NORMA ST.97 DE LA OMPI

Versión 2.0

RECOMENDACIÓN PARA EL PROCESAMIENTO DE DATOS DE PROPIEDAD INTELECTUAL MEDIANTE JSON

Revisión aprobada por el Equipo Técnico sobre API del Comité de Normas Técnicas de la OMPI (CWS) para ser publicada el 15 de julio de 2026

ÍNDICE

INTRODUCCIÓN....3
DEFINICIONES Y TERMINOLOGÍA....3
NOTACIONES GENERALES....4
Identificadores de reglas....4
Ejemplo de estructura de datos JSON....5
ALCANCE DE LA NORMA....5
REGLAS GENERALES DE DISEÑO Y CONVENCIONES JSON....6
Información general....6
Convenciones de denominación JSON....6
REGLAS DE DISEÑO DE ESQUEMAS JSON....7
Información general....7
Modularidad....8
Documentación....16
Convención de denominación de archivos de esquema....18
Estructuración de las propiedades del esquema JSON....18
REGLAS DE DISEÑO DE LAS CONSTRUCCIONES DE ESQUEMA JSON....19
Información general....19
Propiedades....19
Definiciones....19
Definiciones de tipo....20
Tipo primitivo JSON....21
Listas de códigos....21
Matrices....21
Objetos....22
IDENTIFICADORES DE ESQUEMA JSON....22
Información general....22
REGLAS DE DISEÑO DE INSTANCIAS JSON....22
Orden de las propiedades....23

Validación de instancias JSON....23
REFERENCIAS....23
Normas de la OMPI....23
Normas y especificaciones del sector....23

ANNEXOS

Los Anexos están disponibles en: http://www.wipo.int/standards/es/st97/v2-0/

Annexo I – Reglas de transformación de las XSD de la norma ST.96 en esquemas JSON y directrices de uso

Annexo II – Esquema JSON

Annexo III – Ejemplos de instancias JSON

Annexo IV – Lista de siglas y abreviaturas

Annexo V – Términos de representación

Annexo VI – Diccionario de datos de PI

NORMA ST.97 DE LA OMPI

Versión 2.0

RECOMENDACIÓN PARA EL PROCESAMIENTO DE DATOS DE PROPIEDAD INTELECTUAL MEDIANTE JSON

Revisión aprobada por el Equipo Técnico sobre API del Comité de Normas Técnicas de la OMPI (CWS) para ser publicada el 15 de julio de 2026

INTRODUCCIÓN

1. En esta Norma se ofrecen recomendaciones para diseñar, crear o actualizar recursos de notación de objetos de JavaScript (JavaScript Object Notation, JSON) para utilizarlos en la presentación, el procesamiento, el intercambio o la publicación de todo tipo de datos de propiedad intelectual (PI). Asimismo, esta Norma incluye reglas para transformar los esquemas de lenguaje extensible de marcado (XML) (XSD) de la Norma ST.96 de la OMPI en esquemas JSON que se ajusten a las recomendaciones mencionadas, dado que los esquemas JSON se derivan de la Norma ST.96 de la OMPI.

2. La finalidad de esta Norma es:

- Ofrecer esquemas e instancias JSON bien estructurados;
- Proporcionar directrices para diseñar y desarrollar mejores prácticas en materia de datos JSON de PI;
- Recomendar principios de diseño para ampliar los esquemas JSON proporcionados o crear nuevos esquemas JSON conformes; y
- Mejorar la eficiencia del intercambio de datos fomentando la reutilización de esquemas e instancias JSON entre las Oficinas de Propiedad Intelectual (OPI), así como de los datos puestos a disposición del público.

DEFINICIONES Y TERMINOLOGÍA

3. A los efectos de la presente Norma, se utilizan las siguientes definiciones:

- El término “recursos JSON” en esta Norma se refiere a cualquiera de los componentes utilizados para crear y operar una implementación JSON conforme con esta Norma;
- Los términos “JSON”, “JSON text”, “JSON value”, “array”, “element”, “object”, “object type”, “property”, “member”, “property name”, “property value”, “property type”, “keyword”, “number”, “string”, “boolean”, “true”, “false”, “null” y “definition” en esta Norma han de interpretarse según se definen en el esquema JSON Core, versión 2020-12;[1]
- El término “construcción” en esta Norma debe interpretarse como un “componente básico” a partir del cual se construyen los esquemas JSON;
- El término “definición global” se utiliza para una definición a la que pueden hacer referencia otras definiciones del mismo esquema o definiciones de otros esquemas;
- El término “patente” comprende derechos de propiedad industrial tales como las patentes de invención, las patentes de plantas, las patentes de diseño, los certificados de inventor, los certificados de utilidad, los modelos de utilidad, las patentes de adición, los certificados de inventor de adición y los certificados de utilidad de adición;
- El término “marca” significa marca de fábrica o de comercio, marca de servicio u otro tipo de signo distintivo de marca según la definición de marca en la legislación de que se trate, incluidas, entre otras, las marcas colectivas, las marcas de certificación o las marcas de garantía;

[1] La versión del esquema JSON está sujeta a cambios, ya que aún no ha alcanzado el estatus de RFC; no ha sido adoptada por ningún grupo de trabajo de la IETF. Esta versión de la norma se basa en la última versión disponible, es decir, la 2020-12, que puede consultarse en https://json-schema.org/draft/2020-12/json-schema-core.html

- El término “diseño industrial” comprende las características bidimensionales y tridimensionales de la forma y la superficie de los objetos y, por tanto, abarca tanto los conceptos de “dibujos” como de “modelos” cuando se distingue entre unos y otros; el término “diseños industriales” no incluye las patentes de diseño;

- El término “indicación geográfica” significa un signo utilizado en productos que tienen un origen geográfico específico y poseen cualidades o una reputación que se deben a ese origen; y

- El término “derecho de autor” comprende los derechos que los creadores tienen sobre sus obras literarias y artísticas, y la expresión “obra huérfana protegida por derecho de autor” comprende las obras protegidas por derecho de autor cuyo autor no puede ser identificado o localizado.

4. Las palabras clave “DEBE”, “NO DEBE”, “DEBERÍA”, “NO DEBERÍA”, “RECOMENDADO”, “PUEDE” y “OPCIONAL”, cuando aparezcan en mayúsculas en esta Norma, han de interpretarse según se describe a continuación únicamente cuando aparezcan íntegramente en mayúsculas. Las formas no escritas en mayúsculas de estas palabras se utilizan en su sentido habitual en español.

- DEBE: esta palabra, o los términos “OBLIGATORIO” o “HABRÁ DE”, significa que la definición constituye un requisito absoluto de la especificación;

- NO DEBE: esta expresión, o la expresión “NO HABRÁ DE”, significa que la definición constituye una prohibición absoluta de la especificación;

- DEBERÍA: esta palabra, o el adjetivo “RECOMENDADO”, significa que pueden existir razones válidas en determinadas circunstancias para ignorar un elemento concreto, pero las implicaciones completas deben comprenderse y sopesarse cuidadosamente antes de optar por una alternativa;

- NO DEBERÍA: esta expresión, o la expresión “NO RECOMENDADO”, significa que pueden existir razones válidas en determinadas circunstancias en que el comportamiento concreto sea aceptable o incluso útil, pero las implicaciones completas deberían comprenderse y el caso sopesarse cuidadosamente antes de implementar cualquier comportamiento descrito con esta etiqueta; y

- PUEDE: esta palabra, o el adjetivo “OPCIONAL”, significa que un elemento es verdaderamente opcional. Una implementación que no incluya una opción concreta DEBE estar preparada para interoperar con otra implementación que sí la incluya, aunque tal vez con funcionalidad reducida. Del mismo modo, una implementación que sí incluya una opción concreta DEBE estar preparada para interoperar con otra implementación que no la incluya (excepto, naturalmente, en lo que respecta a la función que ofrece la opción).

NOTACIONES GENERALES

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

- <>: indica un término descriptivo provisional que, en la implementación, será sustituido por un valor específico de instancia. Esta convención no se aplica cuando aparece como parte de la sintaxis del esquema;

- “ ”: indica que el texto incluido entre comillas debe utilizarse literalmente en la implementación. Esta convención no se aplica cuando aparece como parte de la sintaxis del esquema;

- { }: indica que los elementos son opcionales en la implementación. Esta convención no se aplica cuando aparece como parte de la sintaxis del esquema; y

- `Courier New`: indica palabras clave JSON y XML, nombres de componentes y códigos.

Identificadores de reglas

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

- El valor “JXX” es un prefijo para clasificar el tipo de regla de la siguiente manera:

  a) JGD para las reglas generales de diseño;

  b) JSD para las reglas de diseño de esquemas JSON;

  c) JCD para las reglas de diseño de construcciones; y

d) JID para las reglas de diseño de instancias.

– El valor “nn” indica el siguiente número disponible en la secuencia de un tipo específico de regla. Cabe observar que el número no indica la posición de la regla, en particular en el caso de una regla nueva. Una nueva regla se colocará en el contexto pertinente. Por ejemplo, el identificador de regla [JGD-10] identifica la décima regla general de diseño. La regla [JGD-10] puede colocarse entre las reglas [JGD-05] y [JGD-06] en lugar de a continuación de la regla [JGD-09] si esa es la ubicación más adecuada para esa regla.

– El identificador de una regla suprimida se conservará, pero la regla será sustituida por el texto “Suprimida”.

Ejemplo de estructura de datos JSON

7. Los ejemplos de estructuras de datos JSON aparecen dentro de cuadros de texto con una fuente de ancho fijo. La sintaxis de los ejemplos de estructuras de datos JSON está resaltada para facilitar la lectura.

ALCANCE DE LA NORMA

8. Esta Norma tiene por objeto proporcionar recursos JSON que se utilizarán para la presentación, la publicación, el procesamiento y el intercambio de datos e información sobre PI. Está destinada a orientar a las OPI y a otras organizaciones que tratan con datos y documentos de patentes, marcas, diseños industriales, indicaciones geográficas u obras huérfanas protegidas por derecho de autor.

9. La finalidad de esta Norma es proporcionar orientación a las OPI y a otras organizaciones que crean o modifican datos de PI como recursos JSON. Para el intercambio de datos entre OPI mediante recursos JSON tales como esquemas, instancias, mensajes y cargas útiles para interfaces de programación de aplicaciones (API), se requiere el cumplimiento de esta Norma.

10. Las reglas y convenciones de diseño contenidas en el presente documento proporcionan un conjunto exhaustivo de reglas para la creación de esquemas e instancias JSON utilizados para todos los tipos de información de PI. Las reglas de transformación se proporcionan para orientar a las OPI sobre cómo transformar sus esquemas ST.96 personalizados en esquemas JSON equivalentes.

11. La presente Norma incluye los siguientes Anexos:

– Anexo I: Reglas de transformación de esquemas XML ST.96 a esquemas JSON.

– Anexo II: Esquemas JSON;[2]

– Anexo III: Ejemplos de instancias JSON;

– Anexo IV: Lista de siglas y abreviaturas;

– Anexo V: Términos de representación; y

– Anexo VI: Diccionario de datos de PI.

12. La presente Norma excluye lo siguiente:

a) Cuestiones de arquitectura de software; y

b) Lenguajes de implementación.

13. Todos los objetos, excepto los objetos de tipo de datos, y los valores de enumeración definidos por la presente Norma DEBEN figurar en el Diccionario de datos de PI, es decir, en el Anexo VI de la Norma.

14. Cada versión del Anexo II DEBE publicarse en su totalidad, incluidos los componentes de esquemas JSON actualizados y no modificados.

[2] Los esquemas JSON transformados tienen, en general, los mismos nombres de componentes y la misma estructura de datos que los definidos en la versión correspondiente del ST.96, anexo III. No obstante, se han introducido varios cambios manuales en algunos esquemas JSON, necesarios para adaptar las diferencias entre las tecnologías XML y JSON. En las notas de la versión correspondiente al ST.97 se ofrecen más detalles sobre los esquemas modificados manualmente.

REGLAS GENERALES DE DISEÑO Y CONVENCIONES JSON

Información general

15. En esta sección se establecen reglas y convenciones generales de diseño JSON, de alto nivel, que se aplican a todos los intercambios de datos JSON y a todas las actividades de desarrollo JSON, y no a la serialización o deserialización de datos hacia o desde JSON en un lenguaje de programación específico. Las reglas y convenciones generales que se enumeran a continuación proporcionan una base común para el desarrollo de esquemas JSON, instancias JSON y estructuras de datos JSON para todos los tipos de datos, incluidos los datos de PI y los datos que no son de PI, como los datos de contenido mixto. Al crear nuevos esquemas JSON, instancias JSON o estructuras de datos JSON que no estén disponibles en la Norma ST.97 de la OMPI, y cuando no sea necesario tener en cuenta la compatibilidad con ST.96, los niveles de anidamiento DEBERÍAN reducirse al mínimo.

Convenciones de denominación JSON

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

17. Las convenciones de denominación JSON establecidas en esta sección se basan en las directrices y principios descritos en el documento ISO 11179, Parte 5[3] (Principios de denominación e identificación). Los nombres de objeto y los nombres de propiedad se componen de los siguientes términos:

- *Clase de objeto* se refiere a una actividad u objeto dentro de un contexto empresarial 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.
- *Término de propiedad* identifica las características de la clase de objeto.
- *Término calificador* es una palabra o conjunto de palabras que ayudan a definir y diferenciar un elemento de datos de otros elementos de datos relacionados, y puede adjuntarse a un término de clase de objeto o a un término de propiedad si es necesario para hacer que un nombre sea único.
- *Término de representación* clasifica el formato del elemento de datos en tipos generales. Deberían utilizarse los términos de representación enumerados en el Anexo V.

[JGD-01] Los nombres de tipo de objeto y de propiedad DEBEN estar compuestos por palabras en lengua inglesa, utilizando las grafías principales del inglés que figuran en el Oxford English Dictionary. Las únicas excepciones permitidas son las siglas, abreviaturas y otras truncaciones de palabras enumeradas en el Anexo IV.

[JGD-02] Los nombres de tipo de objeto y de propiedad DEBERÍAN consistir únicamente en sustantivos, adjetivos y verbos en presente, con excepción de las siglas, abreviaturas y otras truncaciones de palabras enumeradas en el Anexo IV.

[JGD-03] Los caracteres utilizados en los nombres de propiedad DEBEN estar incluidos en el siguiente conjunto: ‘a-z, A-Z y 0-9’.

[JGD-04] La longitud máxima de los nombres de tipo de objeto y de propiedad NO DEBERÍA exceder de 35 caracteres. Se recomienda utilizar abreviaturas y siglas para reducir la longitud de los nombres de componentes. Toda nueva abreviatura o sigla DEBE añadirse al Anexo IV.

[JGD-05] Los nombres de tipo de objeto y de propiedad DEBERÍAN ser concisos y autoexplicativos.

[JGD-06] Los nombres de tipo de objeto y de propiedad DEBEN utilizar la convención lowerCamelCase (LCC). Por ejemplo, "`currencyCode`":"EUR".

[JGD-07] Los nombres de tipo de objeto DEBEN utilizar la convención LCC y llevar el sufijo Type. Por ejemplo, `applicantType`.

[JGD-08] Las siglas y abreviaturas enumeradas en el Anexo IV DEBEN utilizarse siempre en lugar del nombre desarrollado completo.

[3] https://www.iso.org/standard/60341.html

[JGD-09] Las siglas y abreviaturas DEBEN aparecer tal como figuran en el Anexo IV en los nombres de propiedad y de tipo de objeto.

[JGD-24] Las siglas y abreviaturas que aparezcan al comienzo de un nombre de componente DEBEN escribirse en lower camel case, por ejemplo, "`pre`", "`bioDeposit`". Si una sigla se proporciona íntegramente en mayúsculas y aparece al comienzo de un nombre de componente, todos los caracteres deben escribirse en minúsculas, por ejemplo, "`idref`" y "`wipo`" en el nombre de propiedad "`wipoST3Code`". En los demás casos, todos los valores de enumeración, siglas y abreviaturas DEBEN aparecer en mayúsculas, tal como figuran en el Anexo IV.

[JGD-10] Un término de clase de objeto DEBE tener siempre el mismo significado semántico en todo un dominio específico de PI, como patentes, marcas, diseños industriales, indicaciones geográficas o derecho de autor, y PUEDE constar de más de una palabra. Por ejemplo, `contactInformation`.

[JGD-11] Un término de propiedad en un nombre DEBE ser único en el contexto de una clase de objeto, pero PUEDE reutilizarse en distintas clases de objeto.

[JGD-12] Un término calificador PUEDE adjuntarse a un término de clase de objeto o a un término de propiedad si es necesario para hacer que un nombre sea único.

[JGD-13] Cuando un nombre contenga un término de clase de objeto, un término de propiedad y un término de representación, el término de clase de objeto DEBE preceder al término de propiedad, y el término de propiedad DEBE preceder al término de representación. Un término calificador DEBERÍA preceder al término de clase de objeto o al término de propiedad asociado. Por ejemplo, `claimTotalQuantity`.

[JGD-14] Si el término de propiedad termina con la misma palabra que el término de representación (o con una palabra equivalente), el término de representación DEBE suprimirse.

[JGD-15] Cuando se requiera un término de representación, DEBEN utilizarse los términos de representación del Anexo V en los nombres de componentes básicos.

[JGD-16] Dentro de un dominio de PI, todos los nombres de tipo de objeto y de propiedad DEBEN ser únicos.

[JGD-17] Las palabras de un nombre DEBERÍAN estar en singular, salvo que el concepto en sí 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 contextualmente DEBERÍA llevar el sufijo "Bag". Por ejemplo, `emailAddressBag` representa una colección de elementos `emailAddress`.

[JGD-19] Las palabras de enlace como "and", "of" y "the" NO DEBERÍAN utilizarse en los nombres de tipo de objeto ni de propiedad, salvo que formen parte de la terminología propia.

[JGD-20] Los nombres de tipo de objeto y de propiedad NO DEBEN traducirse, modificarse ni sustituirse para ningún propósito.

[JGD-21] Los nombres de tipo de objeto y de propiedad NO DEBEN hacer referencia a números de artículos ni de reglas. Por ejemplo, `PCTRule702C` para el PCT.

[JGD-22] Los niveles de anidamiento DEBERÍAN mantenerse al mínimo al crear nuevos esquemas JSON, instancias JSON y estructuras o fragmentos de datos JSON.

[JGD-23] *Suprimida*

REGLAS DE DISEÑO DE ESQUEMAS JSON

Información general

18. El esquema JSON describe la estructura de la instancia JSON, que expresa las restricciones aplicables a la estructura y el contenido del documento. Esta Norma debe ajustarse a la especificación del sector relativa al esquema

JSON. La última versión disponible en el momento de publicación de la presente Norma es 2020-12[4] y esta versión de la Norma remite a ese proyecto de especificación.

[JSD-01] Los esquemas JSON DEBEN ajustarse a las especificaciones del esquema JSON Core, versión 2020-12, y Validation, versión 2020-12.

[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

```
"$schema": "https://json-schema.org/draft/2020-12/schema"
```

19. El esquema JSON debe codificarse en UTF-8 para lograr la máxima interoperabilidad.

[JSD-03] Los esquemas JSON DEBEN utilizar el conjunto de caracteres ISO/IEC 10646 – UCS – Unicode. DEBE utilizarse UTF-8 para codificar los caracteres Unicode.

Modularidad

20. La modularidad permite crear componentes de esquema para favorecer la flexibilidad del diseño y la reutilización. En el diseño, se recomienda evitar la definición de todas las propiedades y componentes lógicos en un único esquema JSON monolítico, lo que impediría compartir y reutilizar propiedades individuales o componentes lógicos definidos como grupo en un esquema.

21. A continuación figura un esquema que **no** se ajusta al principio de modularidad. La presente Norma NO lo recomienda.

`applicationNumber.json` (ejemplo incorrecto de documento de esquema compuesto)

```
{
  "$id" : "applicationNumber.json",
  "$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: V2_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",
```

[4] https://json-schema.org/draft/2020-12/schema

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

```
Tajikistan: TJ; Timor-Leste: TL; Turkmenistan: TM; Tunisia: TN; Tonga: TO; Türkiye:
TR; Trinidad and Tobago: TT; Tuvalu: TV; Taiwan, Province of China: TW; United
Republic of Tanzania: TZ; Ukraine: UA; Uganda: UG; United States of America: US;
Uruguay: UY; Uzbekistan: UZ; Holy See: VA; Saint Vincent and the Grenadines: VC;
Venezuela (Bolivian Republic of): VE; British Virgin Islands: VG; Viet Nam: VN;
Vanuatu: VU; World Intellectual Property Organization (WIPO) (International Bureau
of): WO; Samoa: WS; Nordic Patent Institute (NPI): XN; International Union for the
Protection of New Varieties of Plants (UPOV): XU; Visegrad Patent Institute (VPI):
XV; Unknown states, other entities or organizations: XX; Yemen: YE; South Africa:
ZA; Zambia: ZM; Zimbabwe: ZW; Version: V2_0"        }, {
           "type" : "string",
            "enum" : [ "AN", "CS", "DL", "DD", "DT", "RH", "SU", "YD", "YU" ],
            "description" : "Version: V2_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: V2_0”
        },
        "applicationNumberText" : {
          "type" : "string",
         "description" : "Description: Free format of application number; Version:
V2_0"
        }
      },
      "oneOf" : [ {
        "required" : [ "st13ApplicationNumber" ]
      }, {
        "required" : [ "applicationNumberText" ]
      } ]
    }
  }
}
```

22. El enfoque de diseño preferido consiste en dividir los datos en un conjunto de pequeños componentes representados por módulos de esquema, como se muestra más abajo en el nuevo esquema de número de solicitud. Este esquema JSON se basa en módulos de esquema JSON más pequeños definidos individualmente en 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" ],
  "$defs" : {
    "applicationNumber" : {
      "description" : "Description: Numbers used by IPOs in order to identify
each application received; Version: V2_0",
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType"

    }
  }
}
```

`applicationNumberType.json` (continuación del ejemplo de esquema modular)

```
{
  "$id" : "applicationNumberType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "$defs" : {
    "applicationNumberType" : {
      "type" : "object",
      "unevaluatedProperties" : false,
      "description" : "Version: V2_0",
      "properties" : {
        "ipOfficeCode" : {
          "$ref" : "ipOfficeCode.json#/$defs/ipOfficeCode"
        }
      },
      "allOf" : [ {
        "oneOf" : [ {
          "allOf" : [ {
            "properties" : {
              "st13ApplicationNumber" : {
                "$ref" :
"st13ApplicationNumber.json#/$defs/st13ApplicationNumber"
              }
            },
            "required" : [ "st13ApplicationNumber" ]
          }, {
            "not" : {
              "required" : [ "applicationNumberText" ]
            }
          } ]
        }, {
          "allOf" : [ {
            "properties" : {
              "applicationNumberText" : {
                "$ref" :
"applicationNumberText.json#/$defs/applicationNumberText"
              }
            },
            "required" : [ "applicationNumberText" ]
```

```
          }, {
            "not" : {
              "required" : [ "st13ApplicationNumber" ]
            }
          } ]
        } ]
      } ]
    }
  }
}
```

`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" : {

      "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: V2_0",
"$ref" : "extendedWIPOST3CodeType.json#/$defs/extendedWIPOST3CodeType",

    }
  }
}
```

`extendedWIPOST3CodeType.json` (continuación del ejemplo de esquema modular)

```
{
  "$id" : "extendedWIPOST3CodeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "$defs" : {
    "extendedWIPOST3CodeType" : {
      "description" : "Version: V2_0",
      "anyOf" : [ {
        "$ref" : "wipoST3CodeType.json#/$defs/wipoST3CodeType"
      }, {
        "$ref" : "wipoFormerST3CodeType.json#/$defs/wipoFormerST3CodeType"
      }, {
        "$ref" : "igoCodeType.json#/$defs/igoCodeType"
      } ]
    }
  }
}
```

`wipoST3CodeType.json` (continuación del ejemplo de esquema modular)

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

```
Taiwan, Province of China: TW; United Republic of Tanzania: TZ; Ukraine: UA;
Uganda: UG; United States of America: US; Uruguay: UY; Uzbekistan: UZ; Holy See:
VA; Saint Vincent and the Grenadines: VC; Venezuela (Bolivian Republic of): VE;
British Virgin Islands: VG; Viet Nam: VN; Vanuatu: VU; World Intellectual
Property Organization (WIPO) (International Bureau of): WO; Samoa: WS; Nordic
Patent Institute (NPI): XN; International Union for the Protection of New
Varieties of Plants (UPOV): XU; Visegrad Patent Institute (VPI): XV; Unknown
states, other entities or organizations: XX; Yemen: YE; South Africa: ZA;
Zambia: ZM; Zimbabwe: ZW; Version: V2_0",
      "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", "FM", "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", "MH", "MK",
"ML", "MM", "MN", "MO", "MP", "MR", "MS", "MT", "MU", "MV", "MW", "MX", "MY",
"MZ", "NA", "NE", "NG", "NI", "NL", "NO", "NP", "NR", "NU", "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",
  "type" : "object",
  "additionalProperties" : false,
  "$defs" : {
    "wipoFormerST3CodeType" : {
      "type" : "string",
      "description" : "Description: Netherlands Antilles: AN; Czechoslovakia:
CS; German Democratic Republic: DD; Federal Republic of Germany: DT; Southern
Rhodesia: RH; Soviet Union: SU; Democratic Yemen: YD; Yugoslavia/ Serbia and
Montenegro: YU; Version: V2_0",
      "enum" : [ "AN", "CS", "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: V2_0"
    }
  }
}
```

st13ApplicationNumberType.json (continuación del ejemplo de esquema modular)

```
{
  "$id" : "st13ApplicationNumberType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "$defs" : {
    "st13ApplicationNumberType" : {
      "type" : "string",      "description" : "Version: V2_0",

      "pattern" : "\\d{2}\\d{4}\\\\w{2}d{7}"
    }
  }
}
```

`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:
V2_0"
    }
  }
}
```

23. Los esquemas JSON deberían utilizar la palabra clave "`$defs`" en la creación de definiciones globales reutilizables para las propiedades y sus contenidos, como se muestra en el ejemplo anterior. Esto equivale aproximadamente a crear declaraciones de elementos globales 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] Los desarrolladores DEBEN utilizar los esquemas JSON existentes que se definen en el Anexo II de la presente Norma, siempre que proceda, antes de crear nuevos esquemas JSON.

[JSD-06] Los desarrolladores DEBERÍAN crear nuevos esquemas JSON solo después de determinar que ningún esquema JSON existente describe adecuadamente la construcción de que se trate.

Documentación

24. Los esquemas JSON deberían ser autodescriptivos. Los desarrolladores deberían procurar que los nombres de las construcciones JSON sean significativos. Además, el esquema JSON debería incluir documentación que describa el esquema y las construcciones JSON. Para favorecer la reutilización manteniendo su carácter general, el esquema JSON no debería proporcionar documentación sobre detalles de implementación específicos del sistema.

[JSD-07] La documentación NO DEBERÍA describir detalles de implementación ni otra información que no esté directamente relacionada con el significado de la construcción.

25. La descripción de un esquema JSON permite a los desarrolladores de esquemas reconocer fácilmente la finalidad, el uso y el contenido de un esquema. Esta información resulta muy útil cuando se necesita seleccionar un esquema para utilizarlo como plantilla en la creación de otro esquema.

[JSD-08] Los esquemas JSON DEBERÍAN incluir documentación del esquema JSON mediante la palabra clave "`description`".

[JSD-09] Los elementos enumerados en el cuadro 1 que figura a continuación DEBERÍAN incluirse en la sección de documentación de todos los esquemas JSON.

Cuadro 1. Elementos de descripción del esquema JSON

Nombre del elemento de descripción	Valor	Obligatorio/opcional
`Description`	Descripción, en texto sin formato, de la información descrita por el esquema	Obligatorio, excepto en esquemas JSON procedentes de tipos simples de las XSD de la Norma ST.96 para los que no se proporciona una descripción, como `DateType`.
`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 o datos de contacto de la organización a la que dirigirse para formular preguntas sobre el esquema	Opcional
`SchemaReleaseNoteURL`	Ubicación donde se publican las notas de la versión del esquema	Opcional

[JSD-10] Los elementos de descripción, como la fecha de creación del esquema y el número de versión indicados anteriormente, DEBERÍAN separarse mediante punto y coma, con espacios permitidos después del punto y coma, y constituir el valor asociado a la palabra clave `description`, como se muestra en el siguiente ejemplo:

Ejemplo de descripción de `applicationBody` (esquema de nivel de documento)

```
"description" : "Description: Body of a patent application; SchemaCreatedDate:
2012-07-13; SchemaLastModifiedDate: 2026-06-09; SchemaContactPoint:
standards@wipo.int; SchemaReleaseNoteURL:
https://www.wipo.int/standards/en/json_material/st97/v2-0/release-notes.html;
Version: V2_0 "
```

Ejemplo de descripción de `ipOfficeCode` (esquema que no es 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: V2_0"
```

Ejemplo de descripción de `appellateBodyCategoryType.json` (esquema de definiciones de tipo de enumeración)

```
"description" : " Office appeal board: Appeal board within the IP office; Court:
Court; Appeal Court: Second instance court; Supreme Court: Highest appellate court,
Version: V2_0;"
```

Ejemplo de descripción de un esquema de definiciones de tipo
`"description" : "Version: V2_0"`

Convención de denominación de archivos de esquema

26. El nombre de archivo del esquema JSON debería utilizar la convención LCC.

27. Los nombres de archivo de esquema y los nombres de esquema suelen ir emparejados. Los nombres de archivo de esquema se basan en los nombres de esquema correspondientes. Por ejemplo, el nombre de archivo `postalAddressType.json` se deriva del nombre de esquema `postalAddressType`. Por lo tanto, las convenciones de denominación de archivos de esquema están relacionadas con las reglas relativas a las convenciones de denominación JSON establecidas en la presente Norma.

28. Un archivo de esquema puede contener información sobre la versión, que consiste en indicadores de versión mayor y menor, o en indicadores de versión mayor, menor y de parche. La información sobre la versión se aplica únicamente a los esquemas de nivel de documento. Un esquema que se encuentre en fase de proyecto puede revisarse. Los esquemas en fase de proyecto deben indicarse como tales en el nombre de archivo del esquema mediante la letra “D” y el número de revisión. Excepcionalmente, pueden proporcionarse versiones de prueba únicamente cuando exista una necesidad empresarial específica; no se considerarán versiones oficiales y, por tanto, no estarán disponibles públicamente. Una versión de prueba se basará en una versión oficial, pero podrá comprender componentes de esquema de distintas versiones oficiales como medio de transición a una versión oficial. Las versiones de prueba DEBEN indicarse como tales en el nombre de archivo del esquema mediante la letra "T" con el número de revisión junto al número de la versión de base.

[JSD-11] Los caracteres utilizados en los nombres de archivo de esquema DEBEN pertenecer al siguiente conjunto: ‘a-z, A-Z, 0-9, guion bajo “_” y punto “.”’.

[JSD-12] Un 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_V2_0.json`.

[JSD-13] El nombre de archivo de un esquema en fase de proyecto 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_V2_0_D1.json`. Si un esquema en fase de proyecto se basa en un esquema existente y contiene información sobre la versión en el nombre de archivo, los números de versión mayor y menor del nombre de archivo del esquema en fase de proyecto DEBERÍAN ser los mismos que se especifican en el archivo de esquema en que se basa. Si el esquema en fase de proyecto es nuevo, el número de versión mayor del nombre de archivo del esquema en fase de proyecto DEBERÍA ser el mismo que se especifica en el dominio de PI correspondiente y el número de versión menor del archivo de esquema DEBERÍA ser cero “0”.

Estructuración de las propiedades del esquema JSON

29. Los esquemas JSON deberían incluir la propiedad `"type": "object"` para que JSON se utilice únicamente para estructuras anidadas, no para valores individuales. A continuación figura un fragmento extraído del ejemplo `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" ],
  "$defs" : {
    "applicationNumber" : {
      "description" : "Description: Numbers used by IPOs in order to identify each
application received; Version: V2_0",
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType"
    }
  }
}
```

[JSD-14] El objeto en el nivel más externo del esquema DEBE tener una palabra clave `"type"` cuyo valor sea `"object"`.

[JSD-15] El objeto en el nivel más externo del esquema DEBE tener una palabra clave `"$defs"` cuyo valor sea la propiedad de dicho objeto.

[JSD-16] El objeto en el nivel más externo del esquema DEBE tener una palabra clave `"required"` cuyo valor sea una matriz que contenga un único elemento, es decir, la propiedad de dicho objeto.

30. Pueden utilizarse extensiones del esquema JSON (personalizaciones). Si el tipo extendido es un tipo de objeto, debe referenciarse en lugar de duplicar sus propiedades, a fin de promover la reutilización.

[JSD-17] Las extensiones del esquema JSON (personalizaciones) para tipos de objeto DEBEN implementarse mediante una referencia al esquema JSON del tipo extendido.

REGLAS DE DISEÑO DE LAS CONSTRUCCIONES DE ESQUEMA JSON

Información general

31. En esta sección se establecen las reglas aplicables a las construcciones de esquema JSON, en concreto, matrices, objetos y valores primitivos. La normalización de los nombres de las construcciones de esquema es fundamental para el desarrollo de una arquitectura de datos sólida.

Propiedades

32. 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 en la mayor medida posible.

[JSC-02] Varias propiedades que puedan agruparse lógicamente PUEDEN declararse en un único archivo de esquema bajo la definición global.

Definiciones

33. Cada propiedad debería tener una definición global definida en su esquema JSON. Esto permitirá reutilizar el nombre de la propiedad en muchos elementos superiores y disponer de una definición coherente en todos ellos. Véase la propiedad `applicationNumber` en el ejemplo que figura a continuación.

[JSC-03] Cada propiedad enumerada en una palabra clave `properties` DEBERÍA hacer referencia a una definición global definida dentro de la palabra clave `$defs`. Esa 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" : {

      "description" : "Description: Numbers used by IPOs in order to identify each
application received; Version: V2_0"
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",

    }
  }
}
```

34. La definición global de una propiedad debería consistir en el nombre de 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.

35. Las propiedades deben tener tipos. Pueden definirse directamente si son tipos primitivos (excepto `object`), o gestionarse mediante una referencia a una definición global de propiedad en otro esquema JSON.

[JSC-05] Una propiedad DEBE tener un tipo especificado mediante la palabra clave `"type"`, ya sea como propiedad directa o mediante una referencia a una definición global.

Definiciones de tipo

36. Los esquemas JSON pueden definir definiciones de tipo reutilizables a las que se haga referencia desde definiciones globales de propiedades. Estas definiciones de tipo reutilizables deberían constar de una palabra clave `"type"`, una palabra clave `"properties"` (si el tipo es `"object"`) y cualquier otra restricción de valor.

[JSC-06] Un esquema PUEDE definir definiciones globales de tipo para reutilizar modelos de contenido en muchas propiedades.

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" ],
  "$defs" : {
    "applicationNumber" : {
      "description" : "Description: Numbers used by IPOs in order to identify each
application received; Version: V2_0",
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType"
    }
  }
}
```

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

Tipo primitivo JSON

[JSC-08] Para una propiedad DEBERÍA utilizarse el tipo primitivo JSON más específico que sea pertinente. Los tipos primitivos incluyen: "string", "number", "integer", "object", "array", "boolean" y "null".

37. 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 utilizarse los formatos integrados, si procede, por ejemplo "date-time" o "duration".

Listas de códigos

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

[JSC-09] DEBE utilizarse la Norma ST.3 de la OMPI para representar OPI, Estados, otras entidades y organizaciones, así como el país u organización de prioridad y designado.

[JSC-10] DEBEN utilizarse los elementos de código ISO 3166-1-Alpha-2 (códigos de país de dos letras) para la representación de nombres de países con fines de direccionamiento y ciudadanía.

[JSC-11] DEBE utilizarse ISO 639-1 (códigos de idioma de dos letras) para los códigos de idioma.

[JSC-12] DEBE utilizarse ISO 4217-Alpha (códigos de moneda de tres letras) para los códigos de moneda.

[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 (_)}

Matrices

39. 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 los desarrolladores de esquemas no especifiquen valores por defecto para los indicadores de ocurrencia, es decir, "minItems": 0, ya que ello puede sobrecargar innecesariamente el esquema.

[JSC-15] Los esquemas JSON DEBERÍAN utilizar las palabras clave `minItems` y `maxItems` para las matrices, excepto en el caso del valor por defecto de `minItems` (0).

40. 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 secuencia de objetos de tipos diferentes, deberían definirse como propiedades separadas de un objeto.

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

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

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

Objetos

Propiedades "comodín"

42. Los esquemas JSON no deberían permitir que propiedades arbitrarias formen parte de la instancia JSON y que esta siga siendo válida, ya que ello puede corromper la integridad del intercambio de datos.

Se requiere el uso de la palabra clave "`additionalProperties`" y debe establecerse en "`false`". De lo contrario, se permitirán propiedades no definidas en las instancias.

[JSC-18] Un esquema JSON DEBE utilizar "`additionalProperties`" y establecer su valor en "`false`" para cada objeto.

43. No se permite utilizar la palabra clave "`patternProperties`", salvo que el esquema sea de tipo de contenido mixto. Esta palabra clave permite asignar expresiones regulares a esquemas. Por ejemplo, permite definiciones implícitas de esquema basadas en el nombre de la propiedad. Dado un determinado tipo de nombre de propiedad, se aplica un esquema determinado.

[JSC-19] Un esquema JSON NO DEBE utilizar la palabra clave "`patternProperties`", salvo que el esquema sea de tipo de contenido mixto**.**

Orden de las propiedades

44. El esquema JSON no impone un orden determinado a las propiedades de un objeto. No obstante, si un esquema JSON tiene un esquema XML correspondiente, se recomienda que las propiedades figuren en el mismo orden que en el esquema XML, tanto en el esquema JSON como en la instancia.

[JSC-20] Un esquema JSON DEBERÍA utilizar el mismo orden de propiedades que el esquema XML correspondiente, si existe.

IDENTIFICADORES DE ESQUEMA JSON

Información general

45. Un ID en un esquema JSON proporciona un URI que identifica una categoría de información basada en el ámbito empresarial, por ejemplo, empresa, patentes y marcas. Este proyecto de Norma utiliza relaciones de varios a uno entre los pocos ID y quizá cientos de construcciones JSON. Un grupo de construcciones JSON relacionadas con nombres únicos se asocia a un ID determinado. Para la identificación debería utilizarse un identificador uniforme de recursos (URI).

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

REGLAS DE DISEÑO DE INSTANCIAS JSON

46. El esquema JSON define la estructura y las restricciones de la instancia JSON. Para mejorar y garantizar un intercambio de datos sólido, tanto dentro de las oficinas como entre ellas, las instancias JSON deberían asociarse al esquema JSON a fin de asegurar su validez y conformidad.

Orden de las propiedades

47. El esquema JSON no impone un orden determinado a 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 el mismo orden de propiedades que el esquema JSON correspondiente, si existe.

Validación de instancias JSON

48. La validación correcta de las instancias JSON garantiza que su contenido satisface todos los requisitos definidos en los esquemas correspondientes.

[JIN-02] Los documentos de instancia JSON PUEDEN validarse con un esquema correspondiente durante el procesamiento.

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

[JIN-04] Todas las modificaciones, actualizaciones, revisiones y nuevas versiones DEBEN someterse primero a la aprobación del Equipo Técnico sobre API 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 se ajuste a las reglas descritas en la presente Norma.

REFERENCIAS

49. Las siguientes normas y recursos son pertinentes para la presente Norma:

Normas de la OMPI

Norma ST.3 Códigos de dos letras para la representación de Estados, otras entidades y organizaciones;

Norma ST.9 Datos bibliográficos sobre patentes y CCP, y relativos a ellos;

Norma ST.16 Identificación de los distintos tipos de documentos de patente;

Norma ST.26 Presentación de listas de secuencias de nucleótidos y aminoácidos mediante XML;

Norma ST.27 Intercambio de datos sobre la situación jurídica de las patentes;

Norma ST.60 Datos bibliográficos relativos a marcas;

Norma ST.61 Intercambio de datos sobre la situación jurídica de las marcas;

Norma ST.67 Gestión electrónica de los elementos figurativos de las marcas;

Norma ST.68 Gestión electrónica de las marcas sonoras;

Norma ST.69 Gestión electrónica de marcas animadas y multimedia;

Norma ST.80 Datos bibliográficos relativos a diseños industriales;

Norma ST.87 Intercambio de datos sobre la situación jurídica de los diseños industriales;

Norma ST.88 Recomendaciones para la representación electrónica de dibujos y modelos industriales;

Norma ST.90 Procesamiento y comunicación de datos de PI mediante API web;

Norma ST.91 Modelos digitales tridimensionales (3D) e imágenes 3D;

Norma ST.96 Procesamiento de información de propiedad intelectual mediante XML.

Normas y especificaciones del sector

Especificación JSON: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

Esquema JSON Core, versión 2020-12: https://json-schema.org/draft/2020-12/json-schema-core

Esquema JSON Validation, versión 2020-12: https://json-schema.org/draft/2020-12/json-schema-validation

Especificación OpenAPI v3.2.0: https://spec.openapis.org/oas/v3.2.0.html

Norma internacional ISO/IEC 11179-5 “Tecnología de la información, Registros de metadatos (MDR), Parte 5: Principios de denominación e identificación”;

Norma internacional ISO 3166-1 “Códigos para la representación de nombres de países y sus subdivisiones, Códigos de país”;

Norma internacional ISO 639-1 “Códigos para la representación de nombres de lenguas, Parte 1: Código Alpha-2”;
Norma internacional ISO 4217 “Códigos para la representación de monedas y fondos”;

Norma internacional ISO/IEC 10646 “Tecnología de la información, Conjunto universal de caracteres codificados con múltiples octetos (UCS)”;

ISO/IEC 21778:2017 Tecnología de la información, Sintaxis de intercambio de datos JSON;

ECMA-404 – Sintaxis de intercambio de datos JSON;

UBL-2.1 Universal Business Language, versión 2.1;

Norma internacional ISO/IEC 10646 “Tecnología de la información, Conjunto universal de caracteres codificados con múltiples octetos (UCS)”;

Grupo de Trabajo de Ingeniería de Internet (IETF), “RFC 2119: Palabras clave que se utilizan en las RFC para indicar niveles de requisito”.

[Sigue el Anexo I]