INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS CIUDAD DE MEXICO

EDI Y XML EN EL COMERCIO ELECTRONICO ENTRE EMPRESAS

T E S I S

QUE PARA OBTENER EL TITULO DE:

MAESTRA EN ADMINISTRACION DE TECNOLOGIAS DE INFORMACION

P R E S E N T A:

LSCA KARLA PAOLA PULIDO DE SOLIS

MEXICO, D. F. 1999

INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS CIUDAD DE MEXICO

EDI Y XML EN EL COMERCIO ELECTRONICO ENTRE EMPRESAS

T E S I S

QUE PARA OBTENER EL TITULO DE:

MAESTRA EN ADMINISTRACION DE TECNOLOGIAS DE INFORMACION

P R E S E N T A:

LSCA KARLA PAOLA PULIDO DE SOLIS

MEXICO. D. F. 1999

A mi querido esposo, Gustavo, porque día a día me hace sentir la presencia de su gran amor que me fortalece en los retos que se me presentan y que constituye la alegría de mi alma.

A mis padres, Cris y Emilio, quienes, con su ejemplo, me enseñaron a ver horizontes lejanos, me infundieron el anhelo y la esperanza para luchar por alcanzarlos y me brindaron la palabra de aliento y de consejo, en momentos claves de mi vida.

A mi hermana Erika, quien con su entusiasmo por cada una de las cosas que emprende, ha sido el modelo a seguir en mi vida.

AGRADECIMIENTOS

Con mi agradecimiento para el Dr. Enrique Espinosa Carrillo, por sus consejos, tanto en mi carrera profesional como ahora en la maestría, y por su valiosa asesoría.

A Fabio Daniel, quien avivó en mi el deseo de aprender y superarme. Por haber creído en mi y por haberme alentado a estudiar esta maestría.

A Filiberto López Durán, por el imperecedero voto de confianza, apoyo moral y profesional, ejemplo recto y tenaz; mi eterno agradecimiento.

A mi familia, quienes a través de su cariño me han hecho sentir su apoyo.

Al Instituto Tecnológico y de Estudios Superiores de Monterrey por haberme dado la oportunidad de adquirir mis conocimientos profesionales y de maestría, y por haberme reforzado los grandes valores comprendidos en su misión.

A mi compañero, Francisco Javier Huerta López, por su gran apoyo.

A mis compañeros de maestría, en especial a Aurora Zenteno, Nancy Corlay, Juan Eligio Duarte, Roberto Murillo, Luis Enrique Cantú y Enrique Murillo, quienes con su ánimo colaboraron para hacer de ésta, una muy buena experiencia.

RESUMEN

EDI Y XML EN EL COMERCIO ELECTRÓNICO ENTRE EMPRESAS

Diciembre 1999

KARLA PAOLA PULIDO GUILLÉN

LICENCIADA EN SISTEMAS DE COMPUTACIÓN ADMINISTRATIVOS

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE

MONTERREY

CAMPUS CIUDAD DE MÉXICO

Dirigida por el Doctor Enrique David Espinosa Carrillo

El Comercio Electrónico negocio a negocio significa el remplazo de los documentos en papel que se intercambian entre compañías, tales como facturas, órdenes de compra, instrucciones de embarque, entre otras; así como la sustitución de los métodos comunes de transferirlos, como por ejemplo el fax o el correo electrónico, por una comunicación electrónica de información entre aplicaciones, minimizando al máximo la intervención humana y con ello la posibilidad de error.

Hasta ahora, las grandes empresas incursionaron en el Comercio Electrónico, entre otras tecnologías, mediante lo que se conoce como EDI (*Electronic Data Interchange*), que trabaja proporcionando una colección de mensajes estándar y su correspondiente diccionario de elementos de forma que "facilite" a los socios de negocio el intercambio de información vía un servicio de mensajes electrónicos.

El EDI parecía ser la solución, pero muchas de las empresas que lo han tratado de implementar, sobre todo las pequeñas y medianas, se han enfrentado a numerosos problemas -principalmente de costos- muchos de los cuales no han podido superar y que las han obligado a quedarse fuera de esta forma de conducir negocios, causando una baja penetración en el mercado del EDI; por lo que al final, las grandes empresas se han visto en la gran dificultad de realmente mejorar sus procesos al no poder incluir a todos sus socios comerciales.

Debido a estos problemas las empresas siguen en búsqueda de una solución que les permita comunicarse electrónicamente con cada uno de sus socios comerciales.

Una tecnología que aparentemente nos propone algo interesante es el XML, iniciales provenientes del inglés "*eXtensible Markup Language*". El XML provee un formato para describir datos estructurados, esto es, a diferencia del HTML no especifica la apariencia de los datos, sino su contenido en sí, es decir, el significado de los datos.

Recientemente se están llevando a cabo estudios y propuestas para, a través de XML, hacer el EDI accesible a un número mayor de usuarios. Así, el método principal de capturar y codificar la información EDI sería a través de formas electrónicas codificadas en XML. De esta forma los mensajes que realmente se intercambiarían serían en XML, y cada participante lo reconstruiría en el formato interno que maneje, ya sea EDI X12, EDIFACT o simplemente el mismo XML. El XML/EDI no es la creación de un nuevo estándar, sino la definición de cómo las compañías pueden utilizar los estándares actuales y atacar los problemas de una forma diferente.

El presente trabajo pretende analizar la situación existente de la tecnología EDI, los problemas y ventajas que provee en el ámbito de la tecnología de información nacional, plantear el porqué de la búsqueda de una nueva tecnología que permita la integración de todos los socios comerciales; y proponer, como solución a dicho problema, el uso de la tecnología XML.

Posteriormente, se hará referencia a los trabajos presentados en el Anexo 1 y 2 de este mismo documento, en donde se describe la construcción de dos traductores de

mensajes EDI X12 a XML y de XML a EDIFACT respectivamente, presentándolos como caso de estudio de la implantación de sistemas para la reconversión informática, permitiendo así la comunicación de dos entidades con estándares diferentes.

Por último, y con el fin de enriquecer el presente trabajo, se mostrarán los resultados de una encuesta realizada a empresas de México, sobre EDI, los problemas a los cuales se han enfrentado en su implementación y mantenimiento y la tendencia hacia las tecnologías basadas en Internet, permitiendo generar en el lector una idea del ambiente mexicano actual, respecto a dichas tecnologías.

INDICE DE CONTENIDO

Página

RESUMEN vi

INDICE DE CONTENIDO x

Capítulo

INTRODUCCIÓN 12

1. QUÉ ES EDI? 18

1.1 Nacimiento de los Estándares EDI 18

1.2 Arquitectura EDI 20

1.3 Porqué parecía interesante el utilizar EDI en lugar de papel? 24

2. FLUJO DE INFORMACIÓN EN UNA CADENA DE VALOR 26

2.1 Comercio Electrónico en una Cadena de Valor 26

2.2 Un ejemplo de un documento EDI. 28

3. PROBLEMAS Y EXPECTATIVAS NO CUMPLIDAS 38

3.1 Poca penetración en el Mercado 38

3.2 Implementando EDI sin cambiar los procesos 39

3.3 Staff y una VAN adicionales. 40

3.4 El estándar EDI no es realmente un estándar 41

4. ALGUNOS CAMPOS DE INNOVACIÓN TECNOLÓGICA EXPLORADOS .. 43

4.1 EDI to Fax 45

4.2 Pantallas de Captura e Intranet 46

4.3 EDI & Internet 48

4.3.1 Seguridad *51*

4.3.2 Integrar la información a las aplicaciones existentes. *52*

4.4 XML 52

4.4.1 Algunas reglas sencillas para la escritura de XML *55*

4.4.2 Beneficios de XML *58*

4.4.3 XML es autodescriptivo. *62*

5. XML/EDI 65

6. METODOLOGÍA DE INVESTIGACIÓN 69

7. RESULTADOS DE LA INVESTIGACIÓN 71

7.1 Resultados de la Encuesta. 71

7.2 Explicación del proyecto de programación. 83

8. CONCLUSIÓN 90

9. RECOMENDACIONES Y TRABAJOS FUTUROS 93

10. ANEXO 1. TRADUCTOR DE EDI X12 A XML 95

10.1 Componentes del traductor EDI2XML 95

10.2 El traductor y su funcionamiento 103

10.3 Manual de Usuario y Procedimiento para ejecutar el Motor "EDI2XML" 105

10.4 Código Fuente del Motor EDI2XML 114

11. ANEXO 2. TRADUCTOR DE XML A EDIFACT 121

12. BIBLIOGRAFÍA 132

INTRODUCCIÓN

Para poder llevar a cabo actividades de comercio, las empresas requieren intercambiar información, tal como órdenes de pedido, instrucciones de embarque, facturas, confirmaciones de reservación de espacio en buques, etc. Esto se hace normalmente por medio de comunicación oral (en persona o en teléfono) o en forma escrita (papel, fax).

La comunicación oral es viable cuando:

a) no se está transmitiendo mucha información, ya que es posible recordarla o anotarla rápidamente;

b) cuando ésta no tiene una estructura complicada;

c) cuando no se requiere de una evidencia de la existencia de dicha información, evitando malos entendidos, sobre todo si se trata de una relación comercial.

Cuando alguna de estas condiciones no se cumple, se requiere el intercambio de información escrita. Sin embargo, esto puede ser altamente costoso para volúmenes muy grandes de información (considerando el proceso de validación, proceso, distribución, almacenaje, etc., que se le da a dicha información); además de que en muchas ocasiones, ésta es transcrita en el lugar de destino, normalmente para cumplir los requisitos de sistemas del destinatario.

En la sección 2.1 de este trabajo, presentaremos un diagrama con el flujo de información que se intercambia en una operación logística típica. Como ejemplo, y situándonos en el contexto del Transporte Marítimo, tendríamos que, cuando una empresa solicita un espacio en el buque para transportar sus mercancías, proporciona datos tales como su razón social, datos de la empresa, itinerario deseado, nombre del buque, fecha de partida, etc. Esta información es recibida por la naviera, validada (si existe es itinerario, el buque, si existe espacio en el buque), y finalmente retecleada para que el sistema emita una confirmación de reservación, misma que le será devuelta al cliente con todos los datos que él proporcionó, más su número de confirmación. El cliente revisará dicha confirmación, si existiere algún error, nuevamente enviará los datos correctos a la naviera, hasta que ambos estén de acuerdo.

Es, a lo largo de todos estos procesos, que la información que fluye a través de la cadena de valor se va integrando. Usualmente todos los involucrados en la cadena van proporcionando algún dato extra que incrementa el valor de esta información [CORNELLA, 1999]. Emmelhainz nos dice que:

> El elemento clave para lograr la integración es el libre-flujo y la información compartida. [EMMELHAINZ, 1996]

Es decir, en la medida que los participantes de la cadena de valor, realmente compartan la información concerniente, existirá una mayor integración de la misma, que,

a su vez, permitirá aumentar la productividad y disminuir el tiempo de acción entre los miembros.

Existen muchas tareas a lo largo de dichas cadenas de distribución, en donde las herramientas de Comercio Electrónico pueden ser utilizadas para obtener una mayor productividad. De hecho, Kalakota y Whinston describen el COMERCIO ELECTRÓNICO como "una metodología de negocio moderna que cubre las necesidades de las organizaciones, comerciantes y consumidores para reducir costos a la vez que se mejora la calidad de los bienes y servicios y se aumenta la velocidad de entrega"[KALAKOTA, 1997]. El conocer la Cadena de Valor específica de una empresa (es decir los diferentes procesos involucrados desde el inicio de un producto o servicios hasta su distribución al usuario final, en donde intervienen los proveedores, la empresa y los clientes, quienes, en teoría, van aumentando el valor del producto o servicio en cada proceso), es lo que nos permitirá saber en dónde y cómo utilizar dichas herramientas para obtener resultados.

A su vez, Porter nos presenta el Comercio Electrónico como: "La colección de procesos, personas, tecnologías y medidas que guían las estrategias para alcanzar una ventaja competitiva en la economía emergente" [PORTER, 1997]

> "El Comercio Electrónico **negocio a negocio** significa el remplazo de los documentos en papel que se intercambian entre compañías, así como la sustitución de los métodos comunes de transferirlos, como por ejemplo el fax o el correo electrónico, por una comunicación electrónica de información, mejor conocida como EDI". [CEDEX, 1998]

Durante la década de los 80's, muchas de las grandes compañías, precisamente, emprendieron el camino hacia el Comercio Electrónico NEGOCIO A NEGOCIO implementando el Intercambio Electrónico de Datos o EDI. [BENÍTEZ, 1997], [CEDEX,1998], [CHUTE,1996].

Existen varias definiciones del EDI [BRYAN,1998], [CEDEX,1998], [JOCSON, 1999], pero partamos de la siguiente:

> "El EDI es el Intercambio de Documentos en una forma electrónica estandarizada, entre organizaciones, en una forma automática, directamente de una aplicación computacional dentro de una organización hacia otra aplicación en otra organización". [CLARKE, 1998]

Sin embargo, como veremos en el Capítulo 3, han descubierto que algunas expectativas alrededor del EDI, se quedaron sin cumplir debido, principalmente, a la falta de capacidad para incluir a todos los socios comerciales y con ello lograr una verdadera reingeniería de procesos [CHUTE, 1996], [BOTTS, 1996].

El objetivo del presente trabajo es describir la situación de la tecnología EDI, los problemas y ventajas que provee, plantear el porqué de la búsqueda de una nueva tecnología y proponer, como solución al problema que se describirá, el uso de la tecnología XML.

Partiremos de la idea de que para que las organizaciones logren una verdadera reingeniería de procesos y con ella una reducción de costos, deben poder integrar a todos

los socios comerciales a la cadena de valor y nuestra hipótesis se centra en el hecho de que con la tecnología XML esto es posible.

De esta forma, en el primer capítulo y a fin de brindar una visión de mayor amplitud, presentaremos el nacimiento de los estándares EDI, la arquitectura de dicha tecnología y los beneficios que prometía a sus suscriptores.

En el segundo capítulo, haremos referencia al flujo de información que puede existir en una cadena de valor, ejemplificando uno de los documentos que se intercambian en dicha cadena, con su formato EDI correspondiente.

Una vez situados en el concepto de EDI, pasaremos a revisar en el tercer capítulo los problemas y expectativas no cumplidas por dicha tecnología, para después, en el capítulo cuatro, mencionar brevemente algunas de los campos de innovación tecnológica que se han explorado, tratando de resolver los problemas mencionados.

En el capítulo cinco, abordaremos una posible solución al conflicto, presentando cómo es posible transferir documentos EDI a través de la nueva tecnología XML, y viceversa. Lo dicho representa una ventaja, si pensamos que aún sin la presencia de EDI, podríamos generar documentos XML directamente y enviarlos a una empresa que únicamente procese documentos EDI.

En el capítulo seis expondremos la metodología de investigación utilizada, tanto en el desarrollo de los capítulos anteriores, como en la encuesta realizada para conocer el entorno de la tecnología EDI en el ambiente mexicano.

En el capítulo siete, presentaremos los resultados de la encuesta y se explicarán brevemente algunos esfuerzos realizados para medir el impacto tecnológico de implantar software de reconversión informática, para lo cual se utilizó tecnología de compiladores en la construcción de dos traductores para comunicar a dos entidades, una con infraestructura EDI X12 y otra con EDIFACT, por medio del XML.

Por último, en el capítulo ocho y nueve, presentaremos las conclusiones y propondremos algunos tópicos para la realización de trabajos futuros, respectivamente.

1. QUÉ ES EDI?

1.1 Nacimiento de los Estándares EDI

> "En los 70's el Comité Coordinador de Datos del Transporte (*Transportation Data Coordinating Comitee o TDCC)* desarrolló el conjunto inicial de transacciones estándar para conducir electrónicamente los negocios del Transporte Aéreo, Motorizado, Ferrocarrilero y Marítimo". [CHUTE, 1998]

Los primeros en adoptar estos estándares fueron las navieras y los transportistas en general, ya que intercambiaban un gran volumen de documentos que además, requerían un esfuerzo manual significativo para ser procesados.

En los inicios del COMERCIO ELECTRÓNICO NEGOCIO A NEGOCIO, la prioridad se centró en la estandarización, tanto de los formatos como de la forma de envío y recepción de los mismos. El que se alcanzara un consenso general entre las industrias sobre el conjunto de documentos estándares que se utilizarían, constituyó el primer paso para el Comercio Electrónico entre negocios. Esto se entiende si pensamos que la forma de llevar a cabo los negocios era totalmente nueva y había que establecer en primer lugar una infraestructura común, un esqueleto sobre el cuál el comercio electrónico de negocio a negocio se pudiera fundar.

> "A principios de los 80's, la desregularización en la industria del transporte en los Estados Unidos clamó con mayor fuerza el uso de EDI para las grandes navieras. Los transportistas se vieron forzados al uso del EDI para competir en precio, además en calidad y en servicio." [CLARKE, 1998]

En la Figura 1, observamos como, la iniciativa del uso del EDI, tanto de los transportistas en general como de las navieras, permitió el crecimiento substancial del EDI a lo largo de los 80's y 90's.

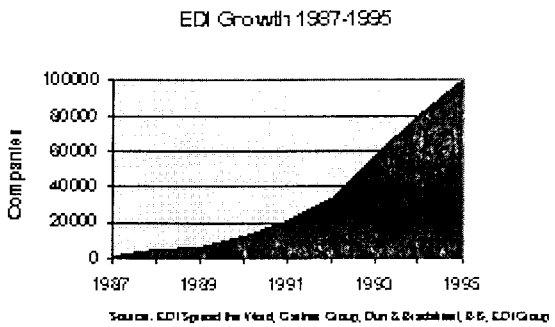


FIGURA 1. CRECIMIENTO DE EDI 1987-1995 [BOTSS, 1997]

En aquel tiempo, en que la tecnología computacional era cara y mucho menos avanzada que hoy en día, tratar con múltiples formatos propietarios para cada transportista constituía una barrera de costo-beneficio para el Comercio Electrónico.

> "En los principios de los 90's, los estándares de la TDCC fueron absorbidos y respaldados por el 'Comité de Estándares ANSI.X12'".[CHUTE, 1996]

> Por la misma época, pero en Europa, específicamente el Departamento de Aduanas de Inglaterra con la asistencia del *'Simplification of Trade Procedures Board'* (SITPRO), estaban desarrollando su propio conjunto estándar de documentos que se utilizarían en el intercambio internacional. A este conjunto se le llamó *'Tradacoms'*.
>
> Posteriormente los *Tradacoms* fueron ampliados por la Comisión Económica de las Naciones Unidas de Europa (UNECE) en lo que se conoció como *General-purpose Trade Data Interchange Standards (GTDI),* los cuales

fueron gradualmente aceptados por unas 2,000 organizaciones exportadoras inglesas.

Los problemas transatlánticos generados por estos dos conjuntos de documentos diferentes, y ampliamente incompatibles, fueron enfrentados con la formación de la Unión Europea y Norteamérica de las Naciones Unidas (UN-JEDI), que empezó a desarrollar el documento de traducción estándar *Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT).* [CLARKE, 1998]

1.2 Arquitectura EDI

El EDI puede compararse y contrastarse con el correo electrónico *(e-mail)*.

El correo electrónico es un sistema de mensajes como el correo tradicional que conocemos, que permite a una persona enviar un mensaje a otra pero a una dirección electrónica, que no es otra cosa más que una aplicación o buzón que almacenará dichos mensajes.

En el correo electrónico el formato de dicho mensaje es libre y se utiliza para transmitir mensajes electrónicamente de una PERSONA A OTRA *(person to person)*.

Ahora bien, El EDI también es un sistema de mensajes, por lo que también existen necesariamente un par de buzones electrónicos en donde serán depositados y almacenados dichos mensajes; pero, a diferencia del correo electrónico, los mensajes del EDI se intercambian ENTRE NEGOCIOS *(business to business)* y respetan un formato con una cierta estructura predeterminada.

Los elementos esenciales del EDI son: [CLARKE, 1998] [JOHNSTOWN,1999]

- Un medio de transmisión electrónico como una Red de Valor Agregado (VAN). Cabe aclarar que la denominación VAN se refiere al concepto organizacional, pero realmente el medio que se está utilizando es una WAN (Wide Area Network) o, recientemente, Internet (de las cuales hablaremos más adelante), en lugar de un medio de transmisión tradicional como puede ser las cintas magnéticas o los discos.
- Mensajes formateados estructuralmente basados en estándares acordados (para que esos mensajes puedan ser traducidos, interpretados y revisados con un conjunto explícito de reglas);
- Facilidades para la entrega relativamente rápida de documentos electrónicos del emisario al receptor (generalmente implica la recepción dentro de horas, o inclusive minutos); y
- La comunicación directa entre las aplicaciones (a diferencia de la comunicación que se lleva a cabo a nivel sólo de las computadoras, como sucede en el correo electrónico, que con que el mensaje llegue al buzón adecuado se considera exitosa, en cambio entre aplicaciones tendrá que hacerse la correspondencia de campos entre las dos bases de datos involucradas para considerarse exitosa).

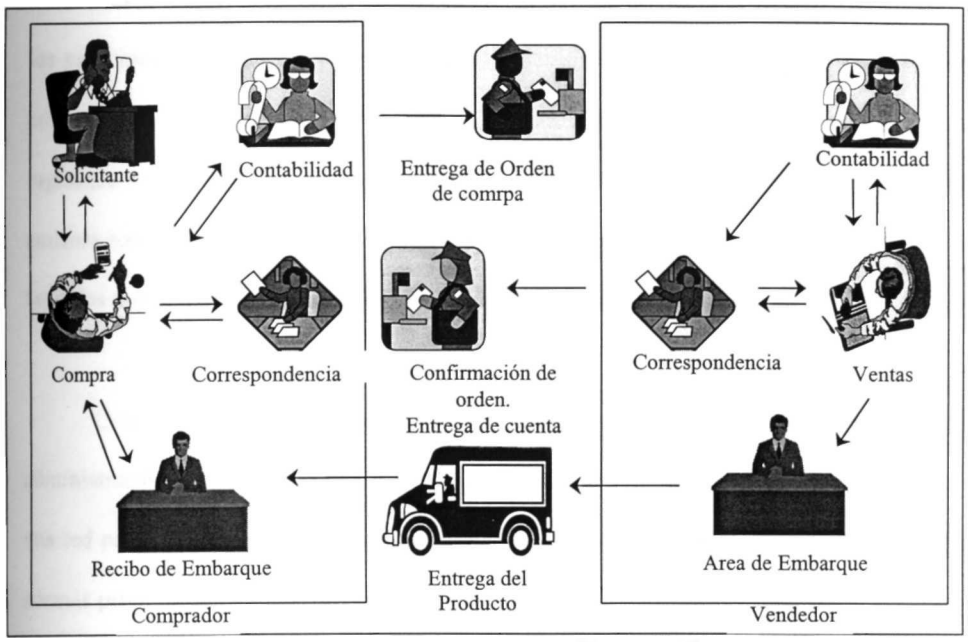


FIGURA 2A. CICLO PARA SURTIR UNA ORDEN SIN EDI [TURBAN, 1990]

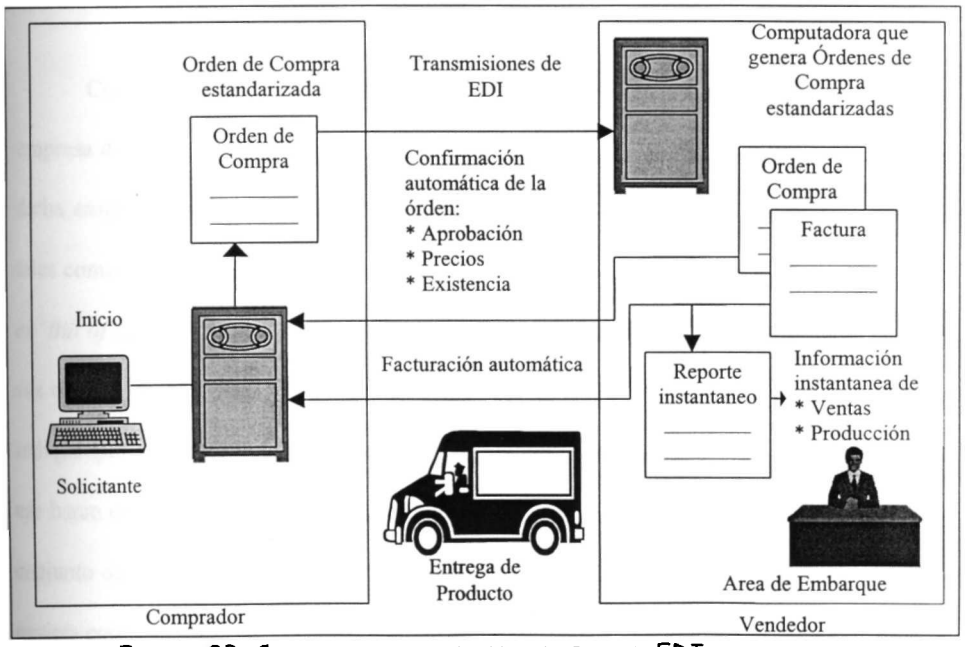


FIGURA 2B. CICLO PARA SURTIR UNA ORDEN CON EDI [TURBAN, 1990]

Como mencionamos, es necesario contar con un punto de conexión común entre los participantes, es decir (y haciendo referencia al correo electrónico), se debe contar con un par de buzones electrónicos. De esta forma, las computadoras de cada organización no se interrumpen mutuamente, ya que los mensajes de la computadora emisora son depositados en el buzón de la receptora, y ésta se conectará a su buzón para bajar los mensajes en determinado momento que sea considerado óptimo.

Generalmente, tanto los buzones como las cuestiones de seguridad y administración de comunicaciones son proporcionados por la VAN, que no es más que una red privada que ofrece estos elementos a los socios comerciales y en algunos casos, además proporciona un procesamiento extra de la información que es intercambiada entre los distintos socios comerciales [JOHNSTOWN,1999].

Como ejemplo de lo anterior podemos citar el servicio que ofrece una VAN a una empresa de transporte marítimo que se encuentra conectado a ella. Además del buzón de dicha empresa, la VAN ofrece el servicio de buzón a sus distintos socios comerciales, tales como puertos, aduanas y clientes importantes. Conforme la naviera vaya enviando el *"Bill of Lading"*, que no es otra cosa más que la factura por el transporte de su carga, a sus clientes, la VAN irá almacenando una copia de dichos documentos y cuando ésta le indique que el puerto ha quedado "cerrado" (es decir que ya no se acepta más carga para ese barco en ese puerto), la VAN emitirá un documento llamado "Manifiesto", que es el conjunto de todas las facturas que llevará ese barco en ese puerto, y lo enviará tanto a la naviera como al puerto correspondiente.

"La VAN provee los servicios de traducción cuando los estándares de los datos difieren entre los socios. Así mismo, provee la revisión del formato, seguridad, transmisión, reporteo y confirmaciones". [JENKINS, 1997]

1.3 Porqué parecía interesante el utilizar EDI en lugar de papel?

"El Comercio Electrónico negocio a negocio significa el remplazo de los documentos en papel así como la sustitución de los métodos comunes de transferirlos entre compañías, como por ejemplo el fax o el correo electrónico, por una comunicación electrónica de información, mejor conocida como Intercambio Electrónico de Datos, (EDI)". [CEDEX, 1998]

El 70 % de la información obtenida a través de un sistema computacional o 'salida' del sistema, se transforma en la 'entrada' de un segundo sistema. En un ambiente basado en papel, esto resulta en que la misma información es retecleada en ambas computadoras. Con el EDI esto desaparece, ya que la información es enviada de un sistema a otro [JOHNSTOWN,1999].

El EDI pretende AHORRAR RECAPTURA de información innecesaria permitiendo una transferencia de la misma mucho más rápida, con menos errores y mejorando, así, la fluidez de un proceso de negocio. [CLARKE, 1998], [BOTTS, 1996].

"El costo de procesar un solo documento en papel multiforma de principio a fin, está en el rango de $10 a $40 dólares americanos. El EDI puede reducir estos costos en una tercera parte y puede eliminar prácticamente una semana trabajo tomando en cuenta el tiempo involucrado por el sistema postal"[JOHNSTOWN,1999].

El Intercambio Electrónico de Datos MINIMIZA LA CANTIDAD DE PERSONAS involucradas en el manejo de la información entre la empresa y sus socios comerciales y ELIMINA LA CANTIDAD DE ERRORES Y RETRASOS que acompañan el procesamiento manual de los documentos. [JENKINS, 1997], [JOHNSTOWN,1999].

Otro factor que nos obliga a centrar la atención en el Intercambio Electrónico de los Datos es el TIEMPO. Hoy la rapidez es una de las armas más poderosas del mercado: quién lo piensa, quién lo crea y quién lo dice primero. Llevando las transacciones de una manera tradicional, la velocidad con que un papel es mandado, recibido, revisado, tecleado y procesado, es infinitamente mayor si se compara con el tiempo en que la misma información es procesada por medios electrónicos.

> "Para competir, hay que responder ágilmente a la demanda y, en la medida de las posibilidades, es necesario también, superar las expectativas. Para atender esta demanda, es preciso fijar objetivos comunes y diseñar estrategias conjuntas entre fabricantes y distribuidores". [ASOCIACIÓN, 1998]

El EDI tiene una característica importante: involucra necesariamente a los socios comerciales. Por tanto va más allá de las fronteras de la propia organización, al hacer posible la comunicación estándar entre los diferentes involucrados en la cadena de valor.

2. FLUJO DE INFORMACIÓN EN UNA CADENA DE VALOR

Para situar al lector en el contexto en donde se utiliza el EDI, veamos un aplicación típica de éste.

A lo largo de toda la cadena de valor existen diferentes oportunidades para reducir los costos, mejorar el servicio al cliente y aumentar la productividad. Dependerá del estudio y conocimiento que se tenga de la misma para saber en dónde y cómo implementar las tecnologías de comercio electrónico.

2.1 Comercio Electrónico en una Cadena de Valor

El Comercio Electrónico, de NEGOCIO A NEGOCIO, se refiere al intercambio de documentos tales como órdenes de compra, facturas, pagos electrónicos con y entre los diferentes involucrados en la cadena de valor. Además, incluye el hecho de proveer a los negocios los servicios de *e-mail* y *bulletin boards*, facilitando la diseminación de la información y mejorando la comunicación entre los socios comerciales. [TOPKEN, 1998]

En la Figura 3 presentamos un ejemplo [TOPKEN, 1998] del flujo de información en una transacción logística común. En él se observan los diferentes tipos de documentos que pueden ser intercambiados entre los distintos actores en una cadena de valor, en donde interviene un proveedor, un transportista, un centro de distribución y un pequeño comerciante.

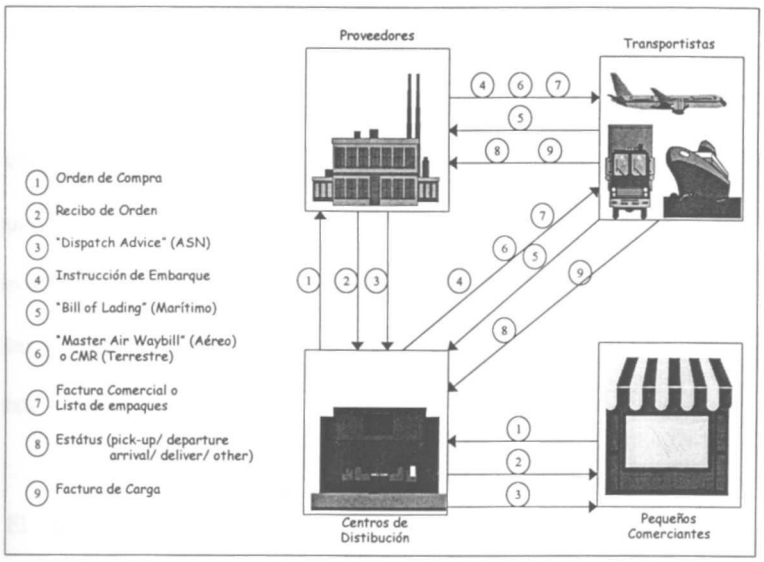


FIGURA 3. FLUJO DE DATOS DE UNA OPERACIÓN LOGÍSTICA

Para cada uno de estos documentos, existe un "documento estándar" en EDI; por ejemplo, la "Instrucción de Embarque" está representado por el documento EDI 304 (ANSI X12), mismo que si todos los involucrados adoptaran, sería susceptible de transmitirse entre las distintas aplicaciones de los involucrados SIN la intervención humana, ya que el software de traducción y mapeo, haría la correspondencia campo a campo del documento EDI con los campos de nuestras bases de datos internas.

De esta forma, crearíamos el documento "Bill of Lading" tomando como base los datos que recibimos en la "Instrucción de Embarque", utilizando el documento EDI 310, y transmitiéndolo a los diferentes involucrados, quienes a su vez deberían ser capaces de procesarlo sin la intervención humana.

2.2 Un ejemplo de un documento EDI.

Veamos a continuación un ejemplo de un documento EDI, "La Instrucción de Embarque", denominada 304 (ANSI X12). Este documento es enviado usualmente, de los centros de distribución o proveedores al transportista marítimo. Su objetivo es el comunicar toda la información necesaria para que la naviera pueda preparar y distribuir un contrato de carga impreso (que será el *"Bill of Lading"*).

El mensaje está compuesto por un "Encabezado" *(Header)* el cual está representado en la Figura 4, un "Detalle" *(Detail)* que observamos en la Figura 5, y, por último, en la Figura 6 presentamos el "Resumen" *(Summary)*. Cada uno de los cuales a su vez está compuesto por uno o varios segmentos, según vemos en la Figura 7, de los cuáles algunos se podrán repetir una o varias veces.

Los segmentos pueden ser Mandatorios (marcados con la letra M) u Opcionales (marcados con la letra O), es decir, que los de éste último tipo pueden o no ser incluidos en el mensaje.

"Header

Segment ID	Name	M/O	Max. Use	Loop Repeat
ST	Transaction Set Header	M	1	
B2	Beginning Segment for Shipment Information Transaction	M	1	
B2A	Set Purpose	O	1	
Y6	Authentication	O	2	
G1	Shipment Type Information	O	1	
G2	Beyond Routing	O	1	
G3	Compensation Information	O	1	
N9	Reference Number	O	100	
V1	Vessel Identification	O	2	
V3	Vessel Shcedule	O	1	
M0	Letter of Credit Reference	O	1	
M1	Insurance	O	5	
M2	Sales/ Delivery Terms	O	1	
C2	Bank ID	O	1	
C3	Currency	O	1	
N1	Name	M	1	10
N2	Additional Name Information	O	2	
N3	Address Information	O	2	
N4	Geographic Location	O	1	
G61	Contact	O	3	
R4	Port	M	1	20
DTM	Date/ Time Reference	O	15	
R2	Route Information	O	13	
K1	Remarks	O	12	
L11	Business Instructions	O	99	
H3	Special Handling Instructions	O	6	
L5	Description, Marks and Numbers	O	999	

FIGURA 4. ENCABEZADO DEL DOCUMENTO EDI X12 304

Detail

Segment ID	Name	Req. Des.	Max. Use	Loop Repeat
LX	Assigned Number	M	1	999
Y2	Container Details	O	10	
N7	Equipment Details	O	1	999
QTY	Quantity	O	1	
N12	Equipment Environment	O	1	
M7	Seal Numbers	O	5	
W09	Equipment and Temperature	O	1	
L1	Rate and Charges	O	1	20
C3	Currency	O	1	
L4	Measurement	O	1	
L7	Tariff Reference	O	1	
X1	Export License	O	1	
X2	Import License	O	1	
N9	Reference Number	O	3	
H1	Hazardous Material	O	1	10
H2	Additional Hazardous Material Description	O	10	
K1	Remarks	O	10	
L0	Line Item - Quantity and Weight	O	1	120
MEA	Measurements	O	10	
L5	Description, Marks and Numbers	O	999	
L12	Altemate Lading Description	O	20	
L1	Rate and Charges	O	1	20
C3	Currency	O	1	
L4	Measurement	O	1	
L7	Tariff Reference	O	1	
X1	Export License	O	1	
X2	Import License	O	1	
C8	Certifications and Clauses	O	10	
H1	Hazardous Material	O	1	10
H2	Additional Hazardous Material Description	O	10	

FIGURA 5. DETALLE DEL DOCUMENTO EDI X12 304

Summary

Segment ID	Name	Req. Des.	Max. Use	Loop Repeat
L3	Total Weight and Charges	M	1	
MEA	Measurements	O	5	
PWK	Paperwork	O	25	
SUP	Supplementary Information	O	1	
L1	Rate and Charges	O	1	20
C3	Currency	O	1	
V9	Event Detail	O	10	
K1	Remarks	O	999	
L11	Busienss Instructions	O	24	
SE	Transaction Set Trailer	M	1	

FIGURA 6. RESUMEN DEL DOCUMENTO EDI X12 304

La columna marcada con "*Max. Use*" significa que el número máximo de veces que puede aparecer ese segmento en el documento, o si el segmento se encuentra en un *loop*, indica el número máximo de veces que puede aparecer en cada uno de los *loops*.

La columna que está nombrada como "*Loop Repeat*" significa que los segmentos contenidos dentro de esa señalización, se pueden repetir un número *n* de veces dentro del mismo mensaje. Por ejemplo los segmentos R4 (*Port*) y DTM (*Date/ Time Reference*) se encuentran dentro de un *loop* marcado con un 20, esto significa que se pueden incluir 20 puertos distintos con su horario asociado, de tal forma que la ruta entera que hará el barco con la mercancía sea especificada en un sólo documento.

Cada segmento se compone a su vez de uno o varios elementos de información [ISA, 1998], por ejemplo:

Segmento	ST Transaction Set Header
Nivel	Header
Loop	----------
Uso	Mandatorio
Máximo uso	1
Propósito	Indica el principio de un conjunto de transacciones y asigna un número de control.

Sumario de Elementos de Información

Ref. Des.	Elemento de Información	Nombre		Atributos	
ST01	143	Transaction Set Identifier Code Code uniquely identifying a Transaction Set.	M	ID	3/3
ST02	329	Transaction Set Control Number Identifying control number that must be unique within the transaction set funcional group assigned by the originator for a transaction set.	M	AN	4/9

FIGURA 7. ELEMENTOS DE INFORMACIÓN DEL SEGMENTO ST

Así mismo para cada uno de los segmentos. El software de traducción, será el responsable de hacer la conexión entre los campos que se mencionan en el documento EDI y los campos de las aplicaciones internas, de tal forma que los datos sean equivalentes a aquellos en las aplicaciones y viceversa.

Ahora bien, veamos a continuación los primeros segmentos de un ejemplo real con su explicación asociada:

Segmento	Explicación
ST*304*24680 N/L	En este segmento como se mencionó, se tienen dos elementos de información: • ST01 -> El identificador del documento EDI que se debe utilizar. • ST02 -> Un número de control. Cada elemento de información estará separado por un asterisco. En este ejemplo, el número del conjunto transaccional o documento EDI que se debe utilizar es el 304, es decir que nos está indicando que es una "Instrucción de Embarque". El número de control es 24680. Las siglas "N/L" que encontraremos al final de cada segmento significan *"new line"* , y funcionan como el separador entre segmentos.
B2*PP*ABCU****PQR24 67*L*CC**B******1 N/L	En seguida, tenemos el segmento B2, que contiene 18 elementos de información, por lo que, en principio, debemos contar con 17 asteriscos para separar a cada uno de dichos elementos. Si dos o más asteriscos se

Segmento	Explicación
	encuentran seguidos uno del otro, significa que ese elemento de información va vacío, muy probablemente porque fuera opcional y en dicho documento no se está incluyendo esta información.
B2*PP*ABCU****PQR24 67*L*CC**B******1 N/L	Se pueden omitir los asteriscos siempre y cuando no se vaya a incluir información en algún elemento de información posterior, por ejemplo, después del 1 que vemos aquí, existen dos elementos de información más, pero como son opcionales y no se van a incluir, ya no se agregan los asteriscos, en cambio como el 1 sí se iba a incluir, se tuvieron que poner los 6 asteriscos anteriores para indicar que los cinco elementos de información anteriores al 1, van vacíos. Ahora bien, este segmento significa que: El segmento se llama B2 El código de servicio de tarifa es PP, que significa puerto a puerto. Enseguida está el código estándar del transportista, que en este ejemplo es ABC Line. Después vemos 4 asteriscos que nos indican que se han omitido los siguientes tres elementos de

Segmento	Explicación
B2*PP*ABCU****PQR24 67*L*CC**B******1 N/L	información. Ahora aparece el número único de referencia de embarque que es PQR2467. La unidad de peso que se utilizará en el documento es L, que equivale a Libras. El método de pago será por cobrar, y este se representa con el código estándar CC. Después se está omitiendo un elemento de información por lo que aparecen dos asteriscos juntos. La B, significa que ese barco está haciendo el transporte único de la mercancía y que no es un traspaso (que sería marcado con la letra W) Enseguida se están omitiendo cinco elementos de información por lo que se presentan 6 asteriscos juntos. Ahora vemos un 1, éste indica el número total de contenedores que se incluyen en el documento. Restan dos elementos de información, pero como no se van a incluir y son los últimos, se omiten los asteriscos y únicamente se cierra el segmento con su respectivo N/L.

Segmento	Explicación
B2A*00 N/L B2A*00 N/L	En seguida tenemos al segmento B2A, cuyo fin es establecer el propósito del documento. Este segmento tiene dos elementos de información, el primero, que aquí tiene el valor de 00, se utiliza para especificar si el documento es un original o una copia, donde el 00 significa que es un original. El siguiente elemento de información es para incluir un código del tipo de aplicación, pero en este caso no se utiliza y como es el último, se omiten los asteriscos y únicamente se cierra el segmento con su respectivo N/L.
N9*BN*SF12345 N/L	Posteriormente aparece el segmento N9, ya que los segmentos Y6, G1, G2 y G3, son opcionales y para este ejemplo no se están empleando. Este segmento se compone de 5 elementos de información. El primero indica de dónde se está obteniendo el número de referencia, (puede ser del número de reservación, del número de orden del consignatario, del número de referencia del agente aduanal o el

Segmento	Explicación
N9*BN*SF12345 N/L	número asignado por la naviera). En este ejemplo aparece el código estándar BN, que significa que se utilizará el número de booking como referencia. En seguida aparece el número de referencia. Posteriormente se podría haber incluido tres elementos de información más, texto libre, la fecha y la hora, pero como se omitieron y son los últimos, no se incluyen los asteriscos y únicamente se cierra el segmento con su respectivo N/L.

3. PROBLEMAS Y EXPECTATIVAS NO CUMPLIDAS

Aunque el EDI parece resolver muchos de los problemas del intercambio de documentos por métodos tradicionales, las compañías que lo han implementado han encontrado ciertos problemas tanto para implementarlo como para mantenerlo. Así mismo, han encontrado puntos que prometía resolver y no lo hace.

3.1 Poca penetración en el Mercado

El EDI, como muchas otras tecnologías, requieren grupos de personas que las adopten antes de que sean realmente útiles para alguien, y crecen muy poco a poco en un principio. Aún si la tecnología ha estado ahí por un largo tiempo y los beneficios potenciales sean muy obvios, es particularmente difícil que éste tipo de innovaciones se implementen rápidamente. Para que alguien se beneficie, prácticamente todos tienen que adoptar la tecnología en cuestión, por lo tanto tendrían que existir muchas compañías capaces de manejar EDI, para que así cada una de éstas contara con muchos socios comerciales EDI y la inversión y el esfuerzo valieran la pena.

El gran costo que implica traducir los formatos nativos a estándares EDI ha mantenido a muchas compañías medianas y chicas fuera del comercio electrónico. Esto impide a dichas compañías disfrutar de los beneficios potenciales de automatizar los procesos de negocio manuales.

"Además de esto, los planes de implementación de EDI de muchas grandes organizaciones habían sido frenados por la imposibilidad de incluir a los pequeños socios comerciales en el EDI. Muchas organizaciones grandes afirmaban que podían automatizar sólo el 60-80% del volumen de sus transacciones. Más aún, el 20-40% de las transacciones manuales restantes eran, usualmente, aquellas que tradicionalmente habían sido las más complejas y que requerían de el mayor esfuerzo para ser procesadas". [CHUTE, 1996]

Las pequeñas y medianas empresas pueden producir documentos electrónicos actualmente pero se resisten a "invertir " en los estándares EDI por temor a que los costos de ello sobrepasen los beneficios potenciales.

3.2 Implementando EDI sin cambiar los procesos

"Muchas implementaciones de EDI únicamente automatizaron el flujo de información entre los socios comerciales, mientras que las actividades de proceso ineficientes y/o manuales permanecieron tanto para el destinatario como para el remitente". [CHUTE, 1996]

Esta ineficiencia llega a tal grado que, muchas compañías simplemente enviaban sus transacciones EDI entrantes a la impresora o las retecleaban, mientras el proceso manual se mantenía intacto. Al realizar esto, los "beneficios" prometidos de EDI se están eliminando automáticamente.

3.3 Staff y una VAN adicionales.

Las compañías que incursionaron en EDI tuvieron que contratar un staff dedicado a traducir la información de su formato nativo al estándar EDI y a mantener los mapas "personalizados" que, a lo largo de la operación del EDI y mientras más socios comerciales se unían a la comunidad, eran más numerosos. Estos mapas personalizados se hacen necesarios en el momento que un par de socios comerciales decide "IMPLEMENTAR SU INTERPRETACIÓN DEL ESTÁNDAR" y con ello las excepciones y casos particulares se presentan, por tanto un "NUEVO" mapeo de los campos, necesita ser programado, como mencionaremos en el siguiente punto.

Muchas de ellas, además, utilizaron VANs en lugar, o además, del staff dedicado. Una VAN, como mencionamos, establece un buzón electrónico para cada socio comercial. El emisor transmitirá la transacción electrónicamente hacia la VAN vía módem o línea telefónica.. La red recibe las transacciones, las ordena por receptor, y las almacena en los correos correspondientes de los receptores hasta que son recogidas por los mismos. Una VAN además provee controles de seguridad para asegurar la integridad de los datos. Cada empresa con una cuenta en una VAN tendrá una clave que deberá utilizar para accesar su buzón y recuperar los mensajes. La encripción de los datos es otro ejemplo de los controles de seguridad que provee una VAN. Además de esto, una VAN ofrece el servicio de traducción entre estándares y provee un sistema de respaldo y almacenamiento de la información contenida en todos los buzones, y ofrece, así mismo, el servicio de detección y corrección de errores [JOHNSTOWN, 1999]

Desafortunadamente, todos estos beneficios tienen un alto costo. Estas VANs no sólo son caras desde el punto de vista del hardware y del software, sino por el cobro que se exige por la transmisión de datos y la comunicación con la VAN, donde los cargos se realizan en base al número de transacciones o comunicación. [CONE, 1998], [GUIDE,1998], [MAYOR,1995]

3.4 El estándar EDI no es realmente un estándar

Hoy en día, la mayoría de los microprocesadores que se encuentran en los escritorios pueden procesar fácilmente grandes volúmenes de transacciones. Además, existe software que puede manejar fácil y económicamente reglas de negocio únicas para cada par de socios comerciales, es decir, que dependiendo de una serie de factores que se especifiquen, (por ejemplo, dependiendo de quién es el remitente y quién el destinatario, o bien, si el documento manejado es el original o una copia, etc.) realizan tal o cual acción sobre el mensaje enviado.

A través de los años, cada empresa que finalmente pudo implementar el EDI, ha desarrollado y documentado su propia interpretación del "estándar" y espera que cada socio comercial se alinee a él.

> "Las discrepancias son resueltas en favor del socio con el poder económico más grande en la relación (usualmente el cliente). El resultado final es que los proveedores típicamente tienen que mantener código único para cada socio comercial". [CHUTE, 1996]

Cuando los estándares fueron primeramente establecidos, alrededor de los años 70's, , el ciclo de vida de los productos era medido en años o décadas en lugar de semanas o meses como hoy en día, y la velocidad de cambio era más lenta. [ETXEBARRIETA, 1995], [NOVELLA, 1999], [RAMAN2, 1999]

En el ambiente de competitividad actual las compañías deben lanzar nuevos productos y servicios en periodos de tiempo muy cortos en ciertas Industrias.

Alan Chute en su artículo "The Mytical Value of EDI Standards" hace la siguiente reflexión al respecto:

> "¿Se puede usted imaginar a *Federal Express* esperando a la siguiente conferencia de ANSI.X12 para lanzar su servicio "*First Overnight*" por que no existía el código de servicio para reflejarlo en un documento estándar EDI? Mientras que este fenómeno es más pronunciado hoy en día es posible pensar que por lógica, los competidores en cualquier Industria se endosaran los conjuntos de transacciones "estándar". Los competidores siempre han buscado oportunidades para diferenciarse entre sí lo que causa una inclinación natural a alejarse del estándar de la Industria. ¿Que tanto podría sobrevivir en el ambiente competitivo actual ofreciendo productos y servicios estándares?" [CHUTE, 1996]

4. ALGUNOS CAMPOS DE INNOVACIÓN TECNOLÓGICA EXPLORADOS

Para lograr sobrevivir en el mundo competitivo actual, las empresas deberían buscar eliminar uno a uno de los problemas presentados:

- En lugar de enfatizar el estándar EDI como formato, buscar la manera que sea más conveniente para cada par de socios comerciales de enviar y recibir la información. El éxito residiría en la capacidad que tengamos de lograr que no importando el formato y el medio de transmisión que utilicen nuestros socios comerciales, pudiéramos procesar de manera automática la información recibida.

- Analizar y reevaluar los procesos internos de la compañía a fin de realizar una reingeniería a los mismos, tomando en cuenta las interacciones que se llevan a cabo tanto con otras áreas funcionales como más allá de las fronteras de la empresa, en la cadena de valor completa, para efectivamente poder procesar automáticamente hasta la transacción más compleja o el socio comercial más pequeño.

Es importante considerar que no por el hecho de automatizar un proceso, significa que se mejore o se espere ganar productividad, es decir, se debe analizar el proceso en si mismo, practicarle una reingeniería y quizá

eficientarlo con las nuevas tecnologías, pero no cometer el error de pensar que por el hecho en si mismo, de implementar alguna nueva tecnología, el proceso se hace eficiente.

- Modificar o desarrollar aplicaciones en la empresa que estén basadas en las reglas de negocio mencionadas, para que estas actúen como el experto en el proceso, siempre y cuando sean fáciles de modificar y mantener.

Actualmente las empresas en general han empezado a buscar soluciones que les permitan la comunicación electrónica con todos sus socios comerciales y que eliminen por completo el proceso manual.

Las pequeñas y medianas compañías han incursionado en algunos campos de innovación tecnológica a fin de poder incorporarse a la comunicación electrónica de datos con sus socios comerciales grandes, como son:

- EDI to fax (*Paper-based Service Bureau*) [HAISTING, 1996]
- Pantallas de captura e Intranet (*Intranet Service Bureau*) [HAISTING, 1996] [BROWN, 1999]
- EDI & Internet [BROWN, 1999]
- XML

4.1 EDI to Fax

Este servicio permite a las pequeñas empresas comunicarse a través de un tercero, llamado CENTRO DE SERVICIO EDI o *Service Bureau*, sin tener que realizar una inversión significativa en la tecnología EDI en cuanto a costo y tiempo. Veamos la gráfica en la Figura 8:

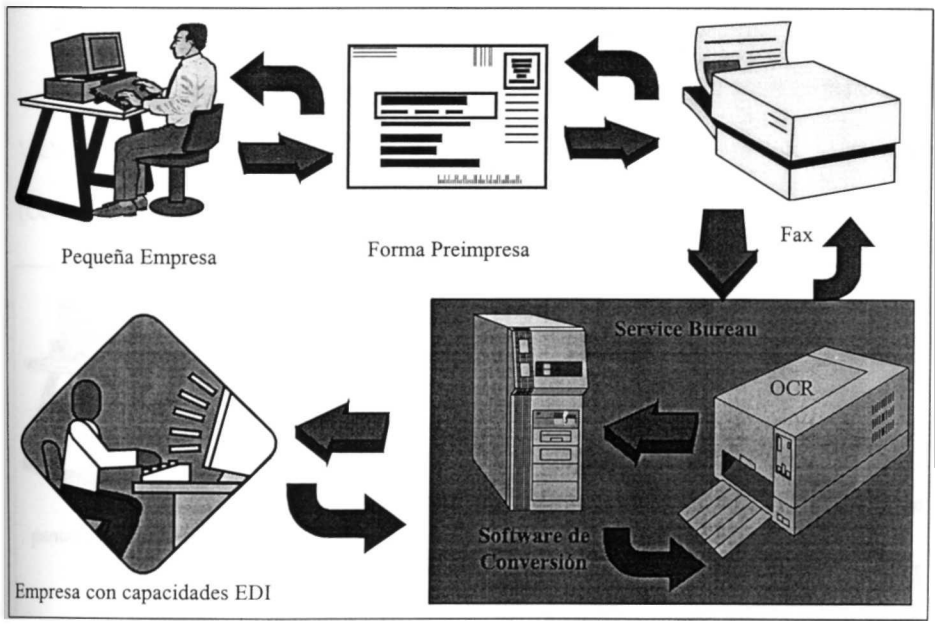


FIGURA 8. EDI TO FAX

El servicio consiste en que la compañía llena una forma preimpresa y la manda por fax al centro de servicio, quien a través de un dispositivo OCR lee los datos de esa hoja preimpresa y utilizando un software de conversión, los transforma en un archivo electrónico, cuyo formato es el de un mensaje EDI.

Más tarde el centro de servicio envía dicho mensaje al receptor (compañía con capacidades EDI) quien lo recibe como si el emisor lo hubiera mandado utilizando la tecnología EDI. Éste lo procesa y contesta al centro de servicio a través de un mensaje EDI. El centro de servicio a la vez transforma dicho mensaje ya sea en un archivo plano o en un fax para el remitente.

4.2 Pantallas de Captura e Intranet

En este servicio, como se aprecia en la Figura 9, la pequeña compañía utiliza un Web Browser que se comunica con el Centro de Servicio desde una línea intranet (Conexión cerrada entre el usuario y el centro de servicio).

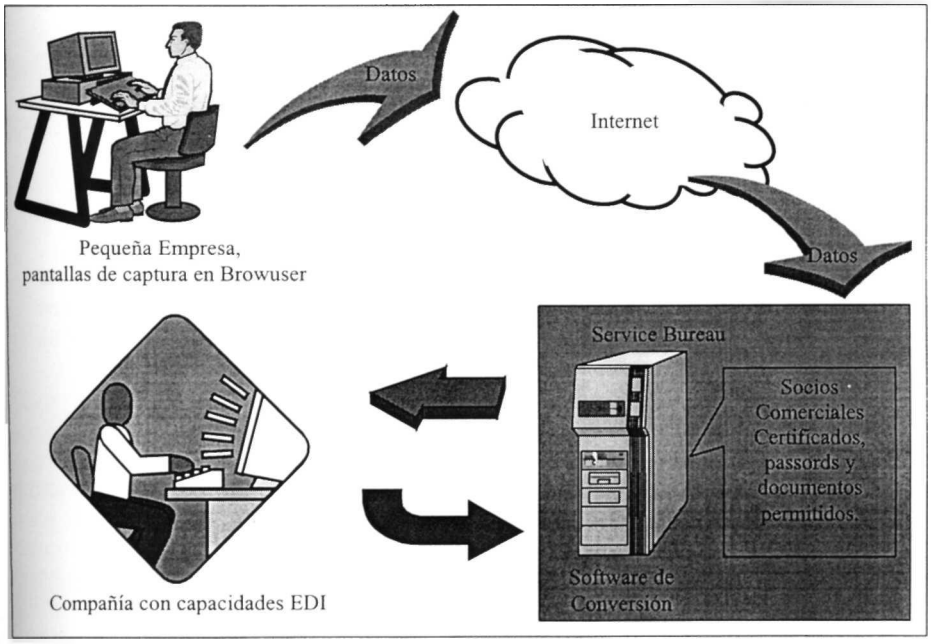


Figura 9. PANTALLAS DE CAPTURA E INTERNET

Los documentos son creados en pantallas de captura y únicamente los datos son transmitidos al centro de servicio, quien a su vez, transforma dichos datos a un formato EDI y los envía utilizando toda la metodología estándar y segura de EDI.

De esta forma las pequeñas compañías se evitan la inversión, la curva de aprendizaje de la tecnología y los problemas de integrar los datos a sus aplicaciones, ya que el bureau es quien se encarga de formatear los datos según convenga.

El centro de servicio mantiene una lista de socios comerciales certificados, *passwords* y documentos permitidos de los diferentes *hubs*.

> "En cuanto al Comercio Electrónico se refiere, el mayor auge que ha tenido Internet en este aspecto se centra en las transacciones *cliente-negocio*. Esto se refiere a individuos comprando bienes directamente del fabricante y pagando por ellos a través de un browser de Internet". [HAISTING, 1996]

Pero existen otros nuevos conceptos que aprovechan las ventajas de la Internet como medio de transmisión de documentos de NEGOCIO A NEGOCIO, haciendo el movimiento de información más sencillo, rápido y potencialmente más económico que el tradicional EDI.

La transferencia electrónica de datos usando la Internet está prevaleciendo y es mucho más económica que los métodos tradicionales de transmisión de transacciones EDI tanto directamente como a través de una Red de Valor Agregado.

> "La industria del transporte ha sido reconocida como la pionera en el intercambio electrónico de datos y se ha convertido en uno de los primeros nichos para integrar la Internet con el tradicional EDI, ayudando a la expansión del Comercio Electrónico". [HAISTING, 1996]

La Internet podría ser una alternativa para proveer al socio comercial pequeño la habilidad para transmitir ciertos documentos EDI con un costo relativamente pequeño de entrenamiento.

En la Figura 10, se presenta una gráfica del crecimiento que ha tenido (y que tendrá) el Comercio Electrónico negocio a negocio, en Estados Unidos. [FORRESTER, 1999]

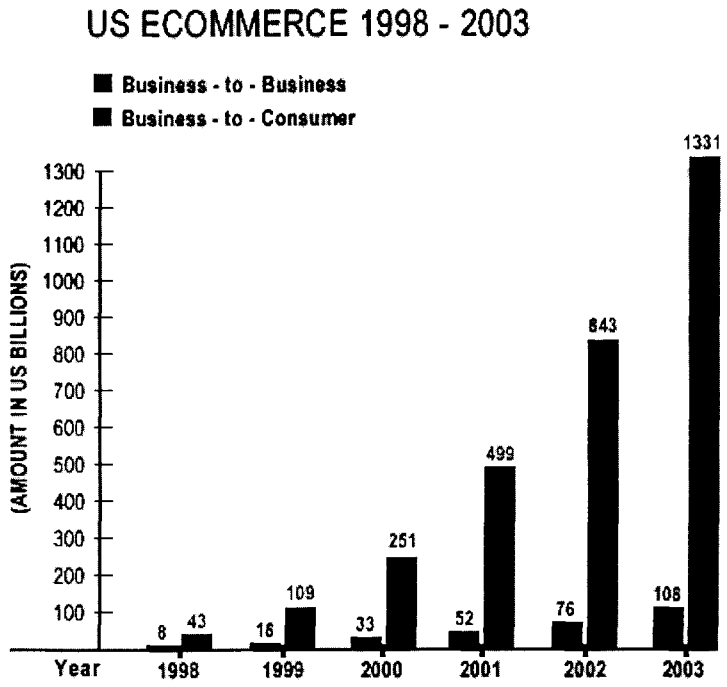


FIGURA 10. CRECIMIENTO DEL COMERCIO ELECTRÓNICO EN EUA, DE 1998-2003

Como podemos observar, Forrester predice un crecimiento muy alto en cuanto al Comercio electrónico negocio a negocio se refiere. No obstante, en la Figura 11[BOTTS, 1996], se muestra la disminución que las VANs han tenido en los últimos años en cuanto a la transmisiones EDI, por lo que esto nos lleva a concluir que las empresas están utilizando otros medios de transmisión, tales como Internet.

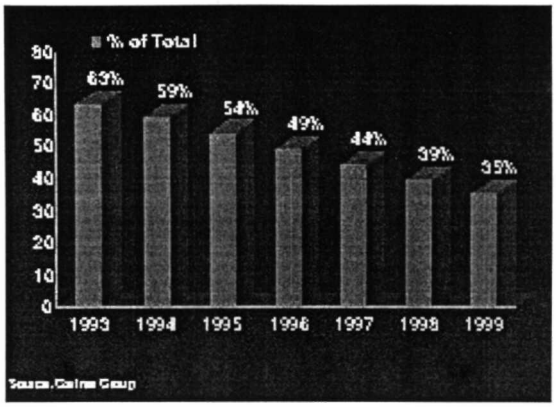


FIGURA 11. DISMINUCIÓN DE TRANSMISIONES EDI A TRAVÉS DE UNA VAN

En Julio de 1993 existían 130 sites de Internet. En Julio de 1995, el número ascendía a 38,800 sites. Actualmente, el número de sites se dobla cada cuarenta y ocho días, según David L. Taylor, vicepresidente y director de investigación de Gartner Group. Recientemente proveedores de EDI, fuertes en el mercado, tales como, Premenos Technology Corp. [PRMO], de Concord, Calif., y Dublin, y Sterling Software [SSW], de Ohio están tomando ventaja sobre cómo la Internet "cambiará el modo en que EDI se lleva a cabo". [PHILLIPS, 1999]

Sin embargo, existen varios problemas que tendrán que resolverse antes de que la Internet pueda ser utilizada universalmente como parte de la estrategia del Comercio Electrónico, especialmente cuando se refiere a la transmisión directa de documentos de negocio estructurados y formales entre aplicaciones computacionales, por ejemplo la seguridad y el cómo integrar los datos con las aplicaciones existentes.

4.3.1 SEGURIDAD

La principal controversia alrededor de la Internet para propósitos de negocios se enfoca en la falta de seguridad para la transmisión de documentos, pero existen numerosas respuestas razonables al respecto.

> Por ejemplo, "muchos vendedores han optado por incorporar paquetes de seguridad en el software EDI, tanto en el *'site'* receptor como en el emisor. Esta opción puede resultar cara e involucra paquetes de seguridad propietarios que podrían diferir entre cada concentrador." [HAISTING, 1996]

Es cierto a su vez que, independientemente de los problemas y las respuestas innovadoras sobre la seguridad, muy pocos documentos son lo suficientemente sensitivos para requerir una metodología de seguridad de transmisión de datos muy exigente. Por ejemplo, la industria del transporte usa un documento estándar llamado *"shipment status" (ANSIX12 214)* [ISA, 1998], para notificar al receptor que su mercancía va en camino. Este documento contiene información no sensitiva, y ha sido ampliamente utilizado por años, por los transportistas, como vehículo para notificar al receptor el estatus de la entrega de su mercancía. El documento *"shipment status"* también es utilizado internacionalmente por los usuarios EDIFACT, el equivalente internacional es la *"Arrival Notice"* o *IFTMAC*. La Internet constituye un muy buen método para transmitir este tipo de datos *"non-mission critical"*[HAISTING, 1996].

La decisión entre si usar una Intranet o la Internet es dictada por los

requerimientos de seguridad que requiere la transacción o documento en cuestión.

4.3.2 INTEGRAR LA INFORMACIÓN A LAS APLICACIONES EXISTENTES.

De cualquier forma, existen otros problemas de igual magnitud que el descrito que tendrán que ser resueltos antes de que el Internet se vuelva de uso común para los negocios.

> Estos problemas podrían no poder resolverse tan fácilmente como el de seguridad. El mayor de ellos podría ser el método utilizado para integrar la información recibida por Internet a las innumerables aplicaciones computacionales con las que operan los negocios de hoy en día. [HAISTING, 1996]

En este punto, es cuando las empresas, que ya tienen su infraestructura EDI altamente probada y estable y que además ya liga exitosamente los datos externos con su infraestructura de información a lo largo de la compañía, deben poner atención y evaluar realmente la conveniencia de cambiar hacia alguna nueva tecnología. Quizá la combinación de algunas les permita reducir sus costos, sin necesidad de cambiar radicalmente su infraestructura.

4.4 XML

Una tecnología que aparentemente nos propone algo interesante es el XML, iniciales provenientes del inglés *"eXtensible Markup Language"*. El XML es un lenguaje de "meta-marcado" *(meta-markup language)* el cual provee un formato para describir datos estructurados. Esto facilita que las declaraciones del contenido sean más precisas y que se reflejen en búsquedas más significativas a través de múltiples plataformas.

> Se le llama "*Extensible*" ya que el XML permite definir un conjunto de "etiquetas"*(tags)* ilimitado. A diferencia de las etiquetas de HTML que se utilizan para desplegar una palabra en negrillas o itálicas, las etiquetas de XML proveen un marco de trabajo para etiquetar datos estructurados. [MICROSOFT, 1999]

Los documentos XML están compuestos, al igual que el EDI, por entidades o elementos, que son unidades de almacenaje que contienen texto o datos binarios. Estos elementos son los que se encuentran marcados con las etiquetas mencionadas.

Un elemento XML puede declarar su dato asociado como un precio, un libro, un puerto, un nombre, una dirección, o cualquier otro tipo de elemento de dato que se desee. Veamos a continuación un pequeño ejemplo:

```
<Header>
    <Reference Number>310198</Reference Number>
    <Contact>
        <Name>Paola Cristina Solís</Name>
        <Geo Location>México</Geo Location>
    </Contact>
</Header>
```

Este fragmento lo hemos extraído de algunos de los campos o elementos de dato, que observamos en el "*Header*" del documento ejemplo que utilizamos en la Figura 4, de la sección 2.2. Aquí vemos como, en XML, existen etiquetas, denotadas entre los

símbolos <>, y </ > para marcar el principio y final de un elemento de dato, que puede incluir a su vez uno o varios elementos de datos anidados. Este fragmento nos indica que tenemos un encabezado, que contiene un número de referencia y un contacto. El contacto, a su vez, incluye el nombre del contacto y su localización geográfica. Los datos reales se encuentran entre las etiquetas, y en este caso, son 310198, Paola Cristina Solís y México.

Una vez que la información ha sido localizada, puede ser presentada al usuario por medio de un browser en diferentes formas, o bien puede ser entregada a otras aplicaciones para su proceso o manipulación posterior.

XML es un subconjunto del lenguaje conocido como *"Standard Generalized Markup Language"* (SGML) optimizado para su funcionamiento a través de la Web.

En 1986 la Organización Internacional para la Estandarización *(International Standard Organization, ISO)* publicó un estándar internacional al definir el SGML que mencionamos. Este lenguaje permitía al usuario:

- Identificar el papel y la sintaxis de cada pieza de información en un documento intercambiado.
- Identificar qué piezas de información debían estar presentes en cada conjunto de datos intercambiados y, si era relevante, el orden en el cual dichos elementos debían presentarse para un documento particular.

- Identificar qué programas debían ser utilizados para controlar cada uno de estos procesos.

Es por eso que el SGML ha conformado la base para múltiples proyectos de documentación multinacionales desde que fue publicado, tal es el caso del propio HTML.

En Julio de 1996 el *Consorcio del la World Wide Web (W3C)* formó un equipo de trabajo para estudiar cómo se podía simplificar el SGML para utilizarlo eficientemente a través de Internet. El resultado fue el desarrollo del XML que combina el poder expresivo del SGML con la funcionalidad que el HTML provee a la Internet. [BRYAN, 1998]

4.4.1 ALGUNAS REGLAS SENCILLAS PARA LA ESCRITURA DE XML

Una de las causas por las que el XML se está considerado como opción a nivel mundial, es por su sencillez de escritura y su fácil aprendizaje. Se dice que un documento XML está "bien formado" *(well-formed)* si sigue un conjunto de reglas básicas:

1. Los datos se anidan necesariamente por medio de pares de etiquetas.

   Los elementos XML pueden contener tanto texto como otros elementos, siempre y cuando sigan las reglas exactas que se dictan en el esquema dado para ese documento. Sin embargo, dichos elementos necesariamente deberán estar anidados: cada etiqueta de inicio deberá contar con su correspondiente etiqueta final.

La sintaxis para la etiqueta inicial deberá incluir el nombre del elemento entre los símbolos <>, y la etiqueta final será el mismo nombre de la inicial entre los símbolos: </ >. Por ejemplo:

Etiqueta inicial: <Inicio>

Etiqueta final: </Inicio>

2. Los elementos de datos no se pueden *sobreponer.*

Los elementos de datos deberán seguir la sintaxis apropiada de XML, por ejemplo, el siguiente párrafo no se apega a la sintaxis de XML:

<titulo>Evolution of Culture <sub>in Animals

</titulo> by John T. Bonner</sub>

El ejemplo estaría correcto de la siguiente forma:

<titulo>Evolution of Culture

_{in Animals}

<autor>by John T. Bonner</autor>

</titulo>

3. Las etiquetas XML son *'case-sensitive'*

Es decir que se tomarán como etiquetas diferentes las palabras que se diferencien en cuanto a mayúsculas y minúsculas, como por ejemplo:

<SuNombre>

<sunombre>

<Sunombre>

se consideran etiquetas distintas.

4. Es posible denotar los elementos vacíos.

XML cuenta con un 'atajo' para denotar los elementos vacíos. Únicamente es necesario colocar al final de la etiqueta inicial el símbolo / (diagonal) para señalar que el elemento no contiene datos. Es decir, en XML es igual escribir <título/> o bien <título></título>

5. Caracteres reservados.

Los siguientes símbolos están reservados por el lenguaje, y si es necesario incluirlos se hará de la siguiente manera:

Símbolo reservado	**Forma de incluirlo**
<	&It;
&	&
>	>
"	"
'	'

Por ejemplo, la oración *"Retiros > a $3000, pasar a la caja E&A"* se codificaría como *"Retiros &It; a $3000, pasar a la caja E&A"*

6. Cada documento XML deberá contar con un sólo elemento raíz.

   Cada documento deberá iniciar y finalizar con una etiqueta única, la cual contendrá todos los demás elementos de información. En el ejemplo del 'La Instrucción de Embarque", presentado en el punto "XML" de este documento, su elemento raíz es: *<header>*.

4.4.2 Beneficios de XML

A continuación presentaremos una serie de beneficios que se tienen al utilizar el XML, los cuales nos permitirán tener una idea más clara de porqué este lenguaje parece presentar una solución para integrar a todos los socios comerciales de una cadena de valor.

4.4.2.1. Búsquedas más significativas

Los datos pueden ser identificados con una etiqueta única, permitiendo, por ejemplo, a un cliente identificar los libros *por su autor*, en lugar de los libros que *'hablen*

del autor' como sucede en las aplicaciones de búsqueda actuales, las cuáles probablemente mezclarán ambos tipos de libros.

Sin XML, es necesario que la aplicación de búsqueda entienda el esquema de cada base de datos, la cual describe cómo está construida. Esto es virtualmente imposible ya que cada base de datos describe sus datos de manera diferente. Sin embargo, con XML, los libros podrían ser fácilmente categorizados en una forma estándar por autor, título, Número ISBN u algún otro criterio.

4.4.2.2. Desarrollo de Aplicaciones flexibles para la Web

Una vez que los datos han sido encontrados, XML los puede entregar a otras aplicaciones, objetos o servidores intermedios para su posterior procesamiento. También puede ser entregado al *desktop* para ser visto con un *browser*.

4.4.2.3. Integración de datos provenientes de distintas fuentes.

En la actualidad, la habilidad para buscar de múltiples e incompatibles bases de datos es prácticamente imposible. XML permite combinar datos estructurados de diferentes fuentes fácilmente. Se pueden utilizar los softwares *'agentes'* para integrar los datos extraídos de las diferentes bases de datos y otras aplicaciones en un servidor

intermedio. Estos datos pueden después ser entregados a los clientes u otros servidores para su posterior edición, proceso o distribución.

EL poder y la belleza de XML es precisamente que mantiene la separación de la interfase del usuario de la estructura de los datos. El *'Hypertext Markup Language'* (HTML) especifica cómo desplegar los datos en un browser, el XML define el contenido. Por ejemplo, en HTML se utilizan las etiquetas para decirle al browser que despliegue un dato en negrillas o itálicas; en XML se utilizan las etiquetas pero para describir los datos, es decir, para señalar que ese dato es una ciudad, un puerto, un nombre, una dirección, etc. En XML se utilizan 'hojas de estilo' *(stylesheets)* tales como las del *"Extensible Style Language"* (XSL) y *"Cascading Style Sheets"* (CSS) para presentar los datos en un browser. XML separa los datos de su presentación y proceso, permitiendo a los usuarios presentar y procesar la información según convenga, aplicando diferentes hojas de estilo y aplicaciones.

Esta separación de los datos de la presentación de los mismos, permite la integración total de los datos provenientes de diversas fuentes.

4.4.2.4. Datos para múltiples aplicaciones.

La extensión y flexibilidad del XML le permite describir datos contenidos en una gran variedad de aplicaciones heterogéneas, desde describir colecciones de páginas Web hasta registros. Ya que los datos basados en XML son auto-descriptivos, éstos pueden ser

intercambiados y procesados sin necesidad de tener una descripción interconstruída de los datos de entrada.

4.4.2.5. Escalabilidad

Una vez que los datos se encuentran en el cliente, pueden ser manipulados, editados y presentados en múltiples vistas, sin los regresos continuos al servidor. Los servidores se vuelven más escalables debido a la reducción de las cargas computacionales y ancho de bando. Además que como los datos son intercambiados en formato XML, pueden mezclarse fácilmente aunque provengan de fuentes distintas.

4.4.2.6. Cálculos y manipulación local

Como se mencionó, después de que los datos, en formato XML, son entregados al cliente, éstos pueden ser editados y manipulados localmente con las aplicaciones del mismo cliente. Los usuarios pueden manipular la información en forma distinta, en lugar de únicamente presentarla, ya que los datos se encuentran separados de la propia forma de presentarlos.

4.4.2.7. Actualizaciones granulares

Los datos pueden actualizarse granularmente con XML, eliminando la necesidad de reenviar un conjunto entero de datos estructurados cada vez que una porción de los mismos cambie. Sólo los elementos cambiados deberán ser enviados del servidor al

cliente, y podrán ser desplegados sin la necesidad de *'refrescar'* la interfase completa del usuario.

4.4.2.8. Entrega de la información a la Red

Ya que el XML es un formato abierto basado en texto, puede ser entregado utilizando HTTP en la misma forma que puede el HTML hoy en día, sin cambiar nada en las redes existentes.

4.4.3 XML ES AUTODESCRIPTIVO.

Una "Definición de Tipo de Documento" *(Document Type Definition. DTDs)* es una estructura que acompaña a un documento, en donde esencialmente se definen las reglas del mismo, tales como cuáles elementos estarán presentes y la relación estructurada entre ellos. Los DTDs ayudan a validar la información cuando la aplicación receptora no tiene interconstruída la descripción de los datos de entrada. Sin embargo con XML los DTDs son opcionales.

> La información enviada junto con su DTD es conocida como **XML válida.** En este caso un traductor XML puede checar la información de entrada contra las reglas definidas en el DTD para asegurar que la estructura de los datos es la correcta.
>
> La información que es enviada sin un DTD es conocida como "**bien-formada**". En este caso, un documento basado en XML que se encuentra estructurado jerárquicamente, puede ser utilizado implícitamente para autodescribirse. [MICROSOFT, 1999]

Las etiquetas describen el esquema de almacenamiento y la estructura lógica del documento. Un documento XML "bien-formado", es exacto, es decir, no permite la ambigüedad, por lo que un browser o editor puede leer las etiquetas que éste contiene y crear con ellas el árbol de su estructura jerárquica sin tener que leer la "Definición del Tipo de Documento" asociada.

En ambos casos, **la información codificada en XML es autodescriptiva**. El XML por tanto se puede utilizar para describir información sobre páginas HTML, o se puede utilizar para describir información contenida en *reglas de negocio* [AGULLÓ, 1997] u objetos en una transacción de comercio electrónico, tales como facturas, órdenes de compra, instrucciones de embarque, etc. Como el XML está separado del HTML, se puede agregar XML dentro de los documentos HTML.

La W3C ha definido un formato por medio del cual la información basada en XML, conocida como islas de información XML *(XML data islands),* se puede encapsular en páginas HTML. Al hacer esto, se pueden generar vistas múltiples de la información que se está presentando, utilizando la semántica incluida en el código XML. Más aún, el XML se puede utilizar para aplicaciones tales como la impresión distribuida, búsquedas en las bases de datos y otras.

El XML fue diseñado principalmente para el intercambio de información en forma de documentos desplegables por computadora. No toda la información comercial es intercambiada en un formato desplegable. Típicamente la información diseñada para

el intercambio electrónico de datos necesita ser procesada antes de que se pueda desplegar. Para que esto sea posible, la información debe de ser mapeada, utilizando alguna clase de plantilla en conjunto con ciertas reglas de procesamiento (exactamente lo que EDI hace).

5. XML/EDI

Recientemente se están llevando a cabo estudios y propuestas para, a través de XML, hacer el EDI accesible a un número mayor de usuarios. El método principal de capturar y codificar la información EDI sería a través de formas electrónicas codificadas en XML. De esta forma a partir de las pantallas XML se formarían los mensajes EDI y viceversa.

Con la llegada de Internet los patrones del comercio electrónico han cambiado dramáticamente. En particular, el Internet ha introducido nuevas formas de realizar comercio, permitiendo la interacción entre grupos que anteriormente no podían enfrentar el gasto económico necesario para hacerlo. [BENÍTEZ, 1997]

El XML/EDI NO ES LA CREACIÓN DE UN NUEVO ESTÁNDAR, sino la definición de cómo las compañías pueden utilizar los estándares actuales y ATACAR LOS PROBLEMAS DE UNA FORMA DIFERENTE.

El XML provee una metodología ideal para el comercio electrónico ya que:

- Permite a los creadores de los mensajes identificar claramente el papel y sintaxis de cada pieza de información intercambiada al utilizar una definición que puede interpretar tanto el humano como la máquina.
- También es posible identificar la fuente de cada estructura compartida utilizando un "Localizador de Recursos Uniforme" *(Uniform Resource Locator, URL).*

- Se conserva la propiedad de poder identificar qué elementos de datos deben estar presente en cierta estructura, y de ser necesario, el orden en que éstos deben aparecer.
- Los documentos XML pueden contener campos *"metadata"* que pueden ser utilizados para identificar quién es responsable por la creación, transmisión, recibo y proceso de cada mensaje, y puede tener facilidades interconstruídas para identificar los puntos de almacenaje de los programas que serán utilizados para controlar el proceso [HALFHILL, 1999].

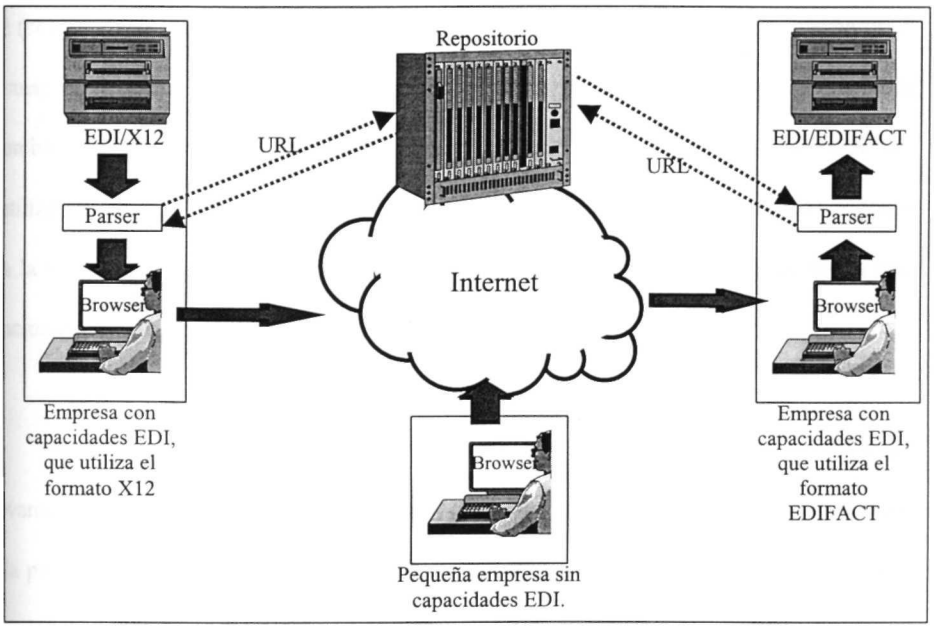


FIGURA 12. XML/EDI

Como vemos en la Figura 12, el Grupo XML/EDI [BRYAN, 1998] nos propone, que a partir de la definición de los documentos estándar EDI, los elementos de datos EDI serán "encapsulados" con elementos XML, que serán creados a partir de reglas específicas

almacenadas en uno o varios *repositorios comunes públicos* [IBM,1999], [PEAT, 1998], [RAMAN,1999]. Existen en la actualidad varias iniciativas sobre estos repositorios comunes de empresas importantes tales como Microsoft, IBM, SUN y Oracle. [RAMAN2,1999]. Una vez encapsulados, serán transportados por la Internet hasta llegar al destino, en donde sufrirán el proceso contrario.

Por ejemplo, una empresa (cliente) con capacidades EDI que utiliza el formato X12 (cómo el que vimos en la sección 2.2) puede enviar una "Instrucción de Embarque" al transportista marítimo, utilizando una pantalla codificada en XML, la cual se despliega en un browser común (con capacidades XML). Para encapsular los datos EDI en XML, se utiliza un "*parser*", el cual mediante los URLs accesa las diferentes reglas de negocio para la traducción correcta del documento, contenidas en los repositorios públicos que mencionamos.

Al llegar a la empresa destino (naviera con capacidades EDI), el browser nuevamente se conectaría a dicho repositorio para realizar el proceso contrario. Además, en la pantalla de la naviera iría codificada la dirección URL que tiene que se tiene que consultar a fin de transformar los datos encapsulados al formato EDI requerido por la naviera (EDIFACT).

Una empresa que no cuente con capacidades EDI, simplemente accesaría la forma XML, llenándola directamente en el browser, lo cual resultaría transparente para las

grandes empresas, ya que, no importando si el origen es EDI X12, EDIFACT o cualquier otro, los datos siempre llegan encapsulados en XML.

De esta forma ELIMINARÍAMOS GRAN PARTE DEL COSTO INVOLUCRADO EN LA IMPLEMENTACIÓN Y MANTENIMIENTO DE EDI, por tanto un número mayor de socios comerciales podría incorporarse al Comercio Electrónico, permitiendo a las empresas que implementaron EDI conservar y aprovechar mejor su infraestructura EDI.

> Jack Shaw, presidente de la empresa de consultoría, *"Electronic Commerce Strategies"*, predice que el uso de tecnologías basadas en Internet pueden ser la clave para las empresas chicas y medianas que, en el pasado, no pudieron enfrentar los costos de la implementación y mantenimiento de un sistema EDI.[PHILIPS,1999]

Conforme la totalidad de los socios comerciales se uniera a la comunidad "XML/EDI" SE IRÍA ELIMINANDO REALMENTE EL ESFUERZO MANUAL, Y POR TANTO SE MEJORARÍAN LOS PROCESOS.

Con el fin de ejemplificar nuestra hipótesis, acerca de que con la tecnología XML se pueden integrar los socios comerciales que manejen EDI con los que no cuenten con esta tecnología, se crearon un par de traductores de EDI X12 a XML y de XML a EDIFACT, mismos que se explican brevemente en el capítulo 7 y a detalle en los Anexos 1 y 2, respectivamente.

Así mismo, en dicho capítulo se presentará el formato del documento utilizado en XML, en donde se podrá apreciar la estructura y reglas mencionadas.

6. METODOLOGÍA DE INVESTIGACIÓN

Para llevar a cabo el presente trabajo se hizo una recopilación y análisis bibliográficos basados en distintos libros de sistemas de información y logística. Por tratarse de un tema concerniente a tecnología de punta, la mayor parte de la información fue extraída de artículos de reciente publicación, o bien, de fuentes obtenidas a través de numerosas búsquedas en Internet (CONSULTAR BIBLIOGRAFÍA).

Para llevar a cabo la encuesta, el Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Ciudad de México, proporcionó una base de datos de direcciones de correo electrónicas de 1200 empresas . Además se obtuvo una base de datos de direcciones electrónicas, del proveedor de EDI de Transportación Marítima Mexicana, de otras 25 empresas. Cabe señalar que la encuesta se mandó hasta en 2 ocasiones, solicitando la respuesta a ella y únicamente se recibieron 52 respuestas, mismas que fueron analizadas y cuyos resultados se presentan en el capítulo siguiente.

Por último, se requirió la colaboración de dos equipos de trabajo, (de la Materia de "Compiladores II" impartida, en el último semestre de la carrera de "Ingeniería en Sistemas Computacionales", por el Dr. Enrique Espinosa, en el Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Ciudad de México), para el desarrollo de dos traductores, uno de EDI X12 a XML y el otro de XML a EDIFACT, con el fin de comprobar el proceso propuesto. Se hará referencia a dichos traductores en el próximo capítulo y en los Anexos 1 y 2 del presente trabajo.

Los problemas principales con los que nos enfrentamos para la elaboración de este documento fue la falta de información cuantitativa sobre el tema y el número reducido de empresas, que en México, conocen y manejan EDI.

7. RESULTADOS DE LA INVESTIGACIÓN

7.1 Resultados de la Encuesta.

Con el fin de enriquecer el presente trabajo, se aplicó una encuesta a empresas mexicanas para conocer, de manera práctica, la realidad de la tecnología EDI en nuestro país y el posible mercado para la solución propuesta.

A continuación se presenta la encuesta aplicada:

Nombre de la Empresa	
Dirección:	
Puesto de quien contesta la encuesta:	
Teléfono:	
Fax:	
E-mail:	

1. Cuántos empleados laboran en la empresa? _____
2. Su negocio intercambia documentos con otras empresas en forma electrónica?

   SI_____ NO_____
3. Cómo es el intercambio de información de su empresa hacia los demás miembros de la cadena de valor (proveedores-empresa-clientes)?
   Preponderantemente Electrónica___Correo Electrónico__Otros (cuál?) _____
   Vía Fax..... _____
   Telefónicamente o por correo tradicional..... _____

4. Manejan Documentos EDI con proveedores?

SI_____ NO_____

5. Manejan Documentos EDI con clientes?

SI_____ NO_____

Si su respuesta a las dos preguntas anteriores fue negativa, pase a la pregunta 14, por favor.

6. Desde cuándo se implementó el EDI en la empresa?

Menos de un año _____
De 1 a 2 años _____
De 3 a 5 años _____
Más de 5 años _____

7. Cuál(es), de los siguientes, fueron los principales problemas encontrados en la implementación del EDI (numere del 1 al 9, en donde el 1 representa el problema mayor):

___ Se contrató staff adicional para realizar los mapas entre los socios comerciales
___ Se contrató una VAN (Value Added Network) para el servicio del EDI
___ Dificultad para conseguir recursos humanos que conozcan el EDI
___ Poco o nulo soporte por parte del proveedor del EDI

___ Los diferentes estándares que manejan los socios comerciales
___ El integrar la información que se transportó por EDI con las aplicaciones internas de la compañía
___ El tiempo invertido para implementar un nuevo documento
___ El tiempo invertido en implementar un nuevo socio comercial
___ La mayoría de mis socios comerciales no manejan EDI

8. Cuál(es), de los siguientes, son los principales problemas que la empresa enfrenta para el mantenimiento del EDI (numere del 1 al 4, en donde el 1 representa el problema mayor):

___ El costo staff adicional que se contrató para mantener los mapas entre los socios comerciales
___ El costo de la VAN o del proveedor que provee el servicio de EDI
___ Los diferentes estándares que se manejan con cada socio comercial
___ El poco o nulo soporte del proveedor del EDI

9. Considera usted que la implementación del EDI ha reducido los costos de su empresa?

SI_____ NO_____

10. Considera que el transportar los datos por medio de EDI es seguro?

SI_____ NO_____

11. Considera que el software de EDI es sencillo de manejar, es decir, que la curva de aprendizaje de éste es pequeña?

SI_____ NO_____

12. Considera que es sencillo realizar alguna modificación a los mapas o a sus aplicaciones internas que tengan que ver con EDI?

SI_____ NO_____

13. Considera usted que el EDI es una tecnología que cumpla con los requerimientos de hoy en día de su empresa?
SI_____ NO_____

Porqué?

_____

_____

_____

14. Ha considerado cambiar su infraestructura EDI por alguna tecnología basada en Internet?

SI_____ NO_____

Porqué?

_____

_____

_____

15. Cuál sería el tema que más le preocuparía al cambiar a una tecnología basada en Internet?
___ La seguridad de los datos

___ La confirmación de recepción de los datos
___ La integración de los datos transmitidos a las aplicaciones internas de la empresa
___ El tiempo transcurrido entre la recepción y recibo de los datos.
___ El tener la seguridad de que quién está enviando los datos es efectivamente la persona autorizada.
___ La resistencia al cambio por parte de quienes manejan la tecnología actual de su empresa.
___ La curva de aprendizaje de la nueva tecnología.

16. Aproximadamente, cuánto estaría dispuesto a gastar para cambiar a una tecnología basada en Internet?

____ Menos de $ 5,000 USD
____ Menos de $ 10,000 USD
____ Menos de $ 50,000 USD
____ Menos de $100,000 USD
____ Menos de $200,000 USD

17. Qué justificación tendría que dar para que se aprobara un proyecto de esta naturaleza en su empresa?

_____
_____
_____

FIGURA 13 . ENCUESTA APLICADA.

Después de analizar las respuestas recibidas, se obtuvieron los siguientes resultados:

Se entrevistaron empresas grandes (con más de 500 empleados), medianas (más de 100 empleados pero menos de 500) y chicas (con menos de 100 empleados).

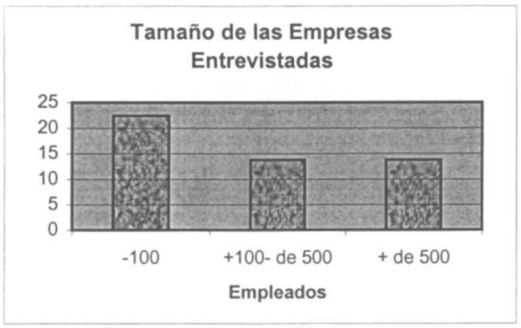


FIGURA 14. TAMAÑO DE LAS EMPRESAS ENTREVISTADAS

De las 52 encuestas recibidas, el 86% declararon intercambiar, por algún medio, información electrónicamente.

En la Figura 15 presentamos una gráfica de la cual podemos concluir lo siguiente:

- La mayoría de las empresas que manejan EDI en México, son empresas grandes.
- Algunas de las empresas medianas ya incursionaron en este campo, pero continúan abarcando, en su mayoría, el del correo electrónico junto con las empresas chicas.
- La mayoría de las empresas mexicanas, por lo menos, ya cuenta con correo electrónico.

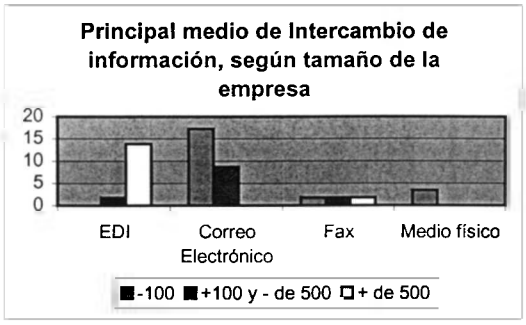


FIGURA 15. PRINCIPAL MEDIO DE INTERCAMBIO DE INFORMACIÓN, SEGÚN TAMAÑO DE LA EMPRESA

Ahora bien, analicemos a los que efectivamente manejan EDI. Pudimos concluir que la mayoría de los que manejan EDI lo hacen tanto con clientes como con proveedores. Sólo unos cuantos lo hacen únicamente con alguno de estos dos grupos, por lo que la tendencia es efectivamente "unir" la cadena de valor, y que la información utilizada en un documento, se reutilize en la creación de los documentos subsecuentes.

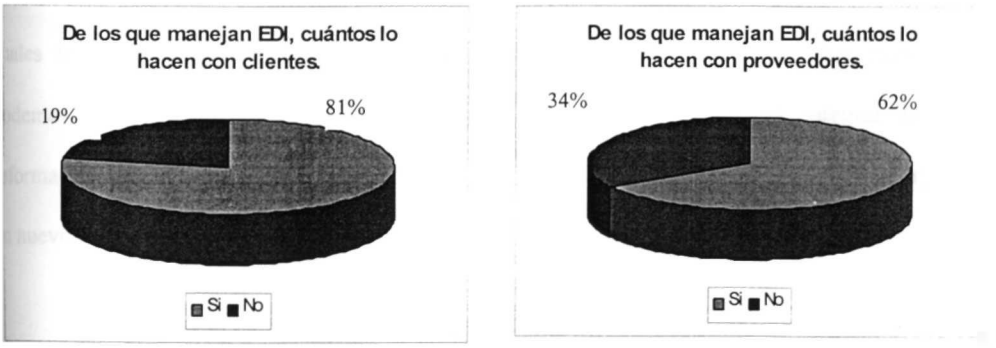


FIGURA 16. LA MAYORÍA MANEJA EDI, TANTO CON CLIENTES COMO CON PROVEEDORES

Los principales problemas que se encontraron las empresas mexicanas entrevistadas en la implementación del EDI, son:

Problema	Porcentaje
Recursos Humanos capacitados en EDI	28%
El integrar la información que llegaba en EDI a sus aplicaciones existentes.	28%
El tiempo que se le necesita invertir al crear o integrar un nuevo documento.	14%
La contratación de staff para realizar los mapeos.	10%
El poco o nulo soporte de parte del proveedor EDI.	10%
La poca penetración que existe en México de la tecnología EDI.	5%
La contratación de una VAN.	5%

En la Figura 17 podemos apreciar gráficamente los principales problemas a los cuales las empresas mexicanas se han enfrentado al implementar EDI. Así pues, podemos observar que el encontrar Recurso Humano capacitado, el integrar la información en las aplicaciones existentes así como el tiempo invertido para implementar un nuevo documento son los problemas que ocupan la mayor parte de la gráfica.

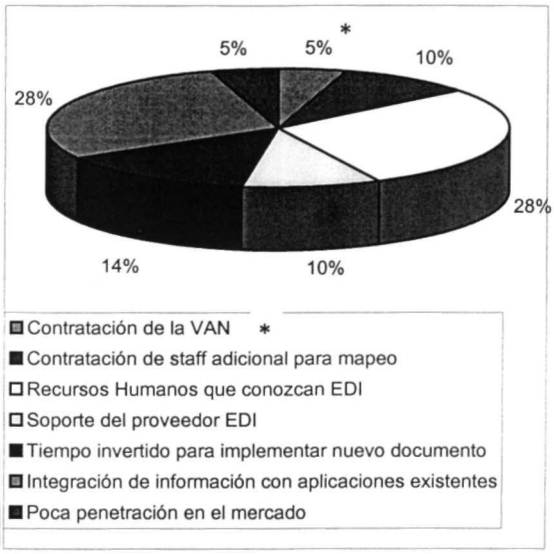


FIGURA 17. PRINCIPALES PROBLEMAS AL IMPLEMENTAR EDI.

NOTA: EL * representa la rebanada de donde partimos en el sentido de las manecillas del reloj

Así mismo, analizamos los problemas que encontraron dichas empresas en el mantenimiento del EDI, encontrando lo siguiente:

- Los diferentes estándares que se manejan con cada socio comercial
- El costo del staff adicional para mantener actualizado el mapeo
- El costo de la VAN o del proveedor EDI.
- El poco soporte del proveedor EDI.

En la Figura 18 podemos observar que, prácticamente en la misma medida, se han presentado estos problemas.

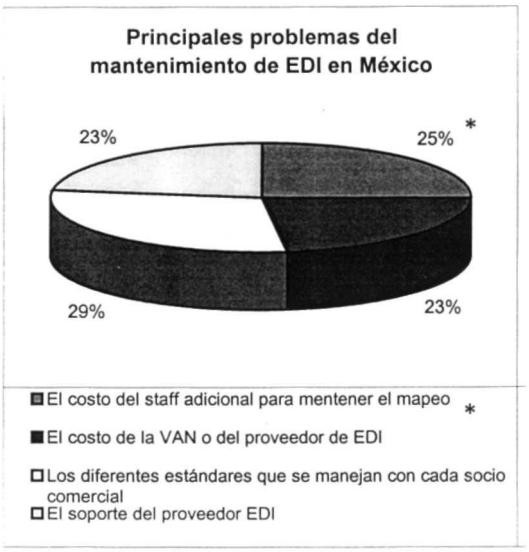


FIGURA 18. PROBLEMAS DE IMPLEMENTACIÓN

NOTA: EL * representa la rebanada de donde partimos en el sentido de las manecillas del reloj

De los problemas encontrados podemos concluir que en México hace falta capacitación en dicha tecnología, recursos humanos que dominen el manejo de la misma. Si hubiera un mayor número de recursos humanos capacitados, el precio del mismo descendería y el tiempo invertido en la creación o mantenimiento de un mapa tendería a ser menor, por la misma capacitación del recurso.

Además es evidente la necesidad de una herramienta que responda al dinamismo del negocio, y que, por tanto, resulte sencillo modificar las reglas de negocio que la rigen, ya que de esta forma, aunque existieran numerosos estándares de los socios comerciales, al ser transportados todos a través del XML, sería transparente para el receptor el formato original del mensaje, y con ello sólo mantendría sus mapas de XML a su formato interno.

Con los resultados de las encuestas, corroboramos las ideas presentadas en cuanto a que:

- El 90% de los encuestados, manifestó que el EDI disminuyó los costos de su empresas comparado con el costo de manejo manual de los documentos .
- El 100% aseguró que EDI es un método seguro de intercambio de documentos.
- EL 70% declaró que no considera sencillo realizar alguna modificación dentro de su mapas, sin embargo el 100% manifestó que el EDI es una tecnología que cumple con los requerimientos de su empresa y sólo el 40% estaría interesado en cambiar a una infraestructura basada en Internet.

Este último punto, nos da la pauta para asegurar que **no** sería conveniente **proponer** una tecnología **sustituta** del **EDI**, sino más bien complementaria, ya que las empresas que ya cuentan con la tecnología EDI, no están dispuestas a eliminar el esfuerzo invertido. He ahí en donde radica la fuerza de la propuesta EDI/XML: no es necesario deshacerse de la infraestructura de las grandes empresas, pero es posible integrar a las pequeñas y medianas a dicho intercambio electrónico de documentos, permitiendo así, mejorar los procesos internos de las empresas al eliminar realmente el esfuerzo manual.

Ahora bien, las preocupaciones que se mostraron para incursionar en una tecnología basada en Internet fueron las siguientes:

- Seguridad de la información
- El tiempo transcurrido entre el envío y la recepción de la información.
- El obtener una confirmación de la recepción de los datos.
- La integración de los datos transmitidos a las aplicaciones existentes.
- La resistencia al cambio por parte de los recursos que manejarían la nueva tecnología.
- La autenticación. Asegurar que quien está mandando los datos, es realmente quien dice ser.
- La curva de aprendizaje de la nueva tecnología.

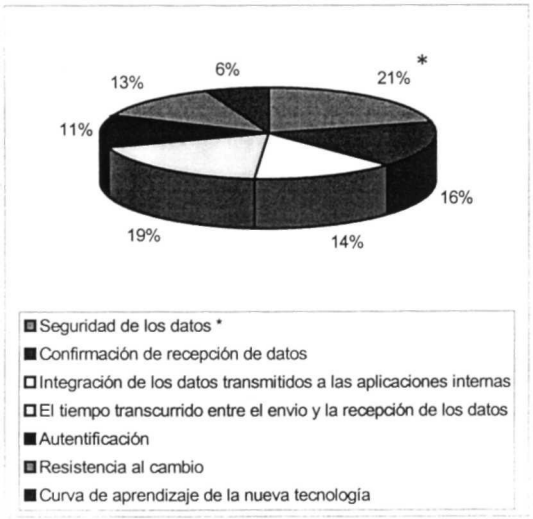


FIGURA 19. PREOCUPACIONES DE LAS TECNOLOGÍAS BASADAS EN INTERNET

NOTA: EL * representa la rebanada de donde partimos en el sentido de las manecillas del reloj

Por último, es importante considerar que las empresas mexicanas, en general, no están dispuestas a invertir grandes cantidades de dinero en la solución propuesta. En la Figura 20 vemos reflejado este aspecto.

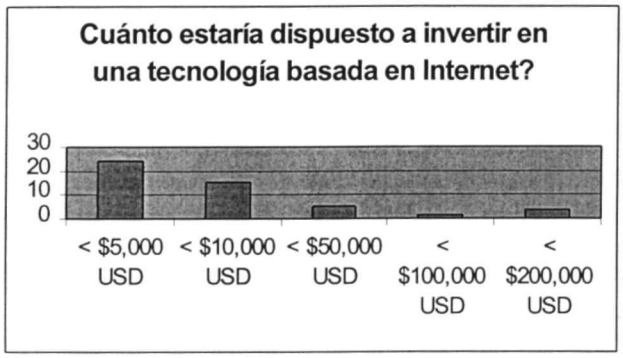


FIGURA 20. LAS EMPRESAS NO ESTÁN DISPUESTAS A INVERTIR MUCHO EN INTERNET

Como observamos la mayoría de las empresas manifestaron que invertirían menos de cinco mil dólares en una solución basada en Internet.

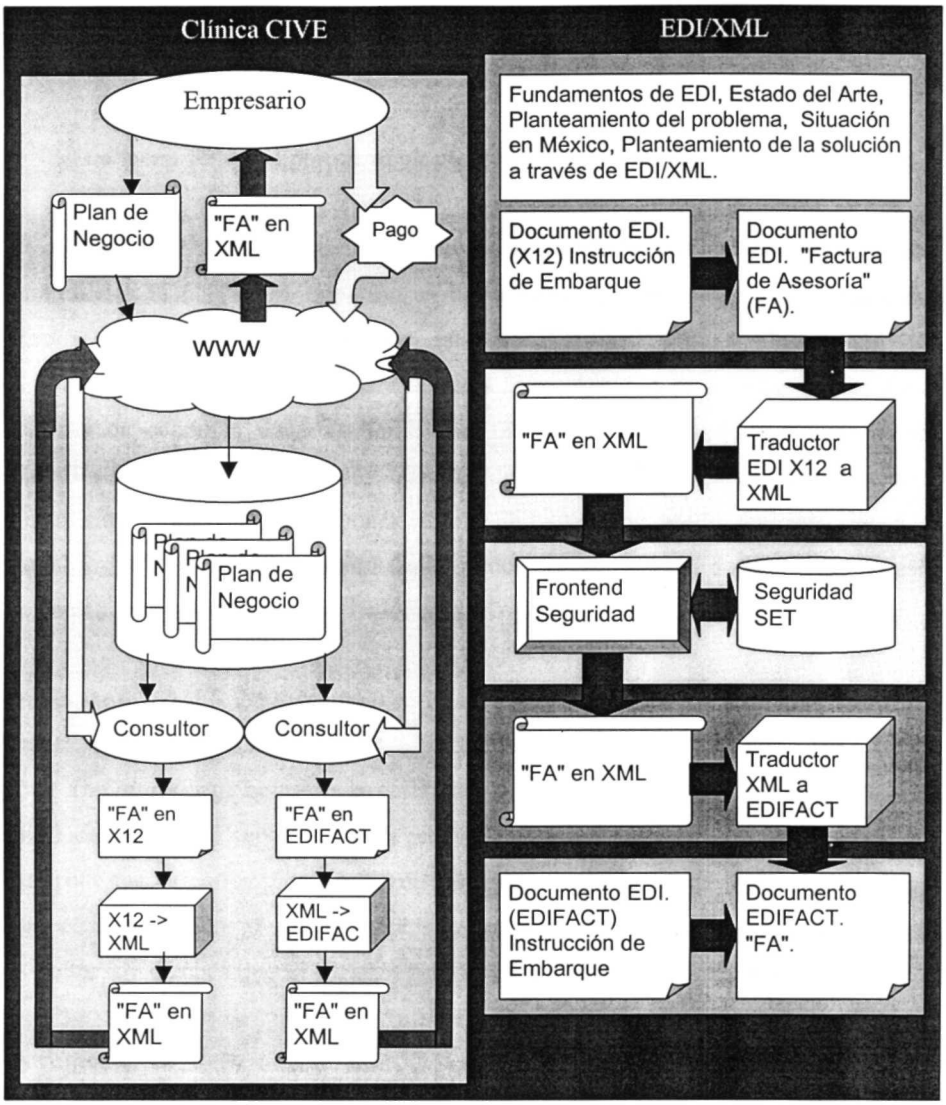
7.2 Explicación del proyecto de programación.

En un esfuerzo para ejemplificar cómo sería posible intercambiar documentos, por ejemplo, entre una entidad que contara con una infraestructura EDI X12 y otra que contara a su vez con una infraestructura EDI pero EDIFACT a través de XML, se colaboró en el desarrollo de dos traductores [CASTRO, DE ANDA, 1999], [ATHIÉ, ESCAMILLA, 1999].

En la Figura 21, mostramos el contexto de todo el proyecto, en donde primeramente se presenta el estudio bibliográfico, base para dicho proyecto. En seguida vemos representado el documento EDI del que hemos hablado en capítulos anteriores (Sección 2.1). A partir de dicho documento se obtuvo un subconjuto del mismo, con el único objeto de hacer manejable nuestro prototipo, que diera vida al documento llamado "Factura de Asesoría".

Partiremos de la idea de que la "Factura de Asesoría" fuera un documento estándar y reconocido por los traductores de EDI. Este documento se enviaría cada vez que un alumno de la "Clínica Incubadora Virtual de Empresas"[ITESM, 1999] solicitara algún tipo de asesoría a un consultor. En el momento que el consultor proporcionara el servicio, cobraría por el mismo a través de la "Factura de Asesoría", y sin importar si esta factura fue emitida originalmente con EDI X12, EDIFACT o simplemente con XML, viajaría a través de Internet en forma de documento XML y se recibiría en el extremo correspondiente a fin de solicitar el pago del servicio.

El primer traductor, que se explica más ampliamente en el Anexo 1, es un traductor de X12 a XML. La función de este traductor es convertir un documento que se recibe del compilador de EDI.X12 - asumiendo que sus partes léxica, sintáctica y semántica son correctas- a un documento XML.



NOTA: "FA" siglas de "Factura de Asesoría"

FIGURA 18. EXPLICACIÓN DEL PROYECTO DE PROGRAMACIÓN

El traductor funciona por medio de cuatro elementos:

1) **Documento EDI.X12.** Este elemento es el resultado que se obtendría de un compilador EDI.X12. En este caso, el documento a procesar será una "Factura de Asesoría", extraída del documento estándar EDI X12 302, o "Instrucción de Embarque", por así convenirlo para nuestros fines. Dicho documento se enfoca a un sistema de cobranza de consultorías que ofrecerá el Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Ciudad de México. En la vida real, dicho documento se tendría que proponer ante la *ISA ("Information System Agreement")*, de tal forma que fuera reconocido como un documento estándar de EDI X12, el cual se podría procesar con un compilador para EDI.X12.

2) **Documento XML.** Este elemento se forma a partir de un par de archivos generados por el "Motor EDI2XML"[VER ANEXO 1]. Un archivo es el llamado *DTD (Document Type Definition)* que contiene la definición de las reglas gramaticales (definiciones del documento) que serán utilizadas por *el Document Element* que es el otro archivo que contiene todos los datos de un documento XML, éste puede estar comprendido por cualquier número de elementos y subelementos que corresponden a las estructuras y campos que EDI utiliza para la definición de sus documentos.

3) **Documento de definiciones EDI-XML: Base de Conocimientos.** Este es un archivo de texto plano que se escribe manualmente por el administrador del sistema; contiene el tipo del segmento EDI que corresponde al mismo en XML, las etiquetas de los segmentos del estándar EDI.X12 y sus equivalentes en XML, las etiquetas de los campos que forman dichos segmentos y sus equivalentes en XML, el tipo de dato de los campos y el número de caracteres mínimo y máximo que puede ocupar cada campo. Esta 'base de conocimientos", vendría siendo el 'repositorio común'[IBM,1999]. [PEAT, 1998] al que nos hemos referido.

4) **Motor EDI2XML o *'parser'*.** Este elemento se encarga de accesar el archivo EDI.X12, analizarlo, traducir su contenido a XML y arrojar un documento completo y correcto de XML. Primeramente se genera el DTD tomando línea por línea de la Base de Conocimientos, en donde cada una de éstas representa una estructura. Para generar el *Document Element* se lee línea por línea del archivo EDI.X12, se identifica la etiqueta del segmento y se busca en la Base de Conocimientos. Una vez encontrada en la BC se identifica la sección del documento a la que dicho segmento pertenece. De la misma línea se van extrayendo los datos de los diferentes campos del segmento los cuales están separados por asteriscos y tienen una correspondencia en la BC, de donde se obtiene el nombre de la etiqueta en XML, el tipo de dato del campo (numérico o alfanumérico), su longitud mínima y máxima. Por último, estos datos se insertan en el *Document Element.*

De esta manera el traductor utiliza el 'Motor EDI2XML' para extraer información de un documento EDI.X12 utilizando la Base de Conocimientos para generar un documento XML.

Después de que se realizó dicho documento, éste pasará a través de un mecanismo de seguridad[FRIAS, 1999] para viajar a través de Internet, asegurando la integridad de los datos, su autenticación y el no repudio de los mismos, llegando a su destinatario, el cuál asumimos que tiene una infraestructura EDIFACT.

Para ello se construyó el segundo traductor, que se explica más ampliamente en el Anexo 2, y cuya principal función es traducir el documento XML a un documento

EDIFACT, completando así el ciclo descrito. Este traductor se compone de los siguientes elementos:

1) **Documento XML.** Este elemento es el resultado de la traducción del documento EDI X12 a XML, y constituye la entrada de la aplicación traductora a EDIFACT.

2) **Base de Datos de Definiciones**. Éste es un archivo que, siguiendo con las reglas de definición de un archivo de inicialización Microsoft (extensión INI), indica las etiquetas del documento XML y su equivalente en EDIFACT. Esta base de datos es la que toma la aplicación para obtener el conocimiento de ambas gramáticas, la fuente y el destino.

3) **Aplicación traductora.** Este elemento abre tanto la base de datos de definiciones, como el documento fuente en XML. Realiza un análisis de tres pasos:

a) Verifica que el documento efectivamente sea un documento XML con las siguientes características: texto plano, encabezado con etiquetas identificables de XML y cuerpo de mensaje.

b) Analiza los identificadores pertenecientes al "*document type definition*", *DTD,* en caso de encontrar errores o identificar partes del documento sin mención o relación en la base de datos, genera los mensajes pertenecientes al error identificado. Si, por el contrario, no encuentra obstáculos en el análisis, continúa al siguiente paso.

c) Identifica dónde inicia el documento y compara los identificadores con su correspondiente en EDIFACT generando así el documento EDIFACT.

4) **Documento de Definición de Tipo (DFD).** Este elemento es la definición de los campos del documento que contiene la información a ser generada en el documento en sí.

5) **Documento EDIFACT.** Este es el resultado final que contiene los campos con sus datos respectivos. Este documento, al igual que el generado para el traductor del EDI X12, se sustrajo del documento "Instrucción de Embarque" o IFTMIN de EDIFACT, por así convenirlo para el ejemplo, pero se tendría que proponer, igualmente, ante la autoridad correspondiente para que fuera aceptado mundialmente y los traductores de EDIFACT fueran capaces de interpretarlo. De cualquier forma conserva la estructura y forma de un documento típico de EDIFACT.

8. CONCLUSIÓN

Las grandes corporaciones no han podido realizar una reingenieria real a sus procesos, ya que las empresas pequeñas y medianas no han podido integrarse a la comunidad EDI, por tanto continúan arrastrando el esfuerzo manual en las operaciones.

El implementar y mantener EDI ha representado un proceso largo, costoso y que requiere, la mayoría de las veces, de la contratación de un staff dedicado a realizar y mantener actualizados los mapas entre la información que se recibe y las aplicaciones existentes, así como la contratación de un tercero, como una VAN, quien contribuya a dicho proceso.

Así mismo, y por la propia naturaleza competitiva de las empresas, el "estándar" EDI cada vez es menos estándar, ya que cada empresa ha realizado modificaciones a éste a fin de lograr una mejor comunicación con sus aplicaciones existentes.

Las pequeñas compañías han empezado a incursionar en soluciones menos costosas en tiempo y dinero para poder integrarse con sus socios comerciales grandes como son las distintas opciones que ofrecen los Centros de Servicio, pero aún así no han implementado en general alguna solución que esté acorde a las necesidades reales del mundo competitivo actual.

Los Gerentes de negocio deben enfocarse en LA HABILIDAD DE HACER NEGOCIOS ELECTRÓNICAMENTE CON TODOS SUS SOCIOS COMERCIALES y no en los estándares que se

siguen para hacer esto posible. Además, para finalmente eliminar el esfuerzo de procesamiento manual, NECESITAN SISTEMAS DE PROCESAMIENTO BASADOS EN REGLAS DE NEGOCIO que puedan ser mantenidas y operadas fácilmente por los expertos de negocio, quienes puedan hacer cambios inmediatos para reflejar los requerimientos cambiantes del negocio sin la necesidad del costoso staff EDI.

La alternativa que iría acorde a la realidad competitiva de hoy en día, sería el permitir a los socios comerciales mandar información electrónicamente en cualquier formato y bajo cualquier método de transmisión, siempre y cuando contenga la información requerida para cumplir con los requerimientos de procesamiento de nuestras aplicaciones internas. La compañías deberán contar con una tecnología suficientemente poderosa como para manejar la diversidad descrita.

En la actualidad se están llevando a cabo estudios para combinar varias de las tecnologías emergentes y hacer posible lo anteriormente descrito, tal es el caso de XML, tecnología que cada vez recibe un mayor apoyo por parte de las grandes corporaciones como IBM, Microsoft, etc.

Por las capacidades que XML ofrece, el futuro del EDI parece vislumbrar una solución, uniendo ambas tecnologías, para integrar a todos los socios comerciales y poder efectuar así una verdadera reingeniería de procesos.

Según los resultados de la encuesta aplicada, pudimos comprobar que la mejor alternativa de solución, sería proponer una tecnología complementaria al EDI, pero no sustituta, ya que las empresas que ya cuentan con su infraestructura EDI no están dispuestas a eliminar el esfuerzo invertido para la construcción y puesta en marcha de su infraestructura EDI.

Con el XML/EDI las empresas chicas y medianas, podrían compartir información electrónicamente con las empresas que contaran con su infraestructura EDI sin tener que hacer una inversión tan fuerte, ya que para éstas últimas sería totalmente transparente el conocer si el documento fue creado originalmente en un formato EDI, XML o algún otro.

Consideraciones importantes a tomar en cuenta, según los resultados de la encuesta, para proponer una solución, son: el contar con un mecanismo altamente seguro para la transferencia y recepción de los documentos electrónicos y la capacitación suficiente a los recursos humanos encargados del proyecto, de tal forma que dominen la tecnología propuesta.

9. RECOMENDACIONES Y TRABAJOS FUTUROS

Como continuación al trabajo aquí presentado, se recomiendan los siguientes temas a desarrollar:

- Realizar un estudio adicional del costo/beneficio de la tecnología EDI/XML con las siguientes variaciones:
  - Tomando en cuenta la seguridad.
  - Sin tomar en cuenta la seguridad.
  - Tomando en cuenta el desarrollo de XML como estándar.

- Llevar a cabo un estudio de factibilidad de los repositorios comunes propuestos por IBM y Microsoft. [IBM, 1999], [MICROSOFT, 1999]

- Integrar totalmente el mecanismo de seguridad al mecanismo de traducción de EDI X12 y EDIFACT a XML.

- Incorporar el mecanismo de conversión informática dentro del contexto del CIVE (Ver Figura 18)

- Implementar un programa gráfico (como las herramientas que existen actualmente para desarrollar páginas WEB) que permita realizar de una manera sencilla, las correspondencias entre los registros EDI y XML.

10. ANEXO 1. TRADUCTOR DE EDI X12 A XML

El traductor EDI2XML es un programa que traduce un archivo de EDI a un archivo XML. Para comprender de una manera más clara su funcionamiento primero se describirá las partes que lo conforman y después se explicará la interacción de las mismas.

10.1 Componentes del traductor EDI2XML

Los componentes del traductor EDI2XML son los siguientes:

Documento EDI.

Base de conocimientos.

Motor EDI2XML

Documento XML

1.- Documento EDI.X12. Este componente es un archivo de texto plano obtenido de un compilador EDI.X12. Este archivo contiene documentos especificados bajo los estándares de EDI. En el estándar EDI los documentos constan de tres secciones principales *Header, Detail y Summary.* Cada sección está conformada por uno o más segmentos *"Document Segments"* y cada segmento está definido por uno o más campos *"Fields"* y cada campo le corresponde un valor.

En un archivo EDI, cada línea corresponde a un segmento del documento. Al inicio de cada renglón se presenta el ID, identificador, del segmento y a continuación, separados por asteriscos, se encuentran los valores de los campos de dicho segmento, si el valor de algún campo no es incluido se indica con asterisco seguido de asterisco.

ID_DEL_SEGMENTO*Valor_del_primer_campo*Valor_del_segundo_campo*...**

En la Figura 19, se presenta un documento EDI. Dicho documento corresponde al mencionado en la sección 7.2, llamado "Factura de Asesoría", el cual se empleará como ejemplo para describir el funcionamiento del traductor. Este archivo lo encontrará con el nombre de "Edido.txt" dentro del directorio en donde se cargue el programa.

```
ST*501*001003
B2*C1*DF*CC**BP
G3*10*5000*ITESM/CCM
N9*991006*130533
M0*00235*990107*991025
M2*C3*001003*990102*990914
C2*1*BANCOMER*00987654*C*990107
C3*MEX*9.60*DLS
N1*ST*PIFF S CO.
N3*Playa Regatas No.491 Col. Marte*D.F. CP08830 MEXICO
N4*Distrito Federal*DF*08830*52
LX*1413
Y2*9*C1
L1*1000*9000*5000
C3*USA*9.60*DLS
L3*9000*5000
L11*001003*Consultoria Plan de Negocios via e-mail PIFF
S Co.
SE*1*0001
```

FIGURA 19. "FACTURA DE ASESORÍA". DOCUMENTO EDI, X12.

2.- Base de conocimientos. Archivo de texto plano que se escribe manualmente por el administrador del sistema. Contiene todos los segmentos que conformarán el documento que se vaya a crear, en este caso, los segmentos del documento *BILL*. Esta 'base de conocimientos", vendría siendo el 'repositorio común'[IBM,1999], [PEAT, 1998] al que nos hemos referido.

En este archivo se indica la sección a la cual pertenece cada segmento (H *Header*, D *Detail*, S *Summary*), la etiqueta EDI del segmento, su etiqueta equivalente en XML la cual definirá el *tag* correspondiente en dicho lenguaje y el tipo o carácter del segmento (M si es mandatorio, O si es opcional).

Debajo de cada segmento se encuentran definidos los campos de dicho segmento. La definición de los campos comienza con un guión "-" seguido de la etiqueta EDI del campo del segmento, de la etiqueta equivalente en XML, del tipo o carácter del campo (M si es mandatorio, O si es opcional), del tipo de información que describe ese campo (ID, AN alfanumérico, N numérico, etc.), y por último, del número mínimo y máximo de caracteres permitidos para el valor de ese campo. Todos los campos están separados por tabuladores.

En la figura 20, se presenta el Archivo Base de Conocimientos que contiene la definición del documento "Factura de Asesoría". Como se puede apreciar, éste contiene las etiquetas en EDI y sus equivalentes en XML. El nombre de dicho archivo es "BC.txt"

H	ST	BILL_HEADER M							
-	ST01	BILL_ID	M	ID	3	3			
-	ST02	BILL_NO	M	AN	4	9			
H	B2	BILL_INFORMATION	O						
-	B201	CONSULTING_TYPE_CODE		O	ID	2	2		
-	B203	CITY_SERVED O		ID	6	9			
-	B208	METHOD_OF_PAYMENT M		ID	2	2			
-	B209	STATUS_REPORT_REQUEST_CODE		O	ID	1	1		
-	B218	RECIVED_DOCUMENTATION_ID		O	ID	2	2		
H	G3	COMPENSATION_INFORMATION		O					
-	G301	TECS_PERCENTAGE_COMISSION		O	R	2	5		
-	G302	TOTAL_COMPENSATION_AMOUNT		M	N0	3	10		
-	G303	NAME_CAMPUS O		AN	1	35			
H	N9	REFERENCE_NUMBER1 O							
-	N904	DATE1 O		DT	6	6			
-	N905	TIME O		TM	4	6			
H	M0	LETTER_OF_CREDIT_REFERENCE		O					
-	M001	NUMBER	M	AN	2	24			
-	M002	DATE2 M		DT	6	6			
-	M003	EXPIRATION_DATE		O	DT	6	6		
-	M004	CHARGE_DATE O		DT	6	6			
H	M2	SALES-DELIVERY_TERMS		O					
-	M201	CONSULTING_CODE	M	ID	2	2			
-	M202	REFERENCE_NUMBER2 O		AN	4	6			
-	M203	SERVICE_REQUESTED_ON_DATE		O	DT	6	6		
-	M206	SERVICE_DELIVERED_ON_DATE		O	DT	6	6		
H	C2	BANK_ID	O						
-	C201	BANK_CLIENT_CODE	M	ID	1	1			
-	C203	BANKS_NAME	O	AN	2	17			
-	C205	BANK_ACOUNT_NUMBER		O	AN	6	17		
-	C206	PAYMENT_METHOD_CODE		O	ID	1	1		
-	C207	DATE3 O		DT	6	6			
H	C3	CURRENCY_INFORMATION		O					
-	C301	CURRENCY_COUNTRY_CODE		M	ID	3	3		
-	C302	EXCHANGE_RATE		O	R	4	6		
-	C303	CURRENCY_RATE		O	ID	3	3		
H	N1	NAME M							
-	N101	ENTITY_IDENTIFIER_CODE		M	ID	2	2		
-	N102	CLIENTS_NAME		C	AN	1	35		
H	N3	ADDRESS_INFORMATION		O					
-	N301	ADDRESS_BUILDING-STREET-PO_BOX				M	AN	1	35
-	N302	CITY-STATE-POSTAL_CODE-COUNTRY				O	AN	1	35
H	N4	CLIENTS_GEOGRAPHIC_LOCATION		O					
-	N401	CLIENTS_CITY_NAME C		AN	2	30			
-	N402	STATE_CODE O		ID	2	2			
-	N403	POSTAL_CODE O		ID	3	9			
-	N404	COUNTRY_CODE		O	ID	2	3		
D	LX	CONSULTING_SERVICE_NUMBER_ASSIGNED				M			
-	LX01	CONSULTATION_NUMBER		M	N0	1	6		
D	Y2	CONSULTATION_DETAILS		O					
-	Y201	NUMBER_OF_INTERCHANGES		M	N0	1	4		

-	Y202	CONSULTATION_CODE	O	ID	1	1				
D	L1	RATE_AND_CHARGES	O							
-	L102	CONSULTATION_RATE	O	R	1	9				
-	L104	CHARGE	C	N2	1	9				
-	L106	PRAPAID_AMOUNT	C	N2	1	9				
D	C31	CURRENCY	M							
-	C311	CURRENCY_COUNTRY_CODE2	M	ID	3	3				
-	C312	EXCHANGE_RATE2	O	R	4	6				
-	C313	CURRENCY_CODE	O	ID	3	3				
S	L3	TOTAL_CHARGES	M							
-	L305	TOTAL_CHARGE_FOR_SERVICES	O	N2	1	9				
-	L307	PREPAID_AMOUNT	O	N2	1	9				
S	L11	BUSINESS_INSTRUCTIONS	O							
-	L1101	REFERENCE_NUMBER3	C	AN	1	30				
-	L1102	CLIENTS_ID	C	ID	2	2				
-	L1103	DESCRIPTION	C	AN	1	80				
S	SE	TRANSACTION_SET_INFORMATION	M							
-	SE01	NUMBER_OF_SEGMENTS_INCLUDED_IN_BILL	M	N0	1	10				
-	SE02	TRANSACTION_SET_CONTROL_NUMBER	M	AN	4	9				
X	XX	X	M							
-	XXX	X	M	X	1	1				
-	XXXX	X	M	X	1	1				

FIGURA 20. BASE DE CONOCIMIENTO PARA LA "FACTURA DE ASESORÍA"

3.- Documento XML. Este componente es un archivo XML que está formado de dos partes principales. La primera parte es el DTD "*Document Type Definition*" y la segunda es el "*Document Element*". En la primera parte se definen las reglas léxicas, sintácticas y semánticas del documento a crear y todos los elementos que lo conformarán para más tarde poder crear el *"Document Element"*, en éste es donde se especifican los valores de cada componente del documento, es decir, es en esta parte en donde, a los campos de los segmentos, se les asignará un valor determinado.

Para crear la "Factura de Asesoría" y cumplir con la estructura de documentos dictada por EDI, en el DTD primero se define el elemento *BILL* que estará conformado por tres elementos *Header, Detail y Summary.* Cada uno de estos elementos estará

conformado por uno o más elementos que corresponden a los segmentos de cada sección, *"Document segments"* y a su vez cada elemento segmento contendrá uno o más elementos que son los campos *"Fields"* de cada segmento. Debido a que cada campo tiene diferentes propiedades, éstas se definen por medio de atributos dentro del DTD.

Estos atributos corresponden a:

* **ID**.- Identificador del campo que es la etiqueta EDI de dicho campo,

* **TYPE.**- Define qué tipo de campo es (opcional o mandatorio),

* **REQ_DES**.- Define qué el tipo de información especificada para dicho campo (ID, alfanumérica, numérica, etc.),

* **Min**.- Mínima longitud en caracteres del campo y

* **Max**.- Máxima longitud en caracteres del campo.

```
<BILL>
  <HEADER>
    <BILL_HEADER>
      <BILL_ID>501</BILL_ID>
      <BILL_NO>001003</BILL_NO>
    </BILL_HEADER>
    <BILL_INFORMATION>
      <CONSULTING_TYPE_CODE>C1</CONSULTING_TYPE_CODE>
      <CITY_SERVED>DF</CITY_SERVED>
      <METHOD_OF_PAYMENT>CC</METHOD_OF_PAYMENT>
      <STATUS_REPORT_REQUEST_CODE></STATUS_REPORT_REQUEST_CODE>
      <RECIVED_DOCUMENTATION_ID>BP</RECIVED_DOCUMENTATION_ID>
    </BILL_INFORMATION>
    <COMPENSATION_INFORMATION>
      <TECS_PERCENTAGE_COMISSION>10</TECS_PERCENTAGE_COMISSION>
      <TOTAL_COMPENSATION_AMOUNT>5000</TOTAL_COMPENSATION_AMOUNT>
      <NAME_CAMPUS>ITESM/CCM</NAME_CAMPUS>
    </COMPENSATION_INFORMATION>
    <REFERENCE_NUMBER1>
```

```
      <DATE1>991006</DATE1>
      <TIME>130533</TIME>
    </REFERENCE_NUMBER1>
    <LETTER_OF_CREDIT_REFERENCE>
      <NUMBER>00235</NUMBER>
      <DATE2>990107</DATE2>
      <EXPIRATION_DATE>991025</EXPIRATION_DATE>
    </LETTER_OF_CREDIT_REFERENCE>
    <SALES-DELIVERY_TERMS>
      <CONSULTING_CODE>C3</CONSULTING_CODE>
      <REFERENCE_NUMBER2>001003</REFERENCE_NUMBER2>
      <SERVICE_REQUESTED_ON_DATE>990102</SERVICE_REQUESTED_ON_DATE>
      <SERVICE_DELIVERED_ON_DATE>990914</SERVICE_DELIVERED_ON_DATE>
    </SALES-DELIVERY_TERMS>
    <BANK_ID>
      <BANK_CLIENT_CODE>1</BANK_CLIENT_CODE>
      <BANKS_NAME>BANCOMER</BANKS_NAME>
      <BANK_ACOUNT_NUMBER>00987654</BANK_ACOUNT_NUMBER>
      <PAYMENT_METHOD_CODE>C</PAYMENT_METHOD_CODE>
      <DATE3>990107</DATE3>
    </BANK_ID>
    <CURRENCY_INFORMATION>
      <CURRENCY_COUNTRY_CODE>MEX</CURRENCY_COUNTRY_CODE>
      <EXCHANGE_RATE>9.60</EXCHANGE_RATE>
      <CURRENCY_RATE>DLS</CURRENCY_RATE>
    </CURRENCY_INFORMATION>
    <NAME>
      <ENTITY_IDENTIFIER_CODE>ST</ENTITY_IDENTIFIER_CODE>
      <CLIENTS_NAME>PIFF S CO.</CLIENTS_NAME>
    </NAME>
    <ADDRESS_INFORMATION>
      <ADDRESS_BUILDING-STREET-PO_BOX>Playa Regatas No.491
Col.Marte</ADDRESS_BUILDING-STREET-PO_BOX>
      <CITY-STATE-POSTAL_CODE-COUNTRY>D.F. CP08830 MEXICO</CITY-
STATE-POSTAL_CODE-COUNTRY>
    </ADDRESS_INFORMATION>
    <CLIENTS_GEOGRAPHIC_LOCATION>
      <CLIENTS_CITY_NAME>Distrito Federal</CLIENTS_CITY_NAME>
      <STATE_CODE>DF</STATE_CODE>
      <POSTAL_CODE>08830</POSTAL_CODE>
      <COUNTRY_CODE>52</COUNTRY_CODE>
    </CLIENTS_GEOGRAPHIC_LOCATION>
  </HEADER>
  <DETAIL>
    <CONSULTING_SERVICE_NUMBER_ASSIGNED>
      <CONSULTATION_NUMBER>1413</CONSULTATION_NUMBER>
    </CONSULTING_SERVICE_NUMBER_ASSIGNED>
    <CONSULTATION_DETAILS>
      <NUMBER_OF_INTERCHANGES>9</NUMBER_OF_INTERCHANGES>
      <CONSULTATION_CODE>C1</CONSULTATION_CODE>
    </CONSULTATION_DETAILS>
    <RATE_AND_CHARGES>
      <CONSULTATION_RATE>1000</CONSULTATION_RATE>
      <CHARGE>9000</CHARGE>
      <PRAPAID_AMOUNT>5000</PRAPAID_AMOUNT>
```

```
    </RATE_AND_CHARGES>
    <CURRENCY_INFORMATION>
      <CURRENCY_COUNTRY_CODE>USA</CURRENCY_COUNTRY_CODE>
      <EXCHANGE_RATE>9.60</EXCHANGE_RATE>
      <CURRENCY_RATE>DLS</CURRENCY_RATE>
    </CURRENCY_INFORMATION>
  </DETAIL>
  <SUMMARY>
    <TOTAL_CHARGES>
      <TOTAL_CHARGE_FOR_SERVICES>9000</TOTAL_CHARGE_FOR_SERVICES>
      <PREPAID_AMOUNT>5000</PREPAID_AMOUNT>
    </TOTAL_CHARGES>
    <BUSINESS_INSTRUCTIONS>
      <REFERENCE_NUMBER3>001003</REFERENCE_NUMBER3>
      <CLIENTS_ID>Consultoria Plan de Negocios via e-mail PIFF S
Co.</CLIENTS_ID>
    </BUSINESS_INSTRUCTIONS>
    <TRANSACTION_SET_INFORMATION>

<NUMBER_OF_SEGMENTS_INCLUDED_IN_BILL>1</NUMBER_OF_SEGMENTS_INCLUDED_
IN_BILL>

<TRANSACTION_SET_CONTROL_NUMBER>0001</TRANSACTION_SET_CONTROL_NUMBER
>
    </TRANSACTION_SET_INFORMATION>
  </SUMMARY>
</BILL>
```

FIGURA 21. EJEMPLO DE UN DOCUMENTO XML OBTENIDO DEL MOTOR EDI2XML

4. Motor EDI2XML o "parser". Este componente se encarga de accesar el archivo EDI.X12, analizarlo, traducir su contenido a XML y arrojar un documento completo y correcto de XML. Primero, el motor genera la estructura de datos KB (Knowledge Base) principal a partir de la cual construye el DTD del documento XML tomando línea por línea de la Base de Conocimientos (BC.txt). Posteriormente genera el "*Document Element*" leyendo línea por línea del archivo EDI.X12, primero identifica la sección del documento a la que dicho segmento pertenece, lo encuentra en la estructura KB, una vez encontrado el segmento en la estructura primero accesa a su etiqueta correspondiente en XML para utilizarla en el "*Document Element*" y a continuación va

accesando los campos del segmento en la estructura, así como sus atributos, para paralelamente ir extrayendo del documento EDI los valores de dichos campos y utilizarlos en la construcción del "Document Element". Una vez analizado todo el documento EDI y, después de haber recorrido toda la estructura de datos KB, la máquina EDI2XML arroja como resultado un archivo XML completo y correcto.

10.2 El traductor y su funcionamiento

Después de haber descrito los componentes del traductor, a continuación se presenta cómo interactúan entre sí dichos componentes y como trabaja el traductor:

El traductor recibe como entrada un documento EDI, utilizando una Base de Conocimientos en donde se definen las etiquetas de los segmentos, genera las estructuras de datos necesarias y traduce el contenido del archivo EDI a un archivo XML, obteniéndose como salida del mismo, un documento XML.

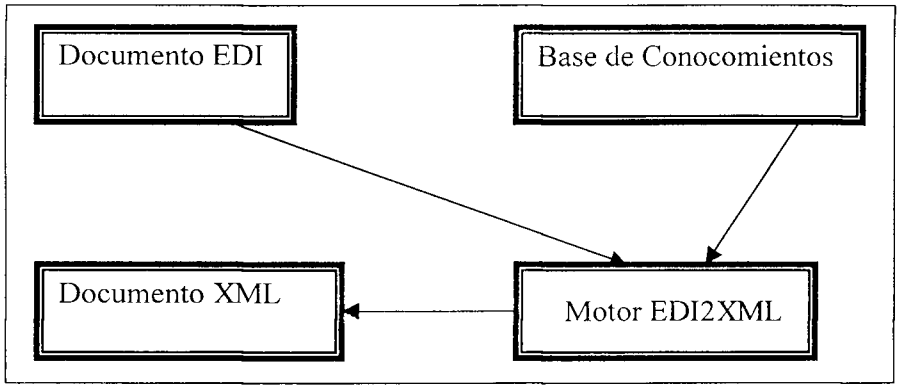


FIGURA 22. FUNCIONAMIENTO DEL MOTOR EDI2XML

En la figura 23 se muestra la estructura de datos principal utilizada en el traductor EDI2XML.

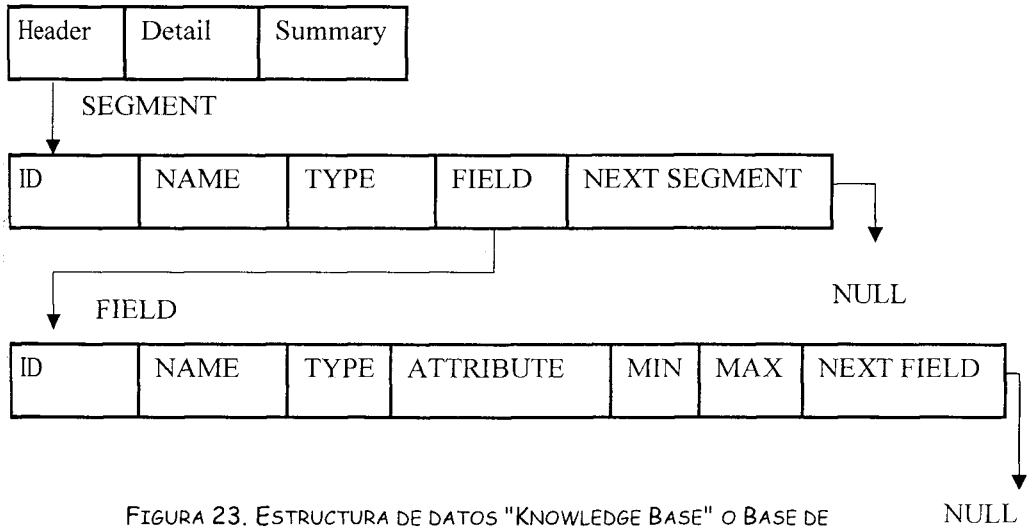


FIGURA 23. ESTRUCTURA DE DATOS "KNOWLEDGE BASE" O BASE DE CONOCIMIENTO

A continuación se muestra un esquema gráfico de cómo trabaja el motor "EDI2XML.EXE"

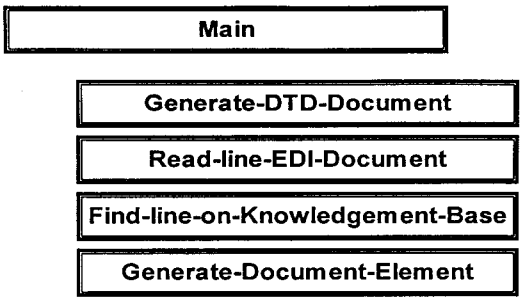


FIGURA 24. MOTOR "EDI2XML"

1) Se genera la estructura lógica del "*Document Type Definition*" a partir de la Base de Conocimientos.
2) Se lee línea por línea del documento EDI para encontrar su correspondiente en el DTD.
3) Una vez encontrado el Segmento o campo de EDI en la Base de conocimiento, se genera el "Document Element".
4) Y así sucesivamente hasta terminar de leer el documento EDI.

10.3 Manual de Usuario y Procedimiento para ejecutar el Motor "EDI2XML"

Requerimientos mínimos:

Para hacer uso del traductor "EDI2XML"

- Computadora con procesador 386 (25 Mhz)
- Sistema operativo MS-DOS Ver. 6

- 8 MB en memoria RAM

Para hacer uso del archivo generado en XML

- Computadora Pentium (100 MHz)
- Cualquier plataforma que permita el uso del navegador Microsoft Explorer 5.0 o superior
- 32 Mb en memoria RAM

Archivos contenidos en el directorio EDI2XML:

- EDI2XML.C: Código fuente del motor "EDI2XML.EXE".
- EDI2XML.EXE: Motor (programa) que hace la traducción de documentos EDI a XML
- BC.TXT: Base de conocimiento EDI-XML para realizar conversiones de documentos EDI como la "Factura de Conocimiento" del ITESM (no es un estándar) a un documento XML.
- EDIDOC1.TXT: Ejemplo 1 de documento EDI
- EJEM1.XML: Resultado de la traducción del documento EDIDOC1.TXT
- EDIDOC2.TXT: Ejemplo 2 de documento EDI
- EJEM2.XML: Resultado de la traducción del documento EDIDOC2.TXT
- EDIDOC3.TXT: Ejemplo 3 de documento EDI
- EJEM3.XML: Resultado de la traducción del documento EDIDOC3.TXT

- EDIDOC4.TXT: Ejemplo 4 de documento EDI
- EJEM4.XML: Resultado de la traducción del documento EDIDOC4.TXT
- EDIDOC5.TXT: Ejemplo 5 de documento EDI
- EJEM5.XML: Resultado de la traducción del documento EDIDOC5.TXT

Para poder hacer uso del traductor "EDI2XML" necesita grabar en un directorio tres archivos:

- Motor "EDI2XML.EXE"
- Base de Conocimientos "BC.TXT"
- Documento EDI a traducir

El motor se tiene que ejecutar en una ventana de MS-DOS, y los tres archivos mencionados anteriormente deben estar en el mismo directorio. La sintaxis es la siguiente:

Prompt> EDI2XML <documento EDI> <documento XML> [1]

Documento EDI: Nombre del documento EDI a ser traducido

Documento XML: Nombre del documento XML a ser generado por el traductor.

1: Opción para ver a detalle la traducción de los documentos

El **documento XML** obtenido del motor "EDI2XML" se puede visualizar en un navegador "Microsoft Explorer 5.0" o superior.

El **documento EDI** se obtiene del compilador EDI.X12.

La **Base de Conocimiento** la puede generar mediante el uso de una hoja de cálculo. Para definir un segmento siga éste orden:

COLUMNA 1:

"H" si el segmento pertenece al *Header*

"D" si el segmento pertenece al *Detail*

"S" si el segmento pertenece al *Summary*

COLUMNA 2:

Valor del ID del segmento en EDI (Definido en el estándar EDI.X12)

COLUMNA 3:

Valor del ID del segmento en XML

COLUMNA 4:

"M" si el segmento es de tipo mandatario (Definido en el estándar EDI.X12)

"O" si el segmento es de tipo opcional (Definido en el estándar EDI.X12)

Ejemplo:

H	B2	BILL_INFORMATION	O			

Para definir un tipo de transacción de un segmento se deben seguir los siguientes pasos:

Los datos que a continuación se muestran se deben poner en el renglón siguiente (o bloque de renglones) en el que se definió el segmento. Los datos están compuestos por los siguientes datos:

COLUMNA 1:

"-" es obligatorio e indica que no es un segmento

COLUMNA 2:

Valor del ID del campo del segmento en EDI (Definido en el estándar EDI.X12)

COLUMNA 3:

Valor del ID del campo del segmento en XML

COLUMNA 4:

"M" si el segmento es de tipo mandatario (Definido en el estándar EDI.X12)

"O" si el segmento es de tipo opcional (Definido en el estándar EDI.X12)

COLUMNA 5:

Primer atributo definido del campo del segmento (Definido en el estándar EDI.X12)

Puede ser "AN" "ID" "DT" "NO" "N2"

COLUMNA 6:

Número mínimo de caracteres de los cuales consta el valor del contenido del campo

COLUMNA 7:

Número máximo de caracteres de los cuales consta el valor del contenido del campo.

Ejemplo:

H	B2	BILL_INFORMATION	O			
-	B201	CONSULTING_TYPE_CODE	O	ID	2	2
-	B203	CITY_SERVED	O	ID	6	9
-	B208	METHOD_OF_PAYMENT	M	ID	2	2
-	B209	STATUS_REPORT_REQUEST_CODE	O	ID	1	1
-	B218	RECIVED_DOCUMENTATION_ID	O	ID	2	2

Cuando se inserte un nuevo campo en un segmento, éste se debe insertar en el mismo orden que tiene al estar "EDI.X12".

Al final de la Base de Conocimiento se debe insertar un segmento con valores "X" (exis).

La Base de Conocimiento se debe guardar como tipo 'TEXTO DELIMITADO POR TABULACIONES' y se debe nombrar "BC.TXT". En la Figura 25, se observa un ejemplo de la Base de Conocimientos, correspondiente a la "Factura de Asesoría".

H	ST	BILL_HEADER	M			
-	ST01	BILL_ID	M	ID	3	3
-	ST02	BILL_NO	M	AN	4	9
H	B2	BILL_INFORMATION	O			
-	B201	CONSULTING_TYPE_CODE	O	ID	2	2
-	B203	CITY_SERVED	O	ID	6	9
-	B208	METHOD_OF_PAYMENT	M	ID	2	2
-	B209	STATUS_REPORT_REQUEST_CODE	O	ID	1	1
-	B218	RECIVED_DOCUMENTATION_ID	O	ID	2	2
H	G3	COMPENSATION_INFORMATION	O			
-	G301	TECS_PERCENTAGE_COMISSION	O	R	2	5
-	G302	TOTAL_COMPENSATION_AMOUNT	M	N0	3	10
-	G303	NAME_CAMPUS	O	AN	1	35
D	LX	CONSULTING_SERVICE_NUMBER_ASSIGNED	M			
-	LX01	CONSULTATION_NUMBER	M	N0	1	6
D	L1	RATE_AND_CHARGES	O			
-	L102	CONSULTATION_RATE	O	R	1	9
-	L104	CHARGE	C	N2	1	9
-	L106	PRAPAID_AMOUNT	C	N2	1	9
D	C31	CURRENCY	M			
-	C311	CURRENCY_COUNTRY_CODE2	M	ID	3	3
-	C312	EXCHANGE_RATE2	O	R	4	6
-	C313	CURRENCY_CODE	O	ID	3	3
S	L3	TOTAL_CHARGES	M			
-	L305	TOTAL_CHARGE_FOR_SERVICES	O	N2	1	9
-	L307	PREPAID_AMOUNT	O	N2	1	9
S	SE	TRANSACTION_SET_INFORMATION	M			
-	SE01	NUMBER_OF_SEGMENTS_INCLUDED_IN_BILL	M	N0	1	10
-	SE02	TRANSACTION_SET_CONTROL_NUMBER	M	AN	4	9
X	X	X	M			
-	X	X	X	X	1	1

FIGURA 25. BASE DE CONOCIMIENTOS GENERADA EN HOJA DE CÁLCULO

Cuando existiera en el documento EDI algún segmento o un campo de segmento que no existe en la Base de Conocimiento, éste se desplegará entre comentarios. Por ejemplo, si en un documento EDIDOC2.TXT se tuvieran las siguientes líneas:

C2*1*BANAMEX*1330991-2*C*991215

A3*MEX*1.65*FR

N1*ST*LA FEMME

Como resultado en el documento XML se obtendría lo siguiente: (EJEM2.XML):

```
<BANK_ID>
<BANK_CLIENT_CODE EDI_ID="C201" REQ_DES="ID" TYPE="M" MIN="1"
MAX="1">1</BANK_CLIENT_CODE>
<BANKS_NAME EDI_ID="C203" REQ_DES="AN" TYPE="O" MIN="2"
MAX="17">BANAMEX</BANKS_NAME>
<BANK_ACOUNT_NUMBER EDI_ID="C205" REQ_DES="AN" TYPE="O" MIN="6"
MAX="17">13309912</BANK_ACOUNT_NUMBER>
<PAYMENT_METHOD_CODE EDI_ID="C206" REQ_DES="ID" TYPE="O" MIN="1"
MAX="1">C</PAYMENT_METHOD_CODE>
<DATE3 EDI_ID="C207" REQ_DES="DT" TYPE="O" MIN="6" MAX="6">991215</DATE3>
</BANK_ID>
- <!-- UNKNOWN: "A3" -->
- <!-- UNKNOWN: "MEX" -->
- <!-- UNKNOWN: "1.65" -->
- <!-- UNKNOWN: "FR" -->
- <!-- CLOSING UNKNOWN -->
<NAME>
<ENTITY_IDENTIFIER_CODE EDI_ID="N101" REQ_DES="ID" TYPE="M" MIN="2"
MAX="2">ST</ENTITY_IDENTIFIER_CODE>
<CLIENTS_NAME EDI_ID="N102" REQ_DES="AN" TYPE="C" MIN="1" MAX="35">LA
FEMME</CLIENTS_NAME>
</NAME>
```

Esto debido a que en el documento EDIDOC2.TXT existe la siguiente línea que no tiene una correspondencia en la Base de Conocimiento:

A3*MEX*1.65*FR

Por tanto se genera un segmento de datos desconocidos, que son mostrados en forma de comentarios:

```
- <!-- UNKNOWN: "A3" -->
- <!-- UNKNOWN: "MEX" -->
- <!-- UNKNOWN: "1.65" -->
- <!-- UNKNOWN: "FR" -->
        <!-- CLOSING UNKNOWN -->
```

Si solamente un campo desconocido existiera dentro de un segmento, éste se mostraría como un comentario dentro del bloque. Se puede observar en el siguiente ejemplo: (EDIDOC5.TXT)

N1*ST*PRODUCCIONES ANTONIETA.*CAMPO ADICIONAL*HOLA

El cual genera el siguiente código XML (ejem5.xml)

```
<NAME>
  <ENTITY_IDENTIFIER_CODE EDI_ID="N101" REQ_DES="ID" TYPE="M" MIN="2" MAX="2">
ST</ENTITY_IDENTIFIER_CODE>
      <CLIENTS_NAME EDI_ID="N102" REQ_DES="AN" TYPE="C" MIN="1"
MAX="35">PRODUCCIONES ANTONIETA.</CLIENTS_NAME>
<!-- UNKNOWN: "CAMPO ADICIONAL" -->
<!-- UNKNOWN: "HOLA" -->
  </NAME>
```

Dentro del segmento -N1- de EDI solamente se tienen 2 campos (consecutivos) registrados en la Base de Conocimientos: - N102 - y - N101-, por lo tanto los campos con el valor "campo adicional" y "hola" no existen y no pueden ser generados en una estructura para XML.

10.4 Código Fuente del Motor EDI2XML

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<io.h>
#define LONG_ID 10
#define LONG_NAME 100
#define LONG_BLOCK 3
#define LONG_TYPE 3
#define LONG_ATTRIBUTE 3
#define LONG_BUFFER 300
int p=0; /* Indicates if the user wants to see the details of
the translation */
                                /* 1 = with details, 0 =
without details */
int printSegmentflag=0;
typedef struct Field{
        char ID[LONG_ID];
        char Name[LONG_NAME];
        char Type[LONG_TYPE];
        char Attribute[LONG_ATTRIBUTE];
        int Min;
        int Max;
        struct Field * NextField;
}Fields;
typedef Fields * Fields_list;

typedef struct Segment{
        char ID[LONG_ID];
        char Name[LONG_NAME];
        char Type[LONG_TYPE];
        struct Field * Fields;
        struct Segment * NextSegment;
}Segments;
typedef Segments * Segments_list;

typedef struct Knowledge_Base {
        struct Segment * Header;
        struct Segment * Detail;
        struct Segment * Summary;
} DTD, Document_Element;


void ReadKBase(char *,DTD*);
void initialize_Field(Fields *);
void initilize_KB(DTD* );
void initialize_Segments(Segments *);
void initialize_KB(DTD*);
void print_KB(DTD*);
void GeneraXML(char*,char *,DTD*);

Segments* insert_Segment(Segments*, Segments*);
Segments* insert_Field(Segments*, Fields*);

/*********************************************
***/
int main(int argc, char *argv[])
{

char
EDIFileName[12]="\0",XMLDOCUMENT[12]="\0";
DTD *DTD1;
int p=atoi(argv[3]);
if(p!=1) p=0;
strcpy(EDIFileName,argv[1]);
strcpy(XMLDOCUMENT,argv[2]);
if(argc<3)
{
 printf("\n\n !!!! Syntax ERROR: ");
 printf("\n\n prompt> EDI2XML <source> <target> [1] ");
 printf("\n\n source: EDI file name to translate");
 printf("\n target: New XML file name to generate
(name.XML)");
 printf("\n     1: To display a detailed progress
translation");
 }
 else
 {
  DTD1=(DTD*)malloc(sizeof(DTD));
  initialize_KB(DTD1);
  ReadKBase("BC.txt",DTD1);
  if(p==1) print_KB(DTD1);
  GeneraXML(XMLDOCUMENT,EDIFileName,DTD1);
  printf("\n:::: The translation has been finished ");
  printf("\n YOU CAN SEE THE XML DOCUMENT
PRODUCED BY EDI2XML TRANSLATOR ");
  printf("\n IN THE FILE \"%s\" ",XMLDOCUMENT);
```

```
 printf("\n\n THANKS FOR USING OUR
TRANSLATOR ");
 printf("\n                GABRIELA CASTRO AND
NAHIM DE ANDA");
 getchar();

 }
return( 0);
}
/*****************************************/
void initialize_KB(DTD* DTD1){
                if(p==1) printf("\n:::: Initializing
Knowledgement Base structure ");
                DTD1->Header=NULL;
                DTD1->Detail=NULL;
                DTD1->Summary=NULL;
}
/*************************************/
void initialize_Segments(Segments * Segmentdata){
                strcpy(Segmentdata->ID,'\0');
                strcpy(Segmentdata->Name,'\0');
                strcpy(Segmentdata->Type,'\0');
                Segmentdata->Fields=NULL;
                Segmentdata-
>NextSegment=NULL;
}
/*********************************`************/
void initialize_Field(Fields * FieldData)
{
        strcpy(FieldData->ID,'\0');
        strcpy(FieldData->Name,'\0');
        strcpy(FieldData->Type,'\0');
        strcpy(FieldData->Attribute,'\0');
        FieldData->Min=0;
        FieldData->Max=0;
        FieldData->NextField=NULL;
}
/*********************************`************/
Segments* insert_Segment(Segments* document_section,
Segments* new_segment)
{
        Segments * document_section_aux;
        document_section_aux = document_section;
        if(document_section==NULL){
                document_section=new_segment;
                return(document_section);
        } else {
                document_section_aux =
document_section;
                while(document_section-
>NextSegment!=NULL)

        document_section=document_section-
>NextSegment;
                document_section-
>NextSegment=new_segment;
                return(document_section_aux);
        }
}
/*********************************************/
void print_KB(DTD *DTD1)
{
        Segments * document_section;
        Fields * field_aux;
        int a=0;
        char name_section[15]="Header";
        document_section = DTD1->Header;
        printf("\n-----------------------------------------------
--");
        printf("\n PRINTING FINAL STRUCTURE
KNOWLEDGEMENT BASE ");
        while(a<3){
                if(document_section!=NULL){
                  printf("\n\n**** DOCUMENT
SECTION = %s ****",name_section);
                  while(document_section!=NULL) {
                                printf("\n**
ID=%s, NAME=%s, TYPE=%s ", document_section->ID,
document_section->Name, document_section->Type);
                                field_aux =
document_section->Fields;

        while(field_aux!=NULL){

        printf("\n   FieldID=%s, NAME=%s,
TYPE=%s, ATTR.=%s, MIN=%d, MAX=%d ",
field_aux->ID, field_aux->Name, field_aux->Type,
field_aux->Attribute, field_aux->Min, field_aux->Max);

        field_aux=field_aux->NextField;
                                }

        document_section=document_section-
>NextSegment;
                  }
                }
                if(a==0){ document_section = DTD1-
>Detail; a++; strcpy(name_section,"Detail "); }
                else if(a==1){ document_section =
DTD1->Summary; a++; strcpy(name_section,"Summary
"); }
                else a++;
        }
}

/*********************************************/
Segments* insert_Field(Segments* document_section,
Fields* new_field)
{
        Segments * document_section_aux;
        Fields * field_aux;
        document_section_aux = document_section;
        while(document_section-
>NextSegment!=NULL)
                document_section=document_section-
>NextSegment;
        if(document_section->Fields==NULL){
                document_section->Fields=new_field;
                return(document_section_aux);
        } else {
                field_aux = document_section-
>Fields;
```

```
                    while(field_aux->NextField!=NULL)
                            field_aux=field_aux->NextField;
                    field_aux->NextField=new_field;
                    return(document_section_aux);
            }
}
/*********************************************
***/
void fill_Field(char *Block, char *ID, char *Name, char
*Type, char *Attribute,int Min, int Max, DTD *DTD1)
{
            Fields_list new_field;
            new_field=(Fields_list)malloc(sizeof(Fields));
//          printf("\n:::: Filling Knowledgement Base
structure fields");
            initialize_Field(new_field);
            strcpy(new_field->ID,ID);
            strcpy(new_field->Name,Name);
            strcpy(new_field->Type,Type);
            strcpy(new_field->Attribute,Attribute);
            new_field->Min = Min;
            new_field->Max = Max;
            new_field->NextField=NULL;
            if(strcmp(Block,"H")==0)
                    DTD1->Header=insert_Field(DTD1-
>Header,new_field);
            else if(strcmp(Block,"D")==0) {
                    DTD1->Detail=insert_Field(DTD1-
>Detail,new_field);
            } else if(strcmp(Block,"S")==0) {
                    DTD1-
>Summary=insert_Field(DTD1->Summary,new_field);
            } else {
                    printf("\n!!!! ERROR: Type Block
Segment Undefined on field's filling");
                    exit(0);
            }
}
/*********************************************
***/
void fill_KB(char *Block, char *ID, char *Name, char
*Type, DTD *DTD1)
{
            Segments_list new_segment;
//          printf("\n:::: Filling Knowledgement Base
structure segments ");
            new_segment=(Segments_list)malloc(sizeof(Se
gments));
            initialize_Segments(new_segment);
            strcpy(new_segment->ID,ID);
            strcpy(new_segment->Name,Name);
            strcpy(new_segment->Type,Type);
            new_segment->NextSegment=NULL;
            if(strcmp(Block,"H")==0)
                    DTD1-
>Header=insert_Segment(DTD1->Header,new_segment);
            else if(strcmp(Block,"D")==0) {
                    DTD1-
>Detail=insert_Segment(DTD1->Detail,new_segment);
            } else if(strcmp(Block,"S")==0) {
                    DTD1-
>Summary=insert_Segment(DTD1-
>Summary,new_segment);
            } else printf("\n!!!! ERROR: Type Block
Segment Undefined");
}

/*********************************************
***/
void generate_document_sections(FILE *KBase,DTD
*DTD1)
{
  int a=0;
  char name_section[15]="HEADER  ";
  char xmlversion[25];
  Segments * document_section;
            printf("\n:::: Generating XML DTD elements'
segments");
            strcpy(xmlversion,"<?xml version=\"1.0\"?>");
            fprintf(KBase, "%s\n<!DOCTYPE BILL
[",xmlversion);
            fprintf(KBase, "\n  <!ELEMENT BILL (
HEADER, DETAIL, SUMMARY)> ");
            document_section=DTD1->Header;
            while(a<3){
                    if(document_section!=NULL){
                      fprintf(KBase, "\n    <!ELEMENT
%s (",name_section);
                      while(document_section!=NULL) {
                                      fprintf(KBase,
"%s",document_section->Name);

            if(strcmp(document_section->Type,"M")==0)
fprintf(KBase, "+");
                                      else
if(strcmp(document_section->Type,"O")==0)
fprintf(KBase, "*");

            if(document_section->NextSegment==NULL)
fprintf(KBase, " )>");
                                      else
fprintf(KBase, ", ");

            document_section=document_section-
>NextSegment;
                      }
                    }
                    if(a==0){ document_section = DTD1-
>Detail; a++; strcpy(name_section,"DETAIL  "); }
                    else if(a==1){ document_section =
DTD1->Summary; a++;
strcpy(name_section,"SUMMARY "); }
                    else a++;
            }
}
/*********************************************
***/
void generate_segments_fields(FILE *KBase, DTD
*DTD1)
{
            Segments * document_section;
```

```
	Fields * field_aux;
	int a=0;
	char name_section[15]="HEADER";
	document_section = DTD1->Header;
	printf("\n:::: Generating XML DTD elements ");
	while(a<3){
		fprintf(KBase,"\n	<!--+++++ %s'S
ELEMENTS +++++ -->",name_section);
		if(document_section!=NULL){
		  while(document_section!=NULL) {
			field_aux =
document_section->Fields;
			fprintf(KBase,"\n
<!ELEMENT %s (",document_section->Name);

	while(field_aux!=NULL){

	fprintf(KBase, "%s",field_aux->Name);

	if(strcmp(field_aux->Type,"M")==0)
fprintf(KBase, "+");
			else
if(strcmp(field_aux->Type,"O")==0) fprintf(KBase, "*");
			if(field_aux-
>NextField==NULL) fprintf(KBase, " )>");
			else
fprintf(KBase, ", ");

	field_aux=field_aux->NextField;
				}

	document_section=document_section-
>NextSegment;
		  }
		}
		if(a==0){ document_section = DTD1-
>Detail; a++; strcpy(name_section,"DETAIL"); }
		else if(a==1){ document_section =
DTD1->Summary; a++;
strcpy(name_section,"SUMMARY"); }
		else a++;
	}
}

/*********************************************
***/
void generate_segments_fields_attribute(FILE *KBase,
DTD *DTD1)
{
	Segments * document_section;
	Fields * field_aux;
	int a=0;
	char name_section[15]="HEADER";
	document_section = DTD1->Header;
	printf("\n:::: Generating XML DTD elements'
attributes");
	while(a<3){
		fprintf(KBase,"\n	<!--+++++
%s'S ELEMENTS' ATTRIBUTES +++++ --
>",name_section);
		if(document_section!=NULL){
			while(document_section!=NULL) {
				field_aux =
document_section->Fields;

	while(field_aux!=NULL){

	fprintf(KBase, "\n	<!ELEMENT %s
(#PCDATA)>",field_aux->Name);

	fprintf(KBase, "\n	<!ATTLIST
%s",field_aux->Name);

	fprintf(KBase, "\n	EDI_ID NMTOKEN
\"%s\"",field_aux->ID);

	fprintf(KBase, "\n	REQ_DES
NMTOKEN \"%s\"",field_aux->Attribute);

	fprintf(KBase, "\n	TYPE	NMTOKEN
\"%s\"",field_aux->Type);

	fprintf(KBase, "\n	MIN	NMTOKEN
\"%d\"",field_aux->Min);

	fprintf(KBase, "\n	MAX	NMTOKEN
\"%d\" >",field_aux->Max);

	field_aux=field_aux->NextField;
				}

	document_section=document_section-
>NextSegment;
			}
		}
		if(a==0){ document_section = DTD1-
>Detail; a++; strcpy(name_section,"DETAIL"); }
		else if(a==1){ document_section =
DTD1->Summary; a++;
strcpy(name_section,"SUMMARY"); }
		else a++;
	}
}
/*********************************************
***/
Fields * Search_Field(int noField, Segments *
SegmentPtr)
{
	int a=1;
	Fields * Fields_aux;
	Fields_aux = SegmentPtr->Fields;
	while(a<noField&&Fields_aux!=NULL) {
	  Fields_aux=Fields_aux->NextField;
	  a++;
	}
	if(Fields_aux!=NULL&&p==1) printf("\n
Writting XML Document Element Field #%d with Name
\"%s\"",noField,Fields_aux->Name);
	return(Fields_aux);
}
```

```
/*********************************************
***/
Segments * Search_Segment(char *SegmentID, DTD *
DTD1)
{
        int flag=0, a=0;
        char tmp[10];
        Segments * Segments_aux;
        Segments_aux=DTD1->Header;
        while(a<3&&flag==0){
          while(flag==0&&Segments_aux!=NULL) {
                if(strcmp(Segments_aux-
>ID,SegmentID)==0){
                        if(a==0) strcpy(tmp,"
Header");
                        else if(a==1) strcpy(tmp,"
Detail");
                        else if(a==2) strcpy(tmp,"
Summary");
                        if(p==1) printf("\n%s
Writting XML Document Element as %s from segment
\"%s\"",Segments_aux->ID,Segments_aux->Name,tmp);
                        flag=1;

        if(a==1&&printSegmentflag==0)
printSegmentflag++;
                        else
if(a==2&&printSegmentflag==2) printSegmentflag++;
                }
                else Segments_aux=Segments_aux-
>NextSegment;
                }
                if(a==0&&Segments_aux==NULL) {
Segments_aux=DTD1->Detail; a++;}
                else
if(a==1&&Segments_aux==NULL) {
Segments_aux=DTD1->Summary; a++;}
            else a++;
        }
        return( Segments_aux );
}
/*********************************************
***/
void values_separation(char *buffer,FILE *KBase,DTD
*DTD1)
{
        int i=0,j=0,segID=0;
        char buffer_value[LONG_BUFFER];
        Segments * SegmentsPtr;
        Fields * FieldPtr;

        while(i<strlen(buffer)) {

while(buffer[i]!='*'&&i<strlen(buffer)&&buffer[i]!='\n') {
                        buffer_value[j]=buffer[i];
                        j++;
                        i++;
                }
                buffer_value[j]='\0';
                if(segID==0) {
        SegmentsPtr=Search_Segment(buffer_value,DT
D1);
                        if(SegmentsPtr==NULL)
                                fprintf(KBase,"\n
<!-- UNKNOWN: \"%s\" -->",buffer_value);
                        else {

        if(printSegmentflag==1){ fprintf(KBase,"\n
</HEADER>\n  <DETAIL>"); printSegmentflag++; }
                                else
if(printSegmentflag==3){ fprintf(KBase,"\n
</DETAIL>\n  <SUMMARY>"); printSegmentflag++; }
                                fprintf(KBase,"\n
<%s>",SegmentsPtr->Name);
                        }
                } else {
                        if(SegmentsPtr!=NULL)
FieldPtr=Search_Field(segID,SegmentsPtr);
                        else FieldPtr = NULL;
                        if(FieldPtr==NULL)
fprintf(KBase,"\n     <!-- UNKNOWN: \"%s\" --
>",buffer_value);
                        else fprintf(KBase,"\n
<%s>%s</%s>",FieldPtr->Name,buffer_value,FieldPtr-
>Name);
                }
                strcpy(buffer_value,"\0");
                i++;
                segID++;
                j=0;
  }
  if(SegmentsPtr==NULL)
        fprintf(KBase,"\n    <!-- CLOSING
UNKNOWN -->");
  else {
        fprintf(KBase,"\n    </%s>",SegmentsPtr-
>Name);
//      if(printSegmentflag>2){ fprintf(KBase,"\n
</DETAIL>\n  <SUMMARY>"); printSegmentflag++; }
  }
}

/*********************************************
***/
void generate_document_element(FILE *KBase,char
*EDIFileName, DTD * DTD1)
{
  FILE *EDIFile;
  char c;
  char buffer[LONG_BUFFER];
  int i=0;
//  printf("ARCHIVO: %s",EDIFileName);
  if(( EDIFile = fopen(EDIFileName,"r")) == NULL ) {
                printf("\n!!!! ERROR: EDI File  %s
was not FOUND",EDIFileName);
                exit(0);
  } else    {
                printf("\n:::: Reading and inserting
EDI'S values in XML File\n ");
                do {
```

```
                        c=fgetc(EDIFile);
                        buffer[i]=c;
                        i++;
                        if(c=='\n') {
                                buffer[i]='\0';
                                if(p==1)
printf("\n\n  Reading fields' values:  %s ",buffer);

        values_separation(buffer,KBase,DTD1);
                                i=0;
                                buffer[0]='\0';
                        }
                } while (!feof(EDIFile));
                fclose(EDIFile);
  }
}

/*********************************************
***/
void GeneraXML(char *File,char *EDIFileName, DTD
*DTD1)
{
  FILE *KBase;
  if(( KBase = fopen(File,"w")) == NULL ) {
                printf("\n!!!! ERROR: Impossible
generate XML Document %s",File);
                exit(0);
  } else {
                printf("\n:::: Generating XML
Document ");

        generate_document_sections(KBase,DTD1);

        generate_segments_fields(KBase,DTD1);

        generate_segments_fields_attributes(KBase,DT
D1);
                fprintf(KBase,"\n]>");
                fprintf(KBase,"\n<!-- INSTITUTO
TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE
MONTERREY, CAMPUS CIUDAD DE MEXICO ");
                fprintf(KBase,"\n\n    Document
generated by \"EDI2XML Translator\" ");
                fprintf(KBase,"\n    Property of
Engineers ");
                fprintf(KBase,"\n       Blanca Gabriela
Castro Campos and Nahim de Anda Martin , from ITESM
CCM");
                fprintf(KBase,"\n    Credits:  ");
                fprintf(KBase,"\n       M.C.S. Enrique
David Espinosa Carrillo; our Compilers II teacher at
ITESM CCM");
                fprintf(KBase,"\n       M.I.T. Karla
Paola Pulido Guillen; thesis: \"EDI y XML en el
Comercio Electronico entre Empresas\", ITESM CCM");
                fprintf(KBase,"\n    November 1999,
Mexico City.");
                fprintf(KBase,"\n-->");
                fprintf(KBase,"\n<BILL>\n
<HEADER>");
        generate_document_element(KBase,EDIFileNa
me,DTD1);
                fprintf(KBase,"\n
</SUMMARY>\n</BILL>");
                fclose(KBase);
                printf("\n:::: Successful XML
Document generation");
  }
}
/*********************************************
***/
void ReadKBase(char *File,DTD *DTD1)
{
  FILE *KBase;
  char Block[LONG_BLOCK], Type[LONG_TYPE],
ID[LONG_ID], Name[LONG_NAME],
Attribute[LONG_ATTRIBUTE], trash[3];
  int Max=0, Min=0, Position=0;
  printf("\n:::: Generating Knowledgement Base structure
");
  if(( KBase = fopen( File, "r" )) == NULL ) {
                printf("\n!!!! ERROR:
Knowledgement Base file %s NOT FOUND-",File);
                exit(0);
  } else {
                printf("\n:::: Building Knowledgement
Base structure with K.B.file %s...",File);
                do {
                        Position = ftell(KBase);

        fscanf(KBase,"%s%s%s%s",&Block,&ID,&Na
me,&Type);
                        if(p==1)
printf("\nBlock=%s, ID=%s, Name=%s,
Type=%s",Block,ID,Name,Type);

        fill_KB(Block,ID,Name,Type,DTD1);
                        strcpy(trash,"-");
                        while(strcmp(trash,"-
")==0&&strcmp(trash,"X")!=0) {
                                Position =
ftell(KBase);

        strcpy(Attribute,'\0');

        fscanf(KBase,"%s%s%s%s%s%d%d",&trash,&
ID,&Name,&Type,&Attribute,&Min,&Max);
                                if(strcmp(trash,"-
")==0)
fill_Field(Block,ID,Name,Type,Attribute,Min,Max,DTD1
);
                                if(p==1)
printf("\n---Tr=%s,ID=%s, Na=%s, Ty=%s, At=%s,
Mn=%d,
Mx=%d",trash,ID,Name,Type,Attribute,Min,Max);
                                }
                        fseek(KBase,Position,0L);
                }
while(strcmp(Block,"X")!=0&&strcmp(trash,"X")!=0);
```

```
        printf("\n:::: Successful
Knowledgement Base structure generation");
        fclose(KBase);
    }
}
```

11. ANEXO 2. TRADUCTOR DE XML A EDIFACT

En la figura 26 observamos el proceso que sigue un documento XML hasta convertirse en un documento EDIFACT.

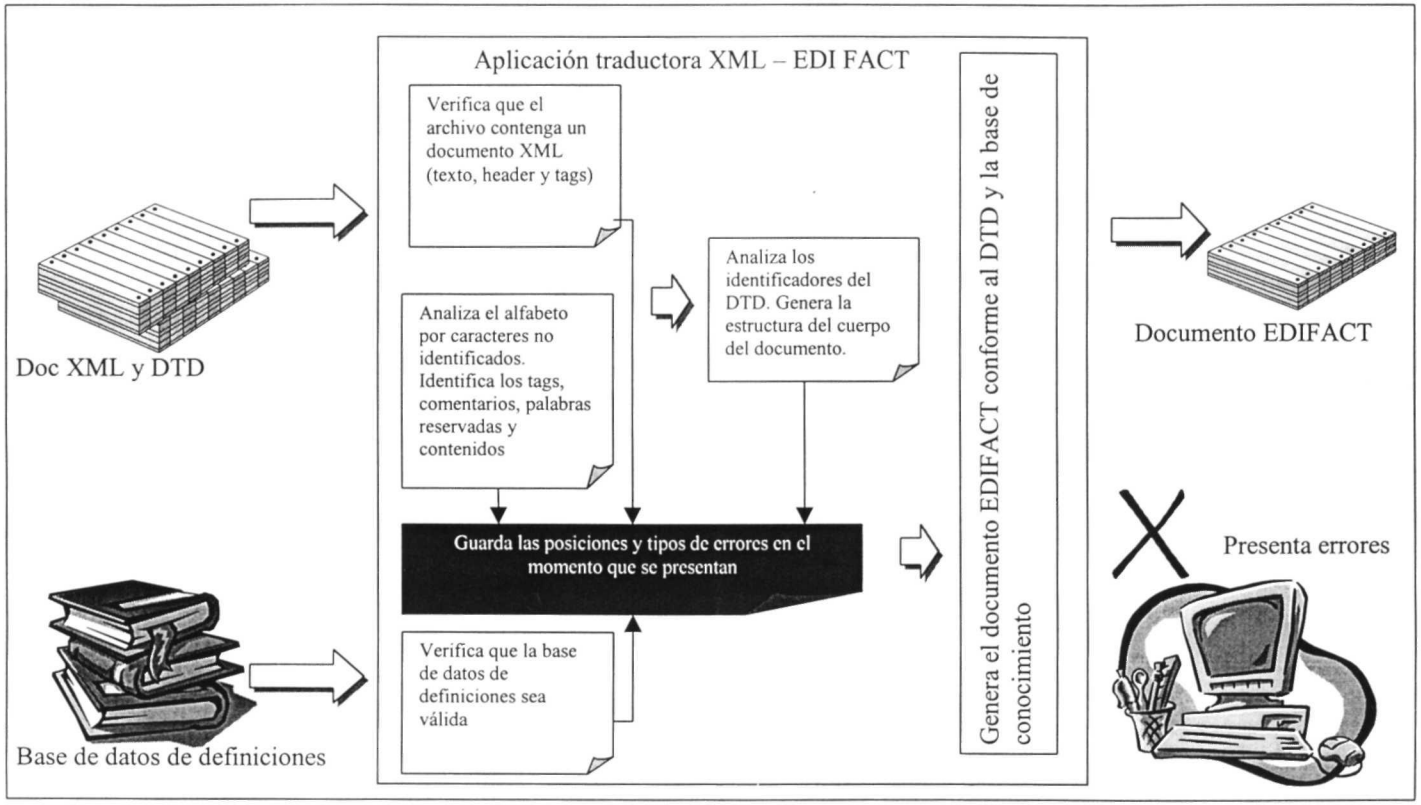


FIGURA 26. APLICACIÓN TRADUCTORA DE XML A EDIFACT

A continuación se presenta el códio fuente del traductor de XML a EDIFACT:

```
' Enum de los tipos de errores
Public Enum ErrType
   ERROR_SINTAXIS = 0
   ERROR_VAR_NO_DECLARADA
   ERROR_CORCHETE_SIN_ABRIR
   ERROR_CORCHETE_SIN_CERRAR
End Enum

' Tipo de dato que guarda la información de un error potencial
Public Type ErrorLocation
   Column  As Integer
   line    As Integer
   Error   As ErrType
End Type

' Constantes con los mensajes de la aplicación
Public Const MSG_FILENOTFOUND = "El archivo no se encontró."
Public Const MSG_INFO = "Información"
Public Const MSG_ANALYZEFILE = "¿Desea analizar este archivo?"
Public Const MSG_INVALIDFILE = "El archivo seleccionado no es válido."
Public Const MSG_QUESTION = "Pregunta"
Public Const MSG_CLOSEAPP = "¿Desea salir de la aplicación?"
Public Const MSG_ERRLANGLOAD = "Error al cargar definición del lenguaje!"
Public Const MSG_WARNING = "Advertencia!"
Public Const MSG_INVALIDLANGFILE = "El archivo de definición no es válido."
Public Const MSG_TRYAGAIN = "¿Desea seleccionar otro?"

Public Const DEFAULTLANGFILE = "\Lang\XML-EDIFACT.ini"
Public Const VALID_LANGSTRING = "1308740"

' Globales de la aplicación
Global gFileName    As String     ' Nombre del archivo a analizar
Global gTotalLines  As Long       ' Número de lineas en el archivo
Global gLangFile    As String     ' Nombre del archivo de definición
Global gReservedWords   As Long
Global gOperators       As Long
Global gProcedures      As Long
Global gComments        As Long
Global gTotalErrors     As Long
Global gTotalBrcktO     As Long
Global gTotalBrcktC     As Long
Global gTotalResIF      As Long
Global gTotalResTHEN    As Long
Global gTotalResBEGIN   As Long
Global gTotalResEND     As Long
Global gTotalVariables  As Long

' Arreglos de control del analizador sintáctico
```

```
Global gErrors()       As ErrorLocation
Global gBrcktOpen()    As ErrorLocation
Global gBrcktClose()   As ErrorLocation
Global gResIF()        As ErrorLocation
Global gResTHEN()      As ErrorLocation
Global gResBEGIN()     As ErrorLocation
Global gResEND()       As ErrorLocation
Global gVariables()    As String

' Estructuras del lenguaje
Global ResWords()      As String
Global Procedures()    As String
```

```
' RAU Declaración de funciones de la API de Windows
Declare Function GetWindowsDirectory Lib "kernel32" Alias "GetWindowsDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
Declare Function GetPrivateProfileInt Lib "kernel32" Alias "GetPrivateProfileIntA" (ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal nDefault As Long, ByVal lpFilename
As String) As Long
Declare Function GetPrivateProfileString Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As String, ByVal
lpReturnedString As String, ByVal nSize As Long, ByVal lpFilename As String) As Long

Public Sub Main()
  frmLangpath.Show 1
  If LoadLanguageStructures = False Then
    MsgBox MSG_ERRLANGLOAD, vbCritical, MSG_WARNING
    End
  Else
    frmMain.Show
  End If
End Sub
```

```
' ---------------------------------------------------------
' RAU
' Función: FileExists
' Parámetros:  Nombre del archivo
'
' Verifica la existencia de un archivo en base
' a su longitud
' ---------------------------------------------------------
Public Function FileExists(FileName As String) As Integer
   Dim x As Long
   On Error Resume Next
   x = FileLen(FileName)
   If x Then FileExists = True Else FileExists = False
End Function

' ---------------------------------------------------------
' RAU
' Función:    ReadFile
' Parámetros:  Nombre del archivo a leer (Global) gFileName
'
' Lee el archivo que se le pasa como parámetro
' y lo sube al lstProgram
' ---------------------------------------------------------
Public Function ReadFile() As Boolean
   Dim FileNumber
   Dim TextLine As String

   gTotalLines = 0
   ReadFile = False

   If Not (FileExists(gFileName)) Then
      MsgBox MSG_FILENOTFOUND, vbExclamation, MSG_INFO
      Exit Function
   End If

   FileNumber = FreeFile
   Open gFileName For Input As #FileNumber

   frmMain.ListProgram.Clear
   frmMain.txtLine = "0"
   frmMain.lblErrors = "0"

   Do While Not EOF(FileNumber)
      gTotalLines = gTotalLines + 1
      Line Input #1, TextLine
      frmMain.ListProgram.AddItem TextLine
   Loop
   Close #FileNumber
End Function
```

```
Private Function LoadLanguageStructures() As Boolean
  Dim Number As Integer

  LoadLanguageStructures = False
  If Not (FileExists(gLangFile)) Then
    MsgBox MSG_FILENOTFOUND, vbExclamation, MSG_INFO
    Exit Function
  End If

' Base de datos default para armado en memoria
  'Number = GetIniInt(R
  ReDim ResWords(70)
  ResWords(0) = "BILL HEADER"
  ResWords(1) = "BILL ID"
  ResWords(2) = "BILL NO"
  ResWords(3) = "BILL INFORMATION"
  ResWords(4) = "CONSULTING TYPE CODE"
  ResWords(5) = "CITY SERVED"
  ResWords(6) = "METHOD OF PAYMENT"
  ResWords(7) = "STATUS REPORT REQUEST CODE"
  ResWords(8) = "RECEIVED DOCUMENTATION ID"
  ResWords(9) = "COMPENSATION INFORMATION"
  ResWords(10) = "TEC'S % COMMISSION"
  ResWords(11) = "TOTAL COMPENSATION AMOUNT"
  ResWords(12) = "NAME CAMPUS"
  ResWords(13) = "REFERENCE NUMBER"
  ResWords(14) = "DATE"
  ResWords(15) = "TIME"
  ResWords(16) = "LETTER OF CREDIT REFERENCE"
  ResWords(17) = "NUMBER"
  ResWords(18) = "DATE"
  ResWords(19) = "EXPIRATION DATE"
  ResWords(20) = "CHARGE DATE"
  ResWords(21) = "SALES/DELIVERY TERMS"
  ResWords(22) = "TYPE OF DOCUMENT"
  ResWords(23) = "REFERENCE NUMBER"
  ResWords(24) = "SERVICE REQUESTED ON DATE"
  ResWords(25) = "SERVICE DELIVERED ON DATE"
  ResWords(26) = "BANK ID"
  ResWords(27) = "BANK CLIENT CODE"
  ResWords(28) = "BANK'S NAME"
  ResWords(29) = "BANK ACCOUNT NUMBER"
  ResWords(30) = "PAYMENT METHOD CODE"
  ResWords(31) = "DATE"
  ResWords(32) = "CURRENCY INFORMATION"
  ResWords(33) = "CURRENCY COUNTING CODE"
  ResWords(34) = "EXCHANGE RATE"
  ResWords(35) = "CURRENCY CODE"
  ResWords(36) = "NAME"
  ResWords(37) = "ENTITY IDENTIFIER CODE"
  ResWords(38) = "CLIENT'S NAME"
  ResWords(39) = "ADDRESS INFORMATION"
  ResWords(40) = "ADDRESS, BUILDING, STREAT, P.D. BOX"
  ResWords(41) = "CITY, STATE, POSTAL CODE, COUNTRY"
  ResWords(42) = "CLIENT'S GEOGRAPHIC LOCATION"
  ResWords(43) = "CLIENT'S CITY NAME"
  ResWords(44) = "STATE CODE"
  ResWords(45) = "POSTAL CODE"
  ResWords(46) = "CONTRY CODE"
  ResWords(47) = "CONSULTING SERVICE NO ASSIGNED"
  ResWords(48) = "CONSULTATION NUMBER"
  ResWords(49) = "CONSUTATION DETAILS"
  ResWords(50) = "NUMBER OF INTERCHANGES"
  ResWords(51) = "CONSULTATION  CODE"
  ResWords(52) = "RATE AND CHARGES"
  ResWords(53) = "CONSULTATION RATE"
  ResWords(54) = "CHARGE"
  ResWords(55) = "PREPAID AMOUNT"
  ResWords(56) = "CURRENCY"
  ResWords(57) = "CURRENCY COUNTRY CODE"
  ResWords(58) = "EXCHANGE RATE"
  ResWords(59) = "CURRENCY CODE"
  ResWords(60) = "TOTAL CHARGES"
  ResWords(61) = "TOTAL CHARGE FOR SERVICE"
  ResWords(62) = "PREPAID AMOUNT"
  ResWords(63) = "BUSINESS INTRUCTIONS"
  ResWords(64) = "REFERENCE NUMBER"
  ResWords(65) = "CLIENT'S ID"
  ResWords(66) = "DESCRIPTION"
  ResWords(67) = "TRANSACTION SET INFORMATION"
  ResWords(68) = "NUMBER OF SEGMENTS INCLUDED IN BILL"
  ResWords(69) = "TRANSACTION SET CONTROL NUMBER"

  LoadLanguageStructures = True
End Function
```

```
Public Function IsLangFileValid() As Boolean
   Dim retVal     As String
   Dim LangFile    As String

   IsLangFileValid = False
   LangFile = frmLangpath.txtFileName.Text
   retVal = String$(255, 0)
   Call GetPrivateProfileString("GLOBAL", "VALID", "", retVal, Len(retVal), LangFile)
   If Left(retVal, InStr(retVal, Chr(0)) - 1) = VALID_LANGSTRING Then IsLangFileValid = True
End Function

Public Sub AnalyzeFile()
   Dim i      As Long
   Dim line    As String

   ' Barre todo el archivo línea por línea
   For i = 0 To frmMain.ListProgram.ListCount
      line = frmMain.ListProgram.List(i)
      If AnalyzeAlphabet(line, i) = True Then
         'LookForLanguageElements Line, i
         AnalyzeLine line, i
      End If
      'AnalyzeLexic (Line)
   Next i
   PresentResults
End Sub

' Regresa falso en caso de encontrar comentario después de espacios y
' tabuladores
Private Function AnalyzeAlphabet(line As String, LineNumber As Long) As Boolean
   Dim length     As Integer
   Dim i          As Integer
   Dim ErrorFound  As Boolean
   Dim SpcOrTabs   As Boolean

   ' Longitud de la línea
   length = Len(line)
   AnalyzeAlphabet = True
   SpcOrTabs = True

   For i = 0 To length - 1
      Select Case Asc(Mid(line, i + 1, 1))
         ' Tab y espacio
         Case 9, 32
         ' Comentarios
         Case 37
            gComments = gComments + 1
            If SpcOrTabs = True Then Exit For
         '  A    Z  a    z
         Case 65 To 90, 97 To 122, 48 To 57, 59, 60, 62, 91, 93
            SpcOrTabs = False
         ' Operadores aritméticos y lógicos
         Case 42, 43, 45, 47, 38, 124, 33
            SpcOrTabs = False
            gOperators = gOperators + 1
         ' Corchetes
         Case 91, 93
            SpcOrTabs = False
         Case Else
            SpcOrTabs = False
            gTotalErrors = gTotalErrors + 1
            ReDim Preserve gErrors(gTotalErrors - 1)
            With gErrors(gTotalErrors - 1)
               .Column = i
               .Error = ERROR_SINTAXIS
               .line = LineNumber
            End With

            AnalyzeAlphabet = False
```

```
        End Select
    Next i
End Function

Private Sub LookForLanguageElements(line As String, LineNumber As Long)
    Dim length      As Integer
    Dim WordLength  As Integer
    Dim i           As Integer
    Dim Start       As Integer
    Dim Space       As Integer
    Dim ErrorFound  As Boolean
    Dim EOL         As Boolean
    Dim NewLine     As String
    Dim Word        As String

    EOL = False
    ' Longitud de la línea
    length = Len(line)
    NewLine = line

    Do  ' Linea vacía
        If Len(NewLine) = 0 Then Exit Do
        ' Busca tabulador y mueve el string al principio de él
        Start = 1
        While Start > 0
            Start = InStr(1, NewLine, Chr$(9))
            NewLine = Mid(NewLine, Start + 1, Len(NewLine) - Start)
        Wend

        ' Se encontró un comentario, obvia el resto
        If Left(NewLine, 1) = "%" Then Exit Do

        ' Busca un espacio para separar la palabra
        Space = InStr(1, NewLine, Chr$(32))

        If Space = 0 Then
            Space = Len(NewLine) + 1
            EOL = True
        End If
        Word = Left(NewLine, Space - 1)
        NewLine = Mid(NewLine, Space + 1, Len(NewLine))
        LookLexicWord (Word)
        If EOL = True Then Exit Do
    Loop
End Sub

Private Sub PresentResults()
    frmMain.txtReservadas = Format(gReservedWords, "###,###")
'   frmMain.txtOperadores = Format(gOperators, "###,###")
    frmMain.txtProcedimientos = Format(gProcedures, "###,###")
    frmMain.txtComentarios = Format(gComments, "###,###")
End Sub

Private Sub LookLexicWord(Word As String)
    Const r_IF = 0
    Dim i           As Integer

    ' Busca si es palabra reservada y guarda el tipo
    For i = 0 To 69
        If Word = ResWords(i) Then
            gReservedWords = gReservedWords + 1
            Exit For
        End If
    Next i
End Sub

Public Sub CleanGlobals()
    gProcedures = 0
    gReservedWords = 0
```

```
    gOperators = 0
    gTotalErrors = 0
    gComments = 0

    ' Limpia el arreglo de errores
    ReDim gErrors(0) As ErrorLocation
    ReDim gBrcktOpen(0) As ErrorLocation
    ReDim gBrcktClose(0) As ErrorLocation
    ReDim gResIF(0) As ErrorLocation
    ReDim gResTHEN(0) As ErrorLocation
    ReDim gResBEGIN(0) As ErrorLocation
    ReDim gResEND(0) As ErrorLocation
    ReDim gVariables(0) As String
End Sub

Public Function GetIniString(Section As String, Entry As String) As String
    Dim retVal

    retVal = String$(255, 0)
    GetPrivateProfileString Section, Entry, "", retVal, Len(retVal), gLangFile
    GetIniString = Left(retVal, InStr(retVal, Chr(0)) - 1)
End Function

Public Function GetIniInt(Section As String, Entry As String) As Integer
    GetIniInt = GetPrivateProfileInt(Section, Entry, 0, gLangFile)
End Function


Private Function AnalyzeLine(line As String, LineNumber As Long)
    Dim iPos       As Integer  ' Posición de corte de palabra
    Dim NewLine    As String   ' Remanente de la linea
    Dim length     As Integer  ' Longitud de la línea
    Dim NewTempLine As String  ' String para análisis

    length = Len(line)
    NewLine = line
    AnalyzeLine = NewLine
    If length = 0 Then Exit Function
    If isComment(NewLine) Then Exit Function

    ' Se busca el primer <
    iPos = InStr(1, NewLine, "<", vbTextCompare)

    If iPos = 0 Then
        NewTempLine = NewLine
    Else
        NewTempLine = Mid$(NewLine, iPos + 1, Len(NewLine) - (iPos + 1))
    End If

    LookLexicWord (NewTempLine)

End Function


Private Function GetSpacePos(line As String) As Integer
    GetSpacePos = InStr(1, line, " ", vbTextCompare)
End Function

Private Function GetTabPos(line As String) As Integer
    GetTabPos = InStr(1, line, Chr$(9), vbTextCompare)
End Function

Private Function GetColonPos(line As String) As Integer
    GetColonPos = InStr(1, line, ";", vbTextCompare)
End Function

Private Function GetBrktOpenPos(line As String) As Integer
    GetBrktOpenPos = InStr(1, line, "[", vbTextCompare)
End Function
```

```
Private Function GetBrktClosePos(line As String) As Integer
   GetBrktClosePos = InStr(1, line, "]", vbTextCompare)
End Function

Private Function GetLesser(iSpcPos As Integer, iTabPos As Integer, iColPos As Integer, iBrkt1Pos As Integer, iBrkt2Pos
As Integer) As Integer
   Dim iLesser As Integer
   Dim iPos    As Integer

   iLesser = 32765
   If iSpcPos < iLesser And iSpcPos > 0 Then
      iLesser = iSpcPos
      iPos = 1
   End If
   If iTabPos < iLesser And iTabPos > 0 Then
      iLesser = iTabPos
      iPos = 2
   End If
   If iColPos < iLesser And iColPos > 0 Then
      iLesser = iColPos
      iPos = 3
   End If
   If iBrkt1Pos < iLesser And iBrkt1Pos > 0 Then
      iLesser = iBrkt1Pos
      iPos = 4
   End If
   If iBrkt2Pos < iLesser And iBrkt2Pos > 0 Then
      iLesser = iBrkt2Pos
      iPos = 5
   End If
   GetLesser = iPos
End Function

Private Function isComment(line As String) As Boolean
   If InStr(1, line, "%", vbTextCompare) > 0 Then isComment = True
End Function

Option Explicit
' Guarda el número de error actual
Dim lCurrentError As Long

Private Sub btnAcercade_Click()
   frmAcercade.Show
End Sub

Private Sub btnAnalyze_Click()
   If txtFileName = "" Then Exit Sub
   CleanGlobals
   AnalyzeFile
   pnlMsg.Caption = ""
   If gTotalErrors > 0 Then
      lCurrentError = 0
      lblErrors.Caption = Format(gTotalErrors, "###,###")
   End If
   GoToError (lCurrentError)
End Sub

Private Sub btnClose_Click()
   If MsgBox(MSG_CLOSEAPP, vbQuestion + vbYesNo, MSG_QUESTION) = vbYes Then _
      Unload Me
End Sub

Private Sub btnExaminar_Click()
   With FileDialog
      .FileName = "doc.xml"
      .InitDir = App.Path & "Lang\"
      .DialogTitle = "Nombre del archivo a traducir"
      .Filter = "XML source file (*.xml,*.txt)|*.xml;*.txt|Todos los archivos (*.*)|*.*"
```

```
    End With

    On Error GoTo CancelSelected
    FileDialog.ShowOpen

    If Len(FileDialog.FileName) > 0 Then
       txtFileName = FileDialog.FileName
    End If

    Screen.MousePointer = vbHourglass
    gFileName = txtFileName.Text
    ReadFile
    lblTotalLines.Caption = Format(gTotalLines, "##,###")
    lblErrors = 0
    lblTotalLines = 0
    txtLine.Text = 1
    ListProgram.Selected(0) = True

    Screen.MousePointer = vbArrow
    Exit Sub
CancelSelected:
    Exit Sub
End Sub

Private Sub btnNextError_Click()
    If txtFileName = "" Then Exit Sub
    If lCurrentError = gTotalErrors - 1 Then Exit Sub
    lCurrentError = lCurrentError + 1
    GoToError (lCurrentError)
End Sub

Private Sub btnPrevError_Click()
    If txtFileName = "" Then Exit Sub
    If lCurrentError = 0 Then Exit Sub
    lCurrentError = lCurrentError - 1
    GoToError (lCurrentError)
End Sub

Private Sub Form_Load()
    CenterForm Me
End Sub

Private Sub ListProgram_Click()
    UpdateLineNumber
End Sub

Private Sub ListProgram_KeyDown(KeyCode As Integer, Shift As Integer)
    UpdateLineNumber
End Sub

Private Sub txtFileName_GotFocus()
    SelectControlText Me.txtFileName
End Sub

Private Sub UpdateLineNumber()
    Dim LineNumber As Long

    LineNumber = ListProgram.ListIndex

    txtLine.Text = Format(Str(LineNumber + 1), "##,###")
End Sub

Private Sub txtLine_GotFocus()
    SelectControlText Me.txtLine
End Sub

Private Sub txtLine_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
       If txtLine.Text > 0 And txtLine.Text <= ListProgram.ListCount Then
```

```
        ListProgram.ListIndex = Val(txtLine.Text - 1)
      Else
        UpdateLineNumber
      End If
    End If

    If KeyAscii = 27 Then UpdateLineNumber
End Sub

Private Sub GoToError(ErrorNum)
    pnlMsg.Caption = GetErrorMessage(gErrors(ErrorNum))
    ListProgram.Selected(gErrors(ErrorNum).line) = True
    pnlMsg.BackColor = RGB(255, 0, 0)
End Sub

Private Function GetErrorMessage(TheError As ErrorLocation)
    Dim sMsg As String
    Select Case TheError.Error
      Case ERROR_CORCHETE_SIN_ABRIR
        sMsg = Space(10) & "Falta abrir corchete"
      Case ERROR_CORCHETE_SIN_CERRAR
        sMsg = Space(10) & "Falta cerrar corchete"
      Case ERROR_SINTAXIS
        sMsg = Space(10) & "Error de sintáxis"
      Case ERROR_VAR_NO_DECLARADA
        sMsg = Space(10) & "Error no se declaró la variable"
    End Select
        sMsg = sMsg & Space(10) & "Columna:" & Str(TheError.Column) _
                & Space(5) & "Línea:" & Str(TheError.line + 1)
    GetErrorMessage = sMsg
End Function
```

12. BIBLIOGRAFÍA

Referencia	Datos
[AGULLÓ, 1997	AGULLÓ Soliveres, Pedro. "Desarrollo Cliente/Servidor: del las reglas de negocio (business rules)"; Revista Profesional para Programadores; 1997. http://www.ctv.es/USERS/pagullo/a_csbr.htm
[ASOCIACIÓN, 1998]	ASOCIACIÓN Civil Argentina de Codificación de Productos Comerciales, "Preparese para la Próxima Gran Revolucion: El Edi", 1998. http://www.codigo.com.ar/BOLETINES/BOLETIN_02/B_02I02.HTM
[ATHIÉ, ESCAMILLA, 1999]	ATHIÉ Rodrigo, ESCAMILLA Gabriel, "Traductor de XML a EDIFACT", 1999.
[BENÍTEZ, 1997]	BENÍTEZ Campoy, Juan Carlos; "Comercio Electrónico en Internet: El Futuro ya está aquí"; 1997 http://www.kriptopolis.com/set.html
[BOTTS, 1996]	BOTTS Steve, "The Internet as a Key Element of your EDI Strategy"; Premenos Corp.; 1996; http://www.commerce.net/events/conference/1996/edi/index.htm
[BROWN, 1999]	BROWN Eryn, "Streamlining the Supply Chain Pitney Bowes"; Fortune, Vol.139, No.10, May 24, 1999.
[BRYAN, 1998]	BRYAN Martin & MARCHAL Benoît. "Guidelines for using XML for Electronic Data Interchange"; XML/EDI Group; January 1998. http://www.geocities.com/WallStreet/Floor/5815/guide.htm O BIEN http://www.xmledi.net/
[CASTRO, DE ANDA, 1999]	CASTRO Blanca Gabriela, DE ANDA Nahim, "Traductor de EDI X12 a XML", 1999.
[CEDEX, 1998]	CEDEX Services International, "All About EDI: Communicating with Trading Partners Electronically", 1998. http://www.cedex.com/csi/Introduction/edi.html

Referencia	Datos
[CHUTE, 1996]	CHUTE Alan, "The Mytical Value of EDI Standards", File Express, Inc., October 1996. http://www.filex.com/filex/edimyth.html
[CLARKE, 1998]	CLARKE, Roger, "Electronic Data Intechange (EDI): An Introduction", Xmax Consultancy Pty Ltd, Australian National University, December 1998. http://www.anu.edu.au/people/Roger.Clarke/EC/EDIIntro.html
[CONE, 1998]	CONE, Eduard; "Affordable EDI"; Information Week On Line; May 18, 1998; http://www.iweek.com/682/82iuedi.htm
[CORNELLA, 1999]	CORNELLA, Alfonso, "Ciclo de Vida y Cadena de Valor en Información", EXTRA!-NET, El impacto de la información online en las organizaciones, Mensaje 162, ESADE Barcelona, 1999. http://www.extra-net.net/articulos/en961009.htm
[EMMELHAINZ, 1996]	EMMELHAINZ, Ph.D Margaret A, "The Strategic Role of Information in the Future of Logistics and Supply Chain Management", Logistics/ Transportation and Electronic Commerce EDI Forum, EDI Group, Ltd., Págs. 9-16, Volume 9, No.3, ISSN 1048-3047, 1996.
[ENIAC, 1998]	ENIAC Page, "Qué es EDI?", 1998, http://www.eniac.com/edihtm.htm
[ETXEBARRIETA, 1995]	ETXEBARRIETA, Jon Andoni; "Cómo mejorar la logística de proveedores y distribuidores?"; 1995 http://www.aecoc.es/notes/aecoc5/21d6.html
[FRÍAS, 1999]	FRÍAS Jeann, "Estudio y Desarrollo de la Seguridad en el Comercio Electrónico entre dos Entidades Productivas a través de Internet"; Tesis publicada por el Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Ciudad de México, 1999.
[FORRESTER, 1999]	FORRESTER Research; "Ecommerce Spend in Us 1998-2003"; NUA Analysis; 1999; http://www.nua.ie/surveys/analysis/graphs_charts/comparisons/ecommerce_us.html
[GOV, 1999]	"Introduction to Electronic Commerce"; 1999; http://www.ec.gov.sg/Intro1.htm

Referencia	Datos
[GUIDE, 1998]	Electronic Commerce - A Guide To Better Practice; "The Internet and EDI"; 1998; http://www.martech-intl.com/Best2/inetedi.htm
[HAISTING, 1996]	HAISTING, Larry, "Transportation Carriers Use Internet-to-EDI Solutions", EDI Forum, EDI Group, Ltd; Volume 9, No.3, Pages 53-57, ISSN 1048-3047.
[HALFHILL, 1999]	HALFHILL, Tom R; "XML: The Next Big Thing"; IBM Research Magazine; 1999; http://www.research.ibm.com/resources/magazine/1999/number_1/xml199.html
[IBM, 1999]	IBM Corporation; "Accesing XML on the Client"; IBM Research Magazine; 1999; http://www.software.ibm.com/xml/education/client/client.html
[ISA, 1998]	ISA, Information System Agreement, "Ocean Transportation Industry Guide to Electronic Data Interchange"; 003030; Standars Work Group; Second Printing; Washintong Publishing Company; July 1994.
[JENKINS, 1997]	JENKINS, Mike. "EDI: Today's Revolution in Logistics Management", EDI Forum, EDI Group, LTD; Volume 10, No1. Pages 68-70, ISNN 1048-3047
[JOCSON, 1999]	JOCSON, Edi P; " E-Commerce: EDI Gateway"; Intercommerce Gateway, Inc.; 1999; http://members.tripod.com/philcustom/ecommerce01.htm
[JOHNSTOWN, 1999]	JOHNSTOWN Regional Electronic Commerce Resource Center; "Background and Peripheral Information for the Electronic Data Interchange (Edi) Course"; 1999; http://www.bus.okstate.edu/sharda/mba5161/edipic.htm
[KALAKOTA, 1996]	KALAKOTA, Ravi & WHINSTON, Andrew B., "Frontiers of Electronic Commerce", Addisnon-Wesley Publishing Company, Inc., USA, 1996.
[LAVERY, 1996]	LAVERY, Hank, "Supply Chain Management and Electronic Commerce", Logistics/ Transportation and Electronic Commerce EDI Forum, EDI Group, Ltd., Págs. 22-28, Volume 9, No.3, ISSN 1048-3047, 1996.

Referencia	Datos
[MAYOR, 1995]	MAYOR, Tracy; "The Economics of EDI"; CIO Magazine; May 1, 1995; http://www.datatech-edi.com/newscio.htm
[MICROSOFT, 1999]	MICROSOFT Corporation; "Introduction to XML" 1999. Http://msdn.microsoft.com/xml/general/benefits.asp
[NOVELLA, 1999]	NOVELLA, Joaquín; " Mundialización, competitividad, comercio internacional, política industrial y empleo"; Fundación Cidoob; Barcelona; 1999. http://www.cidob.es/Catalan/Publicaciones/Afers/novella.html
[PHILLIPS,1999]	PHILLIPS International, Inc.; Industry Success Strategies for Electronic Data Interchange, Vol. III: Marketplace Profiles; 1999; http://www.ectoday.com/edivol3.htm
[PEAT, 1998]	BRUCE, Peat; "New XML Electronic Commerce Repository Proposal Announced"; Press Release; July 1998; http://www.oasis-open.org/cover/xmlEDIReposAnn.html
[PORTER, 1997]	PORTER, Alan L; CAUFFIEL, David A; "Future prospects for Electronic Commerce: A Survey"; EDI Forum, EDI Group, LTD; Volume 10, No1. Pages 64-67, ISNN 1048-3047
[RAMAN, 1999]	RAMAN, Dick ; "Setting up a UN Repository for XML/EDI"; CEFACT - SIMAC; Génova, 1999; http://www.edi-tie.com/edifact/XML-EDIrepositorySetup.htm
[RAMAN2, 1999]	RAMAN, Dick ; "UN-XML: The new way forward for UN/EDIFACT "; EEMA Annual Conference; "ParisSetting up a UN Repository for XML/EDI"; CEFACT - SIMAC; Paris, 1999; http://www.edi-tie.com/edifact/eemaECE99UN-XMLkaal/sld001.htm
[TOPKEN, 1998]	TOPKEN, Doug. "Logistics on the Information Superhighway"; EDI Forum, EDI Group, Ltd; Volume 9, No.3, Pages 73-79, ISSN 1048-3047.
[TURBAN, 1996]	TURBAN, Efraim; McLEAN Ephraim, WETHERBE James; "Information Technology for Management. Improving Quality and Productivity"; John Wiley & Sons, Inc.; USA, 1996.