Desarrollad un trabajo de investigación sobre XML y JSON que al menos incluya: Historia, normas

de uso y sintaxis, aplicaciones en el ámbito empresarial.

## Historia XML

El XML proviene de un lenguaje que inventó IBM allá por los años 70. El lenguaje de IBM se llama GML (General Markup Language) y surgió por la necesidad que tenían en la empresa de almacenar grandes cantidades de información de temas diversos.

IBM manejaba y maneja una cantidad ingente de documentación, por lo que necesitaban una manera de almacenarla, por lo que inventaron el predecesor del actual XML. Este lenguaje es GML.

En 1986 se creo la evolución de GML que fue SGML, igual que GML pero estandarizado.

En 1089 se crea el HTML, que fue utilizado en el nuevo servicio de Internet, la Web. Este lenguaje fue rápidamente adoptado por la comunidad y por varias organizaciones que crearon sus propios visores.

Desde el 96 hasta hoy una entidad llamada W3C ha tratado de poner orden en el HTML y establecer sus reglas y etiquetas para que sea un estándar. Sin embargo el HTML creció de una manera descontrolada y no cumplió todos los problemas que planteaba la sociedad global de Internet.

El mismo W3C en el 98 empezó y continúa, en el desarrollo de XML (Extended Markup Language). En este lenguaje se ha pensado mucho más y muchas personas con grandes conocimientos en la materia están trabajando todavía en su gestación. Pretendían solucionar los carencias del HTML en lo que se respecta al tratamiento de la información. Problemas del HTML como:

El contenido se mezcla con los estilos que se le quieren aplicar.

No permite compartir información con todos los dispositivos, como pueden ser ordenadores o teléfonos móviles.

La presentación en pantalla depende del visor que se utilice.

[Webgrafía](https://desarrolloweb.com/articulos/450.php) <https://desarrolloweb.com/articulos/450.php>

## Reglas de sintaxis XML y sus normas de uso

Debe seguir estas reglas al crear la sintaxis XML:

* Todos los elementos XML deben tener una etiqueta de cierre.
* Las etiquetas XML son sensibles a mayúsculas y minúsculas.
* Todos los elementos XML deben estar debidamente anidados.
* Todos los documentos XML debe tener un elemento raíz.
* Los valores de atributo siempre deben estar entre comillas.

### Todos los elementos XML deben tener una etiqueta de cierre

No está permitido omitir la etiqueta de cierre al crear la sintaxis XML. Los elementos XML deben tener una etiqueta de cierre.

Las etiquetas XML son sensibles a mayúsculas y minúsculas

Cuando crea documentos XML, la etiqueta < Body > es diferente de la etiqueta < body>.

### Todos los documentos XML deben tener un elemento raíz

Todos los documentos XML deben contener un par de etiquetas individuales para definir un elemento raíz. Todos los demás elementos deben estar dentro de este elemento raíz. Todos los elementos pueden tener subelementos (elementos hijo). Los subelementos deben estar anidados correctamente dentro de su elemento padre.

### Los valores de atributo siempre deben estar entre comillas

No está permitido omitir las comillas alrededor de los valores de atributos. Los elementos XML pueden tener atributos en pares nombre/valor; sin embargo, el valor de atributo siempre debe estar entre comillas.

[Webgrafía](https://www.ibm.com/docs/es/b2bis?topic=syntax-xml-rules) <https://www.ibm.com/docs/es/b2bis?topic=syntax-xml-rules>

## Aplicaciones en el ámbito empresarial

¿Qué tipo de XML se utiliza? El lenguaje de marcado extensible (XML) es la tecnología detrás de miles de aplicaciones, desde herramientas de productividad comunes como el procesamiento de textos hasta software de publicación de libros e incluso sistemas complejos de especificación de aplicaciones.

#### Transferencia de datos

XML permite transferir datos entre dos sistemas que almacenan los mismos datos en diferentes formatos. Por ejemplo, su sitio web almacena las fechas como MM/DD/AAAA, pero su sistema de contabilidad almacena las fechas como MM/DD/AAAA. XML le permite transferir información desde un sitio web a su sistema de contabilidad. Los desarrolladores pueden escribir código que convierta automáticamente:

* Datos del sitio web a XML
* Datos XML a datos del sistema de contabilidad.
* La información del sistema de contabilidad se devuelve en formato XML.
* Los datos XML se devuelven a los datos del sitio.

#### Aplicación web

XML proporciona la estructura de los datos que se muestran en las páginas web. Otras tecnologías de sitios web, como HTML, se utilizan junto con XML para proporcionar contenido coherente y relevante a los visitantes del sitio web. Por ejemplo, considere un sitio de comercio electrónico que vende ropa. En lugar de mostrar toda la ropa a todos los visitantes, el sitio utiliza XML para crear páginas web adaptadas a las preferencias del usuario. Filtra una etiqueta para mostrar productos de una marca específica.

#### Documentación

XML permite definir la información estructural de cualquier documento técnico. Luego, otros programas se ocupan de la estructura del documento para una presentación flexible. Por ejemplo, los párrafos, las viñetas numeradas y los títulos tienen etiquetas XML. Utilizando estas etiquetas, otros tipos de software pueden preparar automáticamente documentos para, por ejemplo, imprimir y publicar páginas web.

#### Tipo de datos

Muchos lenguajes de programación admiten XML como tipo de datos. Con este soporte, puede escribir fácilmente programas en otros lenguajes que funcionen directamente con archivos XML..

[Webgrafía](https://aws.amazon.com/es/what-is/xml/) https://aws.amazon.com/es/what-is/xml/

## Material adicional

## Historia JSON

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero que se utiliza para la comunicación entre sitios web y servidores. Se basa en parte en el lenguaje de programación JavaScript y se usa ampliamente para transferir datos en aplicaciones web.

La historia de JSON se remonta a principios de la década de 2000, cuando se desarrolló como una alternativa a XML (eXtensible Markup Language) para el intercambio de datos entre sistemas. Antes de JSON, XML era el formato de intercambio de datos más utilizado, pero se consideraba complejo y difícil de analizar. JSON, por otro lado, está diseñado para ser fácil de leer y escribir, lo que lo convierte en una característica atractiva para los desarrolladores.

Una de las principales razones por las que JSON se ha convertido en el formato dominante en el mundo del intercambio de datos es su simplicidad. Está basado en JavaScript, por lo que la mayoría de los desarrolladores lo conocen. Además, al estar en formato de texto, es fácil de leer y escribir tanto para humanos como para máquinas.

JSON es parte de JavaScript. Básicamente sigue las reglas gramaticales de JavaScript, pero su método de formato es mucho más complejo. Es un formato especial que se puede utilizar en cualquier lenguaje de programación, no solo en JavaScript. Sin embargo, dado que está basado en JavaScript, es fácil convertir datos JSON en objetos JavaScript y viceversa. Esto lo convierte en una excelente característica para la transferencia de datos en aplicaciones web donde los datos se utilizan en código JavaScript.

JSON es un formato para almacenar datos en forma de archivo. A menudo se utiliza para almacenar datos de configuración, ajustes de aplicaciones y otros tipos de datos que deben almacenarse en un formato legible por máquinas.

[Webgrafía](https://ralexrivero.medium.com/entendiendo-json-su-historia-usos-y-tecnicas-de-conversion-c84b5faacf5c) <https://ralexrivero.medium.com/entendiendo-json-su-historia-usos-y-tecnicas-de-conversion-c84b5faacf5c>

## Reglas de sintaxis XML y sus normas de uso

Los archivos en formato JSON tienen las mismas reglas que Javascript, pero además tienen unas propias del lenguaje.

#### Estructuras disponibles

Existen dos tipos de estructuras a la hora de formar un JSON, arrays y objetos.

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

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

#### Uso de llaves para representar objetos

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

Ejemplo:

{

"id": 44,

"game": "age of empires",

"author": "Ensemble Studios",

"ages": "5+"

}

#### Uso de corchetes para representar arrays

Para los arrays es necesario usar corchetes a su inicio y a su fin.

Ejemplo:

**[**

{

"id": 56431

"name": "El Padrino",

"year": 1972

},

{

"id": 7553

"name": "El Padrino 2",

"year": 1974

},

**]**

#### Valores representables con JSON

A la hora de usar valores, tenemos unas posibilidades específicas para representarlos que son:

* Un valor primitivo, que pueden ser de tipos diversos de Javascript, numérico, cadena de caracteres, y boleanos (true / false).
* 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.
* Objetos, es decir, otros elementos JSON con pares clave / valor.
* Además, también es válido el valor null.

#### Uso de comillas dobles

Se deben usar comillas al acotar cadenas y nombres de los atributos del objeto.

#### 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.

#### 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.

[Webgrafía](https://desarrolloweb.com/home/json#track69) https://desarrolloweb.com/home/json#track69

## Aplicaciones en el ámbito empresarial

El hecho de que JSON se representa en archivos de texto plano hace muy fácil su transferencia entre sistemas informáticos, usando protocolos ampliamente extendidos como HTTP.

Por tanto, JSON es un lenguaje habitualmente usado para transferir datos entre aplicaciones. En Internet el uso más importante es el de generar datos en el backend y enviarlos a las aplicaciones frontend. El flujo sería el siguiente:

Las aplicaciones frontend requieren datos para representar al usuario.

Mediante Javascript realizan solicitudes HTTP a un endpoint (URL) en el backend. Estas solicitudes se realizan de manera asíncrona mediante Ajax.

El backend recibe la solicitud sobre el dato que se requiere desde el frontend, o la operación que se desea realizar (ediciones, inserciones, etc.)

El backend realiza la operación, enviando el resultado al frontend en formato JSON.

El frontend recibe el JSON y generalmente construye mediante Javascript el HTML necesario para representar esa información convenientemente al usuario.

Este flujo es el que se realiza en los servicios web o APIs, que encontramos habitualmente en las aplicaciones web modernas.

Para el marketing también es importante. Hay muchas empresas que utilizan JSON, entre ellas Google, Facebook y Twitter, aunque hay muchas más. Esto se debe, sobre todo, a sus ventajas, siendo la principal su simplicidad. Además, también se ha podido comprobar que el software que utiliza JSON tiene un mayor rendimiento, ya que la información se procesa de una forma mucho más rápida. A todo esto hay que sumarle que es muy versátil y accesible, tanto para humanos como para máquinas.

Esto significa que utilizar JSON en una empresa hace que la página web dé una mejor respuesta, ya que el maneo de datos será mucho más fluido y sencillo. Cuanto más rápida sea la carga y más fluida sea la navegación, mejor será la experiencia de usuario, lo que le llevará a tener mejor reputación dentro de los motores de búsqueda.

Otro uso que se le puede dar a JSON, también muy útil para cualquier empresa, es que permite intercambiar datos de un sitio web a otro de forma rápida y sencilla, algo muy útil en caso de problemas de dominio.

A pesar de todo lo mencionado, aún hay profesionales que prefieren el formato XML, que aunque tiene también sus ventajas, no es un archivo tan simple y ligero como un JSON. A pesar de este detalle, sin duda es otra muy buena opción, pero su complejidad hace que no sea tan comprensible.

## Diferencias entre XML y JSON

#### La historia

El Grupo de Trabajo sobre XML concibió XML en 1996 y publicó su versión inicial en 1998. Derivaron el XML del lenguaje de marcado generalizado estándar (SGML). Tras introducir HTML en 1998, desarrollaron XML como herramienta de serialización de datos.

Douglas Crockford y Chip Morningstar publicaron JSON en 2001. Derivaron JSON de JavaScript.

#### Formato

JSON usa pares clave-valor para crear una estructura similar a un mapa. La clave es una cadena que identificará el par. El valor es la información que le da a esa clave. Por ejemplo, podríamos tener "NumberProperty": 10. En este caso, "NumberProperty" es la clave y 10 es el valor.

Por el contrario, XML es un lenguaje de marcado, un subconjunto de SGML con una estructura similar a la HTML. Almacena los datos en una estructura de árbol que presenta capas de información que puede seguir y leer. El árbol comienza con un elemento raíz (principal) antes de proporcionar información sobre los elementos secundarios. Esta estructura expansiva es útil para cargar muchas variables y configuraciones dinámicas.

#### Syntax

La sintaxis utilizada en JSON es más compacta y fácil de escribir y leer. Permite definir objetos fácilmente.

El XML es más detallado y sustituye ciertos caracteres por referencias a entidades. Por ejemplo, en lugar del carácter <, XML utiliza la referencia a la entidad &it;. XML también usa etiquetas finales, lo que lo hace más largo que JSON.

#### Análisis

Debe analizar el XML con un analizador XML, lo que a menudo ralentiza y complica el proceso.

Puede analizar JSON mediante una función de JavaScript estándar, que es más accesible. Debido a sus diferencias de sintaxis y tamaño de archivo, también puede analizar JSON más rápido que XML.

#### Documentación del esquema

La documentación del esquema describe el propósito de un archivo y muestra para qué se debe utilizar.

Los documentos XML tienen un enlace a su esquema en el encabezado. El esquema también está en formato XML, lo que le permite leer lo que puede esperar encontrar en el archivo. A continuación, puede validar el documento con el esquema y comprobar que todo se ha cargado correctamente y sin errores.

JSON también permite utilizar esquemas. Sin embargo, son más simples y permiten una mayor flexibilidad.

#### Soporte de tipos de datos

JSON solo admite un rango limitado de tipos de datos, como cadenas, números y objetos. JSON también admite matrices booleanas, algo que XML no puede hacer sin añadir etiquetas adicionales.

Sin embargo, XML es más flexible y admite tipos de datos complejos, como datos binarios y marcas de tiempo.

#### Facilidad de uso

Como lenguaje de marcado, XML es más complejo y requiere una estructura de etiquetas.

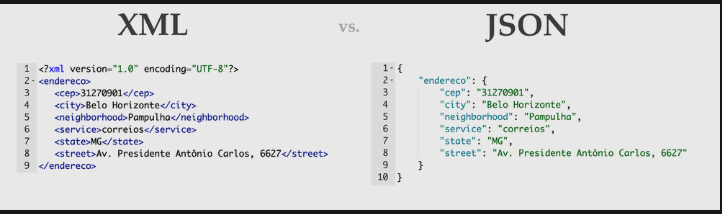
Por el contrario, JSON es un formato de datos que se extiende desde JavaScript. No utiliza etiquetas, lo que lo hace más compacto y fácil de leer para los humanos. JSON puede representar los mismos datos en un tamaño de archivo más pequeño para una transferencia de datos más rápida.

#### Seguridad

El análisis de JSON es más seguro que el XML.

[Webgrafía](https://aws.amazon.com/es/compare/the-difference-between-json-xml/) https://aws.amazon.com/es/compare/the-difference-between-json-xml/

Para finalizar vamos a poner unos ejemplos visuales de ambos.



Ejemplo Json.



Ejemplo XML:

