



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE CIENCIAS

XML en la Estandarización de Documentos

T E S I S

QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

LILIANA CRUZ CRUZ

DIRECTOR DE TESIS:

DRA. AMPARO LOPÉZ GAONA



2011



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno

Cruz

Cruz

Liliana

58444596

Universidad Nacional Autónoma de

México

Facultad de Ciencias

Ciencias de la Computación

301159636

2. Datos del tutor

Dra.

Amparo

López

Gaona

3. Datos del sinodal 1

M. en C.

María Guadalupe Elena

Ibargüengoitia

González

4. Datos del sinodal 2

Mat.

Salvador

López

Mendoza

5. Datos del sinodal 3

M. en I.

Gerardo

Avilés

Rosas

6. Datos del sinodal 4

M. en C.

Egar Arturo

García

Cárdenas

7. Datos del trabajo escrito

XML en la Estandarización de Documentos

130 p

2011

---

# Agradecimientos

*Este trabajo no habría sido posible sin el apoyo de mis padres, mis hermanas,  
mi asesora, mis sinodales y de toda aquellas personas que me rodean.*

---

# Índice general

Agradecimientos	I
Introducción	VII
<b>1. Metadatos</b>	<b>1</b>
1.1. ¿Qué son los metadatos? . . . . .	1
1.2. Utilidad y aplicaciones de los metadatos . . . . .	5
1.3. Tipo de Metadatos . . . . .	7
<b>2. Lenguaje de Marcado y Estructuración de la Información</b>	<b>10</b>
2.1. Lenguajes de marcado . . . . .	11
2.2. HyperText Markup Language (HTML) . . . . .	15
2.2.1. La evolución del HTML . . . . .	18
2.3. Extensible Markup Language (XML) . . . . .	21
2.4. Documentos bien formados y válidos . . . . .	26
2.4.1. Aplicaciones XML . . . . .	31
<b>3. Lenguajes para Textos Científicos</b>	<b>33</b>
3.1. Text Encoding Initiative (TEI) . . . . .	34
3.1.1. Estructura . . . . .	37
3.1.2. Funcionamiento de TEI . . . . .	41
3.1.3. Características del TEI . . . . .	41

---

3.1.4. Implementaciones del TEI . . . . .	42
3.2. North American Research Strategy for Tropospheric Ozone (NARS- TO) . . . . .	43
3.2.1. La plantilla de medidas atmosféricas de NARSTO . . . . .	46
<b>4. Modelado de la Solución Propuesta</b>	<b>51</b>
4.1. Descripción del problema . . . . .	52
4.2. Requerimientos . . . . .	54
4.2.1. Especificaciones de los Requerimientos . . . . .	54
4.3. Análisis . . . . .	55
4.3.1. Caso de Uso General . . . . .	55
4.3.2. Casos de uso . . . . .	56
4.4. Diseño . . . . .	67
4.4.1. Diagrama de Clases . . . . .	67
4.4.2. Diagrama de Secuencia . . . . .	69
4.4.3. Diagrama de Estados . . . . .	71
4.4.4. Diagrama de Bloques . . . . .	72
<b>5. Desarrollo de la Solución Propuesta</b>	<b>74</b>
5.1. Desarrollo . . . . .	85
<b>6. Conclusiones</b>	<b>93</b>
<b>A. Manual de Usuario</b>	<b>96</b>
A.1. Introducción . . . . .	97
A.2. Requerimiento Básico . . . . .	97
A.2.1. Instalación de la Plantilla . . . . .	97
A.3. Cómo utilizar la plantilla . . . . .	97
A.3.1. Datos de Identificación . . . . .	100
A.3.2. Información de la publicación . . . . .	102
A.3.3. Cobertura . . . . .	102

---

A.3.4. Información de Datos Espaciales, Acceso y Uso de la Información, Creación e Identidad . . . . .	103
A.3.5. Campos de Fechas . . . . .	104
A.4. Campos obligatorios opcionales . . . . .	106
<b>B. DTD</b>	<b>109</b>
<b>C. XMLSchema</b>	<b>112</b>
<b>D. Diseño de Pruebas</b>	<b>120</b>
D.1. Pruebas correspondientes a los Caso de Uso . . . . .	120

---

# Índice de figuras

2.1. Estructura básica de un documento HTML. . . . .	16
2.2. Ejemplo: etiquetas de HTML . . . . .	17
2.3. Anidación de etiquetas HTML. . . . .	17
2.4. Atributos en etiquetas HTML. . . . .	17
2.5. Ejemplo de atributos . . . . .	18
2.6. Estructura de un documento XML . . . . .	24
2.7. Estructura jerárquica de un documento XML . . . . .	25
2.8. DTD . . . . .	27
2.9. Esquema XML . . . . .	29
3.1. Estructura genérica de documentos TEI . . . . .	39
3.2. Estructura compuesta de documentos TEI . . . . .	39
3.3. Ejemplo de Archivo TEI . . . . .	40
3.4. NARSTO . . . . .	45
3.5. Plantilla de Excel NARSTO. Parte 1 . . . . .	47
3.6. Plantilla de Excel de NARSTO. Parte 2 . . . . .	49
3.7. Metadatos NARSTO . . . . .	50
3.8. Flujo de la creación de archivos DES . . . . .	50
4.1. Fases de la solución de un problema. . . . .	53
4.2. Caso de Uso General . . . . .	56
4.3. Caso de uso. Entrar . . . . .	57



---

4.4. Caso de Uso. Validar Fechas . . . . .	58
4.5. Caso de Uso. Validar Metadatos . . . . .	59
4.6. Caso de uso. Crear Documentos . . . . .	61
4.7. Crear Documento XML . . . . .	62
4.8. Caso de uso. Crear Documento Excel . . . . .	64
4.9. Caso de Uso. Salir . . . . .	66
4.10. Diagrama de clases de la aplicación. . . . .	69
4.11. Diagrama de Secuencia . . . . .	70
4.12. Diagrama de Estados. . . . .	71
4.13. Diagrama de Bloques . . . . .	73
5.1. Campos - Valor . . . . .	86
5.2. Elementos de Control . . . . .	87
A.1. P-Metadatos2003 . . . . .	98
A.2. Comentarios de Ayuda . . . . .	99
A.3. Exportar XML . . . . .	99
A.4. Mensaje de Advertencia . . . . .	100
A.5. Datos de Identificación . . . . .	101
A.6. Información de la Publicación . . . . .	102
A.7. Cobertura . . . . .	103
A.8. Información de Datos Especiales, Acceso y Uso de la Informa- ción, Creación e Identidad . . . . .	103
A.9. Calendario . . . . .	104
A.10. Validación de Fechas . . . . .	105
A.11. Datos Capturados . . . . .	106
A.12. Exportando documento XML . . . . .	107
A.13. Documento XML generado . . . . .	108

---

# Introducción

El intercambio de información se ha llevado a cabo desde la antigüedad ya que permite comunicar, aprender, e informar principalmente. Pero a su vez genera pérdida de información ya que ésta no siempre llega como se envió, como en el juego del teléfono descompuesto.

El intercambio de información no siempre se ha llevado a cabo con multitud de personas pues antes sólo algunas personas la practicaban. Lo que implicaba que la pérdida de información fuera mínima.

Hoy en día las cosas se han transformado gracias al uso de las tecnologías de la información, ahora casi el 100 % del mundo busca e intercambia información, mediante Internet. Y por ende la pérdida de información sigue estando presente. Esto se debe a que la información que se encuentra en la red es muy variada con respecto a su contenido, y con respecto al formato en que ésta se encuentra. Esta última, la que guarda información en distintos formatos es la que genera la pérdida de información, pues para poder visualizar la información es necesario tener el programa adecuado para el tipo de formato del archivo que este tenga.

Dado que existen multitud de formatos para el intercambio de información, y puesto que no se pueden abarcar todos, en esta investigación sólo se tratará el intercambio de textos científicos, en particular se hablará de TEI y NARSTO.

El problema general que se intenta resolver en esta tesis es precisamente el

intercambio de información de datos científicos en especial los de tipo geográficos (los cuales se obtendrán de TEI y NARSTO). De manera que la información a intercambiar sea sólo la necesaria y suficiente, con el fin de evitar pérdidas de información. Aquí es donde participan los metadatos y XML.

En el presente trabajo se analizan los aspectos más relevantes para redefinir y adaptar la plantilla de NARSTO a una aplicación mas sencilla que permita capturar información y que además posibilite la la construcción de archivos XML con la información capturada, para tener un mejor intercambio de ésta entre las distintas dependencias universitarias.

Los metadatos ayudan a seleccionar el conjunto de datos necesarios a intercambiar, y XML es un lenguaje similar a HTML en su estructura, ya que ambos se definen mediante un conjunto de etiquetas. La diferencia es que HTML tiene definido su propio conjunto y este conjunto está orientado a la presentación de la información contenida en sus etiquetas. Mientras que XML permite definir etiquetas respecto al contenido que almacenarán y también permite definir la estructura que estas etiquetas llevarán.

XML se ha definido como un lenguaje estándar para el intercambio de información estructurada, ya que éste es compatible con muchas aplicaciones, como bases de datos, hojas de cálculo, etc., y por su compatibilidad con tantos programas, trata de impedir que haya pérdida de información.

El producto que se generará en esta investigación, será una aplicación para facilitar el intercambio de información entre algunas dependencias universitarias. La aplicación deberá permitir capturar y validar los datos a intercambiar. El conjunto de datos capturados se podrán exportar a un documento con formato XML que sea válido y bien formado.

Este trabajo presenta los siguientes capítulos:

En el primer capítulo se da un esbozo teórico sobre los metadatos, cómo surgieron, para qué sirven, algunas de sus aplicaciones, y utilidades, etc.

En el segundo capítulo se analizan los lenguajes de marcado como son HTML y XML, lo que estos lenguajes pueden ofrecer y su comportamiento.

En el tercer capítulo se presentan los lenguajes de texto científicos, en donde se plantearan en particular los lenguajes de TEI y NARSTO.

En el cuarto capítulo se muestra el análisis y diseño de la solución al problema, donde se plantean los requerimientos, los casos de uso, el diagrama de secuencias, diagrama de bloques del problema planteado, y de manera resumida la función de las herramientas de análisis.

Y por último el quinto capítulo presenta un esbozo de las herramientas que se utilizarón para la implementación de la solución. Y de manera general el proceso que se siguió para desarrollar la aplicación.

Finalmente se exponen las conclusiones que se obtuvieron al desarrollar el producto, la importancia de XML y de los metadatos sobre el trabajo elaborado.

---

# Capítulo 1

## Metadatos

Suponer que se tiene un conjunto de alimentos enlatados, pero que no se encuentran etiquetados ¿cómo saber que contienen estas latas?. Si se tienen dos o tres latas, lo que se podría hacer es abrirlas y ver su contenido, ¿pero qué pasa cuando hay más de cien? el hecho de abrir cien latas ya no sería una solución eficiente. Una posible solución a este problema sería colocar una etiqueta a cada lata que aporte la información de su contenido, ¿y cuál es la información que necesita esta etiqueta?.

El hecho de poder describir el contenido de algún elemento, ya sea un producto, un libro, un coche, etc., con un conjunto mínimo de palabras y que tal conjunto proporcione la información necesaria y suficiente que describa el elemento en cuestión, es a lo que se le conoce como metadatos.

### 1.1. ¿Qué son los metadatos?

Existe una gran variedad de definiciones sobre el concepto de metadatos, el primer significado que se encuentra a partir de su etimología es “más allá de los datos”; sin embargo la primera definición que se le dio fue la de “datos acerca

de datos”, para comprender esta frase, sólo hay que observar que los primeros datos describen a los segundos.

Es decir, los metadatos, son en general un conjunto de atributos que describen y proporcionan la información suficiente de un objeto<sup>1</sup>.

Al parecer se cree que el uso de metadatos es reciente, pues junto con el inicio de la era de la información digital, este concepto sale a la luz, sin embargo esto no es del todo cierto ya que sus orígenes vienen de siglos atrás.

Esto comienza con la necesidad de organizar libros, para poder resolver esta cuestión, los bibliotecarios crean fichas bibliográficas; ya que éstas describen con palabras clave la información de un libro. Por lo cual los metadatos toman la forma de catálogos de libros, catálogos en tarjetas y en la actualidad catálogos en línea.

Las bases de los primeros metadatos (digitales) se desarrollan a finales del siglo XX, cuando emergen múltiples estándares de codificación, lenguajes y protocolos que se utilizan en la generación y uso de catálogos. Pero ¿en qué consiste realmente el concepto de metadato?

Sus orígenes se remontan a la década de los sesenta, según [FOLD2010] cuando Jack Myers lo empleó para describir conjuntos de datos. Actualmente la difusión más extendida que se le dio fue la de dato sobre dato, pues proporciona la información mínima necesaria para identificar un recurso.

[BERN1999] da el significado de meta como: “prefijo que indica que algo se aplica a sí mismo; por ejemplo, una metareunión es una reunión hablando de reuniones”. De forma análoga podemos decir que los metadatos son datos que ofrecen información de otros datos.

También se encuentra la siguiente definición de metadato como “Datos sobre los datos”. En el procesamiento de datos, los metadatos son datos que

---

<sup>1</sup>Todo lo que puede ser materia de conocimiento o sensibilidad de parte del sujeto, incluso este mismo.

proveen información o documentación sobre otros datos gestionados dentro de una aplicación o contexto determinado.

Los metadatos pueden incluir información descriptiva sobre el contexto, calidad y condición o características de los datos. Por ejemplo, los metadatos aportarán información sobre elementos de datos o atributos (nombre, tamaño, tipo de datos, etc.) e información sobre los registros o la estructura de datos (extensión, cómo están asociados, a quién pertenecen, etc.) [FOLD2010].

Por otra parte, si partimos del concepto de “metalenguaje” entendido, como “lenguaje que se usa para hablar del lenguaje” según el [DRAE2001], de forma paralela, podríamos decir que los metadatos son datos cuando se usan para hablar de ellos mismos, y de ahí que, de manera generalizada y válida en cualquier contexto, se definan los metadatos como datos sobre los datos o datos acerca de datos. [BERN1999]

Inciendo sobre el aspecto básico de la definición, otros autores amplían el concepto de “dato sobre el dato”, ya que afirman que metadato incluye información sobre su contexto, contenido y control, así como todo lo que tenga que ver con “el dato”.

Los metadatos tienen varias funciones, algunas de las cuales se describen a continuación:

- Resumir el significado de los datos.
- Describir el contenido de elementos.
- Permitir la búsqueda de datos.
- Determinar si los datos son los necesarios.
- Recuperar y usar copias de los datos.
- Mostrar instrucciones sobre cómo interpretar datos.

- Aportar información acerca de la vida de los datos.
- Indicar las relaciones con otros recursos.

De todo lo planteado hasta ahora se pueden percibir varios puntos esenciales (datos sobre datos, conjunto de datos, recuperación de información, más allá de los datos) los cuales resultan útiles para la elaboración de una nueva definición que asocie todas las definiciones manifestadas hasta la fecha, de tal forma que resulte posible inferir que metadato es toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato o elemento, y su finalidad es la de facilitar su recuperación, accesibilidad, la preservación y el intercambio en sistemas heterogéneos, entre otros.

De esta forma, son ejemplos de metadatos:

- El encabezado de un archivo multimedia(imagen, video o audio).
- Encabezados de mensajes de correo electrónico.
- El diccionario de una base de datos.
- Catálogos de bibliotecas.
- Etiquetas de productos enlatados.
- Fichas bibliográficas.
- Sinopsis de películas.

Se pueden seguir mencionando más ejemplos, con tan solo observar a nuestro alrededor, seguramente se encontrarán muchos de ellos.



## 1.2. Utilidad y aplicaciones de los metadatos

Una primera utilidad que se le da a los metadatos es describir recursos, como la catalogación de libros. La función principal de los metadatos es describir información de distintos tipos, independientemente del área en donde sean solicitados.

Los metadatos posibilitan el intercambio de información de recursos, lo que facilita búsquedas sobre datos distribuidos. Permitiendo el acceso a éstos de manera controlada. También sirven para preservar los recursos, y de esta manera en un futuro poder migrarlos a nuevas tecnologías de información.

En Internet los metadatos son una fuente esencial de búsqueda e integración de conocimiento, puesto que permite el intercambio de información de un gran número de fuentes heterogéneas. Asimismo se espera optimizar el acceso y recuperación de la información.

Por lo tanto los metadatos contribuyen en el intercambio, la identificación, organización, ordenación, búsqueda y localización de los distintos tipos de recursos de información ya sean analógicos o digitales. De igual forma “los metadatos presuponen procedimientos, normas y especificaciones transparentes y consecuentes” [DEJO2001].

Los metadatos igualmente pueden definir su utilidad en función del sistema y del usuario. A nivel sistema, los metadatos pueden utilizarse para facilitar la interoperabilidad y la posibilidad de compartir datos entre las distintas herramientas de búsqueda, o incluso para facilitar una búsqueda híbrida<sup>2</sup> e integra, tanto en recursos de Internet, como en materiales impresos previamente representados, por ejemplo en MARC(Machine Readable Cataloging) o en otro formato legible por máquina.

---

<sup>2</sup>Una búsqueda híbrida, combina la búsqueda de texto con la búsqueda semántica, para mejorar precisión y exhaustividad.

Como usuario final, los metadatos deberían facilitar detalles sobre qué información está accesible, dónde, cómo y en qué condiciones se puede llegar a ella.

“Los metadatos y la formulación de estándares dependen del área o disciplina en la que son utilizados; muchas de estas las encontramos en Internet: datos médicos, información bibliotecaria, juegos, comercio electrónico y catálogos audiovisuales” [DEJO2001]. La función principal de los metadatos es la recuperación de información. Otra aplicación importante es la revisión de los recursos digitales, tanto para verificar el acceso, como para calificar el contenido o para describir los derechos del documento.

De manera más sistemática, siguiendo a [LAGO1996] y otros investigadores, las principales aplicaciones de los metadatos son: catalogación (documentos, artículos y colecciones), recuperación de información de Internet, comercio electrónico, agentes de software inteligentes, firmas digitales, valoración y clasificación del contenido, derechos de propiedad intelectual y política de preservación y privacidad del contenido Web.

Los metadatos ofrecen la posibilidad de trabajar a diferentes niveles con los recursos digitales, facilitando entre otras cosas:

- El incremento de la accesibilidad: la existencia de un conjunto de metadatos que describa correctamente uno o varios objetos aumenta la posibilidad de acceder a ellos. Por otro lado, los metadatos hacen posible la búsqueda de información en múltiples colecciones a la vez.
- La expansión del uso de la información: ya que facilita la difusión de versiones digitales de un único objeto.
- El control de las diferentes versiones de un objeto: no sólo en lo que se refiere a gestionar la vida de un objeto, sino también en lo que tiene que ver con su difusión, es decir, generar metadatos con distintas cantidades

de información sobre un mismo objeto con el fin de distribuirla a un público heterogéneo.

- La gestión de aspectos legales: los metadatos permiten establecer claramente las restricciones de explotación, informar sobre los derechos de autor, control del uso de todo, o alguna parte del objeto, método de pago por su disfrute, controlar el acceso a información restringida, etc.
- La preservación del objeto original.

### 1.3. Tipo de Metadatos

Existen distintos tipos de metadatos dado que existen diversos tipos de usuarios, con necesidades distintas y diversos tipos de información con finalidades también distintas.

Debido a que se encuentran distintas áreas en las cuales utilizar metadatos, estos se pueden clasificar respecto a su tipo de aplicación. Las siguientes aplicaciones son un ejemplo específico del uso de metadatos:

- Dublin Core<sup>3</sup> (DCMI) Para describir recursos de información Web asociados al concepto de biblioteca digital.
- MARC<sup>4</sup> (Machine Readable Cataloging) Define elementos de los registros catalográficos en bibliotecas.
- CIMI<sup>5</sup> (Consortium for the Interchange Museum Information) Asociados a la información de museos, y por tanto a una información eminentemente iconográfica.

---

<sup>3</sup> *Dublin Core Metadata Initiative* [en línea]. < <http://dublincore.org/> >

<sup>4</sup> *MARC STANDARDS* [en línea]. < <http://www.loc.gov/marc/> >

<sup>5</sup> <http://www.cimi.org/> (El Consorcio cerró sus operaciones en 2003).

- CSDGM<sup>6</sup> (Content Standard for Digital Geospatial Metadata) Del FGDC<sup>7</sup> (Federal Data Geographic Committee). Constituyen normas para la descripción de recursos geográficos o geoespaciales.
- GILS<sup>8</sup> (Government Information Locator Service) Para describir recursos de información electrónica gubernamental.
- EAD<sup>9</sup> (Encoded Archival Description) Para la descripción archivista, como alternativa a la descripción tradicional realizada en los archivos. Proporcionar información descriptiva en los procesos de creación de registros electrónicos. Esta aplicación es un modelo de metadatos basado en SGML/XML que representa una forma estructurada de crear instrumentos de descripción para la búsqueda digital en colecciones constituidas por materiales de archivo.
- MEMRI (Media Error Monitoring and Reporting Information) Es un sistema de metadatos específicos en el área de desarrollo de dispositivos inteligentes para el almacenamiento de datos, destinados a proporcionar datos sobre la integridad de la información almacenada en este tipo de soportes inteligentes.

Otro tipo de metadatos se pueden definir por el formato existente de dominio público, así es como [GILL1998] define los siguientes tipos de metadatos:

- Administrativos: utilizados para la gestión y administración de los recursos de información.
- Descriptivos: para la recuperación y representación de recursos de información.

---

<sup>6</sup> *Content Standard for Digital Geospatial Metadata* [en línea]. < <http://www.fgdc.gov/metadata/csdgm/> >

<sup>7</sup> *The Federal Geographic Data Committee* [en línea].

< <http://www.fgdc.gov/> >

<sup>8</sup> *Global Information Locator Service* [en línea]. < <http://www.gils.net/> >

<sup>9</sup> *Encoded Archival Description* [en línea]. < <http://www.loc.gov/ead/> >

- **Preservativos:** destinados a gestionar la preservación de las fuentes de información.
- **Técnicos:** relacionados al funcionamiento o comportamiento de un sistema.
- **De uso:** relacionados al nivel y tipo de utilización de los recursos de información.

Por otra parte también se pueden separar los metadatos como extrínsecos e intrínsecos. Los metadatos extrínsecos se definen por la forma en que estos están relacionados con la información del recurso, es decir que existe un enlace persistente entre el registro del metadato y el recurso que describe, enlace que existe separadamente del recurso. Los metadatos intrínsecos, son los que están sincronizados al recurso que describen, es decir, que si algún atributo del recurso cambia, el metadato que identifica y describe a este atributo también debe cambiar.

---

## Capítulo 2

# Lenguaje de Mercado y Estructuración de la Información

La necesidad de intercambiar información de una forma fácil y sencilla, es una de las principales causas que ha llevado a transformar la información analógica en digital, el problema que surge no es tanto el digitalizar la información, sino poder visualizar dicha información en distintas computadoras.

Otro inconveniente que surge de intercambiar información es el de estructurar los recursos de una forma lógica y estándar, que los elementos que componen un documento vayan separados de manera que éstos tengan un orden, y se puedan distinguir fácilmente para poder recuperar la información, en cualquier tipo de recurso.

Esto ha llevado a expertos e interesados en este ámbito a investigar la manera de solventar dicha necesidad. Lo cual han logrado profundizando en el tema de estructuración de la información, y han conseguido desarrollar lengua-

jes de marcado que pasan por normas de estandarización, para que el proceso e intercambio de información estén respaldados por dichas normas.

## **2.1. Lenguajes de marcado**

El lenguaje de marcado es un conjunto de etiquetas que indican la estructura y presentación de la información. Este tipo de lenguaje surge ante la necesidad de estandarizar formatos de distintos documentos, por ejemplo cuando se envía un archivo de texto que no se puede visualizar con cualquier programa, necesita de un programa específico que pueda interpretar el contenido del archivo y mostrar la información, de tal forma que se pueda recuperar la información necesaria con la que se desea trabajar.

Este tipo de problemas, que consistían en que cada aplicación utilizaba sus propias etiquetas para codificar los archivos de texto, se trata de resolver cuando IBM en los años 60, propone un sistema de etiquetas para identificar elementos lógicos como títulos, direcciones, páginas, párrafos, listas, etc., con algún tipo de etiqueta dentro del mismo documento. Así estas etiquetas, le dan una estructura lógica a los documentos. De esta forma IBM desarrolla un lenguaje en el que combina texto e instrucciones de formato. A este tipo de instrucciones es lo que IBM llama “Lenguaje de Marcas Generalizado o GML (Generalized Markup Language)”.

De esta forma conociendo el sistema de GML y conociendo el sistema de marcado de las distintas aplicaciones, el intercambio de información de un sistema a otro sería posible sin necesidad de perder el formato y estructura indicada del recurso de información.

GML no se aplicó mucho tiempo, pues éste pasó a formar parte de la ISO (International Organization for Standardization). La cual definió un lenguaje de marcado para la creación de documentos estandarizados, basado en GML,

al cual llamó “Lenguaje de Marcas Generalizado Estándar o SGML (Standard Generalized Markup Language)”.

Los Lenguajes de Marcado (Markup Language), conservan en su denominación el término Markup, que proviene del mundo de la edición, donde la expresión *mark up* es usada para indicar las anotaciones que autores, editores o correctores de pruebas de imprenta o de manuscritos, suelen adjuntar a los textos, para señalar la correcciones o los tratamientos editoriales. Éstas se presentan a través de unas marcas (tags) que constituyen un conjunto de instrucciones, que sirven para describir la estructura y la composición de un documento y que además indican al programa de software cómo debe tratar su contenido[VIAN2004].

[JOHN1998] define el Lenguaje de Marcado como el conjunto de reglas que establecen qué tipo de marcas han de ser utilizadas, de qué modo se distinguirán las marcas del texto del documento, cómo se insertarán éstas (la gramática y su sintaxis), y cuáles son las marcas permitidas en cada una de las partes del texto.

Estos lenguajes son independientes del sistema que genera el documento y describen la finalidad del documento más que su aspecto. Al ser independientes de las reglas de formateo, simplifican al autor la creación de documentos, ya que solamente se deberá preocupar de la organización de las partes y no de su presentación. Para pasar un documento a un nuevo formato solamente se deberán recodificar los parámetros externos, nunca el documento en sí, siendo consecuencia directa de esto su mayor flexibilidad de uso.

La realización de los documentos que adoptan estos lenguajes se suele efectuar en dos etapas:

- Identificación de los elementos posibles según el tipo de documentos dados y sus posibles combinaciones.



- Delimitación en un documento relevante de estas categorías, respetando las reglas editadas para su combinación.

El ejemplo más ilustre de este tipo de lenguajes es el Standard Generalized Markup Language (SGML). El SGML nace como una norma para la representación lógica del contenido de un texto que permite la eventual inclusión de objetos gráficos y está dirigido sobre todo al mundo de la edición. Este lenguaje es en realidad un metalenguaje que prescribe unas reglas sintácticas muy específicas para definir un conjunto de marcadores y de relaciones entre los marcadores.

Es un lenguaje no propietario que al no depender de un programa o de una arquitectura particular permite transportar los datos de un programa a otro o de una plataforma a otra sin perder información. Su sistema de marcas es legible, y está diseñado con texto sencillo, colocado al lado de los elementos textuales a los cuales hace referencia y puede ser leído tanto por una máquina como por una persona.

Un lenguaje de marcado que adopta las especificaciones SGML se define como una aplicación SGML. Una aplicación SGML, describe a su vez, la estructura lógica de una clase de documentos y no su forma física. Esta estructura abstracta se especifica definiendo los elementos que la constituyen (capítulo, título, párrafo, nota, etc.) y las relaciones que existen entre ellos.

Los documentos SGML están formados por los siguientes componentes:

- Elementos. Son partes del documento dotados de un sentido propio (título, autor, etc.). Están caracterizados por una etiqueta inicial, un contenido y una etiqueta final.
- Atributos. Son metainformaciones, informaciones añadidas acerca del elemento, y no pertenecen en sentido estricto al contenido. Están contenidos en la etiqueta inicial del elemento.

- Entidades. El documento SGML se puede distribuir en diferentes archivos. Un archivo puede contener la portada, otro la introducción, otro un gráfico, otro la bibliografía, etc. Estos archivos reciben el nombre de entidades. Estos archivos se invocarán dentro del documento SGML para ser utilizados.
- Texto. Es el contenido propiamente dicho del documento y está compuesto por las palabras, los espacios y la puntuación.
- Comentarios. Son anotaciones que permiten evitar olvidos o facilitar la transmisión de información entre autores, no son parte del contenido del documento y las aplicaciones SGML los ignoran.

Asimismo, los documentos SGML, siempre se componen de tres partes:

- Declaración SGML. Contiene las instrucciones iniciales de las aplicaciones SGML. Permite especificar los valores fundamentales.
- Document Type Definition (DTD) es la verdadera gramática de los documentos SGML que, enumerando los elementos admitidos, definiendo su contexto de utilización y otros eventuales vínculos estructurales, define las reglas que permiten verificar si los documentos son estructuralmente correctos.
- Instancia del documento. Aquella parte del documento que contiene el verdadero texto.

Ante la escasez de organización y recuperación de información electrónica, los lenguajes de marcado se han convertido en un nuevo alfabeto semántico para los productores y gestores de información. El lenguaje de marcado es un sistema complejo de descripción de información, un formato sencillo y universal de documentos electrónicos, de tal manera que represente su significado y permita, así, controlar su procesamiento.

## 2.2. HiperText Markup Language (HTML)

La Web es un sistema de acceso a documentos que se encuentran en Internet, ésta a su vez define los lenguajes empleados para poder enviar los datos que circulan entre servidor y cliente. Los documentos que se encuentran en la Web son documentos de hipertexto, que especifican la codificación de los objetos digitales que constituyen un documento y la manera de representar los enlaces.

El hipertexto se ha definido como una manera de manejar y organizar información. Estos hipertextos utilizan enlaces que son textos en pantalla que pueden conducirnos a otro texto relacionado. Esta tecnología crea una base de información de diversos contenidos, que se conectan entre sí mediante enlaces, de cuya selección se obtiene información.

Actualmente los hipertextos, ya no sólo se limitan a tener enlaces que se representen con texto, sino que ya se pueden encontrar con imágenes, sonido y video. Los programas que leen este tipo de textos son los famosos navegadores de Internet.

HTML es el acrónimo de “HyperText Markup Language” (Lenguaje de Marcado de Hipertexto) , se basa en la idea de hipertextos, es un lenguaje para representación de documentos en formato electrónico que pertenece a la clase de los lenguajes de marcado y tienen su origen en el SGML.

HTML es el lenguaje que mayor difusión ha alcanzado. Fue ideado por Tem Berners Lee[BERN1999], para poder consultar, a través de una interfaz específica, los elementos que figuran en la World Wide Web, pasando de un documento a otro gracias a la aplicación de los lenguajes hipertextuales[VIAN2004].

Su rápido y asombroso éxito, se debe a su facilidad, además de permitir la descripción de documentos textuales ha ido incorporando extensiones para el tratamiento de información multimedia.

El objetivo que tiene HTML es el de etiquetar documentos escritos para darles un formato de presentación y crear enlaces para poder relacionar estos documentos con otros. Las etiquetas definidas en HTML indican al navegador como visualizar los enlaces, párrafos, el tipo de letra, el color de letra, color de fondo, si tiene imágenes, alineación del texto o imagen, etc. Las etiquetas definidas se incluyen dentro del texto del documento, pero estas no son mostradas por el navegador, sólo indican como mostrar su contenido.

Un documento HTML está compuesto, como todas las aplicaciones SGML, por elementos, atributos, entidades, texto y comentarios. Se caracteriza por poseer un elemento raíz, que proporciona la información relativa a la versión de HTML utilizada, y una macroestructura articulada en dos partes: el encabezado (head) y el cuerpo (body). Además el encabezado y el cuerpo, pueden contener otras etiquetas.

La estructura básica de un archivo en formato HTML se puede apreciar en la Figura 2.1.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Página de ejemplo</title>
  </head>
  <body>
    <h1>Página de ejemplo</h1>
    <p>Este es un
      <a href="demo.html">ejemplo</a> básico.
    </p>
    contenido de la página: texto, imágenes, enlaces, ...
    <!-- este es un comentario -->
  </body>
</html>
```

Figura 2.1: Estructura básica de un documento HTML.

Documentos HTML constan de un árbol de elementos y texto. Cada elemento se representa por una etiqueta de inicio, como “<body>”, y una etiqueta final, como “</body>”, ciertas etiquetas de inicio y fin en determinados casos, se omiten Figura 2.2

```
Etiqueta cerrada
<title> Mi página Web </title>
Etiqueta abierta
<br>
Etiqueta con parámetros
<body bgcolor="#FFFFFF"> </body>
```

Figura 2.2: Ejemplo: etiquetas de HTML

Las etiquetas tienen que ser anidadas de manera que los elementos estén completamente dentro de sí Figura 2.3.

```
<p> <em> muy mal <strong> </ em> </ strong> </ p>
<p> <em> <strong> es correcta </ strong>. </ em> </ p>
```

Figura 2.3: Anidación de etiquetas HTML.

HTML define un conjunto de elementos que se pueden utilizar, junto con las reglas sobre la forma en que los elementos se pueden anidar para crear documentos de éste tipo.

Los elementos pueden tener atributos, que controlan el contenido de los elementos. En el siguiente ejemplo Figura 2.4, hay un hipervínculo, formado mediante el elemento **a** y su atributo **href**:

```
<a href="demo.html"> ejemplo con atributos </ a>
```

Figura 2.4: Atributos en etiquetas HTML.

Los atributos se colocan dentro de la etiqueta de inicio, y constan de un nombre y un valor, separados por un carácter “=”. El valor del atributo puede permanecer sin comillas si no contiene espacios o cualquiera de “” = < o >. De lo contrario, tiene que ser citado con comillas simples o dobles. El valor, junto con el carácter “=”, puede omitirse por completo si el valor es la cadena vacía.

```
<!-- atributos vacíos -->
<input name=direccion disabled>
<input name=direccion disabled="">

<!-- atributos con valor -->
<input name=direccion maxlength=200>
<input name=direccion maxlength='200'>
<input name=direccion maxlength="200">
```

Figura 2.5: Ejemplo de atributos

Los navegadores Web analizan este marcado, convirtiéndolo en un árbol DOM (Document Object Model).<sup>1</sup>

Los archivos que contienen documentos HTML suelen tener la extensión .html o .htm.

### 2.2.1. La evolución del HTML

La idea de [BERN1999] surgió por 1989 y en los dos años posteriores definió el HTML como un subconjunto del SGML. En sus primeras versiones(0.9 y 1.0 de 1992), el HTML no llega a ser una verdadera aplicación SGML.

La auténtica formalización en una DTD SGML se produce con la versión 2.0 de 1994, primera versión oficial, con la cual se expresa la voluntad de que las etiquetas sean capaces de marcar la información de acuerdo a su significado, sin dar mucha importancia a la resolución en pantalla [VIAN2004]. Esta primera versión solamente marcaba encabezados, listas y enlaces, lo que sería suficiente para crear la World Wide Web.

Ese mismo año (1994) se crea el World Wide Web Consortium (W3C) fundado por el MIT de Estados Unidos, el INRIA francés y la Keyo University de Japón, se define como un *forum* independiente para la discusión de los problemas que conciernen a la World Wide Web. El objetivo del W3C es guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web. El valor social que aporta la Web,

---

<sup>1</sup>Un árbol DOM es una representación en memoria de un documento.

es que ésta hace posible la comunicación humana, el comercio y las oportunidades para compartir conocimiento. Uno de los objetivos principales del W3C es hacer que estos beneficios estén disponibles para todo el mundo, independientemente del hardware, software, infraestructura de red, idioma, cultura, localización geográfica, o habilidad física o mental [W3C].

En el mes de marzo de 1995 el W3C comenzó a trabajar en la versión 3.0, destinada a proporcionar numerosas y fundamentales modificaciones, agregando capacidades adicionales, tales como tablas y flujo de texto alrededor de imágenes. Algunas de las cuales se vieron implementadas por los nuevos navegadores, que tanto Netscape como Microsoft lanzaron por entonces al mercado, antes de que fuera alcanzado el consenso, mientras otras (por ejemplo el soporte para las matemáticas) no se tomaron nunca en consideración[VIAN2004].

La versión HTML 3.2 desarrollada a principios de 1996, elimina varias funcionalidades de versiones anteriores, esto por la incompatibilidad de los navegadores Web como Netscape y Microsoft, pues cada uno tiene distintas formas de visualizar el código HTML, es decir un mismo documento en HTML se podía ver de distinta forma, dependiendo del navegador.

Esta versión introduce más fuentes de texto; los *applet* Java, que permiten la creación de páginas HTML interactivas; lenguajes de script <sup>2</sup>, y los *Client Side Image*. Mejora la portabilidad del HTML, ya que puede ser empleado en una amplia gama de soportes; teletipos, terminales Dos, Windows o Macintosh, así como en medios no visuales(síntesis vocal y braille). También introduce la posibilidad de utilizar hojas de estilo en cascada(CSS, por sus siglas en inglés), para separar las indicaciones de presentación del documento. Pero sigue permitiendo definir características de diseño dentro del archivo HTML, y no existe un control formal por parte del navegador.

---

<sup>2</sup>Un Lenguaje de Script es un lenguaje de programación que se utiliza dentro de un documento HTML, permite variar dinámicamente su contenido.

La versión 4.0 del HTML marca el restablecimiento de la separación puntual entre presentación y estructura, y cuenta con el compromiso de Microsoft y Netscape de aceptar su resultado y hacer compatibles sus navegadores. El HTML 4.0 se publica en diciembre de 1997. Pasa por una revisión en 1998 y es corregido en diciembre de 1999 con la versión 4.01.

En ésta versión el uso de las hojas de estilo CSS adquieren más peso, llegándose a desaconsejar abiertamente el uso de numerosos atributos de presentación dentro de los documentos HTML, de esta forma HTML da un paso hacia la restauración de de la estructura lógica de sus documentos y extiende sus mecanismos de presentación. La información de estilo puede encontrarse en el elemento concreto al que se refiere, en el encabezado general de cada uno de los documentos, o en un documento aparte, referenciado por el documento HTML. Este último es el procedimiento que aconseja el W3C y que permite que una única hoja sea utilizada por más de un documento, manteniendo así eficazmente la uniformidad del proyecto gráfico por todas las páginas del sitio y simplificando el procedimiento de modificación del diseño, ya que cambios y correcciones pueden ser llevados a cabo sin intervenir en las páginas afectadas.

La última versión de HTML(2007) hasta este momento es la 5, la cual introduce nuevas características para ayuda a los autores de aplicaciones Web, nuevos elementos basados en la investigación dentro de las prácticas actuales de edición, y se ha dado especial atención a la definición de criterios claros para el uso de aplicaciones de usuario, en un esfuerzo por mejorar la interoperabilidad[W3C].

Html 5 incluye elementos destinados a enriquecer la presentación de documentos. Elementos semánticos como `article`, `header`, `hgroup`, `nav`, `section`, `aside` y `footer`. Con ellos se pretende evitar que los autores abusen del elemento `div` para delimitar partes de un documento. Los blogs y los sitios de noticias han influido en gran medida en esta evolución[SCHA2010].

Se puede ver que HTML ha ido evolucionando desde sus inicios, agregando



o eliminando etiquetas. Las primeras versiones de HTML eran sencillas con el fin de que fuera fácil editar un documento HTML. Los navegadores Web mostraban las páginas incluso con errores de sintaxis, tratando de interpretar lo que el usuario habría querido mostrar.

Con el paso del tiempo, los estándares han ido haciendo más rígida y formal la sintaxis del lenguaje.

El lenguaje HTML ya se ha definido como un estándar, esto por las recomendaciones publicadas por el consorcio internacional W3C. Pues en éste se encuentran las especificaciones oficiales de HTML que describen las instrucciones del lenguaje. Esto permite visualizar páginas Web independientemente del sistema operativo o la arquitectura del equipo del usuario.

El HTML, cuya formalización está aceptada universalmente y que todos los navegadores existentes están en condición de interpretar, presenta unas limitaciones que han provocado en el W3C investigar nuevas opciones más potentes y flexibles. Estos esfuerzos han llevado a la definición de otro lenguaje para la creación de documentos web, el Extensible Markup Language(XML).

### **2.3. Extensible Markup Language (XML)**

XML es el acrónimo de “eXtensible Markup Language” (Lenguaje extensible de marcado), se propone integrar, enriquecer y probablemente sustituir, a largo plazo al HTML (HyperText Markup Language) como lenguaje de marcado estándar para la Web.

El lenguaje XML comenzó a desarrollarse en septiembre de 1996, respaldado por el W3C con un claro propósito: diseñar un lenguaje de marcas optimizado para poder ser utilizado en Internet. XML debía combinar la simplicidad de HML con la capacidad expresiva de XML. En su definición participaron empresas como Microsoft, IBM, Sun Microsystems, Novell y Hewlet-Packard.

La versión 1.0 fue aprobada por el W3C en la conferencia sobre SGML/XML celebrada en Washington en diciembre de 1997.[BRUN2001]

La necesidad de separar presentación y datos, ha sido un impulso para la búsqueda de un nuevo estándar, para potenciar la funcionalidad de la gestión editorial y gráfica de los documentos de la World Wide Web, la certificación y el control de los contenidos presentes en los sitios. Nuevamente el punto de partida ha sido SGML, cuyas ventajas respecto al HTML son sustancialmente tres:

- La extensibilidad que permite la definición de conjuntos personalizados de marcas.
- La garantía de los elementos estructurales definidos en un archivo externo (DTD, Document Type Definition).
- La validación que obliga a todos los documentos a pasar por un control que certifica su conformidad a las reglas definidas en el DTD (Document Type Definition).

El W3C ofrece una solución a este problema, con la creación de un lenguaje de Marcado Extendido XML, metalenguaje de marcado diseñado para el intercambio en la red de documentos estructurados, es decir, que permite crear una clase de objetos que almacenan información estructurada y dividida en tres partes: contenido, estructura y presentación.

Con el XML el consorcio W3C propone un instrumento que pretende conservar las ventajas del SGML y por lo tanto es:

- Auto-descriptivo, capaz crear documentos en los cuales la elección de los nombres de los elementos pueden estar en función de la comprensión de su rol estructural.
- Especifica las reglas de composición y las posibles relaciones entre varias partes de los documentos.

- Independiente de la plataforma de uso por ser un estándar abierto.
- Facilita la conversión entre los formatos, y se propone como una sintaxis intermedia para expresar datos de forma independiente de la aplicación que lo ha creado.
- Extensible

Un documento XML, posee una estructura lógica y otra física. Físicamente, el documento está compuesto por etiquetas que poseen un contenido y están identificadas por un nombre. En cuanto a la estructura lógica, el documento está compuesto de declaraciones, elementos, comentarios, referencias a caracteres e instrucciones de procesamiento, todos indicados por una marca explícita.

Un documento XML está formado por marcas y por el contenido textual propiamente dicho, los datos. Para diferenciar en un documento XML, las marcas de los datos, la especificación establece la utilización de los caracteres reservados < y >. Estos caracteres permiten delimitar los caracteres que constituyen una marca. En la Figura 2.6 se muestra un pequeño ejemplo de la estructura de un documento XML.

XML usa el término *elemento* para hacer referencia a cada uno de los componentes estructurales o secciones de un documento XML. Un elemento queda delimitado por una marca o etiqueta de inicio y por una marca o etiqueta de fin. Estas dos marcas comparten el mismo nombre. Para diferenciar qué marca determina el inicio de un elemento y qué marca determina su fin, se tiene que escribir el carácter / a continuación del carácter < en la etiqueta que marca el final del elemento.

En el ejemplo anterior Figura 2.6, la etiqueta </unidades> señala el final del elemento *unidades*, que comienza con la marca <unidades>. El contenido de un elemento puede a su vez contener a otros elementos, con sus correspondientes marcas de inicio y de fin.

```
<?xml version="1.0"?>

<recibo>
  <cabecera>
    <cliente>
      <nombre> James Swart</nombre>
      <rfc>SWJA701211FRE</rfc>
    </cliente>
    <fecha-entrega>10-12-2010</fecha-entrega>
  </cabecera>
  <articulos>
    <articulo>
      <id>Tornillos</id>
      <unidades>300</unidades>
      <precio>10</precio>
      <total>3000</total>
    </articulo>
  </articulos>
  <articulos>
    <articulo>
      <id>tuercas</id>
      <unidades>500</unidades>
      <precio>20</precio>
      <total>10000</total>
    </articulo>
  </articulos>
</recibo>
```

Figura 2.6: Estructura de un documento XML

En el momento de añadir marcas a un documento, es posible utilizar el marcado aplicando distintos niveles de granularidad. El término granularidad se refiere al nivel de detalle con el que se destaca el contenido informativo del documento, mediante la inclusión de marcas. Como resultado de subdividir los elementos de un documento XML en otros elementos, se puede obtener una jerarquía de elementos anidados. En la Figura 2.7 se muestra la jerarquía de elementos del ejemplo anterior Figura 2.6.

La organización jerárquica de la información, permite que no se llegue a perder la relación que hay entre los datos. Así, cuando un elemento está anidado en otro elemento situado en un nivel superior, se sabe que entre esos dos elementos existe una relación lógica.

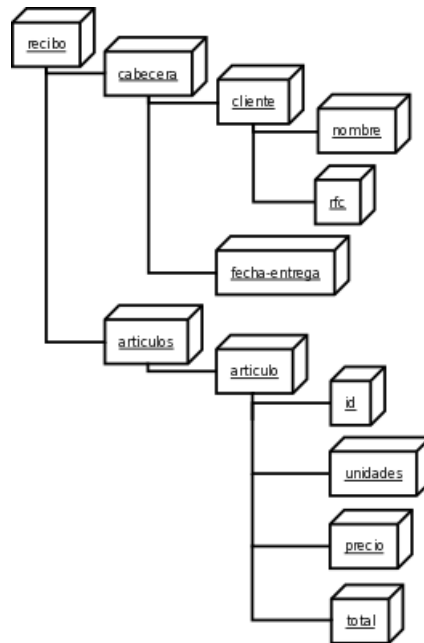


Figura 2.7: Estructura jerárquica de un documento XML

Otra ventaja que ofrece la organización jerárquica de los documentos XML consiste en que ofrece una mayor flexibilidad en los procesos de recuperación textual: una búsqueda sobre el contenido textual de un documento se puede acotar a un conjunto determinado. Por ejemplo, recuperar una palabra que aparece en los datos de un elemento determinado[BRUN2001].

Todos los elementos que conforman un documento XML, deben estar comprendidos en un único elemento: el elemento documento o raíz. El elemento documento constituye la base de la estructura jerárquica de los documentos XML y tiene las siguientes características:

1. Incluye a todos los elementos del documento.
2. No está comprendido en ningún otro elemento.

## 2.4. Documentos bien formados y válidos

Con el fin de que las aplicaciones de software sean capaces de interpretar las marcas que identifican a cada elemento y deducir la estructura jerárquica del documento XML para su posterior procesamiento, es necesario que las marcas estén anidadas correctamente, que el documento este bien formado.

Un documento bien formado debe cumplir con ciertas restricciones sintácticas:

1. Poseer un único elemento raíz que contiene a los demás.
2. Cada elemento está contenido en otro o en la raíz.
3. Todos los elementos deben contener elementos completos, esto es, para cada etiqueta de apertura debe haber una de cierre.
4. Todos los atributos de los elementos deben tener su valor entre comillas, ya sean simples o dobles.
5. Las etiquetas en XML, a diferencia con el HTML, diferencian entre mayúsculas y minúsculas.
6. Los elementos que no contengan información, pueden abrir y cerrarse con una única etiqueta.
7. Todos los elementos se deben cerrar con su etiqueta correspondiente.

La especificación de las etiquetas utilizables, de sus reglas y definiciones, así como el orden que pueden asumir dentro de un documento XML están contenidos en la DTD (Document Type Definition), que puede estar definida a través de un archivo externo, o bien dentro del documento mismo. Una DTD se puede ver en la Figura 2.8, está corresponde al ejemplo de la Figura 2.6.

La DTD del ejemplo, indica que el elemento *recibo* esta compuesto de dos elementos: *cabecera* y *articulos* ; el elemento *cabecera* esta compuesto por : *cliente* y *fecha-entrega*; el elemento *articulos* esta compuesto por un elemento

```
<!ELEMENT recibo ( cabecera, articulos ) >
<!ELEMENT cabecera ( cliente, fecha-entrega ) >
<!ELEMENT cliente ( nombre, rfc ) >
<!ELEMENT nombre ( #PCDATA ) >
<!ELEMENT rfc ( #PCDATA ) >
<!ELEMENT fecha-entrega ( #PCDATA ) >
<!ELEMENT articulos ( articulo+ ) >
<!ELEMENT articulo ( id, unidades, precio, total ) >
<!ELEMENT id ( #PCDATA ) >
<!ELEMENT unidades ( #PCDATA ) >
<!ELEMENT precio ( #PCDATA ) >
<!ELEMENT total ( #PCDATA ) >
```

Figura 2.8: DTD

de tipo *articulo* que puede aparecer una o más ocurrencias dentro del documento XML; el elemento *cliente* esta formado por: un elemento *nombre* y un elemento *rfc*; el elemento *articulo* esta compuesto por un elemento *id*, un elemento *unidades*, un elemento *precio*, y un elemento *total*; y los elementos *nombre*, *rfc*, *fecha-entrega*, *id*, *unidades*, *precio* y *total* son de tipo texto (#PCDATA) y la secuencia de aparición de estos elementos en el documento XML, es la secuencia en que aparecen en la DTD.

La presencia de la DTD da lugar a documentos válidos. Un documento XML será un documento válido si es un documento bien formado y además cumple las restricciones indicadas en su DTD[BRUN2001].

Es necesario precisar que gracias a su concepción modular el XML favorece la reutilización de las aplicaciones web y, por consiguiente, las DTD se pueden mezclar y fundir en forma compleja.

Un mismo documento, por lo tanto, puede estar compuesto de elementos definidos por DTD diferentes. Esto puede provocar problemas de reconocimiento y de ambigüedades, sobre todo cuando los elementos definidos en DTD diferentes presentan el mismo nombre. Para evitar estas posibles colisiones conceptuales, se han establecido los espacios de nombres (Namespaces en XML), cuyo objeto es proporcionar un identificador único que pueda ser usado sin ambigüedad, permitiendo el uso de nombres con significados distintos. Ese identificador úni-

co es introducido a través de una etiqueta “xmlns” que especifica el origen del nombre.

En un principio se pensó que XML serviría solamente para la creación de documentos, sin embargo a continuación se fue consolidando la idea de que se podría emplear también para formular un formato de transporte para la exportación de datos de una aplicación a otra.

La búsqueda de un mecanismo más refinado para especificar los criterios de validación de los documentos XML ha llevado al consorcio a la definición del XML Schema. Éste, al igual que las DTD, tiene por objeto definir las relaciones y las especificaciones formales de los elementos y atributos permitidos, pero además proporciona un mecanismo que permite definir las características de cada elemento tanto en términos de estructura como de contenido??.

El esquema correspondiente al ejemplo anterior se muestra en la Figura 2.9.

Los XML Schemas definen una sintaxis alternativa a la DTD y utilizan la sintaxis del mismo XML. Se pueden ver las siguientes ventajas:

- Fáciles de aprender, ya que utiliza el mismo lenguaje XML.
- Soportan tipos de datos: numéricos, fechas, etc.
- Se procesan igual que los documentos XML.

Dado que un esquema XML define la estructura válida para un documento XML se puede encontrar:

- Los elementos que pueden aparecer en el documento.
- Atributos que se pueden utilizar junto a cada elemento.
- El orden en el que deben aparecer las elementos.
- El número permitido de elementos.



```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="recibo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cabecera" />
        <xs:element ref="articulos" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="cabecera">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cliente" />
        <xs:element ref="fecha-entrega" type="xs:date"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="cliente">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nombre" type="xs:string"/>
        <xs:element ref="rfc" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="articulos">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="articulo" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="id" type="xs:string" />
              <xs:element ref="unidades" type="xs:integer"/>
              <xs:element ref="precio" type="xs:decimal"/>
              <xs:element ref="total" type="xs:decimal"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 2.9: Esquema XML

- Si un elemento puede ser vacío o no.
- Tipos de datos para elementos y atributos.
- Valores por defecto y fijos para elementos y atributos .

La separación física de los elementos, que garantiza la separación conceptual, llega hasta el punto en que XML no se encarga de definir cómo se debe organizar la representación, delegándola a otro lenguaje, el XSL, que asociado a los documentos XML describe las reglas para aplicar a la presentación.

Se puede observar que XML es un lenguaje sencillo y estricto, es fundamental en el intercambio de distintos tipos de datos, y es un formato que permite la lectura de datos a través de distintas aplicaciones.

XML cuenta con un conjunto de tecnologías, que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios, dado que XML sirve para estructurar, almacenar e intercambiar información. En este conjunto XML destacan:

- XSL<sup>3</sup> – Lenguaje Extensible de Hojas de Estilo, su función es crear hojas de estilo, transformar, ordenar, filtrar datos XML, y darles formato.
- XPath<sup>4</sup> – Lenguaje de Rutas XML, sirve para acceder a partes de un documento XML en concreto, como sus atributos y elementos.
- XLink<sup>5</sup> – Lenguaje de enlace XML, inserta elementos en documentos XML para crear enlaces entre recursos XML, permitiendo crear enlaces bidireccionales.
- XPointer<sup>6</sup> – Lenguaje de Direccionamiento XML, permite el acceso a la estructura interna de un documento XML, es decir, que va a un punto específico del documento.
- XQL<sup>7</sup> – Lenguaje de Consulta XML, facilita la extracción de datos desde documentos XML. Se basa en operadores de búsqueda de un modelo de

---

<sup>3</sup><http://www.w3.org/TR/xsl/>

<sup>4</sup><http://www.w3.org/TR/xpath/>

<sup>5</sup><http://www.w3.org/TR/xlink/>

<sup>6</sup><http://www.w3.org/TR/xptr/>

<sup>7</sup><http://www.w3.org/TR/xquery-requirements/>

datos para documentos XML que puede realizar consultas en infinidad de tipos de documentos como son documentos estructurados, colecciones de documentos, bases de datos, catálogos, etc.

### 2.4.1. Aplicaciones XML

Desde el inicio se ha puesto a disposición de los usuarios una gran cantidad de software para elaborar documentos XML: analizadores sintácticos (parser), editores, navegadores, motores de búsqueda. Gran parte de este interés es debido al hecho de que XML, además de formato de representación, puede ser empleado como formato de intercambio entre programas, encontrando aplicación en el área del comercio electrónico y del trabajo en colaboración.

XML permite el intercambio de bases de datos. Con este lenguaje se puede definir e intercambiar tanto la estructura semántica de las bases de datos, como el contenido, por lo tanto se podría convertir en un lenguaje universal para etiquetar las estructuras de datos.

A partir de XML, han surgido numerosos lenguajes que se plantean como solución a las necesidades de distintas comunidades. Entre las numerosas iniciativas se encuentran:

- CML<sup>8</sup> (Chemical Markup Language): lenguaje orientado a la descripción de estructuras moleculares.
- MathML<sup>9</sup> (Mathematical Markup Language): especificación de bajo nivel para la descripción de las matemáticas, proporciona una base para la inclusión de expresiones matemáticas en las páginas Web.
- SMIL<sup>10</sup> (Synchronized Multimedia Integration Language): lenguaje para la sincronización de contenidos multimedia en la Web.

---

<sup>8</sup> < <http://www.xml-cml.org/> >

<sup>9</sup> < <http://www.w3.org/Math/> >

<sup>10</sup> < <http://www.w3.org/AudioVideo/> >

- VML<sup>11</sup> (Vector Markup Language): aplicación que define un formato para la codificación de la información vectorial.
- XBRL<sup>12</sup> (eXtensible Business Reporting Language): lenguaje de comunicación electrónica de datos financieros y de negocio. Ofrece grandes beneficios en la preparación, análisis y comunicación de información empresarial.

Las aplicaciones XML que se describieron son sólo algunas de las muchas aplicaciones que pueden existir hasta el momento.

---

<sup>11</sup> < <http://www.w3.org/TR/NOTE-VML> >

<sup>12</sup> < <http://www.xbrl.org/Home/> >

---

## Capítulo 3

# Lenguajes para Textos Científicos

Hace tiempo el intercambio de información de libros, revistas, periódicos, etc., implicaba bastante tiempo, dinero, y capacidad para transportarla, lo que resultaba una tarea difícil.

Hoy en día estos problemas se han prácticamente eliminado, pues gracias a la digitalización de la información y al Internet el intercambio de información se lleva a cabo de una manera más rápida, que dependiendo del tamaño del archivo de información puede suceder en cuestión de segundos.

Por otra parte, el hecho de que algunos de los problemas se hayan resuelto, no implica que no se generen otros. Como se sabe Internet es un acervo de información extenso, donde se puede encontrar texto, audio, video, etc., todo esto en formatos distintos.

Toda esta información heterogénea que se encuentra en Internet, genera un conflicto sobre la recuperación de la información. Existen diversas soluciones

que proponen mejorar la recuperación de la información en Internet. Una de las cuales es el desarrollo de metadatos distribuidos en la Web.

Un mecanismo que permite recuperar la información con respecto al modelo de metadatos, es el lenguaje de marcado, que ofrece una respuesta interesante a la representación del conocimiento y a las bibliotecas virtuales. Y a su vez permiten establecer mecanismos útiles en la recuperación e intercambio de información.

Como se mencionó en el capítulo anterior, la tendencia a utilizar la noción de metadatos para agilizar el procesamiento de información en multitud de taras, existe desde hace varios años.

El objetivo es localizar e intercambiar información, sin perder datos, en grandes colecciones de documentos en formato electrónico. La solución a este objetivo es estandarizar los documentos a intercambiar, estandarizar no significa que todos los documentos deban lucir igual, ni siquiera que las secciones de un documento deban llamarse de una o de otra manera, significa que estos documentos deben contruirse en base a unas normas, que en este caso sería en base a los metadatos, los cuales proporcionan la información necesaria de los documentos.

Existen varios estándares que facilitan el intercambio de información pero que generalmente están enfocados a áreas de conocimiento específicos. En esta investigación sólo se verá el estándar de TEI<sup>1</sup>(Text Encoding Initiative) y NARSTO<sup>2</sup>(North American Research Strategy for Tropospheric Ozone).

### 3.1. Text Encoding Initiative (TEI)

Es un proyecto que surge de la necesidad de intercambiar y difundir información almacenada en formato electrónico. Se puede utilizar en principio

---

<sup>1</sup><http://www.tei-c.org/>

<sup>2</sup><http://www.narsto.org/>

en cualquier área que se desee, pero se utiliza principalmente en el área de humanidades.

El proyecto del Consorcio Internacional para el etiquetado de textos TEI (Text Encoding Initiative) es una iniciativa que ha partido de diversas asociaciones profesionales en el campo de las humanidades. Se trata de definir unas reglas generales para la representación de materiales textuales en formato electrónico.

El objetivo de TEI es fomentar el uso de etiquetas rigurosas y productivas para cualquier clase de texto, aunque su aportación más directa se produce en el campo de los textos con valor cultural y científico.

El proyecto inicia en 1987, respaldado por tres asociaciones: ACH (Association of Computers in the Humanities), ACL (Association for Computational Linguistics), y ALLC (Association of Literary and Linguistic Computing). Pero es hasta 1994, seis años más tarde del inicio del proyecto, cuando publican sus normas de estandarización.[MEND2002]

Al comienzo de este proyecto, en una conferencia que tiene sede en Vassar Collage N.Y., en noviembre de 1987, definen los objetivos generales del TEI:

- Ser suficientes para representar las características textuales necesarios para la investigación.
- Ser sencillas, claras y concretas.
- Ser fáciles para los investigadores de utilizar sin software de propósito especial.
- Permitir la definición rigurosa y el procesamiento eficaz de textos.
- Prevenir las extensiones definidas por los usuarios.
- Ajustarse a los estándares existentes y en desarrollo.

Dado que el mundo de estudio es amplio y diverso, para que las normas fueran aceptadas era importante asegurar que:

1. El núcleo de las características textuales fuera fácilmente compartido.
2. Agregar o eliminar fácilmente carecterísticas especiales a un texto.
3. Fueran posibles múltiples codificaciones semejantes de una misma característica.
4. La riqueza del marcado debía ser definida por el usuario, con un mínimo nivel de exigencia.
5. Tener siempre una documentación adecuada del texto y su codificación.

En el año 2000 se establece una nueva organización denominada TEI Consortium (Text Encoding Initiative Consortium), cuyo objetivo principal es fomentar una comunidad de usuarios que participen en el desarrollo y uso de las directrices del TEI, además de promover y darle mantenimiento, para que sea académico y económicamente independiente. El consorcio tiene sedes en la Universidad de Virginia<sup>3</sup>(Electronic Text Center e Institute for Advanced Technology in the Humanities, USA) , Universidad de Bergen<sup>4</sup> (Humanities Information Technologies Research Programme), Universidad de Brown<sup>5</sup>(Scholarly Technology Group, USA), y la Universidad de Oxford<sup>6</sup>(Research Technologies Service, UK).

TEI se desarrolló para representar gran variedad de información como: textos literarios, lingüísticos, de preservación, de museos, etc., es decir, se buscaba un estándar que permitiera el fácil intercambio de la información. Así es como TEI pasa a ser un estándar internacional e interdisciplinario de marcado de textos. Pero TEI no sólo se limita al área de humanidades, sino que también

---

<sup>3</sup><<http://www.virginia.edu/>>

<sup>4</sup><<http://www.uib.no/en>>

<sup>5</sup><<http://www.stg.brown.edu/>>

<sup>6</sup><<http://www.ox.ac.uk/>>



se ha utilizado en otras áreas para la estructuración de información electrónica de todo tipo.

TEI ha sido creado para el intercambio de información, con independencia de software y hardware, lo que hace de este estándar interoperable. Además permite un marcado semántico muy detallado.

TEI se relaciona bastante con las bibliotecas digitales, ya que estas son organizaciones que proveen repositorios de acervos y contenidos digitales, almacenados en distintos formatos electrónicos, que requieren distribuir, controlar el acceso, conservar la integridad, etc., y para poder tener un intercambio más uniforme de la información utilizan el lenguaje de marcado que propone el TEI para tener un estándar en sus documentos y poder así compartir e intercambiar dicha información.

### 3.1.1. Estructura

En 1994 TEI define 600 elementos en SGML (Standard Generalized Markup Language), poco a poco van surgiendo otras versiones como TEI P2 (1998), que se caracteriza por ser más operativa y modular, TEI P4 (2001) en XML (Extensible Markup Language) y TEI Lite orientada a bibliotecas digitales.

TEI es un esquema de codificación, que utiliza una DTD (Document Type Definition) para tener una restricción en la estructura y sintaxis del archivo. El estándar para la anotación estructural de textos TEI se compone principalmente de dos elementos: el encabezado `<teiHeader>` que recoge la información con valor documental, por ejemplo lo que se asemeja a los sistemas de catalogación de objetos bibliográficos, y el texto `<text>` que representa el texto a transcribir, todo esto precedido por la declaración del archivo que contiene la definición de la DTD, el cual pasará por un parser ya sea SGML o XML. El estándar TEI inicialmente se codificó en SGML, pero actualmente se codifica en XML.

El encabezado `<teiHeader>` contiene información descriptiva y declarativa del texto, este elemento puede contener:

- `<encodingDesc>` que justifica la relación entre un texto electrónico y la fuente o fuentes de las que éste se deriva.
- `<fileDesc>` que contiene una descripción bibliográfica del archivo electrónico.
- `<profileDesc>` provee una descripción puntualizada de los aspectos no bibliográficos de un texto, explícitamente los idiomas y dialectos usados, los participantes que intervienen, fecha de creación, etc.
- `<revisionDesc>` permite llevar un registro de los cambios realizados sobre la versión electrónica.

Dentro de la etiqueta `<text>` se encuentra el texto que se quiere transcribir, el cual puede ser de cualquier tipo, puede ser único (un texto en verso o teatral, una novela, etc.) o compuesto (una recopilación de ensayos, una antología, etc.) A su vez este elemento puede contener:

- `<front>` contiene información preliminar (encabezado, prefacio, dedicatoria, etc.)
- `<body>` cuerpo de la obra, en caso de un texto combinado se utiliza la etiqueta `<groups>` cada uno conteniendo más grupos o textos
- `<back>` contiene cualquier tipo de apéndice

Todo este tipo de etiquetas que se utilizan para representar información, pueden contener dentro de ellas más etiquetas, atributos, etc.

Un texto individual se etiquetará siguiendo una estructura genérica como esta:

Un texto compuesto se etiquetará usando una estructura genérica como esta:

Los elementos están estructurados en cuatro secciones:

```

<TEI>
  <teiHeader> [ Información del encabezado TEI
                (información descriptiva y declarativa de un texto digital) ] </teiHeader>
  <text>
    <front> [ elementos preliminares (prefacios,dedicatorias, etc.) ] </front>
    <body> [ cuerpo del texto ... ] </body>
    <back> [ elementos finales (contiene anexos, etc.) ] </back>
  </text>
</TEI>

```

Figura 3.1: Estructura genérica de documentos TEI

```

<TEI>
  <teiHeader> [ información del encabezado del conjunto de textos ] </teiHeader>
  <text>
    <front> [ elementos preliminares del conjunto de textos ] </front>
    <group>
      <text>
        <front> [ elementos preliminares del primer texto ] </front>
        <body> [ cuerpo del primer texto ] </body>
        <back> [ elementos finales primer texto ] </back>
      </text>
      <text>
        <front> [ elementos preliminares del segundo texto ] </front>
        <body> [ cuerpo del segundo texto ] </body>
        <back> [ elementos finales del segundo texto ] </back>
      </text>
      [ aquí van los demás textos o grupos de textos ]
    </group>
    <back> [ elementos finales del conjunto de textos ] </back>
  </text>
</TEI>

```

Figura 3.2: Estructura compuesta de documentos TEI

- Descripción del archivo: contiene la descripción bibliográfica del archivo electrónico.
- Descripción sobre la codificación: documenta la relación entre el texto electrónico y la fuente del mismo, así como las prácticas de codificación y transcripción.
- Información sobre el perfil del texto: resuelve aspectos contextuales (fecha de creación, idioma, autor, género, etc.)
- Descripción de revisión: contiene un historial de los cambios hechos al archivo durante su desarrollo.

Los documentos TEI utilizan etiquetas basadas en palabras no numéricas, incluye etiquetas opcionales, obligatorias y recomendadas, repetibles y no repetibles, que contienen atributos, y se sugiere establecer control para algunas de ellas.

En la Figura 3.3, se muestra los datos bibliográficos de un libro codificados con los elementos que proporciona el TEI.

```

<TEI>
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>El mundo de Sofía</title>
        <author>Jostein Gaarder</author>
      </titleStmt>
      <sourceDesc>
        <p>Traducción original del libro,impreso</p>
      </sourceDesc>
      <editionStmt>
        <p>Primera edición, México<date>1995</date> </p>
      </editionStmt>
    </fileDesc>
    <profileDesc>
      <langUsage>
        <language ident="es">Español.</language>
      </langUsage>
      <textDesc n="novel">
        <channel mode="w">impreso</channel>
        <constitution type="single"/>
      <derivation type="original"/>
        <domain type="art"/>
        <factuality type="mixed"/>
      </textDesc>
    </profileDesc>
    <revisionDesc>
      <change when="2008"> última revisión </change>
    </revisionDesc>
  </teiHeader>
  <text>
    <body>
      <p>Poco antes de cumplir los 15 años, la joven Sofía recibe una misteriosa carta anónima con las siguientes preguntas: "¿Quién eres?", "¿De donde viene el mundo?". Este es el punto de partida de una apasionada expedición a través de la historia de la filosofía con un enigmático filósofo. A lo largo de la novela, Sofía irá desarrollando su identidad a medida que va ampliando su pensamiento a través de estas enseñanzas: porque la Verdad es mucho más interesante y más compleja de lo que podría haber imaginado en un principio. </p>
    </body>
  </text>
</TEI>

```

Figura 3.3: Ejemplo de Archivo TEI

### 3.1.2. Funcionamiento de TEI

La operatividad de TEI se ha establecido entorno a cuatro Comités que comparten la responsabilidad de elaborar las directrices.

1. El Comité de Documentación de Textos se encarga de definir las etiquetas para identificar los textos (procedencia, ubicación, clase, categoría, etc.)
2. El Comité de Representación Textual se ocupa de describir física y lógicamente los textos. La descripción lógica abarca cuestiones como su estructura (capítulos, secciones, etc.), la tipografía, la maquetación, las notas, apéndices y referencias diversas, etc.
3. El Comité de Análisis e Interpretación de Textos trata el desarrollo de etiquetas que permiten la descripción lingüística y literaria del texto, así como cuestiones de intertextualidad, indización, etc.
4. El Comité de Cuestiones Metalingüísticas se ocupa de los problemas técnicos de la sintaxis utilizada en la etiquetación.

Las cuestiones resueltas por estos comités de TEI dan idea del grado de complejidad y precisión al que se quiere llegar.

La aplicación de éstas directrices señala directamente a la idea de una transformación radical en la difusión y acceso al conocimiento.

Inicialmente pretendían desarrollar una DTD para los textos que se utilizan en lingüística, literatura y estudios históricos, pero actualmente lo que pretenden es crear un sistema que permita a los investigadores de todas las disciplinas intercambiar y reutilizar recursos, independientemente del software y hardware que utilicen y sin tener en cuenta dónde están localizados.

### 3.1.3. Características del TEI

- Todos los textos TEI deben ir precedidos de un encabezado (TEI Header) que describa el texto.

- Un Comité específico – bibliotecarios y archiveros de Europa y USA- se encarga de diseñar las especificaciones de este encabezado.
- Su propósito es usar el encabezado como medio de control bibliográfico.
- El encabezado puede almacenarse como parte separada del documento al que se refiere o ir unida intrínsecamente a él.
- Se puede utilizar además para describir otros recursos en la red cuando sea necesario.
- Su gran flexibilidad permite adaptarlo fácilmente a cualquier usuario, ya que permite un mayor o menor nivel de detalle en la descripción.

#### 3.1.4. Implementaciones del TEI

- Biblioteca Virtual Miguel de Cervantes. La Biblioteca Virtual Miguel de Cervantes es un proyecto de digitalización documental, pone a disposición de la comunidad científica y de la población hispanoparlante en general, treinta mil obras de autores españoles o hispanoamericanos.
- Oxford Text Archive (OTA) Fundada en 1976 por Lou Burnard, OTA tiene más de veinte años de experiencia en el servicio de búsqueda de documentos electrónicos.

Las guías de TEI proporcionan medios para representar las características de los textos que se necesita identificar de forma explícita para facilitar el procesamiento de textos por programas de computadora. En particular la guía establece un conjunto de etiquetas que pueden ser insertadas en la representación electrónica de un texto, con el objetivo de marcar la estructura del documento teniendo como base características de interés lingüístico. Sin esas marcas explícitas muchas características de interés serían difíciles de localizar por medios mecánicos.

## **3.2. North American Research Strategy for Tropospheric Ozone (NARSTO)**

NARSTO es una alianza público-privada, que se estableció el 13 de febrero de 1995, cuyos integrantes provienen de los sectores gubernamentales, industriales, académicos y de servicios entre México, Estados Unidos y Canadá.[NARST2006]

Su misión principal es coordinar y mejorar la política de la investigación científica y la evaluación del comportamiento de los contaminantes troposféricos; sus actividades aportan elementos para la toma de decisiones con base en investigaciones científicas, y la determinación de estrategias eficientes, efectivas y funcionales para el manejo local y regional de la contaminación del aire.

NARSTO es originalmente el acrónimo de "North American Research Strategy for Tropospheric Ozone" (Estrategia de América del Norte para la Investigación del Ozono Troposférico).

Sin embargo, el término NARSTO ha pasado a significar una asociación trinacional, pública y privada, para tratar los múltiples aspectos de la contaminación troposférica, incluyendo partículas suspendidas y ozono.

La misión de NARSTO es planificar, coordinar, y facilitar la comprensión a largo plazo, de las políticas pertinentes de la investigación científica, la evaluación primaria y secundaria de las especies contaminantes emitidas, formadas, transformadas, y transportadas a la tropósfera sobre el Norte del continente Americano. En la actualidad el estudio está dirigido hacia la capa de ozono, las partículas y sus precursores.

Para lograr esta misión NARSTO proporciona una organización transversal y procesos de planeación que determinen las estrategias más efectivas para la investigación científica, establece y mantiene efectivos canales de comunicación con la policía, control tecnológico, la salud y comunidades de efectos ecológicos.

El objetivo central de los programas de NARSTO es proporcionar datos e información para su uso en la determinación realizable, eficiente, y estrategias efectivas en los entes locales y regionales de la capa de ozono y gestión de partículas finas.

La investigación de NARSTO se centra en cuatro grandes áreas técnicas:

1. Química atmosférica e investigación de modelado.
2. Investigación de emisiones.
3. Investigación de observaciones.
4. Integración de análisis y evaluación.

NARSTO coordina el intercambio de información y colaboración con otros científicos y los encargados de formular políticas (tanto nacionales como internacionales) a través de mecanismos de enlace.

La asociación se ocupa de varias labores, tales como recolectar información, realizar actividades de difusión, incluyendo comentarios críticos de la ciencia, evaluaciones del estado de la ciencia, archivado de datos y desarrollo de sistemas de recuperación.

En la Figura 3.4 se puede observar gráficamente la función de NARSTO.

En enero de 1997, el Department of Energy's Environmental Sciences Division anuncia la afiliación de NARSTO con el Quality Systems Science Center(QSSC). El QSSC se encuentra en el Centro de Análisis de Información del Dioxido de Carbono(CDIAC), Laboratorio Nacional de Oak Ridge(Oak Ridge, Tennessee).

El QSSC apoya programas, líderes de proyecto e investigadores en la definición de objetivos de calidad, garantías de calidad, planificación y coordinación de gestión de datos, y juega un papel de supervisor de calidad. El QSSC garantiza que los sistemas de gestión de calidad se ejecuten en el marco del programa



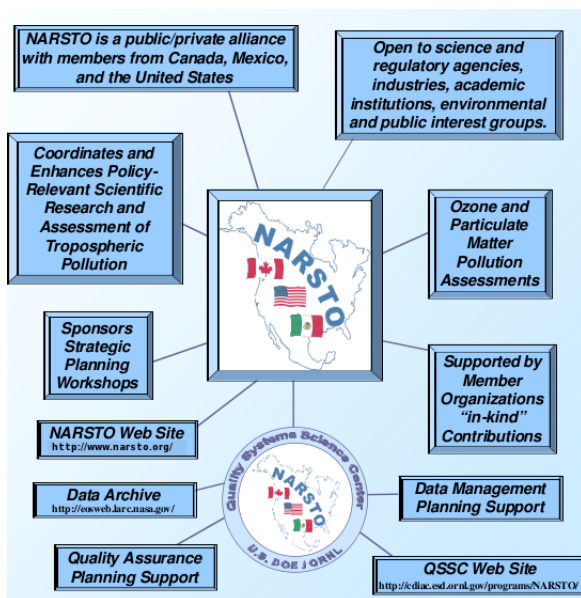


Figura 3.4: NARSTO

de NARSTO; planifica y realiza controles de calidad de elementos importantes del programa NARSTO; y coordina la gestión de datos, archivo de datos y actividades de difusión de datos. El QSSC asegura documentos y transmite información de datos y metadatos a el Permanent Data Archive de NARSTO.

NARSTO apoya al QSSC y al Permanent Data Archive, para almacenar los datos de los afiliados a él, apoya en la elaboración de archivos de datos y recursos de gestión de calidad, para ponerlos a disposición de la comunidad científica.

La lista siguiente son de algunos centros de calidad:

1. QSSC<sup>7</sup> Centro de Ciencia en Sistemas de Calidad (QSSC), Laboratorio Nacional de Oak Ridge.

<sup>7</sup><http://cdiac.ornl.gov/programs/NARSTO/>

2. NAtChem<sup>8</sup> Departamento de Medio Ambiente de Canada.
3. WDCA<sup>9</sup> Centro Mundial de Datos de Aerosoles, Ispra, Italia.

### 3.2.1. La plantilla de medidas atmosféricas de NARSTO

NARSTO trabaja con una plantilla en Excel<sup>10</sup> para el intercambio de información entre las organizaciones que están asociadas a éste. La plantilla es un libro Excel que está constituido por varias hojas, algunas de ellas describen el uso de éste, como la hoja de instrucciones, preguntas y respuestas frecuentes, ejemplos, etc., otras contienen datos que se utilizan para llenar la plantilla, como código de país, nivel de control de calidad, tipo de formato, tamaño de partículas, unidades de medida, métodos de análisis de laboratorio, código de provincia o estado, etc.

También se encuentran hojas que llevan una relación de las versiones de la plantilla, sitios de información sobre la plantilla, etc. Y por último y la más importante es la hoja que representa la plantilla, que es la que será llenada con los datos requeridos. La plantilla se puede observar en la Figura 3.5.

Las organizaciones que están en alianza con NARSTO, casi siempre están llevando a cabo investigaciones que analizan la calidad del aire y la capa de ozono, entre otras cosas con relación a esta área. Estas organizaciones requieren el intercambio de información para poder realizar sus análisis con respecto a otras investigaciones, y lo realizan con la ayuda de la plantilla que les proporciona NARSTO, puesto que esta plantilla les ofrece los datos requeridos para el intercambio de sus análisis.

La hoja principal de la plantilla de NARSTO esta compuesta de varios campos que se deben llenar, hay campos obligatorios y opcionales. Los campos

---

<sup>8</sup><<http://www.ec.gc.ca/natchem>>

<sup>9</sup><[http://gcmd.nasa.gov/records/GCMD\\_WDCA\\_Aerosol.html](http://gcmd.nasa.gov/records/GCMD_WDCA_Aerosol.html)>

<sup>10</sup><<http://cdiac.ornl.gov/programs/NARSTO/>>

presentan comentarios de ayuda de lo que se espera almacenar en el campo, y en algunos casos da ejemplos de la información a almacenar.

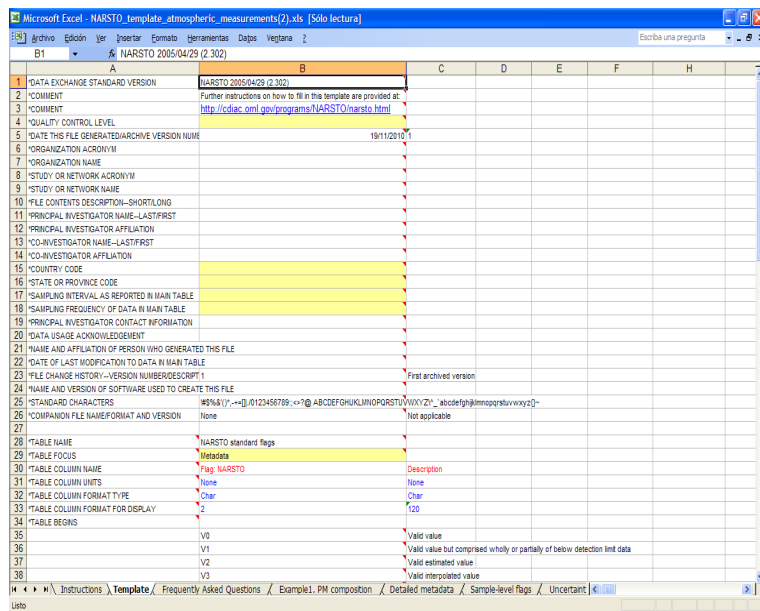


Figura 3.5: Plantilla de Excel NARSTO. Parte 1

Los datos más importantes que se requieren son:

- Fecha de creación del archivo.
- Acrónimo de la organización.
- Nombre de la organización.
- Acrónimo del grupo de estudio.
- Nombre del grupo de estudio.
- Descripción del contenido del archivo.
- Nombre del investigador principal.

- Afiliación del investigador principal.
- Nombre de los co-investigadores.
- Afiliación de los co-investigadores.
- Código del país.
- Código del estado o provincia.
- Intervalo de muestreo.
- Frecuencia de muestreo.
- Información de contacto del investigador principal.
- Reconocimiento del uso de los datos.
- Nombre y afiliación de la persona que genero el archivo.
- Fecha de la última modificación.
- Numero de versión.
- Nombre y versión del software para crear el archivo.

Estos datos son los que proporcionan la información general sobre el grupo de trabajo de la investigación realizada. Algunos de los campos se llenan con ayuda de las hojas que se encuentran en la plantilla, por ejemplo el campo código de país. La otra parte de la plantilla, Figura 3.6, se llena con los datos obtenidos de sus investigaciones realizadas.

En esta plantilla se encuentran datos que no son muy relevantes y algunos que sólo son opcionales, por otra parte la plantilla proporciona una hoja con un ejemplo de los metadatos importantes que se requieren, para poder llenar la plantilla correctamente, estos se muestran en la Figura 3.7.

Esta plantilla es usada por los centros de garantía de calidad (QSSC).

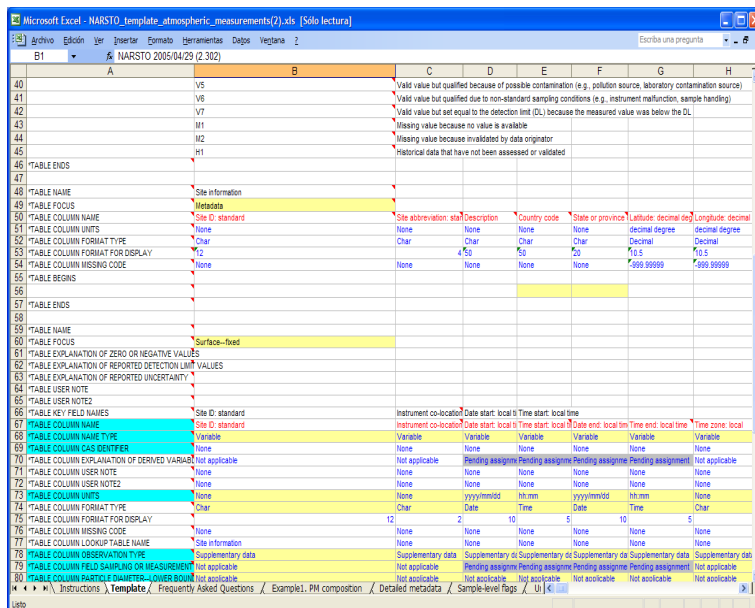


Figura 3.6: Plantilla de Excel de NARSTO. Parte 2

La plantilla Estándar de Intercambio de Datos DES por su acrónimo en inglés “Data Exchange Standard” está diseñada para ayudar a autores de datos a crear archivos DES. Cuando la plantilla ha sido llenada con metadatos y datos, una copia es salvada en formato ASCII CSV (valores separados por coma). El formato ASCII es la codificación final del archivo. El archivo ASCII es sometido al QC(Quality Control), donde es pasado a un conjunto de pruebas de garantías de calidad.

El QC proporciona los resultados de las pruebas de calidad (y algún QC también proporciona resúmenes estadísticos y argumentos) al autor de datos. El autor de datos repasa los informes, hace correcciones cuando es necesario, y somete de nuevo el archivo al QAC. Después de una comprobación final, del archivo ASCII, se hace el original de los datos y es colocado en el repositorio.

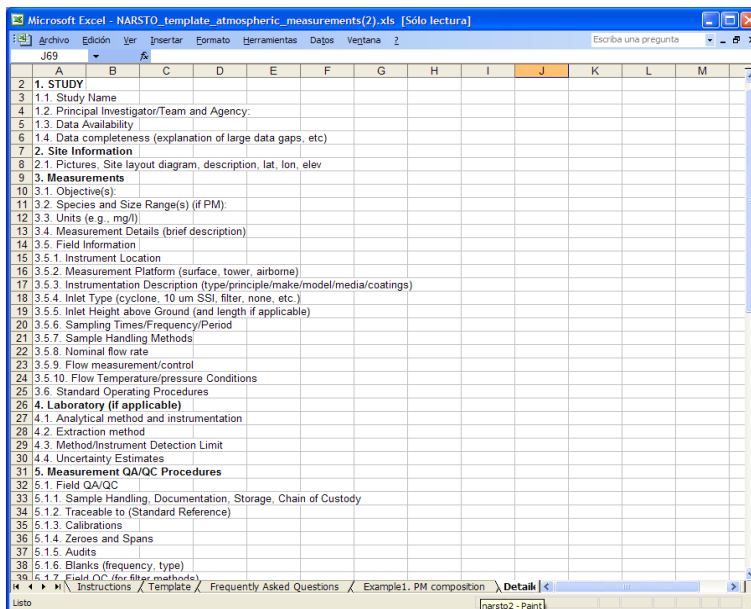


Figura 3.7: Metadatos NARSTO

En la Figura 3.8, se puede observar el posible flujo de la creación de archivos DES.

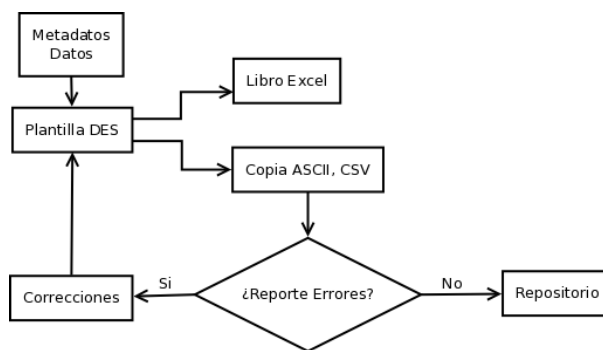


Figura 3.8: Flujo de la creación de archivos DES

---

## Capítulo 4

# Modelado de la Solución Propuesta

El proceso de solución de un problema con una computadora conduce a la escritura de un programa y a la ejecución del mismo.

El proceso que se debe seguir generalmente en el desarrollo de programas se divide en una serie de fases o pasos comunes. Las fases de solución de un problema más sobresalientes según [JOYA2008] son:

- **Análisis.** Se ocupa de entender el dominio del problema, requiere de una clara definición, donde se contemple exactamente lo que debe hacer el programa y el resultado o solución deseada, teniendo presente la especificación de los requerimientos. Los requerimientos son una descripción de las necesidades o deseos de un producto.
- **Diseño.** En la etapa de análisis del proceso de programación se determina *qué* hace el programa. En la fase de diseño se determina *cómo* hace el programa la tarea solicitada. Una vez analizado el problema, se diseña una solución que conducirá a un *algoritmo* que resuelva el problema.

- **Implementación(Codificación).** Fase en la que se adapta la solución a un entorno de programación específico. La codificación es la escritura en un lenguaje de programación de la representación del algoritmo desarrollada en las fases precedentes.
- **Pruebas.** Es el proceso dónde se asegura que todos los requerimientos del sistema se han comprobado, este tipo de pruebas se realizan para la detección de errores. Se requiere probar el software con usuarios reales que puedan evaluar el comportamiento del software con el fin de proporcionar retroalimentación al desarrollo del mismo.
- **Documentación.** Escritura de las diferentes fases del ciclo de vida del software, esencialmente el análisis, diseño y codificación, unidos a manuales de usuario y de referencia, así como normas para el mantenimiento.
- **Mantenimiento.** El programa puede ser actualizado, modificado o corregido, cada vez que sea necesario, de modo que se cumplan todas las necesidades de cambio de sus usuarios. A este tipo de cambios se denominan *mantenimiento del programa*. Después de cada cambio se debe actualizar la documentación para facilitar cambios posteriores.

En la Figura4.1 se puede observar el flujo que pueden presentar las fases de diseño.

## 4.1. Descripción del problema

El trabajo que se desarrollará será para beneficio del Centro de Ciencias de la Atmósfera de la UNAM, este centro utiliza la plantilla DES de NARSTO, para el intercambio de información, pero sólo son algunos datos los que capturan. Lo que implica que la plantilla no se utilice al 100 % aparte de que, como se vio en el capítulo anterior, el flujo del ingreso de datos a la plantilla sea algo laborioso. Por lo tanto lo que se intenta hacer es una adaptación de la plantilla



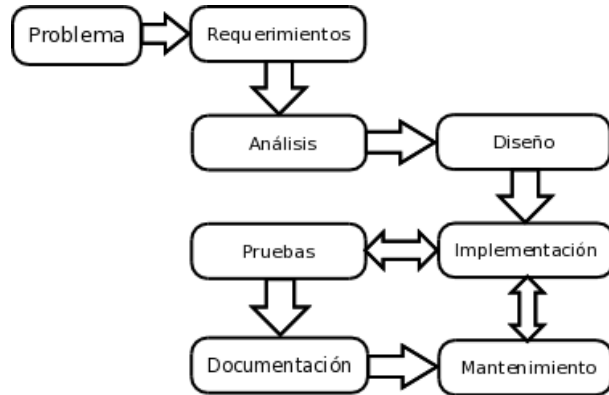


Figura 4.1: Fases de la solución de un problema.

NARSTO a las necesidades del CCA. Por lo tanto el planteamiento general de lo que se requiere, se describe a continuación.

Se requiere un sistema para facilitar la creación e intercambio de documentos, con información relacionada al Centro de Ciencias de la Atmósfera (CCA de la UNAM), por ejemplo datos relacionados a los informes técnicos de sus investigaciones y en general datos con distinto formato que el CCA de la UNAM utiliza. El intercambio de los documentos se lleva a cabo entre distintas dependencias universitarias, de las cuales la mayoría trabaja bajo entornos Windows, ésto implica que la aplicación deberá crearse en éste entorno, y cómo lo que se pretende es adaptar la plantilla de NARSTO, la aplicación deberá generarse en alguna hoja de cálculo (Microsoft Excel 2003 o Microsoft Excel 2007).

La funcionalidad esperada del sistema es:

- Capturar datos.
- Validar datos.
- Exportar datos.

## 4.2. Requerimientos

Un requerimiento se define como un atributo necesario dentro de un sistema, que puede representar una capacidad, característica o de un factor de calidad del sistema de tal manera que le sea útil a los clientes o usuarios finales.

- Requerimientos funcionales. Define que podría ser capaz de hacer el sistema.
- Requerimientos no funcionales. Estos requerimientos son adicionales a los requerimientos funcionales que debe cumplir el sistema, y corresponden a aspectos tales como la disponibilidad, mantenibilidad, flexibilidad, seguridad, facilidad de uso, etc.

El objetivo de capturar los requerimientos, es fundamental para los desarrolladores del software, para la comprensión de la naturaleza de los programas que deben construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión.[PRES2005]

### 4.2.1. Especificaciones de los Requerimientos

Los requerimientos que el sistema deberá satisfacer, son los siguientes :

1. Contar con una plantilla en Microsoft Excel que permita la captura de datos, relacionados a la información de plantilla de NARSTO.
2. El sistema deberá validar las fechas a capturar.
3. El sistema deberá permitir crear un Libro Excel en base a la plantilla para su posterior edición.
4. El sistema deberá generar documentos XML al menos con el conjunto de metadatos obligatorios.
5. El sistema deberá contar con un esquema XML que permita la especificación y validación de documentos XML.

6. El sistema deberá generar documentos XML bien formados y válidos.
7. El almacenamiento de los documentos XML deberá ser sencillo.
8. La capacitación para el uso de la aplicación deberá ser sencilla.

### 4.3. Análisis

Una vez que se han definido los requerimientos del sistema, se hará un análisis detallado del posible funcionamiento de éste con ayuda de los Casos de Uso.

#### 4.3.1. Caso de Uso General

Los casos de uso describen los caminos posibles que puede tener el usuario para interactuar con el sistema y como el sistema podría responder[DOMA2007].

El modelado de Casos de Uso es una técnica simple para modelar los requerimientos de la aplicación desde la perspectiva del usuario. Asimismo se utilizan para modelar cómo el sistema funcionará o cómo los usuarios desean que funcione.

Los Casos de Uso ayudan a describir qué es lo que el sistema debe hacer, y cómo éste interactúa con el usuario.

El Caso de Uso General se muestra en la Figura 4.2.

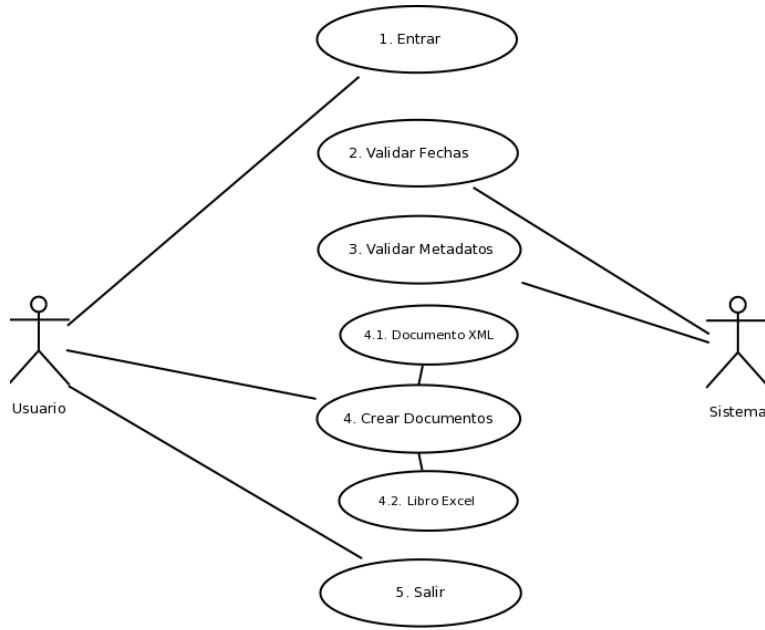


Figura 4.2: Caso de Uso General

### 4.3.2. Casos de uso

A continuación se muestran los Casos de Uso utilizados para el sistema en cuestión.

**Caso de Uso:** Entrar (al sistema)

**Actor:** Usuario

**Descripción:** El usuario desea entrar al sistema para ingresar información y crear nuevos documentos, ya sea XML o libros Excel, o ambos.

**Precodiciones:**

- ◊ El usuario debe tener instalada una versión de Microsoft Office (2003 ó 2007).

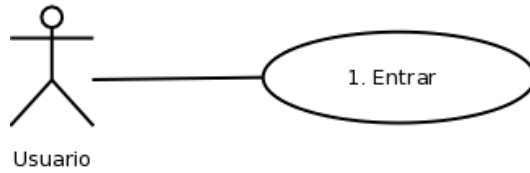


Figura 4.3: Caso de uso. Entrar

- ◊ El usuario debe abrir la plantilla Excel adecuada para acceder a la aplicación.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Abre la aplicación de “Excel”	2	Despliega la interfaz principal de “Excel”	
3	Selecciona el botón “Nuevo”	4	Despliega la interfaz “Nuevo libro”	
5	Selecciona la opción “Mis Plantillas”	6	Despliega la interfaz que tiene las plantillas guardadas	E1
7	Selecciona la plantilla adecuada a la aplicación	8	Despliega la interfaz de la plantilla seleccionada	

**Excepciones:**

Id	Nombre	Acción
E1	No se encuentra la plantilla	La plantilla no se despliega

**Postcondiciones:**

- ◊ El sistema ha desplegado la interfaz de la plantilla

**Plan de prueba del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Entrar	Plantilla Excel	Plantilla Disponible	Despliega interfaz inicial de plantilla
	Plantilla Excel	Plantilla no Disponible	El programa Excel no muestra la interfaz de la plantilla

**Caso de Uso:** Validar Fechas

**Actor:** Sistema

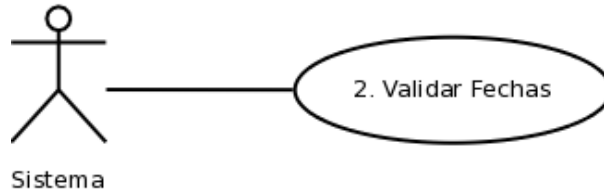


Figura 4.4: Caso de Uso. Validar Fechas

**Descripción:** El sistema válida que las fechas ingresadas sean correctas.

**Precodiciones:**

- ◇ El usuario debe haber ingresado al menos una fecha.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Seleccionar una fecha	2	Se establece la fecha en el campo seleccionado	E1

**Excepciones:**

<b>Id</b>	<b>Nombre</b>	<b>Acción</b>
E1	La fecha ingresada no es válida con respecto a la fecha o fechas ingresadas	

**Postcondiciones:**

- ◇ El sistema establece la fecha en la celda seleccionada

**Plan de prueba del sistema:**

<b>Caso de Uso</b>	<b>Entrada</b>	<b>Esperado</b>	<b>Obtenido</b>
Validar Fechas	Dar clic en una celda correspondiente a una fecha	Se seleccionó correctamente la fecha	La validación se realizó correctamente
	Dar clic en una celda correspondiente a una fecha	La fecha no es válida	El programa Excel muestra un mensaje de advertencia indicando el tipo de error con respecto a las fechas

**Caso de Uso:** Validar Metadatos

**Actor:** Sistema

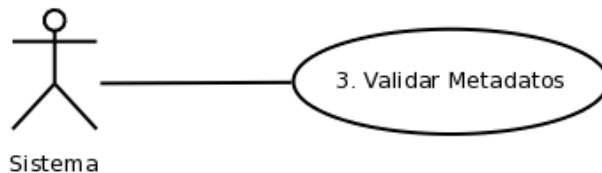


Figura 4.5: Caso de Uso. Validar Metadatos

**Descripción:** El sistema debe validar que se encuentren llenos los campos obligatorios.

**Precodiciones:**

- ◇ El usuario debe haber hecho clic en el botón Exportar.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1		2	Se verifica que no falte ningún campo obligatorio por llenar	E1

**Excepciones:**

Id	Nombre	Acción
E1	Faltan datos obligatorios	La aplicación envía un mensaje de advertencia indicando los campos obligatorios faltantes

**Postcondiciones:**

- ◇ El usuario agregó los datos faltantes

**Plan de prueba del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Validar Meta-datos	Dar clic en el botón Exportar	Verificar que no falte por llenar ningún campo obligatorio	Se verificaron los campos obligatorios
	Dar clic en el botón Exportar	No verifica los campos obligatorios	La aplicación no válida los campos obligatorios



**Caso de Uso:** Crear documentos

**Actor:** Usuario

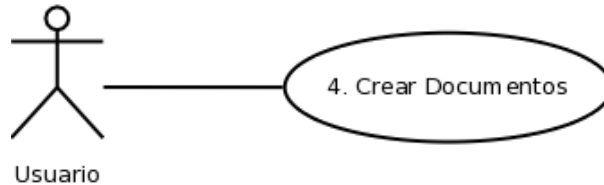


Figura 4.6: Caso de uso. Crear Documentos

**Descripción:** El usuario debe elegir alguna de las opciones disponibles para crear documentos, seleccionando alguna de las opciones ExportarXML o Guardar.

**Precondiciones:**

- ◊ El sistema despliega la interfaz de la plantilla

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Elige la opción “Exportar XML”	2	Despliega la interfaz “Exportar XML como”	E1
3	Elige la opción “Guardar”	4	Despliega la interfaz “Guardar como”	
5	Elige la opción “Salir “	6	Sale del sistema	

**Excepciones:**

Id	Nombre	Acción
E1	Faltan campos obligatorios	Muestra un mensaje de advertencia “Falta el campo___ favor de capturar”

**Postcondiciones:**

- ◇ El sistema crea documentos XML y Excel
- ◇ El sistema finaliza

**Plan de prueba del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Crear documentos	Crear documento XML	Opción Exportar XML	Documento XML
	Crear Libro Excel	Opción Guardar	Libro Excel
	Salir	Opción Salir	Cierra la aplicación

**Caso de Uso:** Crear documento XML

**Actor:** Usuario

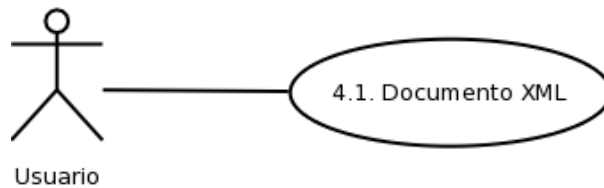


Figura 4.7: Crear Documento XML

**Descripción:** El usuario ingresa la información necesaria (metadatos) para la creación de documentos.

**Precondiciones:**

- ◇ El sistema despliega la interfaz de la plantilla

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Teclea los datos necesarios para crear un documento XML	2		
3	Presiona botón "ExportarXML"	4	Verifica que todos los datos obligatorios se hayan capturado	E1
5		6	Despliega la interfaz "Guardar XML como"	
7	Proporciona la ruta y nombre del archivo a guardar. Presiona el botón "Guardar"	8	Guarda la información XML en la ruta y con el nombre que proporciono el usuario. Regresa a la plantilla actual	E2
9	Presiona botón "Cancelar"	10	Regresa a la interfaz de la plantilla actual	

**Excepciones:**

Id	Nombre	Acción
E1	Faltan campos obligatorios	Muestra un mensaje de advertencia "Falta el campo... favor de capturar"
E2	No se proporciona el nombre del archivo a guardar y se presiona el botón Guardar	La interfaz "Guardar XML como" no hace nada

**Postcondiciones:**

- ◇ El sistema crea un documento XML con los datos proporcionados
- ◇ El sistema regresa a la interfaz de la plantilla actual

**Plan de prueba del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Crear documento XML	Exportar XML	Campos obligatorios completos	Documento XML
	Exportar XML	Campos obligatorios faltantes	Mensaje de advertencia "Falta el campo ___ favor de capturar"

**Caso de Uso:** Crear documento Excel

**Actor:** Usuario

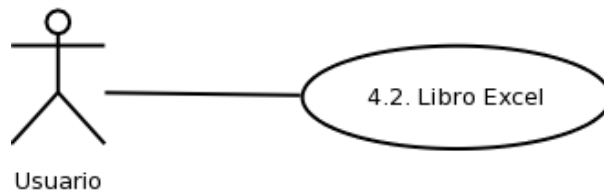


Figura 4.8: Caso de uso. Crear Documento Excel

**Descripción:** El usuario ingresa la información necesaria (metadatos) para la creación de documentos.

**Precondiciones:**

- ◇ El sistema despliega la interfaz de la plantilla

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Teclea los datos para crear un documento Excel	2		
3	Presiona botón "Guardar"	4	Despliega la interfaz "Guardar XML como"	
5	Proporciona la ruta y nombre del archivo a guardar. Presiona el botón "Guardar"	6	Guarda el documento con el nombre y en la ruta especificada	E1
9	Presiona botón "Cancelar"	10	Regresa a la interfaz de la plantilla actual	

**Excepciones:**

Id	Nombre	Acción
E1	No se proporciona el nombre del archivo a guardar y se presiona el botón Guardar	La interfaz "Guardar como" no hace nada

**Postcondiciones:**

- ◇ El sistema crea un Libro Excel basado en la plantilla con los datos proporcionados
- ◇ El sistema regresa a la interfaz del nuevo libro
- ◇ El libro creado se puede seguir editando posteriormente sin perder las propiedades de la plantilla

**Plan de prueba del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Crear documento Excel	Guardar		Libro Excel

**Caso de Uso:** Salir (del sistema)

**Actor:** Usuario

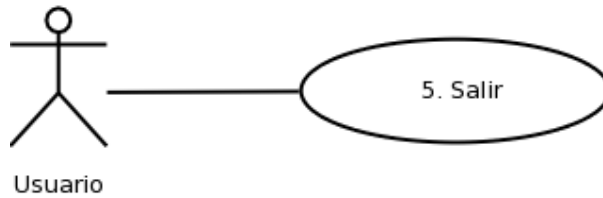


Figura 4.9: Caso de Uso. Salir

**Descripción:** El usuario desea salir del sistema.

**Precodiciones:**

- ◊ El usuario debe haber entrado al sistema.

**Flujo:**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Da clic en el botón cerrar	2	Se cierra la aplicación de Excel	E1

**Excepciones:**

Id	Nombre	Acción
E1	Se ingresaron datos y no se guardo el Libro Excel	La plantilla envia un mensaje preguntando si desea o no guardar el archivo

**Postcondiciones:**

- ◊ El sistema termina

**Plan de prueba del sistema:**

<b>Caso de Uso</b>	<b>Entrada</b>	<b>Esperado</b>	<b>Obtenido</b>
Salir	Dar clic en el icono cerrar	Terminar la ejecución del sistema	El sistema terminó
	Dar clic en el icono cerrar	No termina la ejecución del sistema	El programa Excel muestra un mensaje preguntando si desea guardar los cambios a la plantilla

## 4.4. Diseño

En esta parte de diseño de la aplicación se han utilizado herramientas como : **diagrama de estados**, **diagrama de clases**, **diagrama de secuencia** y **diagrama de bloques** para ver el posible funcionamiento de la aplicación.

### 4.4.1. Diagrama de Clases

El Diagrama de Clases sirve para poder representar las posibles clases que puede contener un sistema. Son abstracciones de clases directamente utilizables en la implementación[FLFJ2005].

Para obtener el diagrama de clases de la aplicación, primero se identifican las clases que ésta puede tener respecto a los casos de uso. Aquí se obtendrá en relación al caso de uso general. Las clases que se encuentran son:

- CrearDocumentos, puede crear dos tipos de documentos: el libro Excel y el documento XML.
- CapturarDatos, se necesita para el ingreso de información necesaria para la creación de documentos.
- ValidarDatos, validará la información a capturar y la información que requiere cada uno de los documentos a generar.

A continuación se identifican las relaciones de las clases anteriores.

- Generalizaciones

- CrearDocumentos se divide en dos tipos: DocumentoExcel y DocumentoXML.
- ValidarDatos se divide en dos tipos de validaciones: ValidarFechas y ValidarMetadatos.

- Asociaciones

- La creación de documentos, se asocian con la captura de datos, y con la validación de datos.

Cada una de las clase anteriores tiene definidos un conjunto de atributos y de métodos que permitirán que el sistema realice la tarea para lo que se desarrollará.

- CrearDocumentos, para poder crear documentos, se necesita información necesaria que el documento llevará,
  - DocumentoExcel: datos para crear un documento Excel.
  - DocumentoXML : datos para crear un documento XML.
- ValidarDatos, esta corresponde al tipo de validación
  - ValidarMetadatos: conjunto de metadatos a validar.
  - ValidarFechas: fecha a validar.

La representación del diagrama de clases correspondiente al diseño de la aplicación se muestra en la Figura 4.10,



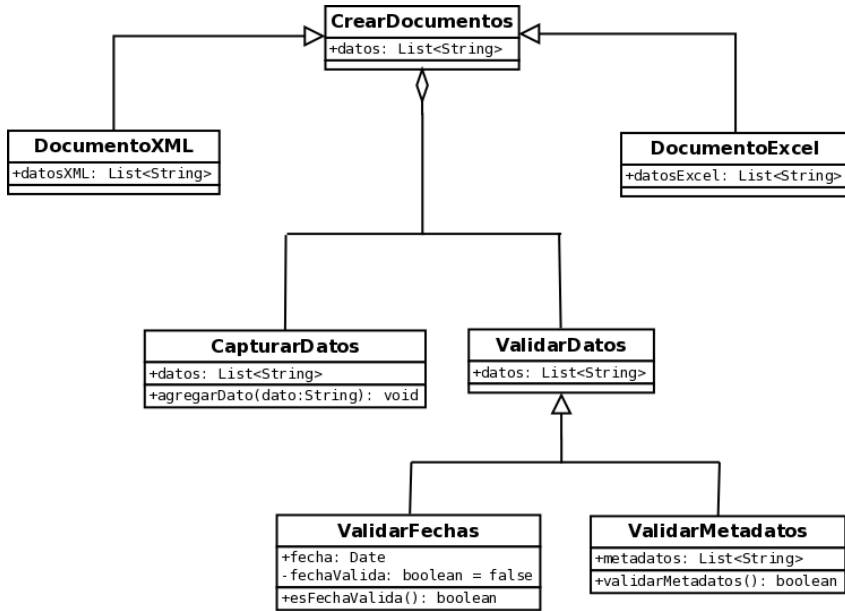


Figura 4.10: Diagrama de clases de la aplicación.

#### 4.4.2. Diagrama de Secuencia

El Diagrama de Secuencia muestra la interacción entre un conjunto de objetos de la aplicación a través del tiempo. En un modelo de secuencia :

1. Los objetos involucrados en la interacción están ordenados horizontalmente con una línea vertical vinculada a cada objeto.
2. El tiempo se representa verticalmente por lo que éste avanza hacia abajo sobre las líneas punteadas. Por lo tanto, en el modelo, la secuencia de operaciones se puede leer fácilmente.
3. Las interacciones entre los objetos se representan por flechas etiquetadas que vinculan a las líneas verticales. Estos *no* son datos que fluyen, sino que representan mensajes o eventos que son fundamentales para la

interacción.

- Los rectángulos delgados sobre la línea de vida del objeto representan el tiempo en el cual el objeto es el que tiene el control del sistema. Un objeto inicia el control en la parte superior de este rectángulo y libera el control a otro objeto en la parte inferior del rectángulo. Si existe una jerarquía de llamadas, el control no se libera hasta que se haya completado el último retorno a la llamada del método inicial.

El Diagrama de Secuencia correspondiente al diseño de la aplicación se puede observar en la Figura 4.11

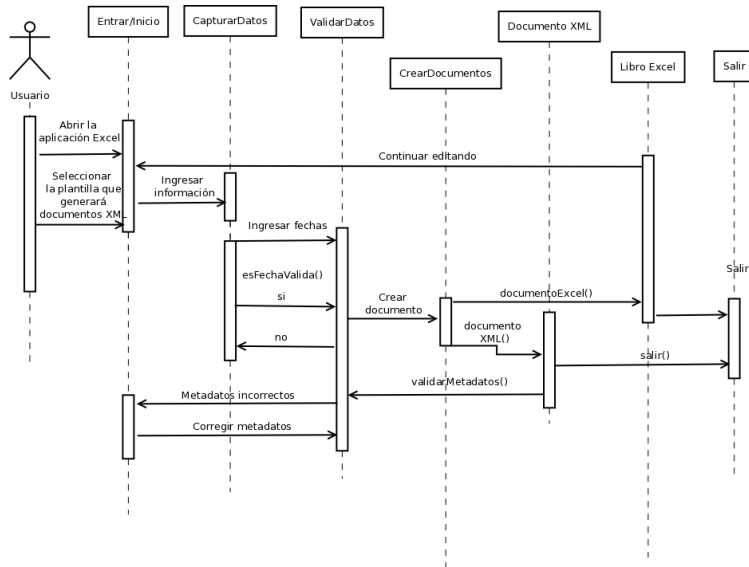


Figura 4.11: Diagrama de Secuencia

### 4.4.3. Diagrama de Estados

Los diagramas ayudan a describir el comportamiento normal de un sistema. El Diagrama de Estados de un sistema supone que en cualquier momento el sistema está en uno de varios estados posibles. Cuando recibe un estímulo, éste dispara una transición a un estado diferente. En la Figura 4.12 se muestra el diagrama de estados correspondiente a la aplicación, se obtuvo a partir de los casos de uso y de los diagramas de clase. El Cuadro 4.1 describe el funcionamiento del diagrama.

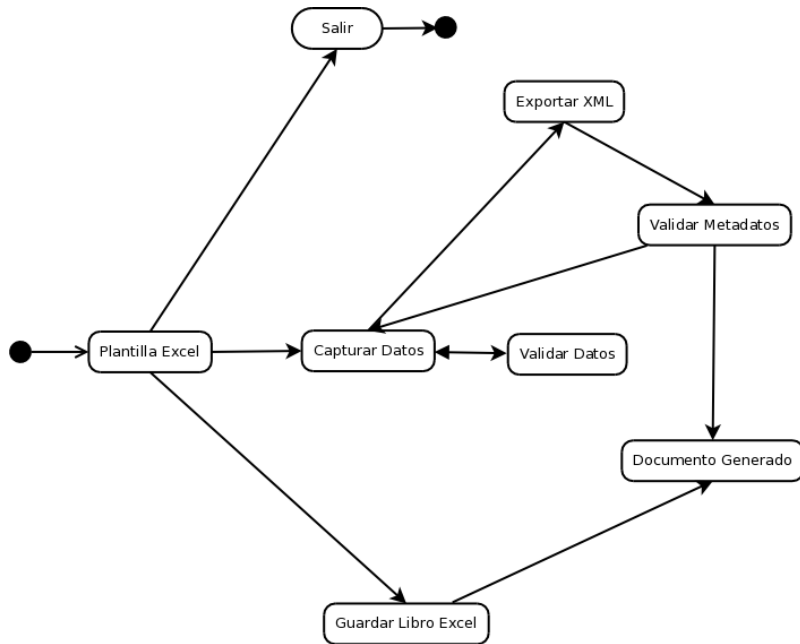


Figura 4.12: Diagrama de Estados.

Estado	Descripción
Plantilla Excel	La plantilla está en espera de datos de entrada. La flecha sin etiqueta con la bola negra indica que “Plantilla Excel” es el estado inicial.
Crear Documento Exel	El estado espera la instrucción de guardar, para crear un Libro Excel basado en la plantilla para su posterior edición.
Salir	El estado espera la instrucción de salir, termina el flujo de la aplicación.
Documento Generado	El estado ha generado algún documento, Libro Excel o Documento XML.
Capturar Datos	El estado va almacenado los datos en la plantilla.
Validar Datos	El estado “Validar Datos” se activa si los datos(fechas) de entrada son inválidos, mostrando algún mensaje de advertencia y regresando al estado “Capturar Datos”, para el ingreso de datos correctos.
Crear Documento XML	El estado se encarga de generar el documento XML, si cumple con las validaciones correspondientes.
Validar Metadatos	El estado revisa que se encuentren llenos los campos que aceptan metadatos, si no es así manda un mensaje de advertencia y regresa al estado “Capturar Datos” para el ingreso de los campos faltantes.

Cuadro 4.1: Flujo Diagrama de Estados

#### 4.4.4. Diagrama de Bloques

Representa de forma gráfica la estructura de un sistema, y la relación entre la entrada y la salida del sistema, e indica la secuencia del proceso del sistema.

La Figura 4.13 muestra el Diagrama de Bloques correspondiente al sistema que será generado.

El diagrama indica que si se recibe como entradas un conjunto de metadatos y un esquema XML el sistema internamente generará un Documento XML bien formado y válido o un Libro Excel, o ambos.

Durante mi estancia en la carrera, me enseñaron varias metodologías de programación, como programación funcional( Matemáticas Discretas, y en Len-

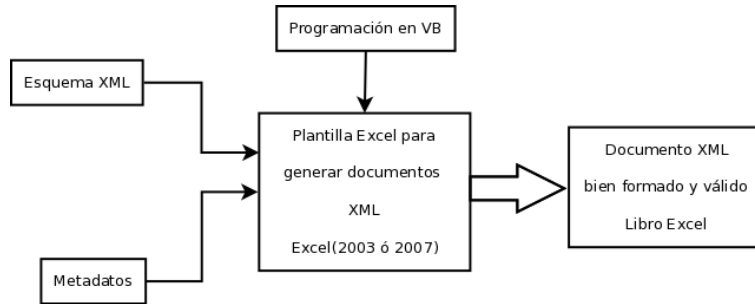


Figura 4.13: Diagrama de Bloques

guajes de Programación), programación orientada a objetos(Introducción a las Ciencias de la Computación 1 y 2), programación lógica(Inteligencia Artificial), etc., de las cuales en la que más enfatizaron fue en la programación orientada a objetos, lo cual generó que aprendiera más sobre este lenguaje. Además en la materia de Ingeniería de Software donde nos enseñan el proceso que se debe seguir para la creación de sistemas computacionales, también estuvo dirigido a la programación orientada a objetos. Por lo tanto la realización del diseño de la aplicación, la desarrolle en función a lo aprendido en la carrera, es decir, orientado a objetos. El proceso de reunir los requerimientos del sistema, es un proceso en donde se debe tener una gran atención hacia el cliente, para entender lo que necesita.

A veces los clientes no tienen muy bien definido la aplicación que necesitan, conocen el problema general, pero al momento de especificarlo van surgiendo más ideas, más requerimientos, lo que implica que el desarrollo del diseño de la aplicación se vaya extendiendo, y el tiempo en iniciar el desarrollo, en comenzar a programar la aplicación se retrase.

Llegando a este punto donde se han comprendido los requerimientos de la aplicación y se ha establecido un diseño, se proceda a implementar la solución de la mejor manera posible. En el siguiente capítulo se describirá el proceso de desarrollo que se siguió para implementar la aplicación.

---

## Capítulo 5

# Desarrollo de la Solución Propuesta

Presentando el diseño de la solución al cliente e indicándole que se iba a crear la aplicación desde cero en un lenguaje de programación Orientado a Objetos (POO), no se convenció del todo, ya que argumentó que quería la aplicación basada en una ya conocida por los usuarios, para que les resultara familiar, y al mismo tiempo se les facilitara el aprendizaje del uso de la aplicación.

En el mundo laboral generalmente se utiliza el Sistema Operativo Microsoft Windows en cualquiera de sus versiones, y por consiguiente los usuarios están familiarizados con el paquete de Microsoft Office.

Dados los argumentos del cliente, y tomándolos en cuenta, se acordó crear la aplicación sobre Microsoft Office Excel 2003, esto también por que la plantilla de NARSTO que se pretendía estandarizar está desarrollada en Excel y los usuarios que llenan dicha plantilla, como lo dijo el cliente ya están familiarizados con ella.

De esta forma se comenzó desarrollando la aplicación sobre una Hoja de

Cálculo dónde el usuario tiene conocimientos básicos de ella.

Una vez que se llegó al acuerdo de por dónde empezar a programar, pues ya solo era cuestión de extender una aplicación, dicho así suena muy sencillo, pero claro no lo es ¿por que?.

Microsoft proporciona un entorno de desarrollo que incluye dos lenguajes de programación, C# y Visual Basic, para la implementación de Microsoft Office que proporciona elementos de la programación orientada a objetos, ésta herramienta es conocida como Visual Studio Tools for Office (VSTO).

Como es de saberse las herramientas de Office no son gratuitas o de distribución libre, por lo que conseguir VSTO de momento no fué algo trivial, ya que no se contaba con los recursos necesarios. Investigando un poco en la página de la UNAM, tratando de conseguir algún descuento, como lo hacen algunas veces con Microsoft Office, se descubrió que se podía obtener varias herramientas de desarrollo para .NET, simplemente con ser alumno de la UNAM, y cumplir con algunos requisitos.

Una vez que se cumplieron con los requisitos necesarios para obtener una herramienta de desarrollo, se logró obtener la versión 2007 de VSTO. Para la configuración del entorno de desarrollo, de igual forma que el conseguir VSTO, no fué tan trivial.

Una vez que se configuró el entorno de desarrollo con la herramienta VSTO y dado que el diseño de la solución del problema se realizó orientado a objetos, se inició con la implementación de la solución. Los datos que en ese momento se emplearon para generar el árbol de elementos XML fue un conjunto de los datos que proporciona la plantilla de NARSTO, que hasta ese momento eran los metadatos a exportar, el conjunto es el siguiente:

- Quality Control Level.
- Date this File Generated/Archive Version Number.

- 
- Organization Acronym.
  - Organization Name.
  - Study or Network Acronym.
  - Study or Network Name.
  - File Contents Description Short/Long.
  - Principal Investigator Name –Last/First.
  - Principal Investigator Affiliation.
  - Co-Investigator Name–Last/First.
  - Co-Investigator Affiliation.
  - Country Code.
  - State or Province Code.
  - Sampling Interval as Reported in Main Table.
  - Sampling Frequency of Data in Main Table
  - Principal Investigator Contact Information.
  - Data Usage Acknowledgement.
  - Name and Affiliation of Person who Generated this File.
  - Date of Last Modification to Data in Main Table
  - File Change History–Version Number/Description
  - Name and Version of Software Used to Create this File
  - Standard Characters.



Durante el transcurso de la carrera aprendí el manejo de algunos lenguajes de programación de distribución libre. Por lo que sumergirme en éste nuevo lenguaje para mí fue toda una aventura.

VSTO proporciona dos lenguajes de programación orientados a objetos: C# y VB, por lo que se tomo la decisión de desarrollar en paralelo la solución, en ese momento C# no contaba con suficiente documentación tanto cómo VB, pero como es un lenguaje orientado a objeto y gracias a los conocimientos que se tienen de éste, se supuso que el programar sobre este lenguaje no sería complicado.

Por otro lado el programar en VB fué relativamente sencillo, ya que se contaba con suficiente documentación y ejemplos que me ayudaron en la programación de la aplicación, el hecho de que la programación bajo este lenguaje no fuera tan fácil, es que a pesar de que es un lenguaje orientado a objetos, al momento de programar éste tipo de programación no se ve tan claro.

Dado que iniciaba en estos lenguajes, se comenzó desarrollando en paralelo con pocos datos la solución, y que estos se enviaran a un documento en formato XML, sin pasar por la DTD o el XMLSchema.

Una vez que tuve una primera versión de como podría funcionar la aplicación, se la llevé al cliente para que le diera el visto bueno. Ejecutando la aplicación desde la máquina de desarrollo al cliente le pareció buena, el problema surgió cuando pidió que se instalara en una de las máquinas que la iba a utilizar.

Para la instalación de la solución creada en VSTO en los equipos de usuarios finales, se requería .NET Framework, instalar el motor en tiempo de ejecución para VSTO, instalar ensamblados de interoperabilidad primario de Office, y algunos requisitos previos para la instalación de cada uno de estos paquetes.

Se intento instalar la plantilla en una máquina con Windows 7, se configuró el ambiente para que la plantilla pudiera funcionar, pero al momento de

abrirla ésta perdía sus funcionalidades. Marcaba un error de compatibilidad por la falta de un paquete, para poder ejecutarse correctamente, se busco dicho paquete pero no se logró encontrar uno adecuado a la versión de Windos 7.

Esto al cliente le pareció un inconveniente, pues la plantilla no sería transportable a las distintas versiones de Windows o de Microsoft Office, por lo que se tomó la decisión de implementar la plantilla con el lenguaje de programación autocontenido que Microsoft Office ofrece, Visual Basic para Aplicaciones (VBA).

El lenguaje Visual Basic para Aplicaciones es un lenguaje orientado a eventos, es una herramienta de diseño de aplicaciones para Windows, en las que éstas se desarrollan en gran parte a partir del diseño de una interfaz gráfica.

Los pasos a seguir para realizar un programa bajo este lenguaje son los siguientes:

- **Creación de una interfaz de usuario.** Esta interfaz será la principal vía de comunicación hombre máquina, tanto para salida de datos como entrada. Será necesario partir de una ventana a la que se le iran añadiendo los controles necesarios.
- **Definición de las propiedades de los controles.** Objetos que se hayan puesto en el formulario. Estas propiedades determinarán la forma estática de los controles, es decir, como son los controles y para qué sirven.
- **Generación del código asociado a los eventos que ocurran a estos objetos.** A la respuesta a estos eventos (clic, doble clic, una tecla pulsada, etc.) que deberá generarse de acuerdo a las necesidades del programa.
- **Generación del Código del programa.** Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. Sin embargo Visual Basic ofrece la posibilidad de

establecer un código de programas separado de estos eventos. Éste código puede introducirse en unos bloques llamados Módulos, en otros bloques llamados Funciones, y otros llamados Procedimientos. Los procedimientos responden a un evento producido durante la ejecución del programa.

Una vez conocido un poco este lenguaje de nueva cuenta se tuvo que implementar la solución, aunque pudiera parecer sencillo trasladar la versión desarrollado en Visual Basic a Visual Basic para Aplicaciones (VBA). Pero no fue así pues aunque sean el mismo lenguaje la sintaxis varía entre las versiones del mismo.

Desarrollé de nueva cuenta una primera versión de la solución en VBA, se la presente al cliente y con esta quedo más convencido. Pero no en todo pues tenía duda en el conjunto de metadatos que se pretendían exportar. El cliente llegó a un acuerdo con sus colegas de que metadatos eran los apropiados a exportar, y éstos fueron los que me proporcionaron.

Los metadatos que debía contener el documento XML generado son:

#### ■ Intellectual Property

- **Creator.** Persona u organización responsable de la creación del contenido intelectual del recurso.
  - personal creator
  - corporate creator
- **Publisher.** Entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual.
- **Contributor.** Persona u organización que haya tenido una contribución intelectual significativa en la creación del recurso, pero secundaria en comparación a la contribución de las personas u organizaciones especificadas en el elemento Creator.

- **Rights.** Forma de manejar los derechos sobre términos y condiciones de acceso y uso del recurso.
- **Creation and Identity**
  - **Date.** Momento o período de tiempo asociado con el ciclo de vida del recurso. La fecha puede ser usada para expresar información temporal en cualquier nivel de granularidad.
  - **Type.** Naturaleza o género del recurso.
  - **Format.** Formato del archivo, medio físico o dimensiones del recurso. Dimensiones incluye el tamaño y duración.
  - **Identifier.** Secuencia de caracteres usados para identificar unívocamente un recurso.
- **Content**
  - **Title.** El nombre dado al recurso, usualmente dada por el autor.
  - **Subject.** Tópicos/temas asociados al contenido del recurso de información.
  - **Keywords.** Palabras significativas no controladas asociadas al contenido del recurso de información.
    - Theme Keywords
    - Place Keywords
  - **Description.** Descripción textual del recurso, un resumen o una descripción del contenido.
  - **Source.** Fuente de la cual se deriva el recurso descrito. Secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual. Puede derivar en parte o totalmente del recurso relacionado.
  - **Lenguaje.** Idioma(s) del contenido intelectual del recurso.

- **Relation.** Identificador de un recurso y su relación con el actual. Permite enlazar los recursos a través de sus descripciones y especificar el tipo de relación.
- **Coverage.** Cobertura espacial y/o temporal del contenido intelectual del recurso.
  - **Spatial Domain.** La cobertura espacial se refiere a una región física (por ejemplo, sector celestial); uso de coordenadas (por ejemplo, longitud y latitud) o nombres de lugares extraídos de una lista controlada.
    - ◇ Box.westLimitCoverage
    - ◇ Box.eastLimitCoverage
    - ◇ Box.northLimitCoverage
    - ◇ Box.southLimitCoverage
  - **Time period of content date.** La cobertura temporal se refiere al contenido del recurso.
    - ◇ Coverage.dateStart
    - ◇ Coverage.dateEnd
- **Spatial Data Information**
  - Data type
  - Data format
  - Projection
  - Scale

Los metadatos descritos anteriormente fueron proporcionados por el cliente, el núcleo básico de estos metadatos son:

- Dublin Core Espacial<sup>1</sup>

---

<sup>1</sup><<http://dublincore.org/>>

- FGDC 111915<sup>2</sup>
- MARC 21<sup>3</sup>

De los metadatos proporcionados se realizaron algunos cambios necesarios, que surgieron al momento de la implementación.

La DTD y esquema XML correspondiente a los metadatos se encuentran en el Apéndice B y Apéndice C, respectivamente.

Los nombres proporcionados a las etiquetas XML del documento generado quedaron en el idioma Inglés, esto con el fin de facilitar el intercambio de información con dependencias universitarias extranjeras y así lograr una mejor estandarización.

Para comenzar con el desarrollo definitivo de la solución, trate de simular la programación orientada a objetos. VBA simula la orientación a objetos de la siguiente forma:

El Editor de Visual Basic es un programa cuya ventana principal permite realizar una serie de acciones sobre una macro, propiedades de libro y hojas, así como el de crear nuevos módulos (procedimientos) y formularios. Aquí se pueden modificar las macros y extenderlas tanto como uno desee darle interactividad.

En un lenguaje de programación orientada a objetos, las instrucciones se basan en variables, constantes, objetos, propiedades de los objetos, etc., pertenecientes a un ámbito, a un entorno. Todos estos elementos, sean sus componentes o características, como sus relaciones entre otros objetos, están agrupados o “encapsulados” en los objetos.

---

<sup>2</sup><<http://www.fgdc.gov/> >

<sup>3</sup><<http://www.loc.gov/marc/marcspa.html> >

Los objetos Excel están constituidos por Celdas, Rangos, Hojas, Libros, Gráficos, Tablas, archivos externos, etc.

Los objetos refiriéndose a Excel, poseen cuatro características empleadas en la codificación de un programa:

Propiedades de los objetos, métodos para manipularlos, eventos que causan la manipulación de los mismos, y colecciones o clases a las que pertenecen los objetos.

### **Propiedades**

Las propiedades son las características, atributos, formas o aspectos del objeto, a las que se hace referencia mediante el uso de variables. Una propiedad de objeto común en Excel es su Nombre, que permite usarlo en las diferentes cosas que se hagan al objeto. De manera que una celda, un rango de celdas, una hoja, gráfico o tabla de Excel, tendrá un nombre con el cual se identificará.

Para hacer referencia a las propiedades mediante los programas en VBA, se usa la sintaxis:

*NombreDelObjeto.Propiedad*

Ejemplo:

Range("A5").Name Aquí se hace referencia al nombre del objeto Range de la celda A5.

### **Métodos**

Un método es una acción, un procedimiento, que tiene efectos sobre un objeto.

Ejemplo:

**Range(“A5”).Select** Aquí se ejecuta el método Select, que permite activar la celda A5 y ponerla en disponible para cualquier operación.

**Range(“B2:G4”).Formato** En este caso, se ejecuta el método Formato sobre el Rango B2:G4. Se supone que el método Formato ya está definido previamente.

### Eventos

Un evento es el resultado de una acción, es la forma cómo queda el objeto después de alguna acción sobre él. Por lo general estas acciones son producidas por los métodos que actúan sobre el objeto.

En la versión 2003 de Excel, se pueden encontrar algunos eventos como:

- *Se seleccionó una celda o rango.*
- *Se seleccionó una hoja (haciendo clic en la pestaña o etiqueta de la misma).*
- *Se ha abierto o cerrado un libro.*

Excel incluye varios controladores de eventos, o rutinas de código, que controlan acciones determinadas. Cuando ocurre una de dichas acciones, y ha comunicado a Excel qué desea que haga cuando ocurra el evento, Excel ejecuta el código del controlador de eventos.

### Colecciones

Una colección es un grupo o conjunto de objetos contenidos en otro objeto cuyas propiedades son comunes a los objetos componentes.

Puesto que un libro contiene una o más hojas de cálculo, se puede decir que un libro es una colección de hojas de cálculo. Así se puede ejecutar algún método sobre esta colección a fin de realizar la misma acción sobre todas ellas.



## 5.1. Desarrollo

Una vez que tuve más claro el como iba a trabajar con VBA, haber estudiado un poco el funcionamiento del lenguaje y teniendo el conjunto de metadatos y las validaciones que se requerían para cada uno de éstos, inicié con el desarrollo de la aplicación.

Como mencione en parrafos anteriores, VBA se basa principalmente en eventos relacionados a alguna interfaz, por lo tanto comencé por definir la interfaz a utilizar, ésta la realicé con un hoja de Excel, tomando las primeras dos columnas, una para la descripción del dato y la otra para capturar su valor. Figura 5.1

Posteriormente se definieron los elementos de control a utilizar en la plantilla, como un botón que es el que realiza la función principal, algunos calendarios (Control Calendar) para seleccionar fechas, y algunos cuadros de selección (checkBox) Figura 5.2

Posteriormente se comenzó por darle funcionalidad a los elementos de control, algunos de ellos asignándolos a algunas celdas específicas, por ejemplo para las fechas, para que al momento de seleccionar una celda que admitiera un valor de tipo Date, el Control Calendar se mostrara para poder seleccionar la fecha.

Para desarrollar el caso de uso principal que es el que exporta los datos a XML, inicié por buscar alguna biblioteca o paquete que proporcionara VBA para facilitar la creación de la estructura del árbol de elementos que conformaría el documento XML.

Una vez encontrado el paquete (Microsoft XML V3.0), desarrollé un procedimiento que obtuviera los valores de la hoja Excel, para comenzar a darle forma al documento XML, esta biblioteca va creando un árbol XML, crea un

	A	B	C	D
1	OBLIGATORIOS	OPCIONALES	OPCIONALES OBLIGATORIOS	
2		ExportarXML		
3				
4		***** DATOS DE IDENTIFICACION*****		
5	Título			
6	Palabras Clave del Tema			
7	Palabras Clave del Lugar			
8	Descripción	del calentamiento global en la atmosfera		
9	Fuente			
10	Idioma	<input checked="" type="checkbox"/> Español <input type="checkbox"/> Inglés <input type="checkbox"/> Francés		
11	Relación	<input type="checkbox"/> Es versión de <input type="checkbox"/> Reemplaza(o) <input checked="" type="checkbox"/> Tiene una versión <input checked="" type="checkbox"/> Es requerido <input type="checkbox"/> Es reemplazado por <input type="checkbox"/> Requiere	<input type="checkbox"/> Es parte de <input type="checkbox"/> Refiere a <input type="checkbox"/> Tiene una parte <input type="checkbox"/> Es referido por	
12		INFORMACION DE LA PUBLICACION		
13		*****AUTOR*****		
14	Personal			
15	Corporativo			
16		*****COLABORADOR*****		
17	Persona			
18	Organización			
19		*****EDITOR*****		
20	Editor			
21		*****COBERTURA*****		
22		PERIODO DE TIEMPO DEL CONTENIDO		
23	Fecha de Inicio			
24	Fecha de Termin			
25		DOMINIO ESPACIAL		
26	Coordenada Oeste			
27	Coordenada Este			
28	Coordenada Norte			
29	Coordenada Sur			
30	Sistema de coordenadas geográficas			
31	Sistema de coordenadas proyectadas			
32	Escala de los Datos			
33		*****INFORMACION DE DATOS ESPACIALS*****		
34	Tipo de Datos			
35	Formato de Datos			
36	Proyección de Datos			
37	Escala de Datos			
38		*****ACCESO Y USO DE LA INFORMACION*****		
39	Restricciones de acceso			
40	Restricciones de uso			
41		*****CREACION E IDENTIDAD*****		
42	Tipo de contenido			
43	Formato de Datos			
44	Fecha de Publicación			
45	Fecha de Actualización			

Figura 5.1: Campos - Valor

nodo raíz, y se van agregando tantos nodos como uno desee. Este paquete también proporciona una función que escribe la estructura XML generada a un archivo con tan sólo pasarle el nombre del archivo.

En el proyecto Visual Basic la función principal que exporta los datos a XML es el procedimiento o subrutina *generaXml(ruta As String)* la cual se encarga de generar el documento XML, y se le pasa como parámetro la ruta en donde el nuevo documento XML será guardado.

La biblioteca de Microsoft que permite la generación de elementos XML es

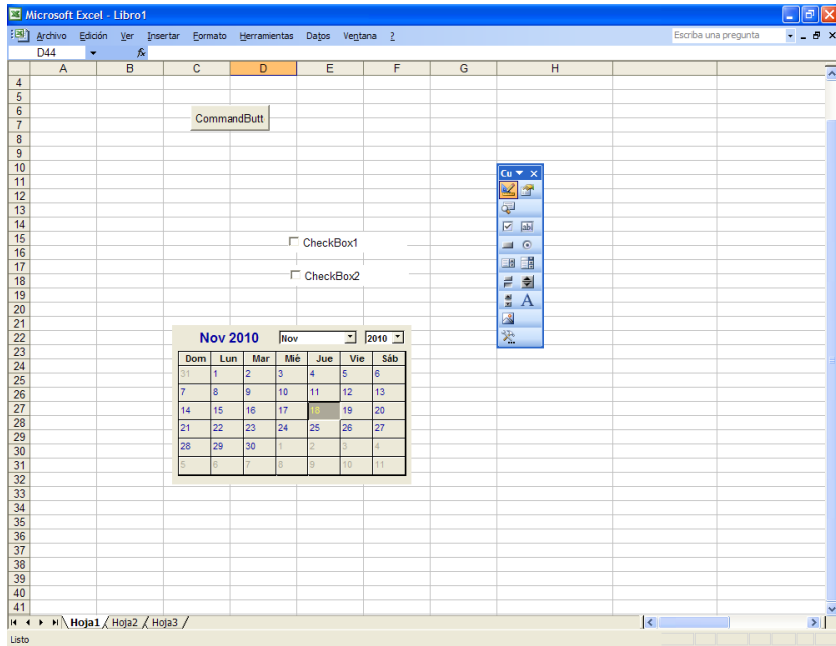


Figura 5.2: Elementos de Control

*Microsoft XML*, para la versión de Excel 2003 del proyecto se utilizó la versión 6.0 El código que se generó tiene la siguiente estructura :

...

- ' Creando la informacion del documento raiz
  - Set oNode = oDoc.createProcessingInstruction("xml",
  - "version='1.0' encoding ='UTF-8' ")
  - Set oNode = oDoc.InsertBefore(oNode, oDoc.ChildNodes.Item(0))
- ' Creando el nodo inicial del documento
  - Set oRoot = oDoc.createElement("metadata")
  - Set oDoc.DocumentElement = oRoot
- ' Agregando nodos a el documento

```

Set oEle = oDoc.createElement("Identification")
Set oNode = oDoc.createElement("title")
oNode.Text = metadatos.Range("Titulo").Value
oEle.appendChild oNode

```

...

Esta función se llama a partir del evento que se crea al hacer clic sobre el botón del formulario, tal evento también esta contenido en una subrutina *ExportarXML()*, que es la que manda a llamar a las funciones principales para la generación del documento XML El código de esta subrutina es:

```

'Subrutina que integra varios funciones para que se
'puedan validar y guardar los elementos
'de la hoja excel en un documento xml
Sub ExportarXML()
Dim quien As String
Dim ruta1 As String
quien = verifica()
If ((IsEmpty(quien)) Or Not (quien = "")) Then
MsgBox "Falta el campo" & vbCr + quien, vbExclamation
+ vbOKOnly, "No se puede guardar"
Else
ruta1 = ruta()
If Not (ruta1 = "") Then
generaXml ruta1
Else
Exit Sub
End If
End If
End Sub

```

La función *verifica()* es la que válida que los campos obligatorios de la

plantilla no se encuentren vacíos, y es el procedimiento que implementa el caso de uso “validar metadatos”.

La implementación de esta función fue la que me llevó un poco más de tiempo. Del conjunto de metadatos que se proporcionó, existe un subconjunto de éste que es más importante, y es necesario para poder generar el archivo XML. Por lo tanto comencé por validar que las celdas que capturaban los datos de éste subconjunto no estuviera vacío, en caso de que alguno de estos campos estuviera vacío, al momento de intentar exportar los datos, se enviaría un mensaje de advertencia, indicando cuales campos eran obligatorios, y que les faltaba por ingresar.

Los datos de la plantilla, se identifican por un color, que indican si son obligatorios u opcionales, o los que son obligatorios opcionales.

Los datos obligatorios opcionales, son subconjuntos de datos, este tipo de datos están relacionados entre sí, es decir que si el usuario captura un dato que pertenezca a uno de estos subconjuntos estos se vuelven automáticamente obligatorios, por lo tanto debe capturar todos los datos relacionados al subconjunto o no capturar ninguno, en caso de que falten datos, se enviarán junto a la lista de validación de los metadatos obligatorios, los datos faltantes.

El validar este tipo de datos hace que se generen muchas condicionales en el código generado y hasta se vuelven complejas. En el caso de los datos opcionales no se encuentra problema alguno.

Otra validación importante que se encuentra en la aplicación, se presenta al seleccionar las fechas. La aplicación cuenta con tres campos de fechas importantes: Fecha de Inicio, Fecha de término y la Fecha de publicación.

La validación que se lleva a cabo para las fechas es: la fecha de inicio debe ser menor igual a la fecha de término, y la de término menor igual a la de

publicación, y la de publicación menor igual a la fecha actual. Por lo que la fecha inicio será la fecha límite inferior y la fecha actual la fecha límite superior.

Esta validación se realiza al momento de ingresar la fecha, cuando la fecha que se ingresa es la primera, sólo se debe validar con respecto a la fecha actual.

Cuando ya se ha ingresado alguna fecha, la siguiente fecha a ingresar se debe validar con respecto a ésta y a la fecha actual, por ejemplo, si la primera que se ingresó fue la fecha de inicio, ésta será la fecha límite inferior para las otras, y si la otra fecha ingresada es menor a la fecha de inicio o mayor a la fecha actual, se envía un mensaje de advertencia indicando cual fecha es la incorrecta, y que la debe actualizar.

Como mencioné anteriormente Excel y VBA trabajan mediante eventos, por lo que el evento que realiza la validación se lleva a cabo al momento de seleccionar la fecha en el Control Calendar, por lo que la validación de fechas se realiza al momento de ir capturando este tipo de datos.

También se creó una función para capturar la ruta y el nombre del archivo XML, de donde se guardará. La función desarrollada para realizar tal tarea se llama ruta() la cual proporciona elementos necesarios para poder capturar los datos de la ruta y el nombre del archivo.

La implementación del código se dividió en módulos, que es lo que VBA ofrece para no hacer tan extenso la generación de éste en un sólo archivo. Estos módulos son los que simularían el uso de clases en la programación orientada a objetos. VBA está basada en objetos pero no es un lenguaje orientado a objetos.

Lo mencionado en éste capítulo principalmente, son las herramientas que se utilizaron para el desarrollo de la aplicación en Excel 2003 y los pasos que seguí al implementar algunos de los casos de uso principales. El código fuente y la aplicación se pueden revisar en el CD anexo.

Para la creación de la plantilla 2007 uno podría pensar que una vez que se ha generado la versión 2003, ésta puede ser un subconjunto de las versiones posteriores, y por lo tanto sólo sería cuestión de adaptar la aplicación 2003 a estas nuevas versiones. Dicho de ésta forma suena muy sencillo, pero pues no fue así ya que me encontré con ciertos problemas de compatibilidad entre las versiones.

Cuando uno abre la aplicación con alguna versión compatible, no causa tantos problemas, pues generalmente sólo se encuentran errores de diseño. Pero como la aplicación es una plantilla de Excel, cuando ésta se trata de importar a un Libro Excel es cuando pierde funcionalidad, ya que pierde las referencias de los paquetes que hacen posible el funcionamiento de la aplicación, además que las versiones de éstas referencias no son compatibles con las nuevas versiones, lo descrito anteriormente se probó con la versión de Microsoft 2007.

Por lo tanto se tuvo que crear la aplicación para la versión de Excel 2007, los pasos que se siguieron relativamente fueron similares, pues las bibliotecas que se utilizan para ésta nueva versión tienen algunas diferencias.

La aplicación 2003, se probó en una máquina con sistema operativo Apple con la versión 2003 de Microsoft Office compatible para Apple, se esperaba que la aplicación funcionara de igual forma que el sistema operativo Windows, pero no fue así pues en este entorno la aplicación definitivamente no levanta.

También probé la aplicación con la versión 2010 de prueba de Microsoft Office, y generó el mismo error que cuando se probó con la máquina de Apple. El error producido, marcaba que hacía falta un `.ddl`, para poder continuar. Pero como no se tenía ni la versión de Microsoft 2010 ni la de Apple, este error no se llegó a analizar a fondo.

Por último probé la aplicación en una hoja de cálculo de distribución libre, y este caso igual le quito toda funcionalidad que la aplicación pudiera

tener, dejándola en modo de diseño y en ésta versión hasta el diseño quedo descuadrado.

Por lo tanto no se puede afirmar que las veriones de Microsoft Office, funcionen 100% como subconjuntos de versiones posteriores, pues aunque la diferencia entre las versiones de los controladores que utilizan estas aplicaciones son mínimas, si afectan al funcionamiento del sistema.



---

## Capítulo 6

# Conclusiones

En base a la investigación que se realizó sobre el intercambio de información, metadatos y XML, y el problema planteado al inicio de la presente, de crear un sistema para adaptar la plantilla de NARSTO a las necesidades del Centro de Ciencias de la Atmósfera de la UNAM, que capture, valide y exporte datos a un formato XML, con la intención de que el intercambio de datos, sea más fiable en cuanto a la pérdida de información se desarrolló el sistema que resuelve el problema planteado, con el fin de observar el comportamiento del intercambio de información.

El sistema se desarrolló para el Sistema Operativo Windows, por la necesidad del cliente y por que la mayoría de usuarios utilizan dicho sistema, por ser uno de los más comerciales.

También por la necesidad de que el sistema fuera sencillo para los usuarios, y que al mismo tiempo estuvieran familiarizados con él, para facilitar la capacitación del uso del sistema.

Esto me llevó a crear el sistema en el entorno de desarrollo que el paquete de Microsoft Office ofrece, al mismo tiempo que el cliente pidió que se realizara sobre una plantilla de Excel.

Esta decisión generó algunos conflictos al momento de la programación, pues como mencione en el capítulo 5, el lenguaje proporcionado por el paquete de Microsoft Office es Visual Basic, hasta ése momento desconocido para mí.

Otros inconveniente que me encuentre fué que el análisis y diseño realizado para la solución del problema se hizo orientado a objetos, y Visual Basic no es un lenguaje orientado a objetos. Lo cual fué aumentando la complejidad al desarrollo del sistema.

Se comenzó por desarrollar el sistema para la versión de Excel 2003. Pero al momento de realizar pruebas del sistema, algunos de los usuarios finales, no tenían esta versión de Excel, por lo que se tuvo que generar también el sistema para la versión de Excel 2007.

En algún momento se mencionó que ya que se tenía la versión 2003 de la aplicación, y la otra versión a implementar era posterior, podría solo adaptarse a esta nueva versión, pues debería ser un subconjunto de versiones posteriores. Primero se realizaron pruebas para ver si esto era posible, pero se llegó a la conclusión de que no puesto que en las pruebas realizadas, no fueron del todo exitosas. Lo cual dio paso a que se desarrollara la versión 2007 de la aplicación, pero ya en este punto el desarrollo fue más fluido.

Por otra parte el hecho de generar un sistema para dos versiones hace que en general el desarrollo sea más lento, por ejemplo en la versión 2007 se proporcionan más herramientas para la creación del sistema, pero al mismo tiempo como es una versión más reciente hay menos documentación que para la versión 2003, pero ésta última versión no tiene las herramientas que tiene la versión 2007 o viceversa, puede si tenerlas, pero la implementación funciona de distinta manera.

A pesar de todos los obstáculos que surgieron durante el desarrollo del sistema, éste se terminó satisfactoriamente, y se encuentra funcionando con algunos miembros del Centro de Ciencias de la Atmósfera de la UNAM.

Lo que se hizo en esta investigación se puede ampliar a cualquier tipo de archivo, para poder así lograr en la mayor parte posible estandarizar los diferentes tipos de documentos.

Tal vez en un futuro se podría desarrollar una herramienta que pueda generar documentos XML dado un conjunto de metadatos, que esté desarrollado en un lenguaje compatible para todos los sistemas operativos, que no necesite un paquete específico para la captura de datos y así poder lograr la estandarización de los distintos tipos de archivos que se encuentran en el mundo de la tecnología de la información.

Que la pérdida de información sea reducida con ayuda de la definición de metadatos y estructurados en documentos XML, los cuales hasta el momento han demostrado ser compatibles con distintas tecnologías de información.

---

# Apéndice A

# Manual de Usuario

PLANTILLA PARA EXPORTAR

METADATOS EN FORMATO XML

VERSIÓN PARA MICROSOFT EXCEL 2003

## A.1. Introducción

La plantilla P-Metadatos2003.xlt es una herramienta que permite capturar, validar y exportar metadatos a un formato XML, para el intercambio de información.

El objetivo que se persigue con la aplicación del presente manual es:

- Dar a conocer a los usuarios finales las características y las formas de funcionamiento de la plantilla.

## A.2. Requerimiento Básico

- Computadora Pentium IV, memoria RAM 512M.
- Windows XP, Windows 7
- Tener instalado un navegador de internet, como por ejemplo: Internet Explorer versión 4 ó superior, Firefox, Mozilla, etc.
- Tener instalada una versión de Excel de Microsoft Office 2003

### A.2.1. Instalación de la Plantilla

La plantilla se puede guardar en cualquier carpeta local que se necesite.

Si requiere importar la plantilla a un libro Excel se debe guardar en la carpeta de plantillas de Microsoft Office.

## A.3. Cómo utilizar la plantilla

La plantilla es una plantilla Excel que esta diseñada para poder capturar datos, estos pueden ser obligatorios, opcionales y opcionales obligatorios, los cuales se identifican por un color distinto, estos campos se deben llenar con datos que proporcionen la información requerida. La plantilla se puede ver en la Figura A.1.

The image shows a screenshot of a metadata form application window. The window title is 'E39' and it has a menu bar with options: Archivo, Edición, Ver, Insertar, Formato, Herramientas, Datos, Ventana, and 2. The form is organized into columns labeled A, B, and C, and rows numbered 1 through 45. The form is divided into several sections:

- Row 1:** OBLIGATORIOS (pink background)
- Row 2:** OPCIONALES (blue background)
- Row 3:** OPCIONALES OBLIGATORIOS (purple background)
- Row 4:** \*\*\*\*\* DATOS DE IDENTIFICACION\*\*\*\*\* (green background)
- Row 5:** Título (pink background)
- Row 6:** Palabras Clave del Tema (blue background)
- Row 7:** Palabras Clave del Lugar (blue background)
- Row 8:** Descripción (pink background) with the text 'del calentamiento global en la atmosfera'.
- Row 9:** Fuente (pink background)
- Row 10:** Idioma (pink background) with radio buttons for Español (checked), Inoles, and Francés.
- Row 11:** Relación (blue background) with checkboxes for 'Es versión de', 'Tiene una versión', 'Es reemplazado por', 'Reemplaz(a)', 'Es requerido', and 'Requiere'.
- Row 12:** INFORMACION DE LA PUBLICACION (green background)
- Row 13:** \*\*\*\*\*AUTOR\*\*\*\*\* (green background)
- Row 14:** Personal (pink background)
- Row 15:** Corporativo (pink background)
- Row 16:** \*\*\*\*\*COLABORADOR\*\*\*\*\* (green background)
- Row 17:** Persona (purple background)
- Row 18:** Organización (purple background)
- Row 19:** \*\*\*\*\*EDITOR\*\*\*\*\* (green background)
- Row 20:** Editor (pink background)
- Row 21:** \*\*\*\*\*COBERTURA\*\*\*\*\* (green background)
- Row 22:** PERIODO DE TIEMPO DEL CONTENIDO (green background)
- Row 23:** Fecha de Inicio (pink background)
- Row 24:** Fecha de Termino (pink background)
- Row 25:** DOMINIO ESPACIAL (blue background)
- Row 26:** Coordenada Oeste (purple background)
- Row 27:** Coordenada Este (purple background)
- Row 28:** Coordenada Norte (purple background)
- Row 29:** Coordenada Sur (purple background)
- Row 30:** Sistema de coordenadas geográficas (purple background)
- Row 31:** Sistema de coordenadas proyectadas (purple background)
- Row 32:** Escala de los Datos (purple background)
- Row 33:** \*\*\*\*\*INFORMACION DE DATOS ESPACIALES\*\*\*\*\* (green background)
- Row 34:** Tipo de Datos (pink background)
- Row 35:** Formato de Datos (blue background)
- Row 36:** Proyección de Datos (blue background)
- Row 37:** Escala de Datos (purple background)
- Row 38:** \*\*\*\*\*ACCESO Y USO DE LA INFORMACION\*\*\*\*\* (green background)
- Row 39:** Restricciones de acceso (blue background)
- Row 40:** Restricciones de uso (blue background)
- Row 41:** \*\*\*\*\*CREACION E IDENTIDAD\*\*\*\*\* (green background)
- Row 42:** Tipo de contenido (pink background)
- Row 43:** Formato de Datos (pink background)
- Row 44:** Fecha de Publicación (pink background)
- Row 45:** Fecha de Actualización (pink background)

Figura A.1: P-Metadatos2003

La plantilla proporciona información de ayuda para llenar alguno de los campos, como se muestra en la Figura A.2, dicha información se muestra al pasar el cursor sobre el campo a llenar.

Para poder exportar los datos a un formato XML, es necesario dar click en el botón ExportarXML que se muestra en la Figura A.3.

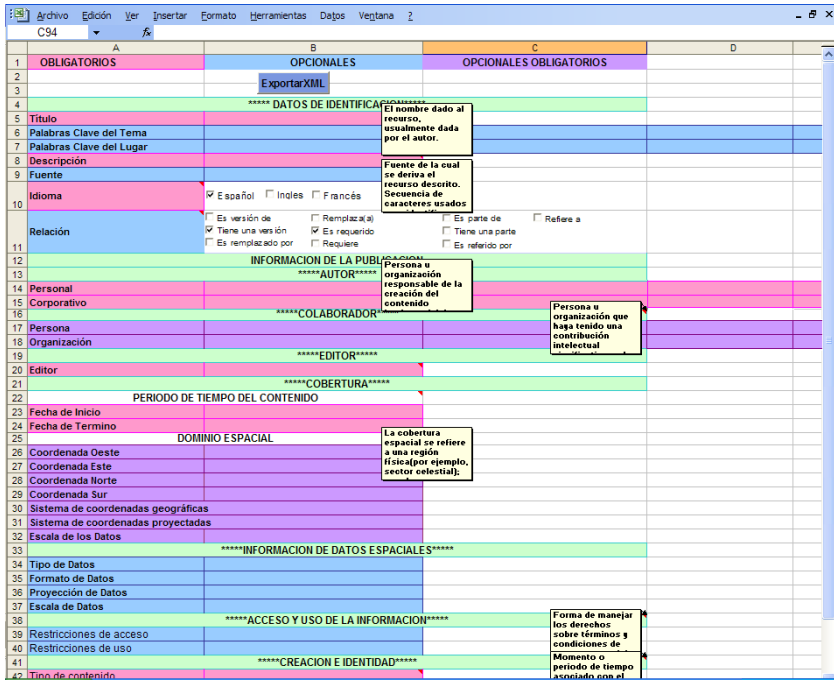


Figura A.2: Comentarios de Ayuda

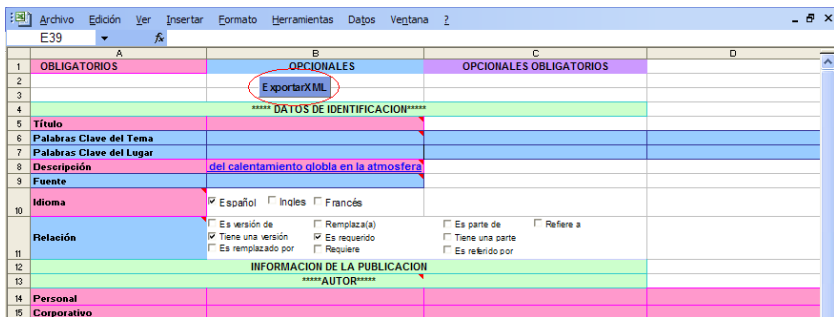


Figura A.3: Exportar XML

Para poder exportar los datos es necesario llenar los campos obligatorios, de lo contrario marcará un mensaje de advertencia, indicando los campos obligatorios faltantes. Como se puede ver en la Figura A.4.

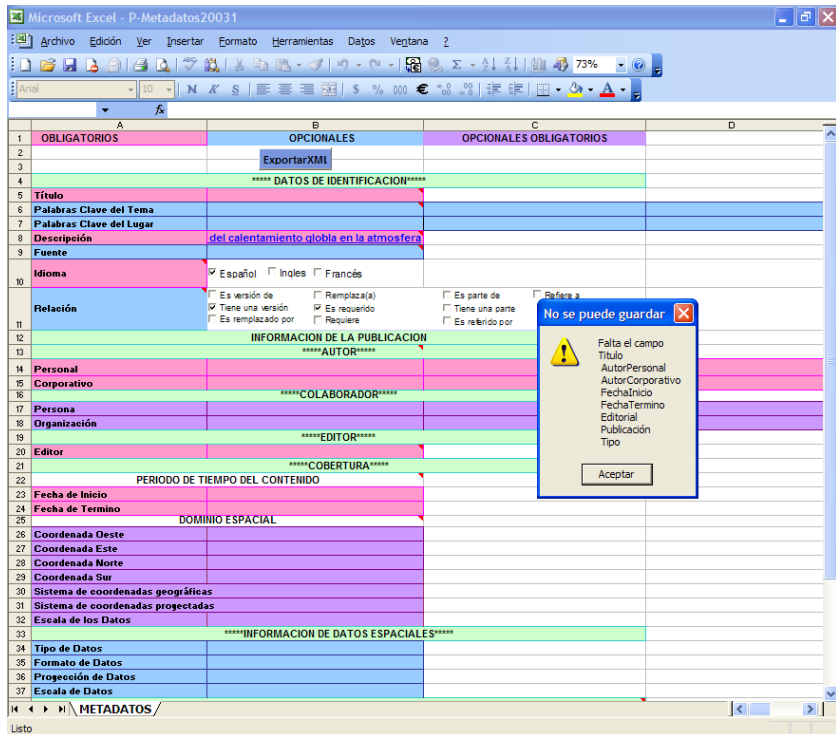


Figura A.4: Mensaje de Advertencia

### A.3.1. Datos de Identificación

La Figura A.5 muestra los campos correspondientes a la sección *Datos de Identificación*



	A	B	C
1	OBLIGATORIOS	OPCIONALES	OPCIONALES OBLIGATORIOS
2		ExportarXML	
3			
4	**** DATOS DE IDENTIFICACION****		
5	Título		
6	Palabras Clave del Tema		
7	Palabras Clave del Lugar		
8	Descripción	del calentamiento global en la atmosfera	
9	Fuente		
10	Idioma	<input checked="" type="checkbox"/> Español <input type="checkbox"/> Inglés <input type="checkbox"/> Francés	
11	Relación	<input type="checkbox"/> Es versión de <input type="checkbox"/> Reemplaza(a) <input checked="" type="checkbox"/> Tiene una versión <input checked="" type="checkbox"/> Es requerido <input type="checkbox"/> Es reemplazado por <input type="checkbox"/> Requiere	<input type="checkbox"/> Es parte de <input type="checkbox"/> Refiere a <input type="checkbox"/> Tiene una parte <input type="checkbox"/> Es referido por

Figura A.5: Datos de Identificación

La descripción de los datos que se observan en la Figura A.5 es la siguiente:

- **Título** – Es un campo obligatorio y sólo está marcada una celda con su correspondiente color, es decir que necesita un dato para llenar la celda.
- **Palabras Clave del Tema y Palabras Clave del Lugar** – Corresponden a campos opcionales, estos se deben llenar con 0 o mas datos. Por ejemplo existen 4 Palabras Clave del Tema, el dato 1 corresponde a la columna B, el dato2 a la columna C, el dato 3 a la columna D, etc. De igual forma con el campo Palabras Clave del Lugar.
- **Descripción** – Es un campo obligatorio y sólo esta marcada una celda con su correspondiente color, es decir que necesita un dato para llenar la celda.
- **Fuente** – Campo opcional, este campo permite 0 ó 1 dato.
- **Idioma** – Campo obligatorio, en este campo se debe seleccionar el idioma o idiomas que se requieran, la plantilla está predeterminada con el idioma Español
- **Relación** – Campo opcional, de las opciones que se muestran se pueden seleccionar 0 ó más opciones.

### A.3.2. Información de la publicación

La Figura A.6 muestra los campos correspondientes a la sección *Información de la publicación*

12	INFORMACION DE LA PUBLICACION		
13	*****AUTOR*****		
14	Personal		
15	Corporativo		
16	*****COLABORADOR*****		
17	Persona		
18	Organización		
19	*****EDITOR*****		
20	Editor		

Figura A.6: Información de la Publicación

La descripción de los datos es la siguiente:

#### Autor, Colaborador y Editor

- **Personal y Corporativo**- Campos obligatorios, se deben tener 1 ó más datos.
- **Persona y Organización**- Campos obligatorios opcionales, puede tener 0 ó más datos. Este tipo de campos se maneja por grupo. Es decir que si se llena una celda del campo Persona, entonces se debe llenar al menos una celda del campo Organización.
- **Editor** - Campo obligatorio, requiere de un dato.

### A.3.3. Cobertura

La Figura A.7 muestra los campos correspondientes a la sección de *Cobertura*

La descripción de los datos es la siguiente:

#### Periodo de Tiempo del Contenido y Dominio Espacial.

- **Fecha de Inicio y Fecha de Terminó**- Campos obligatorios, requieren de un sólo dato.

*****COBERTURA*****	
PERIODO DE TIEMPO DEL CONTENIDO	
Fecha de Inicio	
Fecha de Termino	
DOMINIO ESPACIAL	
Coordenada Oeste	
Coordenada Este	
Coordenada Norte	
Coordenada Sur	
Sistema de coordenadas geográficas	
Sistema de coordenadas proyectadas	
Escala de los Datos	

Figura A.7: Cobertura

- **Coordenada Oeste, Coordenada Este, Coordenada Norte, Coordenada Sur, Sistema de coordenadas geográficas, Sistema de coordenadas proyectadas, Escala de los datos.** Este grupo de datos, son obligatorios opcionales, se deben llenar todos o ningún campo. Cada campo requiere de un sólo dato.

### A.3.4. Información de Datos Espaciales, Acceso y Uso de la Información, Creación e Identidad

Los campos correspondientes a esta información se pueden ver en la Figura A.8.

33	*****INFORMACION DE DATOS ESPACIALES*****	
34	Tipo de Datos	
35	Formato de Datos	
36	Proyección de Datos	
37	Escala de Datos	
38	*****ACCESO Y USO DE LA INFORMACION*****	
39	Restricciones de acceso	
40	Restricciones de uso	
41	*****CREACION E IDENTIDAD*****	
42	Tipo de contenido	
43	Formato de Datos	
44	Fecha de Publicación	
45	Fecha de Actualización	

Figura A.8: Información de Datos Espaciales, Acceso y Uso de la Información, Creación e Identidad

La descripción de los datos es la siguiente:

- **Tipo de Datos, Formato de Datos, Proyección de Datos y Escala**

de Datos. Son campos opcionales, pueden recibir 0 o 1 dato.

- **Restricciones de Acceso y Restricciones de Uso.** Campos opcionales, pueden recibir 0 o 1 dato.
- **Tipo de contenido, Formato de Datos y Fecha de Publicación.** Son campos obligatorios, requieren un sólo dato.
- **Fecha de Actualización.** Campo obligatorio, es un campo que se actualiza cada vez que se abre la plantilla en la cual se esta trabajando.

### A.3.5. Campos de Fechas

Los campos de este tipo, proporcionan un calendario, Figura A.9, al seleccionar la celda que requiere este tipo de dato se despliega el calendario para poder seleccionar la fecha en cuestión.

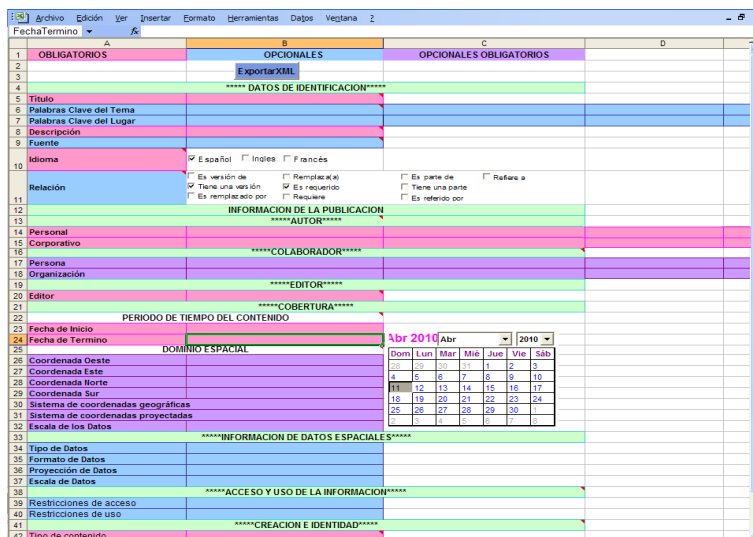


Figura A.9: Calendario

También se proporciona validación con respecto a las fechas, por ejemplo la fecha de inicio no puede ser mayor a la fecha de publicación, si se intenta poner un fecha mayor se enviará un mensaje de advertencia, Figura A.10.

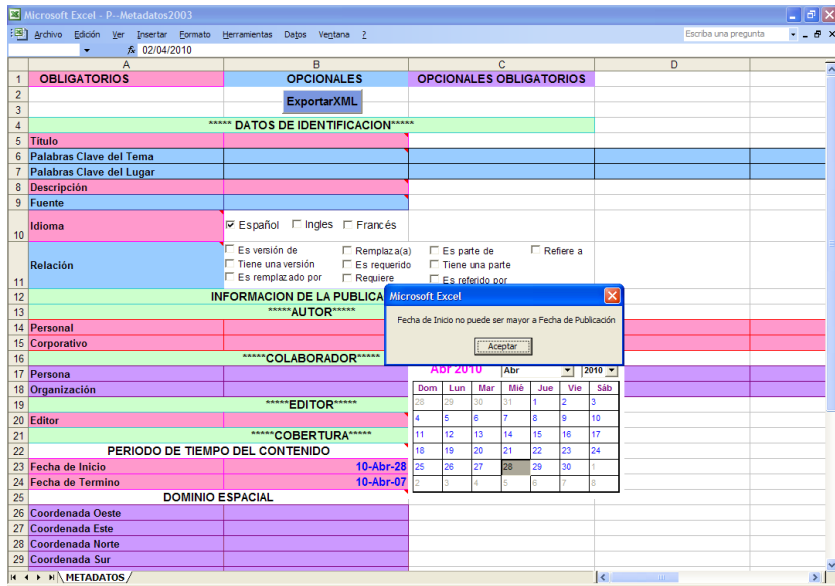


Figura A.10: Validación de Fechas

## A.4. Campos obligatorios opcionales

A	B	C	D
1	OBLIGATORIOS	OPCIONALES	OPCIONALES OBLIGATORIOS
2		ExportarXML	
3			
4	**** DATOS DE IDENTIFICACION****		
5	Título	ento global, y los efectos en la atmósfera	
6	Palabras Clave del Tema	atmosfera	calentamiento
7	Palabras Clave del Lugar	ciudad de mexico	otra palabra
8	Descripción	del calentamiento global en la atmosfera	clave
9	Fuente	repertes de la investigacion	
10	Idioma	<input checked="" type="checkbox"/> Español <input type="checkbox"/> Inoles <input type="checkbox"/> Francés	
11	Relación	<input type="checkbox"/> Es versión de <input type="checkbox"/> Reemplaza(x) <input checked="" type="checkbox"/> Tiene una versión <input checked="" type="checkbox"/> Es requerido <input type="checkbox"/> Es reemplazado por <input type="checkbox"/> Requiere	<input type="checkbox"/> Es parte de <input type="checkbox"/> Refere a <input type="checkbox"/> Tiene una parte <input type="checkbox"/> Es referido por
12	INFORMACION DE LA PUBLICACION		
13		****AUTOR****	
14	Personal	Luis Ruiz	otro autor
15	Corporativo	CCA	otra corporacion
16		****COLABORADOR****	
17	Persona	Eneclinal	otro colaborador
18	Organización	FCA	otro
19		****EDITOR****	
20	Editor	UNAM	
21		****COBERTURA****	
22	PERIODO DE TIEMPO DEL CONTENIDO		
23	Fecha de Inicio	10-Abr-01	
24	Fecha de Termino	10-Abr-30	
25	DOMINIO ESPACIAL		
26	Coordenada Oeste	33"	
27	Coordenada Este	22"	
28	Coordenada Norte	23"	
29	Coordenada Sur	56"	
30	Sistema de coordenadas geográficas	4:56	
31	Sistema de coordenadas proyectada	5:42	
32	Escala de los Datos	1:100	
33	****INFORMACION DE DATOS ESPACIALES****		
34	Tipo de Datos	secuenciales	
35	Formato de Datos	cd	
36	Proyección de Datos	proyeccion	
37	Escala de Datos	escala	
38	****ACCESO Y USO DE LA INFORMACION****		
39	Restricciones de acceso		
40	Restricciones de uso		
41	****CREACION E IDENTIDAD****		
42	Tipo de contenido	dv0	
43	Formato de Datos	ascii	
44	Fecha de Publicación	10-May-07	
45	Fecha de Actualización	10-Nov-07	

Figura A.11: Datos Capturados

Este tipo de campos están agrupados, por ejemplo, en la plantilla se ve el grupo Dominio Espacial, el cual es un grupo de datos que son campos obligatorios opcionales.

Para la captura de éste tipo de campos es necesario llenar todos los campos de este grupo o ninguno, en caso de llenar sólo algunos y tratar de exportar la plantilla el sistema enviará un mensaje de advertencia indicando los campos faltantes de este grupo. Por lo que se recomienda llenar todos los datos o no poner ninguno.

Una vez llenado los campos obligatorios, como se puede ver en la Figura

A.11, se da click en el botón ExportarXML para poder pasar los datos a formato XML.

Se mostrará un cuadro de dialogo, Figura A.12, para poder guardar los datos en XML, buscar la carpeta donde se desea guardar, dar un nombre al archivo y dar aceptar.

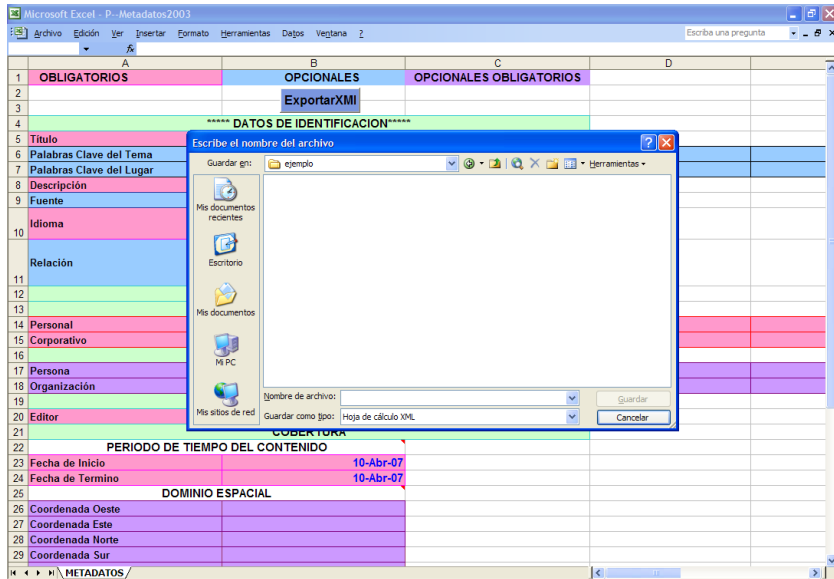


Figura A.12: Exportando documento XML

Una vez guardado el archivo, se busca en la carpeta en la que se guardó, se le da doble click y éste se abrirá en su navegador predeterminado, que mostrará el documento XML de los datos exportados, como en la Figura A.13.

El Libro Excel que se abrió a partir de la plantilla se puede guardar como un libro normal, para poder seguir editando, y poder exportar hasta que se tengan los datos requeridos.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <metadata>
- <Identification>
  <title>Calentamiento global, y los efectos en la atmósfera</title>
  - <keywordsItem>
    <item>atmosfera</item>
    <item>calentamiento</item>
    <item>otra palabra</item>
  </keywordsItem>
  - <keywordsPlace>
    <place>ciudad de mexico</place>
    <place>dave</place>
  </keywordsPlace>
  <description>investigación sobre los efectos del calentamiento global en la atmosfera</description>
  <source>repertes de la investigación</source>
  - <languages>
    <language>Español</language>
  </languages>
  - <relations>
    <relation>Tiene una versión</relation>
    <relation>Es requerido por</relation>
  </relations>
</Identification>
- <author>
- <personals>
  <personal>Luis Ruiz</personal>
  <personal>otro autor</personal>
  <personal>otro mas</personal>
</personals>
- <corporates>
  <corporate>CCA</corporate>
  <corporate>otra corporación</corporate>
</corporates>
</author>
- <collaborator>
- <peoples>
  <people>Enedina</people>
  <people>otro colaborador</people>
  <people>otro</people>
</peoples>
- <organizations>
  <organization>FCA</organization>
</organizations>
</collaborator>
<publisher>UNAM</publisher>
- <coverage>
- <periodOfTime>
  <dateStart>2010-04-01</dateStart>
  <dateEnd>2010-04-30</dateEnd>
</periodOfTime>
- <spatialCoverage>
  <west>33" </west>
  <east>22" </east>
  <north>23" </north>
  <south>56" </south>
  <geographics>4:5:6 </geographics>
  <projecteds>5:4:2 </projecteds>
  <scale>1:100 </scale>
</spatialCoverage>
</coverage>
- <dataSpatial>
  <dataType>secuenciales </dataType>
  <dataFormat>cd </dataFormat>
  <projection>proyeccion </projection>
  <dataScale>escala </dataScale>
</dataSpatial>
- <accessInformation>
  <access />
  <use />
</accessInformation>
- <creation>
  <contentType>dvd </contentType>
  <dataFormat>ascii </dataFormat>
  <datePublication>2010-05-07 </datePublication>
  <dateUpdate>2010-11-07 </dateUpdate>
</creation>
</metadata>

```

Figura A.13: Documento XML generado



---

# Apéndice B

## DTD

```
<!ELEMENT metadata ( Identification, author,
                    collaborator, publisher,
                    coverage, dataSpatial,
                    accessInformation, creation ) >

<!ELEMENT Identification ( title, keywordsItem,
                          keywordsPlace, description,
                          source*, languages, relations ) >
  <!ELEMENT title ( #PCDATA ) >
  <!ELEMENT keywordsItem ( item* ) >
    <!ELEMENT item ( #PCDATA ) >
  <!ELEMENT keywordsPlace ( place* ) >
    <!ELEMENT place ( #PCDATA ) >
  <!ELEMENT description ( #PCDATA ) >
  <!ELEMENT source ( #PCDATA ) >
  <!ELEMENT languages ( language ) >
    <!ELEMENT language ( #PCDATA ) >
  <!ELEMENT relations ( relation* ) >
```

```
    <!ELEMENT relation ( #PCDATA ) >

<!ELEMENT author ( personals, corporates ) >
    <!ELEMENT personals ( personal+ ) >
        <!ELEMENT personal ( #PCDATA ) >
    <!ELEMENT corporates ( corporate+ ) >
        <!ELEMENT corporate ( #PCDATA ) >

<!ELEMENT collaborator ( peoples, organizations ) >
    <!ELEMENT peoples ( people* ) >
        <!ELEMENT people ( #PCDATA ) >
    <!ELEMENT organizations ( organization* ) >
        <!ELEMENT organization ( #PCDATA ) >

<!ELEMENT publisher ( #PCDATA ) >

<!ELEMENT coverage ( periodOfTime, spatialCoverage ) >
    <!ELEMENT periodOfTime ( dateStart, dateEnd) >
        <!ELEMENT dateStart ( #PCDATA ) >
        <!ELEMENT dateEnd ( #PCDATA ) >
    <!ELEMENT spatialCoverage ( west?, east?,north?,
                                south?, geographics?,
                                projecteds?, scale? ) >
        <!ELEMENT west ( #PCDATA ) >
        <!ELEMENT east ( #PCDATA ) >
        <!ELEMENT north ( #PCDATA ) >
        <!ELEMENT south ( #PCDATA ) >
        <!ELEMENT geographics ( #PCDATA ) >
        <!ELEMENT projecteds ( #PCDATA ) >
        <!ELEMENT scale ( #PCDATA ) >
```

```
<!ELEMENT dataSpatial ( dataType?, dataFormat?,
                        projection?, dataScale? ) >
  <!ELEMENT dataType ( #PCDATA ) >
  <!ELEMENT dataFormat ( #PCDATA ) >
  <!ELEMENT projection ( #PCDATA ) >
  <!ELEMENT dataScale ( #PCDATA ) >

<!ELEMENT accessInformation ( access?, use? ) >
  <!ELEMENT access ( #PCDATA ) >
  <!ELEMENT use ( #PCDATA ) >

<!ELEMENT creation ( contentType, formatDate,
                    datePublication, dateUpdate ) >
  <!ELEMENT contentType ( #PCDATA ) >
  <!ELEMENT formatDate ( #PCDATA ) >
  <!ELEMENT datePublication ( #PCDATA ) >
  <!ELEMENT dateUpdate ( #PCDATA ) >
```

---

## Apéndice C

# XMLSchema

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="metadata">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Identification" />
      <xs:element ref="author" />
      <xs:element ref="collaborator" />
      <xs:element name="publisher" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element ref="coverage" />
      <xs:element ref="dataSpatial" />
      <xs:element ref="accessInformation" />
      <xs:element ref="creation" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="Identification">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="title" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element ref="keywordsItem" />
      <xs:element ref="keywordsPlace" />
      <xs:element name="description" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="source" type="xs:string"
        minOccurs="0" maxOccurs="1" />
      <xs:element ref="languages" />
      <xs:element ref="relations" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="author">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="personals" />
      <xs:element ref="corporates" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="collaborator">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="peoples" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        <xs:element ref="organizations" />
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="coverage">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="periodOfTime" />
            <xs:element ref="spatialCoverage" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="dataSpatial">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="dataType" type="xs:string"
                minOccurs="0" maxOccurs="1"/>
            <xs:element name="dataFormat" type="xs:string"
                minOccurs="0" maxOccurs="1"/>
            <xs:element name="projection" type="xs:string"
                minOccurs="0" maxOccurs="1"/>
            <xs:element name="dataScale" type="xs:string"
                minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="accessInformation">
    <xs:complexType>
```

```
<xs:sequence>
  <xs:element name="access" type="xs:string"
    minOccurs="0" maxOccurs="1"/>
  <xs:element name="use" type="xs:string"
    minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="creation">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="contentType" type="xs:string"
        minOccurs="1" maxOccurs="1"/>
      <xs:element name="dataFormat" type="xs:string"
        minOccurs="1" maxOccurs="1"/>
      <xs:element name="datePublication" type="xs:date"/>
      <xs:element name="dateUpdate" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="keywordsItem">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="item" type="xs:string"
        minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="keywordsPlace">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="place" type="xs:string"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="languages">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="language" type="xs:string"
        minOccurs="1" maxOccurs="3" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="relations">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="relation" type="xs:string"
        minOccurs="0" maxOccurs="10" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="personals">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="personal" type="xs:string">
```



```
                minOccurs="1" maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="corporates">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="corporate" type="xs:string"
                    minOccurs="1" maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="peoples">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="people" type="xs:string"
                    minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="organizations">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="organization" type="xs:string"
                    minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="periodOfTime">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="dateStart" type="xs:date"/>
      <xs:element name="dateEnd" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="spatialCoverage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="west" type="xs:string"
        minOccurs="0" maxOccurs="1" />
      <xs:element name="east" type="xs:string"
        minOccurs="0" maxOccurs="1"/>
      <xs:element name="north" type="xs:string"
        minOccurs="0" maxOccurs="1"/>
      <xs:element name="south" type="xs:string"
        minOccurs="0" maxOccurs="1" />
      <xs:element name="geographics" type="xs:string"
        minOccurs="0" maxOccurs="1"/>
      <xs:element name="projecteds" type="xs:string"
        minOccurs="0" maxOccurs="1"/>
      <xs:element name="scale" type="xs:string"
        minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

`</xs:schema>`

---

## Apéndice D

# Diseño de Pruebas

### D.1. Pruebas correspondientes a los Caso de Uso

**Caso de Uso:** Entrar (al sistema)

**Pruebas del sistema:**

<b>Caso de Uso</b>	<b>Entrada</b>	<b>Esperado</b>	<b>Obtenido</b>
Entrar	Plantilla Excel	Plantilla Disponible	Despliega interfaz inicial de plantilla
	Plantilla Excel	Plantilla no Disponible	El programa Excel no muestra la interfaz de la plantilla

**Caso de Uso:** Validar Fechas**Pruebas del sistema:**

<b>Caso de Uso</b>	<b>Entrada</b>	<b>Esperado</b>	<b>Obtenido</b>
Validar Fechas	Dar clic en una celda correspondiente a una fecha	Se seleccionó correctamente la fecha	La validación se realizó correctamente
	Dar clic en una celda correspondiente a una fecha	La fecha no es válida	El programa Excel muestra un mensaje de advertencia indicando el tipo de error con respecto a las fechas

**Caso de Uso:** Validar Metadatos**Pruebas del sistema:**

<b>Caso de Uso</b>	<b>Entrada</b>	<b>Esperado</b>	<b>Obtenido</b>
Validar Metadatos	Dar clic en el botón Exportar	Verificar que no falte por llenar ningún campo obligatorio	Se verificaron los campos obligatorios
	Dar clic en el botón Exportar	No verifica los campos obligatorios	La aplicación no valida los campos obligatorios

**Caso de Uso:** Crear documentos**Pruebas del sistema:**

<b>Caso de Uso</b>	<b>Entrada</b>	<b>Esperado</b>	<b>Obtenido</b>
Crear documentos	Crear documento XML	Opción Exportar XML	Documento XML
	Crear Libro Excel	Opción Guardar	Libro Excel
	Salir	Opción Salir	Cierra la aplicación

**Caso de Uso:** Crear documento XML

**Pruebas del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Crear documento XML	Exportar XML	Campos obligatorios completos	Documento XML
	Exportar XML	Campos obligatorios faltantes	Mensaje de advertencia "Falta el campo ___ favor de capturar"

**Caso de Uso:** Crear documento Excel

**Pruebas del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Crear documento Excel	Guardar		Libro Excel

**Caso de Uso:** Salir (del sistema)

**Pruebas del sistema:**

Caso de Uso	Entrada	Esperado	Obtenido
Salir	Dar clic en el icono cerrar	Terminar la ejecución del sistema	El sistema terminó
	Dar clic en el icono cerrar	No termina la ejecución del sistema	El programa Excel muestra un mensaje preguntando si desea guardar los cambios a la plantilla

---

# Bibliografía

- [DEJO2001] Annemieke de Jong. *Los metadatos en el entorno de la producción audiovisual. Una introducción*. FIAT-IFTA, 2001.
- [DCMI] *Dublin Core Metadata Initiative* [en línea].  
< <http://dublincore.org/> >
- [BERN1999] Berners-Lee, T. *Tejiendo la Red: El inventor de la World Wide Web nos descubre su origen*. Siglo XXI de España Editores, 1999.
- [LAGO1996] Carl Lagoze. *The Warwick Framework. A Container Architecture for Diverse Sets of Metadata*. D-Lib Magazine [en línea].  
< <http://www.dlib.org/> > July/August, 1996.
- [CSDGM] *Content Standard for Digital Geospatial Metadata* [en línea].  
< <http://www.fgdc.gov/metadata/csdgm/> >
- [DAWO2004] Dantzing, G.B. and P. Wolfe, *Descomposition principle for linear programs*. Limusa. Mexico, D.F., 2004.
- [DRAE2001] *Diccionario de la Real Academia Española* [en línea].  
< <http://buscon.rae.es/draeI/> > ESPASA. 2001.
- [FGDC] *The Federal Geographic Data Committee* [en línea].  
< <http://www.fgdc.gov/> >

- [FOLD2010] Denis Howe. *Free On-Line Dictionary of Computing* [en línea]. <<http://foldoc.org>> Febrero 2010.
- [GILS] *Global Information Locator Service* [en línea]. < <http://www.gils.net/>>
- [MARC] *MARC STANDARDS* [en línea]. < <http://www.loc.gov/marc/>>
- [TRAM2006] Tramullas Jesús. *Tendencias en documentación digital*. TREA, 2006.
- [VIAN2004] Vianello Osti Marina. *El Hipertexto entre la utopía y la aplicación: identidad, problemática y tendencia de la Web*. TREA, 2004.
- [MEND2002] Méndez Rodríguez Eva. *Metadatos y recuperación de información. Estándares, problemas y aplicabilidad en bibliotecas digitales*. TREA, 2002.
- [SOMM2002] Sommerville Ian. *Ingeniería de Software*. Pearson Educación, 2002.
- [GILL1998] Gilliland Swetland, A. J. *Defining metadata. Introduction to metadata: pathways to digital information*. Los Angeles, Getty Information Institute, 1998.
- [JOHN1998] Pete Johnston. *What you always wanted to know about SGML, HTML and XML. University of Glasgow Archives and Business Records Centre* [en línea]. < <http://www.gla.ac.uk/InfoStrat/socarcpj/>> noviembre, 1998.
- [JOYA2008] Joyanes Aquilar Luis. *FUNDAMENTOS DE PROGRAMACIÓN. Algoritmos, estructura de datos y objetos*. Cuarta Edición. McGraw-Hill, 2008.
- [PRES2005] Roger Pressman, *Ingeniería del Software: Un enfoque práctico*. McGraw-Hill, 2005.



- [DOMA2007] Doug Rosenberg and Matt Stephens, *Use Case Driven Object Modeling with UML. Theory and Practice* SPRINGER, 2007.
- [BRUN2001] Eíto Brun Ricardo, *Progrmación con XML*. Madrid, Anaya Multimedia, 2001.
- [SCHA2010] Schafer, Steven M. *HTML, XHTML y CSS*. Madrid, Anaya Multimedia, 2010.
- [FLFJ2005] Alonso Amo Fernando, Martínez Normand Loïc y Segovia Pérez Fco. Javier. *INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE: MODELOS DE DESARROLLO DE PROGRAMAS*. Madrid, DELTA PUBLICACIONES, 2005.
- [NARTE2006] *Welcome to the NARSTO Quality Systems Science Center, 2006* [en línea].  
<<http://cdiac.ornl.gov/programs/NARSTO/>>[Consulta: 3 de octubre 2010]
- [NARST2006] *NARSTO Better Air Quality for North America, 2006* [en línea].  
<<http://www.narsto.org/>>[Consulta: 2 de octubre 2010]
- [TEI2006] *Text Encoding Initiative* [en línea].  
< <http://www.tei-c.org/index.html>>[Consulta: 25 de septiembre 2010]
- [W3C] *World Wide Web Consortium* [en línea]. < <http://www.w3.org/>>