extensibility, that is, the tags to be used are defined by the user, since there is no predefined set of tags as in HTML and this allows storing structured data in a very simple way. In other words, XML defines the tags based on the type of data that you are describing and not on the final appearance that they will have on screen or in print. This language completely separates the information or data from its form of presentation or format, that is, it separates the content from the form. This, in addition, facilitates the syntactic analysis and the search and extraction of information. It is also independent of the Operating System and the application that uses it and can be used in any language or alphabet. Another objective of XML was to use and exchange information without being tied to the relational database engines of the time. It became an accepted standard by the industry but with its subsequent development, several tools appeared that, on the one hand, complemented it, but on the other, they complicated the work with it; as an example they can be mentioned among others:

- lenguaje extensible de hojas de estilo XSL (por sus siglas en inglés)
- XSLT o transformaciones XSL
- XPath
- Schema
- XPointer

Resumiendo, el mundo de XML se enriqueció pero a la vez se complejizó y esto supuso un costo elevado asociado a la curva de aprendizaje necesaria para poder trabajar con el mismo. Estos inconvenientes motivaron la búsqueda de nuevas alternativas para el intercambio de información.

JSON fue definido por el desarrollador de software Douglas Crockford a finales de 2002. Es un formato basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos, o sea, soporta dos tipos de estructuras, una de ellas los objetos que contienen una colección de pares llave-valor y el otro arreglos de valores. Esto proporciona gran sencillez en las estructuras. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de dichos archivos y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XML (Aplicaciones Web ServInf UA, 2008).

JSON es un formato ligero, pensado para el intercambio de datos, que surge como alternativa a XML en AJAX, acrónimo de Asynchronous JavaScript And XML, una técnica de desarrollo web para crear aplicaciones interactivas. JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia, cuando la fuente de datos es explícitamente de confianza y donde no es importante el no disponer de procesamiento XSLT para manipular los datos en el cliente (Peñas y Reyero, 2012). En poco tiempo el uso del mismo se extendió y aparecieron implementaciones para la mayoría de los lenguajes de programación. El motivo de su éxito: su sencillez y facilidad de uso ya que es más simple y de fácil implementación que el XML.

Ejemplo de dato en formato XML y su equivalente en JSON:

- Extensible XSL style sheet language (for its acronym in English)
- XSLT or XSL transformations
- XPath
- Schema
- Xpointer

In short, the world of XML was enriched but at the same time more complex and this entailed a high cost associated with the learning curve necessary to be able to work with it. These inconveniences motivated the search for new alternatives for the exchange of information.

JSON was defined by the software developer Douglas Crockford at the end of 2002. It is a format based on a subset of the JavaScript syntax: literals of arrays and objects, that is, it supports two types of structures, one of them the objects that contain one collection of key-value pairs and the other arrays of values. This provides great simplicity in the structures. As it uses JavaScript syntax, JSON definitions can be included within such files and accessed without any additional parsing as required with XML-based languages (ServInf UA Web Applications, 2008).

JSON is a lightweight format, designed for data exchange, which arises as an alternative to XML in AJAX, an acronym for Asynchronous JavaScript And XML, a web development technique for creating interactive applications. JSON is commonly used in environments where the size of the data flow between client and server is of vital importance, when the data source is explicitly trusted and where it is not important not to have XSLT processing to manipulate the data on the client (Peñas and Reyero, 2012). In a short time the use of it spread and implementations appeared for most of the programming languages. The reason for its success: its simplicity and ease of use since it is simpler and easier to implement than XML.

Example of data in XML format and its equivalent in JSON:

XML	JSON
<pre> <?xmlversion="1.0" encoding="UTF-8" ?> <nombre>Juan</nombre> <apellidos>Pérez Iglesias</apellidos> <edad>25</edad> <direccion> <calle_y_num>Campanario, nº34</calle_y_num> <ciudad>La Habana</ciudad> <codigo_postal>4400</codigo_postal> <pais>Cuba</pais> </direccion> <telefono> <tipo>fijo</tipo> <numero>79654321</numero> </telefono> <telefono> <tipo>móvil</tipo> <numero>5467493826</numero> </telefono> </pre>	<pre> { "nombre": "Juan", "apellidos": "Pérez Iglesias", "edad": 25, "direccion": { "calley_num": "Campanario, nº34", "ciudad": "La Habana", "codigo_postal": "4400", "pais": "Cuba", }, "telefono": [{ "tipo": "fijo", "numero": "79654321" }, { "tipo": "móvil", "numero": "5467493826" }] } </pre>

En el ejemplo se puede apreciar que JSON se caracteriza por reducir el volumen de datos y por ello el tamaño de los archivos necesarios para transmitir. Por esto JSON comenzó a ser muy utilizado hasta convertirse en un estándar, lo que no significa que XML haya dejado de usarse.

JSON es fácil de leer y comprender por un humano. El XML es fácilmente procesable tanto por humanos como por software; en este último caso, el análisis sintáctico es factible debido a las estrictas reglas que rigen la composición del documento.

La depuración y corrección de errores se facilita con XML, ya que es relativamente simple verificar que los datos enviados están bien formados y son válidos utilizando Schema, un lenguaje que permite describir la estructura y las restricciones de los contenidos de

In the example you can see that JSON is characterized by reducing the volume of data and therefore the size of the files needed to transmit. For this reason JSON began to be widely used until it became a standard, which does not mean that XML has stopped being used.

JSON is easy for a human to read and understand. XML is easily processable by both humans and software; in the latter case, parsing is feasible due to the strict rules governing the composition of the document.

Debugging and correction of errors is facilitated with XML, since it is relatively simple to verify that the data sent is well formed and is valid using Schema, a language that allows describing the structure and

los documentos. Con JSON la tarea es manual y requiere la verificación de que los atributos correctos del objeto están presentes en la respuesta resultante. Otro aspecto a considerar es la cantidad de recursos que se utiliza para generar una respuesta. Si el servidor debe realizar transacciones muy pesadas y complejas entonces no será muy difícil convertir la data en respuesta XML, por otro lado si la conversión a XML será lo más pesado que hará el servidor entonces es recomendable usar JSON.

Aunque en sus orígenes JSON estuvo ligado al lenguaje JavaScript, con el tiempo se ha convertido en un estándar, que no está ligado a ningún lenguaje ni plataforma. El XML fue diseñado para ser utilizado en cualquier lenguaje o alfabeto (Rodríguez, 2006).

Como explicamos anteriormente, el lenguaje XML se conoce desde hace varios años por lo que muchos lenguajes de programación se integran perfectamente con el mismo. Algunas de las API para XML han existido desde hace algunos años y por esta razón puede ser muy estable cuando se utiliza para aplicaciones complejas. Por otro lado, todas las interfaces para crear una respuesta JSON son relativamente recientes pero, a pesar de ello, es ampliamente utilizado ya que al estar basado en un subconjunto de la sintaxis de JavaScript, un lenguaje de programación interpretado, creado por la empresa Netscape Communications Corp en 1995, que todos los navegadores web modernos interpretan cuando se integra en las páginas web. También la mayoría de los lenguajes de programación han introducido soporte para el intercambio de datos basado en JSON. En la actualidad este formato cuenta con un sitio web oficial [<http://json.org>] y una especificación oficial producida por ECMA International, el mismo organismo encargado de la especificación oficial de JavaScript. La especificación oficial de JSON, publicada por primera vez en 2013, con el nombre de “Standard ECMA-404 The JSON Data Interchange Format”, puede consultarse en la web de Ecma International [<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>].

También fue publicado en forma de norma, por el Grupo de Trabajo de Ingeniería de Internet, IETF por sus siglas en inglés, entidad que regula las propuestas y los estándares de Internet.

restrictions of the contents of the documents. With JSON the task is manual and requires verification that the correct attributes of the object are present in the resulting response.

Another aspect to consider is the amount of resources used to generate a response. If the server must carry out very heavy and complex transactions then it will not be very difficult to convert the data into an XML response, on the other hand if the conversion to XML will be the heaviest the server will do then it is advisable to use JSON.

Although JSON was originally tied to the JavaScript language, over time it has become a standard, which is not tied to any language or platform. XML was designed to be used in any language or alphabet (Rodríguez, 2006).

As we explained above, the XML language has been known for several years so many programming languages integrate seamlessly with it. Some of the APIs for XML have been around for a few years and for this reason it can be very stable when used for complex applications. On the other hand, all the interfaces to create a JSON response are relatively recent but, despite this, it is widely used since it is based on a subset of the syntax of JavaScript, an interpreted programming language, created by the company Netscape Communications Corp in 1995, which all modern web browsers interpret when integrated into web pages. Also most programming languages have introduced support for JSON-based data exchange. Currently this format has an official website [<http://json.org>] and an official specification produced by ECMA International, the same body in charge of the official JavaScript specification. The official JSON specification, published for the first time in 2013, under the name “Standard ECMA-404 The JSON Data Interchange Format”, can be consulted on the Ecma International website [<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>].

It was also published in the form of a standard, by the Internet Engineering Working Group, IETF, the body that regulates Internet proposals and standards.

Una de las diferencias entre XML y JSON es la extensibilidad. Con XML se pueden crear nuevos elementos con facilidad y representar estructuras de datos arbitrarias. Con JSON esto no es posible, pero tampoco es necesario ya que, al no ser un lenguaje de marcado de documentos no lo necesita.

JSON es más efectivo cuando quieres pasar datos entre distintos módulos dentro de una misma aplicación, o sea, los datos son internos y sólo esa aplicación va a trabajar con ellos. Cuando quieres operar con otras aplicaciones, JSON es menos eficiente, ya que no te ofrece ninguna posibilidad de especificar el formato de los datos que tu aplicación necesita o soporta. Los objetos de JSON no tienen identidad, sólo son datos cuya significación está dada por el programa.

Una de las tendencias en la actualidad es la conexión a Internet de dispositivos no informáticos, tales como relojes, termostatos, televisores, refrigeradores, entre otros. Casi todos estos dispositivos son más pequeños y menos potentes que los ordenadores portátiles y de escritorio. En este campo JSON tiene muchas posibilidades de uso debido a sus características.

JSON sobrepasa hoy en día el mero papel de lenguaje de intercambio de datos, ya que se adapta muy bien a las nuevas bases de datos no relacionales hasta el punto de convertirse en formato estándar para muchos sistemas de bases de datos emergentes de tipo NoSQL.

Conclusiones

Aunque el lenguaje XML se creó y utiliza hace mucho más tiempo que JSON, la simplicidad de este último ha dado lugar a la generalización de su uso por lo que, al igual que el XML, ha sido implementado por la mayoría de las herramientas y lenguajes de programación que se utilizan en la actualidad.

JSON se está utilizando habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es muy grande, la fuente de datos es explícitamente de fiar y donde no es importante el no disponer de procesamiento XSLT para manipular los datos en el cliente; de ahí su uso por Yahoo y Google, entre otras empresas.

En la actualidad se utilizan ambos estándares para el intercambio de datos, aunque JSON cada día gana más adeptos. Seleccionar uno u otro depende de las

One of the differences between XML and JSON is extensibility. With XML you can easily create new elements and represent arbitrary data structures. With JSON this is not possible, but it is not necessary either since, as it is not a markup language of documents do not need it.

JSON is more effective when you want to pass data between different modules within the same application, that is, the data is internal and only that application will work with it. When you want to operate with other applications, JSON is less efficient, since it does not offer you any possibility to specify the data format that your application needs or supports. JSON objects have no identity, they are only data whose significance is given by the program.

One of the trends today is the Internet connection of non-computing devices, such as clocks, thermostats, televisions, refrigerators, among others, almost all of these devices are smaller and less powerful than laptops and desktops. In this field JSON has many possibilities of use due to its characteristics.

JSON nowadays surpasses the mere role of data exchange language, since it adapts very well to new non-relational databases to the point of becoming a standard format for many emerging NoSQL-type database systems.

Conclusions

Although the XML language was created and used much longer than JSON, the simplicity of the latter has led to the generalization of its use so that, like XML, it has been implemented by most tools and languages programming used today.

JSON is being commonly used in environments where the size of the data flow between client and server is very large, the data source is explicitly trustworthy, and where it is not important not to have XSLT processing to manipulate the data on the client; hence its use by Yahoo and Google, among other companies.

Currently both standards are used for data exchange, although JSON is gaining more followers every day.

circunstancias y de las preferencias que en cada momento se determinen.

Selecting one or the other depends on the circumstances and the preferences that are determined at each moment.

Bibliografía / References

- Aplicaciones Web ServInf UA. 2020. Curso librerías Web 2.0. [en línea] noviembre 2008, Disponible en: <https://si.ua.es/es/documentacion/mootools/documentos/pdf/json.pdf> [Consulta: enero 4 2020].
- Gener ,N, Enrique, J. 2009. Temas de Informática Básica. Ed. Pueblo y Educación, La Habana. 61p.
- González, N., Torres, J., Febles, J., Pozo, R., Garriga, E., García, M., Mas, M., O’Farrill, E., Piqueras, G., Barrios, L., Rodríguez, E. 1999. Informática Médica, t. 1, pp. 156-157, Ed. Ciencias Médicas, La Habana.
- Montero, R. 2001. XML. Iniciación y referencia, pp. 5-10, Ed. Mc Graw-Hill/Interamericana de España, S.A.U., Madrid.
- Peñas, J., Reyero, L. 2020. JLOP. JSONLanguageOrientedProcessing, Universidad Complutense de Madrid. [en línea] 2012. Disponible en: https://eprints.ucm.es/16693/1/Memoria_JLOPfinal.pdf [Consulta: enero 4 2020].
- Pompa, F., Rodríguez, A. 2006. El mundo de XML, *Giga*. 2:34-36, La Habana.
- Pompa, F. 2006. Guardián de la Web. World Wide Web Consortium. *Giga*. La Habana. 4:40-41pp.
- Rodríguez, A. 2006. ¿Qué es y para qué sirve JSON? .Especificación oficial de Javascript Object Notation (cu01213f), [en línea] 2006. Disponible en: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=956:i-que-es-y-para-que-sirve-json-especificacion-oficial-javascript-object-notation-diferencia-de-xml-cu01213f&catid=83&Itemid=212. [Consulta: enero 4 ,2020].
- Sturm, J. 2000. Developing XML Solutions. Ed. Microsoft Press, Washington. 14-18 pp.