

## ST.96 - ANEXO I

### NORMAS DE DISEÑO Y CONVENCIONES DE XML

Versión 2.0

*Revisión aprobada el 28 de mayo de 2015 por el Equipo Técnico XML4IP del Comité de Normas Técnicas de la OMPI*

#### Índice

ST.96 - ANEXO I .....	1
1. INTRODUCCIÓN.....	3
1.1 Panorama general .....	3
1.2 Alcance.....	3
1.3 ¿Cómo utilizar el presente documento? .....	3
1.4 Estructura del documento .....	3
1.5 Terminología y notación .....	3
1.5.1 Palabras clave .....	3
1.5.2 Notaciones generales .....	4
1.5.3 Identificadores de normas.....	4
2. CONVENCIONES DE DISEÑO DE XML.....	4
2.1 Normas generales de diseño de XML.....	4
2.2 Convenciones sobre denominación en XML.....	5
2.2.1 Convenciones sobre denominación de construcciones de esquema .....	5
2.2.2 Convenciones sobre nombres de archivo de esquema.....	6
2.3 Estrategia de modularidad.....	7
2.3.1 Módulos de esquema.....	7
2.3.2 Referencia a un esquema externo .....	7
2.4 Capacidad de reutilización.....	8
2.5 Espacios de nombres .....	8
2.5.1 Declaración y calificación de espacio de nombres.....	9
2.5.2 Espacios de nombres en un esquema de XML.....	9
2.5.3 Espacios de nombres de destino .....	9
2.5.4 Espacios de nombres por defecto.....	10
2.6 Versiones del esquema .....	11
2.6.1 Cambios mayores y cambios menores .....	11
2.6.1.1 Versiones mayores .....	11
2.6.1.2 Versiones menores .....	11
2.6.2 Estrategia relativa a las versiones de un esquema .....	12
2.6.2.1 Convenciones sobre denominación de ficheros y archivos en las versiones del esquema .....	12
2.6.2.2 Atributo <code>version</code> integrado en el esquema de XML en las versiones del esquema.....	12
2.6.2.3 Atributos <code>st96Version</code> e <code>ipoVersion</code> definidos por el usuario en las versiones del esquema para instancias XML.....	12
2.7 Transformabilidad con otras normas sobre XML de la OMPI.....	14
2.8 Esquemas desarrollados por la industria .....	15
3. CONVENCIONES RELATIVAS A CONSTRUCCIONES DE ESQUEMAS DE XML.....	15
3.1 Definiciones de tipos.....	15
3.1.1 Tipos simples.....	15
3.1.1.1 Tipos de datos integrados del W3C .....	15
3.1.1.2 Tipos de datos definidos por el usuario .....	15
3.1.2 Tipos complejos.....	16

Anexo I, página 2

3.2	Elementos y atributos .....	16
3.2.1	Elementos frente a atributos .....	16
3.2.2	Elementos .....	16
3.2.2.1	Cardinalidad de los elementos .....	16
3.2.2.2	Elementos vacíos .....	17
3.2.3	Atributos .....	17
3.2.4	Agrupación de elementos y atributos .....	17
3.3	Ampliación y restricción .....	17
3.3.1	Ampliación .....	17
3.3.2	Restricción .....	18
3.3.3	Grupos de sustitución .....	18
3.4	Limitaciones de identidad .....	18
3.5	Documentación del esquema .....	18
3.5.2	Documentación del encabezado del esquema .....	19
4.	<b>NORMAS DE DISEÑO DE LA INSTANCIA</b> .....	19
4.1	Espacios de nombres en documentos de instancia XML .....	19
4.1.1	Validación de un documento de instancia XML .....	20
4.1.2	Declaración y calificación del espacio de nombres en documentos de instancia XML .....	20
4.1.3	Espacio de nombres de una aplicación del esquema del W3C .....	21
4.1.4	Alcance del espacio de nombres .....	21
4.2	Entidades externas .....	22
	<b>APÉNDICE A – RESUMEN DE LAS NORMAS DE DISEÑO</b> .....	23
	Normas generales de diseño .....	23
	Normas de diseño de esquema .....	24
	Normas de diseño de instancia .....	28
	<b>APÉNDICE B – TÉRMINOS DE REPRESENTACIÓN</b> .....	29
	<b>APÉNDICE C – LISTA DE SIGLAS Y ABREVIATURAS</b> .....	30
	<b>APÉNDICE D – MATERIAL DE REFERENCIA</b> .....	32
	Normas de la OMPI .....	32
	Normas de la industria .....	32

Anexo I, página 3

## 1. INTRODUCCIÓN

### 1.1 *Panorama general*

1. En el documento *Normas de diseño y convenciones de XML* se ofrece una oportunidad para fomentar la armonización entre todos los tipos de propiedad industrial (P.I.) definidos en la Norma ST.96 de la OMPI y facilitar el intercambio de datos entre Oficinas de P.I., presentando esquemas de XML reutilizables y compatibles.

### 1.2 *Alcance*

2. La finalidad del presente documento es proporcionar un conjunto amplio de normas de diseño y convenciones para crear y utilizar esquemas e instancias XML que abarquen todos los tipos de información de P.I. que faciliten la presentación, el tratamiento, la publicación y el intercambio de datos entre la comunidad de P.I.

### 1.3 *¿Cómo utilizar el presente documento?*

3. Los destinatarios del presente documento son las Oficinas de P.I., los proveedores de datos de P.I. y la comunidad de P.I. en sentido más amplio. La comunidad de P.I. debe tomar el presente documento como el punto de referencia en lo que respecta a las normas de diseño y las convenciones aprobadas que todos los esquemas de XML deben seguir para ser considerados compatibles con la Norma ST.96. Las Oficinas de P.I. pueden basarse en el presente documento para desarrollar sus propias normas de diseño. El presente documento debe consultarse antes de empezar a elaborar un nuevo esquema de XML o de modificar uno ya existente. Una vez desarrollado, el presente documento debe consultarse para verificar que el esquema de XML se ajusta a las normas de diseño.

### 1.4 *Estructura del documento*

4. Se recomienda leer el presente documento en el orden en que se redactó. La presente guía se estructura como sigue:

- Sección 1, Introducción, donde se describen las normas generales que se aplican a lo largo de todo el documento;
- Sección 2, Convenciones de diseño de XML, donde se definen las normas de alto nivel que se aplican en la elaboración de esquemas y aplicaciones;
- Sección 3, Convenciones relativas a construcciones de esquemas de XML, donde se definen normas de diseño concretas para utilizar las especificaciones del esquema del W3C en la creación de esquemas de XML; y
- Sección 4, Normas de diseño de la aplicación, donde se definen normas de diseño concretas para la creación de aplicaciones.

5. Además, la presente guía consta de cuatro apéndices:

- Apéndice A, Resumen de las normas de diseño, donde se resumen las normas de diseño que aparecen en el presente documento;
- Apéndice B, Términos de representación, que contiene las definiciones de los términos de representación;
- Apéndice C, Lista de siglas y abreviaturas, que contiene los términos y las abreviaturas empleados en el presente documento; y
- Apéndice D, Material de referencia, que contiene referencias a Normas de la OMPI y otras normas del sector;

### 1.5 *Terminología y notación*

#### 1.5.1 Palabras clave

6. En general,

- se entenderá por “esquema de XML” un lenguaje para describir la estructura y limitar los contenidos de los documentos de XML; y
- se entenderá por “esquema de XML del W3C” todo esquema de XML plenamente adaptado al conjunto de recomendaciones sobre lenguaje de definición de esquemas de XML del W3C—*Esquema de XML; Parte 1: Estructuras y Esquema de XML; Parte 2: Tipos de datos.*

Anexo I, página 4

7. En el presente documento,
- se entenderá por “esquema” todo esquema de XML de acuerdo con la definición que se proporciona en el Anexo III de la Norma ST.96 de la OMPI; y
  - se entenderá por “componente” un tipo, elemento o atributo.
8. Las palabras clave DEBE, NO DEBE, DEBERÁ, DEBERÍA, NO DEBERÍA y PUEDE han de ser interpretadas según lo indicado en la sección “Definiciones y terminología” de la Norma ST.96 de la OMPI. Cuando aparezcan en minúscula, se utilizarán en el sentido habitual en el idioma español.

### 1.5.2 Notaciones generales

9. A lo largo del presente documento se utilizan las siguientes notaciones:
- <>: Indica un término descriptivo provisional que, en la aplicación, se sustituirá por un valor específico de instancia.
  - “ ”: Indica que el texto entre comillas debe usarse de manera literal en la aplicación.
  - { }: Indica que la aplicación de estos elementos es facultativa.
  - El tipo de letra *Courier* indica palabras clave de XML, nombres de etiquetas de XML y códigos de XML.

### 1.5.3 Identificadores de normas

10. Todas las normas de diseño son normativas. Las normas de diseño se identifican mediante el prefijo [XX-nn].
- a) El valor “XX” es un prefijo para clasificar los tipos de normas de la manera siguiente:
- GD para las normas generales de diseño;
  - SD para las normas de diseño de esquemas; y
  - ID para las normas de diseño de instancias.
- b) El valor “nn” indica el número siguiente disponible en la serie de un tipo específico de norma. Cabe observar que el número no indica la posición de la norma, en particular para una nueva norma. Una nueva norma se colocará en el contexto pertinente. Por ejemplo, el identificador de norma [GD-40] identifica la cuadragésima norma general de diseño. La norma [GD-40] puede colocarse entre las normas [GD-20] y [GD-21] en lugar de colocarse a continuación de [GD-39] si es esa la ubicación más adecuada para esa norma.
- c) El identificador de norma de la norma suprimida se mantendrá, pero la norma será sustituida por el texto “Suprimida”.

## 2. CONVENCIONES DE DISEÑO DE XML

### 2.1 Normas generales de diseño de XML

11. En esta sección de la guía se describen las normas de diseño y las directrices de XML generales y de alto nivel aplicables a todas las iniciativas de desarrollo de XML y no solo a una faceta determinada de la tecnología de XML. Las normas generales y las directrices que se citan a continuación son la base común para recabar datos y elaborar documentos de P.I.

- [GD-01] Todos los esquemas de XML DEBEN basarse en las especificaciones técnicas del W3C que han alcanzado la condición de Recomendación.
- [GD-02] Los esquemas DEBEN ajustarse a Esquema de XML, Parte 1: Estructuras (<http://www.w3.org/TR/xmlschema-1/>) y Esquema de XML, Parte 2: Tipos de datos (<http://www.w3.org/TR/xmlschema-2/>).
- [GD-03] En los esquemas DEBE usarse el conjunto de caracteres ISO/IEC 10646 – UCS – Unicode. DEBE usarse UTF-8 para codificar los caracteres Unicode.

Anexo I, página 5

2.2 *Convenciones sobre denominación en XML*

12. Estas convenciones son necesarias para garantizar la coherencia, la uniformidad y la exhaustividad en la denominación y la definición de todos los recursos de XML.

2.2.1 Convenciones sobre denominación de construcciones de esquema

13. Las convenciones sobre denominación en XML de la Norma ST.96 de la OMPI se basan en las directrices y los principios descritos en el documento *ISO 11179 Part 5 - Naming and Identification Principles*. Los nombres de tipos, elementos y atributos se componen de los siguientes términos:

- Se entenderá por clase de objeto una actividad u objeto en un contexto operativo y representa la agrupación o agregación lógica de datos (en un modelo lógico de datos) a la que pertenece una propiedad. La clase de objeto se expresa mediante un término de clase de objeto.
- Se entenderá por término de propiedades las características de la clase de objeto.
- Se entenderá por término de representación un término que clasifica el formato del elemento de datos en categorías más amplias. En el caso de la Norma ST.96 de la OMPI, deberían usarse los términos de representación que aparecen en el Apéndice B del presente documento.
- Se entenderá por término calificador la palabra o palabras que ayudan a definir y a diferenciar un elemento de datos del resto de elementos de datos conexos y que, si fuera necesario, podrá adjuntarse a un término de clase de objeto o un término de propiedades para que un nombre sea único.

- [GD-04] Los nombres de tipos, elementos y atributos DEBEN estar compuestos de palabras del idioma inglés, utilizando la grafía inglesa básica que figura en el Diccionario Inglés Oxford. Las únicas excepciones permitidas son las siglas, las abreviaturas y otras formas truncadas que se enumeran en el Apéndice C.
- [GD-05] Los nombres de tipos, elementos y atributos solamente DEBERÍAN estar compuestos por nombres, adjetivos y verbos en presente, con la excepción de las siglas, abreviaturas y otras formas truncadas que se enumeran en el Apéndice C.
- [GD-06] Los caracteres empleados en los nombres de tipos, elementos y atributos DEBEN estar contenidos en el conjunto siguiente: 'a-z, A-Z y 0-9'.
- [GD-07] La longitud máxima de un nombre de componente DEBERÍA ser de no más de 35 caracteres.
- [GD-08] Los nombres de tipos, elementos y atributos DEBERÍAN ser concisos y explicarse por sí solos.
- [GD-09] Los nombres de elementos DEBEN seguir la convención *upper camel case* (UCC). Por ejemplo, `CountryCode`.
- [GD-10] Los nombres de tipos DEBEN seguir la convención UCC y poseer el sufijo `Type`. Por ejemplo, `ApplicantType`.
- [GD-11] Los nombres de atributos DEBEN seguir la convención *lower camel case* (LCC). Por ejemplo, `currencyCode="EUR"`.
- [GD-12] Las siglas y las abreviaturas que figuran en el Apéndice C siempre DEBEN utilizarse en lugar del nombre completo ampliado.
- [GD-13] Las siglas y las abreviaturas que aparecen al principio de una declaración de atributos DEBEN estar en minúscula en su totalidad. El resto de siglas y abreviaturas utilizadas en una declaración de atributos DEBEN aparecer en mayúscula, según se indica en el Apéndice C.
- [GD-14] Las siglas y las abreviaturas DEBEN aparecer según lo indicado en el Apéndice C para los nombres de elementos y tipos.
- [GD-15] Los nombres de tipos complejos DEBERÍAN incluir un término de clase de objeto significativo.
- [GD-16] Cuando se necesitan dos términos de clase de objeto en un determinado nombre de componente debido a la naturaleza de la entidad, podrá utilizarse un tipo complejo de asociación. En los nombres de tipo complejo de asociación DEBERÍA utilizarse una estructura de clase de objeto del tipo complejo de asociación, un término de propiedades que describa la naturaleza de la asociación, y la clase de objeto del tipo complejo asociado. Los términos de clase de objeto y de propiedades pueden estar precedidos por calificadores. Por ejemplo, `ApplicantResidenceAddress`, `Applicant` es la clase de objeto del tipo complejo de

Anexo I, página 6

asociación, *Residence* es la propiedad y *Address* es la clase de objeto del tipo complejo asociado.

- [GD-17] Los nombres de elementos básicos (los que se basan en un tipo simple o complejo que no admite elementos derivados) DEBERÍAN componerse de un término de clase de objeto, término de propiedades y, cuando corresponda, un término de representación. Los términos de clase de objeto y de propiedades pueden estar precedidos por calificadores.
- [GD-18] Un término de clase de objeto siempre DEBE tener el mismo significado semántico en todo un espacio de nombres y PUEDE componerse de más de una palabra. Por ejemplo, *ContactInformation*.
- [GD-19] Un término de propiedades de un nombre DEBE ser exclusivo en el contexto de una clase de objeto pero PUEDE reutilizarse en distintas clases de objetos.
- [GD-20] Un término calificador PUEDE adjuntarse a un término de clase de objeto o a un término de propiedades si fuera necesario para individualizar un nombre.
- [GD-21] Cuando un nombre contiene un término de clase de objeto, un término de propiedades y un término de representación, el término de clase de objeto DEBE anteceder al término de propiedades y el término de propiedades DEBE anteceder al término de representación. Un término calificador DEBERÍA anteceder al término de propiedades o al término de clase de objeto asociado.
- [GD-22] Si el término de propiedades acaba con la misma palabra que el término de representación (o una palabra equivalente), DEBE suprimirse el término de representación.
- [GD-23] Cuando es necesario un término de representación, DEBEN emplearse los términos de representación del Apéndice B para los términos de representación en los nombres de componentes básicos.
- [GD-24] En un espacio de nombres, todos los nombres de tipos, elementos y atributos DEBEN ser únicos.
- [GD-25] La palabra o palabras de un nombre DEBERÍAN aparecer en singular salvo que el concepto mismo sea plural. Por ejemplo: *TotalMarkSeries*
- [GD-26] El nombre de un elemento o un tipo que contenga una colección de componentes relacionados por el contexto DEBERÍA poseer el sufijo "Bag". Por ejemplo, *EmailAddressBag* representa una colección de elementos *EmailAddress*.
- [GD-27] Conectores como "and", "of" o "the" NO DEBERÍAN utilizarse en nombres de tipos, elementos y atributos salvo que formen parte de la terminología de la actividad.
- [GD-28] Los nombres de tipos, elementos y atributos NO DEBEN traducirse, modificarse o sustituirse con ningún fin.
- [GD-29] Los nombres de tipos y elementos NO DEBEN remitir a números de artículos o normas. Por ejemplo, *PCTRule702C* para el PCT.

### 2.2.2 Convenciones sobre nombres de archivo de esquema

14. Los nombres de archivo de esquema y los nombres de esquema a menudo van emparejados. Los nombres de archivo de esquema se basan en el nombre de esquema correspondiente. Por ejemplo, el nombre de archivo de *PostalAddressType.xsd* deriva del nombre de esquema *PostalAddressType*. En el presente documento, las convenciones sobre nombres de archivo de esquema están relacionadas con las normas de las convenciones sobre denominación en XML.

15. Un archivo de esquema PUEDE contener información sobre la versión. Un esquema en fase de proyecto puede ser revisado. En los proyectos de esquema, el nombre de archivo del esquema debe incluir la letra "D" y el número de revisión para indicar que se trata de un proyecto de esquema.

- [GD-30] Los caracteres utilizados en el nombre de archivo de esquema DEBEN pertenecer al conjunto siguiente: a-z, A-Z, 0-9, guion bajo "\_", y punto ".".
- [GD-31] El nombre de archivo del esquema DEBE constar de dos partes obligatorias con un delimitador e información facultativa sobre la versión con dos delimitadores adicionales, es decir: <nombre del componente>{"\_"<número de versión mayor>"\_"}<número de versión menor>"}."<extensión del archivo>; por ejemplo, *EmailAddressType.xsd*, *languageCode.xsd*, *ApplicationBody\_V1\_0.xsd*.

Anexo I, página 7

[GD-32] Un nombre de archivo de proyecto de esquema DEBE constar de cuatro partes obligatorias con dos delimitadores e información facultativa sobre la versión con dos delimitadores adicionales, es decir: <nombre del componente>{"\_""V"<número de versión mayor>"\_""<número de versión menor>"}\_""D" <número de revisión>".<extensión del archivo>, por ejemplo, Contact\_D3.xsd, TrademarkApplication\_V1\_1\_D1. Si un proyecto de esquema se basa en un esquema existente y tiene información sobre la versión en su archivo de nombre, los números de la versión mayor y menor del archivo de nombre del proyecto de esquema DEBERÍAN ser los mismos indicados en el archivo de esquema en que se basa el proyecto de esquema. Si un proyecto de esquema es nuevo, el número de versión mayor en el nombre de archivo del proyecto de esquema DEBERÍA ser el mismo número indicado en el espacio de nombres correspondiente, y un número de versión menor en el proyecto de esquema DEBERÍA ser cero "0".

### 2.3 Estrategia de modularidad

16. La Norma ST.96 de la OMPI se basa en gran medida en el concepto de modularidad en el diseño del esquema. Dividir la elaboración del esquema en múltiples módulos de esquema crea un mecanismo que permite importar componentes reutilizables en lugar todo el esquema. Por lo tanto, en la Norma ST.96 se recomienda no definir, en la medida de lo posible, todos los elementos y los componentes lógicos en un único esquema de XML monolítico, ya que eso reduce la posibilidad de compartir y reutilizar elementos individuales o componentes lógicos definidos como un grupo en un esquema.

#### 2.3.1 Módulos de esquema

17. Los componentes definidos en la Norma ST.96 de la OMPI se dividen en componentes comunes, componentes de patente, componentes de marca y componentes de dibujos y modelos industriales. Los componentes comunes DEBERÍAN ser neutrales con respecto al contexto (o independientes de la actividad) o ser compartidos por, al menos, dos tipos de P.I. Excepcionalmente, teniendo en cuenta la evolución futura de la Norma ST.96 de la OMPI, algunos componentes, como el elemento "P", podrán ser definidos como componente común.

18. Existen tres tipos distintos de componente, a saber, básico, agregado y de documento. Los componentes de documento, compuestos por componentes básicos, agregados y/u otros componentes de documento, se definen en función de las necesidades de una actividad. Los componentes básico y agregado se definen en el ámbito común. Los componentes básico, agregado y de documento se definen en los ámbitos de patentes, marcas y dibujos o modelos.

Nivel	Descripción
Componente básico	Un componente básico remite a un tipo de datos integrado del W3C, tipo simple o tipo complejo, y se define <code>xsd:simpleContent</code> . Está representado por un elemento o un atributo. Por ejemplo, <code>FirstName (string)</code> , <code>PhoneNumberCategory (PhoneNumberCategoryType)</code> .
Componente agregado	Un componente agregado es una colección de componentes básicos conexos y/u otros componentes agregados que, juntos, transmiten un significado específico distinto, independiente o no de cualquier contexto operativo concreto. Se representa mediante un elemento o un tipo complejo. Por ejemplo, <code>Name (NameType)</code> , <code>Contact (ContactType)</code> .
Componente de documento	Un componente de documento es un agrupamiento de componentes que comprende componentes básicos, agregados y/u otros tipos de componentes de documento, destinado a un aspecto específico de una actividad de P.I., por ejemplo, la presentación de solicitudes, el examen, la publicación, la concesión, y el intercambio de datos. Por ejemplo, <code>ApplicationBody</code> , basado en <code>ApplicationBodyType</code> , representa un fascículo de patente y puede ser utilizado para la presentación y publicación de un fascículo de patente.

#### 2.3.2 Referencia a un esquema externo

19. Cada módulo de esquema reside en un espacio de nombres distinto. En la Norma ST.96, cada módulo de esquema está definido en uno de los cuatro espacios de nombres asignados, respectivamente, a los módulos de esquema del componente común, los módulos de esquema del componente de patente, los módulos de esquema del componente de marca y los módulos de esquema del componente de dibujos y modelos industriales.

20. Para referirse a esquemas externos, se emplean dos construcciones de esquema del W3C, a saber, `include` e `import`. La construcción `include` debe utilizarse cuando el esquema incluyente y el esquema incluido tengan el mismo espacio de nombres de destino. Se entenderá por *esquema incluyente* todo esquema que incluye un esquema externo, y por *esquema incluido* el esquema externo propiamente dicho. La construcción `import` debe utilizarse cuando el esquema incluyente y el esquema incluido tengan espacios de nombres de destino distintos. Esta técnica puede ser útil si una

Anexo I, página 8

organización no necesita limitar el uso de construcciones de esquemas únicamente a aquellos que pertenecen a un espacio de nombres concreto.

- [SD-01] Los módulos de esquema de componente de patente, los módulos de esquema de componente de marca y los módulos de esquema de componente de dibujos y modelos industriales DEBEN utilizar la construcción `xsd:import` para remitirse a módulos de esquema de componente común.
- [SD-02] Los módulos de esquema de componente común NO DEBEN remitirse a los módulos de esquema de componente de patente, a los módulos de esquema de componente de marca o a los módulos de esquema de componente de dibujos y modelos industriales.
- [SD-03] Un módulo de esquema de componente de patente, de componente de marca o de componente de dibujo o modelo industrial PUEDE remitirse a módulos de esquema de componente común, esquemas del mismo módulo de componente y esquemas desarrollados por la industria aprobados, pero NO DEBEN remitirse a otros módulos de esquema de componente. Por ejemplo, los esquemas de patente NO DEBEN remitirse a esquemas de marca y viceversa.

#### 2.4 Capacidad de reutilización

21. La disociación y la cohesión son principios importantes de diseño de los esquemas de XML. La disociación tiene como finalidad minimizar la dependencia entre elementos, tanto en el documento resultante de una aplicación como en el esquema. Por su parte, la cohesión busca agrupar fragmentos de datos conexos. Existen algunos patrones de diseño que abordan la cuestión de la disociación y la cohesión en los esquemas de XML.

22. Los patrones de diseño también dan respuesta a la cuestión de cómo variar la granularidad de los componentes (elementos y tipos) para su reutilización. La elección de un patrón adecuado es un paso fundamental en la fase de diseño de esquemas. Antes de diseñar el esquema, debe decidirse el patrón.

23. Los patrones de diseño más comunes son *Russian Doll*, *Salami Slice*, *Venetian Blind* y *Garden of Eden*. Los patrones varían en función de su número de componentes globales (elementos o tipos). Para entender los patrones de diseño, es necesario diferenciar entre componentes globales y componentes locales. Un componente global es una consecuencia inmediata de la construcción de esquema en el archivo de definición de esquema de XML. Un componente local no es un derivado inmediato de la construcción de esquema en el archivo de definición de esquema de XML. Los componentes globales se asocian al espacio de nombres de destino del esquema y podrán reutilizarse en otro esquema. También es importante entender que todo elemento definido en el espacio de nombres global puede ser el origen de un documento de instancia XML válido conforme con el esquema definido para dicho espacio de nombres.

24. La mayoría de esquemas existentes adoptan *Venetian Blind* o *Garden of Eden* como patrón de selección porque son los más reutilizables. Desde el punto de vista de la disociación y la cohesión, *Venetian Blind* puede ser mejor que *Garden of Eden*; sin embargo, desde la perspectiva de la capacidad de reutilización, *Garden of Eden* puede ser mejor que *Venetian Blind* ya que utiliza tanto el tipo global como los elementos. El problema principal de un enfoque basado en el tipo (*Venetian Blind*) es el riesgo de elaborar un vocabulario de elementos incoherente cuyos elementos hayan sido declarados localmente y puedan ser reutilizados sin prestar atención a la claridad semántica y a la coherencia entre los tipos. Por ello, la Norma ST.96 de la OMPI recomienda el patrón de diseño *Garden of Eden*.

- [SD-04] DEBEN utilizarse los esquemas existentes cuando proceda antes de crear nuevos esquemas.
- [SD-05] Los esquemas DEBERÍAN utilizar los tipos y los elementos definidos en esquemas existentes siempre que sea posible.
- [SD-06] Todos los tipos, elementos y atributos DEBEN declararse globalmente.
- [SD-07] Los esquemas NO DEBEN utilizar la construcción `xsd:redefine`.

#### 2.5 Espacios de nombres

25. Los espacios de nombres se usan para identificar de manera única elementos y atributos con el mismo nombre cuando se combinan en un único documento. Los espacios de nombres asocian construcciones de esquema a un espacio conceptual que define un vocabulario de marcado. Una instancia XML contener nombres de elementos o atributos procedentes de más de un vocabulario de XML. Asignar a cada vocabulario un espacio de nombres puede permitir resolver la ambigüedad entre elementos o atributos con idéntico nombre. Los espacios de nombres deben ser exclusivos y persistentes. En la Norma ST.96 de la OMPI se recomienda una configuración de espacios de nombres múltiples.

- [SD-08] Los esquemas DEBEN utilizar espacios de nombres.

Anexo I, página 9

[SD-09] Las declaraciones publicadas de espacios de nombres NO DEBERÍAN modificarse.

2.5.1 Declaración y calificación de espacio de nombres

[SD-10] Los esquemas DEBEN declarar el espacio de nombres del esquema del W3C.

[SD-11] Los esquemas DEBEN utilizar calificaciones de espacio de nombres para todas las construcciones de esquema del W3C.

[SD-12] Los esquemas DEBEN utilizar el espacio de nombres en formato URI.

[SD-13] Todos los esquemas DEBEN tener unos atributos `elementFormDefault` y `attributeFormDefault` con un valor "qualified" en el elemento de la raíz `xsd:schema`.

26. En aras de la eficiencia, las Oficinas de P.I. podrán optar por reducir o incluso suprimir los espacios de nombres en la producción de sistemas de tratamiento de XML. No obstante, en el intercambio, deben respetarse las normas anteriormente expuestas.

2.5.2 Espacios de nombres en un esquema de XML

27. La Norma ST.96 define cuatro espacios de nombres: común, de patente, de marca y de dibujo o modelo. Esos espacios de nombres son simbólicos y se mantienen constantes en las distintas versiones.

28. En el ejemplo siguiente se muestra un espacio de nombres debidamente formado:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
.....
</xsd:schema>
```

[SD-14] Los nombres de los espacios de nombres DEBEN tener la siguiente estructura: `http://www.wipo.int/standards/XMLSchema/ST96/<category>`; donde `<category>` es un valor que identifica el ámbito del esquema que se utiliza: `Common`, `Patent`, `Trademark`, o `Design`.

[SD-15] Los esquemas DEBERÍAN utilizar "xsd" como prefijo de espacio de nombres en todas las construcciones de esquema del W3C, "com" para todos los esquemas de componente común, "pat" para todos los esquemas de componente de patente, "tmk" para todos los esquemas de componente de marca, "dgn" para todos los esquemas de componente de dibujos y modelos industriales, "tbl" para los esquemas de componente de cuadro OASIS y "mathml" para los esquemas de componente MathML.

2.5.3 Espacios de nombres de destino

29. Declarar un espacio de nombres de destino en un esquema garantiza que todas las construcciones del esquema se asociarán a dicho espacio de nombres. Sin un espacio de nombres de destino, las construcciones declaradas en el esquema no pertenecerían a ningún espacio de nombres. Toda vez que un esquema podrá tener más de un espacio de nombres declarado, solamente puede designarse un espacio de nombres como espacio de nombres de destino. En las recomendaciones del W3C, no se requiere declarar un espacio de nombre de destino en un esquema. No obstante, en la Norma ST.96 de la OMPI se recomienda utilizar un espacio de nombres de destino.

30. Un esquema declara un espacio de nombres de destino que coincide con uno de los espacios de nombres declarados. De acuerdo con el espacio de nombres asociado, los esquemas deben declarar uno de los siguientes cuatro espacios de nombres de destino definidos en la Norma ST.96 de la OMPI:

Anexo I, página 10

- <http://www.wipo.int/standards/XMLSchema/ST96/Common>, por ejemplo,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:com="http://www.wipo.int/standards/XMLSchema/Common"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/ST96/Patent>, por ejemplo,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
<xsd:include schemaLocation="xxx.xsd"/>
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/ST96/Trademark>, por ejemplo,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tmk="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Trademark"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/ST96/Design>, por ejemplo,

```
<?xml version="1.0" UTF-8?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dgn="http://www.wipo.int/standards/XMLSchema/ST96/Design"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Design"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
...
</xsd:schema>
```

- [SD-16] Todo esquema DEBE declarar el espacio de nombres de destino utilizando el atributo `xsd:targetNamespace`.
- [SD-17] El `targetNamespace` del esquema DEBE coincidir con el nombre del espacio de nombres de uno de los espacios de nombres declarados, pero no con el espacio de nombres del W3C.
- [SD-18] Los esquemas de componente común DEBEN utilizar como espacio de nombres de destino <http://www.wipo.int/standards/XMLSchema/ST96/Common>.
- [SD-19] Los esquemas de componente de patente DEBEN utilizar como espacio de nombres de destino <http://www.wipo.int/standards/XMLSchema/ST96/Patent/>.
- [SD-20] Los esquemas de componente de marca DEBEN utilizar como espacio de nombres de destino <http://www.wipo.int/standards/XMLSchema/ST96/Trademark/>.
- [SD-21] Los esquemas de componente de dibujos y modelos industriales DEBEN utilizar como espacio de nombres de destino <http://www.wipo.int/standards/XMLSchema/ST96/Design>.

#### 2.5.4 Espacios de nombres por defecto

31. Un espacio de nombres por defecto reduce la verbosidad de un esquema. No obstante, declarar un espacio de nombres por defecto en un esquema aumenta la ambigüedad ya que la omisión de un espacio de nombres dificulta más la identificación del espacio de nombres al que pertenece la construcción. La utilización de espacios de nombres por defecto puede limitar el espacio de nombres, haciendo que sea imposible discernir el origen de las construcciones del esquema

Anexo I, página 11

a partir del examen del esquema incluyente. En consecuencia, la Norma ST.96 de la OMPI desaconseja utilizar un espacio de nombres por defecto.

[SD-22] Los esquemas NO DEBERÍAN utilizar espacios de nombres por defecto.

## 2.6 Versiones del esquema

32. La compatibilidad con versiones anteriores plantea un problema cuando es preciso modificar esquemas ya existentes. En general, existe compatibilidad con versiones anteriores si un documento de instancia XML que se valida con arreglo a una versión anterior también se valida con arreglo a una nueva versión del esquema.

### 2.6.1 Cambios mayores y cambios menores

33. Los cambios en los esquemas pueden ser cambios mayores y cambios menores. Los cambios mayores en los esquemas son cambios incompatibles con versiones anteriores del esquema, mientras que los cambios menores son cambios compatibles con versiones anteriores. En las nuevas versiones menores de los esquemas, un documento resultante de una aplicación creado con una versión menor anterior de dicho esquema DEBE poder validarse con arreglo a la misma versión mayor. No obstante, no se debe esperar la validación de un documento resultante de una aplicación con versiones de esquema anteriores a la versión con la que se creó. Además, mientras el esquema está en fase de preparación, están permitidas las revisiones.

#### 2.6.1.1 Versiones mayores

34. En una versión mayor de un esquema, los cambios con respecto a versiones anteriores son significativos y/o incompatibles. Intentar validar una instancia XML basada en la versión mayor anterior del esquema con arreglo a una versión más reciente puede dar lugar a errores de validación. Se elaborará una nueva versión mayor cuando haya cambios significativos y/o incompatibles con versiones anteriores, por ejemplo:

- eliminación o modificación de valores en enumeraciones;
- modificación de nombres de elementos, nombres de tipos y nombres de atributos;
- supresión o adición de elementos o atributos obligatorios; y
- modificación del carácter facultativo de la cardinalidad para convertirlo en obligatorio.

#### 2.6.1.2 Versiones menores

35. En una versión mayor de un esquema conforme a la Norma ST.96 de la OMPI, puede haber una serie de cambios menores o compatibles con versiones anteriores. Las versiones menores de un esquema ST.96 determinan la compatibilidad de un esquema ST.96 con versiones menores anteriores o posteriores de la misma versión mayor. Así, el esquema de las versiones menores ayuda a establecer la compatibilidad con versiones anteriores o posteriores. Las versiones menores solamente se incrementarán cuando se produzcan cambios compatibles, por ejemplo:

- adición de valores a las enumeraciones,
- ampliaciones facultativas, y
- adición de elementos y/o atributos facultativos.

[SD-23] Las nuevas versiones menores de esquemas DEBEN poder validar todos los documentos resultantes de una aplicación creados con versiones menores anteriores de un esquema con la misma versión mayor.

[SD-24] La versión mayor del esquema DEBE incrementarse si el nuevo esquema no puede validar los documentos de instancia XML existentes creados con la versión mayor actual.

[SD-25] *Suprimida*

[SD-26] *Suprimida*

[SD-27] El número de la versión mayor y/o menor de un esquema DEBERÍA modificarse cuando se actualice un esquema incluido o importado.

[SD-28] Cuando una construcción de un esquema se vea alterada, la edición DEBE incrementarse conforme al mismo incremento del número de la versión.

Anexo I, página 12

[SD-29] Al crear un nuevo esquema DEBERÍAN utilizarse las versiones más recientes de todos los esquemas incluidos o importados.

### 2.6.2 Estrategia relativa a las versiones de un esquema

36. Las convenciones siguientes operan conjuntamente para definir la estrategia relativa a las versiones de un esquema:
- Estructura del fichero de la edición de esquema y convenciones sobre denominación de archivos de esquema;
  - utilización del atributo `version` integrado en el esquema de XML para todos los componentes;
  - utilización de un atributo `schemaVersion` definido por el usuario en el nivel de instancia de documento; y
  - utilización de un atributo `ipoVersion` definido por el usuario en el nivel de instancia de documento.

#### 2.6.2.1 Convenciones sobre denominación de ficheros y archivos en las versiones del esquema

37. La información sobre la versión DEBERÍA incluirse en el fichero, el nombre de archivo del esquema en el nivel de documento y el nombre de archivo del esquema aplanado. De acuerdo con las normas sobre denominación de archivos definidas en el presente documento, el fichero de la edición del esquema, el nombre de archivo de esquema en el nivel de documento y el nombre de archivo aplanado contienen tanto un número de versión mayor como un número de versión menor. En la estructura del fichero de la edición del esquema ST.96, el número de versión de la edición está seguido por los tipos de componente, es decir, común, patente, marca, dibujo o modelo y normas externas. Por ejemplo, la versión 2.0 del componente `contact` se sitúa en el fichero `ST96/V2_0/Common`.

[SD-64] El fichero de edición del esquema, un archivo de nombre de esquema en el nivel de documento y un archivo de nombre de esquema aplanado DEBEN contener información coincidente sobre la versión, que comprenda el número de versión mayor y menor.

#### 2.6.2.2 Atributo `version` integrado en el esquema de XML en las versiones del esquema

38. La especificación de esquema del W3C prevé la definición de un atributo `version` en el elemento raíz de un esquema. De conformidad con la Norma ST.96 de la OMPI, los esquemas deben utilizar el atributo `version` del esquema del W3C para incluir un número de versión mayor y un número de versión menor para cada archivo de esquema.

Por ejemplo,

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="unqualified" version=" V2_0">
...
</xsd:schema>
```

[SD-30] El atributo `version` del esquema del W3C DEBE estar compuesto por un número de versión mayor y un número de versión menor en cada archivo de esquema en el formato siguiente:  
"V"<número de versión mayor>"\_ "<número de versión menor>.

#### 2.6.2.3 Atributos `st96Version` e `ipoVersion` definidos por el usuario en las versiones del esquema para instancias XML

39. El atributo `st96Version` definido por el usuario es necesario para que un determinado documento de instancia remita directamente a una versión concreta del esquema XML de la Norma ST.96. Este atributo definirá claramente la versión del esquema ST.96 a la que apunta el documento de instancia o un esquema XML aplicado se remite. Este atributo debería declararse según sea necesario con un valor fijo.

40. El atributo `ipoVersion` definido por el usuario es necesario para que un determinado documento de instancia remita directamente a una versión concreta del esquema aplicado por una oficina de P.I. Este atributo definirá claramente la versión del esquema de la oficina de P.I. a la que apunta el documento de instancia. Este atributo debería declararse como facultativo y no llenarse cuando los componentes del esquema ST.96 se utilizan sin modificarlos.

41. Con miras a la compatibilidad con versiones anteriores o posteriores y al intercambio de datos entre las oficinas de P.I. para las que se utilizan distintas versiones menores, debería proporcionarse el número de la versión mayor y menor. Los ejemplos siguientes ilustran cómo pueden definirse los atributos en el esquema XML de la Norma ST.96, el esquema de aplicación de la oficina y la instancia XML.

Anexo I, página 13

42. El ejemplo siguiente ilustra cómo deberían definirse los atributos de `st96Version` e `ipoVersion` en el esquema ST.96 y en el esquema de aplicación de las oficinas de P.I.; y utilizado en una instancia XML:

`st96Version` de la versión 2.0 de la Norma ST.96:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
  <xsd:attribute name="st96Version" type="xsd:token" fixed="V2_0">
    ...
  </xsd:attribute>
</xsd:schema>
```

`ipoVersion` de la versión 2.0 de la Norma ST.96:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
  <xsd:attribute name="ipoVersion" type="xsd:token">
    ...
  </xsd:attribute>
</xsd:schema>
```

`ApplicationBodyType` de la versión 2.0 de la Norma ST.96:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../../Common/st96Version.xsd"/>
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../../Common/ipoVersion.xsd"/>
  .....
  <xsd:complexType name="ApplicationBodyType">
    .....
    <xsd:attribute ref="com:st96Version" use="required" />
    <xsd:attribute ref="com:ipoVersion"/>
    .....
  </xsd:complexType>
</xsd:schema>
```

`ipoVersion` y `ApplicationBodyType` de la versión 1.0 del esquema de aplicación de la USPTO, basada en `ApplicationBodyType` de la versión 2.0 de la Norma ST.96, *supra*:

```
<?xml version="1.0" encoding="UTF-8"?>
...
  <xsd:attribute name="ipoVersion" type="xsd:token" fixed="US_V1_0">
    ...
  </xsd:attribute>
...
```

Anexo I, página 14

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
targetNamespace="urn:us:gov:doc:uspto:patent"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V2_0">
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../../Common/st96Version.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../../Common/ipoVersion.xsd"/>
.....
<xsd:complexType name="ApplicationBodyType">
.....
<xsd:attribute ref="com:st96Version" use="required" />
<xsd:attribute ref="com:ipoVersion" />
.....
</xsd:complexType>
</xsd:schema>
```

Instancia XML basada en el elemento o `ApplicationBody` de la versión 1.0 del esquema de aplicación de la USPTO:

```
<?xml version="1.0" encoding="UTF-8"?>
<uspat:ApplicationBody xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:uspat="urn:us:gov:doc:uspto:patent"
xmlns:uscom="urn:us:gov:doc:uspto:common" "
xsi:schemaLocation="urn:us:gov:doc:uspto:patent
PE2E/V1_0/USPatent/Document/ApplicationBody_V1_0.xsd" com:st96Version="V2_0"
com:ipoVersion="US_V1_0">
.....
</uspat:ApplicationBody>
```

- [SD-31] El esquema para un componente de documento DEBE declarar un atributo requerido denominado `st96Version` en el elemento raíz con un valor fijo que coincida con la versión de la edición del esquema XML de la Norma ST.96, en el formato siguiente: "V"<número de versión mayor>"\_ "<número de versión menor>".
- [SD-63] El esquema para un componente de documento DEBE declarar un atributo facultativo `ipoVersion` en el elemento raíz con un valor fijo que coincida con la versión de la edición de aplicación de la oficina de P.I., en el formato siguiente: <Código ST.3>"\_ " "V"<número de la versión mayor>"\_ "<número de la versión menor>, por ejemplo, "US\_V2\_0".

## 2.7 Transformabilidad con otras normas sobre XML de la OMPI

43. Antes de la aprobación de la Norma ST.96 de la OMPI, las Oficinas de P.I. ya habían estado utilizando las Normas ST.36, ST.66 y ST.86 de la OMPI, por lo tanto, uno de los principales objetivos de la Norma ST.96 de la OMPI es mantener la transformabilidad entre la Norma ST.96 de la OMPI y las Normas ST.36, ST.66 y ST.86. A fin de garantizar el tratamiento satisfactorio de los datos a los fines de las necesidades operativas de las Oficinas de P.I. y de los proveedores de información de P.I., la norma ST.96 pretende alcanzar el grado necesario de transformabilidad con las Normas ST.36, ST.66 y ST.86 de la OMPI.

44. Se entenderá por *transformabilidad* que el conjunto de unidades atómicas de información (nivel más bajo de elementos de granularidad y ramas terminales del atributo, por lo general un elemento sin elementos derivados) que se podrían incluir en una instancia XML conforme a la Norma ST.96 puede transformarse en unidades atómicas de información que podrían incluirse en una instancia XML conforme a las Normas ST.36, ST.66 y ST.86 y viceversa.

45. Las listas de enumeraciones deben tener en cuenta el conjunto de valores permitidos previsto para las respectivas unidades atómicas de información disponibles en virtud de las Normas ST.36, ST.66 y ST.86 de la OMPI. Si fuere necesario, las listas de enumeraciones deben incluir los valores "Other", es decir que el valor que se proporciona en los datos originales no figura en la lista enumerada, y "Unknown", es decir que la información no figuraba o era incorrecta en los datos originales (aplicaciones de XML conformes a las Normas ST.36, ST.66 y ST.86). En la práctica, en algunas listas de enumeración se utiliza el valor "Undefined" en lugar de "Unknown".

Anexo I, página 15

- [SD-32] Los esquemas ST.96 DEBERÍAN elaborarse con el mismo nivel de cardinalidad y granularidad para facilitar la transformabilidad de las aplicaciones conformes a la Norma ST.96 y las aplicaciones conformes a las Normas ST.36, ST.66 o ST.86.
- [SD-33] Las enumeraciones contenidas en los esquemas NO DEBERÍAN incluir los valores "Other" o "Unknown" a menos que sea necesario.

### 2.8 Esquemas desarrollados por la industria

46. Cuando sea apropiado para el contenido de un documento, es decir, cuando el contenido no sea exclusivo del ámbito de la P.I., deberían utilizarse esquemas desarrollados por la industria. Los esquemas desarrollados por la industria recomendados en el presente documento son el esquema de cuadro de XML OASIS y MathML.
47. El esquema de cuadro OASIS, disponible en: <http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd>, es un modelo de esquema que hay que especificar más para satisfacer las necesidades operativas. En el esquema XML de la norma ST.96 se define OASISTable.xsd a partir del esquema OASIS Table.
48. Para las fórmulas matemáticas, se recomienda la utilización de MathML versión 3. Puede encontrarse más información sobre MathML, versión 3, en: <http://www.w3.org/TR/MathML3>.
49. En el momento de elaborar el presente documento no existe una recomendación del W3C ni un esquema de XML desarrollado por la industria para las estructuras o fórmulas químicas sobre el que exista un amplio consenso. De hecho, existen un gran número de normas de marcado de XML para distintos tipos químicos, pero no se puede considerar que ninguna de ellas goce de aceptación universal. Por este motivo, en el presente documento no se recomienda ninguna norma de XML específica para datos químicos.

- [SD-34] DEBERÍA utilizarse MathML, versión 3, para las fórmulas matemáticas.
- [SD-35] Para los cuadros, DEBERÍA utilizarse OASISTable.xsd, que se basa en el esquema de cuadro de intercambio de OASIS.
- [SD-36] Los esquemas desarrollados por la industria DEBERÍAN incorporarse solamente por referencia y una vez obtenida la aprobación del Comité de Normas Técnicas de la OMPI.

## 3. CONVENCIONES RELATIVAS A CONSTRUCCIONES DE ESQUEMAS DE XML

### 3.1 Definiciones de tipos

50. Los tipos representan la clase de información que pueden contener los elementos y los atributos, por ejemplo, cadenas de caracteres o fechas. A menudo, tipos y elementos van emparejados.

#### 3.1.1 Tipos simples

51. La utilización de tipos simples mejora la calidad de los datos en las aplicaciones de XML ya que todas las aplicaciones que usan tipos simples están sujetas a los mismos mecanismos de validación por medio de procesadores de XML. Si el formato léxico de un tipo simple no es el adecuado, los desarrolladores de esquemas pueden crear sus propios tipos de datos utilizando la sintaxis de expresiones regulares del esquema del W3C. Los tipos simples incluyen tanto tipos de datos integrados del W3C como tipos de datos definidos por el usuario.

##### 3.1.1.1 Tipos de datos integrados del W3C

52. Los tipos de datos integrados del W3C son los tipos de datos definidos por el W3C y que se incluyen en la norma sobre esquemas del W3C, por ejemplo, `date`, `Boolean`, `string` y `token`.

##### 3.1.1.2 Tipos de datos definidos por el usuario

53. Una de las ventajas de los esquemas de XML es su capacidad para crear tipos de datos definidos por el usuario. Los tipos de datos definidos por el usuario se basan en tipos de datos integrados del W3C ya existentes, aunque también pueden obtenerse por derivación a partir de tipos de datos definidos por el usuario ya existentes. Los tipos de datos definidos por el usuario pueden obtenerse por derivación de una de las tres maneras siguientes: `restriction`, `list` y `union`.

- [SD-37] Los esquemas DEBERÍAN utilizar tipos simples siempre que sea posible.
- [SD-38] Las listas de códigos DEBERÍAN declararse como lista de enumeración en un tipo simple.

Anexo I, página 16

- [SD-39] PUEDE declararse una lista de códigos como una unión de tipos simples.
- [SD-40] DEBEN utilizarse los códigos de dos letras de la Norma ST.3 de la OMPI para referirse a las Oficinas de P.I. y para el país/organización designado y prioritario. Por ejemplo, `PriorityCountryCode="EP"`.
- [SD-41] DEBE utilizarse la Norma ISO 3166-1-Alpha-2 *Code Elements (2-Letter Country Codes)* para referirse a nombres de países, dependencias y demás zonas de especial interés geopolítico, a partir de las listas de nombres de países de las Naciones Unidas.
- [SD-42] DEBE usarse la Norma ISO 639-1 (*2-Letter Language Codes*) para los códigos de idioma.
- [SD-43] Los esquemas DEBEN declarar elementos y atributos para valores de fecha y hora utilizando los tipos de datos de fecha y hora del esquema del W3C.
- [SD-44] DEBE usarse la Norma ISO 4217-Alpha (*3-Letter Currency Codes*) para los códigos de divisas.
- [SD-45] Los caracteres utilizados en las enumeraciones DEBEN limitarse al conjunto siguiente: 'a-z, A-Z, 0-9, espacio " " y guion bajo "\_". Los valores de enumeración NO DEBERÍAN comenzar con un carácter numérico.
- [SD-46] Los valores de enumeración DEBERÍAN ser semánticamente suficientes en inglés y emplear el menor número de caracteres posible. Los valores DEBERÍAN proceder del lenguaje común del ámbito de la P.I.

### 3.1.2 Tipos complejos

54. Los tipos complejos son tipos definidos por el usuario que contienen elementos derivados y/o atributos.

- [SD-47] PUEDEN usarse tipos complejos abstractos.

### 3.2 Elementos y atributos

55. Los elementos son las piezas básicas de un documento de instancia XML y se representan mediante etiquetas. Los atributos son construcciones de esquema del W3C asociadas a elementos que proporcionan más información sobre los elementos.

#### 3.2.1 Elementos frente a atributos

56. Una de las decisiones clave en el diseño de esquemas es si un elemento de datos debe representarse como un atributo o un elemento de XML. Mientras que los elementos contienen datos, los atributos contienen metadatos. En cuanto un elemento de datos se ha transformado en atributo, ya no puede ampliarse más, por lo tanto, los atributos solamente DEBERÍAN usarse para describir información que no puede o no podrá seguir ampliándose o subdividiéndose. Los esquemas DEBERÍAN diseñarse de manera que los elementos sean los depositarios principales de la información de P.I. en las aplicaciones de XML. Los atributos DEBERÍAN contener metadatos secundarios, es decir, elementos simples que proporcionan más información sobre el elemento.

- [SD-48] Los esquemas solamente DEBERÍAN utilizar atributos para definir datos no operativos. Por ejemplo, es aceptable utilizar un atributo denominado `sequenceNumber` para los números de serie.

#### 3.2.2 Elementos

##### 3.2.2.1 Cardinalidad de los elementos

57. Se entenderá por *cardinalidad* el número de elementos de un conjunto.

58. En un esquema, la cardinalidad se indica mediante las limitaciones `minOccurs` y `maxOccurs` en una declaración de elemento; estas limitaciones también se conocen como indicadores de ocurrencia. Los indicadores de ocurrencia no pueden aparecer en declaraciones de elemento global.

- [SD-49] Un indicador de ocurrencia NO DEBERÍA utilizarse para especificar una limitación que se dé por defecto en un esquema. Por ejemplo, `minOccurs="1"` y `maxOccurs="1"` NO DEBERÍAN utilizarse, pero es aceptable utilizar `minOccurs="2"` y `maxOccurs="3"`.

Anexo I, página 17

### 3.2.2.2 Elementos vacíos

59. Un elemento vacío es un elemento sin contenido de texto, sin elementos derivados y sin atributo. En general, la ausencia de un elemento en un esquema de XML no tiene un significado concreto; quizá indique que se desconoce la información, que dicha información no es aplicable o que dicha ausencia responde a otro motivo.

60. Aunque en la Norma ST.36 de la OMPI los elementos vacíos funcionan como una especie de indicador `Boolean` de la certeza o la falsedad de una condición, en función de la presencia o ausencia del elemento, en la Norma ST.96 se prescinde de esta práctica. En su lugar, deben crearse elementos que afirmen explícitamente la certeza o la falsedad, despejando así las dudas o las ambigüedades que puedan presentarse a una máquina o a una persona cuando lean la información.

[SD-50] Los elementos vacíos NO DEBEN definirse en los esquemas, a excepción del salto de línea.

### 3.2.3 Atributos

61. Los atributos son construcciones del esquema del W3C asociadas a elementos que proporcionan más información sobre los elementos. A diferencia de los elementos, los atributos no pueden formar parte de otro atributo, es decir, no existen subatributos. Los atributos no se pueden ampliar, a diferencia de lo que sucede con los elementos.

62. La cardinalidad en el caso de los atributos es distinta de la cardinalidad en el caso de los elementos. En el caso de los atributos, al indicador `use` se le puede asignar uno de los tres valores siguientes: "required", "optional" o "prohibited".

[SD-51] El atributo `use` en uso de atributo DEBERÍA omitirse cuando el atributo al que se hace referencia es facultativo, puesto que por defecto se utiliza `use="optional"`.

### 3.2.4 Agrupación de elementos y atributos

63. Las instrucciones de composición son construcciones del esquema del W3C que agrupan declaraciones de elementos. En el esquema del W3C, existen tres tipos de operadores: `sequence`, `choice` y `all`.

64. La instrucción `sequence` indica que los elementos declarados deben aparecer en un documento de instancia XML en el orden en que han sido declarados. La instrucción `sequence` permite velar por el orden de los elementos.

65. La instrucción `choice` indica que solamente uno de los elementos declarados puede aparecer en un documento de instancia XML.

66. En algunos casos, los indicadores de ocurrencia de las instrucciones `sequence` y `choice` permiten la flexibilidad de utilizar una instrucción `sequence/choice`, es decir instrucciones `sequence/choice` compositores optativas o múltiples, para no tener que definir un nivel adicional.

67. La instrucción `all` indica que los elementos declarados pueden aparecer solamente una vez en un documento de instancia XML, en cualquier orden. En el caso de la instrucción `all`, ninguno de los elementos puede aparecer más de una vez y todos los elementos son optativos si el valor del atributo `minOccurs` se fija en 0. Para el intercambio de datos, al menos un elemento debe ser obligatorio. Por lo tanto, no se recomienda utilizar la instrucción `all` en los esquemas.

[SD-52] Los esquemas NO DEBERÍAN utilizar la instrucción `all`.

[SD-53] Los esquemas PUEDEN utilizar indicadores de ocurrencia en las instrucciones de composición `xsd:sequence` y `xsd:choice`.

## 3.3 Ampliación y restricción

68. Algunas técnicas autorizadas en las recomendaciones del W3C pueden afectar a la armonización de datos ya que añaden un cierto margen de maniobra a la hora de proporcionar datos estructurados. Estas técnicas son la ampliación, la restricción y los grupos de sustitución.

### 3.3.1 Ampliación

69. La ampliación de tipos complejos no afecta a la armonización de datos, a diferencia de lo que sucede con la ampliación de tipos simples, ya que los componentes añadidos no se intercambiarán. Sin embargo, derivar tipos simples mediante el uso de las instrucciones `union` o `list` dificulta la interacción ya que propicia la aparición de nuevos valores que no están gestionados por el formato de intercambio de datos.

[SD-54] Un esquema NO DEBE contener un tipo complejo con `xsd:any` en lo relativo a la capacidad de ampliación.

Anexo I, página 18

### 3.3.2 Restricción

70. En el contexto de la armonización de datos, la restricción de tipos complejos modifica la estructura y puede provocar problemas de compatibilidad que bloqueen el intercambio de datos. En el caso de los tipos simples, la restricción modifica los valores; no se prevén problemas en el intercambio de datos.

### 3.3.3 Grupos de sustitución

71. Los grupos de sustitución permiten que un elemento global sustituya a otro elemento global en un documento de instancia XML sin necesidad de introducir más modificaciones adicionales en el esquema. No obstante, los grupos de sustitución no fomentan la armonización de los nombres de elementos. La armonización es un factor clave para el intercambio interactivo de datos, y la utilización de grupos de sustitución es incompatible con la armonización.

[SD-55] Los esquemas NO DEBEN utilizar grupos de sustitución.

### 3.4 Limitaciones de identidad

72. Como sucede con cualquier sistema de almacenado, un documento de XML necesita ofrecer maneras de identificar los fragmentos de información que contiene y referirse a ellos. Con arreglo al esquema de XML del W3C, ello puede hacerse mediante dos funciones, `xsd:key/unique/keyref` y `xsd:ID/IDREF/IDREFS`.

73. Para las referencias dentro del documento XML actual, `xsd:ID/IDREF/IDREFS` PUEDE coexistir con `xsd:key/unique/keyref` en la misma instancia XML. El `xsd:key/unique/keyref` DEBERÍA utilizarse para cada componente dentro de un ámbito, cuando la unicidad deba mantenerse enteramente dentro de ese ámbito. `xsd:ID`, `xsd:IDREF` y `xsd:IDREFS` DEBERÍAN utilizarse en las instancias XML en las que la unicidad y la referencias a la unicidad se exigen en la totalidad del documento. Los atributos `id`, `idref` e `idrefs` están definidos en el espacio de nombres *Common* por `xsd:ID`, `xsd:IDREF` y `xsd:IDREFS`, respectivamente. El atributo `idrefs` DEBERÍA utilizarse cuando se prevén múltiples referencias, por ejemplo, en *Claims*.

74. Para las referencias externas al documento XML actual, DEBERÍA utilizarse el atributo `extRef` definido como `xsd:token` en el espacio de nombres *Common*, por ejemplo *Cross Reference*.

[SD-56] Los esquemas DEBERÍAN utilizar `xsd:key/xsd:keyref/xsd:unique` y/o `xsd:ID/xsd:IDREF/xsd:IDREFS` para identificar las limitaciones, según corresponda.

[SD-57] Los esquemas DEBERÍAN utilizar `xsd:ID/xsd:IDREF/xsd:IDREFS` para una única referencia o múltiples referencias dentro del documento XML actual para identificar las limitaciones, según corresponda.

[SD-62] Los esquemas DEBERÍAN utilizar `xsd:key/xsd:unique/xsd:keyref` dentro de un ámbito, cuando la unicidad debe mantenerse enteramente dentro de ese ámbito.

### 3.5 Documentación del esquema

75. La especificación de esquema del W3C permite incluir documentación en los esquemas, hecho que permite que los esquemas incorporen información documental sobre cada construcción. Cada construcción de esquema definida o declarada DEBE utilizar el elemento `xsd:documentation` para la documentación exigida y, de ser necesario, el elemento `xsd:appInfo` para la información sobre la aplicación dentro del elemento `xsd:annotation`. El elemento `xsd:appInfo` comunicar el contexto y la utilización de elementos de un esquema a un usuario que quizá no tenga la misma formación o los mismos conocimientos técnicos que el desarrollador del esquema. Por lo general, el elemento `xsd:documentation` se utiliza para el material legible por personas y el elemento `xsd:appInfo` se utiliza para proporcionar información para las herramientas, hojas de estilo y otras aplicaciones.

#### 3.5.1 Documentación sobre la construcción del esquema

76. Los esquemas de XML también podrán contener comentarios (es decir, `<!-- comment -->`). Se recomienda a los desarrolladores de esquemas que eviten utilizar esta técnica, sobre todo si la finalidad del comentario es describir algún aspecto de una determinada construcción del esquema.

77. En ocasiones, los comentarios en estilo HTML son aceptables, por ejemplo cuando el desarrollador desea insertar delimitadores visuales entre construcciones como elementos y tipos.

[SD-58] Todos los esquemas DEBERÍAN utilizar el elemento `xsd:documentation` para incluir documentación sobre la construcción del esquema. La documentación solamente DEBERÍA

Anexo I, página 19

describir el elemento o el tipo, y NO DEBERÍA contener detalles sobre la aplicación u otra información que no esté directamente relacionada con el significado de la construcción.

[SD-59] NO DEBERÍAN utilizarse comentarios (es decir, `<!--comment -->`) en un esquema.

[SD-60] La documentación NO DEBERÍA sustituirse por listas de códigos que utilicen la enumeración.

### 3.5.2 Documentación del encabezado del esquema

78. Si la documentación sobre la construcción del esquema añade claridad a un esquema, la documentación del encabezado del esquema permite concentrar determinada información —finalidad, uso y contenidos del esquema— en un único lugar dentro del esquema. La documentación del encabezado del esquema debería incluirse en todos los elementos del nivel de documento y algunos componentes de nivel agregado y básico, como WIPOST3Code. Para facilitar el tratamiento de la información del encabezado por las aplicaciones, debería usarse el elemento `xsd:appInfo` para la documentación del encabezado del esquema.

79. En el Cuadro 1 se muestran los ítems que se incluyen en la sección de encabezado de los esquemas.

Cuadro 1. Información sobre la documentación del encabezado

Nombre del elemento del encabezado	Descripción
"Created date"	Fecha de creación del esquema (obligatorio)
"Last modified date"	Fecha de la última modificación del esquema (obligatorio)
"Contact Point"	Coordinador que se puede contactar en caso de dudas sobre el esquema (obligatorio)
"Release Notes URL"	Un URL en el que se publican las notas explicativas del esquema (opcional)

En el ejemplo siguiente se ilustran os elementos de los esquemas correspondientes a los ítems de encabezado expuestos *supra*:

```
<xsd:annotation>
<xsd:appinfo> <com:SchemaCreatedDate>2011-05-28</com: SchemaCreatedDate>
  <com:SchemaLastModifiedDate>2011-05-28</com: SchemaLastModifiedDate>
  <com: SchemaContactPoint>xml.standards@wipo.int</com: SchemaContactPoint >
  <com:
SchemaReleaseNoteURL>http://www.wipo.int/standards/XMLSchema/ST96/V2_0/ReleaseNotes
.pdf</com:SchemaReleaseNoteURL>
</xsd:appinfo></xsd:annotation>
```

80. La documentación del encabezado del esquema permite que el desarrollador de un esquema identifique fácilmente la finalidad, el uso y los contenidos del esquema. Esta información también es muy útil cuando el desarrollador de un esquema debe elegir un esquema para utilizarlo como modelo para crear otro esquema.

[SD-61] Los esquemas de nivel de documento DEBEN incluir documentación del encabezado del esquema que contenga referencias a la fecha de creación, la fecha de última modificación y el punto de contacto, con carácter obligatorio, y al URL sobre las notas explicativas, con carácter opcional.

## 4. NORMAS DE DISEÑO DE LA INSTANCIA

81. Las normas de diseño de la instancia se ocupan de las recomendaciones que hay que seguir al proporcionar datos estructurados.

### 4.1 Espacios de nombres en documentos de instancia XML

Anexo I, página 20

4.1.1 Validación de un documento de instancia XML

82. Se recomienda validar un documento de instancia XML con un esquema. La validación de un documento de instancia XML garantiza que el contenido cumple todos los requisitos del esquema que se ha utilizado para validarlo. El sistema de validación selecciona el esquema a partir de su ubicación exacta, tal y como se especifica en el documento de instancia XML.

83. La ubicación del esquema debe facilitarse como URI. Por lo general, la ubicación del esquema se define como el URI basado en el URL, dadas las limitaciones de resolución de los URI basados en URN.

84. La ubicación del esquema se puede facilitar en el elemento raíz del documento de instancia XML, tal y como se muestra a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<com:Contact xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/ST96/Common
../V2_0/Common/Contact.xsd">
.....
</com:Contact>
```

85. El atributo `schemaLocation` es una construcción del esquema del W3C que asocia un documento de instancia XML a un esquema. Se utiliza solamente cuando el esquema tiene un espacio de nombres de destino. En el ejemplo anterior, el identificador de espacio de nombres "http://www.wipo.int/standards/XMLSchema/ST96/Common" es el espacio de nombres de destino del esquema `Contact.xsd`.

[ID-01] Las instancias XML DEBEN validarse con el esquema correspondiente.

[ID-02] Los documentos de instancia XML DEBEN utilizar el atributo `schemaLocation` para asociar el espacio de nombres de destino con la localización del archivo del esquema al que se conforma la instancia XML.

[ID-03] Toda declaración de atributo `xsi:schemaLocation` DEBERÍA contener un URL resoluble.

4.1.2 Declaración y calificación del espacio de nombres en documentos de instancia XML

86. Como sucede con los esquemas, la declaración de un espacio de nombres en el elemento raíz de un documento de instancia XML se realiza utilizando un identificador de espacio de nombres y un prefijo de espacio de nombres recomendado. Los documentos de instancia XML deben usar una calificación de espacio de nombres para todos los elementos definidos en el Anexo III de la Norma ST.96 de la OMPI.

87. Conviene señalar que:

- el identificador de espacio de nombres en un documento de instancia XML debe ser el mismo que el identificador de espacio de nombres del espacio de nombres de destino del esquema; y
- el prefijo de espacio de nombres en un documento de instancia XML no tiene por qué ser el mismo que el prefijo de espacio de nombres para el espacio de nombres de destino del esquema.

88. Los elementos y atributos de un documento de instancia XML solamente pueden ser aptos para un espacio de nombres si pertenecen al espacio de nombres de destino del esquema que valida el documento de instancia XML. Por ello, todos los elementos y atributos globales deben ser aptos para un espacio de nombres. No obstante, el requisito para los elementos y atributos locales que pertenecen al espacio de nombres de destino del esquema depende de la creación de un mecanismo de conmutación en el esquema que utilice los dos indicadores siguientes:

- `elementFormDefault`, y
- `attributeFormDefault`.

89. El indicador `elementFormDefault` controla la calificación como espacio de nombres de los elementos locales, mientras que el indicador `attributeFormDefault` controla la calificación como espacio de nombres de los atributos locales. Ambos indicadores aparecen como atributos del elemento raíz de un esquema, y el valor de cada uno de ellos puede ser `qualified` o `unqualified` (por defecto).

Anexo I, página 21

Por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified"
attributeFormDefault="qualified">
.....
</xsd:schema>
```

90. En estas declaraciones, todos los elementos y atributos locales del espacio de nombres de destino del esquema deben ser aptos para un espacio de nombres en un documento de instancia XML, salvo que se altere el valor del atributo `form`.

[ID-04] Todo espacio de nombres declarado DEBERÍA asignarse a un prefijo recomendado, por ejemplo, el prefijo "com" para todas las construcciones del esquema de componentes comunes, "pat" para todas las construcciones del esquema de componente de patente, "tmk" para todas las construcciones del esquema de componente de marca, "dgn" para todas las construcciones del esquema de componente de dibujos y modelos industriales, "tbl" para las construcciones del esquema de componente del cuadro OASIS y "mathml" para las construcciones del esquema de componente de MathML.

[ID-05] Los documentos de instancia XML NO DEBEN utilizar espacios de nombres por defecto.

#### 4.1.3 Espacio de nombres de una aplicación del esquema del W3C

91. El esquema del W3C posee su propio espacio de nombres, conocido como espacio de nombres de la aplicación del esquema del W3C, y contiene todas las construcciones del esquema del W3C utilizadas en documentos de instancia XML (`schemaLocation`, `noNamespaceSchemaLocation`, `type`, y `nil`). Para utilizar dichas construcciones, debe declararse el espacio de nombres de la aplicación del esquema del W3C en el elemento raíz de un documento de instancia XML mediante el identificador de espacio de nombres "http://www.w3.org/2001/XMLSchema-instance":

```
<?xml version="1.0" encoding="UTF-8"?>
<com:Contact xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common">
...
</com:Contact>
```

92. Aunque se trata de un prefijo definido por el usuario, el prefijo `xsi` se suele utilizar en la documentación y en el material de referencia del esquema del W3C como prefijo de espacio de nombres para las construcciones de aplicaciones del esquema del W3C.

[ID-06] Los documentos de instancia XML DEBERÍAN utilizar `xsi` como prefijo de espacio de nombres en todas las construcciones de aplicaciones del esquema del W3C.

#### 4.1.4 Alcance del espacio de nombres

93. Los espacios de nombres en los documentos de instancia XML tienen un alcance de aplicabilidad en un documento de instancia XML. El alcance de un espacio de nombres se aplica al elemento declarado (que podrá ser el elemento raíz) y a todo el contenido de dicho elemento. También puede declararse un espacio de nombres en un elemento distinto al elemento raíz, lo que se conoce como declaración de espacio de nombres local.

94. Aunque la utilización de declaraciones de espacio de nombres local puede mejorar el tratamiento, reviste más valor la capacidad para identificar visualmente todos los espacios de nombres declarados en un documento resultante de una aplicación mediante un examen del elemento raíz.

[ID-07] Los documentos de instancia XML NO DEBEN utilizar declaraciones de espacio de nombres local.

Anexo I, página 22

#### 4.2 Entidades externas

95. Las imágenes integradas son una entidad externa de uso común, es decir, una referencia a un archivo de imagen externo insertada en un documento resultante de una aplicación en el lugar en el que debería aparecer la imagen cuando se procese el documento. A menudo, las imágenes integradas son partes de documentos que no se pueden codificar y almacenar utilizando un conjunto de caracteres. Las imágenes integradas PUEDEN ser dibujos, fórmulas químicas, cuadros complejos, caracteres no definidos, etcétera.

96. Una instancia XML podrá remitir a una entidad externa que sea otro archivo de XML basado en una DTD. En ese caso, el esquema de XML correspondiente no sirve para validar la instancia XML. Para referirse al archivo de XML se podría utilizar `xsd:anyURI` o `xsd:string` para señalar la ubicación del archivo externo de la instancia XML. Sin embargo, en el caso de instancias XML, no se recomienda remitirse al archivo de XML basado en una DTD.

97. Las imágenes pueden estar integradas en una instancia XML como si fueran imágenes binarias integradas codificadas en `base64Binary`, el tipo de datos integrado del W3C, o como si fueran referencias a archivos de imagen externos, es decir, entidades externas. No obstante, en el presente documento las imágenes se calificarán como entidades externas ya que las imágenes binarias integradas pueden contener código dañino, es decir, un virus.

[ID-08] Los archivos de imagen externos DEBERÍAN ajustarse a uno de los formatos de imagen recomendados en la Norma ST.96 de la OMPI.

[ID-09] Las imágenes DEBEN calificarse como archivos externos.

[ID-10] Las listas de secuencias DEBERÍAN seguir la Norma ST.25 de la OMPI.

Anexo I, página 23

**APÉNDICE A – RESUMEN DE LAS NORMAS DE DISEÑO**

*Normas generales de diseño*

ID de la Norma	Norma
[GD-01]	Todos los esquemas de XML DEBEN basarse en las especificaciones técnicas del W3C que han alcanzado la condición de Recomendación.
[GD-02]	Los esquemas DEBEN ajustarse a Esquema de XML, Parte 1: Estructuras ( <a href="http://www.w3.org/TR/xmlschema-1/">http://www.w3.org/TR/xmlschema-1/</a> ) y Esquema de XML, Parte 2: Tipos de datos ( <a href="http://www.w3.org/TR/xmlschema-2/">http://www.w3.org/TR/xmlschema-2/</a> ).
[GD-03]	En los esquemas DEBE usarse el conjunto de caracteres ISO/IEC 10646 – UCS – Unicode. DEBE usarse UTF-8 para codificar los caracteres Unicode.
[GD-04]	Los nombres de tipos, elementos y atributos DEBEN estar compuestos de palabras del idioma inglés, utilizando la grafía inglesa básica que figura en el Diccionario Inglés Oxford. Las únicas excepciones permitidas son las siglas, las abreviaturas y otras formas truncadas que se enumeran en el Apéndice C.
[GD-05]	Los nombres de tipos, elementos y atributos solamente DEBERÍAN estar compuestos por nombres, adjetivos y verbos en presente, con la excepción de las siglas, abreviaturas y otras formas truncadas que se enumeran en el Apéndice C.
[GD-06]	Los caracteres empleados en los nombres de tipos, elementos y atributos DEBEN estar contenidos en el conjunto siguiente: 'a-z, A-Z y 0-9'.
[GD-07]	La longitud máxima de un nombre de componente DEBERÍA ser de no más de 35 caracteres.
[GD-08]	Los nombres de tipos, elementos y atributos DEBERÍAN ser concisos y explicarse por sí solos.
[GD-09]	Los nombres de elementos DEBEN seguir la convención <i>upper camel case</i> (UCC). Por ejemplo, <code>CountryCode</code> .
[GD-10]	Los nombres de tipos DEBEN seguir la convención UCC y poseer el sufijo <code>Type</code> . Por ejemplo, <code>ApplicantType</code> .
[GD-11]	Los nombres de atributos DEBEN seguir la convención <i>lower camel case</i> (LCC). Por ejemplo, <code>currencyCode="EUR"</code> .
[GD-12]	Las siglas y las abreviaturas que figuran en el Apéndice C siempre DEBEN utilizarse en lugar del nombre largo ampliado.
[GD-13]	Las siglas y las abreviaturas que aparecen al principio de una declaración de atributos DEBEN estar en minúscula en su totalidad. El resto de siglas y abreviaturas utilizadas en una declaración de atributos DEBEN aparecer en mayúscula, según se indica en el Apéndice C.
[GD-14]	Las siglas y las abreviaturas DEBEN aparecer según lo indicado en el Apéndice C para los nombres de elementos y tipos.
[GD-15]	Los nombres de tipos complejos DEBERÍAN incluir un término de clase de objetivo significativo.
[GD-16]	Cuando se necesitan dos términos de clase de objeto en un determinado nombre de componente debido a la naturaleza de la entidad, podrá utilizarse un tipo complejo de asociación. En los nombres de tipo complejo de asociación DEBERÍA utilizarse una estructura de clase de objeto del tipo complejo de asociación, un término de propiedades que describa la naturaleza de la asociación, y la clase de objeto del tipo complejo asociado. Los términos de clase de objeto y de propiedades pueden estar precedidos por calificadores. Por ejemplo, <code>ApplicantResidenceAddress</code> , <code>Applicant</code> es la clase de objeto del tipo complejo de asociación, <code>Residence</code> es la propiedad y <code>Address</code> es la clase de objeto del tipo complejo asociado.
[GD-17]	Los nombres de elementos básicos (los que se basan en un tipo simple o complejo que no admite elementos derivados) DEBERÍAN componerse de un término de clase de objeto, término de propiedades y, cuando corresponda, un término de representación. Los términos de clase de objeto y de propiedades pueden estar precedidos por calificadores.
[GD-18]	Un término de clase de objeto siempre DEBE tener el mismo significado semántico en todo un espacio de nombres y PUEDE componerse de más de una palabra. Por ejemplo, <code>ContactInformation</code> .
[GD-19]	Un término de propiedades de un nombre DEBE ser exclusivo en el contexto de una clase de objeto pero PUEDE reutilizarse en distintas clases de objetos.

Anexo I, página 24

ID de la Norma	Norma
[GD-20]	Un término calificador PUEDE adjuntarse a un término de clase de objeto o a un término de propiedades si fuera necesario para individualizar un nombre.
[GD-21]	Cuando un nombre contiene un término de clase de objeto, un término de propiedades y un término de representación, el término de clase de objeto DEBE anteceder al término de propiedades y el término de propiedades DEBE anteceder al término de representación. Un término calificador DEBERÍA anteceder al término de propiedades o al término de clase de objeto asociado.
[GD-22]	Si el término de propiedades acaba con la misma palabra que el término de representación (o una palabra equivalente), DEBE suprimirse el término de representación.
[GD-23]	Cuando es necesario un término de representación, DEBEN emplearse los términos de representación del Apéndice B para los términos de representación en los nombres de componentes básicos.
[GD-24]	En un espacio de nombres, todos los nombres de tipos, elementos y atributos DEBEN ser únicos.
[GD-25]	La palabra o palabras de un nombre DEBERÍAN aparecer en singular salvo que el concepto mismo sea plural. Por ejemplo: <code>TotalMarkSeries</code>
[GD-26]	El nombre de un elemento o un tipo que contenga una colección de componentes relacionados por el contexto DEBERÍA poseer el sufijo "Bag". Por ejemplo, <code>EmailAddressBag</code> representa una colección de elementos <code>EmailAddress</code> .
[GD-27]	Conectores como "and", "of" o "the" NO DEBERÍAN utilizarse en nombres de tipos, elementos y atributos salvo que formen parte de la terminología de la actividad.
[GD-28]	Los nombres de tipos, elementos y atributos NO DEBEN traducirse, modificarse o sustituirse con ningún fin.
[GD-29]	Los nombres de tipos y elementos NO DEBEN remitir a números de artículos o normas. Por ejemplo, <code>PCTRule702C</code> para el PCT.
[GD-30]	Los caracteres utilizados en el nombre de archivo de esquema DEBEN pertenecer al conjunto siguiente: a-z, A-Z, 0-9, guion bajo "_", y punto ".".
[GD-31]	El nombre de archivo del esquema DEBE constar de dos partes obligatorias con un delimitador e información facultativa sobre la versión con dos delimitadores adicionales, es decir: <nombre del componente>{"_" "V"<número de versión mayor>"_"<número de versión menor>"}."<extensión del archivo>; por ejemplo, <code>EmailAddressType.xsd</code> , <code>languageCode.xsd</code> , <code>ApplicationBody_V1_0.xsd</code> .
[GD-32]	Un nombre de archivo de proyecto de esquema DEBE constar de cuatro partes obligatorias con dos delimitadores e información facultativa sobre la versión con dos delimitadores adicionales, es decir: <nombre del componente>{"_" "V"<número de versión mayor>"_"<número de versión menor>"}_"D"<número de revisión>"}."<extensión del archivo>; por ejemplo, <code>Contact_D3.xsd</code> , <code>TrademarkApplication_1_V1_D1</code> . Si un proyecto de esquema se basa en un esquema existente y tiene información sobre la versión en su archivo de nombre, los números de la versión mayor y menor del archivo de nombre del proyecto de esquema DEBERÍAN ser los mismos indicados en el archivo de esquema en que se basa el proyecto de esquema. Si un proyecto de esquema es nuevo, el número de versión mayor en el nombre de archivo de proyecto de esquema DEBERÍA ser el mismo número indicado en el espacio de nombres correspondiente, y un número de versión menor en el proyecto de esquema DEBERÍA ser cero "0".

*Normas de diseño de esquema*

ID de la Norma	Norma
[SD-01]	Los módulos de esquema de componente de patente, los módulos de esquema de componente de marca y los módulos de esquema de componente de dibujos y modelos industriales DEBEN utilizar la construcción <code>xsd:import</code> para remitirse a módulos de esquema de componente común.
[SD-02]	Los módulos de esquema de componente común NO DEBEN remitirse a los módulos de esquema de componente de patente, a los módulos de esquema de componente de marca o a los módulos de esquema de componente de dibujos y modelos industriales.

Anexo I, página 25

ID de la Norma	Norma
[SD-03]	Un módulo de esquema de componente de patente, de componente de marca o de componente de dibujo o modelo industrial PUEDE remitirse a módulos de esquema de componente común, esquemas del mismo módulo de componente y esquemas desarrollados por la industria aprobados, pero NO DEBEN remitirse a otros módulos de esquema de componente. Por ejemplo, los esquemas de patente NO DEBEN remitirse a esquemas de marca y viceversa.
[SD-04]	DEBEN utilizarse los esquemas existentes cuando proceda antes de crear nuevos esquemas.
[SD-05]	Los esquemas DEBERÍAN utilizar los tipos y los elementos definidos en esquemas existentes siempre que sea posible.
[SD-06]	Todos los tipos, elementos y atributos DEBEN declararse globalmente.
[SD-07]	Los esquemas NO DEBEN utilizar la construcción <code>xsd.redefine</code> .
[SD-08]	Los esquemas DEBEN utilizar espacios de nombres.
[SD-09]	Las declaraciones publicadas de espacios de nombres NO DEBERÍAN modificarse.
[SD-10]	Los esquemas DEBEN declarar el espacio de nombres del esquema del W3C.
[SD-11]	Los esquemas DEBEN utilizar calificaciones de espacio de nombres para todas las construcciones de esquema del W3C.
[SD-12]	Los esquemas DEBEN utilizar el espacio de nombres en formato URI.
[SD-13]	Todos los esquemas DEBEN tener unos atributos <code>elementFormDefault</code> y <code>attributeFormDefault</code> con un valor "qualified" en el elemento de la raíz <code>xsd:schema</code> .
[SD-14]	Los nombres de los espacios de nombres DEBEN tener la siguiente estructura: <code>http://www.wipo.int/standards/XMLSchema/ST96/&lt;category&gt;</code> ; donde <code>&lt;category&gt;</code> es un valor que identifica el ámbito del esquema que se utiliza: <code>Common, Patent, Trademark, o Design</code> .
[SD-15]	Los esquemas DEBERÍAN utilizar "xsd" como prefijo de espacio de nombres en todas las construcciones de esquema del W3C, "com" para todos los esquemas de componente común, "pat" para todos los esquemas de componente de patente, "tmk" para todos los esquemas de componente de marca, "dgn" para todos los esquemas de componente de dibujos y modelos industriales, "tbl" para los esquemas de componente de cuadro OASIS y "mathml" para los esquemas de componente MathML.
[SD-16]	Todo esquema DEBE declarar el espacio de nombres de destino utilizando el atributo <code>xsd:targetNamespace</code> .
[SD-17]	El <code>targetNamespace</code> del esquema DEBE coincidir con el nombre del espacio de nombres de uno de los espacios de nombres declarados, pero no con el espacio de nombres del W3C.
[SD-18]	Los esquemas de componente común DEBEN utilizar como espacio de nombres de destino <code>http://www.wipo.int/standards/XMLSchema/ST96/Common</code> .
[SD-19]	Los esquemas de componente de patente DEBEN utilizar como espacio de nombres de destino <code>http://www.wipo.int/standards/XMLSchema/ST96/Patent/</code> .
[SD-20]	Los esquemas de componente de marca DEBEN utilizar como espacio de nombres de destino <code>http://www.wipo.int/standards/XMLSchema/ST96/Trademark/</code> .
[SD-21]	Los esquemas de componente de dibujos y modelos industriales DEBEN utilizar como espacio de nombres de destino <code>http://www.wipo.int/standards/XMLSchema/ST96/Design</code> .
[SD-22]	Los esquemas NO DEBERÍAN utilizar espacios de nombres por defecto.
[SD-23]	Las nuevas versiones menores de esquemas DEBEN poder validar todos los documentos resultantes de una aplicación creados con versiones menores anteriores de un esquema con la misma versión mayor.
[SD-24]	La versión mayor del esquema DEBE incrementarse si el nuevo esquema no puede validar los documentos de instancia XML existentes creados con la versión mayor actual.
[SD-25]	<i>Suprimida</i>
[SD-26]	<i>Suprimida</i>
[SD-27]	El número de la versión mayor y/o menor de un esquema DEBERÍA modificarse cuando se actualice un esquema incluido o importado.

Anexo I, página 26

ID de la Norma	Norma
[SD-28]	Cuando una construcción de un esquema se vea alterada, la edición DEBE incrementarse conforme al mismo incremento del número de la versión.
[SD-29]	Al crear un nuevo esquema DEBERÍAN utilizarse las versiones más recientes de todos los esquemas incluidos o importados.
[SD-30]	El atributo <code>version</code> del esquema del W3C DEBE estar compuesto por un número de versión mayor y un número de versión menor en cada archivo de esquema en el formato siguiente: "V"<número de versión mayor>"_ "<número de versión menor>.
[SD-31]	El esquema para un componente de documento DEBE declarar un atributo requerido denominado <code>st96Version</code> en el elemento raíz con un valor fijo que coincida con la versión de la edición del esquema XML de la Norma ST.96, en el formato siguiente: "V"<número de versión mayor>"_ "<número de versión menor>.
[SD-32]	Los esquemas ST.96 DEBERÍAN elaborarse con el mismo nivel de cardinalidad y granularidad para facilitar la transformabilidad de las aplicaciones conformes a la Norma ST.96 y las aplicaciones conformes a las Normas ST.36, ST.66 o ST.86.
[SD-33]	Las enumeraciones contenidas en los esquemas NO DEBERÍAN incluir los valores "Other" o "Unknown" a menos que sea necesario.
[SD-34]	DEBERÍA utilizarse MathML, versión 3, para las fórmulas matemáticas.
[SD-35]	Para los cuadros, DEBERÍA utilizarse OASISTable.xsd, que se basa en el esquema de cuadro de intercambio de OASIS.
[SD-36]	Los esquemas desarrollados por la industria DEBERÍAN incorporarse solamente por referencia y una vez obtenida la aprobación del Comité de Normas Técnicas de la OMPI.
[SD-37]	Los esquemas DEBERÍAN utilizar tipos simples siempre que sea posible.
[SD-38]	Las listas de códigos DEBERÍAN declararse como lista de enumeración en un tipo simple.
[SD-39]	PUEDE declararse una lista de códigos como una unión de tipos simples.
[SD-40]	DEBEN utilizarse los códigos de dos letras de la Norma ST.3 de la OMPI para referirse a las Oficinas de P.I. y para el país/organización designado y prioritario. Por ejemplo, <code>PriorityCountryCode="EP"</code> .
[SD-41]	DEBE utilizarse la Norma ISO 3166-1-Alpha <i>Code Elements (2-Letter Country Codes)</i> para referirse a nombres de países, dependencias y demás zonas de especial interés geopolítico, a partir de las listas de nombres de países de las Naciones Unidas.
[SD-42]	DEBE usarse la Norma ISO 639-1 <i>(2-Letter Language Codes)</i> para los códigos de idioma.
[SD-43]	Los esquemas DEBEN declarar elementos y atributos para valores de fecha y hora utilizando los tipos de datos de fecha y hora del esquema del W3C.
[SD-44]	DEBE usarse la Norma ISO 4217-Alpha <i>(3-Letter Currency Codes)</i> para los códigos de divisas.
[SD-45]	Los caracteres utilizados en las enumeraciones DEBEN limitarse al conjunto siguiente: 'a-z, A-Z, 0-9, espacio " " y guion bajo "_'. Los valores de enumeración NO DEBERÍAN comenzar con un carácter numérico.
[SD-46]	Los valores de enumeración DEBERÍAN ser semánticamente suficientes en inglés y emplear el menor número de caracteres posible. Los valores DEBERÍAN proceder del lenguaje común del ámbito de la P.I.
[SD-47]	PUEDEN usarse tipos complejos abstractos.
[SD-48]	Los esquemas solamente DEBERÍAN utilizar atributos para definir datos no operativos. Por ejemplo, es aceptable utilizar un atributo denominado <code>sequenceNumber</code> para los números de serie.
[SD-49]	Un indicador de ocurrencia NO DEBERÍA utilizarse para especificar una limitación que se dé por defecto en un esquema. Por ejemplo, <code>minOccurs="1"</code> y <code>maxOccurs="1"</code> NO DEBERÍAN utilizarse, pero es aceptable utilizar <code>minOccurs="2"</code> y <code>maxOccurs="3"</code> .
[SD-50]	Los elementos vacíos NO DEBEN definirse en los esquemas, a excepción del salto de línea.
[SD-51]	El atributo <code>use</code> en uso de atributo DEBERÍA omitirse cuando el atributo al que se hace referencia es facultativo, puesto que por defecto se utiliza <code>use="optional"</code> .
[SD-52]	Los esquemas NO DEBERÍAN utilizar la instrucción <code>all</code> .

Anexo I, página 27

ID de la Norma	Norma
[SD-53]	Los esquemas PUEDEN utilizar indicadores de ocurrencia en las instrucciones de composición <code>xsd:sequence</code> y <code>xsd:choice</code> .
[SD-54]	Un esquema NO DEBE contener un tipo complejo con <code>xsd:any</code> en lo relativo a la capacidad de ampliación.
[SD-55]	Los esquemas NO DEBEN utilizar grupos de sustitución.
[SD-56]	Los esquemas DEBERÍAN utilizar <code>xsd:key/xsd:keyref/xsd:unique</code> y/o <code>xsd:ID/xsd:IDREF/xsd:IDREFS</code> para identificar las limitaciones, según corresponda.
[SD-57]	Los esquemas DEBERÍAN utilizar <code>xsd:ID/xsd:IDREF/xsd:IDREFS</code> para una única referencia o múltiples referencias dentro del documento XML actual para identificar las limitaciones, según corresponda.
[SD-58]	Todos los esquemas DEBERÍAN utilizar el elemento <code>xsd:documentation</code> para incluir documentación sobre la construcción del esquema. La documentación solamente DEBERÍA describir el elemento o el tipo, y NO DEBERÍA contener detalles sobre la aplicación u otra información que no esté directamente relacionada con el significado de la construcción.
[SD-59]	NO DEBERÍAN utilizarse comentarios (es decir, <code>&lt;!--comment --&gt;</code> ) en un esquema.
[SD-60]	La documentación NO DEBERÍA sustituirse por listas de códigos que utilicen la enumeración.
[SD-61]	Los esquemas de nivel de documento DEBEN incluir documentación del encabezado del esquema que contenga referencias a la fecha de creación, la fecha de última modificación y el punto de contacto, con carácter obligatorio, y al URL sobre las notas explicativas, con carácter opcional.
[SD-62]	Los esquemas DEBERÍAN utilizar <code>xsd:key/xsd:unique/xsd:keyref</code> dentro de un ámbito, cuando la unicidad debe mantenerse enteramente dentro de ese ámbito.
[SD-63]	El esquema para un componente de documento DEBE declarar un atributo facultativo <code>ipoVersion</code> en el elemento raíz con un valor fijo que coincida con la versión de la edición de aplicación de la oficina de P.I., en el formato siguiente: <code>&lt;Código ST.3&gt;_ "V"&lt;número de la versión mayor&gt;_ &lt;número de la versión menor&gt;</code> , por ejemplo, <code>"US_V2_0"</code> .
[SD-64]	El fichero de edición del esquema, un archivo de nombre de esquema en el nivel de documento y un archivo de nombre de esquema aplanado DEBEN contener información coincidente sobre la versión, que comprenda el número de versión mayor y menor.

Anexo I, página 28

*Normas de diseño de instancia*

ID de la Norma	Norma
[ID-01]	Las instancias XML DEBEN validarse con el esquema correspondiente.
[ID-02]	Los documentos de instancia XML DEBEN utilizar el atributo <code>schemaLocation</code> para asociar el espacio de nombres de destino con la localización del archivo del esquema al que se conforma la instancia XML.
[ID-03]	Toda declaración de atributo <code>xsi:schemaLocation</code> DEBERÍA contener un URL resoluble.
[ID-04]	Todo espacio de nombres declarado DEBERÍA asignarse a un prefijo recomendado, por ejemplo, el prefijo "com" para todas las construcciones del esquema de componentes comunes, "pat" para todas las construcciones del esquema de componente de patente, "tmk" para todas las construcciones del esquema de componente de marca, "dgn" para todas las construcciones del esquema de componente de dibujos y modelos industriales, "tbl" para las construcciones del esquema de componente del cuadro OASIS y "mathml" para las construcciones del esquema de componente de MathML.
[ID-05]	Los documentos de instancia XML NO DEBEN utilizar espacios de nombres por defecto.
[ID-06]	Los documentos de instancia XML DEBERÍAN utilizar <code>xsi</code> como prefijo de espacio de nombres en todas las construcciones de aplicaciones del esquema del W3C.
[ID-07]	Los documentos de instancia XML NO DEBEN utilizar declaraciones de espacio de nombres local.
[ID-08]	Los archivos de imagen externos DEBERÍAN ajustarse a uno de los formatos de imagen recomendados en la Norma ST.96 de la OMPI.
[ID-09]	Las imágenes DEBEN calificarse como archivos externos.
[ID-10]	Las listas de secuencias DEBERÍAN seguir la Norma ST.25 de la OMPI.

Anexo I, página 29

**APÉNDICE B – TÉRMINOS DE REPRESENTACIÓN**

<b>Término</b>	<b>Definición</b>	<b>Tipo de datos</b>
Amount (Cuantía)	Valor monetario.	AmountType
Category (Categoría)	División o subconjunto en un sistema de clasificación específicamente definido cuyos elementos comparten, todos ellos, el mismo concepto de taxonomía.	xsd:token
Code (Código)	Combinación de uno o más números, letras o caracteres especiales, que se sustituye por un significado concreto. Representa valores finitos y predeterminados o formato libre.	xsd:token
Date (Fecha)	Punto concreto en el tiempo expresado mediante el año, el mes y el día.	DateType o xsd:date
Identifier (Identificador)	Combinación de uno o más números enteros, letras o caracteres especiales que identifican de manera exclusiva una aplicación concreta de un objeto pero que quizá no tenga un significado fácilmente definible.	xsd:token
Indicator (Indicador)	Señal de presencia, ausencia o requisito de algo. Los valores recomendados son Y, N y "?", si fuere necesario.	xsd:boolean
Measure (Medida)	Valor numérico que se determina midiendo un objeto y que se expresa mediante la unidad de medida especificada. MeasureType se utiliza para representar un tipo de dimensión física como la temperatura, la longitud, la velocidad, la anchura, el peso, el volumen o la latitud de un objeto. Más concretamente, MeasureType debe usarse para medir los componentes físicos o intrínsecos de un objeto percibido como un conjunto.	MeasureType
Name (Nombre)	Designación de un objeto expresada mediante una palabra o una frase.	xsd:string
Number (Número)	Cadena de numerales o caracteres alfanuméricos que expresan una etiqueta, un valor, una cantidad o una identificación.	xsd:positiveInteger, xsd:string or xsd:token
Percent (Por ciento)	Número que representa una parte de un conjunto, que se dividirá por 100.	xsd:decimal
Quantity (Cantidad)	Número contado de unidades no monetarias y que posiblemente incluye fracciones. Quantity se utiliza para representar un número de cosas contadas. Quantity debe utilizarse para contar o cuantificar los componentes que integran un objeto percibido como un compuesto, una colección o un contenedor. Quantity debería siempre expresar un número de cosas contadas, y el número se referirá a la cantidad total, enviada, cargada o almacenada. QuantityType debería utilizarse para componentes que requieren información sobre la unidad; y xsd:nonNegativeInteger debería utilizarse para componentes susceptibles de ser contados que no requieren información sobre la unidad.	QuantityType or xsd:nonNegativeInteger
Rate (Índice)	Cantidad o cuantía medida con respecto a otra cantidad.	xsd:decimal
Text (Texto)	Cadena de caracteres sin formato, por lo general en forma de palabras. (incluye: Abreviaturas, Comentarios.)	xsd:string, LocalizedTextType or OrderedTextType
Time (Tiempo)	Designación de un punto cronológico concreto en un período.	xsd:time
DateTime (fecha y hora)	Fecha y hora en la que se produce un evento.	xsd:datetime
URI	El Identificador Uniforme de Recursos identifica la ubicación del archivo.	xsd:anyURI

Anexo I, página 30

**APÉNDICE C – LISTA DE SIGLAS Y ABREVIATURAS**

En lo que respecta a las abreviaturas, en principio, el primer carácter de la palabra o frase debería escribirse en mayúscula y los caracteres que le siguen en minúscula (por ejemplo, "Pre", "BioDeposit"), con excepción de las abreviaturas que ya son notoriamente conocidas, por ejemplo, "IDREF", "W3C", "ST3". En el caso de siglas, todos los caracteres deberían escribirse en letra mayúscula, por ejemplo, "OMPI".

Alt	Texto alternativo para la imagen
B	Negrita
BioDeposit	Depósito biológico
Br	Salto
CDX	<i>CambridgeSoft proprietary ChemDraw file format</i>
CPC	Clasificación de Patentes Cooperativa
DD	Descripción de definición
Del	Texto eliminado
DL	Lista de definición
DOI	Identificador de objeto digital
DT	Término de definición
DTD	Definición de tipo de documento
DWF	<i>Design Web Format</i>
DWG	Dibujo
ECLA	Clasificación europea
ExtRef	Referencias que son externas al documento XML actual
H<n>	La "n" indica el nivel de encabezado con un valor específico, mediante un número de 1 a 15 dígitos. Es decir que, en el valor de enumeración, esta abreviatura representa uno de entre H1 y H15. Por ejemplo, "H1" significa "Heading 1".
I	Cursiva
ID	Identificador para la identificación del sistema
IDREF	Referencia del identificador
IDREFS	Referencias del identificador
IGES	<i>Initial Graphic Exchange Specification</i>
Ins	Texto insertado
IP	Propiedad Industrial
IPC	Clasificación Internacional de Patentes
IPCR	Reforma de la Clasificación Internacional de Patentes
IPO	Oficina de Propiedad Industrial
IPR	Derecho de Propiedad Industrial
ISO	Organización Internacional de Normalización
LCC	<i>Lower Camel Case</i>
LI	Ítem de la lista
MPEG	<i>Moving Picture Experts Group</i>
MOL	Formato de archivo para guardar información sobre átomos, enlaces, conectividad y coordenadas de una molécula
NB	Formato de archivos para cuadernos Mathematica
NPL	Documentos distintos de los de patentes

Anexo I, página 31

O	Tachado
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>
OCR	Reconocimiento óptico de caracteres
OL	Lista ordenada
P	Párrafo
PAN	Número primario de cuenta
PCT	Tratado de Cooperación en materia de Patentes
PKCS7	En <a href="#">criptografía</a> , PKCS es un grupo normas de <a href="#">criptografía asimétrica</a> , y <a href="#">PKCS #7</a> (PKCS7) corresponde a la norma de sintaxis de mensajes codificados, que describe la sintaxis general de los datos a los que puede haberse aplicado criptografía, por ejemplo, firmas y sobres digitales.
Pre	Texto preformateado
S	Texto tachado
ST3	Norma ST.3 de la OMPI
ST13	Norma ST.13 de la OMPI
Sub	Subíndice
Sup	Superíndice
SVG	Imagen <i>Scalable Vector Graphics</i>
SWF	<i>Small Web Format</i>
SWIFT	Sociedad de Telecomunicaciones Interbancarias Mundiales
ThreeDM	Modelado dimensional
ThreeDS	3D Studio
U	Subrayado
UCC	<i>Upper Camel Case</i>
UL	Lista no ordenada
UPOV	Unión Internacional para la Protección de las Obtenciones Vegetales
URI	Identificador uniforme de recursos
URL	Localizador uniforme de recursos
URN	Nombre uniforme de recursos
W3C	Consortio <i>World Wide Web</i>
WIPO	Organización Mundial de la Propiedad Intelectual
WMV	<i>Windows Media Video</i>
XML	Lenguaje extensible de marcado

Anexo I, página 32

## APÉNDICE D – MATERIAL DE REFERENCIA

### *Normas de la OMPI*

- Norma [ST.3](#) de la OMPI: Códigos de dos letras para la representación de Estados, otras entidades y organizaciones
- Norma [ST.13](#) de la OMPI: Numeración de solicitudes de derechos de P.I.
- Norma [ST.25](#) de la OMPI: Listas de secuencias de nucleótidos y aminoácidos
- Norma [ST.36](#) de la OMPI: Tratamiento en XML de la información sobre patentes
- Norma [ST.66](#) de la OMPI: Tratamiento en XML de la información relativa a las marcas
- Norma [ST.67](#) de la OMPI: Gestión electrónica de los elementos figurativos de las marcas
- Norma [ST.86](#) de la OMPI: Tratamiento en XML de información relativa a los diseños industriales

### *Normas de la industria*

- Conjunto de recomendaciones sobre lenguaje de definición de esquemas (XSD) de XML del W3C—*Esquema de XML Parte 1: Estructuras y Esquema de XML; Parte 2: Tipos*
- *Request for Comments 2119*, del Grupo de Tareas sobre Ingeniería de Internet
- Norma ISO/IEC 10646- *Information technology — Universal multiple-octet coded character set (UCS)*
- Norma ISO 11179, *Information Technology -- Metadata registries (MDR), Part 5: Naming and identification principles*
- Norma ISO 3166-1:2006, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*
- Norma ISO 639-1:2002, *Codes for the representation of names of languages – Part 1: Alpha-2 code*
- Norma ISO 8601:2004, *Data elements and interchange formats -- Information interchange -- Representation of dates and times*
- Norma ISO 4217:2008, *Codes for the representation of currencies and funds*
- Esquema de Cuadro de XML OASIS XML: <http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd>
- MathLab: MathML, versión 3. Véase <http://www.w3.org/TR/MathML3> para una descripción completa

[Fin del Anexo I]