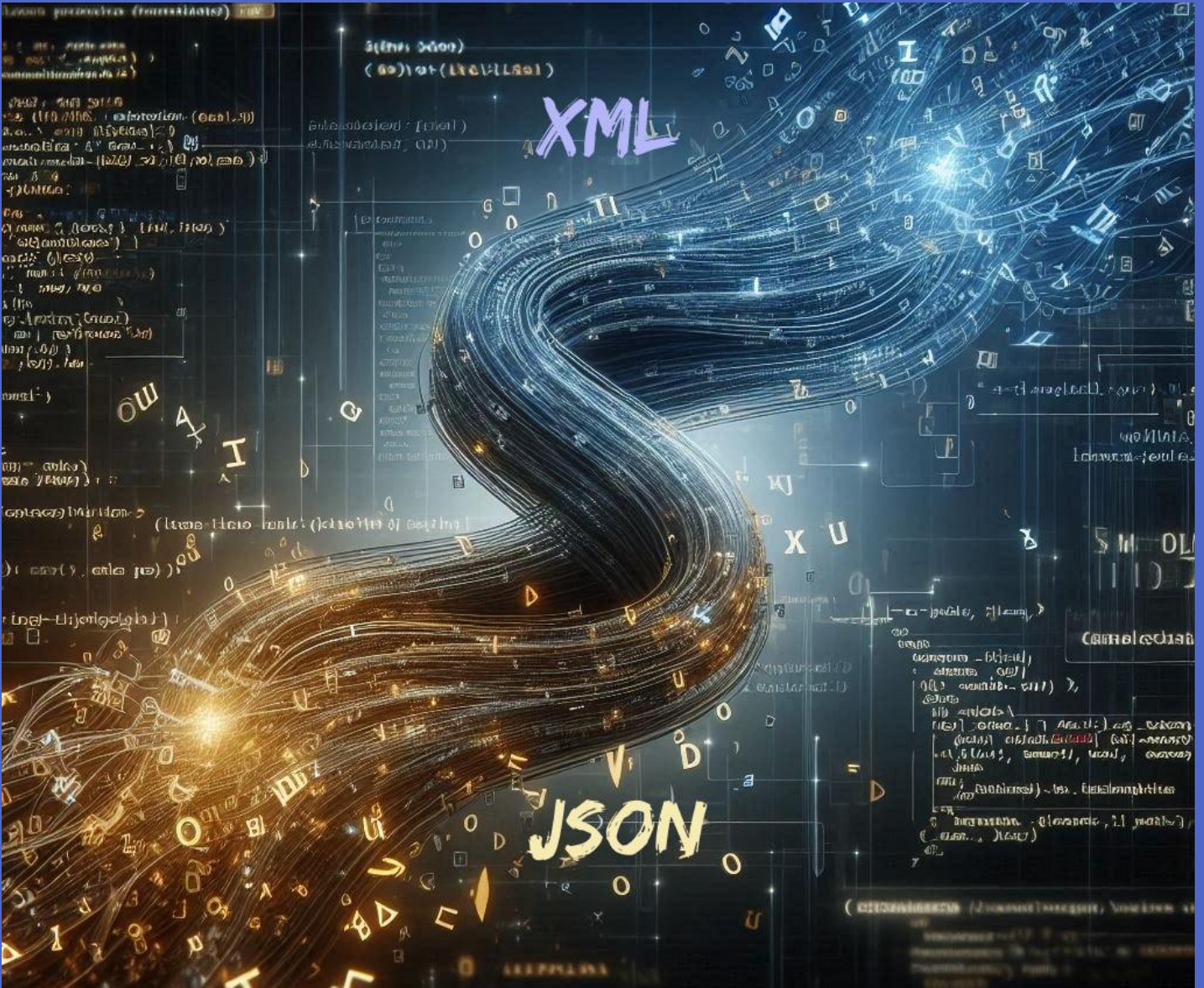


LENGUAJE DE MARCAS

FORO EVALUABLE I



ALUMNO CESUR

24/25

Alejandro Muñoz de la Sierra

PROFESOR

José Carlos Villar Herrera

Introducción

XML (Lenguaje de Marcado Extensible) y JSON (Notación de Objetos de JavaScript) son dos lenguajes importantes para organizar, guardar e intercambiar datos entre sistemas, cada uno con sus propias características y beneficios. XML, que apareció a finales de los años 90, se estableció como un estándar para representar datos gracias a su habilidad de definir estructuras complejas y validar información usando DTDs y esquemas XML. Por otro lado, JSON, que llegó en los años 2000, fue creado para dar prioridad a la simplicidad y eficiencia, convirtiéndose rápidamente en popular en el desarrollo web moderno y en APIs RESTful.

Con el avance de tecnología, JSON se ha vuelto una opción más ligera y fácil de usar con lenguajes como JavaScript, que se usa mucho en aplicaciones web y móviles. No obstante, XML sigue siendo relevante en situaciones que requieren validaciones estrictas o interoperabilidad con sistemas antiguos, como en los sectores financiero y técnico. Este análisis revisa las diferencias principales entre XML y JSON, señalando sus ventajas y desventajas, y busca averiguar cuál es la mejor opción para los retos tecnológicos de hoy.

Resolución:

Comparativa entre XML y JSON: ¿Cuál es mejor hoy?

XML (Lenguaje de Marcado Extensible) y JSON (Notación de Objetos de JavaScript) son dos lenguajes básicos para la gestión de datos. Cada uno tiene sus propias características que pueden ser más útiles en diferentes situaciones.

Características de XML

Formato de marcado: Usa etiquetas anidadas para organizar datos, parecido a HTML.

Ejemplo:

```
xml
<persona>
  <nombre>Juan</nombre>
  <edad>30</edad>
</persona>
```

Extensible y flexible: Permite crear etiquetas personalizadas según se necesite.

Capacidad de validación: Herramientas como DTD o XSD aseguran que el contenido sea correcto y esté bien estructurado.

Rico en características: Permite atributos, metadatos y transformaciones con XSLT.

Uso en documentos complejos: Es mejor para facturas electrónicas, configuraciones y documentos con estructuras complejas.

Características de JSON

Ligero y fácil de leer: Utiliza una sintaxis sencilla de pares clave-valor.

Ejemplo:

```
json
{
  "nombre": "Juan",
  "edad": 30
}
```

Diseñado para la web: Específicamente optimizado para el intercambio de datos en aplicaciones web, sobre todo en APIs RESTful.

Soporte en lenguajes de programación: Se integra bien con JavaScript y otros lenguajes, lo que mejora el desarrollo.

Rendimiento superior: Su estructura simple y tamaño reducido permiten un procesamiento más rápido que XML.

Menor redundancia: JSON usa menos espacio al no necesitar etiquetas de apertura y cierre.

¿Cuál es mejor hoy?

JSON: Se prefiere en el desarrollo moderno, sobre todo en aplicaciones web y móviles. Su simplicidad y bajo peso, además de la compatibilidad nativa con lenguajes como JavaScript lo hacen el estándar para APIs y sistemas que requieren un intercambio rápido de datos.

XML: Aunque ha perdido uso en la web, sigue siendo necesario en situaciones que requieren validación estricta y una estructura compleja, como aplicaciones empresariales, sistemas antiguos y documentos legales.

Conclusión

XML y JSON son herramientas importantes en el desarrollo hoy en día, cada uno con sus ventajas según el contexto y lo que necesite el proyecto. XML sigue siendo útil en situaciones donde son necesarias la validación estricta, estructuras jerárquicas complejas o compatibilidad con sistemas antiguos. JSON, en cambio, se ha vuelto muy popular en el desarrollo moderno por ser ligero, simple y fácil de usar con lenguajes como JavaScript.

En un entorno donde la velocidad y eficiencia son clave, JSON se ha establecido como el estándar principal para el intercambio de datos, especialmente en APIs RESTful y aplicaciones web y móviles. Sin embargo, XML sigue siendo mejor en áreas como la industria financiera, sistemas empresariales y documentación técnica, donde sus capacidades de validación y estructura jerárquica son necesarias.

En resumen, la elección entre XML y JSON debe basarse en lo que necesite el proyecto. Aunque JSON es más efectivo en muchos casos por su practicidad, XML se mantiene como una opción vital en contextos específicos. Para los desarrolladores, es esencial entender ambos formatos, ya que esto asegura una flexibilidad para enfrentar varios desafíos tecnológicos.

Referencias

https://www.w3schools.com/js/js_json_intro.asp

<https://developer.mozilla.org/en-US/docs/Web/XML>

<https://aws.amazon.com/es/compare/the-difference-between-json-xml/#:~:text=JSON%20is%20generally%20a%20better,structures%20that%20require%20data%20exchange>.

<https://stackoverflow.com/questions/5615352/xml-and-json-advantages-and-disadvantages>

<https://stackoverflow.com/search?q=JSON+vs+XML+&s=af282354-7905-409b-80d0-01a345841618>