



PROYECTO FINAL DE CARRERA

Universidad Politécnica de Valencia

Facultad de Informática

XML Y EL COMERCIO ELECTRÓNICO

Director: Juan Vicente Oltra Gutiérrez.

Autor: Miguel Angel Almudéver Galán.

Valencia, Julio 2002.



OBJETO

Tras cinco años estudios en la Facultad de Informática de la Universidad Politécnica de Valencia, el presente Proyecto Final de Carrera tiene por objeto la obtención del título de Ingeniero en Informática, como último paso del proceso formativo en el que he adquirido los conocimientos necesarios para poderlos aplicar en un entorno laboral, y de este modo iniciar una nueva etapa profesional desarrollando las habilidades personales y técnicas alcanzadas.

OBJETIVO

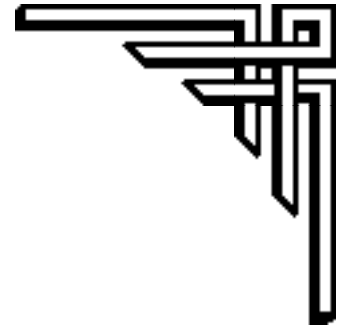
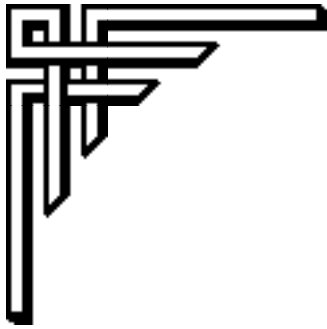
XML es un lenguaje de programación relativamente nuevo que se está introduciendo en todos los sectores de la informática. Se trata de un lenguaje de marcado que sirve para describir la estructura de los datos para que los mismos se puedan procesar, transmitir o almacenar con sencillez y eficiencia.

El comercio electrónico, por otro lado, no es nada nuevo. Se trata de un campo que ha tenido un crecimiento tan rápido como ha sido su recesión en los últimos años, debido a los excesivos costes de implantación, que lo limitaba a las grandes empresas, y a los ínfimos beneficios obtenidos comparados con las expectativas creadas.

La combinación de ambos es la fuerza que debe revitalizar el comercio electrónico, especialmente en la vertiente dedicada a la intercomunicación entre organizaciones. Con las virtudes de XML y con todas las ventajas derivadas de un procesamiento automático de la información, es el medio que permitirá a medianas y pequeñas empresas obtener ventajas competitivas en todos sus aspectos, desde su cadena de suministro hasta una mejor comunicación con sus clientes reales y potenciales.

XML ha iniciado la segunda etapa del comercio electrónico, donde los participantes disponen de la experiencia adquirida en la etapa anterior para no repetir errores previos, y los medios técnicos para superar las adversidades que lo limitaban en sus orígenes.





Bloque 0. Presentación.

- I. Índice y números de página.
- II. Prefacio.
- III. Agradecimientos.

Bloque I. Comercio Electrónico.

- Capítulo 1. *Introducción.*
- Capítulo 2. *Comercio Electrónico.*
- Capítulo 3. *EDI.*

Bloque II. XML.

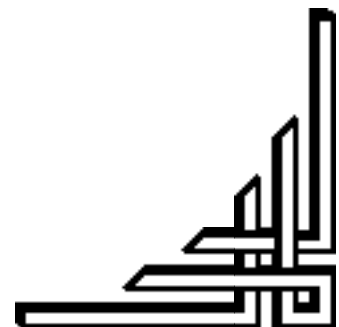
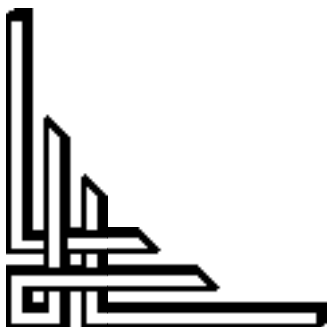
- Capítulo 4. *XML. El lenguaje.*
- Capítulo 5. *Ventajas e inconvenientes de XML.*

Bloque III. Herramientas de desarrollo.

- Capítulo 6. *ebXML.*
- Capítulo 7. *BizTalk.*
- Capítulo 8. *cXML.*

Bloque IV. Recapitulación.

- Capítulo 9. *Análisis del entorno.*
- Capítulo 10. *Conclusiones.*
- Capítulo 11. *Bibliografía.*



Bloque 0. Presentación.

I. Índice y números de página.....	7
II. Prefacio.....	11
III. Agradecimientos.....	13

Bloque I. Comercio Electrónico.

Capítulo 1. <i>Introducción</i>	15
Capítulo 2. <i>Comercio Electrónico</i>	21
Fundamentos del comercio electrónico.....	25
Tipos de comercio electrónico.....	26
AECE.....	28
Aspectos relevantes del comercio electrónico.....	30
Tecnologías esenciales del comercio electrónico.....	33
Mejores prácticas de comercio electrónico.....	36
Capítulo 3. <i>EDI</i>	39
Concepto de sistemas inter-empresariales.....	43
Implantación del EDI.....	44
Ventajas e inconvenientes.....	45
Recapitulación.....	47

Bloque II. XML.

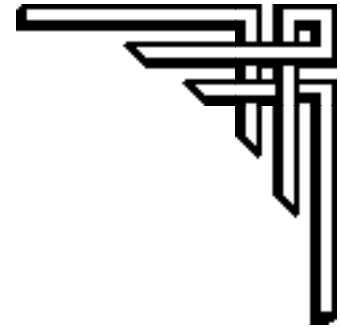
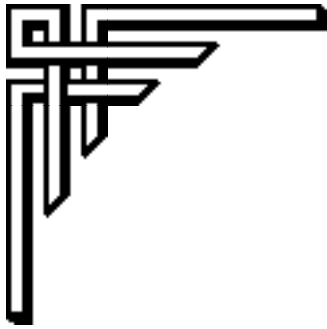
Capítulo 4. <i>XML. El lenguaje</i>	49
Orígenes y objetivos de XML.....	52
Naturaleza de XML.....	53
Características de XML. El DTD.....	55
Namespaces & Schemas.....	58
'Document Object Model' y 'Simple API for XML'.....	60
Hojas de estilo.....	63
XML en un futuro próximo y la Web.....	64
XLink, XPointer y XPath.....	66
Bases de datos XML.....	68
XML en el comercio electrónico.....	71
Conclusiones.....	73
Capítulo 5. <i>Ventajas e inconvenientes de XML</i>	75
Recapitulación.....	83

Bloque III. Herramientas de desarrollo.

Capítulo 6. <i>ebXML</i>	85
Requisitos.....	89
Modelo de interacción.....	88
Comercio electrónico.....	90
Arquitectura.....	91
Procesos de negocio.....	93
Uso de ebXML.....	96
Capítulo 7. <i>BizTalk</i>	99
Integración de aplicaciones.....	102
SOAP - Simple Object Access Protocol.....	104
Terminología.....	105
Anexos y seguridad.....	108
Requisitos.....	110
Caso de estudio.....	111
Conclusión.....	113
Capítulo 8. <i>cXML</i>	115
Especificación del protocolo.....	118
Elementos básicos.....	122
Transacciones de perfil.....	123
Definiciones de pedidos.....	124
Transacción de PunchOut.....	125
Cambios de estado posteriores.....	126
Definiciones del catálogo.....	127
Gestión de suscripciones y de mensajes.....	128
Conclusión.....	129

Bloque IV. Recapitulación.

Capítulo 9. <i>Análisis del entorno</i>	131
Participación de multinacionales.....	134
Legalidad.....	135
Seguridad.....	137
Auditoría.....	138
Buenas prácticas en XML.....	140
Capítulo 10. <i>Conclusiones</i>	143
Capítulo 11. <i>Bibliografía</i>	149
Libros.....	151
White Papers - Documentos técnicos.....	152
Documentos legales.....	154



PREFACIO

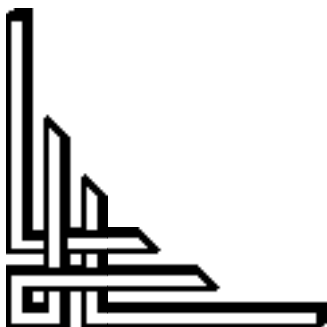
En uno de los libros que aparecen en la bibliografía del presente proyecto, Rafael Bernal, uno de los muchos buenos profesores que he tenido a lo largo de la carrera, citaba a una princesa: *'En la vida hay dos caminos: uno el que buscas, y otro el que te encuentras. El que te encuentras son los interrogantes, y el que buscas son respuestas.'*

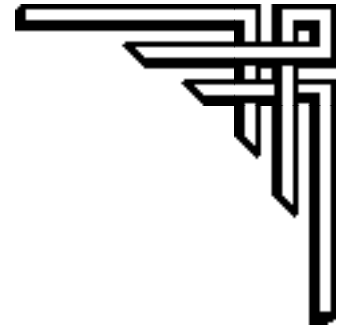
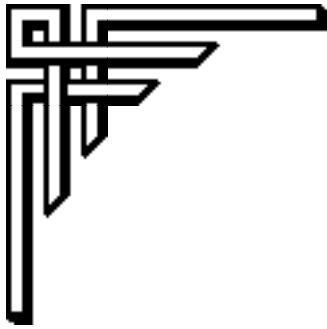
Como me ocurre muchas veces cuando voy al cine y me identifico con uno de los actores, o con los personajes de algún buen libro, e incluso con las canciones, al leer la frase anterior pensé que escondía un mensaje del que todos somos partícipes, al menos en la medida en que lo que nos sucede dependa exclusivamente de nosotros mismos: es necesario buscar un camino propio, tal y como lo hacía el caminante de Antonio Machado.

Durante cierto tiempo, el comercio electrónico ha sido un errante en busca de un lugar para su acomodo. Y a pesar de haber encontrado buenas posadas en forma de empresas capaces de invertir dinero y esfuerzo en su desarrollo, siempre ha dado sensación de fatiga.

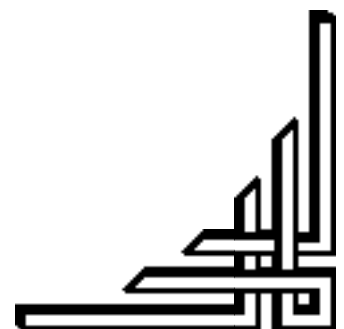
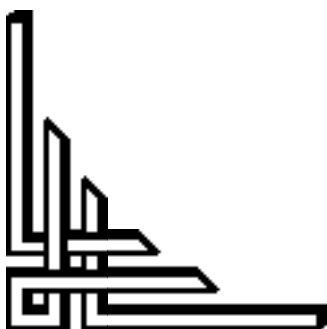
XML surge como un nuevo camino donde el comercio electrónico podrá sentirse tan cómodo como ante el fuego de una chimenea en el frío invierno. De su calor están surgiendo las llamas que deben iluminar el cielo, o la técnica, para que pequeños y medianos dejen de estar al amparo de los grandes y puedan crear su propio camino. O lo que es lo mismo: buscar sus respuestas a los interrogantes del negocio electrónico.

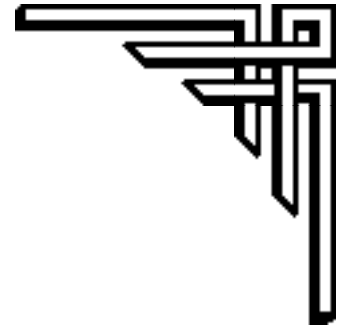
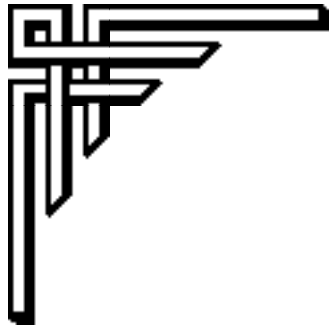
*Miguel Angel Almudéver Galán.
23 de junio de 2002.*





A mis padres.
Ellos me lo han dado todo.
Gracias.





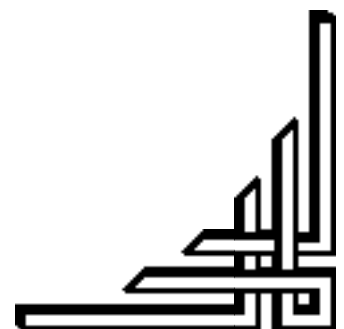
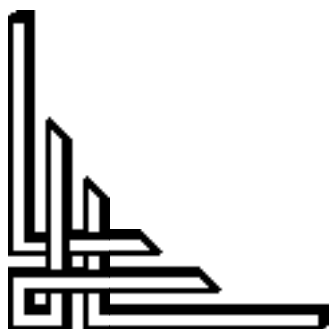
X

Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

M

INTRODUCCIÓN.

L



Hace ya algunos años que estamos viviendo la eclosión de Internet, un medio donde parece que los límites solamente los impone la imaginación. Para buscar su origen nos tenemos que remontar a 1969, cuando el gobierno americano temía que un ataque nuclear destruyera su red de comunicaciones centralizada; entonces se ideó un sistema que permitía, en caso de no poder enviar información a través de nodos preestablecidos, buscar rutas alternativas para asegurar la llegada de los paquetes de datos a su destino final.

El objetivo primordial por el que se impulsó el desarrollo electrónico subyace en la necesidad de la transmisión de la información. En ningún momento se ideó la red bajo el supuesto de un futuro desarrollo económico o comercial. Sin embargo, hoy se trata de uno de los aspectos que están fomentando su crecimiento. El potencial existente para la reducción de costes en las empresas es enorme; cada vez más, se introducen los canales de compras, distribución y logística en Internet, y se busca una mayor integración de la cadena de valor. Una integración que tiene su base en la transmisión de información - la misma base que Internet. Estamos viviendo el resurgir del comercio electrónico, porque esta vez se va a dar cabida tanto a las grandes multinacionales como a las pequeñas empresas.

El factor clave del éxito reside en el *eXtensible Markup Language*, más conocido como XML. Se trata de la energía que permitirá funcionar los motores del nuevo comercio electrónico. Al tratarse de un meta-lenguaje universal permitirá que todas las aplicaciones sean capaces de entenderse entre sí, por eso se le ha llegado a bautizar como el esperanto de la transmisión de datos. Algunos de los mayores problemas que se les presentaba a las empresas, como la comunicación entre aplicaciones de Internet y software tradicional se resolverán con relativa facilidad, y las distintas herramientas informáticas serán capaces de entenderse entre ellas mismas. Para muchas compañías será la aspirina para sus quebraderos de cabeza.

Hay un vocablo de la literatura inglesa que sirve para definir los aspectos más relevantes de XML : SPEED. Sin embargo, a pesar de que puede aumentar la velocidad y la fluidez en la gestión de la información, no es este su significado más importante.

1. S - Storing.
2. P - Publishing.
3. E - Exchanging.
4. E - Electronic.
5. D - Documents.

(almacén, publicación e intercambio de documentos electrónicos).

A pesar de lo que se está describiendo, hace tan sólo tres o cuatro años -entre 1998/99 - nadie hubiera sido capaz de prever el enorme auge de XML. No se trata de una panacea universal a todos los problemas informáticos, aunque se han depositado grandes esperanzas en su capacidad para facilitar todo tipo de comunicaciones. Además, cuando grandes empresas del tamaño de IBM, Microsoft, Sun, SAP o General Motors están de acuerdo en apoyar el estándar que está desarrollando la World Wide Web Consortium (W3C), no es una afirmación gratuita afirmar que se trata de uno de los mayores puntos de inflexión que ha sufrido la ciencia de la informática. James Utzschneider, director de una de las divisiones de negocios de Microsoft, es de la siguiente opinión al respecto:

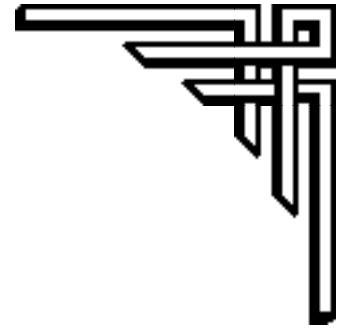
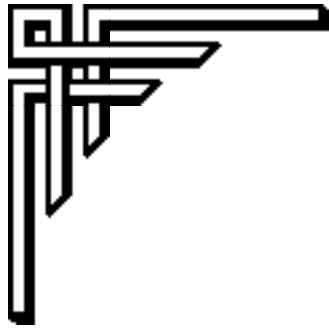
“ XML ha probado ser una forma fácil, efectiva e increíble de transmitir datos entre socios de negocios a través de Internet ”.

Pero no se trata únicamente de un nuevo estándar de Internet. Por sí mismo, XML está generando la introducción de nuevos términos como m-business (negocios Móviles) o s-business (negocios Silenciosos entre ordenadores). Cada día que pasa hay un nuevo anuncio de una aplicación basada en XML.

Uno de los mayores beneficios que va a ofrecer a la industria e centra en la extracción de la información. Una empresa típica utiliza el 40% de su presupuesto en desarrollos y actualizaciones de programas para el tratamiento de datos. Pero nos estamos centrando en el flujo interior de la misma. ¿ Qué pasa con la información que es necesario intercambiar con otras empresas ?. En este punto, el presupuesto se dispara de forma exponencial...

El estándar *de facto* para el comercio electrónico recibe el nombre de EDI - *Electronic Data Interchange*. Funciona muy bien cuando se trata de relaciones de larga duración entre clientes y proveedores, y sobre todo cuando el número de socios incluidos es reducido. Sin embargo, es difícil de implementar y, sobre todo, es muy caro. En los Estados Unidos está siendo utilizado por un reducido número de empresas, que se acercan al 2% del total. En el resto del mundo el porcentaje es todavía menor. En ello influye la dificultad que presenta EDI para introducirse en mercados abiertos o redes de abastecimiento complejas.

Para este año 2002 está prevista una inversión de 40 billones de Euros en todo el mundo en el desarrollo e implantación de herramientas ERP, EAI, software o aplicaciones Internet y comercio electrónico donde intervenga XML. La cifra no es nada despreciable, y merece por sí misma el detallado análisis que va a presentarse en este Proyecto Final de Carrera. El mundo de la informática exige un constante reciclaje en conocimientos. En estos momentos, XML empieza a imponerse en el mercado, y las empresas que lo desconozcan se arriesgan a quedarse aisladas. Es necesario aprender este nuevo esperanto para poderse comunicar con el resto del mundo.



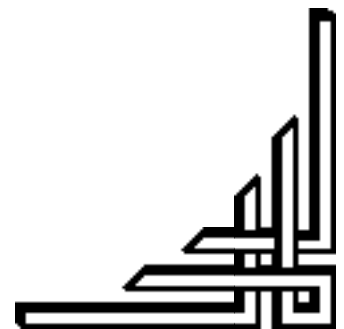
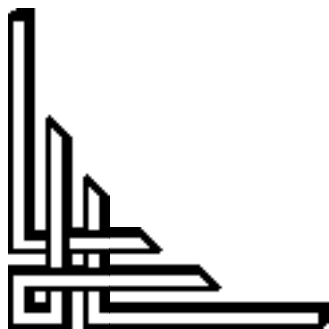
X

Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

**COMERCIO
ELECTRÓNICO.**

M

L



En 1990, Tim Berners-Lee diseñaba una herramienta capaz de poner en contacto a físicos de distintos países que necesitaban compartir grandes cantidades de información. En el CERN había nacido el hipertexto y el verano del año siguiente, HTTP ya era el lenguaje más conocido y admirado en Internet.

Desde aquel momento, los visionarios se dieron cuenta de la importancia que tenía la World Wide Web como recurso económico. A un coste relativamente bajo, se podía ofrecer un producto con un mercado potencial que se restringía ¡¡ a todo el mundo !!.

Poco a poco empezaron a nacer las famosas *punto com*. Empresas que ofrecían sus productos a través de Internet como medio alternativo y, a veces, exclusivo de venta. Después surgieron los portales: en su origen, buscadores de páginas que permitían el acceso a casi cualquier recurso disponible, y por los que navegaban gran cantidad de usuarios. En este último punto se basaron muchas compañías para pagar miles de millones de pesetas - aún no había llegado el Euro - para adquirir sus derechos de explotación.

¿ Y los usuarios ?. ¿ Y las ventas ?. Efectivamente, el mercado potencial era muy grande, pero el real era más bien reducido. Las *punto com* entraron en crisis arrojando pérdidas millonarias, la economía mundial se desplomaba con los atentados del fatídico 11 de septiembre americano y la crisis argentina sacudía a las multinacionales, algunas de ellas españolas.

El panorama era demoledor. Y de eso hace tan sólo un año. Pero desde hacía otros tres, estaba surgiendo una nueva herramienta capaz de aportar optimismo a un sector en decadencia. XML había nacido ya y, como un niño en sus primeros años de vida, empezaba a andar con paso firme. Y también a decir sus primeras palabras - o sus primeros usos. Mientras la imagen superficial invitaba a abandonar las iniciativas de comercio electrónico, una fuerza interior se daba a conocer con sus primeras actuaciones. XML venía al rescate.

Cierto es que sólo hemos hablado de comercio electrónico dirigido a un consumidor final. Muchas empresas, a pesar de la situación de la economía, hicieron fuertes inversiones para desarrollar sus negocios con otras compañías, que ahora mismo están rentabilizando. Otras, desafortunadamente, no pudieron formar parte de este tipo de negocio por los grandes costes y esfuerzo técnico y humano que supone.

XML abre una nueva vía de comunicación. De una forma simple, económica y, sobre todo, estandarizada, multiplica las posibilidades de introducirse en el comercio electrónico. La barrera de entrada que suponía el coste se levanta, y lo mismo sucede con la de la integración. El mundo y su economía no se subsanarán con el uso de XML, pero las empresas podrán mejorar sus hábitos de compras y ventas.

El problema que tuvo en sus orígenes la venta electrónica a través de Internet es que se contaba con compradores potenciales, pero no reales. La demanda estaba muy por debajo de la oferta. Sin embargo, XML ataca a un sector claramente diferenciado. Aunque puede mejorar las transacciones de compra/venta por Internet, su ámbito de actuación se debe centrar en los canales de comunicación ya establecidos. Es decir, su objetivo es el de reducir los costes de la comunicación entre clientes y proveedores. La base de su éxito es que esa relación ya existe y se puede mejorar para hacer las empresas más competitivas.

El secreto de XML no es que permite crear nuevas formas de comercio, a pesar de que lo puede hacer. Su valor añadido es que puede perfeccionar las existentes, y por eso las organizaciones le están abriendo las puertas de par en par.

Fundamentos del Comercio Electrónico.

Existen muchas definiciones de comercio electrónico. Una de las más completas, es la ofrecida por Michael J. Cunningham [B2B, 2001]:

“Transacciones de negocios efectuadas mediante redes públicas o privadas, incluyendo transacciones públicas y privadas en que se utiliza internet como instrumento de entrega. Estas transacciones incluyen transferencias financieras, intercambios en línea, subastas, entrega de productos y servicios, actividades de la cadena de abastecimiento y redes de negocios integradas.”

Sin embargo, según la perspectiva desde la que se enfoque, se pueden ofrecer definiciones más precisas y, sobre todo, adecuadas a cada uno de los sectores. Ignacio Gil, en la descripción de comercio electrónico disponible a través de su página web [<http://ttt.upv.es/igil>] propone las siguientes:

Comunicaciones: *entrega de información, productos, servicios o pagos por medio de líneas telefónicas, redes de ordenadores o cualquier otro medio electrónico.*

Procesos de negocio: *aplicación de la tecnología para obtener la automatización de procesos de negocio y flujos de trabajo.*

Servicio: *herramienta que aumenta la satisfacción de la empresa y la de los clientes mediante una reducción de costes, una mejora en la calidad de los productos y una mayor rapidez en su entrega.*

Una última definición. Quizás la políticamente más correcta, puesto que fue propuesta por la *Comisión de las Comunidades Europeas* en Bruselas, en abril de 1997:

“El comercio electrónico consiste en realizar electrónicamente transacciones comerciales; es cualquier actividad en la que las empresas y consumidores interactúan y hacen negocios entre sí o con las administraciones por medios electrónicos.”

Tipos de Comercio Electrónico.

En sus orígenes, el comercio electrónico fue utilizado por las compañías de ferrocarril y por la industria del automóvil en los años 60. Poco a poco se fue desarrollando y el espaldarazo definitivo lo recibió en los años 70, cuando se empezaron a realizar transferencias de fondos electrónicos desde las diversas instituciones financieras.

El protocolo utilizado era el EDI, o intercambio electrónico de datos que veremos más adelante. Las facturas, órdenes de compra o de venta así como todos los documentos necesarios para formalizar una transacción comercial, se llevaban a cabo de forma automática. Su desarrollo fue cada vez mayor, y finalmente se establecieron distintos tipos de comercio electrónico. Los dos fundamentales son:

1. **B2B.** *Business to business.* Comercio entre empresas.
2. **B2C.** *Business to Consumer.* Comercio negocio-consumidor.

A continuación describiremos un poco más en detalle estos dos tipos de negocio electrónico. Sin embargo, no son los únicos. Las clasificaciones son muy variadas y es posible incluir otras formas de comunicarse electrónicamente. Por ejemplo:

1. **Administración.** Por el volumen de datos que mueve, algunos autores han decidido incluir el comercio electrónico a través de la administración como una nueva clase. Los dos subconjuntos de los que estaría formado son:
 - a. *Consumidor-administración.*
 - b. *Empresa-administración.*

En una clasificación más generalista, se suele entender como una forma de empresa a la administración, y no se particulariza.

2. **C2C.** *Consumer to Consumer.* Comercio entre consumidores. Es relativamente nuevo. Además, necesita de herramientas auxiliares. Un ejemplo podría ser un sitio de subastas, donde un particular ofrece un producto y otro lo compra. A pesar de ello, es necesario pasar por un intermediario - el sitio de las subastas -, con lo que se podría resumir en dos negocios B2C paralelos.

B2B. Business to Business.

Congrega a proveedores, compradores e intermediarios que se ofrecen mutuamente sus productos en base a unas reglas de negocio predeterminadas. Una de las consecuencias directas ha sido la desaparición de los intermediarios, puesto que no ofrecían ningún valor añadido a la cadena de valor.

Las subastas son un ejemplo de este tipo de negocio. Un cliente establece una subasta en Internet para sus proveedores, que le ofrecen el producto que necesita a un precio determinado. Las ofertas son directas, sin intermediarios. Incluso existen las subastas inversas, donde un proveedor dominante puede ofrecer su producto al mejor postor.

A pesar de ello, el mayor beneficio reside en la colaboración entre empresas, y los beneficios para ambas, debido a la agilización de todas sus transacciones. Por ello, el B2B ha creado nuevas necesidades en el sector: administrar eficientemente las relaciones con los clientes, los conocimientos propios y compartidos así como la apertura de la red corporativa a las empresas con las que se colabora.

B2C. Business to Consumer.

En este caso, las empresas ofrecen sus productos a particulares. A cualquiera. El único requisito es conectarse a su web y hacer un pedido. Las estadísticas indican que cada vez se compra más por Internet, pero la evolución es lenta comparada a las expectativas que se habían creado en sus orígenes.

Tiene el riesgo de que los clientes pueden comparar precios fácilmente, con lo que el marketing y el posicionamiento de la marca son muy importantes. Algunos sitios disponen de una posición privilegiada, con lo que se permiten el convertirse en revendedores de otras empresas a cambio de un pequeño margen.

Para fidelizar al cliente, se suelen incluir boletines de noticias o, en la propia web, información relacionada con el producto de interés para el cliente. De este modo se puede crear la necesidad de compra de forma indirecta o, al menos, asegurar que se volverá a visitar el sitio.

AECE

La @ECE [<http://www.aece.org>] es la Asociación Española para el Comercio Electrónico. Dispone de una comisión de B2B cuyas empresas miembro son de la talla del *BBVA*, *ALCATEL*, *IBM España*, *VISA España* o *Retevisión*. En un resumen del estudio sobre el comercio electrónico B2B en España llegaron a las siguientes conclusiones:

[A] Principales oportunidades que plantea el B2B.

1. *Mejorar el nivel de servicio.*
2. *Ampliar el mercado.*
3. *Reducir costes.*
4. *Reducir el tiempo de aprovisionamiento.*
5. *Diferenciarse de la competencia.*

[B] Procesos en los que más se utiliza el B2B.

1. *Recepción de pedidos.*
2. *Logística.*
3. *Facturación.*

[C] Barreras encontradas en el desarrollo del B2B.

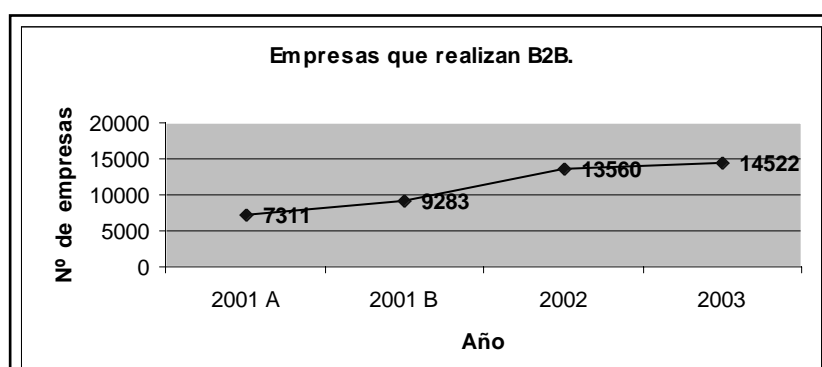
1. *Cultura de la compañía.*
2. *Integración de sistemas.*
3. *Escasa preparación de proveedores / clientes.*
4. *La relación coste / beneficio no justifica la inversión.*
5. *Desconocimiento y falta de perfiles apropiados.*

[D] Factores que contribuyen a la implantación del B2B.

1. *Las personas y su conducta.*
2. *La organización y estructura de la empresa.*
3. *La disponibilidad de sistemas apropiados.*
4. *Conocer los beneficios a obtener.*
5. *Establecimiento de indicadores de mejora.*

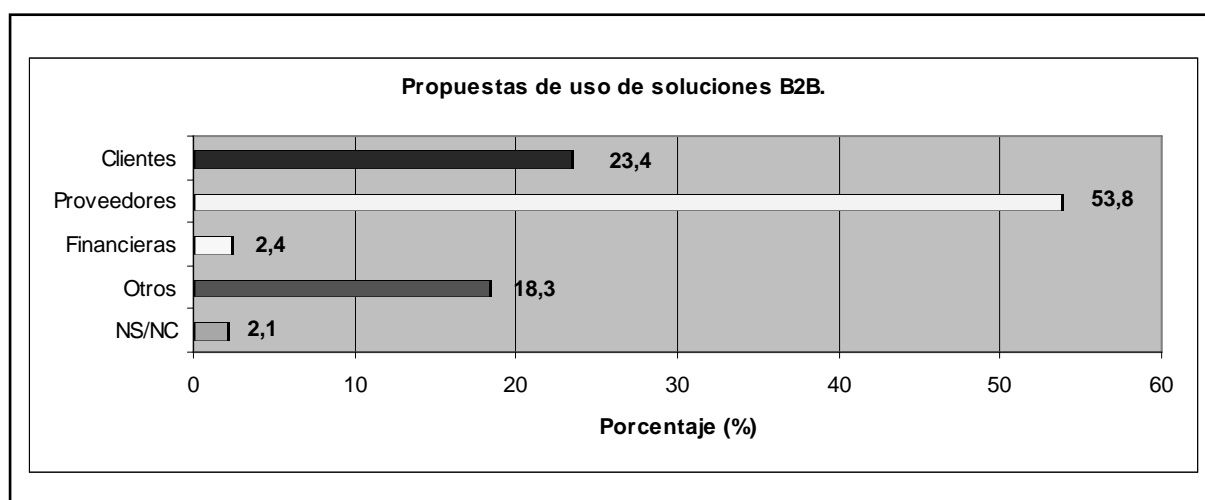
El aspecto más llamativo de los datos anteriores es que entre las barreras que se encuentran para el desarrollo del B2B - y que aparecen de igual modo en el B2C -, se encuentran el desconocimiento, la cultura de la compañía o la escasa preparación de proveedores y clientes.

La conclusión que se puede obtener es que, a medida que se genere más y mejor conocimiento sobre el B2B y tanto proveedores como clientes empiecen a participar del mismo, se producirá un efecto de arrastre. Las mismas empresas que renuncian ahora al comercio electrónico por estas causas, finalmente podrán introducirse en el mismo. Los datos estadísticos que así lo ponen de manifiesto se muestran a continuación:



2001 A / 2001 B. Primer y segundo semestre del año 2001, respectivamente

De un total de 36.375 empresas entrevistadas por la @ECE, sólo un 20,1% están realizando B2B en estos momentos. Sin embargo, de las que no lo utilizan, hasta a un 31,4% se les había propuesto el uso de este tipo de soluciones. En la parte inferior, se muestra un gráfico ilustrativo.



Aspectos relevantes del Comercio Electrónico.

En el año 2000, el comercio electrónico entre empresas, incluyendo todas sus prácticas - desde EDI hasta Internet - se ha estimado en 4,7 billones de pesetas. Un gran volumen que, sin embargo, podría ser mucho mayor.

Es evidente que su desarrollo ha ido creciendo a lo largo de los años, pero también es cierto que, sobre todo el B2C y en menor medida el B2B, no han respondido a las expectativas creadas. Existen una serie de barreras que no fueron detectadas en un primer momento, y que han estado frenando la evolución de las compras electrónicas, en beneficio de las tradicionales. Los motivos más importantes se señalan en la lista que se muestra a continuación:

1. *Seguridad.* El motivo fundamental.
2. *Indisposición a la hora de facilitar datos personales.*
3. *Desconfianza en cuanto a su efectividad.*
4. *Carencia de hábito.*
5. *Poca información sobre servicio post-venta.*
6. *Inexistencia de infraestructuras de acceso a Internet.*
7. *Necesidad de comunicación con un vendedor.*
8. *Imposibilidad de tocar o probar un producto.*

Los motivos anteriores tienen que ver mucho más con el B2C que con el B2B, aunque algunos son válidos para ambos tipos de negocios. Lo bien cierto, sin embargo, es que la mayor barrera de entrada que presente el comercio electrónico entre empresas reside en el coste de la infraestructura necesaria, que además es limitada en el número de participantes. Perdón, era. Con XML esta concepción va a cambiar, pero sigamos analizando el comercio electrónico tradicional.

Las barreras existentes son difíciles de superar, aunque con el tiempo se está logrando. Mucho tiene que ver con las ventajas que se derivan, tanto para empresas como para los compradores, del uso del negocio electrónico.

Ventajas asociadas a las empresas.

La reducción de costes es el motivo fundamental por el que las empresas se rigen, y en este aspecto pueden obtener grandes ventajas. Las podríamos dividir en dos partes:

1. **B2B.** En este sentido, las ventajas surgen por la automatización de los procesos de negocio. La emisión de solicitudes de compra o los recibos se emiten de forma automática, y ya no se necesita la figura del intermediario. El sistema pasa a ser independiente.
2. **B2C.** ¿ Cómo se hace llegar un producto al cliente ?. Tan simple como disponer de catálogos en la red - que además se actualizan de forma automática -. Se consigue, además, un servicio de 24 h. al día y 7 días a la semana.

Es imprescindible reducir costes para ser más competitivo, pero el principal activo de las empresas son siempre sus clientes. Para ellos se pueden modelizar sistemas que permitan su fidelización a la marca. Algunas de las técnicas que se pueden aplicar son:

1. **Marketing personalizado.** En función de las compras anteriores y de las compras de otros particulares, se pueden crear perfiles sobre intereses comunes y ofrecérselos a los clientes.
2. **Información complementaria.** Vendiendo artículos deportivos, por ejemplo, puede ser interesante hacer un seguimiento de las distintas competiciones y eventos, para que el usuario pueda y siga acudiendo a la web aún cuando no haga compras.

Otras ventajas que pueden surgir son la apertura a mercados a nivel internacional, realización de pruebas de mercado o acceso a mercados especializados, aunque en cierto modo eso depende de la política e intereses de la empresa. Lo que sí es cierto es que hay otro tipo de factores intangibles, como el prestigio, que aumentan con la presencia en Internet. Estos valores no se pueden medir, pero convierten a la empresa en más conocida y competitiva.

Ventajas asociadas a los clientes.

Si hablamos de B2B, los beneficios ya se han comentado en la sección anterior, por lo que en este caso nos vamos a centrar en el negocio del B2C, o de venta a particulares.

En cierto modo, las ventajas de las que se beneficia la empresa son las mismas que ofrece al consumidor. Es decir, si se produce una reducción de costes en la empresa que ofrece el producto, esto debería traducirse en una bajada de precios o en un aumento de la calidad en el servicio ofrecido. No siempre es así, pero al menos es de esperar.

Por otro lado, hay una serie de factores que sí que son independientes del servicio que ofrezca la tienda virtual, de los que nos podemos beneficiar directamente:

1. No necesitamos desplazarnos para hacer las compras. Con un simple ordenador y conexión a Internet tenemos suficiente.
2. El tiempo necesario para hacer la compra se reduce de manera drástica. Bastan varios 'clics' para adquirir un producto.
3. Comparativa. Cuantas más tiendas ofrezcan sus productos por Internet, mayor la capacidad para comparar y elegir la mejor oferta.

Es necesario que surja un equilibrio entre las ventajas que supone la compra electrónica y algunas de las barreras que habíamos citado anteriormente. Por ejemplo, mucha gente no es partidaria de entregar los números de su tarjeta de crédito. Para evitar este problema, se han creado las tarjetas recarga, que sólo son válidas cuando tienen saldo. De este modo, antes de hacer una compra se puede cargar la tarjeta y usarla sin miedo alguno, porque aunque se intercepte el número de tarjeta, al no tener ningún saldo, es inútil intentar usarla.

Transacciones.

EDI ha sido el método tradicional de intercambio de documentos electrónicos entre empresas. Establece procedimientos para cada elemento del proceso de una transacción. Sin embargo, está siendo paulatinamente desplazado por un nuevo estándar, XML, como estamos analizando a lo largo del presente proyecto.

A veces es necesario tener en cuenta la actividad fiscal asociada a las transacciones, y por ello muchos proveedores de sistemas tienen sus propios paquetes fiscales o incluso otros de proveedores especializados.

Las bases de datos, la publicación y la presentación son aspectos esenciales de todo tipo de negocio electrónico. En este aspecto XML puede ofrecer novedades como las que se citan a continuación:

1. Reutilizar y redirigir la información.
2. Reducir costos de mantenimiento.
3. Integrar flexibilidad en el sistema.
4. Compartir fácilmente la información, dentro y fuera del sistema.

Auditoría y Administración.

Para completar las soluciones B2B o B2C hay que contar con una clara visión para el desarrollo, administración y mantenimiento de todo el contenido presente. Cada sistema debe incluir los siguientes elementos:

1. Servidor Web.
2. Herramientas de desarrollo de aplicaciones.
3. Una base de datos que dirija el sistema.
4. Herramientas de edición y mantenimiento.

Las características principales que debe agrupar el conjunto anterior se centra en los servicios que ofrece. Velocidad, seguridad así como personalización y adaptación serán claves en su éxito.

Administración del conocimiento.

Son elementos críticos para desarrollar una estrategia efectiva. Tienden a abarcar una gran variedad de fuentes de información que son importantes para la organización. Algunas de las aplicaciones cliente que suelen comprender son las siguientes:

1. *Publicación técnica.*
2. *Catálogos y aplicaciones negocio a negocio.*
3. *Publicación comercial y revistas electrónicas.*

Dispone de componentes genéricos cuya éxito reside en la buena integración que se pueda realizar de los mismos. Estos sistemas no son todos ellos necesarios, pero sí convenientes para tener éxito.

- I. *Almacenes de información.*
- II. *Bases de datos con la información del cliente.*
- III. *Bases de datos con las mejores prácticas.*
- IV. *Tecnologías de recuperación y búsqueda de información.*
- V. *Sistemas expertos.*
- VI. *Software de flujo de trabajo.*
- VII. *Sistemas de administración de almacenamiento.*
- VIII. *Conversión (escáners, OCR, reconocimiento de voz...).*

Finalmente, cabe destacar tres nuevos sistemas que cada vez tienen más aceptación en las empresas por el valor añadido que les otorgan, así como las facilidades de gestión.

1. *CRM. Administración de recursos de clientes.*
2. *Administración de anuncios publicitarios.*
3. *Software de desarrollo de portales.*

Mejores prácticas de Comercio Electrónico.

Para desarrollar estrategias de comercio electrónico efectivas, es necesario tener en cuenta una gran cantidad de factores. En forma de lista y a modo de conclusión final, a continuación tenemos los más importantes, extraídos de [B2B,2001].

1. Pensar rápidamente y trabajar rápidamente.
[hay ocasiones en que ser paranoico puede ser positivo].
2. Los socios son esenciales.
[son necesarios apoyos en marketing, estrategia,...].
3. Hay que reforzar las relaciones existentes.
[objetivo: reforzar unidades de negocio y buscar nuevas fuentes].
4. Internacionalizarse rápidamente.
5. Comprender lo antes posible la cadena de valor.
[es el modelo de entrada de un mercado].
6. Reforzar la marca comercial.
[permite una entrada más económica en el sector].
7. Tener modelos de negocios multidimensionales.
[negocios, patrimonio, publicidad, contenidos, tiendas...].
8. Construir una base sólida para poder reforzarla.
9. Agregar valor.
[para mantener lealtad y fidelidad en el cliente].
10. Es necesario invertir dinero.
[promociones, regalos...].
11. Desechar antiguos programas de mercado y desarrollo.
[integrar investigación de mercado, pruebas, prototipos...].
12. Desarrollar prácticas propias.
[sabiendo que la velocidad mata, pero que el retraso es fracaso].

Cuando se habla de modelos de comercio electrónico, las referencias muestran la capacidad que han tenido algunas empresas para adaptarse a la nueva situación. Veamos la experiencia de algunas en sectores totalmente diferentes:

1. *Música*. La venta de música a través de Internet dispone de un sinfín de sitios. Sin embargo, hay pocos que muestren el detalle de cada disco compacto que se venda, información sobre los distintos conciertos de los grupos cuya música se ofrece e incluso una agenda con todos los eventos musicales más importantes en las distintas ciudades españolas.

[<http://www.fnac.es>].

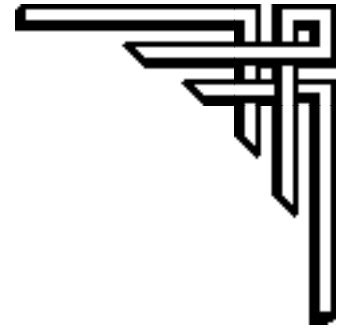
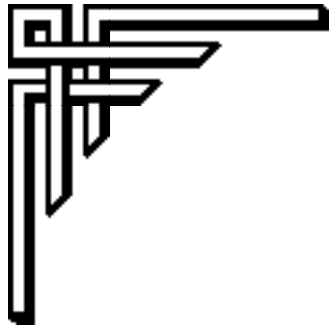
2. *Subastas*. Todos aquellos que alguna vez hayan intentado hacer sus compras por Internet o participar en subastas, conocen o han oído hablar de eBay. Con una gran facilidad para introducir a la vez que pujar por artículos, es el líder en su sector.

[<http://www.ebay.es>].

3. *Literatura*. La eterna pregunta de si el libro electrónico sustituirá alguna vez al de papel queda, por el momento, respondida. Los medios electrónicos venden - y muchos - libros de papel.

[<http://www.amazon.es>].

Como se ha podido comprobar, los modelos anteriores responden a un modelo B2C, o de usuario final. Son los visibles, puesto que las referencias a negocios B2B pasan inevitablemente por multinacionales y el uso de EDI. Aunque cada día que pasa, XML empieza a ganar más adeptos en las comunicaciones entre empresas.



X

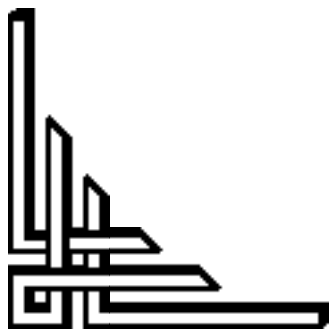
Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

M

EDI.

[ELECTRONIC DATA INTERCHANGE]

L




```
ISA*00* *00* *08*61112500TST *01*DEMO WU000003
*970911*1039*U00302000009561*0*P?
GS*PO*6111250011*WU000003 *970911*1039*9784*X*003020
ST*850*397822
BEG*00*RE*1234**980208
REF*AH*M109
REF*DP*641
REF*IA*000100685
DTM*010*970918
N1*BY*92*1287
N1*ST*92*87447
N1*ZZ*992*1287
PO1*1*1*EA*13.33**CB*80211*IZ*364*UP*718379271641
PO1*1*2*EA*13.33**CB*80211*IZ*382*UP*718379271573
PO1*1*3*EA*13.33**CB*80213*IZ*320*UP*718379271497
PO1*1*4*EA*13.33**CB*80215*IZ*360*UP*718379271848
PO1*1*5*EA*13.33**CB*80215*IZ*364*UP*718379271005
CTT*25
SE*36*397822
GE*1*9784
IEA*1*000009561
```

EDI significa *Electronic Data Interchange*, o lo que es lo mismo, Intercambio Electrónico de Datos. ¿ Y las cifras y letras anteriores ? . No es más que una orden de compra donde se especifican detalles como el comprador, el producto, la cantidad de artículos de la compra o su precio. Y por supuesto se trata de un ejemplo de una implementación realizada con EDI.

XML y EDI son muy diferentes. Los creadores del último tenían como especial preocupación el tamaño de los mensajes y el ancho de banda que era, y a veces sigue siendo, especialmente caro. Los datos se comprimen, el formato es difícil de entender y se requiere gente bien cualificada que cuesta preparar. Las aplicaciones son costosas y difíciles de mantener, pero en los orígenes del comercio electrónico, era la forma óptima para su desarrollo.

EDI se puede utilizar bajo la nomenclatura de dos estándares diferentes, ANSI X12 y EDIFACT. Disponen de varias versiones y a pesar de que su compatibilidad no es siempre evidente y de que existen otros grupos menores, son los más utilizados.

1. ANSI X12 es el estándar americano, y ha evolucionado desde que surgió en los años 60 como una forma de comercio entre las grandes compañías. Ahora se ha convertido en redes entre clientes y proveedores que mueven billones de dólares.
2. EDIFACT es el estándar internacional desarrollado desde cero a partir del año 1985, bajo la tutela de las Naciones Unidas.

EDI mueve, distribuye y genera grandes cantidades de dinero, pero una parte de dichos recursos es necesario haberlos dispuesto antes para poderlo utilizar. Tiene unas características muy peculiares debido a su necesidad de convertirse en un formato optimizado para la compresión de los mensajes que debe distribuir:

1. Necesita servidores dedicados, cuyo precio suele oscilar entre los 15.000 y 150.000 Euros.
2. Usa redes de valor añadido para su difusión, cuyo costo por mensaje varía entre 1 y 25 Euros. A veces incluso más.
3. Su formato obliga a meses de esfuerzo en su aprendizaje.
4. Necesita programadores altamente cualificados en C++.

Todas estas cifras deben orientarnos a la hora de deducir lo costosa de una aplicación de comercio electrónico hasta nuestros días. A pesar de ello un número importante de empresas han apostado por él, obteniendo beneficios. Es necesario destacar que, en algunas ocasiones, las organizaciones sólo adaptan EDI de forma parcial, por ejemplo con una parte de sus proveedores. Pero sigue valiendo la pena.

Concepto de Sistemas Inter-Empresariales.

“Un Sistema Inter-Empresarial (SIE) es un sistema de información automatizado compartido entre dos o más empresas”, [STIG, 2000].

Un SIE es, en cierta medida, una mezcla de la informática y las telecomunicaciones aplicadas al mundo de los negocios, con el objetivo de alcanzar una comunicación sin papeles. Es decir, tomando como base las transacciones electrónicas, se realiza la transmisión del mismo tipo de información legal que se debería realizar sobre papel.

Uno de los problemas que surge cuando se plantea la posibilidad de crear una plataforma de comercio electrónico es la diferencia de objetivos e intereses que pueden plantear las empresas a la hora de afrontar la experiencia. En este sentido, un clima de total fiabilidad y seguridad debe ser establecido bajo los siguientes puntos:

1. Privacidad de la información.
2. Integridad de la información.
3. Existencia de motivaciones comunes a las empresas.
4. Seguridad en cuanto a su viabilidad económica.
5. Confianza en cuanto a la obtención de beneficios, que pueden ser tangibles o intangibles, pero que aporten ventajas competitivas.

La legalidad es un aspecto importante de este tipo de negocios. En concreto, la ley sobre el IVA incluye en el segundo párrafo del artículo número 82 la siguiente sentencia:

“La repercusión del impuesto deberá efectuarse mediante factura o documento análogo que podrá emitirse vía telemática”.

Por lo tanto, a pesar de que el objetivo es dejar de trabajar con el papel, se sigue necesitando almacenar de alguna manera todo tipo de transacciones puesto que la Administración puede presentar un requerimiento de las mismas en cualquier tipo de formato. En tal caso, habría que llevar a cabo la transformación pertinente.

Implantación del EDI.

Puesto que ya conocemos los altos costes de la implantación de EDI en las organizaciones, huelga destacar la importancia de un buen análisis previo antes de su implantación. Análogamente al resto de tecnologías del mercado, incluye dos aspectos interrelacionados:

1. Valor tecnológico.
2. Valor organizacional y sectorial, entendiendo como tal la forma en que EDI puede cambiar la forma de trabajo de una empresa concreta o las de su entorno.

Para realizar un estudio completo de lo que supone el cambio, no hay nada mejor que solicitar un buen informe a una consultora de prestigio y conocedora del sector. Si por el contrario la propia organización es capaz de asumir dicha labor, sus esfuerzos se deberán centrar en tres actividades fundamentales:

1. Estudio previo de viabilidad.

Por un lado, es necesario conocer de antemano todos los costes asociados al proceso de implantación, centrándose en el esfuerzo humano necesario, difícil de estimar. A su vez, hay que evaluar los cambios organizacionales que impondrá la nueva forma de trabajo.

2. Análisis de flujos de información.

Agentes, roles, documentos, datos, interlocutores externos así como internos... todas las personas o elementos que interactúan en una transacción comercial deben ser tenidos en cuenta para un buen desarrollo de la plataforma.

3. Implantación de la solución.

Elección de los productos software y hardware necesarios, así como el estándar a utilizar o los documentos a incluir en los intercambios, por ejemplo. Definición de las formas de trabajo, desarrollo de la solución, ejecución de pruebas y evaluación de resultados. Todo ello para conseguir el objetivo previsto, disponer de una herramienta de *Intercambio Electrónico de Datos*.

Ventajas e inconvenientes.

Las verdaderas ventajas de utilizar sistemas EDI se pueden reducir a dos puntos fundamentales:

1. Reducción de costes.
2. Diferenciación de la empresa.

En el primer caso, la reducción de costes se puede evaluar desde distintos puntos de vista. Siendo el objetivo final la eliminación del papel como medida visible y la obtención de sistemas *Just In Time* como método de trabajo, se observan los siguientes cambios:

- I. Reducción de stock.
- II. Ciclo de producción más corto.
- III. Integración de la cadena de suministro.
- IV. Rapidez de respuesta ante imprevistos.
- V. Agilidad en el tratamiento de la información.
- VI. Facilidad para el crecimiento y la expansión.

Por otro lado, la diferenciación en el sector va a repercutir en aspectos tales como la imagen de la empresa, su posición dominante - asociada al liderazgo tecnológico -, o un mayor poder de negociación con sus proveedores y clientes. El resultado final es el siguiente:

- I. Mejora de servicio al cliente.
- II. Aumento de las barreras de salida.
- III. Reducción de costes de aprovisionamiento.
- IV. Políticas personalizadas de ventas y marketing.

Los beneficios del uso de EDI no son en ningún caso despreciables, puesto que son capaces de transformar todos los segmentos o departamentos de la empresa, así como las relaciones externas - llámense clientes o proveedores.

En cualquier caso, la automatización del proceso introduce una serie de riesgos. En la medida en que se puedan controlar, mayor será la fiabilidad y seguridad del entorno. Por ello, es imprescindible tratar de mantener bajo control:

1. Las redes de comunicaciones.
2. Una previsión de posibles ataques maliciosos.
3. Confidencialidad de los datos:
 - I. Identidad del emisor y del receptor.
 - II. Confidencialidad del contenido.
 - III. Integridad de la transacción.
 - IV. No repudio (irrefutabilidad).
4. Posibles errores en el procesamiento.
5. Falta de honestidad.
6. Incompatibilidad entre versiones.

Ante la existencia de estos riesgos, las organizaciones deben prestar atención especial a sus políticas de protección de datos - en particular, es necesario evaluar la forma en que controlan los accesos a la red y los distintos protocolos de seguridad.

Una de las características de EDI es que hace uso de VANs, o redes de valor añadido. Es decir, que existen empresas externas que se dedican a controlar todas las eventualidades que pueden surgir en su uso. Por supuesto, la diferencia es que en vez de implementar y controlar todos los elementos, simplemente se paga un precio por ellos.

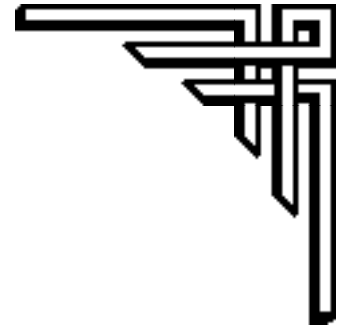
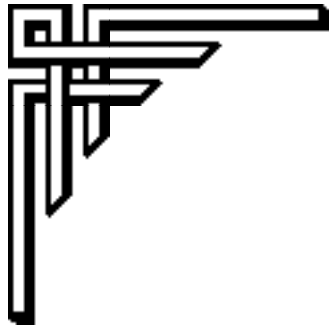
Recapitulación.

EDI presenta una serie de ventajas a las empresas que han impulsado su desarrollo y extensión. Sin embargo, sabemos que sólo un 2% de las empresas americanas se han atrevido a implantarlo - o mejor aún, tras una fase de análisis, se ha descubierto su inviabilidad en algunos casos, mientras que en otros ni siquiera se plantea su uso.

Las soluciones, por lo general, se universalizan rápidamente en cuanto se descubren los beneficios que aportan, y aunque en el caso de EDI se ha producido este fenómeno, en cierta medida se ha visto frenado o limitado por el escaso número de empresas que lo adoptan.

El comercio electrónico es un campo en el que las empresas están predispuestas a participar por la automatización de tareas y reducción de costes que les supone, fundamentalmente. ¿ Por qué resulta, entonces, que hay tan pocas empresas que implementen EDI ?. ¿ Cómo hacer que el resto de empresas implanten soluciones B2B o B2C ?.

Es necesario crear herramientas con mecanismos que permitan su adaptabilidad a todo tipo de sectores, flexibilidad en su uso y, sobre todo, facilidad y reducidos costes en su implantación. EDI no responde a todas estas demandas, por lo que no se puede convertir en la base del comercio electrónico mundial. Se necesitan abrir nuevos caminos para el logro de los objetivos anteriores, y eso es lo que trata de hacer XML desde hace varios años, como veremos en los capítulos posteriores. EDI no va a desaparecer, porque las empresas podrán seguir amortizando sus inversiones, pero tenderá a ser sustituido por otras soluciones.



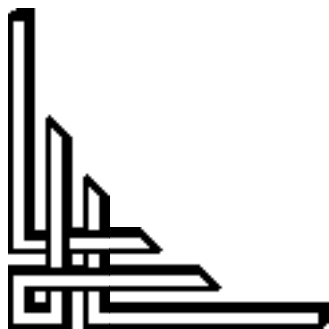
X

Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

XML.
EL LENGUAJE.

M

L



XML son las siglas de *eXtensible Markup Language*. Se está introduciendo en todos los sectores de la informática, y la prueba de ello es que en la mayoría de anuncios y boletines donde se incluye, ni siquiera se explica lo que es. Se empieza a asumir que es algo conocido.

Del mismo modo que HTML convirtió a Internet en la mayor biblioteca del mundo, XML se está erigiendo como el centro de comercio y finanzas universal. Todo ello a través de un proceso de 'dinamización' donde los sitios Web ya no sólo se dedican a mostrar páginas, sino que las generan automáticamente a partir de bases de datos de la empresa e incluso de otras páginas web.

Hay una diferencia fundamental entre HTML y XML. Mientras que el primero se basa en la representación, el segundo se basa en el contenido. Son dos características que es importante saber diferenciar. HTML es un lenguaje cuyas etiquetas indican la forma en que se debe representar - a través de un navegador de Internet - la información que contiene un fichero o documento. Sin embargo, XML nos indica el tipo de información contenida en el mismo. La representación es independiente del contenido, y también puede informar sobre cómo realizarla. Pero la esencia y el aspecto fundamental de este metalenguaje es que sus etiquetas nos indican el tipo de información contenida en el documento, dándonos total libertad para representarla según nos interese. XML representa la metainformación. Es decir, es la información que nos permite clasificar los datos.

Jean Paoli, co-editor de la recomendación sobre XML, junto con Charles F. Goldfarb, una de las mayores autoridades en cuanto a la información estructurada, intentaron crear un estándar abierto, simple, flexible y reconocible por los humanos. Y que pueda viajar entre servidores, clientes y aplicaciones - así como la mezcla de los tres - de manera que se permita una comunicación universal. Aunque suene utópico, 'eso' es XML.

Orígenes y objetivos de XML.

Fue en el año 96 cuando la *World Wide Web Consortium* (W3C - <http://www.w3.org>) inició el proceso de diseño de un lenguaje de marcado extensible. Utilizando como base el SGML - *Standard Generalized Markup Language* - se creó un subconjunto del mismo, mucho más fácil de utilizar. En febrero de 1998 se convirtió en una recomendación de la W3C.

Ha recibido la descripción de 'el ASCII de la Web', por tratarse de una forma simple y estándar de presentar datos en forma de texto. Es tan fácil de utilizar como crear una estructura de datos arbitraria en su lenguaje de programación favorito, y es independiente de cualquier tipo de programa o plataforma. Sus características principales son:

1. Es abierto. Se puede intercambiar datos con otros usuarios o programas independientemente de la plataforma.
2. Es autodescriptivo, con lo que se convierte en una solución excelente para comercio electrónico y soluciones Extranet.
3. Se pueden compartir datos entre programas sin coordinación previa. El documento puede estructurarse dinámicamente.

Para trabajar con documentos XML, la W3C ha estandarizado una API - *Application Programming Interface*. La comunidad de usuarios y desarrolladores ha creado otra alternativa, basada en eventos. Un aspecto importante de su definición es que está basado en el Conjunto de Caracteres Universal (UCS), equivalente al conocido Unicode.

Aunque en realidad XML es en sí mismo datos, no ha dejado de lado la presentación de los mismos. En este sentido se han creado otro tipo de documentos complementarios, las llamadas hojas de estilo, donde el diseñador especifica los estilos de formateo y las reglas que determinan en qué momento se deben aplicar estos estilos.

Naturaleza de XML.

A continuación vamos a ver el modo en el que XML nos permite alcanzar los objetivos descritos anteriormente. Como indica su definición, es un lenguaje de marcado, con lo que son las *etiquetas* del mismo las que nos comunican el tipo de información contenida entre las mismas. Además, podemos extender arbitrariamente el lenguaje para alcanzar nuestros propósitos, y como el mecanismo de extensión es estándar, cualquier programador o máquina que lea nuestros datos será capaz de reconocerlos.

Veamos un ejemplo del modo en que XML podría describir la información referente a una persona y comparémoslos con HTML.

XML	HTML
<pre><Persona> <Nombre>Pedro</Nombre> <Apellidos> <Apellido1>López</Apellido1> <Apellido2>Pérez</Apellido2> </Apellidos> <Edad>25</Edad> </Persona></pre>	<pre><TABLE> <TR><TD>Pedro</TD></TR> <TR> <TD>López</TD> <TD>Pérez</TD> </TR> <TR><TD>25</TD></TR> </TABLE></pre>

De los cuadros anteriores podemos deducir que estamos describiendo a una persona en función de su nombre completo y de su edad. Sin embargo, es evidente que mientras que en el formato de HTML tenemos que deducir que se trata del nombre, apellidos y edad de la persona, en el documento XML se nos indica de qué *atributo* de la persona estamos dando el valor. Con esto, acabamos de definir la M de XML. Se trata del Marcado.

Definamos ahora la X o extensibilidad. ¿Qué ocurre si una organización desea, además del nombre completo y de la edad, el color de ojos de la persona para identificarla?. En tal caso, podemos extender el lenguaje con nuevas etiquetas como, por ejemplo, `<ojos></ojos>`, entre las que indicaríamos el valor del atributo. Tan sencillo y tan potente como flexible y adaptable.

Como hemos visto, la versatilidad de XML supera con grandes creces las limitaciones de HTML. Sin embargo, para poder utilizar el lenguaje de marcado, es evidente que es necesario haberlo definido con anterioridad. Cada empresa puede definir un lenguaje propio de una manera muy sencilla, aunque la gran utilidad de XML reside en la definición de lenguajes comunes. Son los llamados *vocabularios*, cuya puesta en común facilitará la comunicación.

‘Un vocabulario es una descripción de datos que es utilizada como medio para intercambio de información, a menudo dentro de un dominio específico de una actividad humana’ [ProXML, 2000].

Desde la recomendación de XML por parte de la W3C, se han creado una gran cantidad de vocabularios. A su vez, hay muchos que siguen desarrollándose, por la dificultad que implica agrupar la mayor cantidad posible de información común a los miembros de una disciplina concreta. Uno de los elementos que más ralentiza la generación de nuevos vocabularios es su flexibilidad. No hay que olvidar que cualquier vocabulario debe tener implícitas las normas para su futura extensión sin violar la corrección y validez del documento. Algunos de los vocabularios más conocidos son:

1. Chemical Markup Language (CML - <http://www.xml-cml.org>).
[creación de documentos estructurados para uso profesional].
2. Electronic Business XML (ebXML - <http://www.ebxml.org>).
[comercio electrónico mediante el uso de XML].
3. Voice XML (VXML - <http://www.voicexml.org>).
[aplicaciones de telefonía basadas en voz].
4. Wireless XML (WML).
[XML para servicios inalámbricos con tecnología WAP].

Características de XML. El DTD.

‘Cada vez que un sistema cerrado se abre, empieza a interactuar más directamente con otros sistemas existentes, y por lo tanto adquiere el valor acumulado de estos sistemas’.

La frase anterior pertenece a Kevin Kelly, director ejecutivo de Wired - una de las mayores publicaciones científicas americanas. Es extremadamente acertada, aunque hay que matizar algunos puntos. ¿ Cómo se abre un sistema ?. ¿ Cómo interactúa con los demás ?.

Para permitir esta comunicación, debemos asegurar que la sintaxis manejada por los documentos XML es la correcta, y que da lugar, al menos, a *documentos bien formados*. En este momento debemos recordar lo descrito anteriormente sobre los vocabularios, pero debemos ir más lejos todavía. Necesitamos herramientas que se encarguen de verificar esta sintaxis, con lo que nuevos elementos van a formar parte del juego: *los analizadores sintácticos*. Ellos permitirán a nuestras aplicaciones manipular los documentos, informando de los errores en caso de que la sintaxis no sea la correcta. A continuación se presenta un ejemplo de un documento bien formado.

```
<?xml version="1.0"?>
<almacen no_tienda="100">
  <compra cliente="50">
    <producto>
      <marca>Choleck</marca>
      <descripcion>Batido de Vainilla, 1 litro</descripcion>
      <precio>
        <moneda>Euro</moneda>
        <valor>0.50</valor>
      </precio>
    </producto>
    <fecha>
      <dia>20</dia>
      <mes>Mayo</mes>
      <año>2002</año>
    </fecha>
  </compra>
</almacen>
```

Una de las características de XML era y es que cualquier persona puede crearse su propio vocabulario. Pero entonces, ¿ cómo será capaz de usarlo con la seguridad de que es correcto y de que los demás lo entiendan ?. Además, de algún modo será necesario especificar la forma en que se debe construir un documento.

DTD significa *Document Type Definition* y es la respuesta de XML a las preguntas formuladas con anterioridad. Es el primer mecanismo ideado para capturar la estructura de un vocabulario determinado y las reglas que el diseñador ha integrado en su especificación. Con esta herramienta, estamos en disposición de poder validar los documentos que compartan las aplicaciones así como descubrir nuevos vocabularios en tiempo de ejecución. Toda una ventaja y, hasta el la aparición de los *schemas*, la única forma oficial de representar estructuras en XML.

Anteriormente hemos presentado un documento XML relativo a la compra de un cliente en un almacén. A continuación se muestra la estructura o DTD generado para transmitir este tipo de información. Y en la página siguiente se describe el significado de cada etiqueta.

```
<?xml version="1.0"?>
<!DOCTYPE almacen[

<!ELEMENT almacen (compra?) >
<!ATTLIST almacen no_tienda CDATA #IMPLIED >

<!ELEMENT compra (producto*, fecha*) >
<!ATTLIST compra cliente CDATA #IMPLIED >

<!ELEMENT producto (marca, descripcion, precio)+ >
<!ELEMENT      marca (#PCDATA) >
<!ELEMENT      descripcion (#PCDATA) >
<!ELEMENT precio (moneda, valor)+ >
<!ELEMENT      moneda (#PCDATA) >
<!ELEMENT      valor (#PCDATA) >

<!ELEMENT fecha (dia, mes, año)+ >
<!ELEMENT      dia (#PCDATA) >
<!ELEMENT      mes (#PCDATA) >
<!ELEMENT      año (#PCDATA) >

]>
```

1. <!DOCTYPE elemento-raíz [declaracion del documento]>.

Es el nombre del elemento raíz que va a contener las definiciones de la declaración de tipos del documento.

2. <!ELEMENT nombre contenido >.

Asocia el *contenido* descrito al *nombre* que le precede. Dicho de otra manera, cada valor de *contenido* incluido en la etiqueta, está un nivel por debajo de *nombre* en la jerarquía de elementos.

Algunos de los valores que se pueden asignar son:

- **?** : *es un elemento opcional.*
- ***** : *obligatoriamente habrá 0 o más elementos de este tipo.*
- **+** : *obligatoriamente habrá 1 o más elementos de este tipo.*
- **#PCDATA** : *secuencia de caracteres.*

3. <!ATTLIST elemento atributo tipo defecto ... >.

Relaciona la lista de atributos de un elemento dado con el propio elemento al que pertenecen. El tipo fundamental es el CDATA, que es el valor por defecto y que significa que cualquier carácter es válido. Para los valores por defecto, tenemos los siguientes:

- **#REQUIRED** : *debe estar contenido explícitamente.*
- **#IMPLIED** : *es un valor opcional.*
- **#FIXED** : *es el único valor válido, aunque se indique otro.*
- **"valor"** : *si no se especifica otro, es el valor por defecto.*

Como hemos visto hasta el momento, conociendo el DTD de un determinado tipo de documento, somos capaces de descifrarlo o de crear nuevos documentos de este tipo. Sin embargo, ¿ por qué íbamos a quererlo hacer nosotros ? . A lo largo del texto hemos insistido en la ventaja de XML como medio de comunicación universal entre máquinas. No vayamos entonces a modificar su función. El siguiente paso nos conduce a los analizadores sintácticos de los que hemos hablado anteriormente. Ellos serán los encargados de descifrar un DTD y reconocer si el documento está bien formado o no.

Namespaces & Schemas.

A medida que se aprende más sobre XML y las ventajas que ofrece a las aplicaciones, se crean documentos y vocabularios cada vez más complicados. Se introduce en paralelo un afán por realizar de forma automática las tareas necesarias para descubrir la estructura de un documento. Llegados a este punto, se encuentran una serie de limitaciones, algunas de las cuales se pueden resolver mediante el uso de *Namespaces* y *Schemas*.

La manera en la que se describe la estructura de un documento es su DTD. Sin embargo, hay que reconocer que presenta algunas limitaciones. En algunos casos es debido a que no ha sido desarrollado utilizando la sintaxis propia de XML - lo correcto hubiera sido crear un meta DTD para la generación de DTDs inferiores. En otros, puede deberse al hecho de que se mezcla con la propia especificación de XML, con lo que introduce ciertos elementos de confusión.

Hay una gran cantidad de autores que critican el uso de los DTDs, e incluso algunas herramientas desarrolladas con XML han sido definidas mediante el uso de Schemas. Veamos los problemas que presenta el uso de un DTD:

1. No hay restricciones en cuanto a caracteres válidos.
2. Los modelos para los valores de los atributos son muy simples.
3. No soporta los Namespaces, definidos más tarde.
4. Poca modularidad y reusabilidad.
5. No hay soporte para la evolución, extensión o herencia.
6. Falta de control sobre los espacios - como caracteres.
7. No permite la documentación propia.
8. Mecanismo de identificadores de atributos muy simple.
9. No hay valores por defecto para elementos.
10. No permite la especificación parcial.
11. Los valores por defecto están incluidos en la declaración.

Se trata de un conjunto de inconvenientes tan extenso como concreto en su formulación. A continuación veremos el modo en que se han resuelto algunos de ellos.

Namespaces

Nacen como recomendación de la W3C en enero de 1999, y vienen a resolver los problemas derivados de la ambigüedad y colisión en los nombres. Según la W3C se trata de...

“... una colección de nombres, identificados por una referencia URI, que son usados en documentos XML como tipos de elementos y nombres de atributos.”

La palabra reservada utilizada para reconocer la declaración de un namespace es *xmlns*. Declararemos un *namespace* para que sea el valor por defecto cierto ámbito, evitando así la declaración del prefijo.

Schemas

Un concepto que es necesario tener claro, es que un DTD es un tipo de *schema*. En realidad, inicialmente no era una forma oficial de declarar la estructura de un documento, aunque se utilizaban bastante a menudo. Se trata de una sustitución de un DTD por su equivalente escrito en la sintaxis propia de XML.

En Mayo de 2001 surgió la recomendación de la W3C para la definición de *schemas*. Está basado en dos partes, las estructuras y los tipos de datos, y resuelve las carencias que presentaban los DTD de la siguiente forma:

1. Sintaxis XML y uso de *namespaces*.
2. Declaraciones de tipos como en la programación orientada a objetos, locales, globales y con modularización.
3. Restricciones de cardinalidad, valores nulos y por defecto.
4. Auto-documentación estructurada.

Estas son las características más importantes de una especificación de más de 200 páginas. Actualmente, el uso los *schemas* está reemplazando el de los DTDs, pero en estos últimos se cuenta con mayor experiencia que en los primeros, con lo que su introducción es relativamente lenta.

‘Document Object Model’ y ‘Simple API for XML’

Document Object Model se puede abreviar como DOM. De hecho, es la forma en la que se conoce esta especificación creada por la W3C. Existen 3 documentos oficiales - o tres niveles - referentes a la manera en que se debe utilizar la información que se extraiga de los documentos, así como la forma en que se debe extraer. Pero primero será necesario saber el modo en que opera.

Cuando se utiliza DOM para manipular un fichero, lo primero que hace es descomponerlo en partes más pequeñas : elementos, atributos, comentarios, etc. Una vez se ha llevado a cabo este análisis, entonces se crea, en memoria, una representación del documento XML en forma de nodos de un árbol. De esta manera los elementos se pueden acceder fácilmente y proceder a su modificación en caso necesario. En realidad, cada elemento del documento se convierte en un nodo.

Existen cuatro motivos fundamentales para el uso de DOM:

1. Garantiza la corrección de la sintaxis de la gramática y que el documento está bien formado.
2. Abstrae el contenido respecto a la gramática.
3. Simplifica la manipulación interna del documento.
4. Su estructura es muy similar a la de las bases de datos relacionales, lo que facilita la transmisión de información entre ficheros y sistemas gestores de bases de datos.

El primer navegador en incluir de forma nativa el DOM fue Microsoft Internet Explorer en su versión 5. Posteriormente Netscape permitió su uso a través de elementos ActiveX y con código Java o JavaScript en el cliente, para incluirlo definitivamente en versión 6 del Navigator. Tanto los elementos ActiveX como las librerías se pueden descargar gratuitamente en los sitios de Microsoft y Java.

Sin embargo, no todo son ventajas. ¿ Qué ocurre si tenemos un fichero XML de 10 Gb.?. Es evidente que no podemos cargarlo en memoria, tal y como se describe en la especificación. Para resolver este problema y otros derivados de DOM surge una forma alternativa de tratamiento de documentos : la *Simple API for XML*, conocida como SAX.

Fundamentalmente, SAX es una interfaz tipo Java, y para utilizarla con este tipo de aplicaciones es necesario descargar e incluir las clases necesarias para su desarrollo. La ventaja primordial que introduce respecto a DOM es que funciona a base de *eventos*, con lo que resuelve problemas como el mencionado anteriormente sobre el tamaño de los documentos. Los eventos que produce son los siguientes:

1. Inicio de documento y fin de documento.
2. Inicio de una etiqueta y fin de una etiqueta.
3. Al encontrar datos de tipo carácter.
4. Al encontrar una instrucción de procesamiento.

VENTAJAS

1. Permite analizar ficheros de cualquier tamaño.
2. Es útil en la construcción de estructuras de datos propias, porque no es necesario guardarlas en memoria sino recorrerlas e informar mediante eventos al usuario.
3. Cuando sólo se quiere trabajar con un reducido conjunto de la información, es más fácil localizarla e ignorar el resto.
4. Es simple y es rápido.

INCONVENIENTES

1. No permite acceso aleatorio al documento.
2. Las búsquedas pueden ser difíciles de implementar.
3. No se informa sobre los comentarios del documento, porque se asume que van dirigidos al autor y no al lector del mismo.
4. El orden de los atributos se trata de forma indiferente.
5. Es de sólo lectura, a diferencia de DOM que conserva el documento en memoria y permite su modificación.

En cuanto a su relación con los navegadores, los actuales no soportan todavía la especificación de SAX, y la mejor manera de introducirlo son los Applets.

En este punto, ya disponemos de las herramientas suficientes para crear, tratar y modelar documentos en XML. Además, puesto que disponemos de los analizadores sintácticos que se encargan de la validación del documento, las tareas son muy simples.

Algunas de las herramientas que se pueden utilizar para trabajar con XML se citan a continuación. La mayoría son libres, aunque otras, como el caso del editor XMLSpy, son de pago.

1. Analizadores sintácticos.

- Xerces. *Pertenece a AlphaWorks y soporta DOM y SAX.*
- Expat. *Escrito en C y exportado a otros lenguajes.*

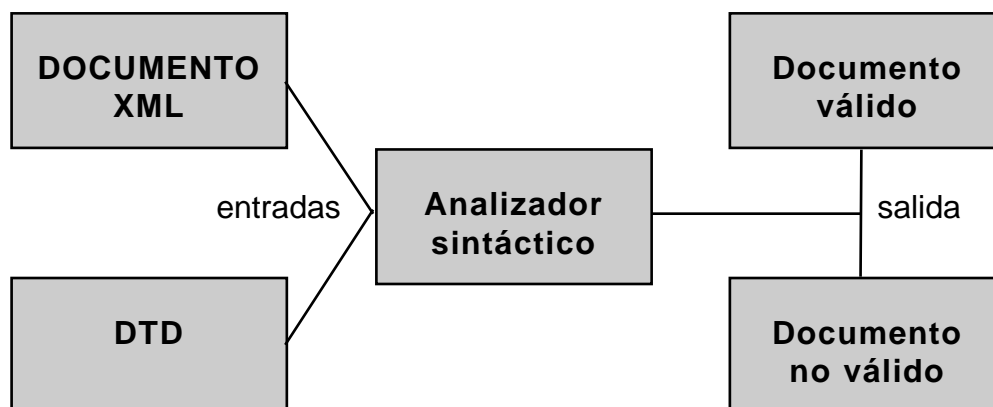
2. Editores.

- Xeena. *También de AlphaWorks, desarrollado en Java.*
- XMLSpy. *Muy popular pero no es gratuito.*

3. Navegadores.

- Apache XML (<http://xml.apache.org>).
- Netscape Navigator 6 e Internet Explorer 5.
- Amaya. *Es a su vez navegador y editor de XML.*

Finalmente, veamos un pequeño esquema de cómo se pueden interrelacionar todos los elementos descritos hasta el momento.

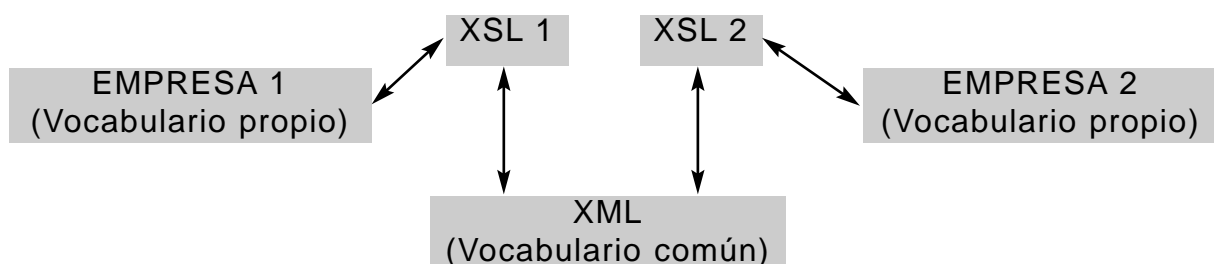


Hojas de estilo.

Una de las técnicas más potentes de XML es la transformación. A pesar de haber afirmado en repetidas ocasiones la importancia de separar el contenido de la presentación, hay situaciones en que necesitaremos hacer uso de los dos. XSL - *eXtensible Stylesheet Language* - es la herramienta adecuada para ello.

Las situaciones donde se requiere la transformación se pueden dar muy a menudo: en cualquier tipo de comercio electrónico, a la hora de enviar información a un navegador o al intercambiar datos con sistemas propietarios. En estos casos, es imprescindible adaptar el lenguaje común o vocabulario a las necesidades del sistema.

La situación más clara se puede ver en el contexto de un intercambio de información entre dos empresas. Cada una de ellas posee su propio vocabulario interno, pero además utilizan un vocabulario común para el intercambio de información:



A través de estas transformaciones, las empresas pueden seguir trabajando con su propio sistema de información y el conocimiento adquirido a lo largo de su experiencia. Lo único que necesitarán hacer para comunicarse con el resto de empresas es una buena adaptación de las hojas de estilo.

Teniendo en cuenta que los navegadores dominan y dominarán durante un largo período la comunicación, la mayoría de transformaciones se realizan para adaptar la información y presentarla en formato HTML. Posiblemente es la transformación más común, aunque no la única ni la más importante.

XML en un futuro proximo y la Web.

El Doctor Nic Fulton, líder del diseño y arquitectura de XML en la agencia Reuters, afirma lo siguiente:

“XML es un estándar muy importante que ha emergido en Internet hace algunos años, y parece haber alcanzado un punto crítico. Nosotros llevamos 150 años en el mundo de la información, y siempre hemos adoptado las últimas tecnologías, lo que es exactamente XML”.

Algunos expertos en documentación y gestión del conocimiento han empezado a hablar de los llamados *Topic Maps*. Se trata de un paso más en la evolución de la Web, donde se organizarían todos los contenidos existentes (documentos HTML, XML, imágenes...) alrededor de *Topics* o temas concretos. De este modo se podría navegar fácilmente entre todos los elementos de un mismo tema.

Pero nuestro objetivo no es obtener una visión de futuro de la Web, sino analizar su situación actual. Ciertamente es que, al escrutar los distintos caminos que puede tomar la evolución de Internet, es positivo que se cuente con XML. Pero aún más cierto es que todavía no se han producido grandes cambios por la irrupción del propio XML. Paso a paso, vamos a descubrir algunas de las alteraciones que está sufriendo la World Wide Web.

Arquitecturas Clásicas

La primera generación de arquitecturas Web seguían la estructuración tradicional dividida en *cliente* y *servidor*. Contrariamente al espíritu de Internet - un sistema abierto, flexible y de protocolos muy simples -, se crearon sistemas demasiado complicados y poco transparentes. Además, se añadió una tercera capa en esta estructura que hacía referencia a una base de datos. Mientras el cliente era, en general, un navegador, el servidor recogía los datos de los parámetros enviados por distintos métodos : GET o POST. Los procesaba y posteriormente enviaba el resultado.

El funcionamiento de la arquitectura era suficientemente bueno. En realidad lo es, porque se sigue utilizando. Pero tiene limitaciones:

1. Estamos restringidos a un navegador cliente.
2. Nadie crea páginas de servidor para comunicarse con otras aplicaciones o servicios. En realidad, los mismos encargados de la parte de servidor deben implementar el cliente.
3. El contenido es siempre HTML.
4. Si se desean contenidos adicionales, es necesario añadir una nueva capa al servidor.
5. Es necesario sincronizar al cliente y al servidor.

Arquitecturas XML

Vamos a pasar de la computación cliente-servidor a un modelo multi-capa. Pero, ¿ qué es multi-capa ? . Si entendemos que una capa es un servidor, tenemos modelos donde interactúan gran cantidad de servidores. Si una capa es uno de los elementos (base de datos, por ejemplo) por el que pasamos antes de devolver la información al cliente, podemos pasar por tantos como queramos.

Todo se basa en un principio al que hemos hecho referencia en varias ocasiones. Un documento XML no debe ser conocido por el receptor del mismo, porque puede buscar su definición para entenderlo. Por lo tanto, un cliente puede enviar una petición en XML - sin tener que ajustarse a las normas del servidor. El servidor descifra y valida los parámetros de la petición que ha recibido, y puede obtener la información solicitada de tantas fuentes como considere necesario. Podrá consultarlas y, finalmente, unificarlas todas en una sola por medio de un fichero XML que devolverá al cliente.

¿ Y por qué XML ? . Es verdad, en realidad podrá enviarlo tal y como se lo haya solicitado el cliente, porque podrá transformar esa información etiquetada como XML en HTML o cualquier otro formato que se desee. El único elemento necesario será una hoja de estilo que podrá haber sido definida tanto por el cliente como por el propio servidor.

XLink, XPointer y XPath.

Al analizar las posibles variaciones que puede sufrir la Web con la introducción de XML, es necesario prestar una especial atención a los enlaces. ¿ Enlaces ?.

Cuando decimos que tanto *Ramón Pérez* como *Carlos Pérez* comparten el mismo apellido - quizás sean hermanos, o primos -, estamos creando un enlace entre ellos. Aquí se establece una ruptura de conceptos importante con el significado de un enlace en HTML. Un enlace, al ser solicitado, nos conduce directamente al destino que tiene indicado. Pero nuestro concepto de enlace es diferente. Simplemente nos relaciona dos elementos, personas, documentos...

XLink es el mecanismo de la W3C para enlazar con otros elementos desde un documento XML. Y tiene la particularidad de que coincide con nuestro concepto de enlace. No necesariamente nos vamos a dirigir al nuevo enlace, sino que sabiendo la relación proporcionada por el mismo decidiremos la manera en la que actuamos: el momento - al cargar el primer documento, por ejemplo -, la forma - dentro o separado del documento que estamos analizando -, el título que recibe, etc.

Curiosamente, además de *Ramón Pérez* y de *Carlos Pérez*, también conocemos a *Cármén Pérez*, a *Miguel Pérez* y a *Manuel Pérez*. El crecimiento de la familia ha sido importante - si es que definitivamente son familia - pero el concepto de enlace debe ser el mismo. Si todos siguen teniendo el mismo apellido, entonces es evidente que debe guardarse el mismo tipo de relación entre los dos primeros que entre el conjunto de los cinco.

Siendo previsor, XML ha sido capaz de dar respuesta a esta situación con lo que denomina enlaces extendidos. Es decir, existe más de un posible enlace con la misma referencia. En tal caso, se procederá como se ha explicado anteriormente. Se decidirá la forma, el momento, etc. en que se desea mostrar el enlace, y se llevará a cabo.

Este tipo de gestión de enlaces es muy eficiente, aunque no se ha puesto demasiado en práctica por tratarse hasta hace poco tiempo de un simple borrador. En estos momentos, ya debe estar funcionando.

Todo el procesado que se realiza en XML depende del concepto de direccionamiento. Para tratar un documento, en primer lugar es necesario encontrarlo o recogerlo de algún lugar y para utilizar los datos es necesario localizarlos. En la Web existen las conocidas URIs - *Uniform Resource Identifiers* - y las omnipresentes URLs - equivalente a las anteriores excepto en la L de *Locator*.

XPath - *XML Path Language* - es la especificación que vamos a utilizar para hacer referencia a datos, elementos, atributos y otro tipo de información que puede surgir en un documento. Su utilidad se basa en que no es necesario duplicar la información. Veamos algunos ejemplos:

1. Un párrafo de un texto puede hacer referencia a otro párrafo. En vez de escribir el valor de la referencia cruzada, podemos escribir la referencia de modo que el valor que se extraiga dinámicamente.
2. Si recibimos una orden de compra de un sistema, para hacer las cuentas deberíamos saber el precio de lo que estamos comprando. O tener una referencia al documento que contiene el precio de lo que estamos comprando.

Viendo la descripción de ambos casos, llegamos a la conclusión de que principalmente se evita la redundancia a la hora de transmitir información. Esta última estará presente en un único sitio desde donde se recuperará. Pero XPath hemos dicho que es, en cierto modo, equivalente a una URL. ¿Cómo localizamos, entonces, el precio de un artículo o el capítulo correspondiente del libro ?.

La respuesta es XPointer. Mientras que XPath se utiliza como mecanismo de direccionamiento, XPointer ofrece una forma estándar de utilizar dicho mecanismo en las referencias. Su capacidad de maniobra es muy importante, y por ello se ha definido de tal manera que pueda apuntar a documentos XML de fuentes distintas, como puede ser HTML.

Bases de datos XML.

Goethe dijo una vez: *“El que no sabe llevar su contabilidad por espacio de tres mil años se queda como un ignorante en la oscuridad y sólo vive al día”*.

Tomando al pie de la letra su proposición, podríamos contestarle que nadie ha podido vivir de la informática durante tres mil años. ¡¡ Los informáticos acabamos de llegar al mundo !!. Puesto que estamos limitados en el tiempo, podemos utilizar mejor este otro proverbio oriental que dice: *“Cuanto más cambian las cosas, más vuelven a parecerse”*.

Cada vez que surge una nueva tecnología aparecen nuevas herramientas, métodos de integración y, sobre todo, cambios en los sistemas de bases de datos. La adopción de XML puede ser muy similar a la que supuso la aparición de SQL en el mercado alrededor de 1980. El resultado no será la sustitución de una tecnología precedente - SQL no impidió que se siguieran efectuando las transacciones que llevaban a cabo los mainframes -, pero se espera una paulatina y progresiva adopción de la misma.

Las bases de datos SQL tienen aspectos importantes como el almacenamiento de la información, transacciones, recuperación ante errores y soporte a la toma de decisiones. XML aparece en el segmento para complementar el sector con gestión de contenido y publicación de la información. Compañías del tamaño de Microsoft y Oracle ya lo han entendido así, y a pesar de no ofrecer bases de datos exclusivamente basadas en XML, sí que ofrecen mecanismos de extensión.

Uno de los empujones definitivos para que la industria empiece a desarrollar bases de datos nativas en XML va a provenir de sus propios clientes. A medida que las organizaciones integren en sus procesos de negocio el nuevo lenguaje, exigirán a las compañías una mayor facilidad de tratamiento para almacenar sus documentos en las bases de datos. Será como una bola de nieve: cuanto mayor sea el uso, crecerá la demanda y aumentará el nivel de exigencia.

Bases de Datos Nativas

La primera base de datos nativa y comercial implementada plenamente para el uso de XML es *Tamino*, de Software AG. El producto se descompone en dos aspectos principales:

1. Almacenamiento de información para Internet.
2. Almacenamiento de datos remotos de la empresa.

Tiene una ventaja fundamental, y es que no necesita de la lógica adicional de la que se tienen que servir los sistemas gestores de bases de datos convencionales.

Bases de Datos Mixtas

Hace ya algunos años la programación orientada a objetos demostró tener un gran número de ventajas sobre la programación tradicional, pero también surgieron pequeños inconvenientes. Del mismo modo que hoy nacen bases de datos con extensiones XML, en aquellos momentos la industria decidió que lo mejor era adaptar los sistemas existentes y no volcarse en la orientación a objetos.

Esta situación, al vivirse de nuevo, plantea los mismos inconvenientes que surgieron en el pasado - quizás Goethe tenía más razón de lo que cabía pensar en un principio. Los fundamentales se derivan de la convivencia de los dos sistemas, porque mientras que por un lado se deben ofrecer capacidades de recuperación y almacén de datos, por otro están obligadas a mantenerse compatibles con viejas aplicaciones de clientes y/o usuarios.

Las comparaciones son odiosas. Tan cierto como que sin las mismas la informática no avanzaría, con lo que se convierten en necesarias y reveladoras. Las bases de datos nativas en XML están un orden de magnitud por encima de las soluciones mixtas.

Data Warehousing

Se trata de los sistemas más complejos y difíciles de implementar para una organización, puesto que son la base de algunas herramientas como los sistemas de ayuda a la toma de decisiones o los sistemas de información para ejecutivos. Su objetivo es doble:

1. Agilizar el complejo procesamiento de consultas del sistema para obtener resultados óptimos - en el tiempo.
2. Ofrecer información relacionada de distintas fuentes de datos o información para el uso por parte de la gerencia.

Los sistemas de Data Warehousing o almacenes de datos se han desarrollado a lo largo de los últimos 15 años, aproximadamente. Algunos de ellos no dieron el resultado esperado, debido a las dificultades de integración entre los distintos sistemas gestores de bases de datos. Cada uno llevaba a cabo su propia optimización de SQL para ofrecer mejores servicios y, posteriormente, cuando se necesitaba agrupar información de distintos orígenes de datos, se producían la mayor parte de las incompatibilidades.

Pero todo esto debe cambiar. No hace falta decir que la solución es XML porque ofrecerá una pasarela única para la transmisión de información. Además, se fomentará el uso del comercio electrónico, lo que supondrá mayores cantidades de datos en los sistemas de las empresas. XML aportará el equilibrio necesario para el buen funcionamiento de todo el conjunto.

XML en el Comercio Electrónico.

Para resolver el problema del intercambio de datos entre aplicaciones, XML está siendo utilizado cada vez por un mayor número de organizaciones que desarrollan actividades de comercio electrónico. En particular, el sector de negocio del B2B - *Business to Business* - o negocio entre empresas, está en auge.

Tal y como afirman la mayor parte de los gerentes de pequeñas y medianas empresas, *“el intercambio de datos mediante EDI es demasiado complejo y, sobre todo, demasiado caro. Con la aparición de XML se abre un nuevo horizonte de comunicación entre empresas”*.

Para hacerlo posible, algunas iniciativas se han adelantado en el tiempo. A continuación se van a describir OFX y OTP.

OFX - Open Financial eXchange.

Es una iniciativa de Checkfree, Intuit y Microsoft para estandarizar el formato de datos financieros que necesitan transmitirse entre distintas partes. Incluye las siguientes áreas:

1. Transacciones de consulta de cuentas.
2. Pagos con tarjeta de crédito.
3. Compra y venta de acciones.
4. Transferencia de fondos.

No se basa directamente en XML, sino en su predecesor SGML, lo que le convierte en uno de los intentos pioneros para obtener un lenguaje estándar de comunicación. Se basa en las siguientes características:

1. Es abierto y extensible.
2. Es independiente del cliente.
3. Es robusto y seguro.
4. Permite procesamiento en segundo plano.
5. Es internacional.

OTP - Open Trade Protocol.

Responde al deseo de ventas al por menor de un conjunto de empresas : AT&T, Hewlett-Packard, Mastercard, Sun o el banco Wells Fargo. La idea es que se presenten al usuario herramientas con una interfaz familiar, consistente y de fácil manejo, que cubra las actividades citadas a continuación:

1. Ofertas de venta.
2. Acuerdos de compra.
3. Pagos.
4. Transferencia de bienes y servicios.
5. Distribución.
6. Facturación.
7. Resolución de problemas.

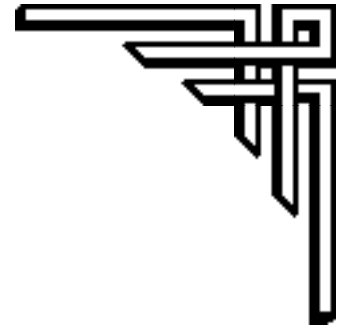
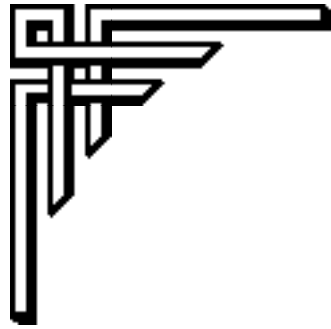
En ambos casos, se trata de iniciativas cuyo fin era el de cubrir los nichos de mercado que ha dejado vacíos EDI. Si tenemos en cuenta que tan sólo el 2% de las empresas utilizan EDI en los Estados Unidos; y que en Europa el porcentaje es todavía menor, existe un amplio campo de actuación donde integrar las nuevas soluciones de comercio electrónico. De este modo se podrán obtener los beneficios que los grandes gurús del sector han pregonado durante años, pero cuya adaptación ha resultado tan complicada en la vida real.

Más adelante descubriremos los protocolos de comercio electrónico en los que vamos a centrar el presente proyecto. Se trata de ebXML - *electronic business XML* -, de cXML - *commerce with XML* - y de BizTalk, sistema propietario de Microsoft. En todos los casos se estudiará detalladamente la manera en la que pueden suplir o colaborar con EDI para facilitar el negocio electrónico y veremos ejemplos prácticos de estas situaciones.

Conclusiones.

A lo largo del presente capítulo, hemos visto la manera de trabajar con XML, su funcionalidad y las ventajas derivadas de su uso. Aún a pesar de ello, es posible que queden algunas dudas por resolver. Para ello, y basándonos en las preguntas más frecuentes que suelen hacer los ejecutivos con respecto a XML, se ha elaborado la siguiente lista de características a modo de conclusión.

1. XML no pertenece a nadie. Ha sido desarrollado por la World Wide Web Consortium y permite que cualquiera pueda elaborar su propio lenguaje en cuestión de días.
2. Tanto los creadores de software como los usuarios están aplicando XML. No es una tecnología cerrada de una industria, sino que está abierta a todos los sectores.
3. XML no pasará de moda. Sería lo mismo que pensar que el español o el inglés pueden desaparecer en varios años.
4. Las mayores ventajas de XML se presentan en el comercio electrónico, donde las pequeñas y medianas empresas apenas han tenido cabida hasta el momento, y ahora buscan su espacio.
5. Cuanto más tarde se inicie en XML, mayor retraso acumulará sobre sus competidores. El mundo ya ha empezado a moverse alrededor de XML y no involucrarse es como cerrar los ojos a la realidad actual.



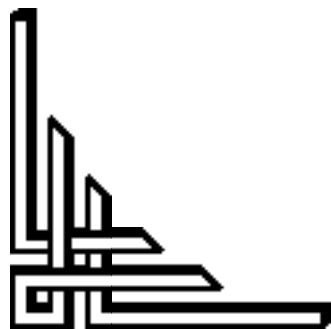
X

Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

**VENTAJAS E
INCONVENIENTES
DE XML.**

M

L



En el capítulo anterior hemos descubierto la funcionalidad que puede ofrecer XML. En forma de recorrido por sus diferentes aspectos, han ido apareciendo las ventajas de trabajar con este lenguaje, pero una cuestión, la más difícil de responder, sigue en el aire: ¿ Se adapta XML a las necesidades de mi negocio ?.

La mayor parte de las empresas se mueven por las exigencias derivadas de su relación con clientes, proveedores y gestión interna. Sólo las grandes multinacionales pueden permitirse el lujo de actuar e investigar en las nuevas tecnologías. Es necesario, por lo tanto, reconocer el momento y aspecto del negocio por el que nos queremos introducir en la herramienta. Si es que finalmente decidimos adoptarla, lo que en estos momentos es mucho más que probable.

Si el capítulo anterior era una invitación al uso de XML en tantos dominios como puede ser aplicable, el presente es un estímulo para la reflexión y la seguridad de que dirigimos nuestros pasos hacia donde verdaderamente queremos hacerlo. Sólo conociendo las limitaciones de una materia seremos capaces de extraerle el máximo partido, y este es nuestro objetivo.

Es evidente que para desenmascarar las ventajas e inconvenientes de XML vamos a necesitar un punto de referencia para llevar a cabo las oportunas comparaciones. El apoyo que pedimos nos lo ofrece el EDI, que aunque presenta un gran número de inconvenientes con respecto a XML ha sido la base sobre la que ha funcionado el comercio electrónico existente hasta la actualidad.

En las páginas siguientes, se ofrecen los que han sido hasta el momento llamados *Pros* y *Contras* de XML. Junto a ellos se ofrece una idea que rebate el argumento original. Adoptando esta actitud se pretende disponer de los argumentos necesarios para justificar la inmersión en el nuevo estándar.

Ventaja**XML ES UN FORMATO DE TEXTO ESTRUCTURADO.**P
E
R
O

Por su propia naturaleza, no sólo representa la información sino los metadatos que la encapsulan. Los ficheros de texto sólo pueden representar una de las dos partes y en su forma se parece a las bases de datos relacionales.

Para los desarrolladores tiene un gran valor añadido, puesto que facilita la interfaz para la creación de documentos, evitando los famosos errores de sintaxis en los que la mejor ayuda suele ser el número de línea del error.



En algunos casos no se necesita de la complejidad que introduce XML, porque ficheros de texto pueden hacer perfectamente esta función. Por ejemplo, si necesitamos un par comando-valor los ficheros son más útiles.

Ventaja**XML SE HA DESARROLLADO CON INTERNET EN MENTE.**P
E
R
O

Uno de los objetivos de diseño de XML era convertirse en el SGML con capacidad Web, para poder aprovechar las distintas capas de transporte como son SMTP, FTP o HTTP. Incluso puede beneficiarse de los mecanismos de seguridad tales como SSL.

Adaptar EDI para alguno de estos protocolos o para incluir información tal que referencias URL es casi una odisea.



EDI funciona generalmente a través de redes de valor añadido (VAN). Sin embargo, aunque no es común, también puede utilizar los protocolos de transporte de Internet. Si no lo hace es para aprovechar los servicios añadidos de seguridad, privacidad, autenticación y confianza de las VAN.

Inconveniente**XML ES UN CONSUMIDOR INSACIABLE**P
E
R
O

Espacio, procesador y ancho de banda. Un documento tradicional puede llegar a crecer entre 3 y 20 veces su tamaño normal, con lo que ocupa mucho más espacio en disco, su procesamiento se vuelve más costoso en el tiempo y el ancho de banda necesario para su transmisión es imperativamente mayor.

El problema es particularmente delicado cuando se procura analizar el fichero bajo la especificación DOM, que lo intenta cargar entero en memoria.



No siempre será necesario procesar en tiempo real, con lo que las tareas se pueden ejecutar asíncronamente. Además, existen técnicas de compresión tanto para almacenamiento como para envío por la red. De todas maneras, es el precio que se suele pagar por las nuevas tecnologías.

Ventaja**XML ES DE FÁCIL LECTURA POR LAS PERSONAS.**P
E
R
O

Es importante que un lenguaje sea entendible con cierta facilidad. Cualquier persona que haya sido programador ha tenido que sumergirse en líneas de código escritas por otros programadores, con la dificultad que eso conlleva. Cuanto más pronto se reconocen los elementos, con mayor facilidad se ajustan a las necesidades.



Uno de los propósitos principales de XML es la comunicación entre ordenadores, con lo que el uso por las personas será relativo. Además, su comprensión dependerá en gran medida de las etiquetas, cuyos nombres pueden seguir siendo insignificantes.

Ventaja**XML ES FLEXIBLE. PERMITE DEFINIR OTROS LENGUAJES.**P
E
R
O

Existen muchas iniciativas, vocabularios, especificaciones y estándares en la industria, pero ninguno ha sido capaz de introducir la extensibilidad en los mismos, porque en tal caso EDI, HTML o SGML habrían iniciado la revolución que ha provocado XML. Validar un documento es muy sencillo y por lo tanto también lo es asegurar que la información que nos llega está 'bien formada'.



A veces puede resultar muy complicado crear un estándar en el que todo el mundo se ponga de acuerdo. Además, es difícil añadir anexos a un documento 'bien formado', puesto que se debe hacer entre etiquetas y en el interior del mismo.

Inconveniente**XML NO ES MÁS QUE UNA SINTAXIS PARA DOCUMENTOS.**P
E
R
O

XML no es una aplicación concreta ni un lenguaje de programación. Su función es tan simple como ofrecer un formato de documentos, a partir del cual se pueden construir analizadores sintácticos fácilmente.

Ha habido intentos de reemplazar lenguajes de programación por equivalentes en XML, pero se ha vuelto demasiado pesado, confuso, extenso y de difícil manejo y corrección de errores.



XML nunca pretendió ser un lenguaje de programación. Es fácil de leer, implementar, leer y escribir. Además es una forma efectiva, portable y personalizable de transmitir datos. XML se debe aplicar donde es útil, pero no en todos los campos de la informática.

Ventaja**XML SEPARA PROCESO DE CONTENIDO.**P
E
R
O

Puesto que representa información y los metadatos referentes a dicha información, no se preocupa de la forma en que se debe procesar o mostrar el documento. Además, se le pueden asociar hojas de estilo, de manera que los datos se transformen en información para páginas Web, teléfonos móviles, EDI o cualquier otro formato imaginable.



Muchas veces es el programador el que realiza la abstracción entre contenido y proceso. Si no es capaz de llevarla a cabo, XML no lo hará por él. Además, esto es una vía de entrada a especificaciones de plataformas o sistemas operativos, contrario a los principios del lenguaje.

Ventaja**LAS PYMES PODRÁN ACCEDER AL COMERCIO ELECTRÓNICO.**P
E
R
O

Herramientas para creación de documentos; procesamiento; validación; mapping; integración; acceso al núcleo de comunicaciones; seguridad... son algunos de los complicados procesos que dificultaban y encarecían el comercio electrónico. XML elimina algunas de las piezas de este puzzle y permite a todo tipo de empresas su entrada en Internet.



No sólo se trata de comunicación. Todas las herramientas integradas en el proceso de negocio deben adaptarse a Internet, y eso supone esfuerzo a todos los niveles. Y no hay que olvidar los costes de integración y personalización de estas herramientas.

Inconveniente**XML ES MUY SIMILAR A HTML Y A SGML.**P
E
R
O

Si es tan similar, ¿por qué no basta con esos estándares?. Mucha gente se pregunta por la necesidad de aprender otro lenguaje de marcado. Además, a HTML se le puede añadir una mezcla de SQL y se pueden crear aplicaciones que utilice todo el mundo. Es más, volveremos a pasar por el trance de las distintas versiones, compatibilidades entre ellas, etc.



XML no surge para sustituir a HTML o SGML. Aparece para representar la metainformación, ofrecer extensibilidad, validación y grandes posibilidades en el mundo del comercio electrónico.

Ventaja**ES TOTALMENTE INDEPENDIENTE Y LIBRE.**P
E
R
O

No tiene ninguna licencia comercial, y al separar proceso de contenido no pertenece a ninguna plataforma. Al ser una tecnología abierta, dispone de gran cantidad de recursos para facilitar su implementación. Se anulan las dependencias, pudiendo recibir asistencia técnica de una empresa, una implementación de otra y resolución de problemas por parte de una tercera.



Aunque estas son las características propias de XML, no se garantiza que las aplicaciones cumplan estos principios, y se pueden desarrollar de manera propietaria, comercial y con bajo soporte.

Ventaja**PERMITE EL COMERCIO ELECTRÓNICO.**P
E
R
O

XML y el comercio electrónico han surgido de forma casi paralela. Esto ha permitido un desarrollo conjunto que se traduce hoy en la existencia de una gran cantidad de plataformas como pueden ser RosettaNet, ebXML, UDDI o cXML, por citar las más comunes.

La manera de lograr una aceptación tan grande reside en el hecho de que XML ha superado los problemas de comunicación del sector.



XML no fue el primero en entrar en el negocio del comercio electrónico, porque EDI ya estaba presente. Acumula un cierto retraso que debe recuperar a medida que se introduzca en las empresas.

Inconveniente**¿ CÓMO TRATA XML LOS DATOS BINARIOS ?.**P
E
R
O

Una de las críticas que ha recibido es que es un buen lenguaje para texto, pero insuficiente para datos binarios. Para representar este tipo de información se puede codificar en ASCII o Unicode, porque leer la representación en texto de una imagen o de un vídeo es tan extenso como difícil de manipular.



Es cierto. Pero, ¿ por qué hay que incluir imágenes o vídeos en un sólo documento?. La respuesta puede ser para disponer del conjunto. De acuerdo, en ese caso se indica un enlace al archivo binario y problema resuelto.

Ventaja**CADA VEZ ES CONOCIDO POR MAS GENTE.**P
E
R
O

Su uso por parte de los individuos a la vez que el uso que hacen las organizaciones va en crecimiento exponencial. Factores de este éxito son su facilidad de uso y la proliferación de herramientas.

Los libros, conferencias, sitios web, seminarios y demás métodos de difusión de XML son cada vez mayores, y esto provoca la sensación y seguridad de buenos augurios de futuro.



Los empleados conocen cada vez mejor XML pero no los componentes asociados como Java o el Comercio Electrónico. Es necesaria una buena formación complementaria a la técnica con la teoría o la experiencia.

Ventaja**ES EL MOMENTO.**P
E
R
O

¿Cuándo han estado Sun, Microsoft, Oracle, Novell o IBM de acuerdo en el uso de una tecnología?. Parece suficiente para indicar que es el momento de ponerlo en práctica. Sin embargo hay más. Están los protocolos abiertos, los anchos de banda y los recursos necesarios para hacer que XML sea una realidad. No hay más que ponerse en movimiento.



A pesar de que una tecnología esté lista para su uso, no se sabe nada sobre su longevidad. Es necesario ser prudente y tener siempre presente que puede convertirse en una simple moda.

Inconveniente**XML ES UNA VUELTA ATRÁS A LOS SISTEMAS DE FICHEROS.**P
E
R
O

Parece una contradicción volver al procesamiento por ficheros cuando se dispone de bases de datos eficientes y centralizadas. Por una lado, los sistemas jerárquicos han fracasado desde los años 80 por su dificultad de almacenamiento. Por otro lado, los SGBD almacenan, indizan y recuperan la información de una manera muy robusta y cómoda.



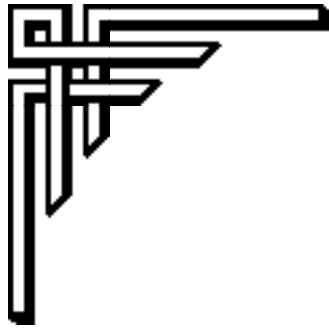
XML es ineficiente para representar información relacional. Pero ese no es su objetivo. Ha sido creado para modelar información jerárquica que puede ser desconocida a priori - al contrario de los SGBD donde es necesario conocer la información previamente.

Recapitulación.

La realidad de las afirmaciones que hemos visto depende siempre del cristal con que se miren. Para las organizaciones es difícil afrontar los cambios y, dentro de las mismas, suelen surgir individuos que los rechazan de entrada, sin valorarlos, por la simple inercia de la rutina que conocen y a la que se han acostumbrado. Por ejemplo, no es lógico que los desarrolladores de Sistemas Gestores de Bases de Datos se lancen al mundo XML por los cambios que les va a suponer, pero sí deben analizar las ventajas que pueden aportar esos cambios.

Lo que hemos descrito hasta el momento se puede clasificar de manera diferente a 'ventajas e inconvenientes de XML'. Podríamos decir que es un breve recorrido de los aspectos en que XML puede aportar beneficio a las empresas y de los que, por el contrario, no agilizan el funcionamiento. Ha quedado claro que su uso para generación de código es posible, aunque no eficiente.

La conclusión más importante que se puede deducir es que la funcionalidad máxima la vamos a extraer a la hora de intercomunicar sistemas heterogéneos, lo cual es particularmente beneficioso para el comercio electrónico. Almacenes de datos, herramientas de gestión de clientes, marketing personalizado o servicios post-venta podrán comunicarse a través de un interfaz común. Independientemente del fabricante o proveedor. Y en este sentido, el esfuerzo que supone el cambio ofrece grandes recompensas.



X

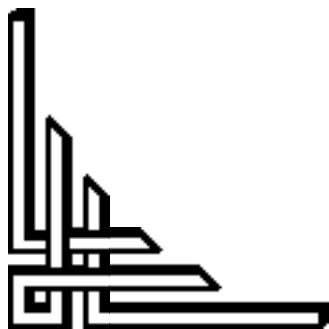
Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
O
N
I
C
O

M

ebXML.

[**E**LECTRONIC **B**USINESS **XML**]

L



El 12 de Mayo del año 2000 apareció la versión 1.0 de la Especificación de Requerimientos de *ebXML*. Era el resultado de una orden de trabajo de entre 15 y 18 meses encabezada por UN/CEFACT [*United Nations Centre for Trade Facilitation and Electronic Business*] y por OASIS [*Organization for the Advancement of the Structured Information Standards*]. El objetivo era proveer un marco de trabajo basado en XML utilizado de manera uniforme para el intercambio de datos de comercio electrónico entre personas, empresas o ambos.

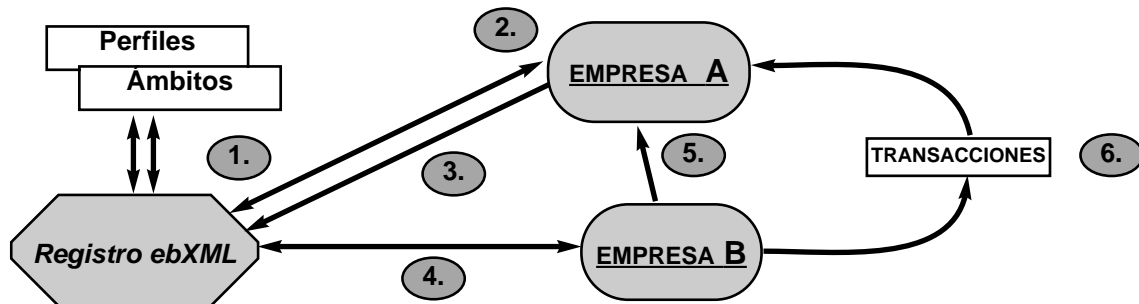
“ebXML pretende crear un conjunto de especificaciones técnicas acordadas internacionalmente, basadas en semántica XML y las estructuras de documentos relacionadas, para facilitar el comercio global”, tal y como reza en su especificación de requerimientos. Para conseguirlo se apoya en los siguientes puntos:

1. Es conforme a las especificaciones técnicas en XML de la W3C.
2. Provee interoperabilidad entre aplicaciones *ebXML*.
3. Es un camino intermedio entre EDI y los estándares XML en desarrollo sobre comercio electrónico que tiende a maximizar tanto la interoperabilidad como la eficiencia.
4. Será enviado a los organismos correspondientes para poder ser acreditado como estándar internacional.

Como todos los nuevos estándares, nace con un alcance que no es otro que todos los sectores de la comunidad de comercio internacional, con particular atención al conjunto de pequeñas y medianas empresas. Para ello, sus principios generales son los siguientes:

1. Permitir el negocio electrónico de manera simple y fácil con XML.
2. Reducir todas las iniciativas basadas en XML a un sólo estándar.
3. Integrar segmentos horizontales y verticales de la industria.
4. Evitar sistemas o soluciones propietarias.
5. Multi-idioma y ajustado a las normas de cada país.
6. Esforzarse en reducir los costes del comercio electrónico.
7. Convertirse en puente entre soluciones EDI y estándares XML.

Modelo de interacción.



1. La empresa A decide introducirse en el comercio electrónico y utilizar la arquitectura de ebXML. Para ello consulta un registro para examinar especificaciones y ejemplos de casos de uso.

2. La compañía A implementa una aplicación conforme al estándar.

3. Posteriormente envía la información relativa a su perfil de negocio, describiendo nuevas funcionalidades que se hayan introducido.

4. La empresa B identifica situaciones de negocio comunes con A, sabiendo que ambas son conformes a ebXML.

5. La compañía B se pone en contacto con A y se propone un acuerdo sobre los procesos de negocio, documentos, mensajes y tipo de seguridad a introducir en sus transacciones. Ambas llegan a un acuerdo.

6. Las dos empresas forman parte del comercio electrónico ebXML.

Vocabulario relacionado.

I. *Registro ebXML*. Es el contenedor de modelos de procesos, de vocabularios y perfiles utilizables.

II. *Perfiles*. Descripciones de las capacidades y restricciones que afectan a una empresa, así como sus ámbitos de actuación.

III. *Ámbitos*. Son las situaciones o entornos de negocio en los que una empresa está dispuesta a desarrollar su labor.

Requisitos.

Para implementar el modelo de interacción presentado, existen una serie de requisitos que ebXML necesariamente debe cumplir. Se pueden descomponer en requisitos técnicos y de negocio.

Requisitos de negocio.

Las necesidades a las que se debe acomodar pueden venir dictadas por soluciones globales. Por lo tanto, es vital la necesidad de acomodarse a requisitos nacionales e internacionales. Además, deberá hacerlo tanto para las implementaciones de B2B como de B2C.

Por otro lado, se deben soportar tanto desarrollos de industrias horizontales como verticales. Una variante de las mismas se pueden considerar los proyectos básicos de coste reducido para las pequeñas y medianas empresas. Para todos, habrá que mantener un sistema abierto y que nadie pueda convertir en propietario.

Requisitos técnicos.

Un aspecto fundamental es la arquitectura capaz de responder a los procesos de negocio y a la semántica introducida por los mismos. Se deben implementar metodologías de transporte, empaquetado y distribución de paquetes robustas y fiables. Y es necesario permitir sistemas de seguridad como las firmas digitales.

En un nivel jerárquico superior, es necesario un soporte que represente los procesos de negocio independientemente de su solución técnica y capaz de evolucionar en ambos sentidos. Estos procesos de negocio se establecerán entre socios tecnológicos independientes, con las necesidades de integración que ello supone, y en la medida de lo posible se debe eliminar la aparición de terceras partes para estructurar un modelo de negocio directo.

Comercio Electrónico.

Como cualquier otro tipo de empresa que participe en algún tipo de comercio, las que se introducen en el comercio electrónico deben cooperar entre ellas en los siguientes aspectos:

1. Mostrando los servicios que pueden ofrecerse.
2. Acordando procesos y documentos que deben intercambiar.
3. Determinando el modo de intercambio de información.
4. Poniéndose de acuerdo en los términos contractuales.

A partir del momento en el que hayan sentado las bases de los acuerdos necesarios anteriores, estarán en condiciones de intercambiar información y servicios de forma automatizada. Para solventar estas necesidades, ebXML se ha construido bajo tres conceptos básicos:

I. Proveer una **infraestructura** que asegure la interoperabilidad en la transmisión de datos. Para ello, es necesario un mecanismo estándar para el transporte de mensajes, con una interfaz bien definida, reglas de empaquetado y modelos de seguridad y reparto. Además, se debe gestionar la entrada y salida de mensajes, tanto en un lado como en el opuesto de la transmisión.

II. Establecer un marco **semántico** que permita la interoperabilidad en cuanto al negocio. Es decir, un metamodelo que defina procesos de negocio y modelos de información; un conjunto de componentes reutilizables; y un proceso para definir estructuras de mensajes y definiciones relativas a un modelo de proceso de negocio.

III. Crear un **mecanismo de descubrimiento** que facilite el que las empresas se pongan en contacto, se hagan socios tecnológicos y puedan generar negocio. Con dicha finalidad se crean los CPA o *Acuerdo de Protocolo de Comunicación* que estará relacionado con un CPP, *Perfil de Protocolo de Comunicación*. Todo ello residirá en una red de almacenes comunes de información, a los que las distintas empresas podrán registrarse. Es la forma más fácil de conocerse y comenzar una relación basada en el comercio electrónico.

Arquitectura.

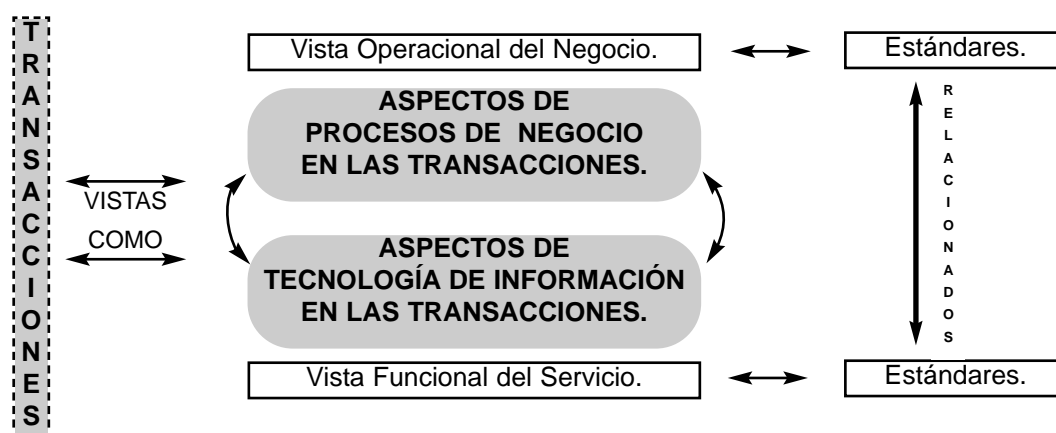
La arquitectura técnica de ebXML está basada en dos vistas, a su vez procedentes del modelo de referencia Open-EDI de la ISO.

1. Vista Operacional del Negocio.

Describe los procesos de negocio de forma totalmente independiente de los distintos lenguajes de programación. Con dicho propósito, ebXML se sirve de UML - Lenguaje de Modelado Unificado - y de UMM - Metodología de Modelado UN/CEFACT.

2. Vista Funcional del Servicio.

Es una referencia de cómo debe ser el marco técnico usado para descubrir y expresar la información relativa al negocio.



La vista operacional del negocio se captura en un registro y se define en tres fases: requisitos, análisis y diseño. El registro actúa como puente entre el lenguaje utilizado en una industria específica y los modelos generados por un lenguaje neutral previamente definido. A la hora de utilizar el registro, se buscan casos de uso similares al que se plantea y, en caso de que no existan, se generan otros nuevos.

La fase final del proceso, la estandarización, se obtiene por medio de los principios de la orientación a objetos, que es lo que asegura la posterior interoperabilidad entre todos los modelos.

En cuanto a la vista funcional del servicio, tiene su corazón en almacenes - *repositories* - que se utilizan por medio de registros ebXML. Estos almacenes/registros contienen los procesos de negocio y modelos de información contenidos en la vista operacional del servicio, el metamodelo ebXML, los perfiles de colaboración y, para terminar, las especificaciones ebXML.

La arquitectura ebXML define como establecer una situación de negocio a partir de los requisitos iniciales, identificando los siguientes aspectos:

1. Metodologías para la definición de:
 - I. perfiles de la compañía.
 - II. acuerdos con socios.
 - III. procesos de negocio.
 - IV. mensajes - documentos del proceso de negocio.
 - V. vocabularios.

2. Catálogo de:
 - I. procesos comunes de negocio.
 - II. semántica común.

3. Especificación de:
 - I. una capa de transporte uniforme.
 - II. interfaces para los almacenes de los catálogos anteriores.

4. Información suplementaria:
 - I. Sinopsis.
 - II. Guía.
 - III. Ejemplos.

Procesos de negocio.

Los modelos de proceso de negocios describen la forma en la que se debe operar sometido a la norma de *proceso a proceso*. La especificación de la definición de un proceso de negocio permite que una organización pueda expresar sus procesos en un marco común, donde otras empresas podrán también entenderlo. El resultado es la integración de los procesos de negocio de una compañía o, mejor aún, entre compañías diferentes.

El metamodelo ebXML que se debe utilizar para construir los modelos de proceso de negocio define:

1. subconjuntos semánticos.
2. esquema de especificación UML.
3. representación en XML de un esquema de especificación UML.

O lo que es lo mismo:

1. Definiciones para los socios tecnológicos.
[Roles, relaciones y responsabilidades].
2. Tipos de interacciones.
[Roles, transacciones de negocio y CPP, CPA].

Componentes centrales.

Definen componentes reutilizables que se pueden aplicar de forma estándar en un contexto de negocio. Representan los objetos del comercio electrónico que se definen utilizando elementos comunes del negocio. Es lo que permite a los usuarios definir datos significativos para su negocio, de manera que sean compatibles con otras aplicaciones. Un ejemplo de un componente central podría ser un país o una dirección.

Las especificaciones de la metodología de componentes centrales y el catálogo resultante se utilizan para capturar conceptos del negocio así como las relaciones entre dichos conceptos.

Registro y almacén.

Se encargan de proveer las funciones clave. Para el usuario - o la aplicación -, conservan perfiles de compañías y especificaciones de socios de negocio. Permiten, por tanto, el acceso a procesos de negocio específicos y modelos de información con la posibilidad de posteriores actualizaciones y adiciones. Para el desarrollador, además, contiene las bibliotecas de componentes centrales.

En general, cuando se considera el registro, debería quedar claro que se trata de una red de registros y almacenes conformes a ebXML.

1. Almacén de relaciones de datos.
 - I. Reglas de intercambio de información.
 - II. Modelos de documentos públicos.
 - III. Ejemplos, como DTDs y Esquemas.
 - IV. Perfiles.
2. Acceso seguro con autenticación.
3. Registro de servicios y clientes.
4. Simple para un uso global.
5. Escalable.
6. Basado en estándares de la industria.
7. Fácilmente identificable en Internet.

Información sobre socios de negocio.

Es lo que define los parámetros técnicos del CPP - *Perfil de Protocolo de Comunicación* - y del CPA - *Acuerdo del Protocolo de Comunicación*. Capturan información crítica para la comunicación entre aplicaciones y procesos de negocio, a la vez que almacenan parámetros técnicos para el desarrollo del comercio electrónico. El CPP incluye:

1. Información de contacto.
2. Clasificación de la industria.
3. Procesos de negocio soportados.
4. Requerimientos de su interfaz.
5. Requerimientos del servicio de mensajes.
6. Seguridad.

Por otro lado, el CPA aglutina los siguientes aspectos:

1. Identificación de servicios de mensajes ya implementados.
2. Identificación de requerimientos de procesos de negocios, como en el caso anterior, también implementados.
3. Si el acuerdo entre dos CPPs será usado por todas las partes implicadas en el proceso de negocio.

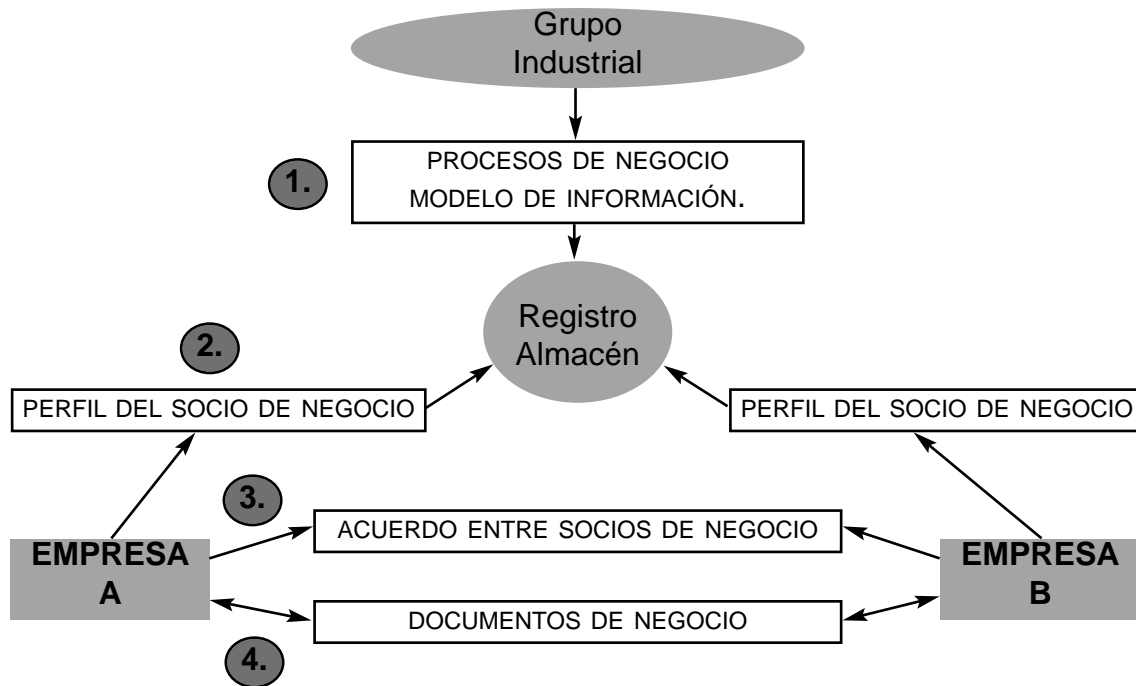
Servicios de mensajes.

La especificación del servicio de mensajes ebXML define el conjunto de servicios y protocolos que permiten a las aplicaciones de comercio electrónico el intercambio de datos. La especificación permite el uso de cualquier protocolo, incluyendo los conocidos y comunes en la red como HTTP, SMTP o FTP.

Se deben utilizar técnicas criptográficas bien implementadas para garantizar una buena seguridad, siendo aconsejable el uso de protocolos del estilo de HTTPS, por ejemplo, que aseguran la confidencialidad. En otro orden, las firmas digitales pueden ser aplicadas a mensajes individuales o a un grupo de mensajes para verificar su autenticidad.

Uso de ebXML.

El comercio electrónico se reduce a un proceso de cuatro tareas bien diferenciadas con ebXML, como se muestra a continuación. El algunos casos concretos pueden variar ligeramente el alcance o el número de empresas involucradas, aunque la base se mantiene similar.



1. Diseño y registro: procesos de negocio y modelos de información.

Se buscan en los almacenes procesos de negocio apropiados o, al menos, que soporte el socio con el que se quiere colaborar.

2. Implementación de interfaces de servicios y registro del CPP.

Se compran, crean o configuran aplicaciones capaces de funcionar en los procesos de negocio y se registra dicha habilidad de las aplicaciones en forma de CPP.

3. Negociar y definir, opcionalmente, un CPA.

4. Intercambio de mensajes entre socios de negocio.

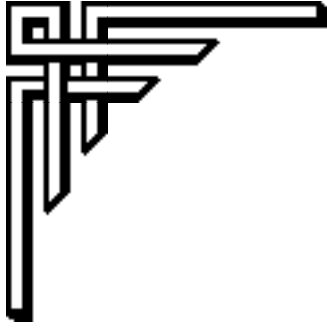
El diseño de ebXML no se ha realizado teniendo como objetivo al usuario final, sino que responde a las necesidades de desarrolladores, asociaciones de comercio o vendedores de plataformas operativas. Estas organizaciones serán las encargadas de crear guías específicas para industrias concretas basadas en ebXML. Las grandes empresas podrán crear sus propias soluciones y las pequeñas y medianas empresas podrán hacer uso de herramientas *prefabricadas* y fácilmente adaptables a su negocio.

El éxito de ebXML no radica en el detalle de sus especificaciones o en el nivel de precisión de su arquitectura, sino que, como en cualquier situación, depende del uso que le de la industria. Por lo tanto, es importante que se introduzca en la misma, para encontrar errores y respuestas más adecuadas a las distintas situaciones de negocio. La aportación más importante que cabe destacar de ebXML una vez más es que se trata de un estándar y que juntamente con XML ofrece flexibilidad y extensibilidad, augurios de un buen porvenir en el comercio electrónico.

A continuación se muestran, a la izquierda, un listado de empresas que han colaborado en su desarrollo y, a la derecha, el conjunto de empresas que actualmente utilizan el estándar ebXML.

Boeing.
Dun & Bradstreet.
Oracle.
XML Global.
Sun Microsystems.
Data Chanel.
IBM.
Mitre Corporation.
Muzmo.

Fujitsu.
Care Data Systems.
PeopleSoft.
Documentum.
Commerce One.
Edifecs.
IPNet Solutions.
SAA Consultants.
Bind Systems.

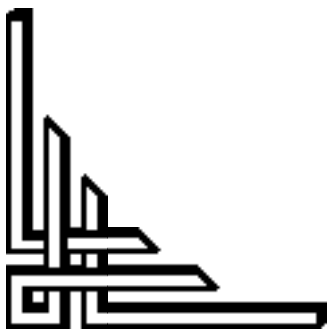


X

Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

M
BIZTALK.

L



Uno de los riesgos que se presentaban en el uso de XML era la posibilidad de que algunas empresas decidieran crear sistemas propietarios o cerrados basados en estándares abiertos. En el caso de Microsoft® BizTalk™, que ha creado todo un entorno de trabajo basado en las propuestas de la *World Wide Web Consortium*, nos encontramos en la frontera de este riesgo.

En realidad no se trata de un sistema exactamente cerrado, porque en tal caso sólo se permitiría el comercio electrónico entre empresas que dispusiesen de esta plataforma. Además, la arquitectura y formato de los mensajes y documentos BizTalk™ está disponible para todos aquellos que quieran utilizarlo. Sin embargo, el uso de Microsoft® BizTalk™ está inevitablemente ligado al sistema operativo de Microsoft®, así como a sistemas de bases de datos del mismo fabricante, como veremos más adelante en sus requerimientos.

Para describir adecuadamente lo que es exactamente BizTalk™ no podemos limitarnos a una estructura de documentos, registros, ámbitos, perfiles o funcionalidades, como era el caso de ebXML. En este caso, estamos frente a esas mismas características complementadas con todas las herramientas necesarias para su desarrollo, que es el valor añadido que presenta. Las más destacadas son:

1. Editores e integradores de documentos XML.
2. Soporte de SMTP, HTTP, hojas de estilo...
3. Concurrencia y sincronización.
4. Encriptación basada en la estructura de claves públicas.
5. Gestión de procesos de negocio, logs, seguimiento...
6. Herramientas de informes de procesos de negocio.
7. Arquitectura escalable.
8. Diseño visual de procesos de negocio.

Integración de aplicaciones.

El objetivo esencial con el que se ha diseñado Microsoft® BizTalk™ es el de integrar aplicaciones, conectarse de forma segura con socios tecnológicos a través de Internet y automatizar de forma rápida y flexible los procesos de negocio. Para facilitar dicha integración, además de usar XML como base tecnológica, se basa en los siguientes puntos:

1. Reducir los costes de integración de proyectos. *las aplicaciones se escalan a lo largo del tiempo para responder de forma efectiva a las demandas de capacidad del negocio.*
2. Integrarse con gran cantidad de productos y tecnologías. *XML junto con librerías de adaptación permiten la interconectividad con un abanico enorme de aplicaciones.*
3. Facilitar el desarrollo y la replicación. *son aplicaciones definidas como de 'apretar un botón', por su facilidad para replicar otras que son similares y se han desarrollado previamente.*

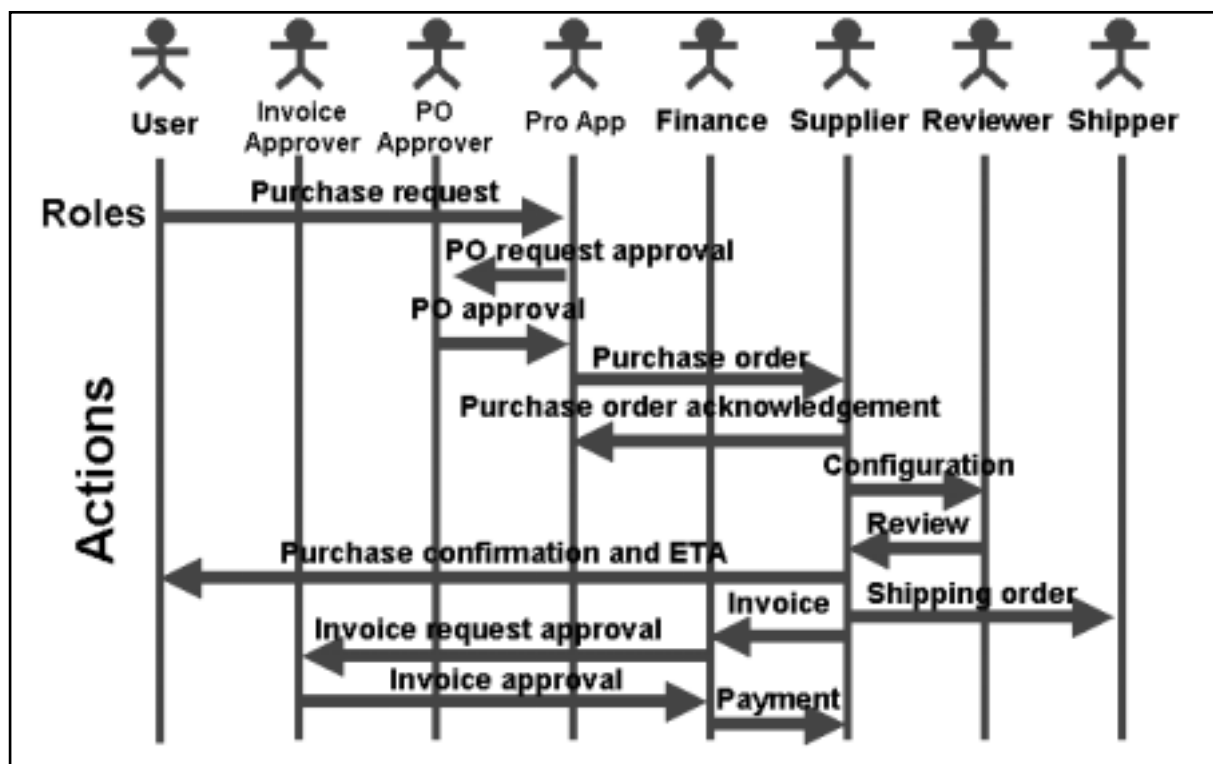
Independientemente del tipo de transporte, plataforma o aplicación que utilicen los socios tecnológicos, el objetivo principal es el de poder establecer situaciones de intercambio de información de manera segura y fiable. Una de las características principales relacionadas con este punto es el uso de un soporte para servicios Web basados en XML, pudiendo incluir en los mismos cualquier línea de negocio de forma fácil, rápida y eficiente.

Es necesario recordar la importancia de la velocidad a la hora de responder a solicitudes de clientes y amenazas del negocio. Esto se hace a través de la automatización de procesos de negocio y gestión y monitorización de operaciones. Sin embargo, este es un aspecto más relacionado con la gestión de los flujos de negocio y las bases de datos que la estructura de información - XML - a utilizar.

En la parte inferior de página tenemos un modelo utilizado por Microsoft® BizTalk™ para definir la relación existente entre los roles y las acciones necesarias en los procesos de negocio. Facturas (invoice), órdenes de compra (PO), logística (Shipping) o proveedores (suppliers) son algunos de los elementos destacados.

Es evidente que son complejos y que, a pesar de disponer de las herramientas necesarias para implementarlos, es necesaria una buena estructuración. Se dispone del alfabeto, que es XML, y de las palabras y de las frases que introducen los DTDs y los Schemas - en el caso de BizTalk™ se trara de Schemas. Para permitir que el negocio sea escalable, dinámico y robusto se definen tres aspectos fundamentales que son la base de la citada estructuración:

1. Separación entre definición e implementación.
2. Procesos dinámicos. *Cambiarán automáticamente en función de las relaciones entre los datos que manejen.*
3. Integración ‘cualquiera a cualquiera’. *Es imposible poder definir relaciones específicas entre plataformas o aplicaciones, por lo que se deberá garantizar una forma de comunicación estándar.*



SOAP - Simple Object Access Protocol.

BizTalk™ tiene una estructura muy similar a la del protocolo SOAP, un estándar de la W3C, a partir del cual se ha realizado una extensión para ajustar la semántica, la nomenclatura y la estructura de algunas etiquetas del lenguaje.

SOAP no es más que un protocolo ligero para el intercambio de información estructurada en un entorno distribuido. Su utilización puede estar basada en una gran cantidad de protocolos de nivel inferior, de forma que se transmitan los mensajes contruidos.

Los dos principios sobre los que se ha diseñado tanto SOAP como BizTalk™, por ser una ampliación del primero, son la simplicidad y la extensibilidad. Para ello, se han omitido algunas de las características que están presentes en los sistemas distribuidos y se han definido como extensiones de otro tipo de especificaciones. Algunas, que no todas, de las características suprimidas son las siguientes:

1. Fiabilidad.
2. Seguridad.
3. Correlación.
4. Direccionamiento.

El objetivo es el de obtener un protocolo que no se preocupe de estas características, cuya obligado cumplimiento recae entonces en los protocolos de nivel inferior en los que se basa.

Los mensajes creados bajo la especificación SOAP son conjuntos específicos de información creada con sintaxis XML, que pueden utilizar otro tipo de representación para su envío, y que sirven como referencia para la transmisión de información entre los distintos nodos que participan del intercambio electrónico de datos.

Terminología.

Un servidor **BFC** - *BizTalk Framework Compliant* - se representa por el conjunto de servicios que proveen la funcionalidad necesaria para el procesamiento de mensajes especificada en los requisitos de Microsoft® BizTalk™. A partir de este punto, se derivan dos definiciones importantes:

Aplicación: *Es el subsistema del negocio donde se almacenan y/o ejecutan tanto los datos como la lógica. También incluye cualquier tipo de adaptador necesario para emitir o procesar documentos de negocio y comunicarse con un servidor BFC.*

Documento de negocio: *Es un documento XML bien formado que contiene datos representando una factura, una orden de compra, publicidad o cualquier otro tipo de información relativa al negocio. Un conjunto de documentos de negocio forman el cuerpo (body) de lo que se denomina un documento BizTalk.*

La nomenclatura asociada a Microsoft® BizTalk™ introduce seis términos necesarios para entender la forma en que se estructuran los documentos.

1. Documento BizTalk. *Descrito en detalle en la siguiente página.*
2. BizTags. *Son las etiquetas XML - tanto las obligatorias como las optativas - que permiten el manejo de documentos de negocio.*
3. Mensaje BizTalk. *Es la unidad de intercambio de datos a nivel de red que se produce entre servidores BFC.*
4. Transporte. *Complementando la definición del punto tres, es el mecanismo de comunicación que permite el intercambio de los mensajes entre dos extremos.*
5. Endpoint. *Es la fuente o destino de un mensaje BizTalk. Cuenta con un atributo definido como address (dirección).*
6. Anexo. *Se trata de información que no se transmite dentro de un documento de negocio como fichero XML. Normalmente se refiere a imágenes, ficheros comprimidos...*

Un documento BizTalk es un mensaje que sigue el estándar de la W3C descrito anteriormente, SOAP 1.1, donde el cuerpo del mensaje contiene los documentos de negocio. Existen una serie de conceptos asociados a este tipo de documentos:

Tiempo de vida: es el tiempo durante el cual un documento tiene validez. Un documento no puede ser enviado, aceptado, procesado o reconocido más allá de su tiempo de vida.

Identidad: es un token único, universal que describe el documento.

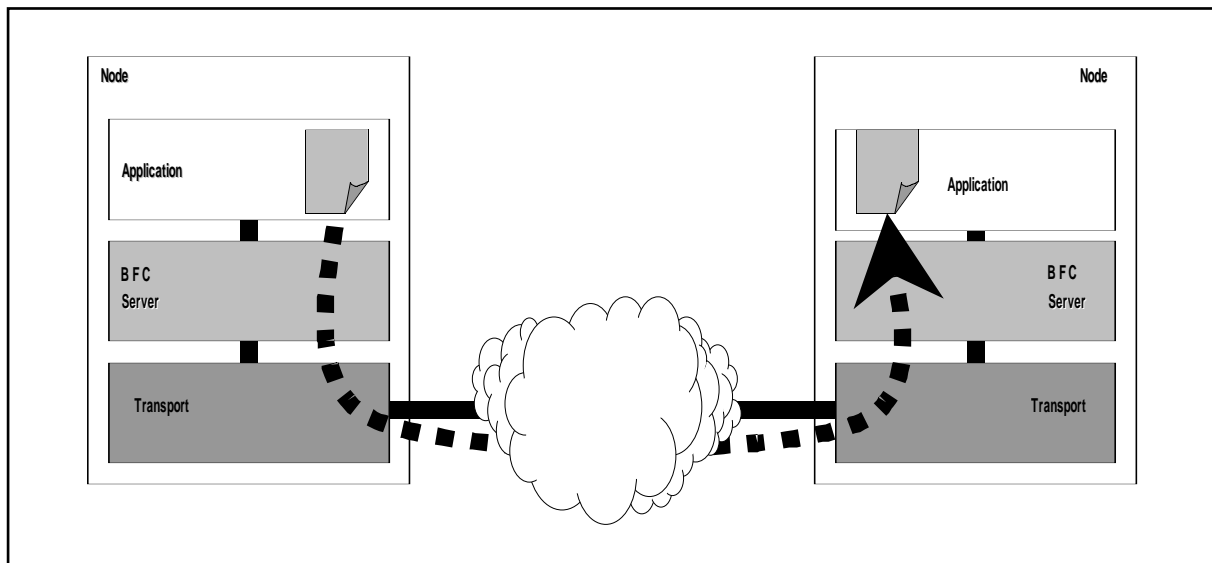
Aceptación: representa el hecho de ser aceptado para su reparto por parte de un receptor. Un documento se acepta si se reconoce un destino servido por el receptor, incluyendo copias o duplicados de documentos previamente recibidos. El que un documento sea aceptado no significa que se haya verificado su contenido.

Idempotencia: es la habilidad de un documento para transmitirse varias veces, recibiendo siempre el mismo tratamiento que si se hubiera transmitido una sola vez.

Recibos: BizTalk™ incluye protocolos punto a punto que prescriben el tipo de recibos que se deben enviar por parte de los receptores para asegurar el reparto en algunos casos. Existen dos tipos de recibos:

1. Reparto. Un recibo para certificar que el receptor acepta un documento recibido para su reparto.
2. Compromiso. Es un recibo que acredita que además de ser aceptado para su reparto, el documento ha sido analizado en su destino, con lo que se ha verificado su corrección y existe un compromiso para su procesamiento.

El modelo implementado por Microsoft® BizTalk™ se compone de tres capas, como se muestra en la siguiente imagen. Las tres capas en las que se divide el procesamiento en un nodo son la aplicación, el servidor BFC y el transporte. Las aplicaciones son, por tanto, el primer y último destino de los mensajes transmitidos, que se envían en uno y otro sentido a través de los servidores BFC. Los protocolos de comunicación pueden ser tan variados como HTTP, SMTP, MSQ - *Microsoft Message Queue*...



La aplicación es la que tiene la responsabilidad de generar los documentos de negocio para transmitirlos al resto de nodos a través de los servidores BFC. Sin embargo, la responsabilidad de empaquetar los documentos de negocio dentro de documentos BizTalk puede residir tanto en la aplicación como en el servidor.

En última instancia se debe destacar que la implementación de los interfaces entre la aplicación, el servidor y la capa de transporte están en función de la implementación realizada.

Anexos y seguridad.

Los distintos procesos de negocio requieren, a menudo, de la transmisión adjunta de otros tipos de documentos desde imágenes de cualquier tipo a diseños de ingeniería. En general, se trata de archivos binarios, para lo cual se especifican los siguientes elementos:

1. Una forma estándar de asociar un documento BizTalk con anexos a través de una estructura MIME para su transporte.
2. La relación entre la estructura MIME para los anexos y las etiquetas en la cabecera de los documentos BizTalk.

En primer lugar es necesario definir MIME - *MultiPurpose Internet Mail Extension*. Se trata de permitir la transmisión de información binaria, no ASCII, usando codificadores ASCII-7 con el fin de poder enviarla por medio de SMTP - *Simple Mail Transfer Protocol*. Para su identificación utiliza cabeceras específicas dentro del protocolo anterior y permite el envío de distintos contenidos y formatos: texto + voz + vídeo.

Los mecanismos de transporte que garantizan la seguridad ya son conocidos. Se trata de SSL que permite privacidad y autenticación, pero que no ofrece medios para firmar digitalmente y encriptar mensajes. La solución que aporta Microsoft® BizTalk™ para resolver el problema es el uso de S/MIME. Una forma segura de utilizar el protocolo MIME y que ofrece hasta tres modos de seguridad distintos:

1. Sólo encriptación.
2. Sólo firma digital.
3. Encriptación y firma digital simultáneas.

Cuando se incluyen anexos en un documento BizTalk, se realiza a través de atributos y referencias SOAP, de manera que se encripta de forma diferente el documento y sus anexos. De este modo, se pueden transmitir los documentos de forma clara y precisa sin comprometer en ningún momento su privacidad.

A continuación se muestra un documento Microsoft® BizTalk™ con la mayoría de los elementos descritos hasta el momento. No incluye anexos, pero sí todos los atributos que hemos descrito en la sección de terminología: por ejemplo, los 'endpoints' o los atributos relativos a las distintas propiedades de un documento BizTalk como tiempo de vida (expiresAt) o identificación (identity). En última instancia, el cuerpo del mensaje (Body) es el que contiene la orden de compra (po).

```

<?xml version='1.0' ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
  <SOAP-ENV:Header>
    <eps:endpoints SOAP-ENV:mustUnderstand="1"
      xmlns:eps="http://schemas.biztalk.org/btf-2-0/endpoints"
      xmlns:agr="http://www.trading-agreements.org/types/">
      <eps:to>
        <eps:address xsi:type="agr:department">
          Book Order Department
        </eps:address>
      </eps:to>
      <eps:from>
        <eps:address xsi:type="agr:organization">
          Booklovers Anonymous
        </eps:address>
      </eps:from>
    </eps:endpoints>
    <prop:properties SOAP-ENV:mustUnderstand="1"
      xmlns:prop="http://schemas.biztalk.org/btf-2-0/properties">
      <prop:identity>uuid:74b9f5d0-33fb-4a81-b02b-5b760641c1d6</prop:identity>
      <prop:sentAt>2000-05-14T03:00:00+08:00</prop:sentAt>
      <prop:expiresAt>2000-05-15T04:00:00+08:00</prop:expiresAt>
      <prop:topic>http://electrocommerce.org/purchase_order</prop:topic>
    </prop:properties>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <po:PurchaseOrder xmlns:po="http://electrocommerce.org/purchase_order/">
      <po:Title>Essential BizTalk</po:Title>
    </po:PurchaseOrder>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Requisitos.

Como ya habíamos descubierto anteriormente, nos encontramos ante un producto de la familia Microsoft®, con lo que la mayoría de sus requisitos vienen en forma de sistema operativo y sistema gestor de base de datos del gigante americano. Existen dos tipos de versiones, cliente y servidor, que analizamos por separado.

SERVIDOR [BIZTALK SERVER 2002]	CLIENTE [BIZTALK CLIENT TOOLS]
<p><u>Procesador.</u> Intel-compatible a 300 MHz.</p> <p><u>Sistema Operativo.</u> Windows® 2000 Server. [con el Service Pack 2 - SP2]</p> <p><u>Memoria.</u> 256 MB. de RAM o superior.</p> <p><u>Disco Duro.</u> 6 GB. de espacio o superior.</p> <p><u>Otros productos.</u> Internet Explorer 5. Microsoft SQL Server™ 7.0, SP2.</p> <p><u>Precio.</u> \$25.000 por procesador.</p>	<p><u>Procesador.</u> Intel-compatible a 166 MHz.</p> <p><u>Sistema Operativo.</u> Windows® 2000 Server o Windows® 2000 Professional.</p> <p><u>Memoria.</u> 16 MB. de RAM o superior.</p> <p><u>Disco Duro.</u> 80 MB. de espacio o superior.</p> <p><u>Otros productos.</u> Internet Explorer 5.</p>

De los datos anteriores derivamos una dependencia total en cuanto al sistema operativo, lo cual era lógico. Sin embargo, también aparece una dependencia en cuanto al sistema gestor de base de datos y el navegador. Por lo tanto nos encontramos ante una solución totalmente integrada pero cerrada por completo a las tecnologías de Microsoft®.

Caso de Estudio.

Introducción.

Coca Cola Beverages (CCB) recibe las órdenes de bebidas de sus mayores y más importantes socios y clientes a través del intercambio electrónico de datos o EDI. Desde mediados de los 80 había sido el procedimiento empleado como base para el intercambio de información estructurada que incluía solicitudes, contabilidad o logística.

Durante los últimos años se han realizado esfuerzos importantes para adaptar EDI a Internet, aunque su coste es elevado. Y la empresa tenía en mente mejorar algunos aspectos de su negocio, sobre todo en el apartado de solicitudes: mensajes de órdenes de compra de clientes y de socios, así como para procesos internos.

En el sistema del que se disponía, estas órdenes llegaban y eran transformadas por un convertidor EDI en ficheros para el sistema de la compañía, guardados en bases de datos de un AS/400. Posteriormente, se ejecutaban en segundo plano, lo cual era ineficiente...

Solución.

El objetivo consistía en liberar recursos, aumentar la productividad y poder hacer partícipes del sistema a pequeños socios y compañías sin el enorme coste de las soluciones EDI, con mayor agilidad y, sobre todo, facilidad de uso.

Después de consolidar la fase de requisitos, se instaló un servidor de Microsoft® BizTalk™ 2000 en fase de pruebas. Desde el principio, por parte de CCB existió una gran atracción por la capacidad de XML para integrarse con estándares precedentes como ANSI X.12 y EDIFACT, así como las nuevas formas de presentación de catálogos, ofertas, campañas de publicidad y otro tipo de información.

Consolidación.

Durante esta fase piloto, se trabajó en paralelo con los dos sistemas, a pesar de las ventajas comparativas que demostraba el nuevo sistema implementado. La mayor ventaja se obtuvo en el procesamiento de los datos importados desde la sede central de recepción de órdenes, que tardaba entre tres o cuatro minutos. Con la nueva solución, se reducía el tiempo necesario a un tercio del mismo. El motivo es que las tareas que anteriormente se ejecutaban en segundo plano pasan ahora a ser ejecutadas por el servidor BizTalk™, temporalmente almacenadas en Microsoft SQL Server™ y finalmente transferidas al AS/400.

El que acabamos de ver es sólo un ejemplo de una industria concreta en la que se ha resuelto un problema con la solución de Microsoft®. Sin embargo, existen muchos otros sectores donde se está implantando con éxito. A continuación se detallan algunas de las empresas más conocidas que hacen uso del sistema:

1. JD Edwards.
2. Avinon, Inc.
3. Navision.
4. Pivotal Software, Inc.
5. Siebel Systems.
6. Scala Business Solutions.
7. ONYX Software Corporation.
8. Teemplate Inc, etc.

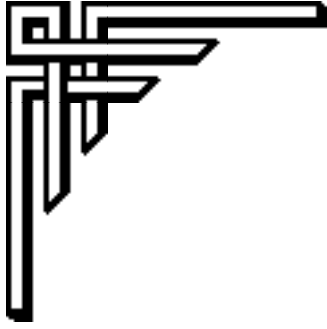
Conclusión.

La implantación de Microsoft® BizTalk™ responde a la necesidad de las empresas de utilizar soluciones *empaquetadas*. Es decir, una solución de comercio electrónico donde se dispone de todos los *paquetes* necesarios para el desarrollo de la actividad y donde el tiempo de inserción de la herramienta en el sistema se reduce a las fases de análisis de necesidades, adaptación sectorial y puesta en marcha.

La contraprestación aparece en el desembolso económico que es necesario llevar a cabo tanto para su funcionamiento como para su mantenimiento, ya que se crea una dependencia en cuanto al sistema operativo y al de gestión de bases de datos muy importante.

Frente a soluciones como *ebXML*, comentada anteriormente, o *cXML*, que aparece en el próximo capítulo, parte con la ventaja competitiva del plazo necesario desde que desarrolla la idea de inserción del comercio electrónico en la empresa, hasta que la misma es operativa. Sin embargo, es una solución excesivamente cara para pequeñas y medianas empresas que, con un buen planteamiento del sistema a desarrollar, en un tiempo mayor pero con total independencia hardware y software pueden crear una herramienta personalizada.

Como parte de la familia de productos Microsoft®, se trata de una aplicación con un elevado coste de adquisición. Por lo tanto, es necesario evaluar si su fácil manejo, experiencias previas del fabricante en la implantación y futuras ampliaciones que puedan acoplarse suplen, a largo plazo, la cuantía económica que sería necesaria para el desarrollo, explotación y mantenimiento de una solución propia.



X

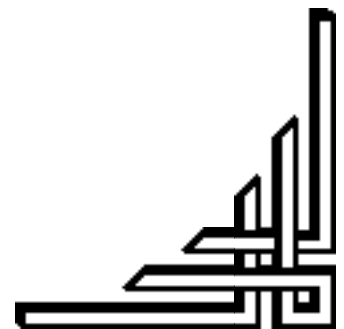
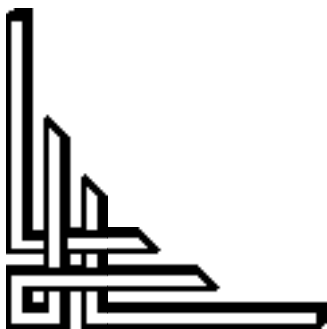
Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
O
N
I
C
O

M

CXML.

[COMMERCE XML]

L



El protocolo cXML es muy diferente del resto de los que implementan comercio electrónico. Está dirigido a asegurar la consistencia en las comunicaciones de documentos de negocios entre aplicaciones de pedidos, concentradores de comercio electrónico y proveedores. No posee todos los elementos de los que les gustaría disponer a las distintas partes para comunicarse, pero sí de los mecanismos necesarios para su expansión. Sus ventajas son varias:

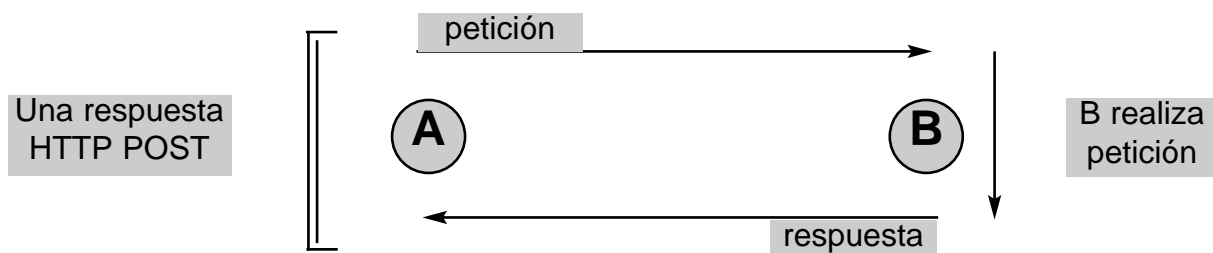
1. Fácil de implementar de forma automática catálogos, órdenes...
2. Es el protocolo mayormente aceptado.
3. Deja atrás la sintaxis relacionada con EDI.
4. Es extensible y adaptable.
5. Permite el empaquetado de soluciones.
6. Fue el primero en permitir mensajes síncronos para el desarrollo de portales de ventas por Internet - PunchOut.

Para llevarse a cabo una mejor implementación del lenguaje, este se ha distribuido en varios DTDs. De este modo se permite una mayor especialización en el desarrollo de los distintos aspectos del comercio electrónico que abarca, como son albaranes o facturas, por ejemplo. El hecho de que su descripción se realice a través de DTDs y no de Schemas responde a la falta de validadores sintácticos que cumplan de forma común y estable en los estándares de la W3C.

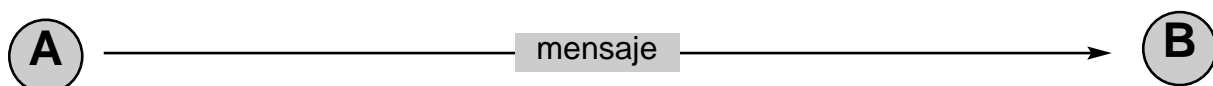
El éxito de cXML reside en su capacidad para distinguirse del omnipresente EDI. Puesto que su orientación es fundamentalmente la de Internet, sin la necesidad de proyecciones punto a punto, reduce su complejidad y facilita su uso en un entorno completamente distribuido como es la World Wide Web. Además, al contrario que otros estándares, provee ejemplos con código fuente para implementaciones basadas en lenguajes tan variados como ASP, Java, Perl o ColdFusion.

Especificación del protocolo.

Existen dos modelos de comunicación para las transacciones cXML: el Petición-respuesta y el Sentido único. Estos dos tipos de modelo permiten realizar de manera sencilla la implementación, ya que se describen con todo detalle las operaciones requeridas. Ambos modelos son necesarios porque se producen situaciones en que uno solo no sería adecuado. A continuación se muestran ambos modelos:



1. A inicia una conexión HTTP/1.x con una máquina B en una URL predeterminada que representa la dirección de B.
2. A utiliza una operación POST para enviar el documento cXML mediante la conexión HTTP.
3. A espera que se devuelva una respuesta a través de la conexión HTTP.
4. B tiene un servidor compatible con HTTP/1.x que remite la petición de HTTP al recurso especificado por la URL que se ha especificado en el paso 1. Este recurso puede ser cualquier ubicación conocida por el servidor HTTP de B, como por ejemplo un programa CGI o una página ASP.
5. El recurso de B, identificado en el paso 4, lee el contenido del documento cXML y asigna la petición al gestor adecuado.
6. El gestor de B encargado de esa petición cXML realiza la tarea que la petición especifica y genera un documento cXML como respuesta.
7. B envía la respuesta cXML a A mediante la conexión HTTP establecida en el paso 1.
8. A lee la respuesta cXML y la devuelve al proceso que ha iniciado la petición.
9. A cierra la conexión HTTP establecida en el paso 1.



1. A da formato y codifica un documento cXML con un transporte que resulte comprensible para B.
2. A envía el documento utilizando el transporte conocido. A no espera de manera activa (ya que no puede) una respuesta procedente de B.
3. B recibe el documento cXML y lo descodifica fuera de la corriente de transporte.
4. B procesa el documento.

cXML utiliza los elementos para describir los distintos artículos, que normalmente son propiedades en los documentos empresariales tradicionales. También la información que contiene claras subdivisiones y relaciones entre éstas, como por ejemplo una dirección, se describe utilizando estos elementos.

En cXML, todos los nombres de los elementos y los atributos utilizan palabras completas en mayúsculas (sin guiones) para separar las palabras. Los nombres de los elementos comienzan por una letra en mayúsculas y los nombres de los atributos por minúsculas. Por ejemplo:

Elementos: *Sender, Credential, Payment, ItemDetail.*

Atributos: *version, payloadID, lineNumber, domain.*

El elemento cXML es el cuerpo de un documento XML normal. Los documentos deben comenzar del siguiente modo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US" payloadID="0c300508b7863dcclb_13550"
timestamp="2000-01-09T01:36:05-08:00">
.....
```

Transmisión de los datos adjuntos.

Cuando se quiere hacer referencia a archivos externos, éstos pueden residir en un servidor accesible al proveedor o se pueden transferir junto con el documento cXML. Esta segunda opción requiere el uso de un sobre MIME con varias partes. Un requisito cXML para este sobre es que se incluyan las cabeceras Content-ID con cada archivo adjunto. En el siguiente ejemplo se muestra la estructura necesaria de un documento cXML con una imagen JPEG adjunta.

```
POST /cXML HTTP/1.0
Content-type: multipart/mixed; boundary=identificador único
Content-type: text/xml; charset="UTF-8"
<?xml version="1.0" encoding="UTF-8"?>
...
--identificador único
Content-type: image/jpeg
Content-ID: <uniqueCID@cxml.org>
...
--identificador único--
```

Header.

El elemento Header contiene información sobre la dirección y la verificación. Este elemento es el mismo independientemente de la petición o respuesta específicas que se contengan en el cuerpo del mensaje cXML. Las aplicaciones necesitan la identidad del solicitante, pero no es necesario comprobar que la información sobre la identidad proporcionada es correcta. En el siguiente ejemplo se muestra el elemento Header:

```
<Header>
<From>
<Credential domain="AribaNetworkUserId">
<Identity>admin@acme.com</Identity>
</Credential>
</From>
<To>
<Credential domain="DUNS">
<Identity>012345678</Identity>
</Credential>
</To>
<Sender>
<Credential domain="AribaNetworkUserId">
<Identity>sysadmin@ariba.com</Identity>
<SharedSecret>abracadabra</SharedSecret>
</Credential>
<UserAgent>Ariba Network 1.1</UserAgent>
</Sender>
</Header>
```

Los elementos From y To son sinónimos de los elementos From y To (De y Para) de los mensajes de correo SMTP, que son la fuente lógica y el destino de los mensajes. Sender es la parte que inicia la conexión HTTP y envía el documento cXML.

Request.

Los clientes envían peticiones para solicitar operaciones. Sólo se permite un elemento Request para cada elemento de sobre cXML, lo que simplifica las implementaciones del servidor, porque no es necesario que se produzca ninguna desmultiplexación al leer los documentos cXML. El elemento Request puede contener virtualmente cualquier tipo de datos XML.

Response.

Los servidores envían respuestas para informar a los clientes de los resultados de las operaciones. Como puede que los resultados de algunas peticiones no tengan datos, el elemento Response puede contener de manera opcional sólo un elemento Status. El elemento Response también puede contener datos de cualquier nivel de la aplicación.

Status: Este elemento indica el éxito o fallo de una operación de petición. Tiene los siguientes atributos:

1. **code.** *Código del estado de la petición. Sigue el modelo de códigos de estados HTTP. Por ejemplo, 200 representa una petición correcta.*
2. **text.** *Texto del mensaje de estado. Este texto facilita al usuario la lectura de los registros y está compuesto de cadenas canónicas en inglés.*
3. **xml:lang** (opcional). *Idioma de los datos en el elemento Status.*

Elementos básicos.

Las siguientes entidades y elementos se utilizan a lo largo de toda la especificación cXML. La mayoría de las definiciones presentan un vocabulario básico con el que se describen los documentos comerciales de los pedidos mayores. Aquí se definen las entidades de tipo común y los elementos más habituales que representan objetos de bajo nivel.

Entidades de tipo.

La mayoría de estas definiciones proceden del envío de la nota de los datos XML al World Wide Web Consortium (W3C). Un número muy reducido de entidades de tipo de alto nivel que también se definen aquí no son datos XML.

isoLangCode: Código de idioma ISO.

isoCountryCode: Código de país ISO.

xmlLangCode: Código de idioma como se define en XML 1.0.

unitOfMeasure: Modo en que el producto se empaqueta o envía.

URL: Tal como se define en la normativa HTTP/1.1.

Es necesario recordar que cuando hacemos referencia códigos ISO, estos van siempre asociados a una de las normativas que rigen este tipo de usos - en particular las ISO 639 e ISO 3166 para *isoLangCode* e *isoCountryCode*, respectivamente.

Elementos base.

Estos elementos, utilizados en la especificación, van desde los más genéricos, como por ejemplo *Name* y *Extrinsic*, a los más específicos, como por ejemplo *Money*.

Transacciones de perfil.

Las implementaciones del servidor cXML 1.1 deben admitir los documentos *ProfileRequest* y *ProfileResponse*. Esta transacción se debe utilizar para recuperar las capacidades del servidor, incluyendo la versión cXML admitida, las transacciones y las opciones de éstas. La respuesta debe contener todas las peticiones admitidas en un sitio Web concreto, no necesariamente todas las que admite la empresa.

ProfileRequest.

Este elemento no tiene contenido. Tan sólo se dirige al servidor cXML adecuado utilizando Header. El servidor responde con una sola *ProfileResponse* como se describe más abajo. Las únicas partes dinámicas de esta respuesta son los atributos *payloadId* y *timestamp* del propio elemento cXML. En este caso concreto, no es necesario que un proveedor proporcione respuestas en varios lugares.

ProfileResponse.

Este elemento contiene una lista de transacciones admitidas, sus ubicaciones y las opciones admitidas. Tiene el atributo *effectiveDate*, fecha y hora en que estos servicios están disponibles.

Option: Valor para una opción definida (para todos los servicios o para una transacción específica). Aún no se ha definido ninguna opción. Option tiene el atributo *name*.

Transacción: La descripción de una transacción admitida por este servicio. La definición de perfil actualmente indica las ubicaciones a las que se envían las peticiones específicas. El elemento Transaction debe contener un elemento URL. Tiene un atributo: *requestName*.

Definiciones de pedidos.

Los documentos de pedidos cXML son *OrderRequest* y una respuesta genérica. *OrderRequest* es similar a un pedido de compra. La respuesta es una confirmación de que el proveedor lo ha recibido. No es necesario ejecutar el citado pedido, sino la confirmación de que se ha recibido correctamente.

OrderRequest.

En el siguiente ejemplo se muestra la estructura del elemento *OrderRequest*. *OrderRequestHeader* incluye información sobre la facturación global y el pago, mientras que *ItemOut* describe los artículos.

```
<OrderRequest>
<OrderRequestHeader ... >
...
</OrderRequestHeader>
<ItemOut ... >
...
</ItemOut>
<ItemOut ... >
...
</ItemOut>
</OrderRequest>
```

Respuesta a una OrderRequest.

Es la parte de respuesta de la transacción de petición-respuesta sincrónica. En el siguiente ejemplo se muestra la respuesta de un documento *OrderRequest*:

```
<cXML version="1.1.007" payloadID="9949494" xml:lang="en"
timestamp="1999-03-12T18:39:09-08:00">
<Response>
<Status code="200" text="OK"/>
</Response>
</cXML>
```

Como se mostraba anteriormente, esta respuesta es sencilla. La respuesta comunica al solicitante que la parte remota de la conexión HTTP ha analizado y ha cumplido correctamente su *OrderRequest*. Sin embargo, no comunica la confirmación en el nivel de pedido, como por ejemplo qué artículos se pueden enviar o cuáles se deben reponer.

Transacción de PunchOut.

Las definiciones de los mensajes de punchout son mensajes de petición-respuesta que están incluidos en los elementos *Request* y *Response*. Los proveedores deben implementar los mensajes que se describen a continuación para admitir el punchout.

PunchOutSetupRequest.

PunchOutSetupRequest y *PunchOutSetupResponse* son el par petición/respuesta utilizado para establecer una sesión de punchout con un sistema remoto. El cliente los utiliza para identificar la aplicación de adquisición, enviar la información de configuración y recibir una respuesta que indique dónde dirigirse para iniciar una sesión de exploración HTML en el sitio Web remoto.

PunchOutSetupResponse.

Una vez haya recibido el sitio Web una *PunchOutSetupRequest*, responde con una *PunchOutSetupResponse*, como se muestra a continuación:

```
<PunchOutSetupResponse>
<StartPage>
<URL>
http://direccion.ejemplo.com/tienda?23423SDFSDF23
</URL>
</StartPage>
</PunchOutSetupResponse>
```

PunchOutOrderMessage.

Este elemento envía el contenido de la cesta de la compra remota al autor del *PunchOutSetupMessage*. Puede contener más datos que los otros mensajes porque debe expresar completamente el contenido de cualquier cesta de la compra que se pueda concebir en el sitio Web externo. Este mensaje no sigue de manera estricta el modelo petición-respuesta.

Cambios de estado posteriores.

Una vez finalizada la transacción de *OrderRequest*, puede que los proveedores y los servidores intermediarios tengan que proporcionar información adicional al sistema comprador. Las transacciones descritas en este apartado se utilizan para este fin. Estas transacciones comparten algunos rasgos semánticos y elementos comunes.

Al igual que sucedía con las respuestas a una *OrderRequest*, ninguna de estas transacciones incluye un elemento *Response* específico. En su lugar, el documento devuelto contiene una *Response* casi vacía - sólo un *Status*. Los documentos devueltos presentan el siguiente formato: (el código devuelto es "200" sólo si la operación ha finalizado correctamente).

```
<cXML version="1.1.007" payloadID="9949494@supplier.com"
timestamp="2000-01-12T18:39:09-08:00" xml:lang="en-US">
<Response>
<Status code="200" text="OK"/>
</Response>
</cXML>
```

DocumentReference.

El elemento *DocumentReference* contiene información suficiente como para asociar la petición actualizada con un documento concreto. Repite un atributo necesario del documento anterior y añade un identificador opcional que el proveedor genera.

StatusUpdateRequest.

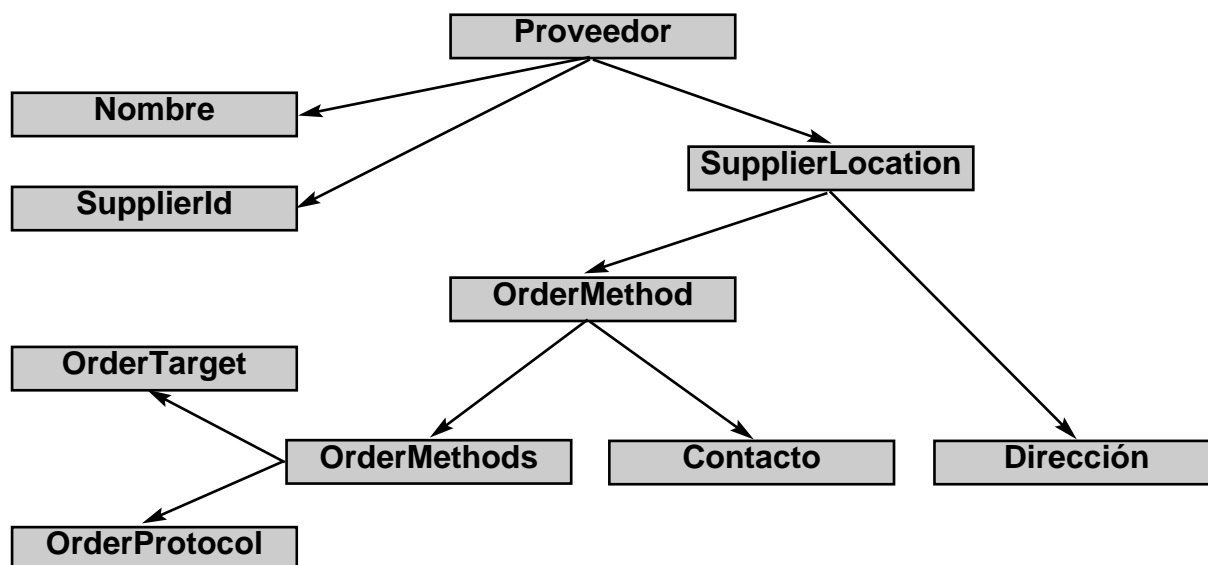
Esta transacción comunica a un nodo anterior acerca de los cambios realizados en el estado del procesamiento de un pedido. Existe un cambio de especial importancia: cuando un concentrador intermedio transmite correctamente una *OrderRequest* hacia delante, puede informar al emisor original o a un concentrador previo acerca de lo ocurrido. Asimismo puede que las transiciones a través de varias colas y fases de procesamiento de un proveedor o concentrador sean también de gran importancia para el comprador.

Definiciones del catálogo.

Las definiciones del catálogo cXML constan de tres elementos principales: Supplier, Index y Contract. Todos estos elementos describen los datos concebidos para uso permanente o temporal en el sistema de adquisición de un comprador o un concentrador.

1. Supplier. Contiene datos básicos acerca del proveedor, como la información sobre la realización de pedidos, la dirección y contacto.
2. Index. Describe los datos acerca del inventario de productos del proveedor, como por ejemplo descripciones, números de pieza y códigos de clasificación.
3. Contract. Describe los datos acerca de aspectos flexibles del inventario acordado entre el comprador y el proveedor, como es el precio.

Uno de los aspectos a tener en cuenta es que *Index* utiliza varios subelementos para describir los artículos de línea en los inventarios del proveedor. Los proveedores pueden enviar la información sobre los precios para entrar en los sistemas del comprador o bien la información sobre *punchout* para que los compradores puedan realizar un *punchout* en los sitios Web remotos para obtener información sobre los precios y otros datos de interés.



Gestión de suscripciones y de mensajes.

Gestión de suscripciones.

Los intermediarios, como los concentradores de red de comercio electrónico, pueden gestionar catálogos de los proveedores que utilizan los sistemas de adquisición de los compradores. Estos intermediarios pueden proporcionar vínculos directos entre los sistemas de adquisición y los sistemas del proveedor.

1. *Datos del proveedor.* Las definiciones para la gestión de datos del proveedor están compuestas principalmente de los elementos *SupplierListRequest*, *SupplierListResponse*, *SupplierDataRequest*, *SupplierDataResponse* y *SupplierChangeMessage*.
2. *Suscripciones del catálogo.*

Gestión de mensajes.

Algunos compradores no tienen puntos de entrada HTTP para recibir mensajes cXML que no proceden de los servidores de seguridad de la empresa. La especificación cXML permite estos entornos. Este apartado es una introducción a las definiciones que permiten a los sistemas de origen dejar los mensajes en cola cuando los objetivos no pueden aceptar directamente los envíos HTTP. Los objetivos, a su vez, van extrayendo los mensajes como consideren conveniente.

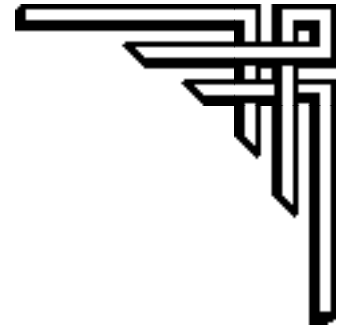
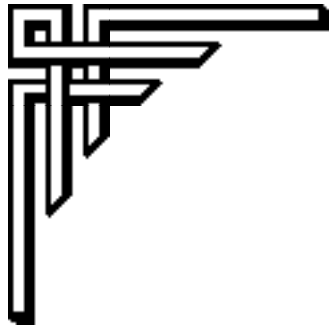
1. *GetPendingRequest.* Este elemento extrae un conjunto de mensajes en espera del solicitante. Tanto el elemento denominado *MessageType* como los atributos *LastReceivedTimestamp* y *maxMessages* controlan el tipo y hacen un recuento de los mensajes recuperados.
2. *GetPendingResponse.* Este elemento contiene uno o varios mensajes que esperan al solicitante para que pueda extraerlos.

Conclusión.

Si queremos implantar una solución de comercio electrónico, cXML responde de manera excelente al problema técnico que plantea el intercambio de información entre las distintas partes, así como los protocolos necesarios para garantizar seguridad, fiabilidad y otras características imprescindibles.

Pero cXML es sólo el lenguaje. Se trata de la piedra angular del proyecto, pero también es necesario tener en cuenta que para un uso eficiente de XML se necesitan editores, herramientas de análisis de negocio con clientes y proveedores o paquetes estadísticos que muestren la evolución de los resultados. Al igual que *ebXML*, cXML garantiza el intercambio efectivo de información, pero es necesario buscar las herramientas para su desarrollo.

El punto fundamental es que en sus propias limitaciones están sus virtudes. Puesto que se trata únicamente del lenguaje, cXML puede adaptarse a cualquier plataforma, sistema operativo o módulo complementario para el desarrollo del comercio electrónico. Esto permite a sus usuarios la libre elección de su entorno técnico de trabajo, así como posibles ampliaciones y nuevos desarrollos que puedan ser adaptables a la solución creada.



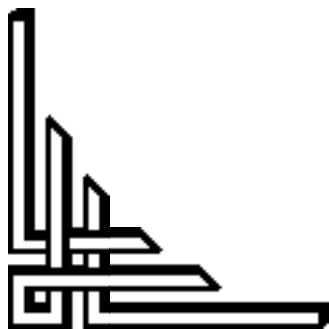
X

Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

ANÁLISIS DEL ENTORNO.

M

L



Desde su introducción, pocas tecnologías han sido tan ensalzadas como ha sido el caso de XML. Sin embargo, a pesar de ser reconocido como el primer actor y el elemento clave del desarrollo del comercio electrónico, en realidad se trata únicamente de un primer paso. El peligro potencial de la creación de *estándares propietarios* amenaza con minar la adopción de XML como *estándar de facto*.

La satisfacción en la implantación de una solución XML para la integración, intercambio y transformación de la información depende, en gran medida, de soluciones genéricas que sean capaces de introducir los distintos proveedores de *middleware*. El motivo es la gran cantidad de procesos de transformación que van a ser necesarios así como la coexistencia de sistemas XML con otros que no lo son.

Los grandes fabricantes llevan siempre la iniciativa por su capacidad para invertir recursos técnicos y económicos tanto en investigación como en desarrollo. Pero no sólo de empresas multinacionales de éxito y de grandes fabricantes se nutre la informática. A su alrededor hay organismos que rigen su funcionamiento.

Un aspecto que cada vez recibe más atención en la formación de profesionales de los sistemas de información es la legalidad y la auditoría. Es necesario que los recién titulados hayan adquirido un buen conocimiento de los reglamentos de seguridad, de leyes sobre el tratamiento automatizado de datos y, en general, de la seguridad de la información que manejan.

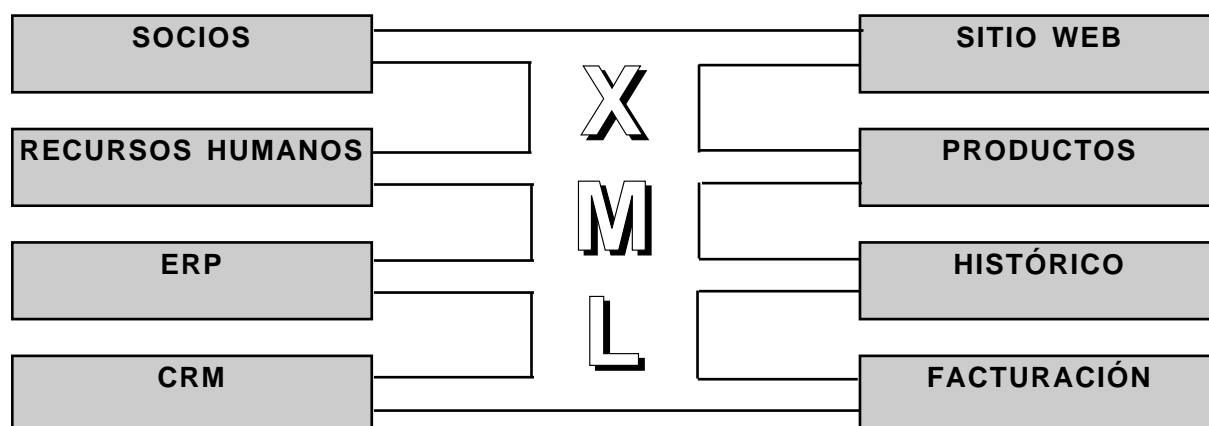
A lo largo del presente capítulo se muestran algunos de los aspectos más relevantes de lo que podríamos denominar el entorno de XML y el comercio electrónico. Es decir, aquellos aspectos no directamente relacionados con su implementación, pero que es fundamental dominar para tener éxito en la implantación de sistemas.

Participación de multinacionales.

Que las grandes empresas hace años que utilizan XML para sus desarrollos no es ningún secreto. Tampoco lo es el hecho de que algunas de ellas hayan aprovechado el nuevo lenguaje para crear soluciones propietarias que luego venden a sus clientes, manteniendo un núcleo cerrado de soluciones y asegurando su cuota de negocio.

La mayor ventaja de la que pueden beneficiarse los clientes de las grandes multinacionales es de que, por fin, van a poder elegir módulos y sistemas de distintos fabricantes y van a poderlos enlazar, no sin cierta dificultad, pero con la seguridad de un buen funcionamiento. La aparición de un estándar como XML ha supuesto que una empresa pueda disponer de un ERP de un fabricante, combinado con un CRM - *Customer Relationship Management* - de otro distinto para acabarlo integrando con un SCM - *Supply Chain Management* - de un tercero. Además, los fabricantes han entendido que una de las mayores ventajas competitivas que pueden ofrecer sus productos es el grado de facilidad del que disponen para adaptarse a soluciones de la competencia.

El ejemplo más claro de lo que acabamos de citar está presente en la página web de Siebel, una de las empresas tecnológicas con mayor proyección en los últimos años y líder en sistemas de gestión de empresas, relaciones con los clientes, etc. En la misma, la imagen central que aparece en una pequeña presentación de FlashTM es similar al siguiente diseño, simulando lo que debería ser la integración de todos los sistemas de una empresa: [<http://www.siebel.com>]



Legalidad.

Convenio 108 del Consejo de Europa.

Artículo 1. Objeto y fin. *El fin del presente Convenio es garantizar, en el territorio de cada Parte, a cualquier persona física sean cuales fueren su nacionalidad o su residencia, el respeto de sus derechos y libertades fundamentales, concretamente su derecho a la vida privada, con respecto al tratamiento automatizado de los datos de carácter personal correspondientes a dicha persona.*

Directiva de la Unión Europea.

Artículo 1. Objeto de la Directiva. *Los Estados miembros garantizarán, con arreglo a las disposiciones de la presente Directiva, la protección de las libertades y de los derechos fundamentales de las personas físicas, y, en particular, del derecho a la intimidad, en lo que respecta al tratamiento de los datos personales.*

LOPD (Ley Orgánica de Protección de Datos).

Artículo 1. Objeto. *La presente Ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.*

Las leyes orgánicas anteriores, la directiva de la Unión Europea así como el convenio del Consejo de Europa tienen por objetivo el garantizar la intimidad de todos los individuos y la confidencialidad de todos sus datos de carácter personal.

A la hora de implementar soluciones de comercio electrónico, intervienen diversos elementos como bases de datos que contienen información de empresas y/o clientes que deben ser tratadas conforme a las normativas anteriores. Por lo tanto, hay que tener en cuenta la forma en que se transmite la información para no incurrir en ningún delito relacionado con el tratamiento automatizado de datos.

El ámbito de aplicación de las leyes o normativas anteriores incluye tanto al sector público como al privado y se centra en los datos de carácter personal en soporte físico. Por datos de carácter personal se entiende cualquier información concerniente a personas físicas identificadas o identificables.

A la hora de proteger los datos de carácter personal en un sistema de información, hay que tener en cuenta el nivel de protección al que deben someterse. La ley establece como datos especialmente protegidos los citados a continuación:

1. Ideología.
2. Religión.
3. Creencias.
4. Origen racial.
5. Vida sexual.
6. Salud.

Un aspecto de especial interés para las empresas es que los interesados que, como consecuencia del incumplimiento de lo dispuesto en las distintas leyes o reglamentos, sufran daño o lesión en sus bienes o derechos tendrán derecho a ser indemnizados.

Seguridad.

La seguridad es la principal inquietud que se genera a la hora de enviar datos a través de la red. Para que deje de ser un problema, se necesitan cuatro características:

1. Confidencialidad.
2. Integridad - que nadie altere los documentos.
3. Autenticación.
4. No repudio.

SSL - Secure Socket Layer - provee las tres primeras funciones. ¿Y la cuarta?. Se trata de las firmas digitales. Para introducirlas en los documentos XML, la W3C está desarrollando XML Signature, o lo que es lo mismo, la forma en que se deberán firmar los documentos. Existen tres mecanismos para la inclusión de firmas electrónicas en los documentos. Se pueden descomponer en cuanto a su integración en el documento original, según la siguiente nomenclatura:

1. *Firma anexa*. En realidad se trata de dos documentos. Por un lado el documento original y por otro la firma electrónica. Es la forma más débil o menos complicada de firmar un documento.
2. *Firma contenida*. En este caso, el documento está protegido por la firma electrónica, que actúa en cierto modo como envoltorio del documento.
3. *Firma múltiple*. ¿Qué ocurre si un documento ha sido generado por varios individuos?. Es posible que todos quieran firmarlo. ¿Y si cada uno genera la firma en distintos momentos?. En tal caso, se podría violar el principio de integridad.

Con la firme múltiple XML va un paso más adelante. Permite que cada segmento de un documento sea firmado independientemente del resto, lo que permite, por ejemplo, la siguiente situación: un comercial realiza una venta e incluye su firma; el responsable de gestión valida al cliente e incluye la suya; el gerente, que lo debe firmar todo, revisa el documento e incluye su firma. Conclusión: se pueden establecer niveles jerárquicos en la firma de los documentos.

Auditoría.

Auditoría es la investigación, consulta, revisión, verificación, comprobación y obtención de evidencia, desde una posición de independencia, sobre la documentación e información de una organización, realizadas por un profesional -el auditor- designado para desempeñar tales funciones.

Los sistemas de información están obligados a pasar revisiones periódicas de manera que se pueda garantizar que se está haciendo un uso eficiente y, sobre todo, conforme a la ley, de los mismos. Por lo tanto, en el momento de implementar soluciones, se deben prever los mecanismos necesarios para que los informes de auditoría derivados sean siempre favorables.

Existen una serie de puntos básicos en cuanto a la separación de funciones que es necesario respetar. No están directamente relacionados con la forma en la que se diseña un sistema de información o una solución de comercio electrónico, pero sí que son importantes en el momento de su explotación.

1. Separar Proceso de Datos de los usuarios funcionales.
2. Separar Proceso de Datos de la auditoría de los sistemas de información.
3. Separar Proceso de Datos de la auditoría de proceso de datos.
4. Separar los distintos departamentos usuarios funcionales.

El rol de los auditores consiste en identificar los riesgos y exposiciones de los sistemas, favoreciendo los controles que se consideren adecuados para mitigarlos. Además, es su tarea evaluar la efectividad de los citados controles. No hay que olvidar que la estrategia de auditoría puede incluir el nivel de dirección, con lo que una adecuada planificación del sistema será más que necesaria.

Una forma de reforzar los controles internos y de reducir la posibilidad de errores, omisiones e irregularidades, consiste en construir e insertar controles en la explotación y en los sistemas, y revisar su adecuación. Algunas de las técnicas y herramientas que facilitan una revisión de los controles son las siguientes:

1. Pistas de Auditoría.
2. Verificaciones de totales de control.
3. Registros (logs) de transacciones.
4. Registros (logs) de errores.
5. Matrices de Control.
6. Cuestionarios de Control interno.
7. Reconciliaciones bancarias.
8. Revisiones independientes.
9. Controles de seguridad de acceso lógico.
10. Informes estadísticos y de excepción.
11. Reconciliaciones manuales o automáticas.

Todos los controles citados anteriormente no pueden ser ofrecidos directamente por el sistema, pero alguno de ellos sí que pueden formar parte de las soluciones de comercio electrónico. De esta manera, se facilita la labor de seguimiento de las actividades no sólo para los informes de auditoría, sino para la evolución continua del sistema de información y sus transmisiones de datos internas y externas.

Para asegurar que se cumple de manera adecuada tanto con las disposiciones legales como con el funcionamiento adecuado del sistema, se muestran a continuación las fases adecuadas para la elaboración de un plan de seguridad:

1. Preparación de un Plan.
2. Identificación y valoración de los activos.
3. Identificación de amenazas.
4. Análisis de riesgos.
5. Ajuste de los controles.
6. Preparación del informe.

Buenas prácticas en XML.

Desde el gerente de la empresa hasta el último implicado en el proceso de implementación de soluciones XML deben de seguir una serie de pautas concretas. Gartner Inc, con más de 20 años analizando las tendencias y tecnologías dominantes en los negocios, ofrece 11 consejos a seguir a la hora de implementar soluciones de comercio electrónico con XML.

1. Utilizar grupos reducidos. ¿Quién necesita aprender XML?. Lo que la mayoría de la gente debe aprender es cómo se definen los datos para su uso en transacciones electrónicas. Y sólo un pequeño grupo de expertos debe decidir si los modelos existentes son suficientes para el desarrollo del negocio, o es necesario implementar otros nuevos.

2. Entender el núcleo de XML. No sólo se trata de comprender cómo crear entidades, sino ser capaz de relacionar el resto de estándares de la W3C, como los relacionados con la creación de hojas de estilo, enlaces y demás.

3. No introducirse sólo. Los expertos pueden hacer que la curva de aprendizaje se reduzca y evitar errores innecesarios. Las experiencias previas que pueden haber tenido con otros clientes pueden ser mucho más que valiosas.

4. Cooperar con competidores. Lo contrario es perjudicial tanto para los clientes - que deben reconocer el uso de estándares diferentes -, y para la empresa, que frena el desarrollo de los estándares al intentar esconderlos o convertirlos en propietarios.

5. Compartir con organizaciones adyacentes. El objetivo es la integración en la cadena de suministro y en la cadena de información necesaria para el desarrollo del negocio.

6. Incluir seguridad desde el principio. Es un requisito de cualquier tipo de transacción. El autenticado, la encriptación, la autorización y la no repudiación se deben garantizar en cada uno de los intercambios.

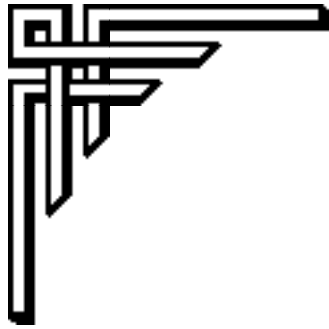
7. Apoyar los registros públicos. Los vocabularios de XML necesitan recursos para ser desarrollados y para gestionarse. Por ello, en la medida en que esto sea posible, se facilitará la rápida evolución y convergencia del estándar hacia objetivos comunes donde todo el mundo pueda sentirse partícipe.

8. Compartir vocabularios, pero no transacciones. Las transacciones son mucho más fáciles de manejar que los vocabularios. Por ello, es necesario estar de acuerdo en el lenguaje que se debe utilizar, para posteriormente adecuar las transacciones al mismo.

9. No discutir sobre los nombres. A los ordenadores no les importa la manera en que denominemos las cosas. A veces se pierden enormes cantidades de tiempo para acordar un nombre común. Es preferible invertir dicho tiempo en el desarrollo, que en discusiones cuya importancia es relativa.

10. No puede hacerse todo de una vez. No es necesario transformar toda la empresa para adaptarla al negocio electrónico y a XML. Es aconsejable iniciar el proceso por aquellas áreas en las que se necesitan introducir cambios y, sobre todo, en las que permitan una mayor interacción con el resto de sistemas de organización.

11. Comprometerse a un largo camino. XML no es tan simple y limitado como se pretende hacer ver en algunas ocasiones. Quizás en su uso, pero no en su significado. Por ello, los pequeños cambios que se van produciendo en la actualidad no son más que el prelude de una era de transformación en las tecnologías de la información.



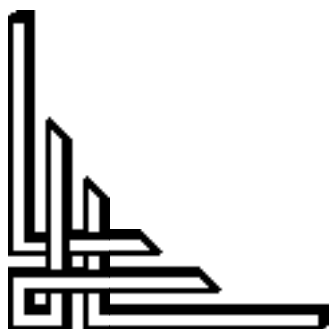
X

Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

M

CONCLUSIONES .

L



Desarrollar una solución de comercio electrónico exige un gran esfuerzo, tanto económico como humano, por parte de todos los componentes de una empresa, entendiendo por componentes tanto los distintos departamentos como las personas que los integran.

La solución nunca será viable si es acometida tan sólo por el departamento de informática. Aunque pueda ser uno de los principales actores en la ejecución del proyecto, es necesaria una colaboración y, sobre todo, una comunicación fluida y constante a nivel interno. Y por supuesto, a nivel externo cuando se implemente una solución en la que participen distintas organizaciones.

Como hemos visto a lo largo del proyecto, la técnica ha dejado de ser un problema en el momento en que se desea llevar a la práctica una idea de negocio electrónico. Dependiendo de las cualidades de la empresa, es decir, del nivel técnico y tecnológico de sus miembros, se puede optar por proceder al desarrollo de una herramienta propia o, por el contrario, optar por soluciones *prefabricadas* donde el mayor esfuerzo se centra en la adaptación de las mismas a la empresa.

El nacimiento del comercio electrónico en una empresa debe analizarse muy detalladamente. Por ello, sería ideal realizar un análisis completo, similar al que se pueda realizar con un plan de empresa o al añadir una nueva línea al negocio. Las preguntas claves que se deberían realizar son las siguientes:

1. *¿Es útil?*. Si nadie va a utilizarlo, no tiene sentido su desarrollo.
2. *¿Puedo hacerlo?*. En el caso de que la respuesta sea que no, se puede optar por una solución existente en el mercado.
3. *¿Se puede vender?*. Lógicamente, si no se van a obtener nuevos beneficios, es mejor no poner en marcha la idea.
4. *¿Quién lo va a financiar?*. La empresa deberá aportar el capital necesario para la puesta en funcionamiento del negocio, a la espera de obtener los frutos más adelante.
5. *¿Con qué apoyos cuento?*. Y no sólo financieros. Si toda la empresa no se involucra en el desarrollo de la solución, es difícil que tenga éxito. Es necesaria una colaboración global.

Hasta hace muy poco tiempo, EDI era la única solución posible a la pregunta de cómo implementar el comercio electrónico. Presentaba sus inconvenientes, pero si se deseaba implementar una herramienta de estas características, era fácil encontrar el modo de hacerlo.

Con XML se solucionan muchos de los problemas que presentaba EDI, sobre todo en su facilidad de implementación y bajo coste para pequeñas y medianas empresas. Pero, como es de esperar, también surgen problemas. Problemas por exceso de posibilidades.

BizTalk, ebXML o CXML son tres de las soluciones posibles, pero existen otras. La relativa facilidad con la que se cuenta para el desarrollo de nuevas plataformas de comercio electrónico ha hecho proliferar el número de las mismas. Por lo tanto, no sólo debemos buscar integrar la cadena de suministro con clientes o proveedores, sino que es necesario buscar la integración de las herramientas que permitan lo anterior. Se convierte en fundamental no sólo el hecho de favorecer el comercio electrónico y enlazar el *modus operandi* de las empresas, sino proceder a una integración tecnológica que permita extraer el mayor partido a las comunicaciones automatizadas. El siguiente paso al análisis de flujos de procesos debe ser el análisis de integración de tecnologías.

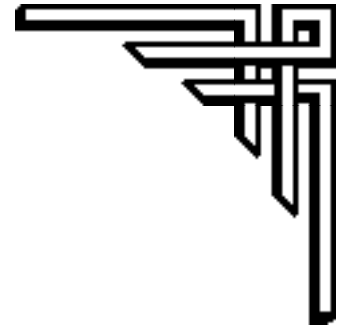
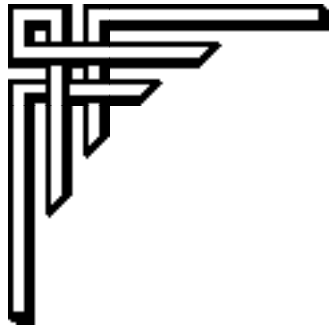
XML ofrece una gran versatilidad y desde un primer momento se ha ensalzado su capacidad integradora. La descripción de sus características así lo ha demostrado, pero no siempre debe de considerarse como la mejor solución. Es necesario saber que existe y que con ella podemos agilizar de forma extraordinaria la automatización de procesos de la empresa, pero es necesario evaluar la empresa con las preguntas formuladas en la página anterior. No se debe aplicar XML y el comercio electrónico porque forma parte de la moda actual, sino porque un buen análisis ha demostrado su viabilidad y por lo tanto se van a derivar beneficios de su uso.

Un buen punto de partida para realizar este tipo de análisis suele ser el seguimiento de empresas con experiencias previas en el sector o en sectores similares. Es importante ser innovador, pero sólo en el caso de tener garantías de que el esfuerzo valdrá la pena.

El último aspecto a tener en cuenta a la hora de tratar con soluciones de comercio electrónico tiene que ver, en cierta medida, con todo lo que no es puramente tecnología.

La legalidad obliga a tener un control sobre distintos aspectos del mismo, como puede ser la gestión de datos de carácter personal. Por otro lado, las auditorías exigen procedimientos de desarrollo y seguimiento adecuados y la seguridad de todo tipo de transacciones debe estar garantizada - no sólo a nivel técnico, sino físico. Es decir, tan importante es que no se descubra la información transmitida como que los procedimientos manuales sean gestionados por las personas adecuadas y designadas para ello.

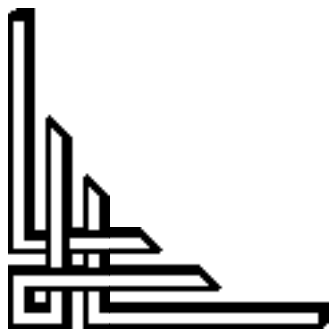
Para tener éxito en el comercio electrónico, al igual que en muchos otros aspectos de la empresa, es fundamental una buena dosis de experiencia. Como no siempre es posible disponer de la misma, es entonces necesario activar los mecanismos de búsqueda de información disponibles y fortalecer la idea con las experiencias previas de otro tipo de organizaciones. Y sobre todo, confiar en tecnologías abiertas e integradoras, como es XML.



X

M

L



Y
E
L
C
O
M
E
R
C
I
O
E
L
E
C
T
R
Ó
N
I
C
O

BIBLIOGRAFÍA.

[AGI, 1996] Auditoría de los sistemas de información.
Rafael Bernal Montañés, Óscar Coltell Simón.
Servicio de Publicaciones UPV. (SPUPV-96.019).

[B2B, 2001] Cómo desarrollar una estrategia de comercio electrónico.
Michael J. Cunningham.
Pearson Educación, México, 2001.

[BXML, 2000] Beginning XML.
David Hunter et alter.
Wrox Press Ltd, Octubre 2000.

[MTI, 2000] Marketing y tecnologías de la Información.
Ignacio Gil Pechuán / Daniel Palacio Marqués / Carlos Rueda Armengot
Servicio de Publicaciones UPV. (SPUPV-2000.4160).

[Portals, 1999] Building Corporate Portals With XML.
Clive Finkelstein et alter.
Computing McGraw-Hill, 1999.

[ProXML, 2000] Professional XML.
Didier Martin et alter.
Wrox Press Ltd, Abril 2000.

[STIG, 1997] Sistemas y tecnologías de la información para la gestión.
Ignacio Gil Pechuán.
McGraw-Hill/Interamericana de España, S.A.U. 1997.

[XML, 2001] The XML Handbook.
Charles F. Goldfarb, Paul Prescod.
Prentice Hall PTR, 2001.

[XMLAp, 1999] Building XML Applications.
Simon St. Laurent, Ethan Cerami.
McGraw-Hill, 1999.

[XMLRev, 2001] The XML Revolution.
Anders Moller, Michael I. Schwartzbach.
BRICS, University of Aarhus, Diciembre 2001.

White Papers - Documentos Técnicos.

AECE [<http://www.aece.org>].

1. Estudio sobre comercio electrónico B2B en España, 2001.
2. Impacto de la economía electrónica en las empresas europeas.

Baltimore Technologies [<http://www.baltimore.com>]

1. Security considerations for progressive e-corporations.

Comercio electrónico global [<http://www.e-global.es>].

1. Mecanismos alternativos para la resolución de los conflictos en infraestructuras de comercio electrónico.

cXML [<http://www.cxml.org>].

1. Guía del usuario cXML.
2. Fulfill examples.
3. InvoiceDetail examples.

DataMirror [<http://www.datamirror.com>].

1. Managing data the XML way.

ebXML [<http://www.ebxml.org>].

1. Business Process and Business Information Analysis Overview.
2. ebXML E-Commerce Patterns.
3. ebXML Catalog of Common Business Processes.
4. Business Process Analysis Worksheets & Guidelines.
5. ebXML Glossary.
6. ebXML Requirements Specification.
7. ebXML Registry Security Proposal.
8. ebXML Security.

ISACA [<http://www.isaca.org>].

1. Automated User Authentication.
2. Cibersecurity and the future of e-commerce.
3. Computer Forensics.
4. Fighting Security Breaches.
5. Securing Emerging Internet Applications.
6. Standard questions.

ITPapers [<http://www.itpapers.com>].

1. The XML Security Suite.
2. TIE Integrator.
3. The data administration newsletter.
4. ebXML: the global standard for electronic business.
5. XML: 11 best practices.

Microsoft [<http://www.microsoft.com/products/biztalk>].

1. BizTalk™ Framework 2.0: Document and Message Specification.
2. Microsoft BizTalk Customer Solution.

OASIS [<http://www.open-oasis.org>].

1. UBL: the next step for Global E-commerce.

Software AG [<http://www.softwareag.com>].

1. The XML Shockwave.

United Nations [<http://www.uncefact.org>].

1. Electronic Commerce on the Web.

World Wide Web Consortium [<http://www.w3.org>].

1. XML in 10 points.

XMLSolutions [<http://www.xmls.com>].

1. XML and EDI: peaceful co-existence.

Zapthink [<http://www.zapthink.com>].

1. The Pros and Cons of XML.

Documentos legales.

1. LOPD.

Ley Orgánica de Protección de Datos de Carácter Personal.
BOE num. 298, 14 de diciembre de 1999.

2. Reglamento de seguridad.

Reglamento de medidas de seguridad de ficheros automatizados que contengan datos de carácter personal.
BOE num. 151, 25 de junio de 1999.