

## **NORMA ST.96 - ANEXO I**

### **NORMAS DE DISEÑO Y CONVENCIONES DE XML**

Versión 1.0

*adoptada por el Comité de Normas Técnicas de la OMPI (CWS)  
en su undécima sesión el 4 de mayo de 2012*

#### **Índice**

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 .....	4
1.5.1 Palabras clave .....	4
1.5.2 Notaciones generales.....	4
1.5.3 Identificadores de normas .....	5
2. CONVENCIONES DE DISEÑO DE XML .....	5
2.1 Normas generales de diseño de XML .....	5
2.2 Convenciones sobre denominación en XML.....	5
2.2.1 Convenciones sobre denominación de construcciones de esquema .....	5
2.2.2 Convenciones sobre nombre de archivo de esquema.....	7
2.3 Estrategia de modularidad.....	7
2.3.1 Módulos de esquema .....	8
2.3.2 Referencia a un esquema externo .....	8
2.4 Capacidad de reutilización .....	9
2.5 Espacios de nombres .....	10
2.5.1 Declaración y calificación de espacio de nombres .....	10
2.5.2 Espacios de nombres en un esquema de XML .....	10
2.5.3 Espacios de nombres de destino .....	11
2.5.4 Espacios de nombres por defecto.....	12
2.6 Versiones del esquema .....	12
2.6.1 Cambios mayores y cambios menores .....	12
2.6.1.1 Versiones mayores.....	12
2.6.1.2 Versiones menores.....	13
2.6.2 Estrategia relativa a las versiones de un esquema.....	13
2.6.2.1 Espacio de nombres en las versiones del esquema.....	13
2.6.2.2 Convenciones sobre denominación de archivos en las versiones del esquema .....	14
2.6.2.3 Atributo <code>version</code> integrado en el esquema de XML en las versiones del esquema .....	14
2.6.2.4 Atributo <code>schemaVersion</code> definido por el usuario en las versiones del esquema para aplicaciones de XML .....	14
2.7 Transformabilidad con otras normas sobre XML de la OMPI .....	15
2.8 Esquemas desarrollados por la industria .....	15

Anexo I, página 2

<b>3. CONVENCIONES RELATIVAS A CONSTRUCCIONES DE ESQUEMAS DE XML</b> .....	<b>15</b>
3.1 Definiciones de tipos.....	15
3.1.1 Tipos simples.....	16
3.1.1.1 Tipos de datos integrados del W3C .....	16
3.1.1.2 Tipos de datos definidos por el usuario .....	16
3.1.2 Tipos complejos.....	16
3.2 Elementos y atributos .....	16
3.2.1 Elementos frente a atributos.....	17
3.2.2 Elementos.....	17
3.2.2.1 Cardinalidad de los elementos .....	17
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 .....	18
3.3.1 Ampliación .....	18
3.3.2 Restricción .....	18
3.3.3 Grupos de sustitución.....	18
3.4 Limitaciones de identidad .....	18
3.5 Documentación del esquema .....	19
3.5.1 Documentación del encabezado del esquema .....	19
<b>4. NORMAS DE DISEÑO DE LA APLICACIÓN</b> .....	<b>20</b>
4.1 Espacios de nombres en documentos resultantes de una aplicación de XML.....	20
4.1.1 Validación de un documento resultante de una aplicación de XML .....	20
4.1.2 Declaración y calificación del espacio de nombres en documentos resultantes de una aplicación de XML .....	20
4.1.3 Espacio de nombres de una aplicación del esquema del W3C .....	21
4.1.4 Alcance del espacio de nombres .....	22
4.2 Entidades externas .....	22
<b>APÉNDICE A – RESUMEN DE LAS NORMAS DE DISEÑO</b> .....	<b>23</b>
Normas generales de diseño.....	23
Normas de diseño de esquema.....	24
Normas de diseño de aplicación .....	27
<b>APÉNDICE B – TÉRMINOS DE REPRESENTACIÓN</b> .....	<b>28</b>
<b>APÉNDICE C – LISTA DE SIGLAS Y ABREVIATURAS</b> .....	<b>29</b>
<b>APÉNDICE D – MATERIAL DE REFERENCIA</b> .....	<b>30</b>
Normas de la OMPI .....	30
Normas de la industria.....	30

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 los tres tipos de P.I. y facilitar el intercambio de datos entre Oficinas de P.I. utilizando unos 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 y aplicaciones de XML relativos a 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;

Annex I, page 4

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*.
- se entenderá por *esquema compatible con la Norma ST.96* todo esquema que sea coherente con los componentes de un esquema ST.96 y con las Normas de diseño y convenciones de XML en materia de P.I., es decir, el Anexo I de la norma ST.96.
- se entenderá por *esquema conforme a la Norma ST.96* todo esquema compatible que no se haya ampliado y que soporte las limitaciones planteadas por un esquema ST.96;

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* el tipo, elemento o atributo.

8. Las presentes Normas de diseño y convenciones contienen determinadas palabras clave que poseen un significado categórico. Estas palabras clave, basadas en las definiciones que figuran en el documento *Request for Comments 2119*, elaborado por el Grupo de Tareas sobre Ingeniería de Internet (IETF), son las siguientes (cuando aparezcan en minúscula, se utilizarán en el sentido habitual en el idioma español);

- **DEBE:** Se entenderá por esta palabra, o por los términos REQUERIDO o DEBERÁ, que la definición es una exigencia de la norma;
- **NO DEBE:** Se entenderá por esta frase, o por la frase NO DEBERÁ, que la definición es una prohibición terminante de la norma;
- **DEBERÍA:** Se entenderá por esta palabra, o por el adjetivo RECOMENDADO, que podrán existir razones válidas en determinadas circunstancias para pasar por alto un punto concreto, pero es preciso ser plenamente consciente de las implicaciones y examinar cuidadosamente la situación antes de optar por una solución distinta;
- **NO DEBERÍA:** Se entenderá por esta frase, o por la frase NO RECOMENDADO, que podrán existir razones válidas en determinadas circunstancias para considerar como aceptable o incluso útil una actuación concreta, pero es preciso ser plenamente consciente de las implicaciones y examinar cuidadosamente la situación antes de proceder a dicha actuación; y
- **PUEDE:** Se entenderá por esta palabra, o por el adjetivo OPCIONAL, que dicho punto es realmente facultativo. Una aplicación que no incluya una opción concreta DEBE poder interactuar con otra aplicación que sí la incluya, aunque tal vez con menos funciones. Asimismo, una aplicación que incluya una opción concreta DEBE poder interactuar con otra aplicación que no la incluya (salvo, lógicamente, en lo relativo a la función que ofrece dicha opción).

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 de la aplicación específico.
- **“ ”:** Indica que el texto entre comillas debe usarse de manera literal en la aplicación.
- **{ }:** Indica que la aplicación de estos elementos es facultativa.

Annex I, page 5

- **Courier:** Indica palabras clave de XML, nombres de etiquetas de XML y códigos de XML que aparecen en tipo de letra courier.

### 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 aplicaciones.
- b) El valor "nn" indica el número de serie de la norma.

Por ejemplo, el identificador de norma [GD-10] se refiere a la décima norma general de diseño.

## 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 aplicaciones 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] Norma ISO/IEC 10646 – UCS – DEBE usarse Unicode para el conjunto de caracteres. DEBE usarse UTF-8 para codificar los caracteres Unicode.

### 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. Estas convenciones también se pueden aplicar a los nombres de archivo.

#### 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, deben 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 al término de clase de objeto o al término de propiedades para individualizar un nombre.

Annex I, page 6

- [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, incluidos los nombres específicos de oficinas, a excepción de 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 DEBEN contener nombres, adjetivos y verbos en presente.
- [GD-06] Los caracteres empleados en los nombres de tipos, elementos y atributos DEBEN limitarse al conjunto siguiente: {a-z, A-Z y 0-9}.
- [GD-07] La longitud máxima de un nombre DEBERÍA ser 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 + 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. El resto de siglas y abreviaturas utilizadas en una declaración de atributos DEBEN aparecer en mayúscula.
- [GD-14] Las siglas y las abreviaturas DEBEN aparecer en mayúscula en todos los nombres de elementos y tipos.
- [GD-15] Los nombres de tipos complejos DEBERÍAN incluir un término de clase de objeto significativo.
- [GD-16] Una asociación de nombres de tipo complejo DEBERÍA utilizar la siguiente estructura: clase de objeto del tipo complejo vinculante, propiedad (naturaleza de la vinculación), clase de objeto del tipo complejo vinculado y cualquier calificador. Por ejemplo, `ApplicantResidenceAddress: Applicant` es la clase de objeto del tipo complejo vinculante, `Residence` es la propiedad y `Address` es la clase de objeto del tipo complejo vinculado.
- [GD-17] Los nombres de tipo simple y de elemento atómico (sin elementos derivados) DEBERÍAN componerse del término de clase de objeto, el término de propiedades, el término de representación y el término calificador.
- [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] El término de clase de objeto DEBE ocupar el primer lugar (más a la izquierda) del nombre, el término de propiedades debe aparecer a continuación y el término de representación debe figurar en último lugar (más a la derecha). El término calificador DEBERÍA anteceder al término de propiedades o al término de clase de objeto vinculado.
- [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] Los términos de representación del Apéndice B DEBEN emplearse para los términos de representación en nombres de componentes.

Annex I, page 7

- [GD-24] En un espacio de nombres, todos los nombres de tipos, elementos y atributos (declarados local o globalmente) DEBEN ser exclusivos.
- [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 una colección DEBERÍA utilizar el sufijo "Bag". Por ejemplo, `EmailAddressBag` representa una colección de `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. Por ejemplo, `GoodsAndServices`.
- [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. Las convenciones sobre nombres de archivo de esquema son un elemento importante de la estrategia de elaboración de versiones del esquema. Debe existir una coherencia en los nombres de archivo de esquema para garantizar que los usuarios puedan distinguir las distintas versiones de un esquema. Así, las normas contenidas en esta sección están estrechamente relacionadas con las normas de la sección *Versiones del esquema* del presente documento.

16. El archivo del esquema debe contener información sobre la versión. Se recomienda incluir el identificador de versión en el nombre de archivo de esquema. En el caso de esquemas en fase de elaboración (proyecto de esquema), es posible revisar el proyecto de esquema. 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 del esquema DEBEN limitarse al conjunto siguiente: {a-z, A-Z, 0-9, guión (-), y punto (.)}.
- [GD-31] El nombre de archivo del esquema DEBE constar de cinco partes y tres delimitadores, es decir:  
<nombre del componente>"-""V"<número de versión mayor>"-""<número de versión menor>"<extensión del archivo>. Por ejemplo, `EmailAddressType-V1-1.xsd`, `languageCode-V1-0.xsd`, `id-V1-2.xsd`.
- [GD-32] El nombre de archivo del proyecto de esquema DEBE constar de siete partes y cuatro delimitadores, es decir: <nombre del componente>"-""V"<número de versión mayor>"-""<número de versión menor>"-""D"<número de revisión>."<extensión del archivo>, por ejemplo, `Contact-V1-0-D3.xsd`.

### *2.3 Estrategia de modularidad*

17. 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, según sea necesario, sin necesidad de importar 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 impide que se puedan compartir y reutilizar elementos individuales o componentes lógicos definidos como un grupo en un esquema.

Annex I, page 8

2.3.1 Módulos de esquema

18. 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) y ser compartidos por, al menos, dos tipos de P.I.

19. Existen tres niveles distintos de componentes, a saber, componentes básicos, componentes agregados y componentes de documento. Los componentes comunes se dividen en componentes básicos y componentes agregados, mientras que los componentes específicos se dividen en componentes básicos, componentes agregados y componentes de documento.

Nivel	Descripción
Componente básico	Un componente básico lo utilizan uno o más componentes agregados. Un componente básico se representa mediante un elemento.  El componente básico remite al tipo de datos integrado del W3C, al tipo simple o al tipo complejo, y se define <code>xsd:simpleContent</code> . Por ejemplo, <code>FirstName (string)</code> , <code>PhoneNumberCategory (PhoneNumberCategoryType)</code> .
Componente agregado	Un componente agregado es una colección de componentes básicos conexos y/o otros componentes agregados que, juntos, transmiten un significado específico distinto, independiente o no de cualquier contexto operativo concreto. Se representa mediante un elemento y un tipo (tipos complejos con elementos múltiples).  Un componente agregado remite a la definición de componente básico, de componente agregado o a las normas externas especificadas en el presente documento. Por ejemplo, <code>Name (NameType)</code> , <code>Contact (ContactType)</code> .
Componente de documento	Un componente de documento es un componente de nivel superior que define los documentos principales intercambiados. Se representa mediante un elemento y un tipo complejo.  Un componente de documento remite a la definición de componente básico, de componente agregado o a una norma externa especificada en el presente documento. Por ejemplo, <code>ApplicationBody (ApplicationBodyType)</code> , <code>MarkRecord (MarkRecordType)</code> .

20. El ejemplo siguiente corresponde a una declaración de componente básico:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:com="http://www.wipo.int/standards/XMLSchema/Common/0"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Common/0"
elementFormDefault="qualified" attributeFormDefault="qualified" version="0.7">...
  <xsd:include schemaLocation="sequenceNumber-V0-5.xsd"/>
  <xsd:include schemaLocation="TextType-V0-5.xsd"/>
  <xsd:complexType name="AddressLineType">
    <xsd:annotation>
      <xsd:documentation>A line in an address. </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="com:TextType">
        <xsd:attribute ref="com:sequenceNumber" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:schema>
```

2.3.2 Referencia a un esquema externo

21. Cada módulo de esquema reside en un espacio de nombres distinto. En el presente documento, 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.

22. 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



Annex I, page 9

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

23. En la Figura 1 se describe la utilización de la estrategia de modularidad de la Norma ST.96 de la OMPI para elaborar esquemas.

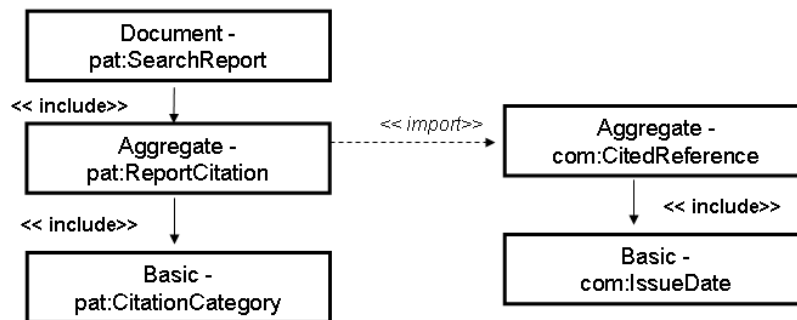


Figura 1. Ejemplo de modularidad del esquema

- [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 `import` para referirse a módulos de esquema de componente común.
- [SD-02] El módulo de esquema de componente común NO DEBE referirse a los módulos de esquema de componente de patente, a los módulos de esquema de componente de marca y a los módulos de esquema de componente de dibujos y modelos industriales.
- [SD-03] Los módulos de esquema de componente de patente, de componente de marca y de componente de dibujos y modelos industriales NO DEBEN referirse los unos a los otros.

#### 2.4 Capacidad de reutilización

24. 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.

25. 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.

26. 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 resultante de una aplicación de XML válido conforme con el esquema definido para dicho espacio de nombres.

27. 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 como el elemento. 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. En consecuencia, la Norma ST.96 de la OMPI recomienda el patrón de diseño *Garden of Eden*.

Annex I, page 10

- [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 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 *redefine* para la redefinición.

**2.5 Espacios de nombres**

28. Los espacios de nombres se usan de manera exclusiva para identificar 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 aplicación de XML podrá 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.
- [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 URL.
- [SD-13] Todos los esquemas DEBEN tener unos atributos `elementFormDefault` y `attributeFormDefault` con un valor "qualified"

29. Por motivos de eficacia, 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**

30. Los nombres de los espacios de nombres son un elemento importante de la estrategia relativa a las versiones del esquema. Las normas para la denominación de un espacio de nombres *deben respetarse* en el diseño de esquemas. Con independencia del número de versión menor de un esquema determinado, el espacio de nombres debe contener solamente el número de versión mayor. Por ejemplo:  
<http://www.wipo.int/standards/XMLSchema/Trademark/1>

31. 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/Common/1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
.....
</xsd:schema>
```

32. Puede encontrarse más información en la sección Versiones del esquema del presente documento.

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

Annex I, page 11

[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

33. 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.

34. 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:

- <http://www.wipo.int/standards/XMLSchema/Common/<major>>, 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/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Common/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xsd:include schemalocation="xxx.xsd"/>
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Patent/<major>>, 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/Patent/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xsd:include schemalocation="xxx.xsd"/>
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Trademark/<major>>, 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/Trademark/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Trademark/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
<xsd:include schemalocation="xxx.xsd"/>
...
</xsd:schema>
```

- <http://www.wipo.int/standards/XMLSchema/Design/<major>>, 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/Design/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Design/1"
elementFormDefault="qualified" attributeFormDefault="qualified">
<xsd:include schemalocation="xxx.xsd"/>
.....
</xsd:schema>
```

Annex I, page 12

- [SD-16] Todo esquema DEBE declarar sus espacios 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 calificadores de espacio de nombres declarados, pero no con el calificador del W3C.
- [SD-18] Los esquemas de componente común DEBEN utilizar como espacio de nombres de destino `http://www.wipo.int/standards/XMLSchema/Common/<major>`.
- [SD-19] Los esquemas de componente de patente DEBEN utilizar como espacio de nombres de destino `http://www.wipo.int/standards/XMLSchema/Patent/<major>`.
- [SD-20] Los esquemas de componente de marca DEBEN utilizar como espacio de nombres de destino `http://www.wipo.int/standards/XMLSchema/Trademark/<major>`.
- [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/Design/<major>`.

#### 2.5.4 Espacios de nombres por defecto

35. 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 los prefijos de 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 a partir del examen del esquema incluyente. En consecuencia, la Norma ST.96 de la OMPI desaconseja utilizar espacios de nombres por defecto.

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

#### *2.6 Versiones del esquema*

36. 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 resultante de una aplicación de 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

37. 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

38. En una versión mayor de un esquema, los cambios con respecto a versiones anteriores son significativos y/o incompatibles. Intentar validar una aplicación de 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.

Annex I, page 13

2.6.1.2 Versiones menores

39. 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 proyecto de esquema no puede validar los documentos resultantes de una aplicación de XML existentes creados con la versión mayor actual.

[SD-25] NO SE DEBERÍAN modificar las versiones menores de un esquema salvo que se altere el esquema o uno de los esquemas importados/incluidos para introducir cambios menores.

[SD-26] En el caso de los esquemas, el nombre de archivo de esquema, el atributo `version` del W3C y el espacio de nombres DEBEN contener información coincidente sobre la versión.

[SD-27] La versión de un esquema DEBERÍA modificarse cuando se actualice un esquema incluido o importado.

[SD-28] Cuando una construcción de un esquema se vea alterada para introducir cambios mayores en un espacio de nombres determinado, DEBE incrementarse la versión mayor de todos los esquemas de dicho espacio de nombres.

[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

40. Los esquemas siguientes operan conjuntamente para definir la estrategia relativa a las versiones de un esquema:

- a) Convenciones sobre denominación de espacios de nombres de esquema;
- b) Convenciones sobre denominación de archivos de esquema;
- c) utilización del atributo `version` integrado en el esquema de XML; y
- d) utilización de un atributo `schemaVersion` definido por el usuario.

2.6.2.1 Espacio de nombres en las versiones del esquema

41. De conformidad con las normas sobre denominación de espacio de nombres que figuran en el presente documento, los espacios de nombres solo contienen un número de versión mayor. Es importante mantener la compatibilidad entre las versiones menores del esquema. Dado que los archivos de una aplicación de una versión menor anterior deben seguir validándose con arreglo a las versiones menores más recientes del esquema, debe conservarse el espacio de nombres de una versión menor a otra.

42. El espacio de nombres no basta para identificar fácilmente la versión exacta de un esquema determinado o de un documento resultante de una aplicación. En este punto cobra importancia el resto de aspectos de las versiones del esquema, tal y como se describe a continuación.

Annex I, page 14

2.6.2.2 Convenciones sobre denominación de archivos en las versiones del esquema

43. La información sobre la versión puede incluirse en el nombre de archivo del esquema. De acuerdo con las normas sobre denominación de archivos que figuran en el presente documento, el nombre de archivo de esquema contiene tanto un número de versión mayor como un número de versión menor.

2.6.2.3 Atributo *version* integrado en el esquema de XML en las versiones del esquema

44. 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/Patent/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
...
</xsd:schema>
```

[SD-30] El atributo *version* del esquema del W3C DEBE incluir un número de versión mayor y un número de versión menor en cada archivo de esquema en el formato siguiente:  
<número de versión mayor>.<número de versión menor>.

2.6.2.4 Atributo *schemaVersion* definido por el usuario en las versiones del esquema para aplicaciones de XML

45. El atributo *schemaVersion* definido por el usuario es necesario para que un documento determinado resultante de una aplicación remita directamente a una versión concreta del esquema. Este atributo definirá claramente la versión del esquema a la que apunta el documento resultante de una aplicación. Por ejemplo:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified" attributeFormDefault="qualified" version="1.0">
<xsd:element name="SearchReport" type="SearchReportType"/>
<xsd:complexType name="SearchReportType"/>
  <xsd:attribute name="schemaVersion" type="xsd:String" use="required"/>
  .....
</xsd:complexType>
```

La aplicación XML correspondiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<pat:SearchReport xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/Patent/1"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/Patent/1/Document
SearchReport-V1.0.xsd" schemaVersion="1.0">
.....
</pat:SearchReport>
```

[SD-31] Los esquemas definidos como componentes de documento DEBEN definir en su elemento raíz un atributo requerido denominado *schemaVersion*, en el formato siguiente: <número de versión mayor>.<número de versión menor>.

Annex I, page 15

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

46. Antes de la aprobación de la Norma ST.96 de la OMPI, las Oficinas de P.I. ya han estado utilizando las Normas ST.36, ST.66 y ST.86 de la OMPI. En consecuencia, 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.

47. 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 aplicación de XML conforme a la Norma ST.96 puede transformarse en unidades atómicas de información que podrían incluirse en una aplicación de XML conforme a las Normas ST.36, ST.66 y ST.86 y viceversa.

48. 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 "Undefined", es decir que la información no figuraba en los datos originales (aplicaciones de XML conformes a las Normas ST.36, ST.66 y ST.86).

[SD-32] Los esquemas ST.96 DEBERÍAN elaborarse con un nivel similar 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 "Undefined" a menos que sea necesario.

*2.8 Esquemas desarrollados por la industria*

49. Cuando sea apropiado para el contenido de un documento, es decir, cuando el contenido no sea exclusivo del ámbito de la P.I., se emplearán 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. El esquema de cuadro OASIS está disponible en: <http://www.oasis-open.org/docbook/xmlschema/1.0b1/calstbl.xsd>; se trata de un modelo de esquema que hay que especificar más para adaptarlo a las necesidades operativas. Para satisfacer las actividades de P.I., en el esquema ST.96 se define OASISTable.xsd a partir del modelo de esquema. Para las formulas matemáticas, se recomienda la utilización de MathML, versión 3. Puede encontrarse más información sobre MathML, versión 3, en: <http://www.w3.org/TR/MathML3>.

50. En el caso de estructuras o fórmulas químicas, en el momento de elaborar el presente documento no existe una recomendación del W3C ni un esquema de XML desarrollado por la industria 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 para datos químicos.

[SD-34] DEBERÍA utilizarse MathML, versión 3, para las fórmulas matemáticas.

[SD-35] En el caso de los cuadros, DEBERÍA utilizarse OASISTable.xsd, que se basa en el esquema de XML 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*

51. 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.

Annex I, page 16

### 3.1.1 Tipos simples

52. 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. En la Norma ST.96 de la OMPI, los tipos simples son tipos de datos definidos por el usuario.

#### 3.1.1.1 Tipos de datos integrados del W3C

53. 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

54. Una de las ventajas de los esquemas de XML es su capacidad para definir tipos de datos de 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] Los tipos simples DEBERÍAN utilizarse para definir las listas de códigos que utilizan la enumeración.
- [SD-39] Los tipos simples PUEDEN utilizar *xsd:union* para mejorar una lista de códigos.
- [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 *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 valores de fecha y hora de las aplicaciones resultantes de un esquema DEBEN adaptarse a los tipos de datos de fecha y hora del esquema de XML 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 listas de enumeraciones DEBEN limitarse al conjunto siguiente: {a-z, A-Z, 0-9, punto (.), coma (,), espacios, guión (-), símbolo & (&#26;) y subrayado (\_)}.
- [SD-46] Los valores disponibles en las listas de enumeraciones 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

55. 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

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



Annex I, page 17

### 3.2.1 Elementos frente a atributos

57. 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 cuando definan datos no operativos. Por ejemplo, el atributo `sequenceNumber`.

### 3.2.2 Elementos

#### 3.2.2.1 Cardinalidad de los elementos

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

59. 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] Los esquemas NO DEBERÍAN utilizar indicadores de ocurrencia para los valores por defecto (es decir, `minOccurs="1"`; `maxOccurs="1"`).

#### 3.2.2.2 Elementos vacíos

60. Un elemento vacío es un elemento sin contenido, 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.

61. 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

62. 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. Por lo tanto, los atributos no se pueden ampliar, a diferencia de lo que sucede con los elementos.

63. 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] Los esquemas NO DEBERÍAN utilizar el indicador `use` para un atributo cuando el valor esperado sea el valor por defecto (es decir, `use="optional"`).

### 3.2.4 Agrupación de elementos y atributos

64. 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`.

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

Annex I, page 18

66. La instrucción `choice` indica que solamente uno de los elementos declarados puede aparecer en un documento resultante de una aplicación de XML.

67. En algunos casos, los indicadores de ocurrencia de las instrucciones `sequence` y `choice` permiten la flexibilidad de utilizar las instrucciones `sequence/choice`, es decir `sequence/choice` optativo o múltiple, para no tener que definir un nivel adicional.

68. La instrucción `all` indica que los elementos declarados pueden aparecer solamente una vez en un documento resultante de una aplicación de 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

69. 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

70. 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; por su parte, derivar tipos simples por medio 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.

#### 3.3.2 Restricción

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

#### 3.3.3 Grupos de sustitución

72. Los grupos de sustitución permiten que un elemento global sustituya a otro elemento global en un documento resultante de una aplicación de 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

73. Como sucede con cualquier sistema de almacenamiento, 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, el XML puede hacerlo de dos maneras:

- `xsd:ID/xsd:IDREF/IDREFS` emula directamente los tipos de atributo `ID`, `IDREF` y `IDREFS` de las definiciones de tipos de documento (DTD) de XML; y
- `xsd:key/xsd:keyref/xsd:unique` se introdujo para mejorar la flexibilidad utilizando expresiones XPATH.

Annex I, page 19

- [SD-56] Los esquemas DEBERÍAN utilizar `xsd:key/xsd:keyref/xsd:unique` para las limitaciones de identidad.
- [SD-57] Los esquemas NO DEBERÍAN utilizar `ID/IDREF/IDREFS` para las limitaciones de identidad.

### 3.5 Documentación del esquema

74. La especificación de esquema del W3C permite incluir documentación en los esquemas, hecho que permite que los esquemas incorporen descripciones de cada construcción. La documentación es necesaria para 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.

75. 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.

76. 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 de documentación del W3C 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, `<!-- comment -->`) en un esquema.
- [SD-60] La documentación NO DEBERÍA sustituirse por listas de códigos que utilicen la enumeración.

#### 3.5.1 Documentación del encabezado del esquema

77. 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. Tal y como sucede con la documentación sobre la construcción del esquema, el elemento de documentación del esquema del W3C debe usarse para la documentación del encabezado del esquema.

78. En el Cuadro 1 se muestran los elementos que se deben incluir en la sección de encabezado de cualquier esquema.

Cuadro 1. Información sobre la documentación del encabezado

Nombre del elemento del encabezado	Descripción
"Last Update"	Fecha de la última actualización del esquema
"Contact Point"	Persona de contacto en caso de dudas sobre el esquema
"Release Notes URL"	La URL en la que se publican las notas explicativas del esquema

El nombre del elemento del encabezado debe preceder el contenido del encabezado, como se ve en el ejemplo siguiente:

```
<xsd:annotation>
  <xsd:documentation>
    Last update: 2011-05-28
    Contact point: xml.standards@wipo.int
    Release notes URL: http://www.wipo.int/standards/XMLSchema/ReleaseNotes.txt
  </xsd:documentation>
</xsd:annotation>
```

Annex I, page 20

79. 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 DEBEN incluir documentación del encabezado del esquema que contenga referencias a "Last Update", "Contact Point" y "Release Notes URL".

#### 4. NORMAS DE DISEÑO DE LA APLICACIÓN

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

##### 4.1 Espacios de nombres en documentos resultantes de una aplicación de XML

###### 4.1.1 Validación de un documento resultante de una aplicación de XML

81. Se recomienda validar un documento resultante de una aplicación de XML con un esquema. La validación de un documento resultante de una aplicación de 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 resultante de una aplicación de XML.

82. 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.

83. La ubicación del esquema se puede facilitar en el elemento raíz del documento resultante de una aplicación de 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/Common/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wipo.int/standards/XMLSchema/Common/1/Aggregate/Contact-V1-0.xsd">
.....
</com:Contact>
```

84. El atributo `schemaLocation` es una construcción del esquema del W3C que asocia un documento resultante de una aplicación de 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/Common/1" es el espacio de nombres de destino del esquema `Contact.xsd`.

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

[ID-02] Los documentos resultantes de una aplicación de XML DEBEN utilizar el atributo `schemaLocation` para describir el espacio de nombres y el nombre de esquema con el que está conforme la aplicación de XML.

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

###### 4.1.2 Declaración y calificación del espacio de nombres en documentos resultantes de una aplicación de XML

85. Como sucede con los esquemas, la declaración de un espacio de nombres en el elemento raíz de un documento resultante de una aplicación de XML se realiza utilizando un identificador de espacio de nombres y un prefijo de espacio de nombres recomendado. Los documentos resultantes de una aplicación de 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.

Annex I, page 21

86. Conviene señalar que:

- el identificador de espacio de nombres en un documento resultante de una aplicación de 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 resultante de una aplicación de XML no tiene por qué ser el mismo que el prefijo de espacio de nombres para el espacio de nombres de destino del esquema.

87. Los elementos y atributos de un documento resultante de una aplicación de XML solamente pueden ser aptos para un espacio de nombres si pertenecen al espacio de nombres de destino del esquema que valida el documento resultante de una aplicación de XML. En consecuencia, 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`.

88. 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).

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/Patent/1"
targetNamespace="http://www.wipo.int/standards/XMLSchema/Patent/1"
elementFormDefault="qualified"
attributeFormDefault="qualified">
.....
</xsd:schema>
```

89. 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 resultante de una aplicación de 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 resultantes de una aplicación de XML NO DEBEN utilizar espacios de nombres por defecto.

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

90. 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 resultantes de una aplicación de 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 resultante de una aplicación de 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/Common/1">
.....
</com:Contact>
```

Annex I, page 22

91. 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 resultantes de una aplicación de 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

92. Los espacios de nombres en los documentos resultantes de una aplicación de XML tienen un alcance de aplicabilidad en un documento resultante de una aplicación de 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.

93. 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 resultantes de una aplicación de XML NO DEBEN utilizar declaraciones de espacio de nombres local.

#### *4.2 Entidades externas*

94. 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.

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

96. Las imágenes pueden estar integradas en una aplicación de 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] Las entidades externas que sean imágenes 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.

**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]	Norma ISO/IEC 10646 – UCS – DEBE usarse Unicode para el conjunto de caracteres. 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, incluidos los nombres específicos de oficinas, a excepción de 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 DEBEN contener nombres, adjetivos y verbos en presente.
[GD-06]	Los caracteres empleados en los nombres de tipos, elementos y atributos DEBEN limitarse al conjunto siguiente: {a-z, A-Z y 0-9}.
[GD-07]	La longitud máxima de un nombre DEBERÍA ser 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, CountryCode.
[GD-10]	Los nombres de tipos DEBEN seguir la convención UCC + sufijo Type. Por ejemplo, ApplicantType.
[GD-11]	Los nombres de atributos DEBEN seguir la convención <i>lower camel case</i> (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 largo ampliado.
[GD-13]	Las siglas y las abreviaturas que aparecen al principio de una declaración de atributos DEBEN estar todas ellas en minúscula. El resto de siglas y abreviaturas utilizadas en una declaración de atributos DEBEN aparecer en mayúscula.
[GD-14]	Las siglas y las abreviaturas DEBEN aparecer en mayúscula en el caso de todos los nombres de elementos y tipos.
[GD-15]	Los nombres de tipos complejos DEBERÍAN incluir un término de clase de objetivo significativo.
[GD-16]	Una asociación de nombres de tipo complejo DEBERÍA utilizar la siguiente estructura: clase de objeto del tipo complejo vinculante, propiedad (naturaleza de la asociación), clase de objeto del tipo complejo vinculado y cualquier calificador. Por ejemplo, ApplicantResidenceAddress: Applicant es la clase de objeto del tipo complejo vinculante, Residence es la propiedad y Address es la clase de objeto del tipo complejo vinculado.
[GD-17]	Los nombres de tipo simple y de elemento atómico (sin elementos derivados) DEBERÍAN componerse del término de clase de objeto, el término de propiedades, el término de representación y el término calificador.
[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.

Annex I, page 24

ID de la Norma	Norma
[GD-21]	El término de clase de objeto DEBE ocupar el primer lugar (más a la izquierda) en el nombre, el término de propiedades debe aparecer a continuación y el término de representación debe figurar en último lugar (más a la derecha). El término calificador DEBERÍA anteceder al término de propiedades o al término de clase de objeto vinculado.
[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]	Los términos de representación del Apéndice B DEBEN emplearse para los términos de representación en nombres de componentes.
[GD-24]	En un espacio de nombres, todos los nombres de tipos, elementos y atributos (declarados local o globalmente) 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 una colección DEBERÍA utilizar el sufijo "Bag". Por ejemplo, <code>EmailAddressBag</code> representa una colección de <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. Por ejemplo, <code>GoodsAndServices</code> .
[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 del esquema DEBEN limitarse al conjunto siguiente: {a-z, A-Z, 0-9, guión (-), y punto (.)}.
[GD-31]	El nombre de archivo del esquema DEBE constar de cinco partes y tres delimitadores, es decir: <nombre del componente>"-""V"<número de versión mayor>"-""<número de versión menor>"."<extensión del archivo>. Por ejemplo, <code>EmailAddressType-V1-1.xsd</code> , <code>languageCode-V1-0.xsd</code> , <code>id-V1-2.xsd</code> .
[GD-32]	El nombre de archivo del proyecto de esquema DEBE constar de siete partes y cuatro delimitadores, 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-V1-0-D3.xsd</code> .

*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>import</code> para referirse a módulos de esquema de componente común.
[SD-02]	El módulo de esquema de componente común NO DEBE referirse a los módulos de esquema de componente de patente, a los módulos de esquema de componente de marca y a los módulos de esquema de componente de dibujos y modelos industriales.
[SD-03]	Los módulos de esquema de componente de patente, de componente de marca y de componente de dibujos y modelos industriales NO DEBEN referirse los unos a los otros.
[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 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>redefine</code> para la redefinición.
[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.



Annex I, page 25

ID de la Norma	Norma
[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 URL.
[SD-13]	Todos los esquemas DEBEN tener unos atributos <code>elementFormDefault</code> y <code>attributeFormDefault</code> con un valor "qualified"
[SD-14]	Los nombres de los espacios de nombres DEBEN tener la siguiente estructura: <code>http://www.wipo.int/standards/XMLSchema/&lt;category&gt;/&lt;major&gt;</code> ; donde <code>&lt;category&gt;</code> es un valor que identifica el ámbito del esquema que se utiliza: <code>Common</code> , <code>Patent</code> , <code>Trademark</code> , o <code>Design</code> ; y <code>&lt;major&gt;</code> , el número de versión mayor.
[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 sus espacios 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 calificadores de espacio de nombres declarados, pero no con el calificador del W3C.
[SD-18]	Los esquemas de componente común DEBEN utilizar como espacio de nombres de destino <code>http://www.wipo.int/standards/XMLSchema/Common/&lt;major&gt;</code> .
[SD-19]	Los esquemas de componente de patente DEBEN utilizar como espacio de nombres de destino <code>http://www.wipo.int/standards/XMLSchema/Patent/&lt;major&gt;</code> .
[SD-20]	Los esquemas de componente de marca DEBEN utilizar como espacio de nombres de destino <code>http://www.wipo.int/standards/XMLSchema/Trademark/&lt;major &gt;</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/Design/&lt;major&gt;</code> .
[SD-22]	Los esquemas NO DEBERÍAN utilizar espacios de nombres por defecto.
[SD-23]	Las nuevas versiones menores de esquemas DEBEN poder validarse 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 proyecto de esquema no puede validar los documentos resultantes de una aplicación de XML existentes creados con la versión mayor actual.
[SD-25]	NO SE DEBERÍAN modificar las versiones menores de un esquema salvo que se altere el esquema o uno de los esquemas importados/incluidos para introducir cambios menores.
[SD-26]	En el caso de los esquemas, el nombre de archivo de esquema, el atributo <code>version</code> del W3C y el espacio de nombres DEBEN contener información coincidente sobre la versión.
[SD-27]	La versión de un esquema DEBERÍA modificarse cuando se actualice un esquema incluido o importado.
[SD-28]	Cuando una construcción de un esquema se vea alterada para introducir cambios mayores en un espacio de nombres determinado, DEBE incrementarse la versión mayor de todos los esquemas de dicho espacio de nombres.
[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 incluir un número de versión mayor y un número de versión menor en cada archivo de esquema en el formato siguiente: <code>&lt;número de versión mayor&gt;.&lt;número de versión menor&gt;</code> .
[SD-31]	Los esquemas definidos como componentes de documento DEBEN definir en su elemento raíz un atributo requerido denominado <code>schemaVersion</code> con el formato siguiente: <code>&lt;número de versión mayor&gt;.&lt;número de versión menor&gt;</code> .
[SD-32]	Los esquemas ST.96 DEBERÍAN elaborarse con un nivel similar 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.

Annex I, page 26

ID de la Norma	Norma
[SD-33]	Las enumeraciones contenidas en los esquemas NO DEBERÍAN incluir los valores "Other" o "Undefined" a menos que sea necesario.
[SD-34]	DEBERÍA utilizarse MathML, versión 3, para las fórmulas matemáticas.
[SD-35]	En el caso de los cuadros, DEBERÍA utilizarse OASISTable.xsd, que se basa en el esquema de XML 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]	Los tipos simples DEBERÍAN utilizarse para definir las listas de códigos que utilizan la enumeración.
[SD-39]	Los tipos simples PUEDEN utilizar <code>xsd:union</code> para mejorar una lista de códigos.
[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 valores de fecha y hora de las aplicaciones resultantes de un esquema DEBEN adaptarse a los tipos de datos de fecha y hora del esquema de XML 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 listas de enumeraciones DEBEN limitarse al conjunto siguiente: {a-z, A-Z, 0-9, punto (.), coma (,), espacios, guión (-), símbolo & (&#amp;#amp;) y subrayado ( <u> </u> }.
[SD-46]	Los valores disponibles en las listas de enumeraciones 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 cuando definan datos no operativos. Por ejemplo, el atributo <code>sequenceNumber</code> .
[SD-49]	Los esquemas NO DEBERÍAN utilizar indicadores de ocurrencia para los valores por defecto (es decir, <code>minOccurs="1"; maxOccurs="1"</code> ).
[SD-50]	Los elementos vacíos NO DEBEN definirse en los esquemas, a excepción del salto de línea.
[SD-51]	Los esquemas NO DEBERÍAN utilizar el indicador <code>use</code> para un atributo cuando el valor esperado sea el valor por defecto (es decir, <code>use="optional"</code> ).
[SD-52]	Los esquemas NO DEBERÍAN utilizar el operador <code>all</code> .
[SD-53]	Los esquemas PUEDEN utilizar indicadores de ocurrencia en componedores <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> para las limitaciones de identidad.
[SD-57]	Los esquemas NO DEBERÍAN utilizar <code>ID/IDREF/IDREFS</code> para las limitaciones de identidad.
[SD-58]	Todos los esquemas DEBERÍAN utilizar el elemento de documentación del W3C 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.

Annex I, page 27

ID de la Norma	Norma
[SD-61]	Los esquemas DEBEN incluir documentación del encabezado del esquema que incluya referencias a la última actualización, el punto de contacto y la URL para las notas explicativas.

*Normas de diseño de aplicación*

ID de la Norma	Norma
[ID-01]	Las aplicaciones de XML DEBEN validarse con el esquema correspondiente.
[ID-02]	Los documentos resultantes de una aplicación de XML DEBEN utilizar el atributo <code>schemaLocation</code> para describir el espacio de nombres y el nombre de esquema con el que está conforme la aplicación de XML.
[ID-03]	Toda declaración de atributo <code>xsi:schemaLocation</code> DEBERÍA contener una 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 resultantes de una aplicación de XML NO DEBEN utilizar espacios de nombres por defecto.
[ID-06]	Los documentos resultantes de una aplicación de 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 resultantes de una aplicación de XML NO DEBEN utilizar declaraciones de espacio de nombres local.
[ID-08]	Las entidades externas que sean imágenes 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.

**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.	xsd:token
Date (Fecha)	Punto concreto en el tiempo expresado mediante el año, el mes y el día.	DateType
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)	Numeral o cadena de numerales que expresan una etiqueta, un valor, una cantidad o una identificación.	xsd:positiveInteger o xsd:string
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. QuantityType se utiliza para representar un número de cosas contadas. QuantityType debe utilizarse para contar o cuantificar los componentes que integran un objeto percibido como un compuesto, una colección o un contenedor. QuantityType siempre debe expresar un número de cosas contadas, y el número se referirá a la cantidad total, enviada, cargada o almacenada.	QuantityType
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
Time (Tiempo)	Designación de un punto cronológico concreto en un período.	xsd:time
Timestamp (Marca temporal)	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

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

ALT	Texto alternativo para la imagen
B	Negrita
BioDeposit	Depósito biológico
Br	Salto
DD	Descripción de definición
DL	Lista de definición
DOI	Identificador de objeto digital
DT	Término de definición
DTD	Definición de tipo de documento
I	Cursiva
ID	Identificador para la identificación del sistema
IDREF	Referencia del identificador
IP	Propiedad Industrial
IPC	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	Punto de la lista
NPL	Documentos distintos de los de patentes
O	Tachado
OCR	Reconocimiento óptico de caracteres
OL	Lista ordenada
P	Párrafo
Pre	Texto preformateado
Sub	Subíndice
Sup	Superíndice
SWIFT	Sociedad de Telecomunicaciones Interbancarias Mundiales
U	Subrayado
UCC	<i>Upper Camel Case</i>
UL	Lista no ordenada
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
XML	Lenguaje extensible de marcado

## **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.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]