LENGUAJES DE MARCAS UD-3

LENGUAJES PARA EL ALMACENAMIENTO Y TRANSMISIÓN DE INFORMACIÓN

EXTENSIBLE MARKUP LANGUAGE (XML)

ÍNDICE

[INTRODUCCIÓN 5](#_Toc134104253)

[A. LENGUAJES DE ALMACENAMIENTO Y TRANSMISIÓN DE INFORMACIÓN 5](#_Toc134104254)

[B. TIPOS DE LENGUAJES 5](#_Toc134104255)

[1. LENGUAJE DE LISTAS: JSON (JAVASCRIPT OBJECT NOTATION) 5](#_Toc134104256)

[2. LENGUAJE DE MARCAS: XML (EXTENSIBLE MARKUP LANGUAGE, LENGUAJE DE MARCADO EXTENSIBLE) 6](#_Toc134104257)

[INTRODUCCIÓN XML 7](#_Toc134104258)

[A. DEFINICIÓN 7](#_Toc134104259)

[B. EJEMPLOS DE USO XML 7](#_Toc134104260)

[1. TRANSFERENCIA Y ORGANIZACIÓN DE INFORMACIÓN 7](#_Toc134104261)

[2. LENGUAJES DERIVADOS DE XML 7](#_Toc134104262)

[3. XML EN OFIMÁTICA 8](#_Toc134104263)

[C. CARACTERÍSTICAS 8](#_Toc134104264)

[D. HERRAMIENTAS BÁSICAS DE XML 9](#_Toc134104265)

[1. EDITOR XML 9](#_Toc134104266)

[2. PROCESADOR XML (XML PARSER) 9](#_Toc134104267)

[E. GRAMÁTICAS Y VOCABULARIOS 10](#_Toc134104268)

[1. DTD (DOCUMENT TYPE DEFINITION) 10](#_Toc134104269)

[2. XSD (XML SCHEMA DOCUMENT) 10](#_Toc134104270)

[F. PRESENTACIÓN DE UN ARCHIVO XML 10](#_Toc134104271)

[G. INCORPORAR HOJA DE ESTILO CSS 10](#_Toc134104272)

[ESTRUCTURA XML 11](#_Toc134104273)

[1. DECLARACIÓN O PRÓLOGO XML (XML DECLARATION) 11](#_Toc134104274)

[2. INSTRUCCIONES DE PROCESAMIENTO 11](#_Toc134104275)

[3. DEFINICIÓN DE TIPO DE DOCUMENTO (DTD, DOCUMENT TYPE DEFINITION) 12](#_Toc134104276)

[4. COMENTARIOS 12](#_Toc134104277)

[ETIQUETAS, ELEMENTOS Y ATRIBUTOS 12](#_Toc134104278)

[1. ETIQUETAS 12](#_Toc134104279)

[2. ELEMENTOS 12](#_Toc134104280)

[3. JERARQUÍA DE ELEMENTOS 13](#_Toc134104281)

[4. ATRIBUTOS 13](#_Toc134104282)

[VALIDACIÓN 14](#_Toc134104283)

[A. DOCUMENTOS BIEN FORMADOS 14](#_Toc134104284)

[B. DOCUMENTOS VÁLIDOS 15](#_Toc134104285)

[C. DIFERENCIA ENTRE DOCUMENTO XML BIEN FORMADO Y DOCUMENTO VÁLIDO 16](#_Toc134104286)

[INCLUIR TEXTO EN XML 16](#_Toc134104287)

[A. ENTIDADES INTERNAS 16](#_Toc134104288)

[B. SECCIONES CDATA (CDATA SECTION) 17](#_Toc134104289)

[INTRODUCCIÓN DTD (DOCUMENT TYPE DEFINITION) 18](#_Toc134104290)

[A. ¿QUÉ ES DTD? 18](#_Toc134104291)

[B. INFERENCIA 18](#_Toc134104292)

[C. TIPOS DE DTD 18](#_Toc134104293)

[D. ¿DÓNDE Y CÓMO SE INCLUYE EL CÓDIGO DTD? 19](#_Toc134104294)

[ELEMENTOS XML 20](#_Toc134104295)

[A. DECLARACIÓN EN DTD 20](#_Toc134104296)

[B. CARDINALIDAD 21](#_Toc134104297)

[C. SECUENCIA DE ELEMENTOS 21](#_Toc134104298)

[ATRIBUTOS 22](#_Toc134104299)

[A. DECLARACIÓN EN DTD 22](#_Toc134104300)

[ENTIDADES 25](#_Toc134104301)

[A. REFERENCIAS ENTIDADES INTERNAS 25](#_Toc134104302)

[B. NOTACIÓN 25](#_Toc134104303)

[INTRODUCCIÓN XSD 26](#_Toc134104304)

[A. ¿QUÉ ES UN XSD (XML SCHEMA DEFINITION)? 26](#_Toc134104305)

[B. COMPARACIÓN DTD-XSD 27](#_Toc134104306)

[C. INSTANCIAS 27](#_Toc134104307)

[VALIDACIÓN DE UN ESQUEMA 28](#_Toc134104308)

[A. DOCUMENTO BIEN FORMADO 28](#_Toc134104309)

[B. DOCUMENTO VÁLIDO 29](#_Toc134104310)

[ASOCIACIÓN DE UN XML (INSTANCIA) A UN XSD (ESQUEMA) 30](#_Toc134104311)

[A. ESPACIO DE NOMBRES 30](#_Toc134104312)

[B. XML SIN ESPACIO DE NOMBRES 32](#_Toc134104313)

[C. XML CON ESPACIO DE NOMBRES 34](#_Toc134104314)

[D. ELEMENTOS CALIFICADOS Y NO CALIFICADOS 35](#_Toc134104315)

[ELEMENTOS 36](#_Toc134104316)

[A. ELEMENTOS SIMPLES 36](#_Toc134104317)

[B. ELEMENTOS COMPLEJOS 37](#_Toc134104318)

[1. INDICADORES DE ORDEN 38](#_Toc134104319)

[2. INDICADORES DE OCURRENCIA (FRECUENCIA) 39](#_Toc134104320)

[3. ELEMENTO <any> 40](#_Toc134104321)

[ATRIBUTOS 40](#_Toc134104322)

[A. DECLARACIÓN DE ATRIBUTO 40](#_Toc134104323)

[B. ELEMENTO EMPTY CON ATRIBUTOS 41](#_Toc134104324)

[C. ATRIBUTO <anyAttribute> 41](#_Toc134104325)

[TIPOS DE DATOS 42](#_Toc134104326)

[A. TIPOS PREDEFINIDOS 42](#_Toc134104327)

[B. RESTRICCIONES EN LOS DATOS (FACETAS) 42](#_Toc134104328)

[C. TIPOS CONSTRUIDOS POR EL USUARIO 45](#_Toc134104329)

[D. EXPRESIONES REGULARES 46](#_Toc134104330)

# INTRODUCCIÓN

## LENGUAJES DE ALMACENAMIENTO Y TRANSMISIÓN DE INFORMACIÓN

Existen distintos tipos de lenguajes que permiten el almacenamiento y la transmisión de información.

Mediante estos lenguajes es posible estructurar datos para compartirlos entre distintos sistemas o aplicaciones.

Diagrama

Descripción generada automáticamente

## TIPOS DE LENGUAJES

Estos lenguajes se clasifican en dos tipos:

* **De listas**: forma de codificar un documento que, junto con el texto, incorpora **símbolos y metadatos** para estructurar los datos. En esta categoría se encuadra el lenguaje **JSON** (JavaScript Object Notation).
* **De marcas**: forma de codificar un documento que, junto con el texto, incorpora **etiquetas** para estructurar los datos. En esta categoría se encuadra el lenguaje **XML** (eXtensible Markup Language). Se profundizará en este lenguaje más adelante.

1. LENGUAJE DE LISTAS: JSON (JAVASCRIPT OBJECT NOTATION)

JSON es un formato de texto sencillo para el **intercambio de datos**.

Se trata de una notación de datos procedente del lenguaje JavaScript: utiliza la sintaxis de objetos de ese lenguaje para representar datos estructurados. Debido a su amplio uso, actualmente se considera un formato independiente del lenguaje JavaScript.

No se considera lenguaje de marcas, ya que no hay diferencia en el texto a través de etiquetas.

Hay dos elementos básicos:

Texto

Descripción generada automáticamente con confianza media

* Mediante estas estructuras se pueden crear jerarquías de datos complejas.
* Un archivo JSON debe ser necesariamente codificado como UTF8. Todo el contenido debe estar comprendido entre los símbolos { y }.
* No existen esquemas de validación.
* No se pueden incluir comentarios.

1. LENGUAJE DE MARCAS: XML (EXTENSIBLE MARKUP LANGUAGE, LENGUAJE DE MARCADO EXTENSIBLE)

Se caracteriza principalmente por:

* Ser un metalenguaje.
* Provenir de SGML.
* Tener la función de compartir y estructurar datos.
* Dar soporte a bases de datos, aplicaciones de ofimática, etc.
* Ser sencillo, fiable, seguro, etc.
* No requerir de Internet.
* Tener tecnologías asociadas que lo complementan (XPath).

# INTRODUCCIÓN XML

## DEFINICIÓN

Es un lenguaje de marcas que surge de SGML y permite definir una gramática para lenguajes específicos. Nace como estándar para el intercambio de información estructurada entre distintas plataformas y es muy utilizado para el intercambio de datos entre programas y en la web.

La forma de XML es muy parecida a la de un documento HTML. Los datos contenidos en las páginas están estructurados mediante etiquetas.

La diferencia es que XML puede definir una gramática y etiquetas propias.

Texto

Descripción generada automáticamente

## EJEMPLOS DE USO XML

1. TRANSFERENCIA Y ORGANIZACIÓN DE INFORMACIÓN

Permite separar los datos de la presentación. XML no indica cómo presentar la información, un mismo documento XML puede presentarse de muchas formas. Es muy utilizado junto a aplicaciones HTML, donde XML guarda o transporta los datos, y HTML les da formato y los muestra.

Cuando los datos cambian, no es necesario tener que editar el archivo HTML (XML es un fichero independiente).

Con los lenguajes de programación web como PHP o Javascript se pueden leer archivos XML y actualizar el contenido de cualquier página HTML.

1. LENGUAJES DERIVADOS DE XML

Muchos lenguajes de propósito específico se han creado a partir de XML: SVG (Scalable Vector Graphics), MusicXML, DocBook, MathML…

1. XML EN OFIMÁTICA

**OpenDocument** es un formato de archivo, abierto y estándar, para almacenamiento de documentos ofimáticos como documentos de texto, hojas de cálculo, presentaciones y gráficos.

Actualmente es el formato de almacenamiento nativo de algunas aplicaciones de ofimática como OpenOffice y LibreOffice.

## CARACTERÍSTICAS

* XML es un metalenguaje.
* No incluye ninguna información relativa al formato (es un lenguaje de marcas descriptivo).

Las etiquetas indican el significado de los datos en lugar del formato con el que se van a visualizar. De esta forma se independiza el contenido y formato de presentación. Si quiere dar formato al documento se puede asociar una hoja de estilo.

* XML es un estándar estructurado para escribir datos en un fichero de texto.
* XML no requiere licencias, es independiente de la plataforma y tiene un amplio soporte.
* XML integra una familia de tecnologías (XML namespaces, Xpath, Xquery…).

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

* XML tiene sintaxis estricta.

Texto

Descripción generada automáticamente con confianza baja

## HERRAMIENTAS BÁSICAS DE XML

1. EDITOR XML

Una característica de los lenguajes de marcas es que se basan en la utilización de ficheros de texto plano por lo que se puede utilizar cualquier procesador de texto para construir un documento XML.

Sin embargo, existen editores especializados que contienen funcionalidades adicionales para este tipo de documentos. Por ejemplo: Amaya (W3C) y xml-copy-editor.

1. PROCESADOR XML (XML PARSER)

Cuando una aplicación necesita leer un documento XML, la aplicación recurre a un procesador XML. El procesador XML (o analizador XML, en inglés XML parser) es el que lee el documento, analiza el contenido y le pasa la información en un formato estructurado a la aplicación. La recomendación XML especifica lo que debe hacer el procesador XML, pero no entra en lo que hace después la aplicación con esa información.

Diagrama

Descripción generada automáticamente con confianza baja

## GRAMÁTICAS Y VOCABULARIOS

Un vocabulario XML es el conjunto de reglas que debe cumplir para ser válido (qué etiquetas y atributos pueden usarse, cómo debe ser el contenido de una etiqueta, las jerarquías válidas entre etiquetas…)

Para definir un vocabulario XML se emplea una gramática. Las gramáticas para documentos XML son:

1. DTD (DOCUMENT TYPE DEFINITION)

Se utiliza una definición de tipo de documento (<!DOCTYPE…>) que enlaza el fichero XML con una serie de reglas que definen el conjunto de etiquetas disponibles y la forma de utilizarlas. Es el modelo más antiguo, heredado del SGML.

1. XSD (XML SCHEMA DOCUMENT)

Se emplea un archivo XSD basado en texto que define las reglas de validación para un archivo XML.

## PRESENTACIÓN DE UN ARCHIVO XML

Los documentos XML no tienen información acerca de cómo mostrar los datos.

Cuando se abre un documento XML en un navegador, es posible que aparezca el siguiente mensaje:

Diagrama

Descripción generada automáticamente con confianza media

Para formatear XML se puede emplear PHP, JavaScript o XSLT entre otros. No es recomendable hacerlo en CSS.

XSLT son las siglas de eXensible Stylesheet Language Transformations, y es el lenguaje de hojas de estilo de CSS.

## INCORPORAR HOJA DE ESTILO CSS

Es posible incorporar una hoja de estilo CSS a un documento XML (no es recomendable). La utilidad es dar formato a la información organizada, para ello habría que incorporar la siguiente línea debajo de la línea <?xml ?>:



# ESTRUCTURA XML

Interfaz de usuario gráfica

Descripción generada automáticamente

1. DECLARACIÓN O PRÓLOGO XML (XML DECLARATION)

Sirve para definir la versión y la codificación. El encoding que se elija (codificación) debe ser igual que el que se utilice a la hora de guardar el fichero.

Debe ser la primera línea del documento XML.

Interfaz de usuario gráfica, Diagrama, Aplicación

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

1. INSTRUCCIONES DE PROCESAMIENTO

Son instrucciones que empiezan por <? y terminan por ?> y que contiene instrucciones dirigidas a las aplicaciones que leen el documento.



1. DEFINICIÓN DE TIPO DE DOCUMENTO (DTD, DOCUMENT TYPE DEFINITION)

DTD es una gramática que define la estructura de un documento XML: los elementos, atributos, entidades, notaciones, etc, que pueden aparecer, el orden y el número de veces que pueden aparecer, cuáles pueden ser hijos de cuáles, etc. El procesador XML la utiliza para verificar si un documento es válido, es decir, si el documento cumple las reglas del DTD.

Imagen que contiene Diagrama

Descripción generada automáticamente

1. COMENTARIOS

Un comentario es una etiqueta que comienza por <!-- y acaba por -->. Los comentarios no pueden estar dentro de otras marcas y no pueden contener los caracteres "--". Dentro de un comentario las entidades de carácter no se reconocen, es decir, sólo se pueden utilizar los caracteres del juego de caracteres del documento. Por ejemplo:



# ETIQUETAS, ELEMENTOS Y ATRIBUTOS

1. ETIQUETAS

Una etiqueta es un identificador que empieza por el carácter < y termina por >. Existen tres tipos de etiquetas:

Diagrama

Descripción generada automáticamente

1. ELEMENTOS

Un elemento es un componente lógico de un documento que cumple una de estas dos opciones:

* Comienza por una etiqueta de apertura y termina por la etiqueta de cierre correspondiente.
* Consiste en una única etiqueta vacía.

El contenido de un elemento es todo lo que se encuentra entre las etiquetas de apertura y cierre, incluso si estos son también elementos en cuyo caso se llaman elementos hijos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. JERARQUÍA DE ELEMENTOS

Todo documento XML tiene que tener un único elemento raíz (padre) del que desciendan todos los demás.

De esta forma, la estructura de cualquier documento XML se puede representar como un árbol invertido de elementos.

Diagrama

Descripción generada automáticamente

1. ATRIBUTOS

Un atributo es un componente de las etiquetas que consiste en una pareja nombre (name) / valor (value) y da información sobre el elemento que lo contiene. Se puede encontrar en las etiquetas de apertura o en las etiquetas vacías, pero no en las de cierre. En una etiqueta no puede haber dos atributos con el mismo nombre.

La sintaxis es siempre nombreAtributo="valorAtributo".

Por ejemplo:

Texto

Descripción generada automáticamente con confianza media

Existen múltiples formas de organizar la información empleando etiquetas y atributos, que pueden ser equivalentes. Ejemplo:

Tabla

Descripción generada automáticamente

# VALIDACIÓN

## DOCUMENTOS BIEN FORMADOS

Un documento XML debe estar bien formado, es decir debe cumplir las reglas de sintaxis de la recomendación XML. Para que un documento esté bien formado, al menos debe cumplir los siguientes puntos:

* El documento contiene únicamente caracteres Unicode válidos.
* Se diferencia entre mayúsculas y minúsculas.
* Hay un elemento raíz que contiene al resto de elementos.
* Los nombres de los elementos y de sus atributos no contienen espacios.
* El primer carácter de un nombre de elemento o de atributo puede ser una letra, dos puntos (:) o guion bajo (\_).
* El resto de caracteres pueden ser también números, guiones (-), guiones bajos (\_) o puntos (.).
* Los caracteres "<" y "&" sólo se utilizan como comienzo de marcas.
* Las letras no inglesas (á, Á, ñ, Ñ...) están permitidas. Sin embargo, es recomendable no utilizarlas para reducir posibles incompatibilidades con programas que puedan no reconocerlas.
* Las etiquetas de apertura, de cierre y vacías están correctamente anidadas (no se solapan) y no falta ni sobra ninguna etiqueta de apertura o cierre.
* Las etiquetas de cierre coinciden con las de apertura (incluso en el uso de mayúsculas y minúsculas).
* Las etiquetas de cierre no contienen atributos.
* Ninguna etiqueta tiene dos atributos con el mismo nombre.
* Todos los atributos tienen algún valor.
* Los valores de los atributos están entre comillas (simples o dobles).
* No existen referencias en los valores de los atributos.
* Puede haber espacio o salto de línea al final de una etiqueta, pero no al principio.

Texto

Descripción generada automáticamente

Ejemplos:

|  |  |
| --- | --- |
| Elemento XML | Correcto/Incorrecto |
| <persona>Eva</persona edad=”34”> | Incorrecto. El atributo está en la etiqueta de cierre. |
| <letra >R</letra> | Correcto. |
| <variables x=”2” y=”5” x=”6” /> | Incorrecto. Se repite el atributo x. |
| <nota-musica>la</nota-musica> | Correcto. |
| <Ciudad>Pamplona</ciudad> | Incorrecto. Mayúscula en la apertura, minúscula en el cierre. |
| <2colores>Rojo y azul</2colores> | Incorrecto. Número en el inicio del nombre de la etiqueta. |
| <color></color> | Correcto. |
| <lugar=”Brasil” año=”1992”/> | Incorrecto. Falta el nombre de la etiqueta. |
| <\_rojo> | Incorrecto. Falta el cierre. |
| <día>18</dia> | Incorrecto. Diferente nombre de etiqueta. |
| <persona edad=34>Eva</persona> | Incorrecto. Faltan las comillas en el valor del atributo. |
| <color principal>rojo</color principal> | Incorrecto. Hay un espacio en el nombre de la etiqueta. |
| <\_ciudad>Madrid</\_ciudad> | Correcto. |
| <mes>6<mes/> | Incorrecto. Cierre mal colocado. |
| <mes>enero< /mes> | Incorrecto. Espacios al inicio de la etiqueta |

## DOCUMENTOS VÁLIDOS

Para ser válido, un documento XML debe:

* Ser un documento bien formado.
* Incluir una referencia a una gramática.
* Incluir únicamente elementos y atributos definidos en la gramática.
* Cumplir las reglas gramaticales definidas en la gramática.

Existen distintos validadores XML.

## DIFERENCIA ENTRE DOCUMENTO XML BIEN FORMADO Y DOCUMENTO VÁLIDO

* **Documento XML bien formado**: está bien escrito sintácticamente, es decir, no hay etiquetas sin su correspondiente cierre, los nombres de elementos no contienen espacios, las comillas están bien utilizadas…
* **Documento XML válido**: está bien formado y además cumple todas las reglas especificadas en su método de validación: su gramática. Existen distintas herramientas que proporcionan el servicio de validación XML.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# INCLUIR TEXTO EN XML

## ENTIDADES INTERNAS

En XML hay algunos caracteres que pueden dar problemas al incorporarse dentro del contenido, y como resultado el documento puede no estar bien formado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para solucionarlo se pueden emplear referencias a entidades internas, cuya codificación coincide con la de HTML. El formato es: &caracteres;

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Pueden emplearse en el contenido de un elemento o en el valor de un atributo:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Existe la posibilidad de definir entidades internas propias a través de DTD.

## SECCIONES CDATA (CDATA SECTION)

Una sección CDATA es una etiqueta que comienza por y cuyo contenido el procesador XML no interpreta como marcas sino como texto. Se emplea en el contenido de los elementos.

Es decir, que si aparecen los caracteres especiales (< & " ') en una sección CDATA, el procesador XML no interpreta que empieza una marca, sino que lo considera un carácter más.

Se suele utilizar en documentos en los que aparecen muchas veces esos caracteres especiales para no tener que estar utilizando las referencias a entidades (< y &) que dificultan la lectura del documento.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

# INTRODUCCIÓN DTD (DOCUMENT TYPE DEFINITION)

## ¿QUÉ ES DTD?

Para que un documento XML sea válido, además de estar bien formado, debe respetar las reglas de la gramática que tenga definida.

Una forma de especificar la gramática es a través del DTD (Definición del Tipo de Documento): establece los elementos que pueden aparecer, su orden, cuáles son optativos, sus atributos, etc.

Actualmente la tendencia es emplear la definición de gramática a través de esquemas.

## INFERENCIA

Cuando los documentos XML tienen muchos datos, los DTD pueden ser muy extensos, por lo que su escritura resulta tediosa.

Para facilitar el proceso, existen herramientas de generación automática de DTD a partir de un documento XML, es lo que se conoce como inferencia.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## TIPOS DE DTD

* **DTD Privado**: no es accesible externamente. La palabra SYSTEM identifica al DTD como privado.
* **DTD Público**: es accesible externamente para su uso compartido. Las palabras PUBLIC y FPI (identificador público de formato) identifican al DTD como público.

Un DTD público siempre debe ser DTD externo (declarado en un fichero aparte).

## ¿DÓNDE Y CÓMO SE INCLUYE EL CÓDIGO DTD?

* **DTD Interno**: incluido en el propio documento XML. Un DTD interno siempre es privado.

Texto, Escala de tiempo

Descripción generada automáticamente

* **DTD externo**: codificado en un fichero externo. La ventaja de emplear este método es que una misma definición DTD puede emplearse para varios documentos XML. Puede ser público o privado. Tiene URL (Identificador de fichero externo).

Texto

Descripción generada automáticamente

* **Forma mixta**: integra DTD interno y externo.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

# ELEMENTOS XML

## DECLARACIÓN EN DTD

Texto

Descripción generada automáticamente con confianza media

Hay que tener en cuenta que el elemento raíz debe estar declarado como elemento. Observa en el siguiente ejemplo como el elemento raíz “mensaje” está declarado como elemento con <!ELEMENT …>.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## CARDINALIDAD

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## SECUENCIA DE ELEMENTOS

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como se ha comentado anteriormente, si se utiliza ANY como contenido de un elemento, no hay ninguna restricción respecto a lo que puede contener.

En ocasiones puede ser necesario mantener flexibilidad en el contenido de un elemento, pero no una libertad completa. Si se quiere limitar el contenido de un elemento a un conjunto de otros elementos, pero sin imponer restricciones de cantidad u orden podría usarse el formato:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

# ATRIBUTOS

## DECLARACIÓN EN DTD

Para que a un elemento se le puedan asignar atributos, hay que declararlos en el DTD:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

No existe limitación en el número de atributos que puede tener cada elemento. Pueden declararse varios atributos de un mismo elemento en un único:

Escala de tiempo

Descripción generada automáticamente con confianza media

A continuación, se exponen una serie de ejemplos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# ENTIDADES

## REFERENCIAS ENTIDADES INTERNAS

Las entidades permiten definir constantes en un documento XML. Cuando se usan dentro del documento XML se limitan por "&" y ";", por ejemplo &entidad;

¿Cómo trabaja el intérprete con ellos? Al procesar el documento XML, el intérprete sustituye la entidad por el valor que se le ha asociado en el DTD.

Hay una serie de entidades predefinidas, no es necesario declararlas. Permiten introducir caracteres especiales dentro del texto de XML:

Tabla

Descripción generada automáticamente

Por otro lado, es posible definir textos particulares como constantes que podrán utilizarse en el texto XML.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## NOTACIÓN

Una notación describe un formato de datos no XML dentro de un documento XML, como por ejemplo imágenes y otros ficheros. La utilidad es indicar el tipo de datos (declaración estándar) o una aplicación que puede interpretar estos datos. Formato:

Texto

Descripción generada automáticamente

Una vez declarado el tipo de dato no xml, es posible utilizarlo en entidades con el formato:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Un atributo de un elemento podrá tener como valor el nombre de la entidad, lo que le llevará a un fichero (definido en la entidad) que se interpretará de una forma definida en la notación:

Texto, Escala de tiempo

Descripción generada automáticamente

# INTRODUCCIÓN XSD

## ¿QUÉ ES UN XSD (XML SCHEMA DEFINITION)?

Es una gramática que permite definir la estructura de documentos XML (elementos y atributos posibles, anidación, orden, valores permitidos, obligatoriedad…), permitiendo su validación.

Es una alternativa a DTD, de hecho, actualmente es más utilizado, ya que presenta algunas ventajas, aunque por otro lado es menos legible para las personas.

Un esquema es un fichero con extensión .xsd, que es a su vez un documento XML, que establece la forma de crear documentos XML. Cada esquema define un vocabulario o lenguaje distinto, que será siempre un lenguaje XML válido.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## COMPARACIÓN DTD-XSD

Una definición DTD tiene algunas limitaciones:

* La definición DTD no es un documento XML válido, por lo que no se puede comprobar si está bien formado.
* No se puede restringir el tipo de datos (solo texto #PCDATA).
* No permite emplear espacios de nombres.
* Otras limitaciones: no se puede dar un valor por defecto a un elemento, no se puede concretar el número de repeticiones en la cardinalidad de los elementos…

Características XSD:

* La definición de un esquema XSD es un documento XML bien formado 4 Se distinguen distintos tipos de datos.
* Es más potente que DTD: existe mayores posibilidades de definición (restricciones y detalles).
* Permite emplear espacios de nombres.
* No admiten la definición de entidades (para ello se emplea DTD).

## INSTANCIAS

Los documentos XML validados por un esquema XSD concreto se denominan instancias del esquema.

El concepto es similar a tener un molde (esquema o schema), del cual se generan varias copias o instancias (documentos XML validados). Esta idea es análoga a las clases y objetos de la programación orientada a objetos.

Imagen que contiene interior, tabla, mostrador, naranja

Descripción generada automáticamente

Un esquema distinto genera un lenguaje distinto que puede instanciarse en múltiples documentos XML.

Diagrama

Descripción generada automáticamente

**Documento de instancia con esquema válido**: documento XML que se describe con un esquema XSD y cumple todas sus restricciones.

# VALIDACIÓN DE UN ESQUEMA

## DOCUMENTO BIEN FORMADO

Un esquema XSD sirve para generar o validar documentos XML, pero a su vez es un documento XML. Por ello sigue las mismas reglas para estar bien formado.

A modo de recordatorio, para que un documento esté bien formado, al menos debe cumplir los siguientes puntos:

* El documento contiene únicamente caracteres Unicode válidos.
* Se diferencia entre mayúsculas y minúsculas.
* Hay un elemento raíz que contiene al resto de elementos.
* Los nombres de los elementos y de sus atributos no contienen espacios.
* El primer carácter de un nombre de elemento o de atributo puede ser una letra, dos puntos (:) o guion bajo (\_).
* El resto de caracteres pueden ser también números, guiones (-), guiones bajos (\_) o puntos (.).
* Los caracteres "<" y "&" sólo se utilizan como comienzo de marcas.
* Las letras no inglesas (á, Á, ñ, Ñ...) están permitidas. Sin embargo, es recomendable no utilizarlas para reducir posibles incompatibilidades con programas que puedan no reconocerlas.
* Las etiquetas de apertura, de cierre y vacías están correctamente anidadas (no se solapan) y no falta ni sobra ninguna etiqueta de apertura o cierre.
* Las etiquetas de cierre coinciden con las de apertura (incluso en el uso de mayúsculas y minúsculas).
* Las etiquetas de cierre no contienen atributos.
* Ninguna etiqueta tiene dos atributos con el mismo nombre.
* Todos los atributos tienen algún valor.
* Los valores de los atributos están entre comillas (simples o dobles).

No existen referencias en los valores de los atributos.

## DOCUMENTO VÁLIDO

Hay que recordar que para que un documento sea válido, debe estar bien formado y además debe respetar las reglas de su gramática.

Diagrama

Descripción generada automáticamente

Existen múltiples validadores online que realizan la validación del fichero XSD (por ejemplo <http://www.utilities-online.info/xsdvalidation>).

Es frecuente que los editores XML permitan realizar la validación de los esquemas.

# ASOCIACIÓN DE UN XML (INSTANCIA) A UN XSD (ESQUEMA)

## ESPACIO DE NOMBRES

Ya que existe libertad para elegir los nombres de los elementos XML, siempre que sigan las normas sintácticas básicas, es posible que si se van a emplear varios esquemas se den conflictos en nombres repetidos.

Por ejemplo, si se tienen los siguientes documentos XML:

Escala de tiempo

Descripción generada automáticamente

Un **espacio de nombres XML** es una recomendación W3C para proporcionar elementos y atributos con nombre único en un archivo XML. **Un archivo XML puede contener nombres de elementos o atributos procedentes de varios vocabularios XML.** Si a cada uno de estos vocabularios se le da un espacio de nombres, un ámbito semántico propio, referenciado a una URI donde se listen los términos que incluye, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual.Cada espacio de nombres se identifica con un URI (uniform resource identifier), que es un identificador único. Por ejemplo: http://www.w3.org/2001/XMLSchema e <http://www.w3.org/2001/XMLSchema-instance>.

Un URI identifica a un espacio de nombres, pero no contiene directamente la especificación del esquema (fichero xsd).

Como se verá más adelante, dentro de un fichero XML, para asociar un prefijo determinado a un espacio de nombres se emplea el formato:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Otro ejemplo:

Fichero xml con datos de clientes y productos solicitados por éste. Tanto cliente como producto tienen un identificador “numero\_ID”. Para que no haya conflicto se pueden usar dos namespaces diferentes:

Diagrama

Descripción generada automáticamente

## XML SIN ESPACIO DE NOMBRES

En el fichero XSD es necesario utilizar el elemento reservado, que contendrá al resto de elementos. El formato básico de un esquema XSD es:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

Opcionalmente puede declararse un prefijo asociado a la norma de esquemas, para indicar que todos los nombres de elementos, atributos y tipos que se usen estarán allí declarados. Es habitual usar xs: o xsd: (schema), aunque puede emplearse otro cualquiera.

Si se emplea, todas las definiciones dentro del esquema deben ir precedidas de ese prefijo. Es recomendable emplearlo para evitar problemas con las definiciones de tipos de datos construidos por el usuario.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En el fichero XML:

Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## XML CON ESPACIO DE NOMBRES

En el fichero XSD es necesario indicar el espacio de nombres asociado con el esquema:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Para asociar el fichero XML a un esquema empleando un prefijo, es necesario declarar el prefijo que se va a utilizar (xmlns:prefijo=”espacio\_nombres”) e indicar dónde está el esquema de ese espacio de nombres (schemaLocation=”espacio\_nombres ruta\_fichero.xsd”:

Escala de tiempo

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

## ELEMENTOS CALIFICADOS Y NO CALIFICADOS

En un esquema, los elementos se dividen el globales y locales:

* + Elementos globales: son hijos directos del elemento raíz, es decir, cuelgan directamente del elemento.
  + Elementos locales: resto de elementos.

Se dice que un elemento está calificado si debe incluir el prefijo de su espacio de nombres. Los elementos globales deben estar calificados obligatoriamente. En el esquema XSD puede indicarse si los elementos están o no calificados empleando el atributo:



Si no se incluye el atributo elementFormDefault, por defecto se considera el valor “unqualified”.

Texto

Descripción generada automáticamente con confianza baja

# ELEMENTOS

## ELEMENTOS SIMPLES

Un elemento se identifica por una etiqueta en xml. Los elementos simples:

* Solo contienen texto, aunque puede ser cualquier tipo de dato (string, integer, date…)
* Pueden tener un valor por defecto o fijo.
* NO pueden contener atributos.
* NO pueden contener otros elementos hijos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## ELEMENTOS COMPLEJOS

Los elementos complejos:

* Pueden ser elementos vacíos.
* Pueden contener otros elementos.
* Pueden contener solo texto.
* Pueden contener otros elementos y texto (se denominan mixtos).
* Pueden tener atributos.

Escala de tiempo

Descripción generada automáticamente

1. INDICADORES DE ORDEN

Establecen el orden que deben tener los elementos hijos.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. INDICADORES DE OCURRENCIA (FRECUENCIA)

Establecen el número de veces que puede aparecer un elemento hijo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

1. ELEMENTO <any>

El elemento es cualquier elemento.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Puede usarse para permitir que un elemento tenga cualquier otro elemento como hijo. Hay que destacar que “cualquier otro elemento” permite elegir un elemento sin restricciones, pero para que el documento sea válido, ese elemento debe estar declarado en algún sitio (en caso contrario no podría validarse).

Texto

Descripción generada automáticamente con confianza media

# ATRIBUTOS

## DECLARACIÓN DE ATRIBUTO

Un elemento complejo puede contener atributos.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente con confianza media

## ELEMENTO EMPTY CON ATRIBUTOS

En ocasiones se desea que un elemento no tenga ningún contenido, es decir, que sea un elemento vacío o EMPTY. En este caso, es habitual configurar el elemento a través del valor de sus atributos.

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

## ATRIBUTO <anyAttribute>

El elemento es cualquier atributo.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Puede usarse para permitir que un elemento tenga cualquier atributo. Hay que destacar que “cualquier atributo” permite elegir un atributo sin restricciones, pero para que el documento sea válido, ese atributo debe estar declarado en algún sitio (en caso contrario no podría validarse).

Ejemplo completo:

Texto, Escala de tiempo

Descripción generada automáticamente

# TIPOS DE DATOS

Indica cómo debe ser el contenido de los elementos o atributos. Se distinguen tipos predefinidos y tipos construidos por el usuario.

## TIPOS PREDEFINIDOS

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## RESTRICCIONES EN LOS DATOS (FACETAS)

En ocasiones es interesante limitar el posible parlo que puede tener un dato. Formato básico:

Diagrama

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto, Escala de tiempo

Descripción generada automáticamente

## TIPOS CONSTRUIDOS POR EL USUARIO

Empleando restricciones, es posible crear tipos de datos personalizados. La ventaja es que, una vez declarado el tipo de datos, es posible utilizarlo por varios elementos o atributos.

Existen dos posibilidades:

Escala de tiempo

Descripción generada automáticamente

Cuando se usan tipos propios, es obligatorio utilizar un prefijo para los tipos de datos predefinidos (por ejemplo sx: o xsd:).

Ejemplos (solo se incluye la definición y uso el elemento):

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## EXPRESIONES REGULARES

Las expresiones regulares se utilizan en las restricciones de tipos de datos para imponer un patrón en el contenido de un elemento o atributo. Ejemplo:

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Tabla

Descripción generada automáticamente