UNIVERSIDAD DE PANAMÁ
CENTRO REGIONAL UNIVERSITARIO DE VERAGUAS

XX CONGRESO CIENTÍFICO NACIONAL

PONENCIA: ***XML: ORIGEN E IMPORTANCIA.***

EXPOSITOR: ***RAÚL ENRIQUE DUTARI DUTARI.***

FECHA: ***04 DE OCTUBRE DE 2001.***

HORA: ***04:55 P. M.***

LUGAR: ***AUDITÓRIUM DEL CENTRO REGIONAL UNIVERSITARIO DE VERAGUAS.***

DIRIGIDA A: ***PROFESORES UNIVERSITARIOS, PROFESIONALES Y ESTUDIANTES QUE PARTICIPARON EN EL EVENTO.***

DURACIÓN: ***20 MINUTOS.***

OBJETIVOS GENERALES

1. Elevar el nivel de cultura informática de los participantes.

2. Presentar la tecnología XML como una alternativa de intercambio de información en Internet.

OBJETIVOS ESPECÍFICOS

1. Describir el origen de la tecnología XML.

2. Describir las características principales de la tecnología XML.

3. Identificar algunos de los estándares relacionados con la tecnología XML.

4. Resaltar las ventajas que ofrece la tecnología XML.

5. Señalar algunas de las aplicaciones futuras de la tecnología XML.

RESUMEN DE LA PONENCIA

Se expone una visión general del metalenguaje XML como un estándar para el intercambio de datos.

Se describe el origen de la tecnología XML.

Se describen las características principales de la tecnología XML.

Se identifican algunos de los estándares relacionados con la tecnología XML.

Se resaltan las ventajas que ofrece la tecnología XML.

Se puntualizan algunas de las aplicaciones futuras de la tecnología XML.

TABLA DE CONTENIDOS

Objetivos Generales ii

Objetivos Específicos iii

Resumen de la Ponencia..... iv

Tabla De Contenidos v

1. Origen de XML. 1

2. Características de XML. 3

3. Estándares abiertos relacionados. 5

4. Ventajas de XML. 6

5. El Futuro de *XML.* 8

6. Referencias Bibliográficas. 10

1. ORIGEN DE XML.

Históricamente se puede observar que los “avances” que se dan en las tecnologías de la información presentan un patrón regular: factorización de problemas. Como ejemplos de este proceder tenemos:

- Las librerías de uso general en los sistemas operativos, simplemente colocan un conjunto de funciones en archivos que son compartidos por muchas aplicaciones;

- Las máquinas virtuales de Java permiten que el código escrito en un mismo lenguaje pueda ser interpretado en diferentes plataformas, cuya estructura es bien distinta;

- Las API's[1] para el desarrollo de aplicaciones, como JDBC[2] y ODBC[3] son una forma común de acceder a datos cuyos orígenes y formatos son bien diversos, pero que son vistos por el programador a través de una misma jerarquía de objetos, independientemente del sistema de soporte;

---

[1] API: Interfaz de Programación de Aplicaciones -Application Programming Interface-.

[2] JDBC: es un API Java para ejecutar sentencias SQL, en palabras de la propia Sun. Comúnmente se asocia a conectividad con bases de datos para Java -Java Database Connectivity-. JDBC es una marca registrada tal cual y no es un acrónimo.

Principios similares son los que animan la creación del XML[4]: separar dos partes que hasta ahora estaban indisolublemente unidas en un documento HTML[5]: la presentación y los datos que son presentados.

Se puede pensar que las CSS[6] pueden resolver este problema. Pero éstas solamente permiten definir patrones de presentación independientes que son aplicados mediante etiquetas HTML a un documento. Se trata de una separación conceptual; en esencia, los datos siguen estando en la página HTML.

Con XML la separación es total. Un documento XML contiene datos que se autodefinen, exclusivamente. Un documento HTML contiene datos, mezclados con elementos de formato. En XML se separa el contenido de la presentación de forma total.

Una forma de entender rápidamente la estructura de un documento XML, es viendo un pequeño ejemplo:

---

3 ODBC: Conectividad con objetos de bases de datos -Object Database Connectivity-.

4 XML: Lenguaje de Marcas Extensibles -Extensible Markup Language-.

5 HTML: Lenguaje de Marcas de Hipertexto -Hypertext Markup Language-.

6 CSS: Hojas de Estilo en Cascada -Cascade Style Sheets-.

```
<?xml version="1.0"?>
<mensaje>
        <remite>
                <nombre>Juan Pérez</nombre>
                <correo>jperez@coldmail.com</correo>
        </remite>
        <destinatario>
                <nombre>Juana López</nombre>
                <correo>jlopez@coldmail.com</correo>
        </destinatario>
        <asunto>Hola Bill</asunto>
        <texto>
                <parrafo>
                        ¿Hola qué tal? Hace <enfasis>mucho</enfasis> que no
                        escribes. A ver si llamas y quedamos para tomar algo.
                </parrafo>
        </texto>
</mensaje>
```

2. CARACTERÍSTICAS DE XML.

¿Estamos hablando entonces de otro lenguaje de marcas para Internet, simplemente? ¿Una especie de *HTML* mejorado? La respuesta es un **NO** rotundo. Una descripción de sus características nos dará una idea más aproximada de lo que este lenguaje puede hacer.

1. **Representación estructural de los datos:** *XML* ofrece una representación estructural de los datos que se puede implementar ampliamente y es fácil de distribuir. *XML* es un subconjunto de *SGML*[7] optimizado para el Web. *XML* garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes. La

---

7 SGML: Lenguaje de Marcas Generalizado -Standard Generalized Markup Language-.

interoperabilidad resultante está creando rápidamente una nueva generación de aplicaciones de comercio electrónico en el Web.

2. **Extensible:** En *XML* se puede definir un conjunto ilimitado de etiquetas. Un elemento *XML* puede indicar que los datos asociados son un precio de venta, un impuesto de ventas, un índice de pluviosidad o cualquier otro elemento de datos. Dado que las etiquetas *XML* son iguales en toda una organización y para todos los usuarios de Internet, existirá la capacidad de buscar y manipular datos independientemente de las aplicaciones en las que se encuentren. Una vez encontrados los datos, se pueden entregar a través de la red y presentar en un explorador de muy distintas formas, o bien se pueden pasar a otras aplicaciones para procesarlos y verlos.

3. **Los datos están separados entre la presentación y el proceso:** La potencia y la belleza de *XML* residen en el hecho de que mantiene la separación entre la interfaz de usuario y los datos estructurados. *HTML* especifica la forma de presentar los datos en un explorador, mientras que *XML* define su contenido. En *HTML*, las etiquetas se utilizan para indicar al explorador que presente los datos en negrita o en cursiva, mientras que en *XML* las etiquetas sólo se utilizan para describir los datos, como el nombre de la ciudad, la temperatura y la presión atmosférica. *XML* separa los datos de la presentación y del proceso, lo que permite mostrar y procesar los datos a su gusto con sólo aplicar distintas hojas de estilo y aplicaciones.

4. **Conversión de los datos XML en autodescriptivos:** Con *XML*, se pueden incluir *DTD's*[8], con un documento para definir sus normas, como los elementos que están presentes y la relación estructural entre los elementos. Los archivos *DTD* ayudan a validar los datos cuando la aplicación receptora no tiene una descripción incorporada de los datos entrantes. No obstante, los *DTD* son opcionales con *XML*.

5. **Interplataforma:** El formato interno de los archivos *XML* (texto plano, o *ASCII*[9]) permite su transporte y lectura bajo cualquier plataforma y le da un valor de universalidad que no se puede conseguir con ningún formato nativo, por mucha aceptación que tenga.

3. ESTÁNDARES ABIERTOS RELACIONADOS.

La iniciativa *XML* consta de un conjunto de estándares relacionados entre sí, apoyado por la W3C[10]. Entre ellos, se encuentra a:

1. **Namespaces in *XML* (Espacios de nombres, en inglés):** describe la sintaxis y la compatibilidad de los espacios de nombres para los analizadores *XML* que reconocen espacios de nombres.

---

8 DTD: Definición de Tipo de Documento -Document Type Definition-.

9 ASCII: Código estándar americano para el intercambio de información -American Standard Code for Information Interchange-.

10 W3C: World Wide Web Consortium.

2. **Document Object Model (*DOM* o Modelo de Objetos de Documento, en inglés):** Ofrece un estándar para el acceso, mediante programación, a datos estructurados (a través de secuencias de comandos), de modo que los desarrolladores puedan interactuar de forma coherente con documentos *XML* y procesarlos.

3. **Extensible Stylesheet Language (*XSL* o lenguaje de hojas de estilo extensibles, en inglés):** Es un estandar que define modos de presentación de datos y dispone de marcas de programación recursivas y condicionales para recorridos de conjuntos de registros.

4. ***XML* Linking Language (Lenguaje de enlaces *XML*, en inglés):** Es un lenguaje de vinculación que ofrece vínculos en *XML* parecidos a los de *HTML*, pero más potentes. Con *XLL*, los vínculos pueden tener varias direcciones y pueden existir en el nivel de los objetos, no sólo en el nivel de las páginas.

4. VENTAJAS DE XML.

XML aporta mucha potencia y flexibilidad a las aplicaciones basadas en el Web, proporcionando numerosas ventajas a los programadores y usuarios. Entre ellas tenemos:

1. **Búsquedas con más significado:** Los datos se pueden etiquetar de forma exclusiva con *XML*. Las búsquedas que utilizan los métodos actuales, por ejemplo, no permiten diferenciar libros escritos por XXXX, de los que se escriben sobre XXXX. Sin embargo, con *XML* los libros se pueden clasificar fácilmente en categorías estándar por autor, título, ISBN u otros criterios.

2. **Programación de aplicaciones Web flexibles:** Una vez encontrados los datos, el documento *XML* se puede distribuir a otras aplicaciones, objetos y servidores de nivel medio para continuar su procesamiento, o bien se puede entregar al escritorio para su visualización en un explorador.

3. **Integración de datos procedentes de fuentes dispares:** La capacidad de buscar en varias bases de datos no compatibles entre sí es, hoy en día, prácticamente imposible. Sin embargo, la capacidad de ampliación y la flexibilidad de *XML* le permiten describir los datos contenidos en una gran variedad de aplicaciones muy diversas, desde las recopilaciones descriptivas de páginas Web hasta los registros de datos. Además, dado que los datos basados en *XML* son autodescriptivos, se pueden intercambiar y procesar sin necesidad de una descripción incorporada de los datos entrantes.

4. **Computación y manipulación locales:** Después de entregarlos al cliente, los datos en formato *XML* se pueden analizar, editar y manipular de forma local, siendo las aplicaciones clientes quienes realizan los cálculos. Los usuarios pueden manipular los datos de diversas formas, y no limitarse a presentarlos. El DOM de XML también permite manipular datos con secuencias de comandos u otros lenguajes de programación. Los cálculos relativos a los datos se pueden realizar sin volver al servidor. La separación entre la interfaz de usuario que ve los datos y los propios datos permite crear, de forma natural, potentes aplicaciones para el Web que antes sólo se encontraban en bases de datos avanzadas, todo con un formato simple, flexible y abierto.

5. **Varias vistas de los datos:** Una vez entregados los datos al escritorio, se pueden ver de varias formas. Al describir los datos estructurados de una forma simple, abierta y extensible, *XML* sirve de complemento para el

HTML, que se utiliza ampliamente para describir las interfaces de usuario. Una vez más, mientras que el lenguaje *HTML* describe el aspecto de los datos, *XML* describe los propios datos. Dado que la presentación está separada de los datos, la definición de dichos datos en *XML* permite especificar varias vistas, lo que significa que los datos se pueden representar de la forma adecuada. Los datos locales se pueden presentar de una forma dinámica determinada por la configuración del cliente, las preferencias del usuario u otros criterios. *CSS* y *XSL* proporcionan mecanismos declarativos para describir una vista de los datos en particular.

6. **Actualizaciones granulares:** Los datos se pueden actualizar de forma granular con *XML*, por lo que no es necesario volver a enviar un conjunto completo de datos estructurados cada vez que cambia parte de dichos datos. Sólo es preciso enviar el elemento modificado del servidor al cliente, y los datos modificados se pueden presentar sin necesidad de actualizar toda la interfaz de usuario. En la actualidad, aunque sólo cambie un elemento de los datos, es preciso volver a construir toda la página, incluso si la vista permanece igual, lo que reduce drásticamente la escalabilidad del servidor.

5. EL FUTURO DE *XML.*

En su calidad de estándar del sector para expresar datos estructurados, XML ofrece muchas ventajas a las organizaciones, desarrolladores de software, sitios Web y usuarios finales. Las oportunidades aumentarán cuantos más formatos de datos de mercado vertical se creen para mercados claves, como el mercado de búsqueda avanzada en bases de datos, banca en línea, médico, legal, comercio

electrónico, etc. Cuando los sitios ofrezcan datos, en lugar de limitarse a las vistas de datos, las oportunidades serán extraordinarias.

Un mercado vital y todavía por descubrir es el de las herramientas de desarrollo que simplifican a los usuarios finales la creación de sus propios sitios Web cooperativos, lo que incluye las herramientas para generar datos XML heredados de información de bases de datos e interfaces de usuario ya existentes. Las herramientas declarativas y visuales para describir XML generadas a partir de bases de datos heredadas constituyen una oportunidad muy potente. Las herramientas personalizadas para ver datos XML se pueden escribir en el sistema de programación de Visual Basic®, en Java y en C++.

XML va a necesitar herramientas nuevas y potentes para la presentación de datos XML ricos y complejos dentro de un documento. Esto se consigue asignando una capa de presentación fácil de usar por encima de un conjunto complejo de datos jerárquicos que pueden cambiar de forma dinámica. Entre los diseños que se podrán utilizar para los datos XML se incluyen los esquemas contraíbles, las vistas dinámicas de tablas dinámicas y una sencilla hoja para cada portafolio.

Y finalmente, una llamada de advertencia: El XML y sus tecnologías nos pueden parecer muy útiles y potentes, pero... usemos la herramienta más adecuada a cada problema que se desea resolver, sin importar sí esta de moda o no. Así se logrará optimizar el uso de las tecnologías de información y sus recursos colaterales.

6. REFERENCIAS BIBLIOGRÁFICAS.

1. **MICROSOFT CORPORATION.** *Por qué XML.* http://www.microsoft.com/spain/msdn/xml/default.asp enero del 2000.

2. **REINO R., A.** *Introducción al XML en Castellano.* http://www.ibium.com/alf/xml/index.asp 26 de enero del 2000.

3. **POSADAS, M.** XML: ¿Un lenguaje de marcas o el ASCII del Futuro? Grupo EIDOS, 4 de diciembre del 2000. (http://www.eidos.es/).