

# RESUMEN XML

## ¿El porqué de xml?

XML es un mecanismo de propósito general para describir datos jerárquicos. Los elementos de datos se encuentran dentro de elementos y atributos. Los elementos pueden contener datos textuales, atributos y otros elementos, mientras que los atributos pueden contener únicamente datos textuales.

Siempre que se necesita almacenar datos complejos, ya sea para transferirlos entre sistemas o guardarlos en un archivo, estos deben estar marcados de alguna manera. Al volver a leerlos, debe ser posible distinguir dónde termina un registro y dónde empieza otro, y qué registro está dentro de otro. XML llenó esta laguna.

Antes de XML, los desarrolladores creaban sus propios formatos propietarios para almacenar datos, con distintos niveles de éxito. No es difícil diseñar un sistema propio para representar datos, pero también era necesario desarrollar un analizador y serializador personalizados, lo cual puede ser una tarea compleja y propensa a errores. XML proporciona un mecanismo general para marcar y representar datos. Dado que XML se ha convertido en un estándar ampliamente utilizado, existen otras herramientas y tecnologías disponibles en la mayoría de las plataformas. El uso de XML le permitirá aprovechar muchas otras tecnologías compatibles con XML, como XPATH, DTD, XSLT, XSD, XQUERY y el enlace de datos XML.

- XML simplifica el intercambio de datos
- XML simplifica el transporte de datos
- XML simplifica los cambios de plataforma
- XML simplifica la disponibilidad de datos

Muchos sistemas informáticos contienen datos en formatos incompatibles. El intercambio de datos entre sistemas incompatibles (o sistemas actualizados) es una tarea laboriosa para los desarrolladores web. Se deben convertir grandes cantidades de datos y, con frecuencia, se pierden los datos incompatibles.

XML almacena datos en formato de texto plano. Esto proporciona una forma independiente del software y del hardware de almacenar, transportar y compartir datos.

XML también facilita la expansión o actualización a nuevos sistemas operativos, nuevas aplicaciones o nuevos navegadores, sin perder datos.

Con XML, los datos pueden estar disponibles para todo tipo de "máquinas lectoras", como personas, ordenadores, máquinas de voz, fuentes de noticias, etc.

## ¿Qué es XML?

- XML significa Lenguaje de Marcado Extensible
- XML es un lenguaje de marcado muy parecido a HTML.
- XML se diseñó para almacenar y transportar datos.
- XML fue diseñado para ser autodescriptivo.

## Diferencias entre XML y HTML

XML y HTML fueron diseñados con objetivos diferentes:

- XML se diseñó para transportar datos, centrándose en qué datos son.
- HTML se diseñó para mostrar datos, con especial atención a la apariencia de los datos.
- Las etiquetas XML no están predefinidas como las etiquetas HTML.

Característica	XML (Lenguaje de Marcado Extensible)	HTML (Lenguaje de Marcado de Hipertexto)
Propósito principal	Almacenar y transportar datos.	Mostrar datos en un navegador web.
Etiquetas	Permite etiquetas personalizadas definidas por el usuario.	Utiliza un conjunto de etiquetas predefinidas.
Flexibilidad de estructura	Muy flexible, permite definir la estructura de los datos según sea necesario.	Menos flexible, la estructura está fijada por las etiquetas estándar.
Sintaxis	Estricta. Distingue entre mayúsculas y minúsculas, requiere que las etiquetas se cierren y los atributos deben ir entre comillas.	Más flexible. Los navegadores intentan corregir errores de sintaxis.
Uso	Intercambio de datos entre aplicaciones y sistemas.	Crear y diseñar páginas web con contenido visible para los usuarios.

## Caracteres de control de escape

Carácter original	Carácter con escape
<	&lt;
>	&gt;
&	&amp;
'	&apos;
"	&quot;

## Lectura de datos XML mediante programación

### DOM (Modelo de Objetos del Documento)

Un analizador DOM es un analizador XML que lee datos XML y los almacena en un conjunto de objetos. Estos objetos pueden examinarse y extraerse los datos de ellos. La estructura de los objetos DOM está estandarizada por el W3C, por lo que el código para leer XML es prácticamente estándar en múltiples plataformas.

Una de las desventajas de un analizador DOM es que todo el documento XML debe leerse en su forma de objeto (y, por lo tanto, en la memoria del sistema), lo que significa que no es posible trabajar con archivos XML muy grandes.

Los analizadores DOM existen para todas las plataformas y lenguajes principales, y normalmente están integrados en el marco central de las plataformas.

### SAX (API simple para XML)

Un analizador SAX es un lector XML más básico. Por cada entidad que lee en los datos XML, activa un evento (o devolución de llamada) que la aplicación debe gestionar o ignorar. Esta interfaz básica permite gestionar archivos XML de cualquier tamaño, ya que la aplicación solo necesita almacenar su estado actual y descartar la información ya leída cuando ya no la necesita. Sin embargo, complica el manejo de los datos, ya que la aplicación debe controlar su estado (es decir, su posición en el documento XML).

Normalmente, esta técnica se combina con un analizador DOM. La aplicación registra su posición en el XML hasta que encuentra un fragmento de datos que necesita procesar. Entonces, construye un árbol DOM basado en una pequeña sección del documento XML completo, lo procesa y lo descarta antes de continuar.

### Enlace de datos XML

La vinculación de datos XML es similar al mecanismo DOM. El documento XML se lee en un conjunto de objetos; sin embargo, en lugar de leerse en un conjunto de objetos DOM de propósito general, se lee en un conjunto de clases generadas específicamente para el tipo de documento XML que se lee. Estas clases se generan mediante un esquema XML que reconoce la forma de los documentos XML válidos.

Este mecanismo hace que sea mucho más fácil para los desarrolladores trabajar con datos XML, ya que tratan con objetos fuertemente tipados (es decir, tienen nombres y propiedades que reflejan los elementos y atributos de los datos XML).

Esta técnica también debe leer el documento XML completo en la memoria (como con DOM), pero esta restricción se puede solucionar; consulte Enlace de datos XML líquidos: cómo trabajar con archivos grandes.

Hay varias herramientas para el enlace de datos XML, Liquid Studio proporciona una solución que genera clases para C#, C++, Java, VB.Net y Visual Basic.

## Esquemas XML

Un esquema XML especifica formalmente la estructura de un documento XML. Tiene diversos usos:

- Validación: el esquema XML se puede utilizar para validar un documento XML y garantizar que contenga todos los datos correctos en los lugares correctos.
- Interoperabilidad: debido a que la forma del documento XML se describe formalmente, no hay ambigüedad, lo que significa que cada equipo que trabaja con un esquema XML determinado sabe cómo debe lucir el documento XML resultante; no hay especificaciones ambiguas con las que trabajar.
- Generación de código: El esquema XML permite generar código que permita a los desarrolladores leer y escribir datos XML mediante clases fuertemente tipadas. Esto significa que los desarrolladores solo tienen que trabajar con objetos simples con propiedades fuertemente tipadas. Esta técnica se conoce como enlace de datos XML. Liquid Studio ofrece enlace de datos XML para C#, C++, Java, VB.NET y Visual Basic.
- Visualizaciones: es posible mostrar la estructura de un esquema XML de forma gráfica, facilitando su comprensión para los desarrolladores.
- Documentación: un esquema XML puede contener documentación, que se puede generar en un formato legible para humanos; consulte la Biblioteca de estándares de esquemas XML.

Existen varios mecanismos para describir un esquema XML.

- **DTD (Definición de Tipo de Documento)** : el estándar original, definido dentro del estándar XML del W3C. El estándar DTD está prácticamente obsoleto, reemplazado por el estándar XSD del W3C. Los DTD tienen su propio formato y pueden definir sustituciones internamente, lo que requiere múltiples análisis para extraer el documento normalizado. Además, eran bastante limitados, lo que permitía la validación de cursos y una reutilización mínima.
- **XDR (XML-Data Reduced)**: un estándar desarrollado por Microsoft que superó la brecha entre los esquemas DTD y XSD. Se implementó un analizador sintáctico en MSXML hasta la versión 6, cuando se eliminó. También se utilizaba para describir datos en versiones anteriores de Biz Talk. El documento se describía en términos de XML y era muy simple, ofreciendo una validación o reutilización mínima, pero fácil de analizar y extensible.

- **XSD (Definición de Esquema XML):** ratificado por el W3C, es actualmente el mecanismo de facto para describir documentos XML. Permite la validación compleja, la reutilización mediante herencia y la creación de tipos. Se describe en términos de XML, por lo que es fácil de analizar y es compatible con la mayoría de las plataformas. Casi todos los principales estándares de datos se describen actualmente en términos de XSD.
- **RELAX NG (Lenguaje Regular para XML de Próxima Generación) :** RELAX NG tiene una estructura relativamente simple y comparte muchas características con el estándar XSD del W3C, como la tipificación de datos, la compatibilidad con expresiones regulares y espacios de nombres, y la capacidad de referenciar definiciones complejas. Existen analizadores de código abierto en la mayoría de las plataformas, pero su uso no está muy extendido.

### Ejemplo de xml:

```
<? xml version= "1.0" encoding= "UTF-8" ?>
< book isbn = "0123-456-789" >
  < title > El león, la bruja y el armario </ title >
</ book >
```

En el ejemplo, "book" y "title" son elementos, y "isbn" es un atributo. La primera línea es el encabezado XML que define la codificación de caracteres.