

INTRODUCCIÓN A OFFICE OPEN XML



ECMA TC45
TOM NGO (NEXTPAGE), EDITOR

1 INTRODUCCIÓN

Office Open XML (OpenXML) es una propuesta de estándar abierto para documentos de procesamiento de texto, presentaciones y hojas de cálculo que diversas aplicaciones pueden implementar libremente en distintas plataformas. Su publicación beneficia a las organizaciones que desean implementar aplicaciones capaces de utilizar el formato, así como a entidades comerciales y gubernamentales que obtienen dicho software, y formadores y autores que enseñan el formato. En última instancia, cualquier usuario puede aprovecharse de las ventajas de un estándar XML para sus documentos, entre las que se cuentan la estabilidad, conservación, interoperabilidad y evolución permanente.

El proceso para la estandarización de OpenXML fue llevado a cabo por Ecma International a través de su Comité Técnico 45 (TC45), que incluye representantes de Apple, Barclays Capital, BP, la Biblioteca Británica, Essilor, Intel, Microsoft, NextPage, Novell, Statoil, Toshiba y la Biblioteca del Congreso de Estados Unidos (1).

Estas notas del producto constituyen una introducción a OpenXML. Léalas para:

- Entender la finalidad de OpenXML y la estructura de su especificación.
- Conocer sus características: cómo soluciona la compatibilidad con versiones anteriores, la conservación, la extensibilidad, los esquemas personalizados, los subconjuntos, la multiplicidad de plataformas, la internacionalización y la accesibilidad.
- Saber cómo seguir la estructura de alto nivel de cualquier archivo OpenXML y navegar rápidamente a cualquier parte de la especificación sobre la que necesita más detalles.

2 FINALIDAD DEL ESTÁNDAR

Desde el principio, OpenXML fue diseñado para poder representar de manera fiel el corpus preexistente de documentos de procesamiento de texto, presentaciones y hojas de cálculo codificados en los formatos binarios definidos por Microsoft Corporation. El proceso de estandarización consiste en trasladar a XML las capacidades requeridas para representar el corpus existente, ampliarlas, proporcionar documentación detallada y hacer posible la interoperabilidad. En el momento de escribir este documento, más de 400 millones de usuarios generan documentos en formatos binarios que se estima superan los 40 mil millones y cada año se generan miles de millones más.

Los formatos binarios originales de estos archivos se crearon en un momento en el que el espacio era un bien escaso y el tiempo de análisis afectaba gravemente a la experiencia del usuario. Se basaban en la serialización directa de estructuras de datos en memoria que utilizaban las aplicaciones de Microsoft® Office®. La infraestructura de estándares, redes y hardware de hoy en día (especialmente XML) hacen posible un nuevo diseño que favorece la implementación en diversas plataformas por parte de diferentes proveedores, al mismo tiempo que permite la evolución.

En consonancia con estos avances técnicos, los mercados se han diversificado y ahora incluyen una nueva gama de aplicaciones que los programas elementales de edición de documentos no contemplaban originalmente. Entre estas nuevas aplicaciones se encuentran las que:

- Generan documentos automáticamente a partir de datos económicos
- Extraen datos económicos de documentos y suministran dichos datos a aplicaciones empresariales
- Realizan tareas restringidas sobre una pequeña parte de un documento, manteniendo siempre su capacidad de edición
- Proporcionan accesibilidad a grupos de usuarios con necesidades especiales, como los invidentes
- Se ejecutan en diversos tipos de hardware, incluidos los dispositivos móviles

Quizás el problema de mayor importancia sea el de la conservación a largo plazo. Nos hemos acostumbrado a producir información en cantidades que crecen exponencialmente. Al mismo tiempo, codificamos dicha información a través de representaciones digitales tan estrechamente unidas a los programas que las generaron que, al cabo de una década o dos, nos resulta muy difícil leerlas sin que se pierdan datos importantes. Conservar la inversión financiera e intelectual de estos documentos, tanto de los existentes como de los nuevos, se ha convertido en una prioridad acuciante.

La emergencia de estas cuatro fuerzas (la adopción generalizada de formatos binarios, los avances tecnológicos, las fuerzas del mercado que demandan una diversidad de aplicaciones y la dificultad creciente de la conservación a largo plazo) ha generado una necesidad imperiosa de definir un formato XML abierto y migrar a dicho formato los millones de documentos existentes con el mínimo de pérdida de datos posible. Asimismo, la estandarización del formato XML abierto y su mantenimiento a lo largo del tiempo crea un entorno que permite a cualquier organización confiar en la estabilidad actual de la especificación y tener la seguridad de que su evolución posterior contará con las comprobaciones y controles propios de los procesos de estándares.

Existen varios estándares y especificaciones de documentos. Entre ellos se incluyen HTML, XHTML, PDF y sus subconjuntos, ODF, DocBook, DITA y RTF. Al igual que ocurre con los numerosos estándares que representan imágenes de mapa de bits, como TIFF/IT, TIFF/EP, JPEG 2000 y PNG, cada uno de ellos se creó por una finalidad distinta. OpenXML nace de la necesidad de un estándar que englobe las características representadas en el corpus actual de documentos. Hasta donde sabemos, se trata del único formato de documento XML compatible con todas las características de los formatos binarios.

OpenXML define formatos para documentos de procesamiento de texto, presentaciones y hojas de cálculo. Cada tipo de documento se especifica a través de un lenguaje de marcado principal: WordprocessingML, PresentationML o SpreadsheetML. Los mecanismos incrustados permiten que un documento de cualquiera de estos tres tipos contenga material en otros lenguajes de marcado principales y en varios lenguajes de marcado compatibles.

La especificación contiene material normativo (material que define OpenXML) e informativo (material que ayuda a la comprensión por parte del lector pero que no es prescriptivo). Está estructurado en partes que se ajustan a las necesidades de una variedad de audiencias.

- | | |
|--|--|
| Parte 1. Aspectos básicos.
165 páginas | <ul style="list-style-type: none"> — Define el vocabulario, convenciones de nomenclatura y abreviaturas. — Resume los tres lenguajes de marcado principales y los lenguajes de marcado compatibles. — Establece condiciones para la conformidad y proporciona directrices de interoperabilidad. — Describe las restricciones incluidas en las Convenciones de empaquetado abierto que se aplican a cada tipo de documento. |
| Parte 2. Convenciones de empaquetado abierto.
125 páginas | <ul style="list-style-type: none"> — Define las Convenciones de empaquetado abierto (OPC, siglas de Open Packaging Conventions). Cada archivo OpenXML consta de un conjunto de secuencias de bytes denominadas “partes” recopiladas en un contenedor denominado “paquete”. El formato de empaquetado lo definen las OPC. — Describe una implementación física recomendada de las OPC que utiliza el formato de archivo Zip. — Declara los esquemas XML para las OPC como definiciones de esquemas XML (XSD) (2), en un anexo publicado solamente en soporte electrónico. El anexo incluye también representaciones no normativas de los esquemas que utilizan RELAX NG (ISO/IEC 19757-2) (3). |
| Parte 3. Manual.
466 páginas | <ul style="list-style-type: none"> — Presenta las características de cada lenguaje de marcado, proporcionando el contexto e ilustrando los distintos elementos mediante ejemplos y diagramas. Esta parte es informativa, no normativa. — Describe los recursos para el almacenamiento de datos XML personalizados en un paquete para hacer posible la integración con datos económicos. |
| Parte 4. Referencia de lenguaje de marcado.
5.756 páginas | <ul style="list-style-type: none"> — Define cada elemento y atributo, la jerarquía de relaciones principal y secundaria de los elementos y las semánticas adicionales según corresponda. Esta parte está destinada a su uso como referencia siempre que se necesite información detallada acerca de un elemento o atributo. |

- Define los recursos para el almacenamiento de datos XML personalizados.
 - Declara los esquemas XML para los lenguajes de marcado como XSD (2), en un anexo publicado solamente en soporte electrónico. El anexo los expresa también de una forma no normativa mediante RELAX NG (ISO/IEC 19757-2) (3).
- Parte 5. Extensibilidad y compatibilidad de marcado.
34 páginas
- Describe recursos para la extensión de documentos OpenXML.
 - Especifica elementos y atributos mediante los cuales pueden interoperar aplicaciones con extensiones diferentes.
 - Expresa las reglas de extensibilidad mediante NVDL (ISO/IEC 19757-4) (4).

Para poder leer y navegar con facilidad por estos documentos, las versiones electrónicas tienen muchos vínculos internos activos. En concreto, toda la parte 4 cuenta con vínculos a elementos principales y secundarios.

4 PROPIEDADES DEL ESTÁNDAR

Esta sección le prepara para investigar OpenXML mediante la descripción de algunas de sus propiedades más importantes. Cada subsección describe una de estas propiedades y corresponde a características específicas de OpenXML.

- En “Interoperabilidad” se describe cómo OpenXML es independiente de los formatos, características y entornos de tiempo de ejecución propietarios, lo que otorga a los programadores una amplia gama de posibilidades de elección.
- En “Internacionalización” se mencionan unos cuantos procedimientos representativos mediante los cuales OpenXML permite la compatibilidad con todos los grupos de lenguajes más importantes.
- En “Escasas barreras para la adopción por parte del programador”, “Diseño compacto” y “Modularidad” se muestran los procedimientos concretos a través de los cuales OpenXML evita o elimina los obstáculos prácticos que puedan tener las distintas entidades para la implementación: curva de aprendizaje, conjunto mínimo de características y rendimiento.
- En “Migración de alta fidelidad” se describe cómo OpenXML se ajusta al objetivo global que es la conservación de la información, que incluye la intención general del creador original, tanto en documentos preexistentes como nuevos.
- En “Integración con datos económicos” se describe el modo en el que OpenXML incorpora información económica en esquemas personalizados para posibilitar la integración y la reutilización de información entre aplicaciones de productividad y sistemas de información.
- En “Espacio para la innovación” se expone cómo OpenXML se prepara para el futuro mediante la definición de mecanismos de extensibilidad adicionales y la garantía de la interoperabilidad entre aplicaciones con conjuntos de características distintos.

El resto del presente documento, incluida esta sección, es un recorrido temático por OpenXML. Las referencias a la especificación presentan todas la forma §Parte: sección. Subsección. Por ejemplo, §1:2.5 corresponde a la Parte 1, Sección 2.5 de la especificación. En las referencias a otros títulos de este documento se utiliza el nombre.

4.1 INTEROPERABILIDAD

Los programadores pueden escribir aplicaciones que consumen y producen OpenXML en una multiplicidad de plataformas.

En primer lugar, la interoperabilidad de OpenXML se ha logrado gracias a numerosas contribuciones, modificaciones y revisiones de la especificación por parte de los miembros del comité Ecma TC45 (1), cuyas procedencias e intereses corporativos son diferentes. Representantes implicados:

- Proveedores (Apple, Intel, Microsoft, NextPage, Novell y Toshiba) con varios sistemas operativos (Linux, MacOS y Windows) y múltiples posibilidades de utilización de OpenXML.
- Empresas (BP, Barclays Capital, Essilor, Statoil) con fuertes inversiones en contenido existente, incluidos sistemas de transacción esenciales.
- La Biblioteca Británica y la Biblioteca del Congreso de Estados Unidos, ambas muy directamente interesadas en la conservación de documentos.

Durante la preparación, los miembros del comité plantearon y resolvieron cientos de problemas relativos a la directiva, claridad, semántica y posible dependencia del entorno. Grupos representativos de problemas y otras actividades incluidas:

- Características que hacen posible la independencia de plataforma de mecanismos que eran propietarios en los formatos binarios originales.
- Condiciones para la conformidad.
- Contenido de los esquemas.
- Representaciones alternativas para esquemas y mecanismos de extensibilidad mediante RELAX NG (ISO/IEC 19757-2) y NVDL (ISO/IEC 19757-4) (4).
- Desarrollo de herramientas para analizar automáticamente y visualizar los esquemas.
- Internacionalización.
- Descripciones completas, correctas y claras en toda la especificación, en muchos casos como resultado del intento de implementar fragmentos de la misma.

El resto de esta subsección destaca áreas específicas en las cuales OpenXML se desvía de los formatos binarios originales en bien de la interoperabilidad.

Uno de los requisitos fundamentales para la interoperabilidad es la independencia respecto a cualquier tipo concreto de contenido de origen.

- OpenXML no presenta restricciones en cuanto al tipo de imagen, audio o vídeo. Por ejemplo, las imágenes pueden ser GIF, PNG, TIFF, PICT, JPEG o cualquier otro tipo (§1:14.2.12).
- Los controles incrustados pueden ser de cualquier tipo, como Java o ActiveX (§1:15.2.8).
- Las especificaciones de fuentes de WordprocessingML pueden incluir parámetros de medición de fuentes e información PANOSE para ayudar a encontrar una fuente de sustitución si no está disponible la original (§3:2.10.5).

Además, OpenXML se escapa a la dependencia del entorno de tiempo de ejecución de la aplicación que generó el documento.

- El ejemplo clásico se produce con una aplicación o control externo que genera una imagen para un fragmento de la superficie de presentación. Para el caso en que el control o la aplicación no estén disponibles o no se puedan ejecutar en un entorno de tiempo de ejecución dado, el archivo de documento puede contener una representación de imagen. Este mecanismo existe también en los formatos binarios antiguos.
- OpenXML presenta un mecanismo más general denominado Bloque de contenido alternativo (§3:2.18.4), que se puede utilizar en varias situaciones en las que una aplicación consumidora puede no ser capaz de interpretar lo escrito por una aplicación productora. Normalmente, se utiliza en el contexto de la extensibilidad. Este mecanismo se describe más adelante en la subsección “Espacio para la innovación”.
- OpenXML se escapa a la dependencia de cualquier parámetro significativo en un entorno de producción de documentos, pero puede que no ocurra lo mismo en un entorno de consumo. Por ejemplo, el parámetro CT_SYSCOLOR es un índice en una tabla de colores en el entorno de producción. Para hacer posible la portabilidad a un entorno de consumo distinto, PresentationML permite al productor almacenar en caché el color del sistema que estaba en uso en el momento en que se creó el documento.

Y lo que es más importante, para terminar, Office OpenXML se ajusta a los estándares W3C como XML (5) y espacios de nombres XML (6). Solamente este hecho ya permite un nivel básico de interoperabilidad entre todas las plataformas y sistemas operativos adheridos a estos estándares abiertos.

4.2 INTERNACIONALIZACIÓN

OpenXML es compatible con las características de internacionalización requeridas por idiomas tan distintos como el árabe, chino (tres variantes), hebreo, hindi, japonés, coreano, ruso y turco.

OpenXML admite de manera intrínseca Unicode porque es XML. Además, OpenXML cuenta con un completo conjunto de características de internacionalización que se han ido perfeccionando con el paso de los años. El que sigue es un listado representativo:

Orientación de texto: OpenXML admite idiomas de orientación tanto de izquierda a derecha como de derecha a izquierda. También admite idiomas bidireccionales como el árabe, persa, urdu, hebreo o yiddish, los cuales se escriben de derecha a izquierda pero pueden presentar fragmentos de texto insertado escrito de izquierda a derecha. En WordprocessingML, la dirección del texto se puede controlar tanto a nivel de párrafo (§4:2.3.1.6) como a nivel de un segmento dentro de un párrafo (§4:2.3.2.28). Del mismo modo, el texto DrawingML se puede controlar a nivel de cuerpo de texto (§4:5.1.5.1.1), de párrafo (§4:5.1.5.2.2) y dentro de viñetas con números (§4.5.1.5.4).

Flujo de texto: en WordprocessingML, la dirección del flujo de texto se puede controlar a nivel de sección o tabla (§4:2.3.1.41), o a nivel de párrafo (§4:2.3.2.28). En los niveles de sección y tabla, el flujo de texto se puede controlar en dirección vertical y horizontal. Esto permite a OpenXML la compatibilidad con todos los diseños de texto posibles. Por ejemplo, líneas verticales que van de arriba abajo y que se organizan de izquierda a derecha, para admitir el mongol. Esto afecta al diseño de listas, tablas y otros elementos de presentación. DrawingML también utiliza parámetros Kumimoji en los niveles de párrafo y segmento para que el texto discurra horizontalmente y los números verticalmente (§4:5.1.5.2.3, §4:5.1.5.3.9). En WordprocessingML (§4:2.3.1.16) y PresentationML (§4:4.3.1.15), el flujo de caracteres también se puede especificar mediante el empleo de parámetros Kinsoku para definir qué caracteres pueden iniciar o terminar una línea de texto.

Representación de números: para el formato de campo en WordprocessingML (§4:2.16.4.3), numeración de párrafos y listas en WordprocessingML (§4:2.9), y numeración en DrawingML (§4:5.1.5.4, §4:5.1.12.61), se puede aplicar formato a los números utilizando cualquiera de las varias docenas de formatos de número posibles, incluyendo hiragana, árabe, abjad, tailandés, texto cardinal (por ejemplo, “ciento treinta y tres”), chino, coreano (Chosung o Ganada), hebreo, hindi, japonés, vietnamita o números romanos. Estos recursos también admiten valores de base decimal arbitrarios (por ejemplo, “1,00” frente a “1.00”) y separadores de lista. El formato de número internacionalizado es especialmente sólido en SpreadsheetML, que admite todas estas características en los formatos de celda (§4:3.8.30) y en referencias a datos externos (§4.3.13.12).

Representación de fechas: en WordprocessingML (§4:2.18.7) y SpreadsheetML (§4:3.18.5), las fechas del calendario se pueden escribir utilizando el sistema gregoriano (tres variantes), el hebreo, el hijri, el japonés (era del emperador), el coreano (era de Tangún), el Saka, el taiwanés y el tailandés.

Fórmulas: la especificación de fórmula en SpreadsheetML proporciona varias funciones de conversión relacionadas con la internacionalización, como BAHTTEXT (§4:3.17.7.22), JIS (§4:3.17.7.185) y ASC (§4:3.17.7.11).

Identificadores de idioma: en WordprocessingML (§4:2.3.2.18) y DrawingML (§4:5.1.5.3), cada párrafo y segmento pueden etiquetarse con un identificador de idioma, el cual permite a una aplicación seleccionar las herramientas de revisión y otras funcionalidades propias del idioma. Además de un identificador para cada idioma, OpenXML admite la asignación de un nombre a un conjunto de caracteres, una familia de fuentes y un valor PANOSE para que la aplicación pueda elegir un conjunto de caracteres de sustitución adecuado cuando no haya compatibilidad local.

4.3 ESCASAS BARRERAS PARA LA ADOPCIÓN POR PARTE DEL DESARROLLADOR

Un programador experimentado puede empezar a escribir aplicaciones sencillas conformes con OpenXML a las pocas horas de haber empezado a leer la especificación.

Aunque la especificación describe un amplio conjunto de características, una aplicación conforme con OpenXML no necesita ser compatible con todas ellas. La declaración de conformidad (§1:2) requiere simplemente que un consumidor conforme “no rechace ningún documento compatible del tipo de documento [previsto]” y que un productor conforme “pueda producir documentos compatibles” (§1:2.5). También proporciona directrices de interoperabilidad que establecen la función de las semánticas de los elementos (§1:2.6).

Una aplicación conforme puede tener funcionalidades muy bien definidas. Puede tratarse, por ejemplo, de un procesador por lotes que simplemente actualiza avisos de derechos de autor en un conjunto de documentos de procesamiento de texto. O de un lector de texto a voz que entiende lo suficiente de una presentación con diapositivas como para representar su contenido de texto en audio a medida que el usuario navega de una diapositiva a otra. La estructura del formato del archivo permite que dichos programas se escriban partiendo de un conocimiento mínimo de OpenXML. En concreto:

- El formato de archivo es compatible con estándares consolidados, especialmente XML y ZIP, para los cuales existen herramientas avanzadas.
- El formato de archivo utiliza las Convenciones de empaquetado abierto, que combinan XML y ZIP con mecanismos estándar para expresar las relaciones dentro de un archivo. Por esta razón, se puede explorar el contenido de un archivo sin tener conocimiento de la semántica de etiquetas de ningún lenguaje de marcado principal o auxiliar en OpenXML.
- Se puede tener acceso y modificar elementos en niveles profundos del árbol XML sin alterar el resto de la estructura.

Pequeños detalles de todos los formatos de archivo, algunos de los cuales no estaban presentes en los formatos binarios, son compatibles con aplicaciones de funcionalidad mínima a través de valores en caché. Por ejemplo:

- Sin necesidad de implementar un paginador, una aplicación como un lector para invidentes puede permitir la navegación por la página utilizando los últimos saltos de página calculados (§4:2.3.3.13).
- Sin necesidad de implementación de fórmulas ni integración con una fuente de datos externa, un programa de hoja de cálculo puede operar a partir de cálculos en caché (§3:3.2.9) y datos externos en caché (§4:3.14 y §4:3.10.1.76).

Un documento conforme mínimo es muy sencillo; consulte la subsección “Documento WordprocessingML mínimo”.

4.4 DISEÑO COMPACTO

El formato de archivo OpenXML admite la creación de aplicaciones de alto rendimiento. En esta subsección se describen algunos de los aspectos de diseño que hacen posible que un archivo sea compacto, aumentando de este modo la velocidad de tratamiento y análisis. En la siguiente subsección se muestra cómo la estructura modular de archivo permite que la aplicación lleve a cabo numerosas tareas analizando o modificando solamente un pequeño subconjunto de un documento.

Tradicionalmente, un archivo OpenXML se almacena en un archivo ZIP por razones de empaquetado y compresión, según lo recomendado por las Convenciones de empaquetado abierto. Aunque parezca sorprendente, los archivos OpenXML son en media un 25% de menor tamaño, a veces incluso hasta un 75%, que sus equivalentes binarios. Por ejemplo, el tamaño de estas notas del producto es un 85% mayor en formato binario.

Una segunda razón que explica este diseño compacto, sobre todo cuando se requiere una representación sin comprimir, es la longitud de identificadores en XML. Los nombres de etiqueta usados frecuentemente son cortos. Igualmente, se recomienda a los implementadores que utilicen prefijos de espacio de nombres cortos. Por ejemplo, el prefijo convencional para el nombre de espacio de WordprocessingML es “w”.

Se logra un diseño aun más compacto evitando la repetición en todo el formato de archivo. Una clase de ejemplos elimina el almacenamiento redundante de objetos de gran tamaño.

- En SpreadsheetML, las cadenas repetidas se almacenan en una tabla de cadenas en el libro y se hace referencia a ellas en el índice (§3:3.3).
- En SpreadsheetML, una fórmula que llena una o varias celdas se almacena como una única fórmula “patrón” en la celda superior izquierda. El resto de celdas del intervalo de relleno hacen referencia a dicha fórmula mediante un índice de agrupación (§3:3.2.9.2).
- En DrawingML, los nombres de forma (§4:5.1.12.56), geometrías de texto (§4:5.1.12.76) y otros elementos preestablecidos (varios en §3:5.8, §3:5.9 y §4:5.1.12) se representan mediante un nombre o número, en lugar de hacerlo de manera explícita. En estos casos, los significados de los nombres y números residen en la especificación, y no en el archivo. Aquí, la representación elegida es el resultado de una decisión de compromiso explícita durante los procesos de estándares. Esto es compacto y permite realizar modificaciones en el nivel adecuado de abstracción. Por ejemplo, un rectángulo se puede cambiar a un óvalo mediante el cambio de un atributo (§4:5.1.11.18).

En otra clase de ejemplos, la jerarquía se utiliza para proporcionar semánticas de herencia. Una consecuencia positiva es que mejora el rendimiento mediante la reducción del tamaño del archivo.

- En WordprocessingML, los estilos son jerárquicos (§3:2.8.9).
- En DrawingML, las formas se agrupan de manera jerárquica (§4:5.1.2.1.20).
- En PresentationML, una jerarquía predeterminada hace referencia a patrones de diapositiva, diseños de diapositiva y diapositivas (§3:4.2).

Hay otros aspectos de OpenXML que también están diseñados para permitir una implementación eficaz. Por ejemplo, en SpreadsheetML, la tabla de celdas solamente almacena celdas no vacías y puede representar celdas combinadas como una unidad. Esta técnica representa una economía importante para hojas de cálculo dispersas.

4.5 MODULARIDAD

Una aplicación puede llevar a cabo numerosas tareas analizando o modificando solamente un pequeño subconjunto de un documento.

Hay tres características del formato OpenXML que contribuyen a proporcionar esta modularidad.

- Un documento no es monolítico, sino que se compone de varias partes.
- A su vez, las relaciones entre las partes se almacenan en partes.
- El formato de archivo ZIP que se utiliza habitualmente para la compatibilidad con documentos OpenXML admite el acceso aleatorio a cada parte.

Por ejemplo:

- Una aplicación puede mover una diapositiva limpiamente de una presentación a otra, junto con recursos tales como imágenes y diseños, sin que sea necesario en absoluto analizar el contenido de la diapositiva (§3:13.3.8). Este ejemplo utiliza datos llamados relaciones explícitas para encontrar la diapositiva y sus recursos. Las relaciones explícitas se definen mediante las Convenciones de empaquetado abierto y se pueden analizar sin necesidad de tener conocimientos de semánticas de etiquetas de PresentationML (§1:9.2, §2:8.3).
- Una aplicación puede eliminar todos los comentarios de un documento WordprocessingML sin necesidad de analizar su contenido (§1:11.3.2). Este ejemplo utiliza datos llamados relaciones implícitas para encontrar los comentarios. Las relaciones implícitas son específicas de OpenXML y por ello no exigen tener conocimientos del lenguaje de marcado correspondiente (§1:9.2).

4.6 MIGRACIÓN DE ALTA FIDELIDAD

OpenXML está diseñado para su compatibilidad con todas las características de los formatos binarios de Microsoft Office 97-2003.

No es exageración si afirmamos que lograr este objetivo es harto difícil y hay que subrayar el carácter único de OpenXML al conseguirlo. Algunos formatos, como PDF, están diseñados para suministrar un facsímil visual de documentos terminados a un usuario final. En contraste, OpenXML está pensado para permitir futuras modificaciones o manipulaciones al mismo nivel de abstracción que estaba disponible para el creador original. Por ejemplo, la reducción de un gráfico vectorial a un mapa de bits no respondería a esta intención, puesto que podría hacer que una jerarquía de estilo se desintegrara en estilos independientes. Además, un documento puede contener semánticas informáticas que el creador original desea conservar, como lógicas de fórmula que dependen de resultados de cálculo intermedio, incluyendo códigos de error o reglas de animación que generan un comportamiento dinámico.

Estas referencias a la especificación ejemplifican la capacidad de OpenXML para representar aspectos sutiles de los formatos binarios.

- La descripción de SpreadsheetML incluye una especificación de fórmula extensa (§4:3.17.7).
- La especificación de WordprocessingML documenta las reglas mediante las cuales las propiedades de párrafo, caracteres, numeración y tabla se componen con formato directo (§3:2.8, especialmente §3:2.8.10).
- La especificación de PresentationML documenta las características de animación (§3:4.4).

OpenXML permite distintas implementaciones para lograr la compatibilidad sin necesidad de tener que coincidir en los detalles más nimios. Este hecho es especialmente importante cuando están implicados cálculos numéricos, como diseño, representación de efectos y evaluación de fórmulas. Requerir más coherencia de lo que es factible crearía dificultades a los programadores a la hora de lograr la conformidad. Estas afirmaciones ponen de relieve el tipo de decisiones tomadas por el comité a este respecto.

- OpenXML define efectos como aspectos de superficie (§5.1.12.50) sin imponer restricciones al programador en cuanto a tener que buscar la coincidencia píxel a píxel.
- OpenXML define parámetros como márgenes de página (§4:2.6.11), fuente (§4:2.8) y justificación (§4:2.3.1.13). Esto permite a los programadores implementar distintos algoritmos de flujo siempre que respeten dichos parámetros.

- La especificación de fórmula de SpreadsheetML (§4:3.17.7) no intenta eliminar variaciones de cálculo de punto flotante porque hacerlo requeriría, en general, que las aplicaciones compatibles implementen emulación lenta en lugar de servirse de hardware nativo. En lugar de ello, especifica el número mínimo de bits de precisión para cálculos numéricos (§4:3.17.5).
- La especificación de fórmula SpreadsheetML también deja ciertas decisiones condicionadas a la implementación, para permitir de este modo posibles innovaciones futuras. Por ejemplo, no limita el número de veces que se puede repetir un cálculo como NORMINV (§4:3.17.7.227). (La función NORMINV efectúa una distribución inversa a la normal mediante la realización de una búsqueda iterativa.)

Algunas antiguas características, como VML (§3:6), se incluyen fundamentalmente por razones de compatibilidad con versiones anteriores. Al escribir nuevos documentos, se recomienda utilizar estándares más recientes ya en OpenXML, como DrawingML (§3:5).

4.7 INTEGRACIÓN CON DATOS ECONÓMICOS

OpenXML permite a las organizaciones integrar aplicaciones de productividad con sistemas de información que administran procesos empresariales mediante el uso de esquemas personalizados dentro de documentos OpenXML. El objetivo de una organización que adopte este enfoque sería reutilizar y automatizar el procesamiento de una información empresarial que de otro modo permanecería encerrada e impenetrable, dentro de documentos que las aplicaciones empresariales no pueden leer o escribir.

Las aplicaciones incluyen:

- *Búsqueda*: un usuario final puede buscar un conjunto de hojas de cálculo de empresas cuyos márgenes de beneficios superen el 20%.
- *Etiquetado de metadatos*: una empresa puede etiquetar presentaciones que se han aprobado desde una perspectiva normativa.
- *Ensamblado de documentos*: un grupo de propuestas puede optimizar la generación de propuestas a través de la automatización de la preparación de los datos base.
- *Reutilización de datos*: un ejecutivo de ventas puede generar un informe de todos los contratos de ventas en un determinado intervalo de fechas, listado de consumidores, volúmenes de transacciones y cualquier término y condición modificado.
- *Aplicaciones de línea de negocio*: los profesionales de una vertical especializada pueden preparar entregas en un entorno de creación familiar, y aún así que sus productos de trabajo crezcan automáticamente en los sistemas empresariales.

El cumplimiento de estos objetivos requiere definir la estructura y el tipo de datos que puede contener una clase de documentos, así como permitir que la información se muestre siempre que se produzca de manera natural dentro del flujo de cada documento. Pensemos en el simple ejemplo de un currículum vitae. Se podría definir una estructura de datos que incluya campos denominados nombre, número de teléfono, dirección, objetivos profesionales y cualificaciones. También se podría establecer después que estos campos se muestren siempre que una persona los introduzca en un documento. En otro ámbito económico, por ejemplo un grupo financiero o una entidad médica, la estructura y campos de datos serían diferentes.

OpenXML permite que estos procesos se produzcan de una manera estandarizada.

En primer lugar, la estructura de los datos económicos se expresa primero mediante un esquema XML personalizado. Esto permite a una organización expresar datos con etiquetas que son significativas desde una perspectiva empresarial. Una organización puede crear sus propios esquemas o utilizar esquemas estándar de la industria tales como XBRL para informes financieros (7) y HL7 para información sobre atención sanitaria (8). Los esquemas se están creando en el sector público, en el seno de corporaciones y, como estándares de la industria, con propósitos que van desde certificados de nacimiento hasta información sobre seguros. Se puede utilizar cualquier esquema personalizado, siempre que se exprese en forma de XSD (2).

En segundo lugar, los datos personalizados están incrustados en cualquier documento OpenXML en una parte de XML personalizado (§3.7.3) y se pueden describir utilizando una parte de propiedades de datos XML personalizados (§4:7.5). Separando estos datos personalizados de la presentación, OpenXML permite la integración de datos sin errores, al tiempo que hace posible la presentación y manipulación por parte del usuario final dentro de una amplia variedad de contextos, incluyendo documentos, formularios, diapositivas y hojas de cálculo. Así, se puede conseguir la interoperabilidad a un nivel más fundamental y preciso desde el punto de vista semántico.

4.8 ESPACIO PARA LA INNOVACIÓN

OpenXML está diseñado para impulsar entre los programadores la creación de nuevas aplicaciones que no eran posibles cuando se definieron los formatos binarios de Office, e incluso en los momentos iniciales de la definición de OpenXML.

Primero abordamos los mecanismos de extensibilidad que operan conjuntamente para posibilitar la interoperabilidad entre aplicaciones que cuentan con conjuntos de características distintos. Pensemos en una aplicación de *alto nivel* (una que contenga una nueva característica que no esté documentada en OpenXML) y una aplicación de *nivel bajo* (una que no entienda esta característica). Los tres objetivos principales de la extensibilidad son:

- *Fidelidad visual*: capacidad de la aplicación de nivel bajo para mostrar lo que mostrará la aplicación de nivel alto. De manera intrínseca, esto exige que el archivo almacene varias representaciones de los mismos datos.
- *Editabilidad*: capacidad para editar una o varias representaciones.
- *Privacidad*: capacidad para garantizar que las antiguas versiones de una representación no permanezcan después de haber editado otra representación, conservando de este modo información que el usuario consideraba borrada o modificada. Una aplicación puede lograr este objetivo mediante la eliminación o sincronización de representaciones.

Si un programador desea ampliar el conjunto de características de OpenXML, cuenta para ello con dos opciones principales:

- *Bloques de contenido alternativos*: un bloque de contenido alternativo (§3:2.18.4 y §5:9.2) almacena varias representaciones del mismo contenido, cada una dentro de su propio bloque de elección. Una aplicación de nivel bajo lee un bloque de elección que es capaz de leer. Al editar, escribe tantas elecciones de bloque como es capaz de escribir.
- *Listas de extensiones*: una lista de extensiones (§3:2.6) almacena XML personalizado sin representación visual.

Los programadores tienen posibilidades de innovar fuera de estos mecanismos de extensibilidad.

- *Paradigmas de interacción alternativa.* OpenXML especifica más que la sintaxis del documento pero menos que el comportamiento de la aplicación. Tal como se describe en la Declaración de conformidad, se centra en las semánticas (§1:2.2, §1.2.3). En consecuencia, una aplicación conforme es libre de comunicar con el usuario final a través de diversos medios o de no comunicar en absoluto con el usuario final, siempre que respete las semánticas especificadas.
- *Entornos informáticos nuevos.* La Declaración de conformidad admite aplicaciones con baja capacidad, para que éstas puedan ejecutarse en pequeños dispositivos, así como aplicaciones que implementan solamente un subconjunto de OpenXML (§1:2.6). El mecanismo de características adicionales permite que una aplicación productora comunique sus límites de capacidad (§3:8.1).

Tal como se indicaba en la sección anterior, algunas de las posibilidades más importantes para la innovación no implican la representación de documentos para la interacción con el usuario directo. En lugar de ello, hay un procesamiento entre máquinas que utiliza formatos de mensaje XML, por ejemplo a través de servicios Web XML (9). Aunque estas aplicaciones no tienen comportamientos visibles al usuario que no sean sus operaciones con datos contenidos en los documentos OpenXML, están sujetas a conformidad de documento (§1:2.4) y conformidad de aplicación (§1:2.5), las cuales son puramente sintácticas, y a directrices de interoperabilidad (§1:2.6), las cuales incorporan semánticas.

Aunque es imposible enumerar todos los casos posibles de uso del procesamiento de XML personalizado, se pueden anticipar servicios centrados en XML que procesan documentos OpenXML para la extracción e inserción de datos personalizados, servicios de seguridad personalizada como firma digital XML (10) o cifrado XML (11), incluso transformaciones XSLT arbitrarias (12) que convierten en y desde otros formatos XML. OpenXML no introduce prohibiciones o limitaciones en dichos procesos.

5 ESTRUCTURA DE UN DOCUMENTO OFFICE OPEN XML

Objetivo principal de estas notas del producto es permitir al lector seguir la estructura de alto nivel de cualquier archivo OpenXML. Para lograrlo, proporcionamos un nivel moderado de detalles sobre las Convenciones de empaquetado abierto (OPC) y menos detalles sobre los lenguajes de marcado individuales.

5.1 CONVENCIONES DE EMPAQUETADO ABIERTO

Las Convenciones de empaquetado abierto (OPC) proporcionan una manera de almacenar distintos tipos de contenido (por ejemplo, XML, imágenes y metadatos) en un contenedor, como un archivo ZIP, para representar de manera completa un documento. Describen un modelo lógico para representar contención y relaciones.

La implementación recomendada para las OPC utiliza el formato de archivo ZIP. Se puede examinar la estructura de un archivo OpenXML a través de cualquier visor de ZIP. Es útil examinar de este modo el contenido de un archivo OpenXML de pequeño tamaño al mismo tiempo que se lee esta descripción. En el sistema operativo Windows, solamente se tiene que agregar una extensión “.zip” al nombre del archivo y hacer doble clic.

Lógicamente, un documento OpenXML es un *paquete* OPC (§5:8). Un paquete es un conjunto plano de *partes* (§5:8.1). Cada parte tiene un *nombre de parte* que distingue mayúsculas de minúsculas y que consiste en una secuencia de nombres de segmentos delimitada por barras diagonales del siguiente modo “/pres/slides/slide 1 .xml” (§5:8.1.1). Cada parte tiene también un *tipo de contenido* (§5:8.1.2). Físicamente, el archivo ZIP es un paquete, cada elemento ZIP del archivo es una parte y las rutas de acceso dentro del archivo ZIP corresponden directamente a nombres de partes.

En la implementación de ZIP, “[Content_Types].xml” permite al usuario determinar el tipo de contenido de cada parte del paquete (§2:9.2.6). La sintaxis y definición de tipos de media se detalla en la sección 3.7 de RFC 2616 (13).

Los paquetes y partes pueden contener *relaciones explícitas* (§1:9.2) con otras partes dentro del paquete, al igual que con recursos externos. Cada relación explícita tiene un Id. de relación, que permite al contenido de una parte hacer referencia al mismo; y un tipo, que permite a una aplicación decidir como procesarlo. Los tipos de relaciones se nombran mediante URI, lo que permite a las entidades no coordinadas crear nuevos tipos de manera segura sin que surjan conflictos.

El conjunto de relaciones explícitas para un paquete o parte de origen dado se almacena en una *parte de relaciones*. La parte de relaciones de la totalidad de un paquete se llama “/_rels/.rels”; la parte de relaciones de una parte llamada “/a/b/c.xml” se llama “/a/b/_rels/c.xml.rels”. Las partes de relaciones (y, en la implementación de ZIP, la parte de tipo de contenido) son las únicas partes nombradas de manera especial en un paquete. Para abrir un paquete, una aplicación debe analizar el paquete y las partes de relaciones, y seguir las relaciones del tipo apropiado.

Las demás partes del documento OpenXML incluyen OpenXML, XML personalizado o contenido de tipo arbitrario como objetos multimedia. La capacidad de una parte para incluir XML personalizado es un mecanismo especialmente eficaz para incrustar datos y metadatos económicos.

5.2 WORDPROCESSINGML

Un documento WordprocessingML se compone de un conjunto de narraciones (§3:2.1). Cada narración es uno de los siguientes elementos: el documento principal (§3:2.2), el documento del glosario (§3:2.13), un subdocumento (§3:2.18.2), un encabezado (§3:2.11.1), un pie de página (§3:2.11.2), un comentario (§3:2.14.5), un marco, un cuadro de texto (§3:2.18.1), una nota al pie (§3:2.12.1) o una nota al final (§3:2.12.2).

La única narración necesaria es el documento principal. Es el destino de la relación del paquete cuyo tipo es:

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

Una ruta típica desde la raíz a la hoja en el árbol XML comprendería estos elementos XML (§3:2.2):

- `document`: el elemento raíz del documento principal (§3:2.3).
- `body`: cuerpo (§3:2.7.1). Puede contener varios párrafos. También puede contener propiedades de sección especificadas en un elemento `sectPr`.
- `p`: párrafo (§3:2.4.1). Puede contener uno o más segmentos. También puede contener propiedades de párrafo especificadas en un elemento `pPr`, que a su vez puede contener propiedades de segmento predeterminadas (a las que también se hace referencia como propiedades de carácter) especificadas en un elemento `rPr` (§3:2.4.4).

- `r`: segmento (§3:2.4.2). Puede contener varios tipos de contenido de segmento, fundamentalmente intervalos de texto. También puede contener propiedades de segmento (`rPr`). El segmento es un concepto fundamental en OpenXML. Un segmento es un fragmento contiguo de texto con propiedades idénticas; un segmento no contiene marcado de texto adicional. Por ejemplo, si una frase debe contener las palabras “esto es **tres** segmentos”, entonces estará representado por al menos tres segmentos: “esto es”, “**tres**” y “segmentos”. A este respecto, OpenXML difiere significativamente de los formatos que permiten el anidamiento arbitrario de propiedades, como HTML.
- `t`: intervalo de texto (§3:2.4.3.1). Contiene una cantidad arbitraria de texto sin formato, saltos de línea, tablas, gráficos u otro material no textual. El formato de texto es heredado de las propiedades de segmento y párrafo. Este elemento utiliza con frecuencia el atributo `xml:space="preserve"`.

En esta subsección, hemos tocado el formato directo de texto especificando propiedades de párrafo y segmento. El formato directo se encuentra al final de un orden de aplicación que también incluye estilos de tabla, párrafo y numeración, al igual que valores predeterminados de documento (§3:2.8.10). Estos estilos se organizan a su vez en jerarquías heredadas (§3:2.8.9).

La subsección “Documento WordprocessingML mínimo”, más adelante, muestra la totalidad de un documento WordprocessingML.

5.3 PRESENTATIONML

Un documento PresentationML se describe mediante una parte de presentación. La parte de presentación es el destino de la relación del paquete cuyo tipo es:

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

La presentación corresponde a estos constructos principales (§3:4.2), que enumeramos de arriba abajo en la jerarquía predeterminada:

- Patrones de diapositiva, patrones de notas y patrones de documento (§3:4.2.2), todos los cuales heredan propiedades a partir de la presentación.
- Diseños de diapositiva (§3:4.2.5), que heredan propiedades a partir del patrón de diapositiva.
- Páginas de notas (§3:4.2.4) y diapositivas (§3:4.2.3), que heredan propiedades a partir de patrones de notas y diseños de diapositivas respectivamente.

Cada patrón, diseño y diapositiva se almacena en su propia parte. El nombre de cada parte se especifica en la parte de relación de la parte de presentación. Cada una de las seis partes que no sea la presentación está estructurada básicamente de la misma forma. Una ruta típica desde la raíz a la hoja en el árbol XML comprendería estos elementos XML (§3:2.2):

- `sld`, `sldLayout`, `sldMaster`, `notes`, `notesMaster` o `handoutMaster`: el elemento raíz.
- `csld`: diapositiva (§4:4.4.1.15). Puede contener elementos DrawingML (tal como se describe en las dos siguientes viñetas) y otros elementos estructurales (tal como se describe a continuación).
- `spTree`: árbol de forma (§4:4.4.1.42). Puede contener propiedades de forma de grupo en un elemento `grpSpPr` (§4:4.4.1.20) y propiedades de forma de grupo no visuales en un elemento `nvGrpSpPr` (§4:4.4.1.28). Este nodo y sus descendientes son todos elementos DrawingML. Aquí mostramos algunos elementos DrawingML debido a su papel crucial en PresentationML.
- `sp`: forma (§4:4.4.1.40). Puede contener propiedades de forma en un elemento `spPr` (§4:4.4.1.41) y propiedades de forma no visuales en un elemento `nvGrpSpPr` (§4:4.4.1.31).

Además del contenido de forma DrawingML, un `slide` puede contener otros elementos estructurales, dependiendo del elemento raíz en el que reside, tal como se sintetiza en esta tabla:

	Diapositiva	Diseño de diapositiva	Patrón de diapositiva	Patrón de documento	Patrón de notas	Página de notas
Datos comunes	X	X	X	X	X	X
Transición	X	X	X			
Tiempo	X	X	X			
Encabezados y pies de página		X	X	X	X	
Nombre coincidente		X				
Tipo de diseño		X				
Conservar		X	X			
Lista de diseño			X			
Estilo de texto			X			

Las propiedades especificadas por objetos inferiores en la jerarquía predeterminada (patrón de diapositiva, diseño de diapositiva y diapositiva) sobrescriben las correspondientes propiedades especificadas por objetos superiores en la jerarquía. Por ejemplo, si no se especifica una transición para una diapositiva, entonces aquella se toma del diseño de la diapositiva; si no se especifica ahí, entonces se toma del patrón de diapositiva.

5.4 SPREADSHEETML

Un documento SpreadsheetML se describe en el nivel superior mediante una parte de libro. La parte del libro es el destino de la relación del paquete cuyo tipo es:

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

La parte del libro almacena información acerca del libro y su estructura, como la versión del archivo, la creación de aplicación y la contraseña que se va a modificar. Lógicamente, el libro contiene una o varias hojas (§3:3.2). Físicamente, cada hoja se almacena en su propia parte y se hace referencia a ella del modo habitual desde la parte del libro. Cada hoja puede ser una hoja de cálculo, una hoja de gráfico o una hoja de diálogo. Solamente trataremos la hoja de cálculo, que es el tipo más común. En de un objeto de hoja de cálculo, una ruta típica desde la raíz a la hoja en el árbol XML comprendería estos elementos XML:

- `worksheet`: el elemento raíz de una hoja de cálculo (§3:3.2).
- `sheetData`: la tabla de celdas, que representa las celdas no vacías de la hoja de cálculo (§3:2.4).
- `row`: una fila de celdas en la tabla de celdas (§3:2.8).
- `c`: una celda (§3:3.2.9). El atributo `r` indica la ubicación de la celda utilizando coordenadas de estilo A1. La celda puede tener también un identificador de estilo (atributo `s`) y un tipo de dato (atributo `t`).
- `v` y `f`: el valor (§3:3.2.9.1) y fórmula opcional (§3:3.2.9.2) de la celda. Si una celda tiene una fórmula, entonces el valor es el resultado del cálculo más reciente.

Ambas cadenas y fórmulas se almacenan en tablas compartidas (§3:3.3 y §3:3.2.9.2.1) para evitar el almacenamiento redundante y agilizar la carga y el guardado.

5.5 LENGUAJES DE MARCADO COMPATIBLES

Se pueden usar varios lenguajes de marcado para describir el contenido de un documento OpenXML.

- DrawingML (§3:5): utilizado para representar formas y otros objetos representados gráficamente dentro de un documento.
- VML (§3:6): un formato para gráficos vectoriales que se incluye para la compatibilidad con versiones anteriores y que será reemplazado por DrawingML.
- ML compartidos: Math (§3:7.1), Metadata (§3:7.2), Custom XML (§3:7.3) y Bibliography (§3:7.4).

5.6 DOCUMENTO WORDPROCESSINGML MÍNIMO

Esta subsección contiene un documento WordprocessingML mínimo que se compone de tres partes.

La parte de tipo de contenido “[Content_Types].xml” describe los tipos de contenido de las otras dos partes necesarias.

```
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
  <Default Extension="rels"
    ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
  <Default Extension="xml"
    ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml"/>
</Types>
```

La parte de relación del paquete “/_rels/.rels” describe la relación entre el paquete y la parte de documento principal.

```
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId1"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument"
    Target="document.xml"/>
</Relationships>
```

La parte de documento, en este caso “/document.xml”, contiene el contenido del documento.

```
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello, world.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

La especificación proporciona documentos mínimos y detalles adicionales de WordprocessingML (§1:11.2), PresentationML (§1:13.2) y SpreadsheetML (§1:12.2).

OpenXML es el resultado de un esfuerzo considerable realizado por representantes de muchas industrias e instituciones públicas con distintas procedencias e intereses organizativos. Engloba la totalidad del conjunto de características utilizadas en el corpus de documentos existentes, así como las necesidades de internacionalización propias de los grupos de idiomas más importantes de todo el mundo. Como resultado del trabajo de estandarización realizado por Ecma TC45 (1) y de las contribuciones realizadas a través de comentarios públicos, OpenXML ha hecho posible un alto nivel de interoperabilidad e independencia de plataforma. Actualmente, su documentación es completa (a través de un amplio material de referencia) y accesible a la vez (mediante descripciones no normativas). También incluye suficiente información para los productos de tecnología de asistencia para procesar documentos de manera adecuada. Las implementaciones de OpenXML pueden ser muy pequeñas y proporcionar funcionalidades muy específicas, o pueden abarcar la totalidad del conjunto de características. Los mecanismos de extensibilidad que incorpora el formato garantizan las posibilidades de innovación.

La estandarización de la especificación del formato y su mantenimiento a lo largo del tiempo garantiza que entidades de diferente naturaleza puedan confiar en él y tengan la seguridad de que su evolución posterior contará con las comprobaciones y controles propios de un proceso de estándar abierto. Existe la necesidad imperiosa de un estándar de formato de documento abierto capaz de conservar los millones de documentos creados en los formatos binarios preexistentes, y de los millones que se siguen creando cada año. Los avances tecnológicos en hardware, redes e infraestructura de software basada en estándares lo hacen posible. La tremenda diversificación de la demanda del mercado, incluyendo las cuantiosas inversiones actuales en sistemas empresariales críticos, lo convierte en esencial.

1. **Ecma International.** TC45 – Formatos Office Open XML. *Ecma International*. (En línea; en inglés) <http://www.ecma-international.org/memento/TC45.htm>.
2. **W3C.** Esquemas XML. *World Wide Web Consortium*. (En línea; en inglés) <http://www.w3.org/XML/Schema>.
3. **ISO.** ISO/IEC 19757-2:2003. *Organización Internacional de Normalización*. (En línea; en inglés) <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=37605&ICS1=35&ICS2=240&ICS3=30>.
4. **ISO.** ISO/IEC 19757-4:2006. *Organización Internacional de Normalización*. (En línea; en inglés) <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38615&ICS1=35&ICS2=240&ICS3=30>.
5. **W3C.** Lenguaje de marcado extensible (XML) 1.0. (Cuarta edición). *World Wide Web Consortium*. (En línea; en inglés) 2006. <http://www.w3.org/TR/2006/REC-xml-20060816/>.
6. **W3C.** Espacios de nombres de XML 1.0 (Segunda edición). *World Wide Web Consortium*. (En línea; en inglés) 2006. <http://www.w3.org/TR/2006/REC-xml-names-20060816/>.

7. **XBRL International.** Especificaciones XBRL. *Extensible Business Reporting Language*. (En línea; en inglés)
<http://www.xbrl.org/Specifications/>.
8. **Health Level Seven.** HL7 ANSI-Estándares aprobados. *Health Level Seven*. (En línea; en inglés)
http://www.hl7.org/about/directories.cfm?framepage=/documentcenter/public/faq/ansi_approved.htm.
9. **W3C.** Arquitectura de servicios Web W3C. *World Wide Web Consortium*. (En línea; en inglés) 2002. <http://www.w3.org/2002/ws/>.
10. **W3C.** Firma XML W3C. *World Wide Web Consortium*. (En línea; en inglés)
<http://www.w3.org/Signature/>.
11. **W3C.** Cifrado XML W3C. *World Wide Web Consortium*. (En línea; en inglés) 2001.
<http://www.w3.org/Encryption/2001/>.
12. **W3C.** XSL y XSLT. *World Wide Web Consortium*. (En línea; en inglés)
<http://www.w3.org/Style/XSL/>.
13. **W3C.** Protocolo de transferencia de hipertexto - HTTP/1.1. *World Wide Web Consortium*. (En línea; en inglés)
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>.