

CARLOS EDUARDO JARA RODRIGUEZ

102509

QUIZ TALLER 25%

1. Indique las diferencias entre DTD y XML SHEMA (Informe).

XML SHEMA:

- Es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa.
- Permite crear nuevos elementos
- Usa sintaxis de XML
- Permite tipos definidos por el usuario, llamados Arquetipos. Dando un nombre a estos arquetipos, se pueden usar en distintas partes dentro del Schema.
- Es posible trabajar con espacios de nombre, según la Especificación XML Schema Part 0: Primer, permitiendo validar documentos con varios namespaces. Es posible trabajar con espacios de nombre, según la Especificación XML Schema Part

DTD

- Describe un formato de datos
- Usa un formato común de datos entre aplicaciones
- Verifica los datos al intercambiarlos
- Verifica un mismo conjunto de datos
- Para el procesamiento del documento, las herramientas y analizadores (parsers) empleados para tratar los documentos XML deben ser capaces de procesar también las DTD.
- El mecanismo de extensión es complejo y frágil ya que está basado en sustituciones sobre cadenas y no hace explícitas las relaciones, es decir, que dos elementos que tienen definido el mismo modelo de contenido no presentan ninguna relación.
- Posee un lenguaje propio de escritura lo que ocasiona problemas a la hora del aprendizaje, pues no sólo hay que aprender XML, sino también conocer el lenguaje de las DTD.

2. Especifique los tipos de datos soportados por DTD y XML SHEMA?

DTD

- ✓ CDATA (Texto Plano).

XML SCHEMA:

- ✓ anyType
- ✓ anySimpleType
- ✓ duration
- ✓ dateTime
- ✓ time
- ✓ date
- ✓ gYearMonth
- ✓ gYear
- ✓ gMonthDay
- ✓ gDay
- ✓ gMonth
- ✓ boolean
- ✓ base64Binary
- ✓ hexBinary
- ✓ float
- ✓ double
- ✓ anyURI
- ✓ QName
- ✓ NOTATION
- ✓ String
- ✓ Decimal
- ✓ normalizedString
- ✓ integer
- ✓ token
- ✓ language
- ✓ name
- ✓ long
- ✓ nonPositiveInteger
- ✓ nonNegativeInteger
- ✓ NCName
- ✓ Short
- ✓ unsignedInt
- ✓ ID
- ✓ IDREF
- ✓ ENTITY
- ✓ Byte
- ✓ ENTITIES
- ✓ Tipos de datos típicos de lenguajes de programación.

Tipos de datos personalizados simples y complejos.

3. Es posible definir otro tipo de dato en DTD y XML SCHEMA.

En DTD no es fácil expresar la información por esta razón tiene que recurrir a esquemas, por otro lado en XML Schema si es posible definir cualquier tipo de dato, esto permite que tenga mayor ventaja que DTD.

4. Especifique las restricciones y reglas para la definición de una estructura en DTD y XML SCHEMA

Un esquema basado en una DTD tiene bastantes limitaciones. Una DTD no permite definir elementos locales que sólo sean válidos dentro de otros elementos. Por ejemplo, si queremos tener un elemento <Manager> que describa al gestor de una compañía o al de una delegación, y la definición de Manager es diferente en cada caso, con una DTD tendríamos que crear los elementos “CompanyManager” y “DelegationManager” para evitar el conflicto de nombres. Es decir, la falta de jerarquía en una DTD obliga a introducir una jerarquía a base de guiones o puntos en el espacio de nombres (Namespace). En una DTD es poco flexible la definición de elementos con contenido mixto, es decir, que incluyan otros elementos además de texto. Además no es posible indicar a qué tipo de dato (número, fecha, moneda) ha de corresponder un atributo o el texto de un elemento.

La necesidad de superar estas limitaciones propicia la aparición de otros lenguajes de esquema como XML Schema, herramientas más completas de descripción que son una alternativa a las DTD.

5. Especifique librerías para utilizar DTD y XML SCHEMA en (java, C#, Python, JavaScript PHP, C++)

XML SCHEMA:

- ✓ **Java:** xsom.jar, relaxngDatatype.jar
- ✓ **C#:** MSXML3.dll
- ✓ **Python:** scrapy, requests
- ✓ **JavaScript:** xml.js
- ✓ **PHP:** PHPDocX.lib
- ✓ **C++:** MSXML3.dll

DTD:

- ✓ **Java:** PCDATA
- ✓ **C#:** libxml2
- ✓ **Python:** libxml2

- ✓ JavaScript: xml.js
- ✓ PHP: libxml_dtdload, libxml_dtdattr, libxmldtdvalid
- ✓ C++: libxml2