

Comité de Normas Técnicas de la OMPI (CWS)

Décima sesión
Ginebra, 21 a 25 de noviembre de 2022

PROPUESTA DE UNA NUEVA NORMA DE LA OMPI SOBRE JSON

Documento preparado por la Oficina Internacional

RESUMEN

1. En nombre del Equipo Técnico de XML4IP, la Oficina Internacional presenta un borrador final para una nueva norma de la OMPI sobre la notación de objetos en JavaScript (JSON), para que sea examinado en la décima sesión del Comité de Normas Técnicas de la OMPI (CWS). El borrador final incluye una serie de reglas de diseño, así como un conjunto de esquemas JSON, que se basan en los esquemas XML de la Norma ST.96 de la OMPI, y ejemplos de instancias JSON.

INTRODUCCIÓN

2. En su quinta sesión de 2017, el CWS aprobó la adición de la nueva Tarea n.º 56, con el fin de proporcionar el marco para que el Equipo Técnico XML4IP comenzara a trabajar en una nueva norma de la OMPI que proporcione un conjunto de recomendaciones para el procesamiento y la comunicación de datos de propiedad intelectual (PI) utilizando interfaces de programación de aplicaciones (API). En la descripción de esa tarea se indica que se puede utilizar la notación JSON como carga útil.

3. En su séptima sesión, el CWS creó la nueva Tarea n.º 64, cuya descripción se reproduce a continuación, y asignó la tarea al Equipo Técnico XML4IP (párrafos 58 y 59 del documento CWS/7/29):

"Preparar una propuesta de recomendaciones de recursos de notación de objetos en JavaScript (JSON) compatibles con la Norma ST.96 de la OMPI a fin de que sean utilizados para la presentación, tratamiento, publicación y/o intercambio de información sobre propiedad intelectual."

4. En su octava sesión, celebrada en 2020, el CWS aprobó la nueva Norma ST.90 de la OMPI, "Recomendación para el procesamiento y la comunicación de datos de propiedad intelectual mediante el uso de API (interfaces de programación de aplicaciones) en la web". La Norma ST.90 de la OMPI incluye ejemplos tanto en formato XML como JSON. Si bien la Norma ST.90 remite a la Norma ST.96 como referencia para los esquemas XML, no hay ninguna norma de referencia para JSON, ya que en ese momento no existía ninguna norma de la OMPI sobre JSON. La Norma ST.90 incluye una nota a pie de página que dice "La norma JSON de la OMPI es objeto de debate, pero se basará en la Norma ST.96 de la OMPI".

5. En su novena sesión, celebrada en 2021, el CWS señaló que un borrador final de la nueva norma sobre JSON estaría disponible para su examen y aprobación en la décima sesión (párrafo 20 del documento CWS/9/25). Desde la novena sesión del CWS, en el marco de la Tarea n.º 64, el Equipo Técnico de XML4IP ha conseguido finalizar el borrador de la norma tras haber debatido el tema en reuniones, correos electrónicos y en la wiki.

PROPUESTA DE NUEVA NORMA DE LA OMPI

6. Las Oficinas de PI y el sector de la PI han adoptado y utilizado JSON progresivamente, mientras que el XML (lenguaje de marcado extensible), basado en las normas XML de la OMPI, sigue siendo ampliamente utilizado. La Norma ST.96 de la OMPI define una recomendación sobre cómo deben capturarse los recursos XML durante la presentación, la publicación, la tramitación y el intercambio de información sobre diversos tipos de PI, es decir, patentes, marcas, diseños industriales, indicaciones geográficas y derechos de autor. Las Oficinas de PI aplican la Norma ST.96 tal como se ha publicado o bien la adaptan según sus necesidades.

7. El Equipo Técnico de XML4IP desarrolló el borrador de la norma JSON teniendo en cuenta la necesidad de coherencia y compatibilidad de los datos entre los formatos XML y JSON, con el fin de facilitar el intercambio de datos entre las Oficinas de PI y la difusión de datos por parte de estas en estos dos formatos. Esta coherencia y compatibilidad de los datos puede lograrse utilizando los esquemas XML y JSON compatibles que se utilizarán para validar las instancias XML y JSON respectivamente.

8. En el momento de la elaboración de este documento, todavía no existe una norma JSON internacional aprobada por el sector. Solo existen borradores de especificaciones y el formato JSON sigue evolucionando. El [borrador 2020-12](#) es la última versión de la especificación del esquema JSON y la versión 5.0 es la última Norma ST.96 de la OMPI. Por consiguiente, la norma JSON propuesta se basa en este borrador de la especificación del esquema JSON y en la versión 5.0 de la Norma ST.96 de la OMPI. Cabe señalar que la nueva versión 6.0 de la Norma ST.96 se publicará en octubre de 2022.

Objetivos

9. Esta propuesta de norma tiene por objeto proporcionar un conjunto de recomendaciones para la presentación de datos de propiedad intelectual en formato JSON. Los principales objetivos de la norma son ofrecer las siguientes ventajas:

- Orientar sobre el marcado de datos en formato JSON;
- Garantizar la coherencia mediante esquemas e instancias JSON basados en la Norma ST.96 de la OMPI para el intercambio de datos de PI;
- Recomendar principios de diseño para ampliar los esquemas JSON proporcionados o crear nuevos esquemas JSON compatibles; y
- Mejorar la eficiencia del intercambio de datos promoviendo la reutilización de los recursos JSON entre las Oficinas de PI, así como de los datos proporcionados al público.

Ámbito de aplicación

10. Esta propuesta de norma tiene por objeto orientar a las Oficinas de PI y a otras organizaciones que crean o modifican datos de PI como recursos JSON. La compatibilidad con esta norma es necesaria para el intercambio de datos entre las Oficinas de PI que utilizan JSON como carga útil para las API.

11. La propuesta de norma se estructura de la siguiente manera:

- **Cuerpo principal:** definición de las reglas generales de diseño, reglas de diseño del esquema JSON y de la construcción del esquema, identificadores del esquema JSON y reglas de diseño de la instancia JSON;
- **Anexo I:** reglas de transformación de esquemas XML ST.96 a esquemas JSON, que contienen el apéndice "Herramienta de transformación de esquemas XSD ST.96 a esquemas JSON";
- **Anexo II:** esquemas JSON transformados a partir de los esquemas XML de la Norma ST.96 de la OMPI, versión 5.0;
- **Anexo III:** instancias JSON;
- **Anexo IV:** siglas y abreviaturas basadas en la Norma ST.96 de la OMPI; y
- **Anexo V:** términos de representación basados en la Norma ST.96 de la OMPI.

12. La Oficina Internacional propone el siguiente nombre para la nueva norma de la OMPI:

"Norma de la OMPI ST.97: Recomendación para el procesamiento de datos de propiedad intelectual mediante JSON (notación de objetos en JavaScript)"

13. El CWS debe tener en cuenta que esta propuesta de norma JSON excluye los aspectos relacionados con la arquitectura informática y los lenguajes de aplicación. La norma propuesta, incluidos todos los anexos definidos anteriormente, se reproducen como anexo del presente documento.

Desarrollo futuro

14. Al preparar la propuesta de norma JSON, el Equipo Técnico de XML4IP detectó y resolvió muchos problemas. No obstante, quedan varias cuestiones pendientes que deberían resolverse junto con los cambios derivados de la evolución de la especificación del esquema JSON, entre ellas:

- Simplificar la estructura de los esquemas JSON de acuerdo con las prácticas del sector, por ejemplo, eliminando el anidamiento innecesario en el esquema JSON que fue el resultado de la transformación de los esquemas XML ST.96;
- Mejorar el diseño de los datos para lograr una validación más precisa, lo que requerirá un análisis por parte de los expertos en la materia;
- Actualizar el conjunto de esquemas JSON de acuerdo con los cambios introducidos en la versión 6.0 de la Norma ST.96, que se publicará en octubre de 2022;
- Añadir reglas de transformación y herramientas pertinentes para las instancias entre XML y JSON; y
- Revisar la nueva norma, si es necesario, para tener en cuenta los cambios en la especificación del esquema JSON, ya sea en el próximo borrador o en la versión oficial.

REVISIÓN DE LA TAREA N.º 64

15. Una vez que la nueva norma propuesta sobre JSON haya sido aprobada por el CWS, la Tarea n.º 64 deberá considerarse terminada y el Equipo Técnico de XML4IP habrá concluido la labor de esta tarea. No obstante, como se ha indicado anteriormente, esta norma debería mantenerse de cara a futuras revisiones. A este respecto, se propone revisar la descripción de la Tarea n.º 64 de la siguiente manera:

“Garantizar las revisiones y actualizaciones necesarias de la Norma ST.97 de la OMPI”

MANTENIMIENTO DE LA NUEVA NORMA DE LA OMPI

16. Si bien el Equipo Técnico XML4IP ha concluido la Tarea n.º 64 y ha elaborado esta primera versión de la norma JSON de la OMPI a partir de la Norma ST.96 sobre XML, la Oficina Internacional propone asignar al Equipo Técnico de las API la Tarea n.º 64 revisada, ya que también gestiona la Norma ST.90 de la OMPI. Esto se debe a que JSON se utiliza muy a menudo como carga útil para las API.

17. Habida cuenta de la continua revisión y actualización de la Norma ST.96 de la OMPI, el CWS estableció el procedimiento de "vía rápida" para el examen y/o la aprobación de las revisiones de la norma por el Equipo Técnico de XML4IP. Como está previsto que la nueva norma sobre JSON se revise continuamente junto con las revisiones de la Norma ST.96 y la evolución de la [especificación del esquema JSON](#), se propone establecer otro procedimiento de "vía rápida" para el examen y/o la aprobación de las revisiones de la nueva norma JSON, como se indica a continuación:

- a) las propuestas de revisión de la Norma ST.97 de la OMPI se presentarán directamente o a través de la Secretaría al Equipo Técnico designado para su examen y aprobación;
- b) el Equipo Técnico designado está temporalmente autorizado a aprobar las revisiones de la Norma ST.97 de la OMPI;
- c) si una propuesta de revisión de la Norma ST.97 de la OMPI resulta controvertida, se presentará al CWS para que la examine, por ejemplo, cuando no sea posible alcanzar un consenso entre los miembros designados del Equipo Técnico; y
- d) el jefe designado del Equipo Técnico informará al CWS de toda revisión de la Norma ST.97 de la OMPI aprobada por el Equipo Técnico en la siguiente sesión del CWS.

MODIFICACIONES EN LA REDACCIÓN DE LA NORMA ST.90 DE LA OMPI

18. Puesto que el Equipo Técnico de las API no ha presentado ningún documento en esta sesión, la Oficina Internacional ha señalado varias modificaciones en la redacción que son necesarias para la Norma ST.90 si se aprueba la Norma ST.97 propuesta. Estas propuestas de modificación de la redacción que se someten a la consideración del CWS incluyen:

- Añadir a la sección de referencias una referencia a la nueva Norma ST.97 de la OMPI;
- Modificar el párrafo 33 de la Norma ST.90 de la OMPI para introducir una referencia a la Norma ST.97 de la OMPI, añadiendo la frase subrayada, de modo que el párrafo diga:

"Las API deben admitir peticiones y respuestas XML y JSON. En el caso de XML, las respuestas deben ser compatibles con las normas de la OMPI que utilizan XML, como la Norma ST.96, y en el caso de JSON, las respuestas deben ser compatibles con la Norma ST.97 de la OMPI. Estos dos formatos deberían estar correlacionados de manera coherente."; y

- Eliminar la nota a pie de página 7 del párrafo 33 de la Norma ST.90, que se reproduce a continuación.

"Actualmente, el Equipo Técnico de XML4IP está examinando una especificación JSON y un esquema JSON basados en la Norma ST.96, con el objetivo de que se examinen en la octava sesión del CWS, en noviembre de 2020, y se considere su adopción como una nueva norma de la OMPI. Mientras tanto, esta norma recomienda la convención BadgerFish debido a su simplicidad hasta que se proporcione el esquema JSON. Algunas Oficinas de PI, como la OEP, también hacen referencia a ella: www.epo.org/searching-for-patents/data/web-services/ops.html."

19. Las modificaciones de redacción propuestas en el párrafo 18 no son exhaustivas. Por consiguiente, se sugiere que la Secretaría introduzca las modificaciones necesarias en la redacción, según proceda, antes de volver a publicar la nueva versión revisada de la Norma ST.90 de la OMPI.

20. Se invita al CWS a:

a) *tomar nota del contenido de este documento y de su anexo (el borrador final de la norma JSON);*

b) *examinar y aprobar el nombre de la norma propuesta "Norma ST.97 de la OMPI: Recomendación para el procesamiento de datos de propiedad intelectual mediante JSON", como se indica en el párrafo 12;*

c) *examinar y aprobar la nueva Norma ST.97 de la OMPI, que se reproduce en el anexo del presente documento;*

d) *examinar y aprobar la revisión de la descripción de la Tarea n.º 64, como se indica en el párrafo 15;*

e) *designar al Equipo Técnico de las API para que se ocupe de la revisión de la Tarea n.º 64, como se indica en el párrafo 16;*

f) *examinar y aprobar la vía rápida para revisar la Norma ST.97 de la OMPI, como se indica en el párrafo 17; y*

g) *aprobar la modificación de la Norma ST.90 de la OMPI para que haga referencia a la Norma ST.97 adoptada y solicitar a la Secretaría que introduzca otras modificaciones necesarias en la redacción y publique la Norma ST.90 revisada, como se indica en los párrafos 18 y 19.*

[Sigue el anexo (borrador de la norma JSON)]

NORMA ST.XX DE LA OMPI**RECOMENDACIONES SOBRE EL TRATAMIENTO EN JSON DE DATOS DE PROPIEDAD INTELECTUAL**

Propuesta presentada por el Equipo Técnico de XML4IP para su examen en la décima sesión del Comité de Normas Técnicas de la OMPI

Nota editorial: con el fin de facilitar la comprensión de determinadas normas, se han incluido notas a lo largo del proyecto de norma propuesta, que serán eliminadas del documento cuando se publique la Norma tras su adopción.

ÍNDICE

NORMA ST.XX DE LA OMPI.....	1
ÍNDICE	1
ÍNDICE DE CUADROS.....	2
1. INTRODUCCIÓN	3
2. DEFINICIONES Y TERMINOLOGÍA.....	3
3. NOTACIONES GENERALES	3
3.1 Identificadores de normas	3
3.2 Ejemplo de estructura de datos JSON.....	4
4. ALCANCE	4
5. NORMAS GENERALES DE DISEÑO DE JSON	4
5.1 Información general.....	4
5.2 Convenciones de denominación en JSON	4
6. NORMAS DE DISEÑO DE ESQUEMAS JSON	6
6.1 Información general.....	6
6.2 Modularidad.....	6
6.3 Documentación	14
6.4 Nombre de archivo	15
6.5 Estructuración de las propiedades del esquema JSON	15
7. NORMAS DE DISEÑO DE LAS CONSTRUCCIONES DE ESQUEMA JSON	16
7.1 Información general.....	16
7.2 Propiedades	16
7.3 Definiciones.....	16
7.4 Definiciones de tipo	17
7.5 Tipo primitivo JSON.....	17
7.6 Listas de códigos.....	17
7.7 Matrices.....	18
7.8 Objetos.....	18
8 IDENTIFICADORES DE ESQUEMA JSON	19
8.1 Información general.....	19
9. NORMAS DE DISEÑO DE INSTANCIAS JSON	19
9.1 Ordenación de las propiedades.....	19
9.2 Validación de instancias JSON.....	19
8. MATERIAL DE REFERENCIA	19
Normas de la OMPI.....	19
Normas y especificaciones del sector.....	19
ST.XX - ANEXO I.....	21
NORMAS SOBRE LA TRANSFORMACIÓN DE LA DEFINICIÓN DE ESQUEMA XML (XSD) DE LA NORMA ST.96	21
EN ESQUEMA JSON Y DIRECTRICES DE USO	21
INTRODUCCIÓN	21
ALGORITMO DE TRANSFORMACIÓN Y DIRECTRICES.....	21
ALCANCE.....	21
TRANSFORMACIÓN DE NOMBRES DE ARCHIVO.....	21
TRANSFORMACIÓN DE ESPACIOS DE NOMBRES	21
TRANSFORMACIÓN DE TIPOS DE DATOS INTEGRADOS	22
DEFINICIÓN DE ESQUEMA XML (XSD).....	23
REFERENCIACIÓN DE ESQUEMAS GLOBALES	29
INSTRUCCIONES DE COMPOSICIÓN XSD	30
ELEMENTOS.....	35
ATRIBUTOS	36
SIMPLETYPE	37
COMPLEXTYPE	37

<i>Contenido simple</i>	37
<i>Contenido complejo</i>	37
<i>Contenido mixto</i>	37
ANOTACIÓN	39
<i>xsd:appinfo</i>	39
<i>xsd:documentation</i>	40
UNIÓN	40
AMPLIACIÓN	41
RESTRICCIÓN	42
ENUMERACIÓN.....	42
FACETAS DE RESTRICCIÓN.....	43
<i>Patrón</i>	44
GRUPO.....	44
TRANSFORMACIÓN DE DEPENDENCIAS EXTERNAS DE LAS XSD	45
APÉNDICE	46
<i>Requisitos</i>	46
<i>Uso</i>	46
<i>Descarga del archivo JAR ejecutable</i>	46
ANEXO II	47
ESQUEMA JSON	47
ANEXO III	48
EJEMPLOS DE INSTANCIAS JSON	48
ANEXO IV	49
LISTA DE SIGLAS Y ABREVIATURAS	49
ANEXO V	51
TÉRMINOS DE REPRESENTACIÓN	51

ÍNDICE DE CUADROS

Cuadro 1. Elementos de la documentación del encabezado del esquema JSON.	14
Cuadro 2. Transformación de tipos simples de datos XSD.....	22
Cuadro 3. Transformación de las facetas de restricción en las XSD ("X" es el valor numérico de la restricción).	43

1. INTRODUCCIÓN

En la presente Norma se ofrecen recomendaciones para el diseño, la creación o la actualización de recursos de notación de objetos de JavaScript (JavaScript Object Notation, JSON) para utilizarlos en la presentación, el tratamiento, el intercambio o la publicación de todo tipo de datos de propiedad intelectual (PI). Asimismo, se incluyen normas para transformar las definiciones de esquema de lenguaje extensible de marcado (definiciones de esquema XML [XSD]) de la Norma ST.96 de la OMPI en esquemas JSON que se ajusten a las referidas recomendaciones.

La finalidad de esta Norma es:

- Proporcionar directrices sobre el marcado de datos en formato JSON;
- Garantizar la coherencia ofreciendo esquemas e instancias JSON basados en la Norma ST.96 de la OMPI para el intercambio de datos de PI;
- Recomendar principios de diseño para ampliar los esquemas JSON proporcionados o crear nuevos esquemas JSON conformes con la Norma; y
- Mejorar la eficiencia del intercambio de datos fomentando la reutilización de recursos de JSON entre las Oficinas de Propiedad Intelectual (OPI), así como el suministro de datos al público.

2. DEFINICIONES Y TERMINOLOGÍA

En esta Norma se utiliza la siguiente terminología:

- Los términos "object", "object type", "property", "member", "property name", "property value", "property type", "keyword" y "definition" en esta Norma han de interpretarse según se definen en JSON Schema Core (versión draft-2020-12);¹
- El término "construcción" en esta Norma debe interpretarse como un bloque de construcción a partir del cual se construyen los esquemas JSON;
- El término "definición global" hace referencia a una definición a la que pueden referirse otras definiciones del mismo esquema o definiciones de otros esquemas; y
- Las palabras DEBE, NO DEBE, DEBERÍA, NO DEBERÍA, RECOMENDADO, PUEDE y OPCIONAL han de interpretarse según se describe en el documento [RFC 2119](#). Cuando aparezcan en minúscula, se utilizarán en el sentido habitual en español.

3. NOTACIONES GENERALES

A lo largo de la presente Norma se utilizan las siguientes notaciones:

- `<>`: Indica un término descriptivo provisional que, en la aplicación, se sustituirá por un valor específico de instancia;
- `" "`: Indica que el texto entre comillas debe usarse de manera literal en la aplicación;
- `{ }`: Indica que la aplicación del elemento en cuestión es facultativa; y
- La fuente `Courier New` se utiliza para palabras clave JSON, nombres de propiedades de JSON y elementos y atributos XSD.

3.1 Identificadores de normas

Todas las normas de diseño son normativas. Las normas de diseño se identifican mediante el prefijo [JXX-nn].

- El valor "JXX" es un prefijo para clasificar los tipos de normas según se indica a continuación:
 - a) JGD para las normas generales de diseño;
 - b) JSD para las normas de diseño de esquemas JSON;
 - c) JCD para las normas de diseño de construcciones; y
 - d) JID para las normas de diseño de instancias.
- El valor "nn" indica el número siguiente disponible en la serie de un tipo específico de norma. Cabe observar que el número no indica la posición de la norma, en particular para una nueva norma. Una nueva norma se

¹ La versión del esquema JSON puede ser objeto de modificaciones porque no ha alcanzado el estado de RFC; no ha sido adoptada por un grupo de trabajo del IETF. La presente versión de la Norma se basa en la última versión del esquema JSON, a saber, la versión 2020-12, disponible en <https://json-schema.org/draft/2020-12/json-schema-core.html>

colocará en el contexto pertinente. Por ejemplo, el identificador de norma [JGD-10] identifica la décima norma general de diseño. La norma [JGD-10] puede colocarse entre las normas [JGD-05] y [JGD-06] en lugar de a continuación de la norma [JGD-09] si esa es la ubicación más adecuada para esa norma.

- El identificador de norma de una norma suprimida se mantendrá, pero la norma será sustituida por el texto “Suprimida”.

3.2 Ejemplo de estructura de datos JSON

Los ejemplos de estructuras de datos JSON se proporcionan en una fuente monoespaciada dentro de cuadros de texto. Se han resaltado los ejemplos de sintaxis de estructuras de datos JSON para que se puedan distinguir con mayor facilidad.

4. ALCANCE

La finalidad de esta Norma es proporcionar orientación a las OPI y a otras organizaciones que crean o modifican datos de PI como recursos de JSON. Para el intercambio de datos entre las OPI mediante mensajes JSON se requiere el cumplimiento de esta Norma.

Las normas de diseño y las normas sobre transformación se han redactado teniendo en cuenta las normas de diseño y las convenciones de la Norma ST.96 de la OMPI. La correspondencia entre las normas de diseño y convenciones de la Norma ST.96 y las normas de diseño y convenciones de JSON no es biunívoca, por lo que algunas normas de diseño de la Norma ST.96 se duplican y, si es necesario, se modifican ligeramente.

La presente Norma incluye los siguientes Anexos:

- [Anexo I](#): Normas sobre la transformación de los esquemas XML de la Norma ST.96 en esquemas JSON. Incluye el Apéndice titulado “Herramienta de transformación de la definición de esquema XML (XSD) de la Norma ST.96 en esquema JSON”;
- [Anexo II](#): Esquemas JSON obtenidos de la transformación de esquemas XML de la Norma ST.96 de la OMPI (versión 5.0);²
- [Anexo III](#): Instancias JSON;
- [Anexo IV](#): Siglas y abreviaturas; y
- [Anexo V](#): Términos de representación.

En esta Norma no se abarcan cuestiones de:

- a) arquitectura de *software*; y
- b) lenguajes de implementación.

5. NORMAS GENERALES DE DISEÑO DE JSON

5.1 Información general

En esta sección se describen las normas de diseño y las directrices de JSON generales y de alto nivel que se aplican a cualquier intercambio de datos JSON e iniciativa de desarrollo de JSON, en lugar de a un lenguaje de programación específico de empaquetado y desempaquetado de datos en objetos JSON. Las normas y directrices generales que se proporcionan son la base común para el desarrollo de esquemas JSON, instancias JSON y estructuras de datos JSON aplicables a todos los datos, sean o no de PI, como los datos de contenido mixto. El nivel de anidamiento DEBERÍA mantenerse lo más bajo posible cuando se desarrollen nuevos esquemas JSON, instancias JSON y estructuras de datos JSON que no estén disponibles en la Norma ST.96 de la OMPI o cuya compatibilidad con la Norma ST.96 no sea necesaria.

5.2 Convenciones de denominación en JSON

Las convenciones de denominación son necesarias para garantizar la coherencia, la uniformidad y la exhaustividad en la denominación y definición de todos los recursos de JSON.

Estas convenciones de denominación en JSON se basan en las directrices y los principios descritos en el documento [ISO 11179 Part 5 - Naming and Identification Principles](#). El nombre de objetos y los nombres de propiedades se componen de los siguientes términos:

² Los esquemas JSON obtenidos tienen los mismos nombres de etiquetas y la misma estructura de datos que se definen en el Anexo III de la Norma ST.96, incluidos los componentes XML de contenido mixto y las normas externas, como MathML y Oasis Table, para la interoperabilidad con los datos en el formato de la Norma ST.96.

- Por *clase de objeto* se entenderá 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.
- El *término de propiedad* identifica características de la clase de objeto.
- 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, si fuera necesario, podrá adjuntarse a un término de clase de objeto o un término de propiedad para individualizar un nombre.
- El *término de representación* clasifica el formato del elemento de datos en categorías más amplias. Deberían utilizarse los términos de representación que figuran en el Anexo V.

- [JGD-01] Los nombres de propiedades y de tipos de objetos DEBEN estar compuestos de palabras del idioma inglés, utilizando la grafía inglesa básica que figura en el Diccionario Inglés Oxford. Las únicas excepciones permitidas son las siglas, las abreviaturas y otras formas truncadas que figuran en el [Anexo IV](#).
- [JGD-02] Los nombres de propiedades y de tipos de objetos DEBERÍAN estar compuestos únicamente por sustantivos, adjetivos y verbos en presente, con la excepción de las siglas, las abreviaturas y otras formas truncadas que figuran en el [Anexo IV](#).
- [JGD-03] Los caracteres empleados en los nombres de propiedades DEBEN estar contenidos en el conjunto siguiente: 'a-z, A-Z y 0-9'.
- [JGD-04] La longitud máxima de los nombres de propiedades y de tipos de objetos DEBERÍA ser de no más de 35 caracteres.
- [JGD-05] Los nombres de propiedades y de tipos de objetos DEBERÍAN ser concisos y explicarse por sí solos.
- [JGD-06] Los nombres de propiedades y de tipos de objetos DEBEN seguir la convención *Lower Camel Case* (LCC). Por ejemplo, "currencyCode": "EUR".
- [JGD-07] Los nombres de tipos de objetos DEBEN seguir la convención LCC y tener el sufijo Type. Por ejemplo, applicantType.
- [JGD-08] Las siglas y las abreviaturas que figuran en el Anexo IV siempre DEBEN utilizarse en lugar del nombre completo ampliado.
- [JGD-09] Las siglas y las abreviaturas DEBEN aparecer según lo indicado en el Anexo IV para los nombres de propiedades y de tipos de objetos.
- [JGD-10] Un término de clase de objeto siempre DEBE tener el mismo significado semántico en un ámbito específico de la PI, como las patentes, las marcas, los diseños industriales, las indicaciones geográficas o el derecho de autor, y PUEDE componerse de más de una palabra. Por ejemplo, contactInformation.
- [Nota: la expresión "espacio de nombres" que figura en la Norma ST.96 ha sido sustituida por "ámbito de la PI" en JSON.]
- [JGD-11] Un término de propiedad de un nombre DEBE ser exclusivo en el contexto de una clase de objeto, pero PUEDE reutilizarse en distintas clases de objetos.
- [JGD-12] Un término calificador PUEDE adjuntarse a un término de clase de objeto o a un término de propiedad si fuera necesario para individualizar un nombre.
- [JGD-13] Cuando un nombre contiene un término de clase de objeto, un término de propiedad y un término de representación, el término de clase de objeto DEBE anteceder al término de propiedad, y el término de propiedad DEBE anteceder al término de representación. Un término calificador DEBERÍA anteceder al término de propiedad o al término de clase de objeto asociado. Por ejemplo, claimTotalQuantity.
- [JGD-14] Si el término de propiedad acaba con la misma palabra que el término de representación (o una palabra equivalente), DEBE suprimirse el término de representación.
- [JGD-15] Cuando es necesario un término de representación, DEBEN emplearse los términos de representación del Anexo V para los términos de representación en los nombres de componentes básicos.
- [JGD-16] En un ámbito de la PI, todos los nombres de propiedades y de tipos de objetos DEBEN ser únicos.

- [JGD-17] La palabra o palabras de un nombre DEBERÍAN aparecer en singular salvo que el concepto mismo sea plural. Por ejemplo, `totalMarkSeries`.
- [JGD-18] El nombre de una propiedad o de un tipo de objeto que contenga una colección de componentes relacionados por el contexto DEBERÍA llevar el sufijo "Bag". Por ejemplo, `emailAddressBag` representa una colección de elementos `emailAddress`.
- [JGD-19] Conectores como "and", "of" o "the" NO DEBERÍAN utilizarse en los nombres de propiedades y de tipos de objetos salvo que formen parte de la terminología de la actividad.
- [JGD-20] Los nombres de propiedades y de tipos de objetos NO DEBEN traducirse, modificarse o sustituirse con ningún fin.
- [JGD-21] Los nombres de propiedades y de tipos de objetos NO DEBEN remitir a números de artículos o normas. Por ejemplo, `PCTRule702C` para el PCT.
- [JGD-22] El nivel de anidamiento DEBERÍA mantenerse lo más bajo posible cuando se desarrollen nuevos esquemas JSON, instancias JSON y estructuras de datos o fragmentos JSON que no estén disponibles en la Norma ST.96 de la OMPI.
- [JGD-23] Para los nombres nuevos de propiedades y de tipos de objetos que no estén definidos en la Norma ST.96 de la OMPI o que, previsiblemente, no serán compatibles con un nombre de componente que esté definido en la Norma ST.96, se DEBERÍAN utilizar términos o nombres descriptivos en línea en lugar de nombres genéricos o abreviados de propiedades o de tipos de objetos que sean de propiedad única. Por ejemplo, es preferible utilizar `"inventorFullName": "Thomas Edison"` en lugar de `"inventor": { "fullName": "Thomas Edison" }`.

[Nota: esta norma recomienda seguir la práctica propia de JSON en caso de que los nombres no estén relacionados con datos XML basados en las versiones actual o futuras de la Norma ST.96.]

6. NORMAS DE DISEÑO DE ESQUEMAS JSON

6.1 Información general

El esquema JSON describe la estructura de la instancia JSON, incluidas las restricciones sobre la estructura y el contenido del documento. La presente Norma debe ajustarse a la especificación del esquema JSON. La última versión disponible del esquema JSON en el momento de la publicación de esta Norma es la versión [draft 2020-12](#), por lo que la presente Norma se basa en ese borrador de la especificación del esquema JSON.

- [JSD-01] Los esquemas JSON DEBEN ajustarse a las especificaciones del esquema JSON: JSON Schema Core, versión 2020-12, disponible en <https://json-schema.org/latest/json-schema-core.html>, y JSON Schema Validation, versión 2020-12, disponible en <https://json-schema.org/latest/json-schema-validation.html>.
- [JSD-02] Los esquemas JSON DEBEN indicar que se ajustan a la versión 2020-12 del esquema JSON mediante la palabra clave `$schema` con el valor "<https://json-schema.org/draft/2020-12/schema>".

Ejemplo: indicación de la versión del esquema JSON
--

<code>"\$schema": "https://json-schema.org/draft/2020-12/schema"</code>

El esquema debe ser codificado utilizando UTF-8 para lograr la mayor interoperabilidad posible.

- [JSD-03] En los esquemas JSON DEBE usarse el conjunto de caracteres ISO/IEC 10646 - UCS – Unicode, y DEBE usarse UTF-8 para codificar los caracteres Unicode.

6.2 Modularidad

La modularidad permite la creación de componentes de esquema para facilitar la flexibilidad en el diseño y la reutilización. En el diseño, se recomienda no definir, en la medida de lo posible, todas las propiedades y los componentes lógicos en un único esquema JSON monolítico, ya que se reduciría la posibilidad de compartir y reutilizar las propiedades individuales o los componentes lógicos definidos como un grupo en un esquema.

A continuación figura un esquema que **no** se ajusta al principio de modularidad, lo que NO es recomendado por la presente Norma.

applicationNumber.json (ejemplo de documento de esquema compuesto inadecuado)

```
{
  "$id" : "applicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "applicationNumber" : {
      "$ref" : "#/$defs/applicationNumber"
    }
  },
  "required" : [ "applicationNumber" ],
  "$defs" : {
    "applicationNumber" : {
      "description" : "Description: Numbers used by IPOs in order to identify
each application received; Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "ipOfficeCode" : {
          "anyOf" : [ {
            "type" : "string",
            "enum" : [ "AD", "AE", "AF", "AG", "AI", "AL", "AM", "AO", "AP", "AR",
"AT", "AU", "AW", "AZ", "BA", "BB", "BD", "BE", "BF", "BG", "BH", "BI", "BJ",
"BM", "BN", "BO", "BQ", "BR", "BS", "BT", "BV", "BW", "BX", "BY", "BZ", "CA",
"CD", "CF", "CG", "CH", "CI", "CK", "CL", "CM", "CN", "CO", "CR", "CU", "CV",
"CW", "CY", "CZ", "DE", "DJ", "DK", "DM", "DO", "DZ", "EA", "EC", "EE", "EG",
"EH", "EM", "EP", "ER", "ES", "ET", "EU", "FI", "FJ", "FK", "FO", "FR", "GA",
"GB", "GC", "GD", "GE", "GG", "GH", "GI", "GL", "GM", "GN", "GQ", "GR", "GS",
"GT", "GW", "GY", "HK", "HN", "HR", "HT", "HU", "IB", "ID", "IE", "IL", "IM",
"IN", "IQ", "IR", "IS", "IT", "JE", "JM", "JO", "JP", "KE", "KG", "KH", "KI",
"KM", "KN", "KP", "KR", "KW", "KY", "KZ", "LA", "LB", "LC", "LI", "LK", "LR",
"LS", "LT", "LU", "LV", "LY", "MA", "MC", "MD", "ME", "MG", "MK", "ML", "MM",
"MN", "MO", "MP", "MR", "MS", "MT", "MU", "MV", "MW", "MX", "MY", "MZ", "NA",
"NE", "NG", "NI", "NL", "NO", "NP", "NR", "NZ", "OA", "OM", "PA", "PE", "PG",
"PH", "PK", "PL", "PT", "PW", "PY", "QA", "QZ", "RO", "RS", "RU", "RW", "SA",
"SB", "SC", "SD", "SE", "SG", "SH", "SI", "SK", "SL", "SM", "SN", "SO", "SR",
"SS", "ST", "SV", "SX", "SY", "SZ", "TC", "TD", "TG", "TH", "TJ", "TL", "TM",
"TN", "TO", "TR", "TT", "TV", "TW", "TZ", "UA", "UG", "US", "UY", "UZ", "VA",
"VC", "VE", "VG", "VN", "VU", "WO", "WS", "XN", "XU", "XV", "XX", "YE", "ZA",
"ZM", "ZW" ],

```

"description" : "Description: This code list is inline with WIPO Standard ST.3 (two-letter codes for the representation of states, other entities and organizations) published on September, 2019.; Version: V5_0; AD: Andorra; AE: United Arab Emirates; AF: Afghanistan; AG: Antigua And Barbuda; AI: Anguilla; AL: Albania; AM: Armenia; AO: Angola; AP: African Regional Intellectual Property Organization (ARIPO); AR: Argentina; AT: Austria; AU: Australia; AW: Aruba; AZ: Azerbaijan; BA: Bosnia and Herzegovina; BB: Barbados; BD: Bangladesh; BE: Belgium; BF: Burkina Faso; BG: Bulgaria; BH: Bahrain; BI: Burundi; BJ: Benin; BM: Bermuda; BN: Brunei Darussalam; BO: Bolivia (Plurinational State of); BQ: Bonaire, Sint Eustatius and Saba; BR: Brazil; BS: Bahamas; BT: Bhutan; BV: Bouvet Island; BW: Botswana; BX: Benelux Office for Intellectual Property (BOIP); BY: Belarus; BZ: Belize; CA: Canada; CD: Democratic Republic of the Congo; CF: Central African Republic; CG: Congo; CH: Switzerland; CI: Côte D'Ivoire; CK: Cook Islands; CL: Chile; CM: Cameroon; CN: China; CO: Colombia; CR: Costa Rica; CU: Cuba; CV: Cabo Verde; CW: Curaçao; CY: Cyprus; CZ: Czech Republic; DE: Germany; DJ: Djibouti; DK: Denmark; DM: Dominica; DO: Dominican Republic; DZ: Algeria; EA: Eurasian Patent Organization (EAPO); EC: Ecuador; EE: Estonia; EG: Egypt; EH: Western Sahara; EM: European Union Intellectual Property Office (EUIPO); EP: European Patent Office (EPO); ER: Eritrea; ES: Spain; ET: Ethiopia; EU: European Union; FI: Finland; FJ: Fiji; FK: Falkland Islands (Malvinas); FO: Faroe Islands; FR: France; GA: Gabon; GB: United Kingdom; GC: Patent Office of the Cooperation Council for the Arab States of the Gulf (GCC Patent Office); GD: Grenada; GE: Georgia; GG: Guernsey; GH: Ghana; GI: Gibraltar; GL: Greenland; GM: Gambia; GN:

Guinea; GQ: Equatorial Guinea; GR: Greece; GS: South Georgia and South Sandwich Islands; GT: Guatemala; GW: Guinea-Bissau; GY: Guyana; HK: Hong Kong, China; HN: Honduras; HR: Croatia; HT: Haiti; HU: Hungary; IB: International Bureau of the World Intellectual Property Organization (WIPO); ID: Indonesia; IE: Ireland; IL: Israel; IM: Isle of Man; IN: India; IQ: Iraq; IR: Iran, Islamic Republic of; IS: Iceland; IT: Italy; JE: Jersey; JM: Jamaica; JO: Jordan; JP: Japan; KE: Kenya; KG: Kyrgyzstan; KH: Cambodia; KI: Kiribati; KM: Comoros; KN: Saint Kitts and Nevis; KP: Democratic People's Republic of Korea; KR: Republic of Korea; KW: Kuwait; KY: Cayman Islands; KZ: Kazakhstan; LA: Lao People's Democratic Republic; LB: Lebanon; LC: Saint Lucia; LI: Liechtenstein; LK: Sri Lanka; LR: Liberia; LS: Lesotho; LT: Lithuania; LU: Luxembourg; LV: Latvia; LY: Libya; MA: Morocco; MC: Monaco; MD: Republic of Moldova; ME: Montenegro; MG: Madagascar; MK: North Macedonia; ML: Mali; MM: Myanmar; MN: Mongolia; MO: Macao, China; MP: Northern Mariana Islands; MR: Mauritania; MS: Montserrat; MT: Malta; MU: Mauritius; MV: Maldives; MW: Malawi; MX: Mexico; MY: Malaysia; MZ: Mozambique; NA: Namibia; NE: Niger; NG: Nigeria; NI: Nicaragua; NL: Netherlands; NO: Norway; NP: Nepal; NR: Nauru; NZ: New Zealand; OA: African Intellectual Property Organization (OAPI); OM: Oman; PA: Panama; PE: Peru; PG: Papua New Guinea; PH: Philippines; PK: Pakistan; PL: Poland; PT: Portugal; PW: Palau; PY: Paraguay; QA: Qatar; QZ: Community Plant Variety Office (European Community) (CPVO); RO: Romania; RS: Serbia; RU: Russian Federation; RW: Rwanda; SA: Saudi Arabia; SB: Solomon Islands; SC: Seychelles; SD: Sudan; SE: Sweden; SG: Singapore; SH: Saint Helena, Ascension and Tristan da Cunha; SI: Slovenia; SK: Slovakia; SL: Sierra Leone; SM: San Marino; SN: Senegal; SO: Somalia; SR: Suriname; SS: South Sudan; ST: Sao Tome and Principe; SV: El Salvador; SX: Sint Maarten (Dutch part); SY: Syrian Arab Republic; SZ: Eswatini; TC: Turks and Caicos Islands; TD: Chad; TG: Togo; TH: Thailand; TJ: Tajikistan; TL: Timor-Leste; TM: Turkmenistan; TN: Tunisia; TO: Tonga; TR: Turkey; TT: Trinidad and Tobago; TV: Tuvalu; TW: Taiwan, Province of China; TZ: United Republic of Tanzania; UA: Ukraine; UG: Uganda; US: United States of America; UY: Uruguay; UZ: Uzbekistan; VA: Holy See; VC: Saint Vincent and the Grenadines; VE: Venezuela (Bolivian Republic of); VG: British Virgin Islands; VN: Viet Nam; VU: Vanuatu; WO: World Intellectual Property Organization (WIPO) (International Bureau of); WS: Samoa; XN: Nordic Patent Institute (NPI); XU: International Union for the Protection of New Varieties of Plants (UPOV); XV: Visegrad Patent Institute (VPI); XX: Unknown states, other entities or organizations; YE: Yemen; ZA: South Africa; ZM: Zambia; ZW: Zimbabwe"

```

    }, {
      "type" : "string",
      "enum" : [ "AN", "CS", "DL", "DD", "DT", "RH", "SU", "YD", "YU" ],
      "description" : "Version: V5_0; AN: Netherlands Antilles; CS: Czechoslovakia; DL: German Democratic Republic; DD: German Democratic Republic; DT: Federal Republic of Germany; RH: Southern Rhodesia; SU: Soviet Union; YD: Democratic Yemen; YU: Yugoslavia/ Serbia and Montenegro"
    } ]
  },
  "st13ApplicationNumber" : {
    "type" : "string",
    "pattern" : "\\d{2}\\d{4}\\d{9}",
    "description" : "Description: Application number format recommended in WIPO Standard ST.13. The sequence of indispensable elements in the application number format is IP type (2 digits), year designation (4 digits) and serial number (9 digits).; Version: V5_0"
  },
  "applicationNumberText" : {
    "type" : "string",
    "description" : "Description: Free format of application number; Version: V5_0"
  }
},
"oneOf" : [ {
  "required" : [ "st13ApplicationNumber" ]
}, {
  "required" : [ "applicationNumberText" ]
} ]
}

```

```
}  
}
```

El enfoque de diseño preferido es dividir los datos en un conjunto de pequeños componentes representados por módulos de esquema, como se ilustra a continuación con un ejemplo de esquema de número de solicitud nuevo. Este esquema JSON está construido a partir de módulos de esquema JSON más pequeños definidos individualmente con sus propios esquemas.

applicationNumber.json (ejemplo de esquema modular)

```
{  
  "$id" : "applicationNumber.json",  
  "$schema" : "https://json-schema.org/draft/2020-12/schema",  
  "type" : "object",  
  "additionalProperties" : false,  
  "properties" : {  
    "applicationNumber" : {  
      "$ref" : "#/$defs/applicationNumber"  
    }  
  },  
  "required" : [ "applicationNumber" ],  
  "$defs" : {  
    "applicationNumber" : {  
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",  
      "description" : "Description: Numbers used by IPOs in order to identify each  
application received; Version: V5_0"  
    }  
  }  
}
```

applicationNumberType.json (continuación del ejemplo de esquema modular)

```
{  
  "$id" : "applicationNumberType.json",  
  "$schema" : "https://json-schema.org/draft/2020-12/schema",  
  "$defs" : {  
    "applicationNumberType" : {  
      "description" : "Version: V5_0",  
      "type" : "object",  
      "additionalProperties" : false,  
      "properties" : {  
        "ipOfficeCode" : {  
          "$ref" : "ipOfficeCode.json#/$defs/ipOfficeCode"  
        },  
        "st13ApplicationNumber" : {  
          "$ref" : "st13ApplicationNumber.json#/$defs/st13ApplicationNumber"  
        },  
        "applicationNumberText" : {  
          "$ref" : "applicationNumberText.json#/$defs/applicationNumberText"  
        }  
      },  
      "oneOf" : [ {  
        "required" : [ "st13ApplicationNumber" ]  
      }, {  
        "required" : [ "applicationNumberText" ]  
      } ]  
    }  
  }  
}
```

ipOfficeCode.json (continuación del ejemplo de esquema modular)

```
{
  "$id" : "ipOfficeCode.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "ipOfficeCode" : {
      "$ref" : "#/$defs/ipOfficeCode"
    }
  },
  "required" : [ "ipOfficeCode" ],
  "$defs" : {
    "ipOfficeCode" : {
      "$ref" : "extendedWIPOST3CodeType.json#/$defs/extendedWIPOST3CodeType",
      "description" : " Description: Two-letter alphabetic codes which represent
the names of states, other entities and intergovernmental organizations the
legislation of which provides for the protection of IP rights or which
organizations are acting in the framework of a treaty in the field of IP;
Version: V5_0"
    }
  }
}
```

extendedWIPOST3CodeType.json (continuación del ejemplo de esquema modular)

```
{
  "$id" : "extendedWIPOST3CodeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "extendedWIPOST3CodeType" : {
      "description" : "Version: V5_0",
      "anyOf" : [ {
        "$ref" : "wipoST3CodeType.json#/$defs/wipoST3CodeType"
      }, {
        "$ref" : "wipoFormerST3CodeType.json#/$defs/wipoFormerST3CodeType"
      } ]
    }
  }
}
```

wipoST3CodeType.json (continuación del ejemplo de esquema modular)

```
{
  "$id" : "wipoST3CodeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "wipoST3CodeType" : {
      "description" : "Description: This code list is inline with WIPO Standard
ST.3 (two-letter codes for the representation of states, other entities and
organizations) published on September, 2019.; Version: V5_0; AD: Andorra; AE:
United Arab Emirates; AF: Afghanistan; AG: Antigua And Barbuda; AI: Anguilla;
AL: Albania; AM: Armenia; AO: Angola; AP: African Regional Intellectual Property
Organization (ARIPO); AR: Argentina; AT: Austria; AU: Australia; AW: Aruba; AZ:
Azerbaijan; BA: Bosnia and Herzegovina; BB: Barbados; BD: Bangladesh; BE:
Belgium; BF: Burkina Faso; BG: Bulgaria; BH: Bahrain; BI: Burundi; BJ: Benin;
BM: Bermuda; BN: Brunei Darussalam; BO: Bolivia (Plurinational State of); BQ:
Bonaire, Sint Eustatius and Saba; BR: Brazil; BS: Bahamas; BT: Bhutan; BV:
Bouvet Island; BW: Botswana; BX: Benelux Office for Intellectual Property
(BOIP); BY: Belarus; BZ: Belize; CA: Canada; CD: Democratic Republic of the
Congo; CF: Central African Republic; CG: Congo; CH: Switzerland; CI: Côte
D'Ivoire; CK: Cook Islands; CL: Chile; CM: Cameroon; CN: China; CO: Colombia;
CR: Costa Rica; CU: Cuba; CV: Cabo Verde; CW: Curaçao; CY: Cyprus; CZ: Czech
Republic; DE: Germany; DJ: Djibouti; DK: Denmark; DM: Dominica; DO: Dominican
Republic; DZ: Algeria; EA: Eurasian Patent Organization (EAPO); EC: Ecuador; EE:
Estonia; EG: Egypt; EH: Western Sahara; EM: European Union Intellectual Property
Office (EUIPO); EP: European Patent Office (EPO); ER: Eritrea; ES: Spain; ET:
Ethiopia; EU: European Union; FI: Finland; FJ: Fiji; FK: Falkland Islands
(Malvinas); FO: Faroe Islands; FR: France; GA: Gabon; GB: United Kingdom; GC:
Patent Office of the Cooperation Council for the Arab States of the Gulf (GCC
Patent Office); GD: Grenada; GE: Georgia; GG: Guernsey; GH: Ghana; GI:
Gibraltar; GL: Greenland; GM: Gambia; GN: Guinea; GQ: Equatorial Guinea; GR:
Greece; GS: South Georgia and South Sandwich Islands; GT: Guatemala; GW: Guinea-
Bissau; GY: Guyana; HK: Hong Kong, China; HN: Honduras; HR: Croatia; HT: Haiti;
HU: Hungary; IB: International Bureau of the World Intellectual Property
Organization (WIPO); ID: Indonesia; IE: Ireland; IL: Israel; IM: Isle of Man;
IN: India; IQ: Iraq; IR: Iran, Islamic Republic of; IS: Iceland; IT: Italy; JE:
Jersey; JM: Jamaica; JO: Jordan; JP: Japan; KE: Kenya; KG: Kyrgyzstan; KH:
Cambodia; KI: Kiribati; KM: Comoros; KN: Saint Kitts and Nevis; KP: Democratic
People's Republic of Korea; KR: Republic of Korea; KW: Kuwait; KY: Cayman
Islands; KZ: Kazakhstan; LA: Lao People's Democratic Republic; LB: Lebanon; LC:
Saint Lucia; LI: Liechtenstein; LK: Sri Lanka; LR: Liberia; LS: Lesotho; LT:
Lithuania; LU: Luxembourg; LV: Latvia; LY: Libya; MA: Morocco; MC: Monaco; MD:
Republic of Moldova; ME: Montenegro; MG: Madagascar; MK: North Macedonia; ML:
Mali; MM: Myanmar; MN: Mongolia; MO: Macao, China; MP: Northern Mariana Islands;
MR: Mauritania; MS: Montserrat; MT: Malta; MU: Mauritius; MV: Maldives; MW:
Malawi; MX: Mexico; MY: Malaysia; MZ: Mozambique; NA: Namibia; NE: Niger; NG:
Nigeria; NI: Nicaragua; NL: Netherlands; NO: Norway; NP: Nepal; NR: Nauru; NZ:
New Zealand; OA: African Intellectual Property Organization (OAPI); OM: Oman;
PA: Panama; PE: Peru; PG: Papua New Guinea; PH: Philippines; PK: Pakistan; PL:
Poland; PT: Portugal; PW: Palau; PY: Paraguay; QA: Qatar; QZ: Community Plant
Variety Office (European Community) (CPVO); RO: Romania; RS: Serbia; RU: Russian
Federation; RW: Rwanda; SA: Saudi Arabia; SB: Solomon Islands; SC: Seychelles;
SD: Sudan; SE: Sweden; SG: Singapore; SH: Saint Helena, Ascension and Tristan da
Cunha; SI: Slovenia; SK: Slovakia; SL: Sierra Leone; SM: San Marino; SN:
Senegal; SO: Somalia; SR: Suriname; SS: South Sudan; ST: Sao Tome and Principe;
SV: El Salvador; SX: Sint Maarten (Dutch part); SY: Syrian Arab Republic; SZ:
Eswatini; TC: Turks and Caicos Islands; TD: Chad; TG: Togo; TH: Thailand; TJ:
Tajikistan; TL: Timor-Leste; TM: Turkmenistan; TN: Tunisia; TO: Tonga; TR:
Turkey; TT: Trinidad and Tobago; TV: Tuvalu; TW: Taiwan, Province of China; TZ:
United Republic of Tanzania; UA: Ukraine; UG: Uganda; US: United States of
America; UY: Uruguay; UZ: Uzbekistan; VA: Holy See; VC: Saint Vincent and the
Grenadines; VE: Venezuela (Bolivian Republic of); VG: British Virgin Islands;
VN: Viet Nam; VU: Vanuatu; WO: World Intellectual Property Organization (WIPO)
(International Bureau of); WS: Samoa; XN: Nordic Patent Institute (NPI); XU:
International Union for the Protection of New Varieties of Plants (UPOV); XV:
```

```

Visegrad Patent Institute (VPI); XX: Unknown states, other entities or
organizations; YE: Yemen; ZA: South Africa; ZM: Zambia; ZW: Zimbabwe",
  "type" : "string",
  "enum" : [ "AD", "AE", "AF", "AG", "AI", "AL", "AM", "AO", "AP", "AR",
"AT", "AU", "AW", "AZ", "BA", "BB", "BD", "BE", "BF", "BG", "BH", "BI", "BJ",
"BM", "BN", "BO", "BQ", "BR", "BS", "BT", "BV", "BW", "BX", "BY", "BZ", "CA",
"CD", "CF", "CG", "CH", "CI", "CK", "CL", "CM", "CN", "CO", "CR", "CU", "CV",
"CW", "CY", "CZ", "DE", "DJ", "DK", "DM", "DO", "DZ", "EA", "EC", "EE", "EG",
"EH", "EM", "EP", "ER", "ES", "ET", "EU", "FI", "FJ", "FK", "FO", "FR", "GA",
"GB", "GC", "GD", "GE", "GG", "GH", "GI", "GL", "GM", "GN", "GQ", "GR", "GS",
"GT", "GW", "GY", "HK", "HN", "HR", "HT", "HU", "IB", "ID", "IE", "IL", "IM",
"IN", "IQ", "IR", "IS", "IT", "JE", "JM", "JO", "JP", "KE", "KG", "KH", "KI",
"KM", "KN", "KP", "KR", "KW", "KY", "KZ", "LA", "LB", "LC", "LI", "LK", "LR",
"LS", "LT", "LU", "LV", "LY", "MA", "MC", "MD", "ME", "MG", "MK", "ML", "MM",
"MN", "MO", "MP", "MR", "MS", "MT", "MU", "MV", "MW", "MX", "MY", "MZ", "NA",
"NE", "NG", "NI", "NL", "NO", "NP", "NR", "NZ", "OA", "OM", "PA", "PE", "PG",
"PH", "PK", "PL", "PT", "PW", "PY", "QA", "QZ", "RO", "RS", "RU", "RW", "SA",
"SB", "SC", "SD", "SE", "SG", "SH", "SI", "SK", "SL", "SM", "SN", "SO", "SR",
"SS", "ST", "SV", "SX", "SY", "SZ", "TC", "TD", "TG", "TH", "TJ", "TL", "TM",
"TN", "TO", "TR", "TT", "TV", "TW", "TZ", "UA", "UG", "US", "UY", "UZ", "VA",
"VC", "VE", "VG", "VN", "VU", "WO", "WS", "XN", "XU", "XV", "XX", "YE", "ZA",
"ZM", "ZW" ]
  }
}
}

```

wipoFormerST3CodeType.json (continuación del ejemplo de esquema modular)

```

{
  "$id" : "wipoFormerST3CodeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "wipoFormerST3CodeType" : {
      "description" : "Version: V5_0; AN: Netherlands Antilles; CS:
Czechoslovakia; DL: German Democratic Republic; DD: German Democratic Republic;
DT: Federal Republic of Germany; RH: Southern Rhodesia; SU: Soviet Union; YD:
Democratic Yemen; YU: Yugoslavia/ Serbia and Montenegro",
      "type" : "string",
      "enum" : [ "AN", "CS", "DL", "DD", "DT", "RH", "SU", "YD", "YU" ]
    }
  }
}

```

st13ApplicationNumber.json (continuación del ejemplo de esquema modular)

```
{
  "$id" : "st13ApplicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "st13ApplicationNumber" : {
      "$ref" : "#/$defs/st13ApplicationNumber"
    }
  },
  "required" : [ "st13ApplicationNumber" ],
  "$defs" : {
    "st13ApplicationNumber" : {
      "$ref" :
"st13ApplicationNumberType.json#/$defs/st13ApplicationNumberType",
      "description" : "Description: Application number format recommended in
WIPO Standard ST.13. The sequence of indispensable elements in the application
number format is IP type (2 digits), year designation (4 digits) and serial
number (9 digits).; Version: V5_0"
    }
  }
}
```

st13ApplicationNumberType.json (continuación del ejemplo de esquema modular)

```
{
  "$id" : "st13ApplicationNumberType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "st13ApplicationNumberType" : {
      "description" : "Version: V5_0",
      "type" : "string",
      "pattern" : "\\d{2}\\d{4}\\d{9}"
    }
  }
}
```

applicationNumberText.json (continuación del ejemplo de esquema modular)

```
{
  "$id" : "applicationNumberText.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "applicationNumberText" : {
      "$ref" : "#/$defs/applicationNumberText"
    }
  },
  "required" : [ "applicationNumberText" ],
  "$defs" : {
    "applicationNumberText" : {
      "type" : "string",
      "description" : "Description: Free format of application number; Version:
V5_0"
    }
  }
}
```

Los esquemas JSON deberían utilizar la palabra clave "\$defs" en la creación de definiciones globales reutilizables para las propiedades y sus contenidos (véase el ejemplo anterior), lo que equivale básicamente a la creación de declaraciones de elemento global y tipos con nombre en un esquema XML.

- [JSD-04] Los esquemas JSON DEBERÍAN utilizar la palabra clave "\$defs" que incluye una definición reutilizable para cada propiedad y tipo de propiedad.
- [JSD-05] DEBEN utilizarse, cuando proceda, los esquemas JSON existentes, definidos en el [Anexo II](#) de este proyecto de norma, antes de crear nuevos esquemas JSON.
- [JSD-06] DEBERÍAN crearse nuevos esquemas JSON solo si se comprueba que no hay esquemas JSON existentes que describan adecuadamente la construcción en cuestión.

6.3 Documentación

Los esquemas JSON deberían ser autodescriptivos. Se debería procurar que los nombres de las construcciones JSON sean significativos. Además, se debería incluir documentación que describa el esquema JSON y las construcciones JSON.

Con el fin de favorecer la reutilización del esquema JSON, se debería mantener su carácter general y no proporcionar documentación con detalles específicos sobre la aplicación del sistema.

- [JSD-07] La documentación NO DEBERÍA contener detalles sobre la aplicación u otra información que no esté directamente relacionada con el significado de la construcción.

El encabezado de un esquema JSON permite que quienes desarrollan esquemas identifiquen fácilmente la finalidad, el uso y los contenidos del esquema. Esa información es muy útil si se ha de elegir un esquema para utilizarlo como modelo en la creación de otro esquema.

- [JSD-08] Los esquemas JSON DEBERÍAN incluir documentación del encabezado del esquema JSON que utilice la palabra clave "description".
- [JSD-09] La sección del encabezado de todos los esquemas JSON DEBERÍA incluir los elementos que figuran en el cuadro 1.

Cuadro 1. Elementos de la documentación del encabezado del esquema JSON.

Nombre del elemento del encabezado	Descripción	Obligatorio/opcional
Description	Descripción, en texto sin formato, de la información incluida en el esquema	Obligatorio, excepto en esquemas JSON creados a partir de XSD de tipo simple de la Norma ST.96 para las que no se proporciona una descripción, como <code>DateType</code> .
Version	Número de versión mayor y menor del esquema	Obligatorio
SchemaCreatedDate	Fecha de creación del esquema	Opcional
SchemaLastModifiedDate	Fecha de la última modificación del esquema	Opcional
SchemaContactPoint	Nombre de la organización de contacto para preguntas sobre el esquema	Opcional
SchemaReleaseNoteURL	URL donde están publicadas las notas explicativas del esquema	Opcional

- [JSD-10] Los elementos de la documentación del encabezado, como "Publicado en" y "Número de versión", DEBERÍAN estar separados por punto y coma, con espacios permitidos después del punto y coma, y conforman el valor asociado a la palabra clave "description". Si no hay un valor disponible para el elemento del encabezado, debería incluirse solo la etiqueta, según se muestra en el siguiente ejemplo:

Ejemplo de documentación del encabezado para applicationBody (esquema de nivel de documento)

```
"description" : "Description: Body of a patent application; Version: V5_0;
SchemaCreateDate: 2012-07-13; SchemaLastModifiedDate: 2021-10-01;
SchemaContactPoint: xml.standards@wipo.int; SchemaReleaseNoteURL:
http://www.wipo.int/standards/XMLSchema/ST96/V5\_0/ReleaseNotes.pdf"
```

Ejemplo de documentación del encabezado para ipOfficeCode (ningún esquema de nivel de documento)

```
"description" : " Description: Two-letter alphabetic codes which represent the
names of states, other entities and intergovernmental organizations the legislation
of which provides for the protection of IP rights or which organizations are acting
in the framework of a treaty in the field of IP; Version: V5_0"
```

Ejemplo de documentación del encabezado de appellateBodyCategoryType.json (esquema de definiciones de tipo de enumeración)

```
"description" : "Version: V5_0; Office appeal board: Appeal board within the IP
office; Court: Court; Appeal Court: Second instance court; Supreme Court: Highest
appellate court"
```

Ejemplo de documentación del encabezado para un esquema de definiciones de tipo

```
"description" : "Version: V5_0"
```

6.4 Nombre de archivo

Para los nombres de archivo del esquema JSON se siguen las normas previstas en la Norma ST.96, pero se debe aplicar la convención LCC.

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 `postalAddressType.json` deriva del nombre de esquema `postalAddressType`. En la presente Norma, las convenciones sobre nombres de archivo de esquema están relacionadas con las normas de las convenciones sobre denominación en JSON.

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

- [JSD-11] Los caracteres utilizados en el nombre de archivo de esquema DEBEN pertenecer al conjunto siguiente: 'a-z, A-Z, 0-9, guion bajo (_), y punto (.)'.
- [JSD-12] El nombre de archivo de esquema DEBE constar de dos partes obligatorias con un delimitador e información facultativa sobre la versión con dos delimitadores adicionales, es decir: `<component name>{"_""V"<major version number>"_""<minor version number>"}.<file extension>`; Por ejemplo, `emailAddressType.json`, `languageCode.json`, `applicationBody_V1_0.json`.
- [JSD-13] Un nombre de archivo de proyecto de esquema DEBE constar de cuatro partes obligatorias con dos delimitadores e información facultativa sobre la versión con dos delimitadores adicionales, es decir: `<component name>{"_""V"<major version number>"_""<minor version number>"_""D"<revision number>"_""<file extension>`, por ejemplo, `trademarkApplication_V1_1_D1.json`. Si un proyecto de esquema se basa en un esquema existente y tiene información sobre la versión en su nombre de archivo, los números de la versión mayor y menor en el nombre de archivo del proyecto de esquema DEBERÍAN ser los mismos indicados en el archivo de esquema en que se basa el proyecto de esquema. Si un proyecto de esquema es nuevo, el número de versión mayor en el nombre de archivo del proyecto de esquema DEBERÍA ser el mismo número indicado en el ámbito correspondiente de la PI, y el número de versión menor en el nombre de archivo del proyecto de esquema DEBERÍA ser cero "0".

6.5 Estructuración de las propiedades del esquema JSON

Los esquemas JSON deberían incluir la propiedad `"type": "object"` para asegurar que se utilizan solo para estructuras anidadas, y no para valores individuales. A continuación figura un fragmento extraído del ejemplo de `applicationNumber.json`:

```
{
  "$id" : "applicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
```

```
"type" : "object",
"additionalProperties" : false,
"properties" : {
  "applicationNumber" : {
    "$ref" : "#/$defs/applicationNumber"
  }
},
"required" : [ "applicationNumber" ],
"$defs" : {
  "applicationNumber" : {
    "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",
    "description" : "Description: Numbers used by IPOs in order to identify each
application received; Version: V5_0"
  }
}
}
```

- [JSD-14] El objeto en el nivel más externo del esquema DEBE tener la palabra clave "type" con el valor "object".
- [JSD-15] El objeto en el nivel más externo del esquema DEBE tener la palabra clave "\$defs" cuyo valor es la propiedad de dicho objeto.
- [JSD-16] El objeto en el nivel más externo del esquema DEBE tener la palabra clave "required" cuyo valor es una matriz de un solo elemento, a saber, la propiedad de dicho objeto.
- Se pueden utilizar extensiones del esquema JSON (personalizaciones). Si el tipo extendido es un tipo de objeto, debe referenciarse en lugar de duplicar sus propiedades para promover la reutilización.
- [JSD-17] Las extensiones del esquema JSON (personalizaciones) para los tipos de objeto DEBEN utilizarse mediante la referencia al esquema JSON del tipo extendido.

7. NORMAS DE DISEÑO DE LAS CONSTRUCCIONES DE ESQUEMA JSON

7.1 Información general

En esta sección se establecen las normas para las construcciones de esquema JSON, concretamente para las matrices, los objetos y los valores primitivos. La normalización de los nombres de las construcciones de esquema es fundamental para el desarrollo de una arquitectura de datos sólida.

7.2 Propiedades

Las propiedades, también conocidas como miembros, son los bloques de construcción básicos de una construcción JSON.

- [JSC-01] Las definiciones DEBERÍAN utilizar los esquemas existentes siempre que sea posible.
- [JSC-02] Si varias propiedades pueden ser lógicamente agrupadas PUEDEN declararse en un único archivo de esquema mediante una definición global.

7.3 Definiciones

Cada propiedad debería tener una definición global en su esquema JSON. De esta manera, el nombre de la propiedad podrá ser reutilizado en varios padres y tendrá una definición uniforme en todos ellos. Véase la propiedad "applicationNumber" en el ejemplo que figura más abajo.

- [JSC-03] Cada propiedad incluida en una palabra clave de propiedades DEBERÍA hacer referencia a una definición global definida dentro de la palabra clave "\$defs". La definición global DEBERÍA tener el mismo nombre que la propiedad.

Ejemplo de una propiedad que hace referencia a una definición global

```
{
  "$id" : "applicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "applicationNumber" : {
      "$ref" : "#/$defs/applicationNumber"
    }
  },
  "required" : [ "applicationNumber" ],
  "$defs" : {
```

```
"applicationNumber" : {  
  "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",  
  "description" : "Description: Numbers used by IPOs in order to identify each  
application received; Version: V5_0"  
}  
}
```

La definición global de una propiedad debería consistir en el nombre del archivo y la descripción de la propiedad.

[JSC-04] La definición global de una propiedad DEBERÍA consistir en el nombre de archivo y la "description" de la propiedad.

Las propiedades deben tener tipos. Pueden ser definidos directamente si son tipos primitivos (excepto el tipo objeto), o especificados mediante una referencia a una definición global de propiedad en otro esquema JSON.

[JSC-05] Una propiedad DEBE tener un tipo que se especifica utilizando la palabra clave "type", ya sea directamente o mediante una referencia a una definición global.

7.4 Definiciones de tipo

Los esquemas JSON pueden incluir definiciones de tipo reutilizables a las que se puede hacer referencia desde definiciones globales de propiedades. Esas definiciones de tipo global deberían consistir en la palabra clave "type", la palabra clave "properties" (si el tipo es "object") y cualquier otra restricción de valor.

[JSC-06] Un esquema PUEDE incluir definiciones de tipo global para reutilizar los modelos de contenido en varias propiedades.

Ejemplo de definición de tipo reutilizable

```
{  
  "$id" : "applicationNumber.json",  
  "$schema" : "https://json-schema.org/draft/2020-12/schema",  
  "type" : "object",  
  "additionalProperties" : false,  
  "properties" : {  
    "applicationNumber" : {  
      "$ref" : "#/$defs/applicationNumber"  
    }  
  },  
  "required" : [ "applicationNumber" ],  
  "$defs" : {  
    "applicationNumber" : {  
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",  
      "description" : "Description: Numbers used by IPOs in order to identify each  
application received; Version: 5_0"  
    }  
  }  
}
```

[JSC-07] Las definiciones que representan tipos DEBEN tener nombres que sigan la convención LCC y tengan el sufijo "Type".

7.5 Tipo primitivo JSON

[JSC-08] Para las propiedades DEBERÍA utilizarse el tipo primitivo JSON más específico y adecuado. Los tipos primitivos incluyen: "string", "number", "integer", "object", "array", "boolean", y "null".

Por ejemplo, si el valor de una propiedad va a ser un número entero, debería utilizarse el tipo "integer" en lugar del tipo más genérico "number" o el tipo más permisivo "string". Para el tipo "string", deberían usarse, si procede, los formatos integrados, por ejemplo, "date-time" o "duration".

7.6 Listas de códigos

En ciertos casos, resulta ventajoso restringir un valor a una lista de códigos que estén normalizados y sean aceptables para el intercambio de datos. Las listas de códigos son un medio para crear un vocabulario controlado con valores permitidos para un elemento de datos (por ejemplo, una lista de códigos normalizados para códigos de país, códigos de idioma, códigos de oficina de PI, etc.). Deberían utilizarse las listas de códigos que ya existen en el dominio público y que son mantenidas por comités de normas, como los comités de la ISO.

- [JSC-09] DEBE utilizarse la Norma ST.3 de la OMPI para hacer referencia a las OPI, los Estados, otras entidades y organizaciones y para el país u organización designado y prioritario.
- [JSC-10] DEBE utilizarse la Norma ISO 3166-1-Alpha-2 Code Elements (two-letter country codes) para referirse a nombres de países con fines de direccionamiento y ciudadanía.
- [JSC-11] DEBE utilizarse la Norma ISO 639-1 (two-Letter Language Codes) para los códigos de idioma.
- [JSC-12] DEBE utilizarse la Norma ISO 4217-Alpha (three-Letter Currency Codes) para los códigos de divisa.
- [JSC-13] DEBERÍA utilizarse la palabra clave JSON `enum` para definir las listas de códigos.
- [JSC-14] Los caracteres utilizados en los valores de enumeración DEBEN limitarse al conjunto siguiente: {a-z, A-Z, 0-9, punto (.), coma (,), espacios, guion (-) y guion bajo (_)}.

7.7 Matrices

El término cardinalidad se define como el número de elementos de una matriz. En un esquema, la cardinalidad se indica mediante las palabras clave `minItems` y `maxItems`. Se recomienda que, al desarrollar un esquema, no se especifiquen valores por defecto para los indicadores de ocurrencia (por ejemplo, `"minItems": 0`), ya que ello podría embrollar innecesariamente el esquema.

- [JSC-15] Los esquemas JSON DEBERÍAN utilizar las palabras clave `minItems` y `maxItems` para las matrices, con la excepción del valor por defecto `"minItems"=0`.

El tipo de elementos de una matriz debe definirse mediante la palabra clave `"items"`. Para simplificar, todos los elementos de una matriz deben tener el mismo tipo. Si se desea una serie de objetos de diferentes tipos, deberían definirse como propiedades separadas de objeto.

- [JSC-16] Para cada objeto de tipo matriz, DEBE haber una palabra clave `"items"` y su valor DEBE ser un objeto de esquema único y no una matriz. Todos los elementos de una matriz DEBEN tener el mismo tipo.

La palabra clave `"additionalItems"` no debe utilizarse para las matrices, ya que no es relevante cuando el valor de `"items"` es un objeto de esquema único.

- [JSC-17] La palabra clave `"additionalItems"` NO DEBERÍA utilizarse cuando el valor de `"items"` es un objeto de esquema único.

7.8 Objetos

7.8.1 *Propiedades comodín*

Los esquemas JSON no deberían permitir que propiedades arbitrarias formen parte de la instancia JSON y esta sea válida, ya que podrían perjudicar la integridad del intercambio de datos.

Se debe utilizar la palabra clave `"additionalProperties"` con el valor `"false"`. De lo contrario, se permitirá el uso de propiedades no definidas en las instancias.

- [JSC-18] Un esquema JSON DEBE utilizar la palabra clave `"additionalProperties"` con el valor `"false"` para cada objeto.

No está permitido utilizar la palabra clave `"patternProperties"`. Esta palabra clave permite asociar expresiones ordinarias a esquemas. Por ejemplo, permite definiciones implícitas de esquemas basadas en el nombre de la propiedad. Para cada tipo de nombre de propiedad, se aplica un esquema particular.

- [JSC-19] Un esquema NO DEBE utilizar la palabra clave `"patternProperties"`.

7.8.2 *Ordenación de las propiedades*

El esquema JSON no impone un orden determinado para las propiedades de un objeto. No obstante, si un esquema JSON se corresponde con un esquema XML, se recomienda que las propiedades aparezcan en el mismo orden que en el esquema XML, tanto en el esquema JSON como en la instancia JSON.

- [JSC-20] El esquema JSON DEBERÍA utilizar la misma ordenación de las propiedades que el esquema XML, si es que se corresponde con alguno.

8 IDENTIFICADORES DE ESQUEMA JSON

8.1 Información general

Un identificador (ID) en un esquema JSON proporciona un identificador uniforme de recursos (URI) que identifica una categoría de información basada en el ámbito empresarial (por ejemplo, empresas, patentes y marcas). En este proyecto de norma se ha optado por los casos en que un único grupo de ID corresponde a varios, tal vez a cientos, de construcciones JSON. Un conjunto de construcciones JSON relacionadas y con nombres únicos será asociado a un ID determinado. Para la identificación debería utilizarse un URI.

[JID-01] Los identificadores DEBEN utilizarse en los esquemas mediante la palabra clave "\$id".

9. NORMAS DE DISEÑO DE INSTANCIAS JSON

El esquema JSON define la estructura y las restricciones de la instancia JSON. Con el fin de mejorar el intercambio de datos (entre oficinas y dentro de ellas) y garantizar su eficacia, deberían asociarse las instancias JSON con el esquema JSON para garantizar su validez y conformidad.

9.1 Ordenación de las propiedades

El esquema JSON no impone un orden determinado para las propiedades de un objeto. No obstante, se recomienda que las propiedades aparezcan en la instancia en el mismo orden que en el esquema JSON.

[JIN-01] Un documento de instancia JSON DEBERÍA utilizar la misma ordenación de las propiedades que el esquema JSON, si es que se corresponde con alguno.

9.2 Validación de instancias JSON

Si una instancia JSON es validada, se garantiza que su contenido satisface todos los requisitos establecidos en el esquema correspondiente.

[JIN-02] Los documentos de instancia JSON PUEDEN ser validados con el esquema correspondiente durante su desarrollo.

[JIN-03] PUEDE crearse un esquema en tiempo de ejecución para satisfacer los requisitos de rendimiento de la solicitud en el entorno de ejecución. Por ejemplo, el esquema compuesto para el número de solicitud puede utilizarse como esquema en tiempo de ejecución.

[Nota: El esquema JSON en tiempo de ejecución es similar al esquema aplanado de la Norma ST.96 que se utiliza en el entorno de producción por motivos de rendimiento, entre otros. Por consiguiente, consideramos que el esquema JSON en tiempo de ejecución podría servir para propósitos similares. No obstante, en el momento de la preparación de esta Norma no existen tales prácticas ni apoyo de herramientas o editores en el sector de JSON.]

[JIN-04] Todas las modificaciones, actualizaciones, revisiones y nuevas ediciones DEBEN someterse primero a la aprobación del Equipo Técnico XML4IP antes de que los cambios puedan incorporarse al esquema en tiempo de ejecución.

Conviene que una instancia JSON se ajuste a un esquema.

[JIN-05] Una instancia JSON DEBERÍA ajustarse a un esquema JSON determinado que satisfaga las normas previstas en la presente Norma.

8. MATERIAL DE REFERENCIA

Normas de la OMPI

- Norma [ST.96](#) de la OMPI: *Recomendación sobre el tratamiento en lenguaje extensible de marcado (XML) de la información relativa a la propiedad intelectual.*
- Norma [ST.90](#) de la OMPI: *Recomendaciones para el tratamiento y la comunicación de datos de propiedad intelectual mediante interfaces de programación de aplicaciones (API) para servicios web.*

Normas y especificaciones del sector

- Especificación JSON: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- JSON Schema Core, 2020-12 draft: <http://json-schema.org/latest/json-schema-core.html>
- JSON Schema Validation, 2020-12 draft: <http://json-schema.org/latest/json-schema-validation.html>
- Especificación OpenAPI v3.1.0: <https://spec.openapis.org/oas/v3.1.0#schema>
- Norma ISO 21778 ECMA-404 – The JSON Data Interchange Syntax: <https://www.ecma-international.org/publications/standards/Ecma-404.htm>
- Universal Business Language Version 2.1: <http://docs.oasis-open.org/ubl/UBL-2.1.html>

- Norma ISO 3166 - Country Codes: <https://www.iso.org/iso-3166-country-codes.html>
- Norma ISO 639 - Language Codes: <https://www.iso.org/iso-639-language-codes.html>
- Norma ISO 4217 - Currency codes: <https://www.iso.org/iso-4217-currency-codes.html>
- Norma ISO 11179: <https://www.iso.org/standard/60341.html>
- RFC 2119: <https://www.ietf.org/rfc/rfc2119.txt>

[Sigue el Anexo I]

ST.XX - ANEXO I

NORMAS SOBRE LA TRANSFORMACIÓN DE LA DEFINICIÓN DE ESQUEMA XML (XSD) DE LA NORMA ST.96 EN ESQUEMA JSON Y DIRECTRICES DE USO

INTRODUCCIÓN

Las Oficinas de Propiedad Intelectual (OPI) y la industria de la PI han venido adoptando y utilizando JSON gradualmente, mientras que siguen utilizando ampliamente XML, conforme a las normas de la OMPI sobre XML. La Norma ST.96 de la OMPI recomienda los recursos de XML que deben utilizarse para la presentación, la publicación, el tratamiento y el intercambio de todo tipo de datos de propiedad intelectual (PI), a saber, patentes, marcas, diseños industriales, indicaciones geográficas y derecho de autor. Las OPI aplican la Norma ST.96 tal y como se haya publicado o la adaptan según sus necesidades.

El intercambio de datos entre las OPI y la difusión de datos por parte de las OPI en los dos formatos, XML y JSON, requiere la coherencia y la compatibilidad de los datos en los dos formatos, lo que puede lograrse utilizando esquemas XML y JSON compatibles, con los que se validarán las instancias XML y JSON, respectivamente.

El Anexo I tiene por objeto proporcionar normas y directrices sobre la transformación de las definiciones de esquema XML (XSD) de la Norma ST.96 en los correspondientes esquemas JSON. Los esquemas JSON proporcionados en el Anexo II de la Norma ST.XX se han obtenido aplicando esas normas de transformación. Es recomendable que las OPI apliquen esas normas cuando desarrollen esquemas JSON, de modo que estos sean compatibles con las XSD de la Norma ST.96 adaptadas a sus necesidades.

Además, el Anexo I incluye, en un apéndice, una herramienta de transformación que aplica esas normas para que las OPI que lo deseen puedan desarrollar sus propios esquemas JSON basados en las XSD de la Norma ST.96 adaptadas a sus necesidades.

En el momento de la elaboración de este Anexo, la última versión del esquema JSON es la draft-2020-12 y la última versión de la Norma ST.96 es la versión 5.0. Por consiguiente, las normas y directrices sobre transformación se basan en la versión 2022-12 del esquema JSON y en la versión 5.0 de la Norma ST.96, que utiliza la versión 1.1 de la XSD. Las normas proporcionadas han de ajustarse a las definidas en el cuerpo principal de la Norma ST.XX. En este Anexo, se utiliza el prefijo 'TR' como identificador de las normas sobre transformación.

ALGORITMO DE TRANSFORMACIÓN Y DIRECTRICES

Para transformar las XSD de la Norma ST.96 en esquemas JSON, debería aplicarse el siguiente algoritmo:

1. Transformación del nombre de archivo de XSD en nombre de archivo de esquema JSON (véanse las normas sobre nombres de archivo en la Sección **Error! Reference source not found.**).
2. Transformación de un fragmento de XSD a un fragmento de esquema JSON mediante:
 - a) XPath absolutos; o
 - b) XPath relativos.

Conviene señalar que la transformación de las XSD en esquemas JSON no requiere seguir un orden estricto. Ahora bien, la validación de los esquemas JSON obtenidos debería iniciarse una vez generados todos ellos, ya que de lo contrario podría ocurrir que uno de los archivos de esquema JSON generados incluya dependencias "`$ref`" aún no generadas.

ALCANCE

Dado que los esquemas JSON que figuran en la presente Norma provienen de la transformación de las XSD de la Norma ST.96, este Anexo solo abarca construcciones, palabras clave y otros elementos de las XSD utilizados en la Norma ST.96 de la OMPI.

TRANSFORMACIÓN DE NOMBRES DE ARCHIVO

[TR-01] Los nombres de archivo de esquema JSON y los nombres de objetos o propiedades de esquema JSON DEBEN coincidir con los correspondientes nombres de archivo o de componentes de la XSD de la Norma ST.96, pero ajustándose a las normas definidas en la Sección **Error! Reference source not found.** "Convenciones de denominación en JSON" y, en el caso de los nombres de archivo, se debe utilizar el sufijo ".json" en lugar de ".xsd".

TRANSFORMACIÓN DE ESPACIOS DE NOMBRES

Los espacios de nombres XSD no pueden utilizarse en los esquemas JSON, ya que la función nativa de los espacios de nombres no es compatible con la versión 2022-12 del esquema JSON. Si en el futuro dicha función fuera compatible con el esquema JSON, la transformación de espacios de nombres XSD sería reconsiderada. No obstante, la emulación

de espacios de nombres solo puede realizarse si los esquemas JSON se almacenan en subcarpetas según el ámbito de la PI. En el diseño de la Norma ST.96, se destina una subcarpeta a cada uno de los ámbitos de la PI.

[TR-02] Para la emulación de espacios de nombres XSD, los esquemas JSON DEBEN almacenarse en subcarpetas según el ámbito de la PI y componentes comunes, a saber, componentes comunes, derecho de autor, diseños, indicaciones geográficas, patentes, marcas y normas externas, tal como se almacenan las correspondientes XSD en la Norma ST.96.

TRANSFORMACIÓN DE TIPOS DE DATOS INTEGRADOS

De acuerdo con la norma JSC-09, para las propiedades DEBERÍA utilizarse el tipo primitivo JSON más específico y adecuado. Por ejemplo, si el valor de una propiedad va a ser un número entero, debería utilizarse el tipo "integer" en lugar del tipo más genérico "number" o el tipo más permisivo "string".

Error! Reference source not found. se proporcionan correspondencias biunívocas entre tipos de datos XSD y tipos de datos JSON (integrados o personalizados / en línea o en un archivo separado) que se utilizan para la transformación de la XSD de la Norma ST.96 en esquema JSON.

[TR-03] Los tipos de datos XSD incluidos en el cuadro 2 DEBERÍAN transformarse en los correspondientes tipos de datos JSON, según se indica en dicho **Error! Reference source not found.**

Cuadro 2. Transformación de tipos simples de datos XSD.

Tipo de datos XSD	Tipo de datos de esquema JSON o definición de tipo de esquema JSON
xsd:string	"type": "string"
Token	"type": "string"
xsd:integer	"type": "integer"
xsd:float xsd:double xsd:decimal	"type": "number"
xsd:Boolean	"type": "boolean"
xsd:positiveInteger	"type": "integer", "minimum": 0, "exclusiveMinimum": true
xsd:negativeInteger	"type": "integer", "maximum": 0, "exclusiveMaximum": true
xsd:nonPositiveInteger	"type": "integer", "maximum": 0, "exclusiveMaximum": false
xsd:nonNegativeInteger	"type": "integer", "minimum": 0, "exclusiveMinimum": false
xsd:date, xsd:dateTime, xsd:time	"type": "string", "format": "date-time"
gYearMonth	No existe ningún tipo de datos integrados. Se define el archivo específico "gYearMonth.json". "gYearMonth": { "anyOf": [{ "type": "object", "properties": { "year": { "type": "integer" }, "month": { "type": "integer", "minimum": 1, "maximum": 12 }, "timezone": { "type": "integer", "minimum": -1440, "maximum": 1439 } } } } }

	<pre>] } </pre>
gYear	No existe ningún tipo de datos integrados. Se define el archivo específico "gYear.json". <pre> "Year": { "anyOf": [{ "type": "object", "properties": { "year": { "type": "integer" }, "timezone": { "type": "integer", "minimum": -1440, "maximum": 1439 } } }] } </pre>
xsd:anyURI	<pre> { "type": "string", "format": "uri" } </pre>

DEFINICIÓN DE ESQUEMA XML (XSD)

Para transformar una XSD, debería añadirse la propiedad "\$schema" de objeto *wrapper* [objeto que envuelve a otro] a la definición del componente correspondiente. Los atributos del elemento raíz de un esquema XML "xsd:schema" no tienen marcadores de posición equivalentes en el esquema JSON.

[TR-04] `xsd:schema` y los valores de sus atributos DEBERÍAN ser transformados en las propiedades JSON correspondientes a partir de las XSD de origen según se indica a continuación.

XSD	JSON	Notas
xsd:schema	"\$schema"	
@xmlns	ignorado	
@targetNamespace	ignorado	
@elementFormDefault	ignorado	
@attributeFormDefault	ignorado	
@version	"description"	Concatenado con el valor de la descripción que lleva la etiqueta "Version: " (véanse JSD-10 y TR-20).

A continuación se proporcionan algunos ejemplos:

Definición de esquema XML de componente (AbstractNumber.xsd)
<pre> <?xml version="1.0" encoding="UTF-8"?> <xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common" elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0"> <xsd:element name="AbstractNumber" type="xsd:string"> <xsd:annotation> <xsd:documentation>Number assigned to an abstract published without the full document in a collection of abstracts. This collection can be a journal, conference proceedings, a patent collection of abstracts (e.g. Soviet Patent Abstracts), etc.</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:schema> </pre>
Definición de esquema JSON de componente (abstractNumber.json)
<pre> { "\$id" : "abstractNumber.json", </pre>

```
"$schema" : "https://json-schema.org/draft/2020-12/schema",
"additionalProperties" : false,
"properties" : {
  "abstractNumber" : {
    "$ref" : "#/$defs/abstractNumber"
  }
},
"required" : [ "abstractNumber" ],
"$defs" : {
  "abstractNumber" : {
    "type" : "string",
    "description" : "Description: Number assigned to an abstract published without
the full document in a collection of abstracts. This collection can be a journal,
conference proceedings, a patent collection of abstracts (e.g. Soviet Patent
Abstracts), etc.; Version: V5_0"
  }
}
}
```

Definición de esquema XML de tipo de componente (AdditionalRemarkType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
<xsd:include schemaLocation="P.xsd"/>
<xsd:include schemaLocation="languageCode.xsd"/>
  <xsd:complexType name="AdditionalRemarkType">
    <xsd:sequence>
      <xsd:element ref="com:P"/>
    </xsd:sequence>
    <xsd:attribute ref="com:languageCode"/>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON de tipo de componente (additionalRemarkType.json)

```
{
  "$id" : "additionalRemarkType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "additionalRemarkType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "languageCode" : {
          "$ref" : "languageCode.json#/$defs/languageCode"
        },
        "p" : {
          "$ref" : "p.json#/$defs/p"
        }
      },
      "required" : [ "p" ]
    }
  }
}
```

Definición de esquema XML de componente de nivel de documento (DesignApplication_V5_0.xsd)

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

```
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Design"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:annotation>
    <xsd:appinfo>
      <com:SchemaCreatedDate>2012-07-13</com:SchemaCreatedDate>
      <com:SchemaLastModifiedDate>2021-10-01</com:SchemaLastModifiedDate>
      <com:SchemaContactPoint>xml.standards@wipo.int</com:SchemaContactPoint>

      <com:SchemaReleaseNoteURL>http://www.wipo.int/standards/XMLSchema/ST96/V5_0/Release
Notes.pdf</com:SchemaReleaseNoteURL>
    </xsd:appinfo>
  </xsd:annotation>
<xsd:include schemaLocation="DesignApplicationType_V5_0.xsd"/>
  <xsd:element name="DesignApplication" type="dgn:DesignApplicationType">
    <xsd:annotation>
      <xsd:documentation>Details on a design application</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

Definición de esquema JSON de componente de nivel de documento (designApplication_V5_0.json)

```
{
  "$id" : "designApplication_V5_0.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "designApplication" : {
      "$ref" : "#/$defs/designApplication"
    }
  },
  "required" : [ "designApplication" ],
  "$defs" : {
    "designApplication" : {
      "$ref" : "designApplicationType_V5_0.json#/$defs/designApplicationType",
      "description" : "Description: Details on a design application; Version: V5_0;
SchemaCreatedDate: 2012-07-13; SchemaLastModifiedDate: 2021-10-01;
SchemaContactPoint: xml.standards@wipo.int; SchemaReleaseNoteURL:
http://www.wipo.int/standards/XMLSchema/ST96/V5_0/ReleaseNotes.pdf"
    }
  }
}
```

Definición de esquema XML de tipo de componente de nivel de documento (DesignApplicationType_V5_0.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:dgn="http://www.wipo.int/standards/XMLSchema/ST96/Design"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Design"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/RequestSoftware.xsd"/>
  <xsd:include schemaLocation="../../DesignApplicationFormName.xsd"/>
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/RequestExamination.xsd"/>
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/RegistrationOfficeCode.xsd"/>
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/ReceivingOfficeCode.xsd"/>
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../Common/ReceivingOfficeDate.xsd"/>
  <xsd:include schemaLocation="../../ReceiptNumber.xsd"/>
  <xsd:include schemaLocation="../../SealedDepositIndicator.xsd"/>
```

```

<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ApplicationNumber.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/FilingPlace.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ApplicantFileReference.xsd"/>
<xsd:include schemaLocation="../../DesignApplicationLanguageCode.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/SecondLanguageCode.xsd"/>
<xsd:include schemaLocation="../../DesignTotalQuantity.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/CorrespondenceLanguageCode.xsd"/>
<xsd:include schemaLocation="../../DesignApplicationCurrentStatusCategory.xsd"/>
<xsd:include schemaLocation="../../DesignApplicationCurrentStatusDate.xsd"/>
<xsd:include schemaLocation="../../DesignatedCountryBag.xsd"/>
<xsd:include schemaLocation="../../DesignBag.xsd"/>
<xsd:include schemaLocation="../../ApplicantBag.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/RepresentativeBag.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/Authorization.xsd"/>
<xsd:include schemaLocation="../../DesignApplicationEventBag.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/DocumentIncludedBag.xsd"/>
<xsd:include schemaLocation="../../DesignApplicationStatementBag.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/PaymentBag.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ReimbursementBag.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/SignatureBag.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ApplicationDate.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ApplicationDateTime.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/CorrespondenceAddress.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/BusinessEntityStatusCategory.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/InternationalRegistrationNumber.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/operationCategory.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/st96Version.xsd"/>
<xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/ipoVersion.xsd"/>
<xsd:complexType name="DesignApplicationType">
  <xsd:sequence>
    <xsd:element ref="com:RequestSoftware" minOccurs="0"/>
    <xsd:element ref="dgn:DesignApplicationFormName" minOccurs="0"/>
    <xsd:element ref="com:RequestExamination" minOccurs="0"/>
    <xsd:element ref="com:RegistrationOfficeCode"/>
    <xsd:element ref="com:ReceivingOfficeCode" minOccurs="0"/>
    <xsd:element ref="com:ReceivingOfficeDate" minOccurs="0"/>
    <xsd:element ref="dgn:ReceiptNumber" minOccurs="0"/>
    <xsd:element ref="dgn:SealedDepositIndicator" minOccurs="0"/>
    <xsd:element ref="com:ApplicationNumber" minOccurs="0"/>
    <xsd:element ref="com:InternationalRegistrationNumber" minOccurs="0"/>
    <xsd:element ref="com:FilingPlace" minOccurs="0"/>
    <xsd:element ref="com:ApplicantFileReference" minOccurs="0"/>
    <xsd:element ref="dgn:DesignApplicationLanguageCode" minOccurs="0"/>
    <xsd:element ref="com:SecondLanguageCode" minOccurs="0"/>
    <xsd:element ref="dgn:DesignTotalQuantity" minOccurs="0"/>
    <xsd:element ref="com:CorrespondenceLanguageCode" minOccurs="0"/>
  
```

```

        <xsd:element ref="dgn:DesignApplicationCurrentStatusCategory"
minOccurs="0"/>
        <xsd:element ref="dgn:DesignApplicationCurrentStatusDate" minOccurs="0"/>
        <xsd:element ref="dgn:DesignatedCountryBag" minOccurs="0"/>
        <xsd:element ref="dgn:DesignBag"/>
        <xsd:element ref="dgn:ApplicantBag"/>
        <xsd:element ref="com:RepresentativeBag" minOccurs="0"/>
        <xsd:element ref="com:Authorization" minOccurs="0"/>
        <xsd:element ref="dgn:DesignApplicationEventBag" minOccurs="0"/>
        <xsd:element ref="com:DocumentIncludedBag" minOccurs="0"/>
        <xsd:element ref="dgn:DesignApplicationStatementBag" minOccurs="0"/>
        <xsd:element ref="com:PaymentBag" minOccurs="0"/>
        <xsd:element ref="com:ReimbursementBag" minOccurs="0"/>
        <xsd:element ref="com:SignatureBag" minOccurs="0"/>
        <xsd:choice>
            <xsd:element ref="com:ApplicationDate" minOccurs="0"/>
            <xsd:element ref="com:ApplicationDateTime"/>
        </xsd:choice>
        <xsd:element ref="com:CorrespondenceAddress" minOccurs="0"/>
        <xsd:element ref="com:BusinessEntityStatusCategory" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute ref="com:operationCategory"/>
    <xsd:attribute ref="com:st96Version" use="required"/>
    <xsd:attribute ref="com:ipoVersion"/>
</xsd:complexType>
</xsd:schema>

```

Definición de esquema JSON de tipo de componente de nivel de documento (designApplicationType_V5_0.json)

```

{
  "$id" : "designApplicationType_V5_0.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "designApplicationType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "operationCategory" : {
          "$ref" : "../../../Common/operationCategory.json#/$defs/operationCategory"
        },
        "st96Version" : {
          "$ref" : "../../../Common/st96Version.json#/$defs/st96Version"
        },
        "ipoVersion" : {
          "$ref" : "../../../Common/ipoVersion.json#/$defs/ipoVersion"
        },
        "requestSoftware" : {
          "$ref" : "../../../Common/requestSoftware.json#/$defs/requestSoftware"
        },
        "designApplicationFormName" : {
          "$ref" :
"../../designApplicationFormName.json#/$defs/designApplicationFormName"
        },
        "requestExamination" : {
          "$ref" : "../../../Common/requestExamination.json#/$defs/requestExamination"
        },
        "registrationOfficeCode" : {
          "$ref" :
"../../Common/registrationOfficeCode.json#/$defs/registrationOfficeCode"
        },
        "receivingOfficeCode" : {
          "$ref" : "../../../Common/receivingOfficeCode.json#/$defs/receivingOfficeCode"
        },
        "receivingOfficeDate" : {

```

```
"$ref" : "../../../Common/receivingOfficeDate.json#/$defs/receivingOfficeDate"
},
"receiptNumber" : {
  "$ref" : "../../../receiptNumber.json#/$defs/receiptNumber"
},
"sealedDepositIndicator" : {
  "$ref" : "../../../sealedDepositIndicator.json#/$defs/sealedDepositIndicator"
},
"applicationNumber" : {
  "$ref" : "../../../Common/applicationNumber.json#/$defs/applicationNumber"
},
"internationalRegistrationNumber" : {
  "$ref" :
"../../../Common/internationalRegistrationNumber.json#/$defs/internationalRegistrationNum
ber"
},
"filingPlace" : {
  "$ref" : "../../../Common/filingPlace.json#/$defs/filingPlace"
},
"applicantFileReference" : {
  "$ref" :
"../../../Common/applicantFileReference.json#/$defs/applicantFileReference"
},
"designApplicationLanguageCode" : {
  "$ref" :
"../../../designApplicationLanguageCode.json#/$defs/designApplicationLanguageCode"
},
"secondLanguageCode" : {
  "$ref" : "../../../Common/secondLanguageCode.json#/$defs/secondLanguageCode"
},
"designTotalQuantity" : {
  "$ref" : "../../../designTotalQuantity.json#/$defs/designTotalQuantity"
},
"correspondenceLanguageCode" : {
  "$ref" :
"../../../Common/correspondenceLanguageCode.json#/$defs/correspondenceLanguageCode"
},
"designApplicationCurrentStatusCategory" : {
  "$ref" :
"../../../designApplicationCurrentStatusCategory.json#/$defs/designApplicationCurrentStatusC
ategory"
},
"designApplicationCurrentStatusDate" : {
  "$ref" :
"../../../designApplicationCurrentStatusDate.json#/$defs/designApplicationCurrentStatusDate"
},
"designatedCountryBag" : {
  "$ref" : "../../../designatedCountryBag.json#/$defs/designatedCountryBag"
},
"designBag" : {
  "$ref" : "../../../designBag.json#/$defs/designBag"
},
"applicantBag" : {
  "$ref" : "../../../applicantBag.json#/$defs/applicantBag"
},
"representativeBag" : {
  "$ref" : "../../../Common/representativeBag.json#/$defs/representativeBag"
},
"authorization" : {
  "$ref" : "../../../Common/authorization.json#/$defs/authorization"
},
"designApplicationEventBag" : {
  "$ref" :
"../../../designApplicationEventBag.json#/$defs/designApplicationEventBag"
},
}
```

```

    "documentIncludedBag" : {
      "$ref" : "../../../Common/documentIncludedBag.json#/$defs/documentIncludedBag"
    },
    "designApplicationStatementBag" : {
      "$ref" :
"../../designApplicationStatementBag.json#/$defs/designApplicationStatementBag"
    },
    "paymentBag" : {
      "$ref" : "../../../Common/paymentBag.json#/$defs/paymentBag"
    },
    "reimbursementBag" : {
      "$ref" : "../../../Common/reimbursementBag.json#/$defs/reimbursementBag"
    },
    "signatureBag" : {
      "$ref" : "../../../Common/signatureBag.json#/$defs/signatureBag"
    },
    "applicationDate" : {
      "$ref" : "../../../Common/applicationDate.json#/$defs/applicationDate"
    },
    "applicationDateTime" : {
      "$ref" : "../../../Common/applicationDateTime.json#/$defs/applicationDateTime"
    },
    "correspondenceAddress" : {
      "$ref" :
"../../Common/correspondenceAddress.json#/$defs/correspondenceAddress"
    },
    "businessEntityStatusCategory" : {
      "$ref" :
"../../Common/businessEntityStatusCategory.json#/$defs/businessEntityStatusCategory"
    }
  },
  "oneOf" : [ {
    "required" : [ "applicationDate" ]
  }, {
    "required" : [ "applicationDateTime" ]
  } ],
  "required" : [ "st96Version", "registrationOfficeCode", "designBag",
"applicantBag" ]
}
}
}

```

REFERENCIACIÓN DE ESQUEMAS GLOBALES

[TR-05] Lo dos tipos convencionales de definiciones globales utilizados en las XSD, `xsd:import` y `xsd:include`, DEBEN ser transformados con la propiedad "\$ref" en esquema JSON independientemente de si en la XSD se utiliza el tipo "import" o "include".

Por ejemplo:

Definición de esquema XML para un ejemplo de `xsd:import` (RelatedApplicationDate.xsd)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:dgn="http://www.wipo.int/standards/XMLSchema/ST96/Design"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Design"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:import namespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
schemaLocation="../../../Common/DateType.xsd"/>
  <xsd:element name="RelatedApplicationDate" type="com:DateType">
    <xsd:annotation>
      <xsd:documentation>Application date of the related
application</xsd:documentation>
    </xsd:annotation>
  </xsd:element>

```

```
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:import (relatedApplicationDate.json)

```
{
  "$id" : "relatedApplicationDate.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "relatedApplicationDate" : {
      "$ref" : "#/$defs/relatedApplicationDate"
    }
  },
  "required" : [ "relatedApplicationDate" ],
  "$defs" : {
    "relatedApplicationDate" : {
      "$ref" : "../Common/dateType.json#/$defs/dateType",
      "description" : "Description: Application date of the related application;
Version: V5_0"
    }
  }
}
```

Definición de esquema XML para un ejemplo de xsd:include (AffectedDesign.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:dgn="http://www.wipo.int/standards/XMLSchema/ST96/Design"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Design"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:include schemaLocation="AffectedDesignType.xsd"/>
  <xsd:element name="AffectedDesign" type="dgn:AffectedDesignType">
    <xsd:annotation>
      <xsd:documentation>Design affected by the decision, either all designs or
enumeration</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:include (affectedDesign.json)

```
{
  "$id" : "affectedDesign.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "affectedDesign" : {
      "$ref" : "#/$defs/affectedDesign"
    }
  },
  "required" : [ "affectedDesign" ],
  "$defs" : {
    "affectedDesign" : {
      "$ref" : "affectedDesignType.json#/$defs/affectedDesignType",
      "description" : "Description: Design affected by the decision, either all
designs or enumeration; Version: V5_0"
    }
  }
}
```

INSTRUCCIONES DE COMPOSICIÓN XSD

Las instrucciones de composición son construcciones de esquema del W3C que agrupan declaraciones de elementos. En el esquema del W3C, hay tres tipos de operadores: `sequence`, `choice` y `all`. No está permitido utilizar `xsd:all` en la Norma ST.96 de la OMPI de acuerdo con la norma de diseño [SD-52] prevista en el Anexo I de la Norma ST.96.

[TR-06] DEBERÍA añadirse una propiedad de objeto o matriz a la definición del componente correspondiente de acuerdo con el valor de "maxOccurs", según se indica a continuación.

XSD	JSON	Notas
<code>xsd:sequence</code>	objeto	
<code>maxOccurs=1</code>	objeto	<code>minOccurs=1</code> se requiere que aparezca un elemento
<code>maxOccurs=unbounded</code>	objeto o matriz	<code>minOccurs=1</code> se requiere que aparezca un elemento
<code>xsd:choice</code>	objeto	
<code>maxOccurs=1</code>	objeto	<code>minOccurs=1</code> se requiere que aparezca un elemento
<code>maxOccurs=unbounded</code>	objeto o matriz	<code>minOccurs=1</code> se requiere que aparezca un elemento

Por ejemplo:

Definición de esquema XML para un ejemplo de `xsd:sequence maxOccurs=1` (AdditionalRemarkType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:include schemaLocation="P.xsd"/>
  <xsd:include schemaLocation="languageCode.xsd"/>
  <xsd:complexType name="AdditionalRemarkType">
    <xsd:sequence>
      <xsd:element ref="com:P"/>
    </xsd:sequence>
    <xsd:attribute ref="com:languageCode"/>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de `xsd:sequence maxOccurs=1` (AdditionalRemarkType.json)

```
{
  "$id" : "additionalRemarkType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "additionalRemarkType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "languageCode" : {
          "$ref" : "languageCode.json#/$defs/languageCode"
        },
        "p" : {
          "$ref" : "p.json#/$defs/p"
        }
      },
      "required" : [ "p" ]
    }
  }
}
```

Definición de esquema XML para un ejemplo de `xsd:sequence maxOccurs=unbounded` (InventionClaimBagType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
<xsd:include schemaLocation="InventionNumber.xsd"/>
<xsd:include schemaLocation="ClaimNumber.xsd"/>
<xsd:include schemaLocation="ClaimNumberRange.xsd"/>
  <xsd:complexType name="InventionClaimBagType">
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element ref="pat:InventionNumber"/>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element ref="pat:ClaimNumber"/>
        <xsd:element ref="pat:ClaimNumberRange"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:sequence maxOccurs=unbounded (inventionClaimBagType.json)

```
{
  "$id" : "inventionClaimBagType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "inventionClaimBagType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "inventionNumber" : {
          "type" : "array",
          "minItems" : 1,
          "items" : {
            "$ref" : "inventionNumber.json#/$defs/inventionNumber"
          }
        },
        "claimNumber" : {
          "anyOf" : [ {
            "$ref" : "claimNumber.json#/$defs/claimNumber"
          }, {
            "type" : "array",
            "minItems" : 1,
            "items" : {
              "$ref" : "claimNumber.json#/$defs/claimNumber"
            }
          } ]
        },
        "claimNumberRange" : {
          "anyOf" : [ {
            "$ref" : "claimNumberRange.json#/$defs/claimNumberRange"
          }, {
            "type" : "array",
            "minItems" : 1,
            "items" : {
              "$ref" : "claimNumberRange.json#/$defs/claimNumberRange"
            }
          } ]
        }
      }
    },
    "anyOf" : [ {
      "required" : [ "claimNumber" ]
    }, {
      "required" : [ "claimNumberRange" ]
    } ],
  }
}
```

```
"required" : [ "inventionNumber" ]
    }
  }
}
```

Definición de esquema XML para un ejemplo de xsd:choice maxOccurs=1 (ChemicalFormulaeType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
<xsd:include schemaLocation="Image.xsd"/>
<xsd:include schemaLocation="InlineFormula.xsd"/>
<xsd:include schemaLocation="id.xsd"/>
<xsd:include schemaLocation="chemicalFormulaeNumber.xsd"/>
<xsd:include schemaLocation="ExternalDocumentBag.xsd"/>
  <xsd:complexType name="ChemicalFormulaeType">
    <xsd:choice>
      <xsd:element ref="com:Image"/>
      <xsd:element ref="com:InlineFormula"/>
      <xsd:element ref="com:ExternalDocumentBag"/>
    </xsd:choice>
    <xsd:attribute ref="com:id"/>
    <xsd:attribute ref="com:chemicalFormulaeNumber"/>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:choice maxOccurs=1 (chemicalFormulaeType.json)

```
{
  "$id" : "chemicalFormulaeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "chemicalFormulaeType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "id" : {
          "$ref" : "id.json#/$defs/id"
        },
        "chemicalFormulaeNumber" : {
          "$ref" : "chemicalFormulaeNumber.json#/$defs/chemicalFormulaeNumber"
        },
        "image" : {
          "$ref" : "image.json#/$defs/image"
        },
        "inlineFormula" : {
          "$ref" : "inlineFormula.json#/$defs/inlineFormula"
        },
        "externalDocumentBag" : {
          "$ref" : "externalDocumentBag.json#/$defs/externalDocumentBag"
        }
      }
    },
    "oneOf" : [ {
      "required" : [ "image" ]
    }, {
      "required" : [ "inlineFormula" ]
    }, {
      "required" : [ "externalDocumentBag" ]
    } ]
  }
}
```

```
}
}
```

Definición de esquema XML para un ejemplo de xsd:choice maxOccurs=unbounded (InventionClaimBagType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:include schemaLocation="Heading.xsd"/>
  <xsd:include schemaLocation="P.xsd"/>
  <xsd:include schemaLocation="id.xsd"/>
  <xsd:complexType name="ContentType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="com:Heading"/>
      <xsd:element ref="com:P"/>
    </xsd:choice>
    <xsd:attribute ref="com:id"/>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:choice maxOccurs=unbounded (inventionClaimBagType.json)

```
{
  "$id" : "contentType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "contentType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "id" : {
          "$ref" : "id.json#/$defs/id"
        },
        "heading" : {
          "anyOf" : [ {
            "$ref" : "heading.json#/$defs/heading"
          }, {
            "type" : "array",
            "minItems" : 1,
            "items" : {
              "$ref" : "heading.json#/$defs/heading"
            }
          } ]
        },
        "p" : {
          "anyOf" : [ {
            "$ref" : "p.json#/$defs/p"
          }, {
            "type" : "array",
            "minItems" : 1,
            "items" : {
              "$ref" : "p.json#/$defs/p"
            }
          } ]
        }
      }
    },
    "anyOf" : [ {
      "required" : [ "heading" ]
    }, {
      "required" : [ "p" ]
    } ]
  }
}
```

```

    }
  }
}

```

ELEMENTOS

Los elementos son las piezas básicas de los documentos XML y deben transformarse en una propiedad en los documentos JSON.

[TR-07] DEBE añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente de acuerdo con el valor de `maxOccurs`, según se indica a continuación.

XSD	JSON	Notas
<code>xsd:element</code>	Propiedad	Véase la sección "Transformación de tipos de datos integrados"
<code>maxOccurs=1</code>	Propiedad con tipo primitivo o tipo personalizado de datos XSD	<code>minOccurs=1</code> se requiere que aparezca una propiedad
<code>maxOccurs=unbounded</code>	Matriz de tipo primitivo o de tipo personalizado	<code>minOccurs=1</code> se requiere que aparezca una propiedad

Por ejemplo:

Definición de esquema XML para un ejemplo de <code>xsd:element maxOccurs=1 (DocumentTotalQuantity.xsd)</code>
<pre> <?xml version="1.0" encoding="UTF-8"?> <xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common" elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0"> <xsd:element name="DocumentTotalQuantity" type="xsd:nonNegativeInteger"> <xsd:annotation> <xsd:documentation>Total number of documents available or provided. </xsd:documentation> </xsd:annotation> </xsd:element> </xsd:schema> </pre>

Definición de esquema JSON para un ejemplo de <code>xsd:element maxOccurs=1 (documentTotalQuantity.json)</code>
<pre> { "\$id" : "documentTotalQuantity.json", "\$schema" : "https://json-schema.org/draft/2020-12/schema", "additionalProperties" : false, "properties" : { "documentTotalQuantity" : { "\$ref" : "#/\$defs/documentTotalQuantity" } }, "required" : ["documentTotalQuantity"], "\$defs" : { "documentTotalQuantity" : { "type" : "integer", "minimum" : 0, "description" : "Description: Total number of documents available or provided. ; Version: V5_0" } } } </pre>

Definición de esquema XML para un ejemplo de <code>xsd:element maxOccurs=unbounded (IPOfficeCodeBagType.xsd)</code>
<pre> <?xml version="1.0" encoding="UTF-8"?> </pre>

```
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
<xsd:include schemaLocation="IPOfficeCode.xsd"/>
  <xsd:complexType name="IPOfficeCodeBagType">
    <xsd:sequence>
      <xsd:element ref="com:IPOfficeCode" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:element maxOccurs=unbounded (ipOfficeCodeBagType.json)

```
{
  "$id" : "ipOfficeCodeBagType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "ipOfficeCodeBagType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "ipOfficeCode" : {
          "type" : "array",
          "minItems" : 1,
          "items" : {
            "$ref" : "ipOfficeCode.json#/$defs/ipOfficeCode"
          }
        }
      },
      "required" : [ "ipOfficeCode" ]
    }
  }
}
```

ATRIBUTOS

Los atributos son construcciones de esquema del W3C asociadas a elementos que proporcionan más información sobre elementos y deben ser transformados como una propiedad del componente correspondiente en JSON.

[TR-08] DEBE añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente, según se indica a continuación.

XSD	JSON	Notas
xsd:attribute	Propiedad del componente correspondiente	Véase la sección “Transformación de tipos de datos integrados”
xsd:annotation	“\$description”	Véase la sección “Anotación”

Por ejemplo:

Definición de esquema XML para un ejemplo de xsd:attribute (changeDateTime.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
<xsd:attribute name="changeDateTime" type="xsd:dateTime">
  <xsd:annotation>
    <xsd:documentation>Date and time of change</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:attribute (changeDateTime.json)

```
{
  "$id" : "changeDateTime.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "additionalProperties" : false,
  "properties" : {
    "changeDateTime" : {
      "$ref" : "#/$defs/changeDateTime"
    }
  },
  "required" : [ "changeDateTime" ],
  "$defs" : {
    "changeDateTime" : {
      "format" : "date-time",
      "type" : "string",
      "description" : "Description: Date and time of change; Version: V5_0"
    }
  }
}
```

SIMPLETYPE

[TR-09] DEBERÍA añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente, según se indica a continuación.

XSD	JSON	Notas
xsd:simpleType	objeto	Véase la sección “Transformación de tipos de datos integrados”
xsd:union	objeto - “anyOf”	Véase la sección “Unión”
xsd:restriction	objeto	Véase la sección “Restricción”
xsd:enumeration	enumeración	Véase la sección “Enumeración”

COMPLEXTYPE

[TR-10] DEBERÍA añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente, según se indica a continuación.

XSD	JSON	Notas
Xsd:complexType	objeto	Véase la sección “Transformación de tipos de datos integrados”
xsd:simpleContent	objeto	Véase la sección “Contenido simple”
xsd:complexContent	objeto	Véase la sección “Contenido complejo”
xsd:sequence	objeto	Véase la sección “Instrucciones de composición XSD”
xsd:choice	objeto o matriz	Véase la sección “Instrucciones de composición XSD”
@mixed	objeto	Véase la sección “Contenido mixto”

Contenido simple

[TR-11] DEBERÍA añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente, según se indica a continuación.

XSD	JSON	Notas
xsd:simpleContent	objeto	
xsd:extension	objeto	Véase la sección “Ampliación”
xsd:restriction	objeto	Véase la sección “Restricción”

Contenido complejo

[TR-12] DEBERÍA añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente, según se indica a continuación.

XSD	JSON	Notas
xsd:complexContent	objeto	
xsd:extension	objeto	Véase la sección “Ampliación”
xsd:restriction	objeto	Véase la sección “Restricción”

Contenido mixto

[TR-13] DEBERÍA añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente, según se indica a continuación.

XSD	JSON	Notas
xsd:complexType	objeto	
@mixed	objeto	

Por ejemplo:

Definición de esquema XML para un ejemplo de xsd:complexType@mixed=true (CrossReferenceType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
<xsd:include schemaLocation="PhraseType.xsd"/>
<xsd:include schemaLocation="id.xsd"/>
<xsd:include schemaLocation="idrefs.xsd"/>
<xsd:include schemaLocation="extRef.xsd"/>
<xsd:include schemaLocation="crossReferenceCategory.xsd"/>
<xsd:include schemaLocation="sourceURI.xsd"/>
<xsd:include schemaLocation="sourceSystemName.xsd"/>
<xsd:include schemaLocation="sourceSystemIdentifier.xsd"/>
  <xsd:complexType name="CrossReferenceType" mixed="true">
    <xsd:complexContent>
      <xsd:extension base="com:PhraseType">
        <xsd:attribute ref="com:id"/>
        <xsd:attribute ref="com:idrefs"/>
        <xsd:attribute ref="com:extRef"/>
        <xsd:attribute ref="com:crossReferenceCategory" use="required"/>
        <xsd:attribute ref="com:sourceURI"/>
        <xsd:attribute ref="com:sourceSystemName"/>
        <xsd:attribute ref="com:sourceSystemIdentifier"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:complexType@mixed=true (crossReferenceType.xsd)

```
{
  "$id" : "crossReferenceType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "crossReferenceType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "phraseType" : {
          "$ref" : "phraseType.json#/$defs/phraseType"
        },
        "id" : {
          "$ref" : "id.json#/$defs/id"
        },
        "idrefs" : {
          "$ref" : "idrefs.json#/$defs/idrefs"
        },
        "extRef" : {
          "$ref" : "extRef.json#/$defs/extRef"
        },
        "crossReferenceCategory" : {
          "$ref" : "crossReferenceCategory.json#/$defs/crossReferenceCategory"
        },
        "sourceURI" : {
          "$ref" : "sourceURI.json#/$defs/sourceURI"
        },
        "sourceSystemName" : {
          "$ref" : "sourceSystemName.json#/$defs/sourceSystemName"
        },
        "sourceSystemIdentifier" : {
```

```

        "$ref" : "sourceSystemIdentifier.json#/$defs/sourceSystemIdentifier"
      },
      "required" : [ "crossReferenceCategory" ]
    }
  }
}

```

ANOTACIÓN

En la Norma ST.96 de la OMPI se utilizan dos tipos de anotaciones, a saber, `xsd:appinfo` y `xsd:documentation`.

`xsd:appinfo`

El elemento `xsd:appinfo` especifica la información que se utilizará en la solicitud y debe ir dentro de un elemento de anotación. En la Norma ST.96 se utiliza `xsd:appinfo` en todas las XSD de nivel de documento que contienen los siguientes elementos de esquema:

- `com:SchemaCreatedDate`
- `com:SchemaLastModifiedDate`
- `com:SchemaContactPoint`
- `com:SchemaReleaseNoteURL`

[TR-14] La información proporcionada en `<xsd:appinfo>` DEBERÍA especificarse en el valor de "description" del esquema JSON.

Por ejemplo:

Definición de esquema XML de componente (ApplicationBodyType.xsd)

```

<xsd:annotation>
  <xsd:appinfo>
    <com:SchemaCreatedDate>2012-07-13</com:SchemaCreatedDate>
    <com:SchemaLastModifiedDate>2021-10-01</com:SchemaLastModifiedDate>

    <com:SchemaContactPoint>xml.standards@wipo.int</com:SchemaContactPoint>

    <com:SchemaReleaseNoteURL>http://www.wipo.int/standards/XMLSchema/ST96/V5_0/ReleaseNotes.pdf</com:SchemaReleaseNoteURL>
  </xsd:appinfo>
</xsd:annotation>
<xsd:include schemaLocation="ApplicationBodyType_V5_0.xsd"/>
<xsd:element name="ApplicationBody" type="pat:ApplicationBodyType">
  <xsd:annotation>
    <xsd:documentation>Body of a patent application</xsd:documentation>
  </xsd:annotation>...

```

Definición de esquema JSON para un ejemplo de `xsd:annotation` (applicationBodyType.json)

```

"$defs" : {
  applicationBody" : {
    $ref" : "applicationBodyType_V5_0.json#/$defs/applicationBodyType",
    "description" : "Description: Body of a patent application; Version:
V5_0; SchemaCreatedDate: 2012-07-13; SchemaLastModifiedDate: 2021-10-01;
SchemaContactPoint: xml.standards@wipo.int;
SchemaReleaseNoteURL: http://www.wipo.int/standards/XMLSchema/ST96/V5\_0/ReleaseNotes.pdf"
  }
}

```

xsd:documentation

En la Norma ST.96, el elemento `xsd:documentation` especifica dentro de un elemento de anotación la información que debe ser leída o utilizada por los usuarios. De acuerdo con la Norma ST.96:

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

[TR-15] La información sobre la documentación y la versión DEBERÍA transferirse al valor de "description" del esquema JSON.

Por ejemplo:

Definición de esquema XML de componente (AbstractNumber.xsd)
<pre><xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common" elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0"> <xsd:element name="AbstractNumber" type="xsd:string"> <xsd:annotation> <xsd:documentation>Number assigned to an abstract published without the full document in a collection of abstracts. This collection can be a journal, conference proceedings, a patent collection of abstracts (e.g. Soviet Patent Abstracts), etc.</xsd:documentation> </xsd:annotation> </xsd:element> </xsd:schema></pre>

Definición de esquema JSON para un ejemplo de <code>xsd:documentation</code> (abstractNumber.json):
<pre>{ "\$id" : "abstractNumber.json", "\$schema" : "https://json-schema.org/draft/2020-12/schema", "additionalProperties" : false, "properties" : { "abstractNumber" : { "\$ref" : "#/\$defs/abstractNumber" } }, "required" : ["abstractNumber"], "\$defs" : { "abstractNumber" : { "type" : "string", "description" : "Description: Number assigned to an abstract published without the full document in a collection of abstracts. This collection can be a journal, conference proceedings, a patent collection of abstracts (e.g. Soviet Patent Abstracts), etc.; Version: V5_0" } } }</pre>

UNIÓN

[TR-16] DEBERÍA añadirse una propiedad con el tipo adecuado a la definición del componente correspondiente, según se indica en el siguiente cuadro.

XSD	JSON	Notas
<code>xsd:union</code>	objeto - "anyOf"	Véase la sección "Transformación de tipos de datos integrados"

Por ejemplo:

Definición de esquema XML para un ejemplo de <code>xsd:simpleType</code> <code>xsd:union</code> (DocumentNameType.xsd)
<pre><?xml version="1.0" encoding="UTF-8"?> <xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common" elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0"> <xsd:include schemaLocation="DocumentNameCategoryType.xsd"/></pre>

```
<xsd:simpleType name="DocumentNameType">
  <xsd:union memberTypes="xsd:string com:DocumentNameCategoryType"/>
</xsd:simpleType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:simpleType\xsd:union (documentNameType.json)

```
{
  "$id" : "documentNameType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "documentNameType" : {
      "description" : "Version: V5_0",
      "anyOf" : [ {
        "type" : "string"
      }, {
        "$ref" : "documentNameCategoryType.json#/$defs/documentNameCategoryType"
      } ]
    }
  }
}
```

AMPLIACIÓN

[TR-17] DEBERÍA añadirse un objeto a la definición del componente correspondiente, según se indica en el siguiente cuadro.

XSD	JSON	Notas
xsd:extension	objeto - "anyOf"	Véase la sección "Transformación de tipos de datos integrados"

Por ejemplo:

Definición de esquema XML para un ejemplo de xsd:extension (AmountType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:include schemaLocation="currencyCode.xsd"/>
  <xsd:complexType name="AmountType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:decimal">
        <xsd:attribute ref="com:currencyCode"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de xsd:extension (amountType.json)

```
{
  "$id" : "amountType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "amountType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "$" : {
          "type" : "number"
        },
        "currencyCode" : {
          "$ref" : "currencyCode.json#/$defs/currencyCode"
        }
      }
    }
  }
}
```

```

    }
  }
}
}
}

```

RESTRICCIÓN

[TR-18] DEBERÍA añadirse un objeto a la definición del componente correspondiente, según se indica en el siguiente cuadro.

XSD	JSON	Notas
xsd:restriction	objeto	
xsd:pattern	patrón	Véase la sección “Patrón”
xsd:enumeration	enumeración	Véase la sección “Enumeración”

ENUMERACIÓN

[TR-19] El elemento `xsd:enumeration` DEBE transformarse en la propiedad “enum” en la definición del componente correspondiente del esquema JSON.

Definición de esquema XML para un ejemplo de `xsd:simpleType`\`xsd:restriction`\`xsd:enumeration` (`BusinessEntityStatusCategoryType.xsd`)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
  elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:simpleType name="BusinessEntityStatusCategoryType">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="Undiscounted">
        <xsd:annotation>
          <xsd:documentation>Undiscounted entity</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="Small">
        <xsd:annotation>
          <xsd:documentation>Small entity discount</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="Micro">
        <xsd:annotation>
          <xsd:documentation>Micro entity discount</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

Definición de esquema JSON para un ejemplo de `xsd:simpleType`\`xsd:restriction`\`xsd:enumeration` (`businessEntityStatusCategoryType.json`)

```

{
  "$id" : "businessEntityStatusCategoryType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "businessEntityStatusCategoryType" : {
      "description" : "Version: V5_0; Undiscounted: Undiscounted entity; Small: Small
entity discount; Micro: Micro entity discount",
      "type" : "string",
      "enum" : [ "Undiscounted", "Small", "Micro" ]
    }
  }
}

```

FACETAS DE RESTRICCIÓN

Los esquemas del W3C utilizan las facetas de restricción que figuran en el cuadro siguiente. Conviene señalar que `xsd:minInclusive`, `xsd:maxInclusive`, `xsd:minExclusive`, `xsd:maxExclusive`, y `xsd:minLength` no se utilizan en la Norma ST.96. Por consiguiente, esas facetas no utilizadas no se incluyen en el Anexo I.

[TR-20] Las facetas de restricción en las XSD DEBEN transformarse en las correspondientes palabras clave JSON con la longitud mínima o máxima adecuada, según se indica en el cuadro siguiente.

Cuadro 3. Transformación de las facetas de restricción en las XSD ("X" es el valor numérico de la restricción).

Faceta de restricción en la XSD	Equivalente en el esquema JSON
<code><xsd:minLength value="X" /></code>	<code>{ "minLength": X }</code>
<code><xsd:maxLength value="X" /></code>	<code>{ "maxLength": X }</code>
<code><xsd:length value="X" /></code>	<code>{ "minLength": X, "maxLength": X }</code>
<code><xsd:pattern value="X" /></code>	<code>{ "pattern": "X" }</code>
<code><xsd:minExclusive value="X" /></code>	<code>{ "minimum": X, "exclusiveMinimum": true }</code>
<code><xsd:maxExclusive value="X" /></code>	<code>{ "maximum": X, "exclusiveMaximum": true }</code>
<code><xsd:minInclusive value="X" /></code>	<code>{ "minimum": X, "exclusiveMinimum": false }</code>
<code><xsd:maxInclusive value="X" /></code>	<code>{ "maximum": X, "exclusiveMaximum": false }</code>

Por ejemplo:

Definición de esquema XML para un ejemplo de `xsd:length` (ClassType.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pat="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Patent"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
  <xsd:simpleType name="ClassType">
    <xsd:restriction base="xsd:token">
      <xsd:length value="2"/>
      <xsd:pattern value="[0-9][1-9]|[1-9][0-9]"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

Definición de esquema JSON para un ejemplo de `xsd:length` (classType.json)

```
{
  "$id" : "classType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "classType" : {
      "description" : "Version: V5_0",
      "type" : "string",
      "maxLength" : 2,
    }
  }
}
```

```

    "pattern" : "[0-9][1-9]|[1-9][0-9]"
  }
}
}

```

Patrón

[TR-21] El elemento `xsd:pattern` DEBE transformarse en la propiedad “pattern” en la definición del componente correspondiente.

Por ejemplo:

Definición de esquema XML para un ejemplo de `xsd:pattern` (WIPONotificationNumberType.xsd)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:com="http://www.wipo.int/standards/XMLSchema/ST96/Common"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.wipo.int/standards/XMLSchema/ST96/Common"
elementFormDefault="qualified" attributeFormDefault="qualified" version="V5_0">
<xsd:simpleType name="WIPONotificationNumberType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[A-Z]{3}[0-9]{6}" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

Definición de esquema JSON para un ejemplo de `xsd:pattern` (wipoNotificationNumberType.json)

```

{
  "$id" : "wipoNotificationNumberType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "wipoNotificationNumberType" : {
      "description" : "Version: V5_0",
      "type" : "string",
      "pattern" : "[A-Z]{3}[0-9]{6}"
    }
  }
}

```

GRUPO

El elemento `xsd:group` no se utiliza en la Norma ST.96, pero sí en normas externas a las que se refieren las XSD de la Norma ST.96. En la sección “Transformación de dependencias externas de las XSD” se puede encontrar más información sobre normas externas.

[TR-22] DEBERÍA añadirse el tipo de objeto a la definición del componente correspondiente.

XSD	JSON	Notas
<code>xsd:group</code>	objeto	

Por ejemplo:

Definición de esquema XML para un ejemplo de `xsd:group` (fragmento de FlattenedMathML3.json)

```

<xs:group name="anyElement">
  <xs:choice>
    <xs:any namespace="##other" processContents="skip"/>
    <xs:any namespace="##local" processContents="skip"/>
  </xs:choice>
</xs:group>

```

Definición de esquema JSON para un ejemplo de `xsd:group` (fragment from flattenedMathML3.json)

```

"anyElement" : {
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {

```

```
"other_SKIP" : {
  "patternProperties" : {
    "^@\\w+$" : {
      "type" : [ "string", "number", "boolean" ]
    }
  },
  "additionalProperties" : false,
  "type" : [ "object", "string", "number", "boolean" ],
  "properties" : {
    "$" : {
      "type" : [ "string", "number", "boolean" ]
    }
  }
},
"local_SKIP" : {
  "patternProperties" : {
    "^@\\w+$" : {
      "type" : [ "string", "number", "boolean" ]
    }
  },
  "additionalProperties" : false,
  "type" : [ "object", "string", "number", "boolean" ],
  "properties" : {
    "$" : {
      "type" : [ "string", "number", "boolean" ]
    }
  }
},
"oneOf" : [ {
  "required" : [ "other_SKIP" ]
}, {
  "required" : [ "local_SKIP" ]
} ]
}
```

TRANSFORMACIÓN DE DEPENDENCIAS EXTERNAS DE LAS XSD

En la Norma ST.96 de la OMPI se hace referencia a los siguientes esquemas desarrollados por la industria, en lugar de redefinirlos:

- MathML versión 3³ (FlattenedMathML3.xsd); y
- Esquema OASIS Table⁴ (OASISTable_V1_0.xsd).

En el momento de elaborar la presente Norma, esas normas externas no proporcionan esquemas JSON. No obstante, para solucionar desde el punto de vista de la programación el problema de que los esquemas JSON basados en la Norma ST.96 no son posibles sin disponer de esquemas JSON equivalentes, las XSD de esas dos normas externas se transforman en esquemas JSON equivalentes razonablemente estables mediante la herramienta de transformación que figura en el Apéndice del Anexo I de la presente Norma. Los esquemas JSON obtenidos mediante dicha herramienta se incluyen en el conjunto de esquemas JSON del Anexo II.

[Sigue el Apéndice del Anexo I de la Norma propuesta]

³ <http://www.w3.org/TR/MathML3>

⁴ <http://www.oasisopen.org/docbook/xmlschema/1.0b1/calstbl.xsd>

APÉNDICE

HERRAMIENTA DE TRANSFORMACIÓN DE LA DEFINICIÓN DE ESQUEMA XML (XSD) EN ESQUEMA JSON

Este Apéndice del Anexo I contiene una herramienta para transformar esquemas XML en esquemas JSON, que consiste en una biblioteca Java que facilita a las OPI la transformación de una XSD basada en la Norma ST.96 de la OMPI en su esquema JSON equivalente, conforme a las normas sobre transformación previstas en el Anexo I. La herramienta de transformación se proporciona como parte de la Norma, de modo que las OPI también pueden utilizarla para transformar en esquemas JSON sus propias XSD basadas en la Norma ST.96.

Requisitos

Java Runtime Environment 1.8 (o una versión posterior).

Uso

#Proporciona opciones de ayuda

```
java -jar xsd2JsonSchema.jar -help
```

#Transforma un único archivo (XSD)

```
java -jar xsd2JsonSchema.jar -f "C:\XSD_Folder_Path\Common\AbstractNumber.xsd"
```

#Transforma un único archivo (XSD) y los esquemas incluidos o importados

```
java -jar xsd2JsonSchema.jar -f -r "C:\XSD_Folder_Path\Common\AbstractNumber.xsd"
```

#Transforma todos los esquemas del directorio proporcionado (el indicador recursivo no está disponible en esta opción)

```
java -jar xsd2JsonSchema.jar -d "C:\XSD_Folder_Path\Common"
```

Descarga del archivo JAR ejecutable

La herramienta puede descargarse en el Apéndice del Anexo I en el siguiente enlace:

https://www.wipo.int/edocs/mdocs/cws/en/cws_10/cws_10_6-appendixi.zip

[Sigue el Anexo II de la Norma propuesta]

ANEXO II

ESQUEMA JSON

En el Anexo II se proporciona un conjunto completo de esquemas JSON que se corresponden con las XSD de la Norma ST.96 de la OMPI (versión 5.0). Esos esquemas JSON se generaron automáticamente utilizando la herramienta de transformación proporcionada en el Apéndice del Anexo I, conforme a las normas y directrices sobre transformación definidas en el Anexo I. Conviene señalar que se trata de un proceso de transformación unidireccional, de XSD a JSON. En el conjunto de esquemas JSON proporcionado se incluyen los esquemas JSON obtenidos transformando los esquemas XML basados en las normas externas MathML y OASIS Table.

Los esquemas JSON pueden descargarse en el Apéndice del Anexo II en el siguiente enlace:

https://www.wipo.int/edocs/mdocs/cws/en/cws_10/cws_10_6-appendixii.zip

[Sigue el Anexo III de la Norma propuesta]

ANEXO III

EJEMPLOS DE INSTANCIAS JSON

En el Anexo III se proporcionan ejemplos de instancias JSON obtenidas a partir de las XSD de la Norma ST.96 de la OMPI, con el fin de facilitar a las OPI la creación de instancias similares. Cada una de las instancias debería ser validada con el esquema JSON correspondiente incluido en el Anexo II.

Los siguientes ejemplos de instancias, que no son reales, corresponden a algunos de los componentes de nivel de documento que se proporcionan en el Anexo VII de la Norma ST.96 de la OMPI:

- **patentPublication.json**: este componente de nivel de documento se utiliza para registrar datos de la publicación de una solicitud de patente. Este ejemplo puede descargarse en el siguiente enlace:
patentPublication.json
- **trademarkApplication.json**: este componente de nivel de documento se utiliza para registrar datos relativos a una solicitud de registro de marca. Este ejemplo puede descargarse en el siguiente enlace:
trademarkApplication.json
- **designApplication.json**: este componente de nivel de documento se utiliza para registrar datos sobre una solicitud de registro de diseño industrial. Este ejemplo puede descargarse en el siguiente enlace:
designApplication.json

[Nota: los enlaces de descarga de esos ejemplos de instancias se facilitarán cuando se publique este proyecto de norma.]

Los ejemplos de instancias JSON pueden descargarse en el Apéndice del Anexo III en el siguiente enlace:
https://www.wipo.int/edocs/mdocs/cws/en/cws_10/cws_10_6-appendixiii.zip

[Sigue el Anexo IV de la Norma propuesta]

ANEXO IV

LISTA DE SIGLAS Y ABREVIATURAS

Las siglas y abreviaturas que aparecen al principio de un nombre de propiedad o de tipo de objeto DEBEN escribirse en minúsculas. Ejemplos: "pre", "bioDeposit". Si una sigla se proporciona entera en mayúsculas al principio de un nombre, todos los caracteres deben ir en minúsculas, por ejemplo, "idref" y "wipo" en el nombre de propiedad "wipoST3Code". En los demás casos, los valores de una enumeración, siglas y abreviaturas DEBEN aparecer en mayúsculas, según se indica a continuación.

Las siglas y abreviaturas que figuran a continuación NO DEBERÍAN considerarse en el contexto de los códigos de idioma, divisa, oficina y país, que figuran en la Norma ST.96 de la OMPI, ya que puede haber valores duplicados. Esos códigos se basan en los códigos de idioma ISO 639-1, los códigos de divisa ISO 4217, los códigos de la Norma ST.3 de la OMPI y los códigos de país ISO 3166-1, respectivamente.

AF	Fichero de referencia
Alt	Texto alternativo para la imagen
B	Negrita
BioDeposit	Depósito biológico
Br	Salto
CDX	<i>CambridgeSoft proprietary ChemDraw file format</i>
CPC	Clasificación de Patentes Cooperativa
DD	Descripción de definición
Del	Texto eliminado
DL	Lista de definición
DOI	Identificador de objeto digital
DT	Término de definición
DTD	Definición de tipo de documento
DWF	<i>Design Web Format</i>
DWG	Dibujo
ECLA	Clasificación europea
EIDR	Registro identificador de entretenimiento
ExtRef	Referencias que son externas al documento XML actual
GI	Indicación geográfica
H<n>	La "n" indica el nivel de encabezado con un valor específico, mediante un número de 1 a 15 dígitos. Es decir que, en el valor de enumeración, esta abreviatura representa uno de entre H1 y H15. Por ejemplo, "H1" significa "Heading 1".
I	Cursiva
IB	Oficina Internacional
ID	Identificador para la identificación del sistema
IDREF	Referencia del identificador
IDREFS	Referencias del identificador
IGES	<i>Initial Graphic Exchange Specification</i>
IGO	Organización intergubernamental cuya legislación prevé la protección de los derechos de propiedad intelectual o que actúa en el marco de un tratado en el ámbito de la propiedad intelectual
INID	Números acordados internacionalmente para la identificación de datos (bibliográficos)
Ins	Texto insertado
IP	Propiedad intelectual
IPC	Clasificación Internacional de Patentes
IPCR	Reforma de la Clasificación Internacional de Patentes
IPO	Oficina de propiedad intelectual
IPR	Derecho de propiedad intelectual
ISMN	Número internacional normalizado para música
ISNI	Identificador internacional normalizado de nombres
ISO	Organización Internacional de Normalización
JSON	Notación de objetos de JavaScript
LCC	<i>Lower Camel Case</i>
LI	Ítem de la lista
LOR	Licencia de pleno derecho
MathML	<i>Mathematical Markup Language</i>
MPEG	<i>Moving Picture Experts Group</i>

MOL	Formato de archivo para guardar información sobre átomos, enlaces, conectividad y coordenadas de una molécula
NB	Formato de archivos para cuadernos Mathematica
NPL	Documentos distintos de los de patentes
NUTS	Nomenclatura de las Unidades Territoriales Estadísticas
O	Tachado
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>
OCR	Reconocimiento óptico de caracteres
OL	Lista ordenada
P	Párrafo
PAN	Número primario de cuenta
PCT	Tratado de Cooperación en materia de Patentes
PKCS7	En criptografía , PKCS es un grupo de normas de criptografía asimétrica , y PKCS #7 (PKCS7) corresponde a la norma de sintaxis de mensajes codificados, que describe la sintaxis general de los datos a los que puede haberse aplicado criptografía, por ejemplo, firmas y sobres digitales.
Pre	Texto preformateado
S	Texto tachado
SQL	Listas de secuencias
SOC	Código de sociedad
SPC	Certificado complementario de protección
ST3	Norma ST.3 de la OMPI
ST13	Norma ST.13 de la OMPI
Sub	Subíndice
Sup	Superíndice
SVG	<i>Imagen Scalable Vector Graphics</i>
SWF	<i>Small Web Format</i>
SWIFT	Sociedad de Telecomunicaciones Interbancarias Mundiales
ThreeDM	Modelado dimensional
ThreeDS	3D Studio
TISA	Código del sistema de información territorial de la CISAC - alfanumérico
TISN	Código del sistema de información territorial de la CISAC - numérico
TSG	Especialidades tradicionales garantizadas
U	Subrayado
UCC	<i>Upper Camel Case</i>
UL	Lista no ordenada
UPOV	Unión Internacional para la Protección de las Obtenciones Vegetales
URI	Identificador uniforme de recursos
URL	Localizador uniforme de recursos
URN	Nombre uniforme de recursos
W3C	Consorcio <i>World Wide Web</i>
WIPO	Organización Mundial de la Propiedad Intelectual
WMV	<i>Windows Media Video</i>

[Sigue el Anexo V de la Norma propuesta]

ANEXO V
TÉRMINOS DE REPRESENTACIÓN

Término	Definición	Tipo de datos
Amount (Cuantía)	Valor monetario.	number
Category (Categoría)	División o subconjunto en un sistema de clasificación específicamente definido cuyos elementos comparten, todos ellos, el mismo concepto de taxonomía.	string
Code (Código)	Combinación de uno o más números, letras o caracteres especiales, que se sustituye por un significado concreto. Representa valores finitos y predeterminados o formato libre.	string
Date (Fecha)	Punto concreto en el tiempo expresado mediante el año, el mes y el día.	string, con la palabra clave "format": "date"
Identifier (Identificador)	Combinación de uno o más números enteros, letras o caracteres especiales que identifican de manera exclusiva una aplicación concreta de un objeto pero que quizá no tenga un significado fácilmente definible.	string
Indicator (Indicador)	Señal de presencia, ausencia o requisito de algo. Los valores booleanos son <code>true</code> o <code>false</code> (sin comillas). Estos valores distinguen entre mayúsculas y minúsculas.	boolean or string
Measure (Medida)	Valor numérico que se determina midiendo un objeto y que se expresa mediante la unidad de medida especificada. <code>MeasureType</code> 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, <code>MeasureType</code> debe usarse para medir los componentes físicos o intrínsecos de un objeto percibido como un conjunto.	number
Name (Nombre)	Designación de un objeto expresada mediante una palabra o una frase.	String
Number (Número)	Cadena de numerales o caracteres alfanuméricos que expresan una etiqueta, un valor, una cantidad o una identificación.	integer, number, 0 or string
Percent (Por ciento)	Número que representa una parte de un conjunto, que se dividirá por 100.	number
Quantity (Cantidad)	Número contado de unidades no monetarias y que posiblemente incluye fracciones. <code>Quantity</code> se utiliza para representar un número de cosas contadas. <code>Quantity</code> debe utilizarse para contar o cuantificar los componentes que integran un objeto percibido como un compuesto, una colección o un contenedor. <code>Quantity</code> debería siempre expresar un número de cosas contadas, y el número se referirá a la cantidad total, enviada, cargada o almacenada. <code>QuantityType</code> debería utilizarse para componentes que requieren información sobre la unidad; e <code>Integer</code> debería utilizarse para componentes susceptibles de ser contados que no requieren información sobre la unidad.	number con la palabra clave "minimum": 0 or integer
Rate (Índice)	Cantidad o cuantía medida con respecto a otra cantidad o cuantía.	number
Text (Texto)	Cadena de caracteres con o sin formato, por lo general en forma de palabras (incluye: abreviaturas y comentarios).	string
Time (Tiempo)	Designación de un punto cronológico concreto en un período.	string, con la palabra clave "format": "time"
DateTime	Fecha y hora en la que se produce un evento.	string, con la palabra clave "format": "date-time"
URI	Identificador uniforme de recursos que identifica la ubicación del archivo.	string, con la palabra clave "format": y los valores "uri" or "uri-reference"

[Fin del Anexo V de la Norma propuesta y fin de la Norma propuesta]

[Fin del Anexo y del documento]