# Guía práctica para la mejora de la calidad de datos abiertos

**Enero de 2025**

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA PRIMERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
red.es
Iniciativa aporta
datos.gob.es
reutiliza la información pública

# ÍNDICE DE CONTENIDOS

# 1. Introducción

Los datos abiertos son datos que pueden ser utilizados, modificados y compartidos libremente por cualquier persona para cualquier propósito. En los últimos años, la cantidad y variedad de datos abiertos publicados por las administraciones públicas a nivel mundial -uno de los principales usuarios y productores de datos- ha aumentado de manera tangible, de la misma manera que ha aumentado la voluntad política de apertura y con ello las normativas, hojas de ruta y directrices técnicas. La disponibilidad de la información del sector público como datos abiertos puede aportar un considerable valor añadido y satisfacer una demanda creciente que proviene de empresas, organizaciones gubernamentales, desarrolladores y de la sociedad en su conjunto. No obstante, la apertura, desde el punto de vista normativo y técnico, no es suficiente para crear un ecosistema de reutilización de datos próspero puesto que los fallos en la disponibilidad y la calidad de los datos pueden perjudicar, no solo a la reutilización de los datos, sino también a la credibilidad de las instituciones que los publican.

La calidad de los datos es una de las principales trabas para el sector reutilizador y es un aspecto fundamental para garantizar su reutilización y respaldar el ecosistema de la creación de servicios y aplicaciones. La baja calidad de los datos abiertos obstaculiza una reutilización eficiente de los mismos debido a la necesidad de invertir recursos, por parte de los usuarios, en la comprensión de conjuntos de datos mal documentados y la realización adicional de tareas de depuración y procesamiento. Para incentivar la reutilización de los datos abiertos, es necesario que las administraciones inviertan en la mejora de su calidad.

Esta guía se orienta a publicadores de datos y se presenta como un compendio de **directrices para mejorar la calidad de datos** actuando directamente sobre cada una de las características que la definen. Por otro lado, la recopilación de estas pautas pretende orientar a los reutilizadores de datos sobre cómo afrontar las debilidades de calidad que pueden presentar los conjuntos de datos con los que trabajar.

En esta guía se utiliza el término conjunto de datos o dataset indistintamente, asociado al concepto de colección de datos que poseen vínculos entre sí de acuerdo con alguna estructura, esquema o modelo de representación. La disponibilidad de datasets para su reutilización se puede realizar mediante la descarga de archivos o accediendo a servicios de datos (APIs). Los archivos de datos descargables constituyen la materialización completa de un dataset. En cambio, el acceso a datos mediante servicios permite obtener múltiples datasets en función de las consultas que el reutilizador configure. Igualmente, se denominan distribuciones a la forma en la que se representan los datasets es decir, los formatos. Estos son diversos y responden a las necesidades del reutilizador: puede ser una hoja de cálculo CSV o Excel, un archivo XML, un archivo de imagen, datos vinculados en RDF, etc. La disponibilidad de formatos alternativos de cada dataset facilita la reutilización.

La estructura del documento comienza con una introducción a la calidad de datos, la definición de sus características según las diferentes normas técnicas de referencia internacionales, para centrar posteriormente el cuerpo del documento en una descripción de pautas generales y específicas de los formatos de datos abiertos más habituales, que detallan los problemas que es necesario afrontar, las características de calidad afectadas y las recomendaciones para su resolución con ejemplos prácticos que ayudarán a entender cada caso presentado. El documento cierra con dos aspectos importantes que

contribuyen a mejorar la calidad general como son la estandarización, el enriquecimiento y la documentación de datos abiertos.

Este documento toma como referente la guía para la calidad de datos de data.europe.eu, publicada en 2021 por la Oficina de Publicaciones de la Unión Europea[1]. De dicha guía se han recopilado y adaptado a la estructura del documento las pautas y los ejemplos más relevantes para alcanzar un buen nivel de calidad de datos abiertos. Otros ejemplos utilizados se han adaptado del dataset "Auto MGP Dataset" disponible en el popular *UCI Machine Learning Repository* de la Universidad de California.

[1] Publications Office, *Data.europa.eu data quality guidelines*, Publications Office, 2021, https://data.europa.eu/doi/10.2830/79367

# 2. Definición y requisitos mínimos de la calidad de los datos abiertos

## 2.1. ¿Qué es la calidad de los datos abiertos?

Existen múltiples definiciones para la calidad de datos, pero la más extendida la define como la idoneidad de un conjunto de datos para servir a su propósito específico. Esta definición implica que la calidad de los datos debe ser gestionada en base al cumplimiento de unos requisitos determinados por las características que los definen. La gestión de la calidad de los datos, además, debe proporcionar métodos y herramientas para evaluar y establecer procesos de mejora siempre que sean precisos.

A lo largo de estos últimos años, la mayoría de las iniciativas de datos abiertos aplican métodos y herramientas para evaluar y establecer procesos de mejora de la calidad de sus datos. Muchas de estas iniciativas han utilizado el método de 5 estrellas para los datos abiertos publicado por Tim Berners-Lee en 20061. Aunque este esquema es ampliamente utilizado para medir la calidad de los datos, sólo cubre un aspecto específico de la calidad, la codificación utilizada para publicar los datos, por lo que un conjunto de datos publicado puede alcanzar el nivel de 5 estrellas, pero al mismo tiempo mostrar una calidad deficiente ya que puede presentar otros tipos de errores, como errores de sintaxis, duplicidad de registros o datos obsoletos, entre otros que se irán revisando en esta guía práctica.

Igualmente es común identificar iniciativas que ligan la calidad de los datos con los principios que debe regir toda política de apertura de datos. Estos principios nos indican que los datos deben ser completos, primarios, actuales accesibles, procesables por maquinas, no discriminatorios, no propietarios y sin restricciones de utilización. Pero estos principios, aunque guardan relación, no ponen el foco en la calidad de los datos, sino en una serie de propiedades que deben poseer los conjuntos de datos para considerarlos abiertos y reutilizables.

En la literatura existen diferentes enfoques sobre qué características deben tener los datos de calidad y esto conlleva a que existan diferentes modelos de medición y aseguramiento. Algunas referencias que establecen características imprescindibles para conseguir una alta calidad en los datos se citan a continuación.

- Los principios definidos por la carta internacional de los Datos Abiertos (Open Data Charter) establecen ya una serie de cualidades que los datos de calidad deben cumplir, tales como completitud, exhaustividad, puntualidad, oportunidad, comparabilidad e interoperabilidad. Sin embargo, hay otros aspectos que también definen la calidad de los datos, y deben tenerse en cuenta a la hora de producir y valorar cualquier tipo de datos.

- La OCDE, en su informe "Marco de calidad y directrices para las actividades estadísticas de la OCDE" publicado en 2011, considera la calidad en términos de 7 dimensiones: pertinencia, precisión, credibilidad, actualidad, accesibilidad, interpretabilidad y coherencia, y además, establece que está ampliamente vinculada con las perspectivas, necesidades y prioridades de los reutilizadores.

- Por otro lado, la norma ISO/IEC 25012 también establece un modelo de Calidad del Producto de Datos compuesto por 15 características, clasificadas en dos grandes categorías, 12 de ellas relacionadas directamente con los datos como producto: exactitud, completitud, consistencia, credibilidad,

actualidad, accesibilidad, conformidad, confidencialidad, eficiencia, precisión, trazabilidad y comprensibilidad.

Otro enfoque distinto, aunque en línea con la calidad que deben presentar los datos, es la propuesta que, en 2016, publica la revista Scientific Data de Nature, sobre los "Principios FAIR para la gestión y administración de datos científicos". Los principios FAIR, son un conjunto de directrices precisas y medibles que debe seguir cualquier científico para la publicación de sus datos con la mayor calidad posible. Estos principios hacen hincapié en la capacidad de acción de los sistemas informáticos para encontrar, acceder, interoperar y reutilizar los datos con la mínima intervención humana, ya que los seres humanos dependen cada vez más del apoyo informático para tratar los datos como resultado del aumento del volumen, la complejidad y la velocidad de creación de los datos. En 2018, se creó la comunidad GO FAIR con el fin de ayudar e implementar en la comunidad científica los principios FAIR.

| Principio | Descripción |
|---|---|
| **Encontrables (Findable)** | Los datos y metadatos pueden ser encontrados por la comunidad de manera sencilla después de su publicación, mediante herramientas de búsqueda.<br>• Asignar un identificador único y persistente.<br>• Describir los datos con metadatos de manera prolija.<br>• Indexar los datos y metadatos en el recurso de búsqueda.<br>• En los metadatos se debe especificar el identificador de los datos que se describen. |
| **Accesibles (Accessible)** | Los datos y metadatos están accesibles y por ello pueden ser descargados por otros investigadores utilizando sus identificadores.<br>• Los datos y metadatos son recuperables por su identificador utilizando protocolos de comunicación estandarizados.<br>  ○ El protocolo es abierto, gratuito y de aplicación universal.<br>  ○ El protocolo permite un procedimiento de autenticación y autorización cuando sea necesario.<br>• Los metadatos son accesibles, incluso cuando los datos ya no están disponibles. |

| Principio | Descripción |
|---|---|
| **Interoperables (Interoperable)** | Los datos deben poder integrarse con otros datos. Además, los datos deben poder interoperar con aplicaciones o flujos de trabajo para su análisis, almacenamiento y procesamiento.<br>• Los datos y metadatos deben utilizar un lenguaje formar, accesible, compartido y ampliamente aplicable.<br>• Los datos y metadatos utilizan vocabularios que siguen los principios FAIR.<br>• Los datos y metadatos incluyen referencias cualificadas a otros datos o metadatos. |
| **Reutilizables (Reusable)** | Los datos y metadatos deben estar bien descritos para que puedan ser replicados y/o combinados en diferentes entornos.<br>• Los datos y metadatos están bien descritos con una pluralidad de atributos precisos y relevantes.<br>• Los datos y metadatos están asociados a una procedencia detallada.<br>• Los datos y metadatos cumplen con las normas de la comunidad pertinentes para el sector. |

*Figura 1. Principios FAIR para la gestión y administración de datos*

## 2.2. Características de calidad de datos

Los referentes mencionados anteriormente definen un amplio conjunto de características comunes que se espera que cumplan los conjuntos de datos considerados de alta calidad


*Figura 2. Características de calidad de los datos según la norma ISO 25012*

A continuación, se describe someramente cada una de las características:

- **Exactitud/Precisión**, aunque no son términos equivalentes se refieren a la veracidad que proporcionan los datos. Los datos que presentan esta característica representan correctamente el valor verdadero del atributo al cual simbolizan en el mundo real. Además, las mediciones que son precisas son consistentes y replicables.
- **Completitud**: los datos se consideran completos cuando está disponible toda la información requerida para un atributo. Los datos deben presentar un nivel de detalle y una desagregación adecuada para ser relevantes y reutilizables.
- **Consistencia/Coherencia**, los datos deben estar libres de contradicciones y tener coherencia lógica en un contexto específico, por ejemplo, de formato o temporal.
- **Credibilidad** tanto para los datos en sí como para la fuente de información. Los datos deben ser objetivos, deben estar publicados con los estándares estadísticos apropiados y las prácticas y políticas para su recogida y publicación deben ser transparentes. La credibilidad, también incluye el concepto de autenticidad (la veracidad de los orígenes de datos, atribuciones y compromisos).
- **Actualidad y actualización/Puntualidad**, los datos deben estar disponibles a tiempo y sin retrasos que afecten a su relevancia y se actualizarán regularmente, manteniendo así su valor.
- **Accesibilidad**, referida a la facilidad de acceso a los datos. Los datos deben estar disponibles para la más amplia gama de usuarios y propósitos.
- **Conformidad**, los datos se adhieren a estándares o normativas vigentes.
- **Confidencialidad**, los datos se deben publicar respetando la privacidad y seguridad de estos. En contextos específicos, los datos sólo serán accedidos e interpretados por usuarios autorizados. La confidencialidad es un aspecto fundamental de la seguridad de la información.
- **Eficiencia**, los datos tienen atributos que puede ser procesados y proporcionados con unos recursos razonables.
- **Trazabilidad** respecto a la fuente u origen de los datos. Los datos tienen atributos que proporcionan un histórico del camino de acceso auditado a los datos o cualquier otro cambio realizado sobre ellos.
- **Comprensibilidad /Interpretabilidad** los datos pueden ser interpretados y leídos por los usuarios y ser expresados utilizando lenguajes, símbolos y unidades coherentes con el contexto de los datos. Cierta información sobre la comprensibilidad puede ser expresada mediante metadatos.

A lo largo de esta guía veremos cómo los diferentes problemas que se relatan afectan a una o más de las características indicadas.

## 2.3. ¿Por qué es importante disponer datos de la mayor calidad posible?

Generalmente, la solución de los problemas de calidad de los datos en conjuntos de datos abiertos implica una inversión significativa, a veces con rendimientos decrecientes, tanto por parte de los

publicadores como por parte de los usuarios. Ya la OCDE en el "Marco de calidad y directrices para las actividades estadísticas de la OCDE" mencionado con anterioridad, determina que la rentabilidad (costes + beneficios) es un factor que se debe tener en cuenta en cualquier análisis de calidad, ya que puede afectar a todas las características antes mencionadas.

Haug et al., en 2011, llevaron a cabo un análisis de las causas y una estimación de los costes asociados a la deficiente calidad de los datos. Los autores afirman que, en la práctica, la baja calidad de los datos puede implicar perjuicios económicos a la organización que publica los datos de múltiples formas. Además, determinan que las organizaciones tienden a sobreestimar la calidad de sus datos e infravalorar el coste generado por los errores derivados. En numerosas ocasiones, los publicadores de datos gastan más dinero en solucionar errores derivados de datos deficientes que en evitar con antelación los potenciales problemas que su uso a largo plazo puede generar en los reutilizadores.

Este análisis sostiene que el nivel óptimo de mantenimiento de calidad de los datos no es lograr datos perfectos, si no alcanzar un equilibrio entre el coste de las tareas asociadas con el aseguramiento de la calidad de los datos y el ahorro del coste causado por datos de baja calidad. Además, los autores, clasifican los tipos de costes como:

- **Costes ocasionados por la deficiente Calidad de los Datos ocasionados por**:
  - Costes directos asociados a la verificación, re-entrada y compensación de datos
  - Costes indirectos derivados de decisiones o acciones incorrectas y pérdida de inversiones.
- Costes en mejora o de aseguramiento de la Calidad de los Datos derivados de:
  - Costes de **prevención** asociados a la formación, monitorización, desarrollo e implementación estándar.
  - Costes de **detección** asociados a la creación de análisis e informes asociados a los datos
  - Costes de **reparación** asociados a la planificación de reparaciones y la reparación de implementaciones.

En el concepto de los datos abiertos es importante tener en cuenta que los costes derivados de la mala calidad de los datos, trasciende a la repercusión que tiene sobre la propia organización que los gestiona y en primera instancia utiliza, dado que se trasladan al sector reutilizador, multiplicando así el efecto negativo que conlleva.

Por último, aunque no es el objetivo de esta guía, es esencial que los conjuntos de datos estén acompañados de unos metadatos de excelente calidad, ya que es el primer contacto que el usuario tiene con el conjunto de datos. Para llevar a cabo una reutilización efectiva de los datos, es imprescindible que estén acompañados de una documentación que ayude a los reutilizadores a comprender el contenido de los datasets y cómo usarlos, incluidos los defectos ya detectados. Los metadatos forman parte del conjunto de datos y proporcionan información crucial sobre el origen de los datos, el grado de actualización, las restricciones legales sobre su uso y otra información relevante. Además, los metadatos deberían de informar sobre la calidad que presentan los datos. Unos metadatos deficientes disminuyen la probabilidad de reutilización de los datos.

# 3. Pautas generales para garantizar la calidad de los datos abiertos

La insuficiente calidad de los datos afecta negativamente al proceso de reutilización de estos, dificultando el objetivo prioritario que persiguen las Iniciativas de Datos Abiertos, que es lograr la máxima reutilización y generación de valor a partir de datos públicos.

Algunas de las causas de los problemas de calidad de datos más frecuentes se detallan a continuación.

- Utilizar formatos de datos no procesables
- Utilizar codificación de caracteres no estandarizada
- Nombrar inadecuadamente columnas
- Publicar datos incompletos o con valores ausentes
- Incluir registros duplicados
- Falta de estandarización de valores de datos
- Proporcionar una cantidad inadecuada de datos para facilitar su análisis
- Formato inadecuado de variables de fecha y hora
- Formato inadecuado de tipos de datos numéricos
- Mezcla de escalas numéricas
- Mezcla de rangos en un mismo conjunto de datos
- No incorporar variables con información geográfica
- Incorporar subtotales, totales o agrupamientos
- Dificultades para el acceso a los datos
- Organización inadecuada de datasets disponibles

Repasamos estos errores comunes y veamos porqué se producen, así como ciertas recomendaciones y buenas prácticas para evitar cada problema.

## 3.1. Evitar formatos de datos no procesables

**Descripción de problema**: Es frecuente distribuir información pública en forma de documentos de todo tipo cuyo contenido es predominantemente texto y normalmente en formato PDF. En numerosos casos, estos informes están basados en análisis de datos que los publicadores gestionan. Sin embargo, muy pocas veces se publican estos datos de manera complementaria al informe en un formato abierto y procesable, lo que hace muy difícil que esos datos puedan ser reutilizados por cualquier usuario.

En otras ocasiones se publican datos en sitios web públicos mediante el uso de mapas, gráficos o tablas, pero no se incluye la URL de acceso para su descarga, lo que hace inservibles esos datos para una eficiente reutilización.

Por otra parte, también es frecuente que se utilicen formatos no abiertos o simplemente poco comunes para distribuir la información sin ofrecer alternativas a los usuarios, lo que en la práctica reduce las posibilidades de reutilización.

Finalmente, se pueden encontrar errores físicos en los propios archivos fruto de una codificación de caracteres inadecuada, o una compresión incorrecta de archivos, entre otras circunstancias. Esto impide la apertura y procesamiento automático de los mismos, lo que redunda nuevamente en la reducción de la calidad y con ello, de la reutilización de esos datos.

**Características de calidad afectadas**: Accesibilidad/disponibilidad, Legibilidad por maquinas/Procesabilidad, Apertura.

**Recomendaciones**: Cualquier informe, mapa, gráfico, infografía, tabla, o cualquier otra representación visual elaborada a partir de datos, **debe ir siempre acompañada de la serie de archivos en formato abierto y reutilizable que faciliten el acceso a los datos** en los que se basan o a los que hacen referencia en dicha representación

Los publicadores **deben proporcionar los datos en un número razonable de formatos alternativos**. Es recomendable, además, dar preferencia a aquellos formatos con mayor nivel de compatibilidad (por ejemplo, CSV frente a XLS), pero sin relegar a otros formatos populares entre los usuarios o adecuados para determinados tipos de datos, por ejemplo, el formato SHP para datos espaciales.

Así mismo, **siempre que se dispongan datos de alto valor, dinámicos o con alta frecuencia de actualización, además de la posibilidad de descarga masiva, cuando proceda, se deben proporcionar a través de interfaces de programación de aplicaciones (APIs) o de puntos de consulta SPARQL**. Con ello, los publicadores conseguirán abrir los datos a un rango más amplio de reutilizadores facilitando, además, la utilización de éstos por maquinas u otras aplicaciones implementando así soluciones basadas en datos más sofisticadas.

## 3.2. Utilizar una codificación de caracteres estandarizada

**Descripción del problema**: Es relativamente frecuente encontrar datasets cuyos valores contienen caracteres especiales, como acentos, eñes, exclamaciones, entre otros, que son interpretados de manera diferente por las maquinas, lo cual termina haciendo difícilmente legibles los conjuntos de datos, lo que en la práctica reduce las posibilidades de reutilización o aumenta el esfuerzo a la hora de hacer interpretable el conjunto de datos.

**Características de calidad afectadas**: Interoperabilidad, Conformidad/Cumplimiento

**Recomendaciones**: Para que los caracteres se muestren correctamente y se garantice la mayor compatibilidad posible con las aplicaciones que procesan datos, se puedan reutilizar y mezclar con otros datos de fuentes internacionales y evitar problemas durante el procesamiento por máquinas, es

recomendable **utilizar una codificación de caracteres internacionalmente reconocida, estandarizada y ampliamente utilizada. Normalmente la codificación elegida es** UTF-8, que es una codificación de caracteres Unicode y un estándar internacional para la representación de todos los caracteres significativos, ya sean del alfabeto latino o japonés. Sin embargo, en general, se debe evitar el uso de caracteres especiales en los conjuntos de datos, incluso aunque formen parte de la codificación UTF-8, asegurando la compatibilidad con sistemas más antiguos.

Si se utiliza un conjunto de caracteres diferente a UTF-8 en el conjunto de datos, es esencial especificarlo en los metadatos.

Existen herramientas como UTF-8 Tools o CSVLint para ayudar a los publicadores en la validación de la codificación de datos.

**Ejemplo:** En la tabla de la izquierda (en formato CSV), se observa un conjunto de datos que no utiliza UTF-8, lo que ocasiona que aparezcan símbolos no entendibles por el usuario. La tabla derecha muestra los mismos valores codificados como UTF-8, legibles tanto para el usuario como para la máquina.

| identificador,nombre |
|---|
| 1,Universidad Europea CEES |
| 2,Universidad PolitÃ©cnica de Madrid |
| 3,Universidad Carlos III |
| 4,Universidad AutÃ³noma de Madrid |
| 5,Universidad Pablo de Olavide |
| 6,Universidad Alfonso X El Sabio |
| 7,Universidad Antonio Nebrija |
| 8,Universidad Internacional MenÃ©ndez Pelayo |
| 9,Universidad de Cantabria |
| 10,Universidad de AlmerÃ-a |

| identificador,nombre |
|---|
| 1,Universidad Europea CEES |
| 2,Universidad Politécnica de Madrid |
| 3,Universidad Carlos III |
| 4,Universidad Autónoma de Madrid |
| 5,Universidad Pablo de Olavide |
| 6,Universidad Alfonso X El Sabio |
| 7,Universidad Antonio Nebrija |
| 8,Universidad Internacional Menéndez Pelayo |
| 9,Universidad de Cantabria |
| 10,Universidad de Almería |

## 3.3. Nombrar adecuadamente columnas

**Descripción de problema**: Los nombres de las columnas o variables de un conjunto de datos deben ser perfectamente interpretables por las personas. En numerosas ocasiones los nombres de las columnas de los conjuntos de datos mantienen los nombres asignados en los sistemas de información de los cuales se extraen, haciendo que sean difícilmente comprensibles para las personas.

Puede ocurrir, además, que las distribuciones de los conjuntos de datos contienen diferentes nombres en columnas que representan los mismos datos, afectando de esta manera la consistencia del conjunto de datos.

**Características de calidad afectadas**: Reusabilidad, Consistencia, Precisión, Comprensibilidad

**Recomendaciones**: Es aconsejable que los nombres de los campos se mantengan a lo largo de todas las distribuciones del conjunto de datos. Los nombres deben ser cortos, pero siempre teniendo en cuanta que el ahorro de caracteres no debe inducir a errores. Los nombres de los campos y sus

especificaciones deben estar recogidos en el diccionario de datos que documenta el dataset. Además, se deben utilizar solo caracteres ASCII en minúsculas, no se deben usarse caracteres especiales, tildes o signos de puntuación y los espacios deben ser sustituidos por guiones.

**Ejemplo:** nombrado comprensible de columnas

| Identificador-M | Annio | Cil. | Consumo-por-cada-100-Kms-de-recorrido-urbano | HP | m/seg^2 |
|---|---|---|---|---|---|
| **chevrolet chevelle malibu** | 1970 | 8 | 18 | 130 | 12 |
| **buick skylark 320** | 1970 | 8 | 18 | 165 | 11.5 |
| **plymouth satellite** | 1970 | 8 | 18 | 150 | 11 |

| marca | año | cilindros | consumo | potencia | aceleración |
|---|---|---|---|---|---|
| **chevrolet chevelle malibu** | 1970 | 8 | 18 | 130 | 12 |
| **buick skylark 320** | 1970 | 8 | 15 | 165 | 11.5 |
| **plymouth satellite** | 1970 | 8 | 18 | 150 | 11 |

## 3.4. Publicar datos completos y evitar valores ausentes

**Descripción del problema:** La ausencia de valores en tablas es una problemática habitual en muchos conjuntos de datos que afecta directamente a la calidad de los datos. En numerosas ocasiones se debe a fallos en la trascripción de los datos o problemas durante la recogida de estos, debido, por ejemplo, a la imposibilidad para obtener cierta medida u observación. Esta circunstancia ocasiona importantes problemas de reutilización, ya que puede generar resultados erróneos o visualizaciones de datos engañosas, entre otros.

**Características de calidad afectadas:** Accesibilidad/disponibilidad, Completitud.

**Recomendaciones:** Una entidad que recoge y publica datos reutilizables debe hacerlo acorde a un plan de recogida y publicación. En este documento se deben especificar las variables que se incorporarán al conjunto de datos, garantizando que sus valores serán recogidos y publicados en su totalidad.

Si finalmente por algún motivo justificado, existe ausencia de datos en el conjunto, es necesario que el publicador especifique en los metadatos la razón por la cual esos datos no se encuentran presentes,

ayudando al usuario en las decisiones sobre la reutilización de tales datos. Además, para evitar confusiones en su tratamiento, el publicador debe marcar claramente los valores ausentes como valores nulos. De esta manera, los usuarios que no estén familiarizados con los datos puedan identificar que valores faltan.

Hay varias formas de indicar un valor nulo, por ejemplo, marcando el valor que falta con "NULL" o "NA" y siempre de forma consistente, es decir, todos los datos ausentes de la misma manera. Es importante, además, si se observa que una fila o columna tiene un alto porcentaje de valores nulos, valorar la posibilidad de eliminar dicha fila o columna, ya que probablemente no aporte ninguna información significativa.

**Ejemplo 1:** Ventas de coches por año (en miles). En la primera tabla, los valores ausentes se indican de manera inconsistente utilizando campos vacíos, "N/A" o "null", indistintamente. En cambio, la tabla siguiente, muestra los mismos datos, pero los valores ausentes están marcados como nulos utilizando siempre la misma codificación (NA).

| marca | año | consumo | ventas |
|---|---|---|---|
| **Chevrolet chevelle malibu** | 1998 | Alto | 2.50 |
| **Chevrolet chevelle malibu** | 1999 | Bajo | 2.63 |
| **Chevrolet chevelle malibu** | 2000 | Medio | |
| **Buick skylark 320** | 1998 | | 3.40 |
| **Buick skylark 320** | 1999 | Medio | 3.57 |
| **Buick skylark 320** | 2000 | Medio | N/A |
| **Plymouth satellite** | 1998 | | 2.40 |
| **Plymouth satellite** | 1999 | null | 2.52 |
| **Plymouth satellite** | 2000 | Alto | 3.60 |

| marca | año | consumo | ventas |
|---|---|---|---|
| **Chevrolet chevelle malibu** | 1998 | Alto | 2.50 |
| **Chevrolet chevelle malibu** | 1999 | Bajo | 2.63 |
| **Chevrolet chevelle malibu** | 2000 | Medio | NA |
| **Buick skylark 320** | 1998 | NA | 3.40 |
| **Buick skylark 320** | 1999 | Medio | 3.57 |

| marca | año | consumo | ventas |
|---|---|---|---|
| **Buick skylark 320** | 2000 | Medio | NA |
| **Plymouth satellite** | 1998 | NA | 2.40 |
| **Plymouth satellite** | 1999 | NA | 2.52 |
| **Plymouth satellite** | 2000 | Alto | 3.60 |

**Ejemplo 2:** el siguiente ejemplo, muestra un CSV con el número de visitantes a lo largo de los años, en la tabla de la izquierda, los valores ausentes se indican como campos vacíos. A la derecha, muestra los mismos datos, pero los valores ausentes están marcados como tal (null).

| year; visitors; viewing-time |
|---|
| 2014; 768954;00:03:18 |
| 2013;;00:02:59 |
| 2013;822101;00:02:59 |
| 2011;707402; |
| 2010;707402;00:03:50 |

| year; visitors; viewing-time |
|---|
| 2014; 768954;00:03:18 |
| 2013;null;00:02:59 |
| 2012;792967;00:02:52 |
| 2011;707402;null |
| 2010;707402;00:03:50 |

## 3.5. Evitar la duplicidad de registros

**Descripción del problema:** Se da cuando en un conjunto de datos existen dos o más registros idénticos. Los datos duplicados aumentan la probabilidad de que los resultados obtenidos a partir de su análisis estén sesgados, lo que limita su utilidad. Este caso puede ocurrir cuando se lleva a cabo una integración desde múltiples fuentes de datos con origen en diferentes departamentos u organizaciones, debido a diferencias en el formato de registro, existencia de errores tipográficos o carencia de estandarización.

**Características de calidad afectadas:** Reusabilidad, Consistencia.

**Recomendaciones**: La duplicidad de los datos es un problema grave y en ocasiones difícil de detectar, que se debe atajar para incrementar la calidad de los datos y garantizar su confiabilidad. Para reducir el número de duplicados es aconsejable estandarizar la recogida de datos y almacenamiento de los mismos, centralizando el proceso en un único sistema de información, de tal forma que sean fácilmente detectables y puedan ser eliminados automáticamente.

**Ejemplo:** La tabla de la izquierda muestra un archivo CSV con determinados registros duplicados. A la derecha se muestran los mis datos con todas sus filas distintas.

| year; visitors; viewing-time |
|---|
| 2014; 768954;00:03:18 |
| 2013;822101;00:02:59 |
| 2013;822101;00:02:59 |
| 2011;707402;00:03:44 |
| 2010;707402;00:03:50 |
| 2010;707402;00:03:50 |

| year; visitors; viewing-time |
|---|
| 2014; 768954;00:03:18 |
| 2013;822101;00:02:59 |
| 2012;792967;00:02:52 |
| 2011;707402;00:03:44 |
| 2010;707402;00:03:50 |
| 2009;429430;00:03:16 |

## 3.6. Estandarizar valores de datos

**Descripción de problema:** En muchas ocasiones, los valores de las variables de los conjuntos de datos no están estandarizados. Esta problemática dificulta de forma significativa la correlación entre datos de diferentes distribuciones o datasets, la interoperabilidad y el enlazado de datos e incluso, la simple comparación de datos entre y dentro de las organizaciones.

**Características de calidad afectadas:** Conformidad, Legibilidad/Procesabilidad por maquinas.

**Recomendaciones:** Para normalizar la estructura y los valores de los campos es recomendable, siempre que sea posible, el uso de vocabularios de referencia. La Oficina de Publicaciones de la Unión Europea proporciona diferentes vocabularios para su aplicación en el ámbito del sector público. Otras referencias útiles son los vocabularios aplicables en contexto de ciudades abiertas o para un propósito más general, los relacionados en el sitio web Linked Open Vocabularies (LOV).

En el caso de no usarlos, el valor que se asigne a un determinado atributo debe ser único y consistente en toda utilización de dicho valor en los datasets que lo incluyan. Es decir, si se opta por usar el valor "Castilla y León", para referirse a esa región, no se debe usar el valor "Comunidad Autónoma de Castilla y León" o cualquier otro similar.

La norma AENOR 137801:2015, Ciudades Inteligentes, Datos Abiertos, considera datos técnicamente correctos aquellos que utilizan la misma codificación y normalización para el mismo tipo de dato publicado en los diferentes conjuntos de datos de un catálogo y que la codificación y normalización utilizada se base en algún estándar común y utilizado por otras organizaciones de codificación (por ejemplo estándares aprobados por EUROSTAT o el INE), entre otras características.

Ejemplo 1: En la tabla superior no se utiliza ninguna codificación estandarizada. Sin embargo, en la tabla inferior se está utilizando la nomenclatura estadística de actividades económicas de la Comunidad Europea de EUROSTAT para la estandarización de las actividades económicas de los vendedores de vehículos.

| marca | actividad-vendedor |
|---|---|
| **chevrolet** | Venta de coche |
| **buick** | Venta de vehículos |
| **plymouth** | Venta |

| marca | codigo-vendedor | actividad-vendedor |
|---|---|---|
| **chevrolet** | 45.11 | venta de automóviles y vehículos de motor ligeros |
| **buick** | 45.11 | venta de automóviles y vehículos de motor ligeros |
| **plymouth** | 15.19 | venta de otros vehículos de motor |

**Ejemplo 2:** a continuación, se muestra el metadato que define las condiciones de uso de un dataset utilizando DCAT que toma el valor de una cadena de caracteres. Esta forma de presentación está dificultando el procesamiento posterior a la vez que está expuesto a errores ortográficos. En cambio, la misma codificación se puede realizar utilizando el vocabulario estandarizado por la Unión Europea para describir condiciones de reutilización de conjuntos de datos.

```
<dcterms:license rdf:resource=”http://CCC_BY_4_0”/>
```

```
<dcterms:license rdf:resource=http://publications.europa.eu/resource/authority/licence/CC_BY_4_0/>
```

Volveremos a tratar esta pauta en la sección recomendaciones para la estandarización y enriquecimiento de datos de este documento.

## 3.7. Proporcionar una cantidad adecuada de datos para facilitar su análisis

**Descripción del problema:** Es bastante frecuente que existan datos publicados en abierto, y que, al mismo tiempo, estos datos estén tan limitados o, por el contrario, sea una cantidad tan excesiva porque pueda exceder significativamente el ámbito del dataset, que no tenga sentido considerar que dicha información sea reutilizable por parte del usuario o aporte valor tal como está disponible.

**Características de calidad afectadas:** Reusabilidad, Relevancia, Completitud.

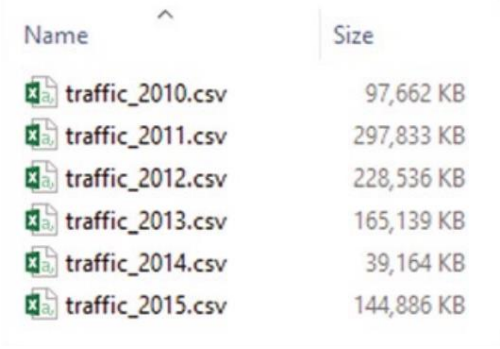
**Recomendaciones:** Puede resultar obvio que hay que disponer la cantidad apropiada de datos para hacer factible su reutilización. Dependiendo de los datos que se publiquen, el significado del término “apropiado” puede ser muy diferente. Es importante publicar todos los datos que resulten pertinentes considerando su utilidad. Los publicadores deben asegurar que se publica una cantidad razonable de datos para que haya suficiente contexto y los usuarios puedan obtener valor de su explotación. Resulta bastante inútil para los reutilizadores descargar un archivo CSV con solo cuatro filas de datos, por ejemplo.

Sin embargo, tampoco hay una indicación clara sobre cuál debe ser la cantidad adecuada de datos, ya que depende en gran medida del objetivo que tenga el usuario. Para encontrar un buen equilibrio, el publicador, aunque a priori no conoce el potencial uso de los datos, debe analizar si todos los datos

que va a publicar se ciñen al dominio de conocimiento que se representa en el dataset y con ello se aporta valor al reutilizador, reduciéndolos si cree que son excesivos o por el contrario insuficientes y, por tanto, incrementar la cantidad para añadir contexto y maximizar su valor.

Otra perspectiva en esta línea está relacionada con la forma en la que se organizan los distintos conjuntos de datos disponibles. Nos referimos a ello en una pauta específica dedicada a la organización adecuada de conjuntos de datos más adelante en este documento.

**Ejemplo:** se observan dos imágenes que muestran archivos de datos con diferentes resoluciones temporales. Una solución aceptable para el caso que se presenta es la disponibilidad de ambas opciones, es decir, la posibilidad de descarga de los archivos con la resolución temporal mínima (anualidades) y el agregado con todos los años.

| | |
|---|---|
|  | El archivo contiene datos de tráfico ficticios agregados durante el transcurso de 6 años. Inconveniente, si los usuarios solo están interesados en un año determinado tienen que descargar y procesar el archivo que los integra. |
|  | En contraste a la imagen anterior, se muestran los mismos datos, pero divididos por anualidades. En este caso, los usuarios pueden descargar únicamente los archivos de datos que necesiten. |

## 3.8. Formateo de variables de fecha y hora

**Descripción de problema:** Los datos refieren con frecuencia fechas y horas. Dependiendo del contexto regional existen diferentes formas de indicar las fechas, lo que puede llevar a confusión a la hora de interpretar los datos.

**Características de calidad afectadas:** Interoperabilidad, Conformidad/Cumplimiento, Legibilidad por maquinas/Procesabilidad.

**Recomendaciones:** Las fechas deben codificarse siempre utilizando el estándar internacional de referencia ISO 8601, que codifica los valores de fecha con el formato AAAA-MM-DD, en su forma abreviada y AAAA-MM-DDThh:mm:ss, en su versión extendida (la letra "T" debe aparecer literalmente en la cadena, para indicar el comienzo del elemento de tiempo, como se especifica en ISO 8601). Además, si procede deberá especificarse el huso horario utilizado, que se toma siempre la diferencia

temporal respecto al Tiempo Universal Coordinado (UTC). Existen herramientas como DenCode para ayudar a los publicadores en la conversión de datos tipo fecha al formato ISO 8601.

**Ejemplo 1:** En el ejemplo siguiente observamos datos relativos al tiempo medio anual de visita a una determinada página web. En la tabla de la izquierda, el formato de la hora no sigue un esquema consistente, lo que hace muy difícil procesarlo correctamente. En cambio, en la tabla de la derecha, se muestran los mismos datos con todos los valores de tiempo formateadas de manera uniforme utilizando la codificación ISO 8601.

| year; visitors; viewing-time |
|---|
| 2014;768954;**3:18** |
| 2013;822101;**00:02:59** |
| 2012;792967;**0:02:52** |
| 2011;707402;**03:44** |
| 2010;707402;**3m:50s** |
| 2009;429230;**3:16** |

| year; visitors; viewing-time |
|---|
| 2014; 768954;**00:03:18** |
| 2013;822101;**00:02:59** |
| 2012;792967;**00:02:52** |
| 2011;721519;**00:03:44** |
| 2010;707402;**00:03:50** |
| 2009;429430;**00:03:16** |

**Ejemplo 2:** En este ejemplo se observan las fechas de inicio y fin de un determinado periodo temporal. En la tabla de la izquierda, la fecha de fin puede dar lugar a errores debido a la ambigüedad que implica interpretar mes y día. Codificado correctamente, como se ve en la tabla de la derecha, no habrá lugar a confusión dado que el formato es AAA-MM-DD

| start-date; end-date |
|---|
| **01.01.2014**; **12.03.2014** |
| **01.01.2014**; **01.07.2016** |

| start-date; end-date |
|---|
| **2014-01-01**; **2014-03-12** |
| **2014-01-01**; **2016-07-01** |

## 3.9. Formateo de datos numéricos

**Descripción de problema:** En ocasiones, los datos numéricos se representan utilizando separadores entre la parte entera y la decimal que, dependiendo de la configuración regional, pueden ser un punto o una coma. En los separadores de unidades de millar, ocurre de forma similar e incluso, se pueden llegar a usar espacios en blanco. Estas diferencias pueden ocasionar una interpretación errónea, especialmente cuando los datos se procesan automáticamente.

**Características de calidad afectadas:** Interoperabilidad, Conformidad/Cumplimiento, Legibilidad por maquinas/Procesabilidad.

**Recomendaciones**:

- Con el fin de internacionalizar los datos, debe utilizarse un punto como separador entre la parte entera y decimal de un número.

- En el caso de los separadores de unidades de millar, la recomendación es no usarlos y en ningún caso, usar como separadores caracteres en blanco.
- Los valores negativos deben ir precedidos de un signo menos (-). No se deben usar paréntesis para indicar valores negativos.
- Si una columna contiene valores enteros y decimales, el tipo de dato debe ser decimal y, por tanto, se incluirá el separador ',' o '.' y el número de cifras decimales que proceda, normalmente dos decimales.
- Si una columna solo contiene valores enteros, se expresarán sin separador decimal.
- No se debe mezclar texto con valores numéricos. Por ejemplo: no se debe usar 50€ o 27 km como valor en un campo numérico.

**Ejemplo:** En la tabla de la izquierda vemos representaciones erróneas de datos numéricos, mientras que, en la tabla de la derecha, vemos como se deberían de representar los números de manera correcta en un dataset.

| |
|---|
| 0,53 |
| 789.654 |
| 789 654 |
| 25.026,8 |

| |
|---|
| 0.53 |
| 789654.0 |
| 789654.0 |
| 25026.8 |

## 3.10. Evitar la mezcla de escalas numéricas

**Descripción de problema:** La publicación de calidad de un conjunto de datos implica que exista consistencia en todas las distribuciones que de éste se publiquen. Las características que presenta cada una de las variables condiciona su análisis. Uno de los errores más comunes que se producen cuando se publican series temporales, es el cambio repentino de criterio en la representación de la escala de medición de los datos, sea ésta categórica o numérica, es decir, en un primer momento los datos se han recogido y distribuido atendiendo a un determinado nivel de medición y posteriormente esa escala se ve modificada sin motivo aparente. Por ejemplo, una serie que comienza en meses pasa al cabo de un determinado periodo a medirse en trimestres sin explicar el cambio.

Esta problemática puede ocasionar errores en los resultados obtenidos a partir del análisis de esos datos, o incluso los datos podría ser difícilmente reutilizables. El usuario debe detectar que se ha producido un cambio en la escala, si no se indica en la descripción del conjunto de datos, y ocupar sus propios recursos, para solventar esta anomalía.

**Características de calidad afectadas:** Reusabilidad, Consistencia, Precisión.

**Recomendaciones:** Lo más adecuado es garantizar que la escala de medida no variara a lo largo del tiempo, siendo exactamente la misma en todas las distribuciones publicadas del conjunto de datos. Para ello es necesario que el organismo que recoge los datos genere un protocolo de recogida de datos preciso y exhaustivo, donde se detalle cómo se recoge y representa cada una de las variables.

En algunas ocasiones, es completamente imprescindible el cambio de escala, por diversas circunstancias, por ejemplo, el instrumento de medida cambia o cambia la normativa que obliga a recoger los datos de otra determinada manera, entre otras circunstancias. En este caso, el publicador debería invertir recursos en proporcionar los datos en ambas escalas, impidiendo de esta manera que el coste de procesamiento recaiga en los usuarios. Si no es posible asumir esta tarea por parte del publicador, **la recomendación es indicar claramente en la descripción del conjunto de datos el momento exacto en el que ha cambiado la escala de medición de la variable y a qué nivel concreto de escala afecta**, para que los usuarios puedan interpretar los datos y optimicen la cantidad de recursos para su reutilización.

**Ejemplo:** En la primera tabla se observa un cambio de escala en las ventas de coches por año (medidas en miles).

| marca | año | consumo | ventas |
|---|---|---|---|
| **buick skylark 320** | 1998 | Alto | 2.50 |
| **buick skylark 320** | 1999 | Bajo | 2.63 |
| **buick skylark 320** | 2000 | Medio | 3050 |
| **buick skylark 320** | 2001 | Alto | 3400 |
| **buick skylark 320** | 2002 | Medio | 3570 |
| **plymouth satellite** | 1998 | Medio | 2.37 |
| **plymouth satellite** | 1999 | Bajo | 2.40 |
| **plymouth satellite** | 2000 | Medio | 4800 |
| **plymouth satellite** | 2001 | Alto | 3600 |

| marca | año | consumo | ventas |
|---|---|---|---|
| **buick skylark 320** | 1998 | Alto | 2.50 |
| **buick skylark 320** | 1999 | Bajo | 2.63 |
| **buick skylark 320** | 2000 | Medio | 3.05 |
| **buick skylark 320** | 2001 | Alto | 3.40 |
| **buick skylark 320** | 2002 | Medio | 3.57 |
| **plymouth satellite** | 1998 | Medio | 2.37 |
| **plymouth satellite** | 1999 | Bajo | 2.40 |
| **plymouth satellite** | 2000 | Medio | 4.80 |
| **plymouth satellite** | 2001 | Alto | 3.60 |

## 3.11. Evitar la mezcla de rangos en un mismo conjunto de datos

**Descripción de problema:** El uso de rangos en un conjunto de datos, limita la información a la cual tiene acceso el usuario. Además, en numerosas ocasiones, los rangos de datos utilizados en las distintas distribuciones son inconsistentes. Esta problemática ocasiona perdida de efectividad por parte del usuario para poder reutilizar esos conjuntos de datos y en la mayoría de los casos, finalmente, ocasiona perdida de información en su procesamiento.

**Características de calidad afectadas:** Reusabilidad, Consistencia, Precisión.

**Recomendaciones:** Es aconsejable que los conjuntos de datos se publiquen con el mayor nivel de desagregación posible evitando el uso de rangos de datos, facilitando a los usuarios el mayor detalle de toda la información. Si no es posible representar los datos con el máximo nivel de desagregación es imprescindible mantener la consistencia en todos los valores de la variable y evitar la combinación de texto e información numérica añadiendo columnas adicionales que representen el valor inicial y final de rango.

**Ejemplo 1:** Características y ventas de diferentes vehículos en la que la potencia se representa utilizando un rango.

| marca | año | consumo | ventas | potencia | aceleración |
|---|---|---|---|---|---|
| **ford torino** | 1970 | Alto | 2.50 | De 100 a 150 | 12 |
| **buick skylark 320** | 1970 | Medio | 2.63 | De 150 a 200 | 11.5 |
| **plymouth satellite** | 1970 | Medio | 2.37 | De 150 a 200 | 11 |
| **chrevrolet chevelle malibu** | 1970 | Bajo | 2.40 | Mas de 200 | 13 |

| marca | año | consumo | ventas | potencia | aceleración |
|---|---|---|---|---|---|
| **ford torino** | 1970 | Alto | 2.50 | 130 | 12 |
| **buick skylark 320** | 1970 | Medio | 2.63 | 165 | 11.5 |
| **plymouth satellite** | 1970 | Medio | 2.37 | 150 | 11 |
| **chrevrolet chevelle malibu** | 1970 | Bajo | 2.40 | 210 | 13 |

| marca | año | consumo | ventas | rango-potencia-final | aceleración |
|---|---|---|---|---|---|
| **ford torino** | 1970 | Alto | 2.50 | 150 | 12 |
| **buick skylark 320** | 1970 | Medio | 2.63 | 200 | 11.5 |
| **plymouth satellite** | 1970 | Medio | 2.37 | 200 | 11 |
| **chrevrolet chevelle malibu** | 1970 | Bajo | 2.40 | NA | 13 |

## 3.12. Incorporar variables con información geográfica

**Descripción de problema:** La disponibilidad de conjuntos de datos que incluyen referencias espaciales ha aumentado en los últimos años. Los campos que incluyen información geográfica deben estar representados de una determinada manera para su eficiente procesamiento. Por ejemplo, en numerosas ocasiones, estos campos se especifican en grados sexagesimales lo que dificulta su representación en un mapa. Por otro lado, ocurre también que estos datos en ocasiones se publican únicamente en formato CSV, pero no en formatos específicos para procesar datos espaciales, como SHP o KML.

**Características de calidad afectadas:** Interoperabilidad, Conformidad/Cumplimiento, Legibilidad por maquinas

**Recomendaciones**: Es aconsejable **publicar las coordenadas geográficas de latitud y longitud en grados decimales, cuyos valores se representen en dos columnas independientes que deben llamarse: "latitud" y "longitud", respectivamente**. Además, siempre que se publique este tipo de datos, deben usarse formatos específicos, como por ejemplo SHP o KML, complementando a otro tipo de formato, por ejemplo CSV o XLS, para facilitar su reutilización con diferentes programas.

Para la publicación de este tipo de datos, es recomendable seguir la Guía práctica para la publicación de Datos Espaciales. También es conveniente seguir las especificaciones Well-known text representations of coordinate reference systems, (WKT-CRS) del Open Geospatial Consortium.

**Ejemplo:** en el primer caso se observa una forma inadecuada de representar valores de coordenadas geográficas utilizando grados sexagesimales. A continuación, los mismos datos en grados decimales.

| concesionario | latitud | longitud |
|---|---|---|
| **chevrolet** | 43° 14' 04" | 5° 07' 24" |
| **buick** | 43° 20' 44" | 5° 14' 14" |

| concesionario | latitud | longitud |
|---|---|---|
| **chevrolet** | 43,2345678 | -5,1234567 |
| **buick** | 43,3456789 | -5,2345678 |

## 3.13. Evitar la incorporación de subtotales, totales o agrupamientos

**Descripción de problema:** Con frecuencia se incluyen filas o columnas de totales o subtotales en tablas, generando de esta forma agregaciones dentro del conjunto de datos. Cuando se incluyen este tipo de filas o columnas, como consecuencia de aplicar una determinada operación, resulta muy difícil y en ocasiones imposible, recuperar el dato desagregado.

Otro de los problemas relacionados con este caso es el agrupamiento, que implica agrupar filas relacionadas con una entidad, dejando ciertas celdas vacías. Este problema es común y puede ocasionar problemas cuando se modifica el orden original de las filas.

**Características de calidad afectadas:** Accesibilidad/disponibilidad, Reusabilidad, Precisión, Completitud.

**Recomendaciones:** Un dataset de calidad siempre debe presentar el **mayor nivel de desagregación posible y ser consistente con el nivel de granularidad de los datos que contiene**. Un nivel de granularidad superior de los datos siempre se puede obtener a partir de un nivel inferior, pero difícilmente se puede lograr a la inversa. Por ejemplo, es posible obtener el número de ventas anuales a partir de los datos de ventas mensuales, pero no a la inversa.

Por otro lado, la forma de evitar las celdas vacías de un agrupamiento relacionadas con una entidad es repitiendo dicha entidad en todas las filas del agrupamiento.

**Ejemplo 1:** Venta semestral de coches (en miles), con totales donde se están mezclando niveles de granularidad. En la siguiente tabla se observa la misma información sin totales y por tanto con el mismo nivel de granularidad)

| Marca | | Año | Ventas-semestrales |
|---|---|---|---|
| **chevrolet chevelle malibu** | | 1998 | 2.5 |
| **chevrolet chevelle malibu** | | 1998 | 2.63 |
| | Total anual | 1998 | 5.13 |
| **buick skylark 320** | | 1999 | 3.4 |
| **buick skylark 320** | | 1999 | 3.57 |
| | Total anual | 1999 | 6.97 |

| Marca | | Año | Ventas-semestrales |
|---|---|---|---|
| **plymouth satellite** | | 2000 | 2.4 |
| **plymouth satellite** | | 2000 | 2.52 |
| | Total anual | 2000 | 4.92 |

| Marca | Año | Ventas-s1 | Ventas-s2 |
|---|---|---|---|
| **chevrolet chevelle malibu** | | 1998 | 2.5 |
| **buick skylark 320** | | 1998 | 2.63 |
| **plymouth satellite** | Total anual | 1998 | 5.13 |

**Ejemplo 2:** A continuación, se muestra cómo evitar el agrupamiento en base al uso de celdas vacías.

| Marca | Año | Ventas-semestrales |
|---|---|---|
| **chevrolet chevelle malibu** | 1998 | 2.5 |
| | 1999 | 2.63 |
| | 2000 | 3.13 |
| **buick skylark 320** | 1998 | 3.4 |
| | 1999 | 3.57 |
| | 2000 | 3.97 |

| Marca | Año | Ventas-semestrales |
|---|---|---|
| **chevrolet chevelle malibu** | 1998 | 2.5 |
| **chevrolet chevelle malibu** | 1999 | 2.63 |
| **chevrolet chevelle malibu** | 2000 | 3.13 |
| **buick skylark 320** | 1998 | 3.4 |
| **buick skylark 320** | 1999 | 3.57 |
| **buick skylark 320** | 2000 | 3.97 |

## 3.14. Evitar la fragmentación de datos y de difícil localización

**Descripción del problema:** Es frecuente que los datos que se pretende localizar estén disponibles en algún portal web, pero estén poco documentados, sean de difícil acceso o estén fragmentados.

En abundantes ocasiones ocurre de los datos están divididos y distribuidos en distintas secciones o páginas dentro del portal de un organismo o entidad, o incluso en distintos sitios web. Esta circunstancia dificulta su encontrabilidad y por tanto su acceso.

En ocasiones ocurre un problema añadido, la existencia de múltiples versiones de los datos replicadas en distintos espacios web que presentan distintos metadatos y características, lo que contribuye a confundir a los usuarios.

**Características de calidad afectadas:** Accesibilidad/disponibilidad, Apertura.

**Recomendaciones:** Mejorar la usabilidad de los sitios web en general, y concretamente la organización de los contenidos y el etiquetado de éstos permitiendo una búsqueda más pormenorizada. Además, es necesario establecer conexiones entre los distintos conjuntos de datos para visibilizar las relaciones existentes entre ellos.

Una buena práctica, es la creación de un catálogo centralizado con todos los datos disponibles del organismo o la empresa, facilitando así su acceso y encontrabilidad. Para ello, una solución es hacer uso de un sistema de gestión de datos abiertos o DMS (Data Management System, por sus siglas en inglés). Uno de los DMS más populares es CKAN, donde es posible publicar todos los conjuntos de datos que el organismo tenga disponibles, mejorando de esta manera su visibilidad, y con ello el acceso y la reutilización de los mismos. Como alternativa, el catálogo nacional, datos.gob.es, proporciona un Widget que permite al organismo usar el Catálogo Nacional como catálogo propio de conjuntos de datos.

Otra de las recomendaciones, es el uso de los metadatos, de manera correcta, completa y proporcionados también en un formato legible por máquinas como RDF. Estándares de referencia son DCAT, DCAT-AP y GeoDCAT-AP. Los catálogos de datos en España se rigen por la Norma Técnica de Interoperabilidad de Recursos de Información (NTI-RISP), de la manera que se describe en la guía de aplicación de la Norma Técnica de Interoperabilidad[8], facilitando la encontrabilidad de los datos a través de motores de búsqueda, agregadores u otras herramientas. La NTI-RISP es compatible con el estándar DCAT.

## 3.15. Organizar adecuadamente los datasets disponibles

**Descripción del problema:** Los datos describen hechos de distinta naturaleza. Si los datos refieren observaciones numéricas e incluyen referencias temporales, podemos encontramos ante datasets del ámbito estadístico. Si el dato, además hace referencia, directa o indirectamente a una localización o zona geográfica específica, entonces el contexto podría ser el de datos espaciales. Los datasets, en definitiva, son proyecciones de la realidad de cualquier ámbito.

La disponibilidad de datasets para su reutilización se puede realizar mediante la descarga de archivos o accediendo a servicios de datos (APIs). Los archivos de datos descargables constituyen la materialización completa de un dataset. En cambio, el acceso a datos mediante servicios permite obtener múltiples datasets en función de las consultas que el reutilizador configure.

Como estamos viendo a lo largo de esta guía, la forma en la que se representan los datasets, es decir, los formatos, es diversa y responde a las necesidades del reutilizador: puede ser una hoja de cálculo CSV o Excel, un archivo XML, un archivo de imagen, datos vinculados en formato RDF, etc. La disponibilidad de formatos alternativos de cada dataset facilita la reutilización.

**Características de calidad afectadas:** Accesibilidad/disponibilidad, Apertura.

**Recomendaciones:** Cuando se planifica la publicación de un dataset, hay que organizar su disponibilidad adecuadamente. En primer lugar, hay que tener en cuenta que cada forma de representación -formato de archivo descargable o punto de acceso a un servicio de datos-, constituye un recurso o distribución del conjunto de datos. Por ejemplo,

Dataset X
- dataset X en formato CSV
- dataset x en formato XML
- acceso a la API que sirve el dataset X
- …

Además, atendiendo al contenido, un dataset se puede publicar de forma fragmentada, entre otras razones, debido a su excesivo tamaño o porque el origen de los datos proporciona vistas parciales de los mismos. Por ejemplo, cuando los datos están organizados en forma de series temporales, éstas contienen conjuntos de observaciones que incluyen referencias de tiempo asociadas y, por tanto, los datos se presentan de acuerdo a periodos de observación temporales. Un caso puede ser la publicación del presupuesto anual de gasto de un organismo que se podría organizar publicando las siguientes distribuciones:

Dataset: Presupuesto anual de gasto de un organismo A
- Presupuesto 2018 del organismo A
- Presupuesto 2019 del organismo A
- Presupuesto 2020 del organismo A
- Presupuesto 2021 del organismo A
- …

Cuando se realiza este tipo de organización es importante tener en cuenta que cualquier cambio en la estructura de datos introduce una ruptura en la serie temporal, es decir, una discontinuidad. Los puntos de discontinuidad en una serie temporal deben identificarse y explicarse y por esta razón, las revisiones sustanciales de las series temporales deben, si es posible, proporcionar a los usuarios información suficiente sobre los cambios conceptuales y metodológicos de forma que los datos puedan mantener su consistencia a lo largo del tiempo.

Por otro lado, los datos pueden estar organizados transversalmente (cross-sectional) incluyendo múltiples observaciones en un momento concreto de tiempo y, por tanto, se organizan en torno a otro tipo de dimensiones como la geográfica. Por ejemplo, las posibles distribuciones de un conjunto de datos que representa el presupuesto de gasto en un periodo concreto para diferentes CCAA pueden ser las siguientes:

Dataset: Presupuesto gasto de las CCAA de 2017
- Presupuesto de gasto de la CCAA A
- Presupuesto de gasto de la CCAA B
- Presupuesto de gasto de la CCAA C
- Presupuesto de gasto de la CCAA D
- …

En cualquier caso, el reutilizador debe tener disponible **una colección estructurada, coherente y completa del conjunto de recursos que constituyen el dataset.** Para facilitar la reutilización es recomendable organizar la publicación del dataset aplicando los siguientes criterios:

- Si el dataset se distribuye en diferentes formatos de datos, cada distribución se corresponderá con cada uno de los formatos disponibles y en su caso, el acceso al servicio de datos disponible.

- Si el dataset contiene información temporal, es aconsejable publicar como una serie temporal de distribuciones dividida en años, semestres, trimestres, meses, etc.

- Si el dataset contiene delimitaciones geográficas, es razonable publicar distribuciones por unidades territoriales, por ejemplo, CCAA, provincias, zonas, etc.

- Si el dataset contiene dimensiones o categorías temáticas de información atemporales, se pueden publicar tantas distribuciones como categorías existentes.

En todo caso, siempre que se publiquen datasets descargables de forma fragmentada es recomendable **publicar de forma complementaria una distribución con la opción de descarga o acceso al dataset completo.** De esta forma, se cubrirán las expectativas de cualquier reutilizador y se evitará que un exceso en la fragmentación de los archivos de datos dificulte el acceso a los reutilizadores ya que puede incrementar sustancialmente el número de descargas necesarias.

# 4. Pautas para asegurar la calidad usando formatos específicos de datos

Las recomendaciones generales detalladas en la sección anterior son aplicables a cualquier formato de representación de datos. Más adelante, en el apartado "Recomendaciones para la documentación de datos", nos centraremos en cómo componer y disponer diccionarios de datos para facilitar la interoperabilidad y procesamiento de los datos. Antes, tengamos en cuenta que también existen pautas que aplican de forma específica a ciertos formatos de uso común en los portales de datos abiertos. A continuación, te detallamos las más relevantes.

## 4.1. Formato CSV

Las pautas de calidad de datos abiertos para el formato CSV están detalladas en la Guía Práctica para la publicación de datos tabulares en archivos CSV. La mayoría de ellas se han incluido en el apartado anterior como pautas generales porque aplican a todos los formatos de datos. En la guía específica de archivos CSV, encontrarás, además, información útil para estructurar adecuadamente datos tabulares, geocodificar direcciones postales y pautas para exportar/importar datos tabulares desde herramientas de hojas de cálculo a archivos CSV, entre otros aspectos aplicables en este tipo de archivos. A continuación, te resumimos algunas de las pautas específicas que debes tener en cuenta para garantizar una buena calidad en archivos CSV:

### 4.1.1. Usar punto y coma (";") como delimitador

**Características de calidad afectadas:** Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** En un archivo de datos en formato CSV, cada campo debe estar separado del siguiente por un carácter singular: por ejemplo, una coma [",“], un punto y coma [";“], un carácter pipe ["|"] o un carácter tabulador [TAB]. Cuando los campos están separados por un carácter tabulador [TAB], el formato de archivo es TSV. Alternativamente, los campos pueden tener una longitud fija de caracteres. El carácter delimitador o separador, siempre se establece entre dos valores y el último valor del registro no va seguido de un delimitador. Dado que los valores de los campos -principalmente de texto-, pueden contener habitualmente el carácter coma [",“], **se recomienda usar punto y coma [";“] como carácter separador**, ya que se usa con menos frecuencia. No obstante, para evitar que una coma se interprete como un separador, debería enmascararse. De esta forma, los valores de los campos que incluyen comillas, comas o retornos de carro deben ir entre comillas. Por otro lado, es importante asegurar que no hay espacios ni tabulaciones ni al principio ni al final de cada registro del archivo.

**Ejemplo:** En el ejemplo se observa como en el archivo CSV se utilizan caracteres ";" como separadores, pero en la versión de la izquierda se incluyen delimitadores al final de cada registro, así como caracteres en blanco alrededor de los propios delimitadores.

| año; visitantes: tiempo-de-visita |
|---|
| 2013; 822101;00:02:59; |
| 2012;792967;00:02:52; |
| 2011; 721519;00:03:44; |

| año; visitantes: tiempo-de-visita |
|---|
| 2013;822101;00:02:59 |
| 2012;792967;00:02:52 |
| 2011;721519;00:03:44 |

### 4.1.2. Incluir una tabla de datos por fichero

**Características de calidad afectadas:** Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** Cada archivo CSV solo debe contener una tabla. Si se trata de publicar una hoja de cálculo que contiene varias hojas, se debe crear un archivo CSV para cada hoja. Una estructuración diferente dificultara la interpretación y procesamiento por máquinas.

### 4.1.3. Evitar incluir información adicional en el fichero de datos

**Características de calidad afectadas:** Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** Es importante asegurar que el archivo CSV solo contiene los datos que van a ser procesables en una reutilización. Concretamente, **solo debe contener los encabezados de columna y los valores de cada registro de la tabla**. En numerosas ocasiones, los archivos CSV se generan a partir de una exportación de una hoja de cálculo que está diseñada para una interpretación visual por las personas incorporando contenido adicional y determinadas características de diseño que dificultan el procesamiento automático de los datos. Por ejemplo, el uso de cabeceras con títulos, líneas en blanco para separar registros, explicaciones sobre los datos o determinados formatos visuales para enriquecer el contenido. Estos contenidos también se procesan automáticamente y constituyen un foco de posibles problemas. Cualquier información adicional sobre los datos debe incluirse en la descripción de estos utilizando los metadatos apropiados en el Diccionario de Datos. Por ejemplo, cualquier explicación sobre los datos puede ser muy útil para que los usuarios comprendan mejor el contenido, pero nunca se deben incluir directamente en el archivo CSV. En su lugar, las explicaciones y descripciones deben almacenarse en propiedades de metadatos adecuadas, como la propiedad dct:description de DCAT.

**Ejemplo:** En el siguiente ejemplo (arriba) se observa una hoja de cálculo visualmente bien organizada e interpretable por personas, con títulos, varias secciones y líneas en blanco. La exportación directa a un archivo CSV (abajo) incluye todos los elementos de la tabla original obligando a interpretar y procesar en la reutilización la línea del título de la tabla, las líneas en

blanco y demás elementos incluidos. Dado que esto puede provocar fallos en el procesamiento, se debe evitar incluir contenido adicional que no sea exclusivamente el encabezado de las columnas y los valores de los datos.

## Ejemplo de hoja de cálculo con formato

**Indicar valores**

| | |
|---|---|
| Importe del préstamo | 5.000,00 € |
| Tasa de interés anual | 4,00% |
| Periodo del préstamo en años | 1 |
| Número de pagos por año | 12 |
| Fecha de inicio del préstamo | 26/08/2022 |
| **Pagos adicionales opcionales** | 100,00 € |

**Resumen del préstamo**

| | |
|---|---|
| Pago programado | 425,75 € |
| Número de pagos programados | 12 |
| Número real de pagos | 10 |
| Importe total de pagos anticipadc | 900,00 € |
| Importe total de intereses | 89,62 € |
| **Nombre del prestamista** | Banco Woodgrove |

| Número de pago | Pago Fecha | Inicio Saldo | Pago programado | Adicional Pago | Total Pago | Director | Interés | Fin Saldo | Acumulado Interés |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 26/08/2022 | 5.000,00 € | 425,75 € | 100,00 € | 525,75 € | 509,08 € | 16,67 € | 4.490,92 € | 16,67 € |
| 2 | 26/09/2022 | 4.490,92 € | 425,75 € | 100,00 € | 525,75 € | 510,78 € | 14,97 € | 3.980,14 € | 31,64 € |
| 3 | 26/10/2022 | 3.980,14 € | 425,75 € | 100,00 € | 525,75 € | 512,48 € | 13,27 € | 3.467,65 € | 44,90 € |
| 4 | 26/11/2022 | 3.467,65 € | 425,75 € | 100,00 € | 525,75 € | 514,19 € | 11,56 € | 2.953,46 € | 56,46 € |
| 5 | 26/12/2022 | 2.953,46 € | 425,75 € | 100,00 € | 525,75 € | 515,90 € | 9,84 € | 2.437,56 € | 66,31 € |
| 6 | 26/01/2023 | 2.437,56 € | 425,75 € | 100,00 € | 525,75 € | 517,62 € | 8,13 € | 1.919,94 € | 74,43 € |
| 7 | 26/02/2023 | 1.919,94 € | 425,75 € | 100,00 € | 525,75 € | 519,35 € | 6,40 € | 1.400,59 € | 80,83 € |
| 8 | 26/03/2023 | 1.400,59 € | 425,75 € | 100,00 € | 525,75 € | 521,08 € | 4,67 € | 879,50 € | 85,50 € |
| 9 | 26/04/2023 | 879,50 € | 425,75 € | 100,00 € | 525,75 € | 522,82 € | 2,93 € | 356,69 € | 88,43 € |
| 10 | 26/05/2023 | 356,69 € | 425,75 € | 0,00 € | 356,69 € | 355,50 € | 1,19 € | 0,00 € | 89,62 € |

```
;;;;;;;;;
;Ejemplo de hoja de cálculo con formato;;;;;;;;
;;;;;;;;;
Indicar valores;;;;;Resumen del préstamo;;;;
Importe del préstamo;;;5.000,00 €;;Pago programado;;425,75 €;;
Tasa de interés anual;;;4,00%;;Número de pagos programados;;12;;
Periodo del préstamo en años;;;1;;Número real de pagos;;10;;
Número de pagos por año;;;12;;Importe total de pagos anticipados;;900,00 €;;
Fecha de inicio del préstamo;;;26/08/2022;;Importe total de intereses;;89,62 €;;
;;;;;;;;;
Pagos adicionales opcionales;;;100,00 €;;Nombre del prestamista;;Banco Woodgrove;;
;;;;;;;;;
Número de pago;"Pago
Fecha";"Inicio
Saldo";Pago programado;"Adicional
Pago";"Total
Pago";Director;Interés;"Fin
Saldo";"Acumulado
Interés"
1;26/08/2022;5.000,00 €;425,75 €;100,00 €;525,75 €;509,08 €;16,67 €;4.490,92 €;16,67 €
2;26/09/2022;4.490,92 €;425,75 €;100,00 €;525,75 €;510,78 €;14,97 €;3.980,14 €;31,64 €
3;26/10/2022;3.980,14 €;425,75 €;100,00 €;525,75 €;512,48 €;13,27 €;3.467,65 €;44,90 €
4;26/11/2022;3.467,65 €;425,75 €;100,00 €;525,75 €;514,19 €;11,56 €;2.953,46 €;56,46 €
5;26/12/2022;2.953,46 €;425,75 €;100,00 €;525,75 €;515,90 €;9,84 €;2.437,56 €;66,31 €
6;26/01/2023;2.437,56 €;425,75 €;100,00 €;525,75 €;517,62 €;8,13 €;1.919,94 €;74,43 €
7;26/02/2023;1.919,94 €;425,75 €;100,00 €;525,75 €;519,35 €;6,40 €;1.400,59 €;80,83 €
8;26/03/2023;1.400,59 €;425,75 €;100,00 €;525,75 €;521,08 €;4,67 €;879,50 €;85,50 €
9;26/04/2023;879,50 €;425,75 €;100,00 €;525,75 €;522,82 €;2,93 €;356,69 €;88,43 €
10;26/05/2023;356,69 €;425,75 €;0,00 €;356,69 €;355,50 €;1,19 €;0,00 €;89,62 €
```

### 4.1.4. Incluir una primera fila única de cabecera

**Características de calidad afectadas:** Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** Las tablas de datos pueden contener, opcionalmente, **una y solo una línea de cabecera para especificar los nombres de los campos**. Es importante tener en cuenta que los nombres de las columnas que se incluyen en la línea de cabecera son un tipo de anotación o metadato que nombra cada columna y no forma parte de los datos, es decir, no se debe considerar cuando se cuenta el número de filas de datos en una tabla. Si el origen de datos es una hoja de cálculo, para nombrar las columnas se deben usar celdas simples y en ningún caso, celdas combinadas.

Por otro lado, hay que considerar que no existe un mecanismo para discernir automáticamente si el primer registro de un CSV es una línea de cabecera ya que ésta se codifica como cualquier otro registro. Por tanto, es buena práctica especificar la presencia o ausencia de línea de cabecera en el diccionario de datos mediante el uso de una propiedad "title = ". Otra forma de indicar la presencia o ausencia de la línea de cabecera es mediante un parámetro del tipo de contenido cuando el archivo de datos es transmitido vía HTTP, de la forma: Content-Type: text/csv;header=absent.

**Ejemplo:** A continuación, se muestran dos vistas de una tabla. En la superior se observa el uso de celdas combinadas para definir la cabecera de datos y en la inferior una forma alternativa utilizando celdas simples en una única fila.

| Marca | Contacto- concesionario | |
|---|---|---|
| | Concesionario-mail | Concesionario-teléfono |
| **chevrolet chevelle malibu** | mail@concesionario_chevrolet.com | +34-1111111 |
| **buick skylark 320** | mail@concesionario_buick.com | +34-2222222 |

| Marca | contacto-concesionario-mail | contacto-concesionario-teléfono |
|---|---|---|
| **chevrolet chevelle malibu** | mail@concesionario_chevrolet.com | +34-1111111 |
| **buick skylark 320** | mail@concesionario_buick.com | +34-2222222 |

### 4.1.5. Asegurar que todas las filas tengan el mismo número de columnas

**Características de calidad afectadas:** Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** Cada fila o registro de un archivo CSV debe contener el mismo número de columnas y de esta forma asegurar la coherencia de la estructura en el archivo completo. Esto implica que cada fila debe tener el mismo número de delimitadores. El hecho de que no sea así puede conducir a un procesamiento erróneo de los datos. Los motivos por los que las filas de un CSV pueden contener diferente número de columnas, son varios. Por un lado, puede ocurrir que algunos valores se hallan "escapado" incorrectamente. Por ejemplo, un valor que contiene un punto y coma que no está enmascarado y, por lo tanto, se interpreta como un delimitador. Otro problema común deriva del tratamiento de valores ausentes o desconocidos. Cualquier valor ausente en una fila, debe interpretarse como un valor "nulo" y es imprescindible expresarlo de forma coherente en todo el archivo de datos. Como norma general, **hay que rellenar todos los valores de una tabla y mantener un código común para los datos desconocidos**.

Hay que tener en cuenta, que los valores desconocidos, cuando se dejan sin explicar o simplemente se encuentran ausentes, suelen generar confusión, especialmente cuando la columna de datos es numérica. Además, generan resultados erróneos en tareas de ordenación.

**Recomendaciones para evitar valores de datos desconocidos:**

- Si la celda en blanco representa un cero, entonces el valor debe ser 0.
- Si la celda en blanco representa un valor "desconocido" o "no obtenido", entonces esta posibilidad debe explicarse en el diccionario de datos e indicarse con un código específico.
- Si un valor en blanco tiene un significado, se debe valorar la opción de añadir una nueva columna para incluir la explicación del valor "en blanco" como un valor posible.
- Una terminología aceptada para indicar valores desconocidos o ausentes es el valor o código específico NA o N/A. [2]
- El código que se utilice para indicar los valores desconocidos o ausentes, por ejemplo, NA, debe especificarse en el diccionario de datos.

**Ejemplo:** En el ejemplo se observa que todas las filas tienen igual número de celdas y para que el archivo CSV que se genere no contenga errores, se deben tratar los valores ausentes, nulos o en blanco. Por tanto, se debe entender que el valor 0 en la columna “ventas” indica que para ese año las ventas de coches de ese modelo han sido 0. En cambio, cuando el dato de “ventas”, al

[2] Del inglés, not available (no disponible), not applicable (no corresponde en el caso) o no answer (sin respuesta; aunque este significado solo se usa en ciertas situaciones). https://es.wikipedia.org/wiki/N/a

igual que el de "consumo" se desconoce, se indica con NA. Además, todos los valores desconocidos en cualquier columna se indican con el mismo código: NA.

| Marca | Año | Consumo | Ventas |
|---|---|---|---|
| **chevrolet chevelle malibu** | 1998 | Alto | 2.50 |
| **chevrolet chevelle malibu** | 2000 | Medio | |
| **buick skylark 320** | 1998 | | 3.40 |
| **buick skylark 320** | 199 | Medio | 3.57 |
| **buick skylark 320** | 2000 | Medio | N/A |
| **plymouth satellite** | 1998 | | 2.40 |
| **plymouth satellite** | 2000 | Medio | 3.60 |

| Marca | Año | Consumo | Ventas |
|---|---|---|---|
| **chevrolet chevelle malibu** | 1998 | Alto | 2.50 |
| **chevrolet chevelle malibu** | 2000 | Medio | 0 |
| **buick skylark 320** | 1998 | NA | 3.40 |
| **buick skylark 320** | 199 | Medio | 3.57 |
| **buick skylark 320** | 2000 | Medio | NA |
| **plymouth satellite** | 1998 | NA | 2.40 |
| **plymouth satellite** | 2000 | Medio | 3.60 |

## 4.2. Formato XML

XML es un formato estándar de amplísima utilización y que históricamente ha sido la piedra angular de la interoperabilidad técnica al ser el lenguaje por excelencia de intercambio de datos entre sistemas de información. A continuación, te describimos algunas pautas específicas de calidad para este formato.

### 4.2.1. Proporcionar una declaración XML

**Características de calidad afectadas:** Reusabilidad, Consistencia.

**Recomendaciones:** Cada archivo XML debe tener una declaración XML completa. Debe contener metadatos relativos a la estructura del documento para que las aplicaciones puedan procesar correctamente el archivo. Por ejemplo, la información sobre la versión XML y la codificación de caracteres debe especificarse en los metadatos que se incluyen en dicha declaración. El atributo

"version" es obligatorio, mientras que el uso de los atributos "encoding" o "standalone" es opcional.

**Ejemplo:**

```
<fruits>
        <fruit>
                <type>Apple</type>
                <origin>Germany</origin>
                <drupe>true</drupe>
        </fruit>
</fruits>
```

**La captura de pantalla que se muestra a la izquierda muestra un archivo XML sin declaración.**

```
<?xml version="1.0" encoding="UTF-8"?>
<fruits>
        <fruit>
                <type>Apple</type>
                <origin>Germany</origin>
                <drupe>true</drupe>
        </fruit>
        <fruit>
                <type>Grape</type>
                <origin>Italy</origin>
                <drupe>false</drupe>
        </fruit>
</fruits>
```

**Esta captura de pantalla muestra a el mismo archivo XML con una declaración correctamente formateada que indica la versión XML y la codificación de caracteres utilizada**

### 4.2.2. Uso de caracteres de escape

**Características de calidad afectadas:** Reusabilidad, Consistencia

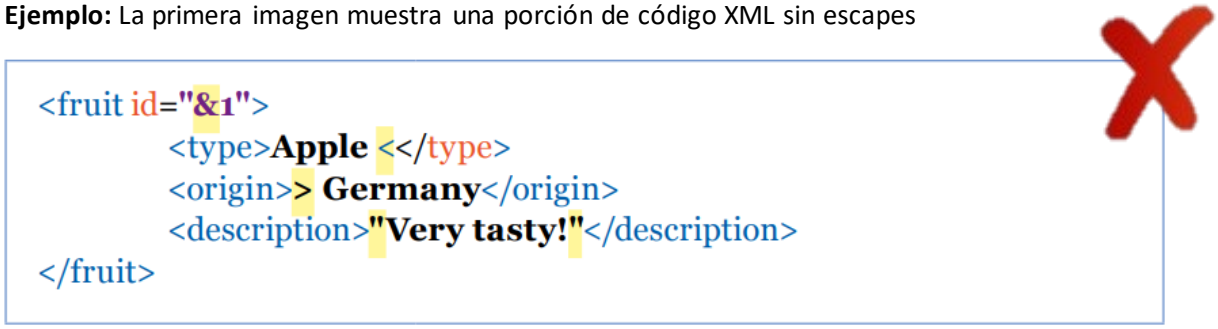
**Recomendaciones:** Cuando se utilizan caracteres especiales en XML, es necesario "escaparlos", es decir, invocar una interpretación alternativa a ese carácter o secuencia de caracteres. Esto garantiza una estructura de archivo sólida y evita que las aplicaciones utilizadas para procesar el archivo malinterpreten los datos. El escape se realiza sustituyendo los caracteres afectados por las entidades XML equivalentes. En la tabla siguiente se muestra un resumen de los caracteres especiales.

**Tabla:** Caracteres que deben escaparse en XML

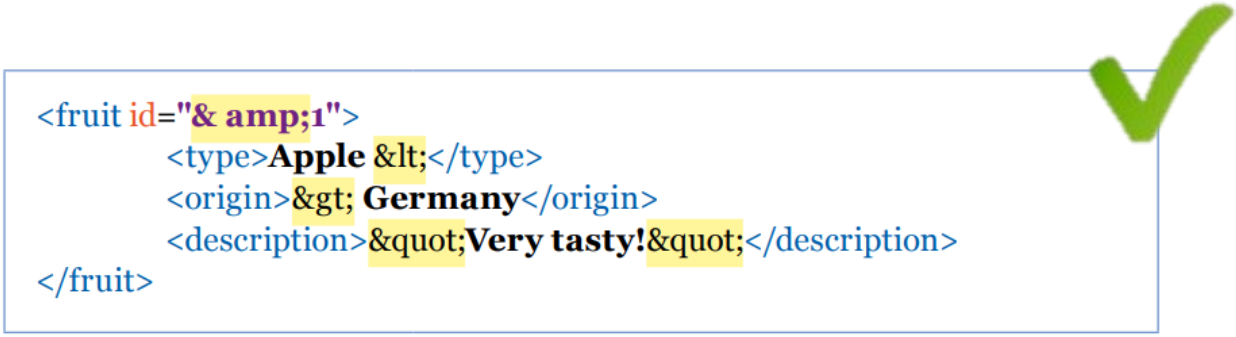
| | Formato de escape | Sustituido por |
|---|---|---|
| **"y", "et" o ampersand** | &amp; | & |
| **Menor que** | &lt; | < |
| **Mayor que** | &gt; | > |
| **Comillas** | &quot; | " |
| **Apóstrofe** | &apos; | ` |

Una herramienta online de código abierto que puede utilizarse para el escape de los caracteres especiales en XML y realizar el proceso inverso es XML Escape/Unescape.

**Ejemplo:** La primera imagen muestra una porción de código XML sin escapes




A continuación, el mismo código XML con los caracteres completamente escapados




### 4.2.3. Uso de nombres significativos para los identificadores

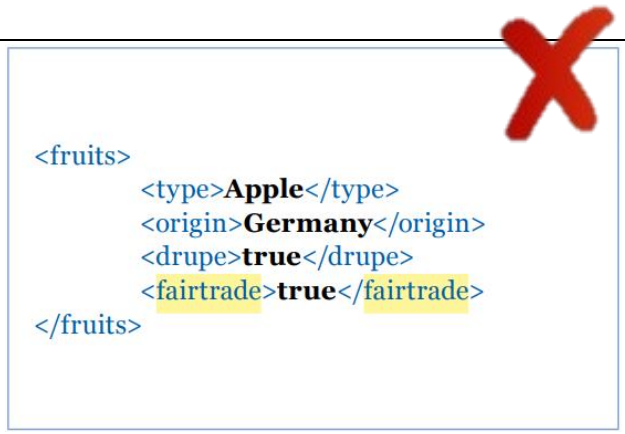
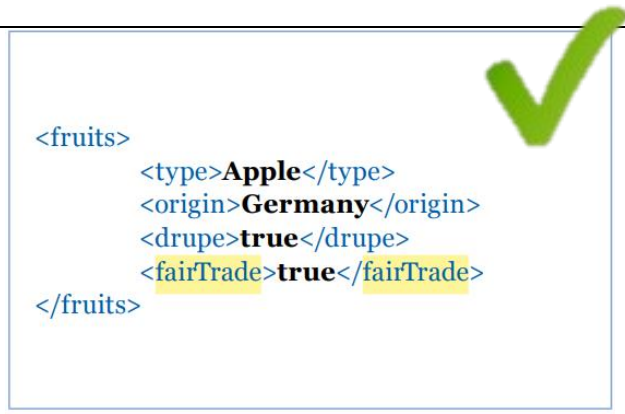
**Características de calidad afectadas:** Reusabilidad, Consistencia

**Recomendaciones:** Todos los identificadores, ya sean elementos o atributos, deben tener nombres significativos y ser utilizados de forma univoca, es decir, en ningún caso usar el mismo identificador para nombrar dos o más elementos. No hay recomendaciones oficiales sobre la forma de expresar identificadores, por lo que se puede utilizar, por ejemplo, el estilo de escritura camelCase o su

variante PascalCase, sin embargo, no deben mezclarse las diferentes formas. Además, es importante tener en cuenta que nunca deben utilizarse caracteres especiales en los identificadores.

La herramienta de código abierto Title Case puede ser útil para la conversión de frases compuestas por múltiples palabras en varios formatos de mayúsculas y minúsculas.

**Ejemplo:**

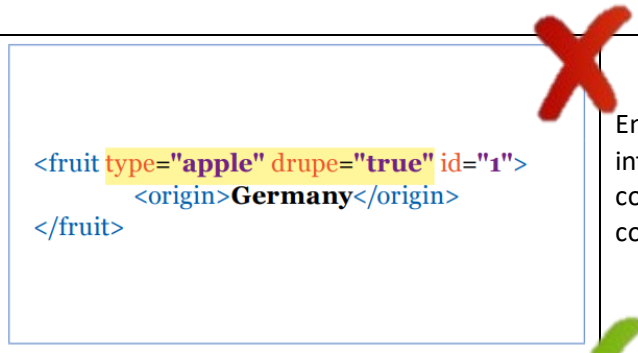
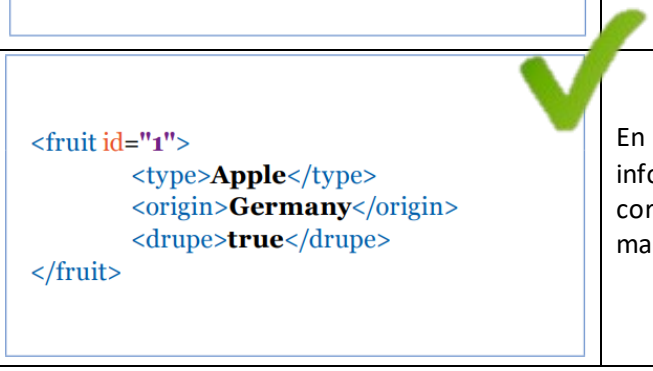
| | |
|---|---|
|  | **Este ejemplo muestra el código XML con el identificador 'fairtrade' que no está escrito siguiendo el estilo camelCase o PascalCase, lo que dificulta su lectura por parte de los usuarios y por tanto, propenso a errores de procesamiento.** |
|  | **Este ejemplo muestra el mismo código XML con el identificador 'fairTrade', compuesto por dos palabras concatenadas escrito siguiendo el estilo camelCase.** |

### 4.2.4. Utilizar correctamente los atributos y elementos

**Características de calidad afectadas:** Interoperabilidad, Conformidad, Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** Aunque no existe una directriz obligatoria sobre si los datos deben codificarse como elementos o atributos, el consorcio W3C establece unas recomendaciones a seguir para la publicación de conjuntos de datos en formato XML. Estas recomendaciones recogen como buena práctica que la información principal que forma parte del dato debe representarse mediante elementos y los metadatos o propiedades que contienen información adicional deben implementarse como atributos.

**Ejemplo:** Aunque ambas propuestas son correctas sintácticamente, la segunda es semánticamente más expresiva.

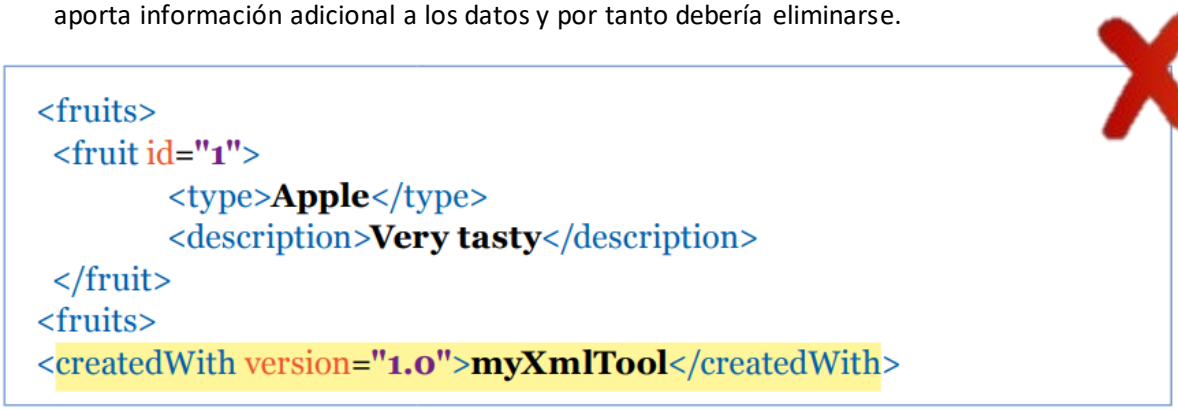
| | |
|---|---|
|  | En esta porción de código XML, parte de la información principal del dato se ha codificado como atributos, cuando se debería de haber codificado como elementos. |
|  | En esta modificación del ejemplo, la información principal se ha codificado correctamente como elementos y tan solo se mantiene como atributo el identificador. |

### 4.2.5. Eliminar datos específicos relativos al software de edición

**Características de calidad afectadas:** Interoperabilidad, Conformidad, Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** XML, como cualquier otro formato abierto, debe ser siempre independiente de los programas o herramientas utilizados para procesar los archivos. Esto permite al usuario elegir la herramienta que prefiera para procesar los datos sin tener que preprocesarlos con antelación.

**Ejemplo:** En el siguiente ejemplo podemos observar que el código XML contiene datos relativos al programa que se ha utilizado para la creación o el procesamiento del archivo. Esta información no aporta información adicional a los datos y por tanto debería eliminarse.



```
<fruits>
 <fruit id="1">
        <type>Apple</type>
        <description>Very tasty</description>
 </fruit>
<fruits>
```

## 4.3. Formato JSON

JSON es uno de los formatos estándar con mayor aceptación entre los reutilizadores profesionales. A continuación, procederemos a describir de forma no exhaustiva algunas pautas útiles aplicables en este formato.

### 4.3.1. Utilizar tipos de datos adecuados

**Características de calidad afectadas:** Interoperabilidad, Reutilización por maquinas/Procesabilidad.

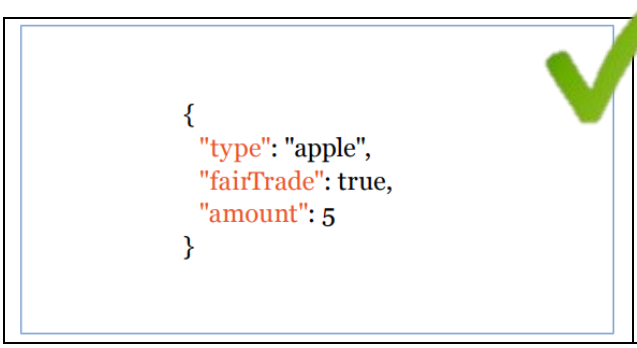
**Recomendaciones:** JSON, soporta principalmente seis tipos de datos: cadenas de caracteres, números, arrays, objetos, valores booleanos, y valor nulo (null). Para el procesamiento de datos es esencial utilizar los tipos adecuados. Por ejemplo, los números deben codificarse utilizando el tipo "number", y los valores booleanos con el tipo "boolean". De esta forma se evitan errores derivados de la codificación de valores no permitidos, por ejemplo, un valor distinto de "true", "false" o "null" para una variable booleana.

La herramienta online jsonlint ayuda a chequear si el código JSON es válido.

**Ejemplo:**

```
{
  "type": "apple",
  "fairTrade": "true",
  "amount": "5"
}
```

Se muestra código JSON con varios tipos de datos. En este caso toda la información ha sido codificada como cadenas de caracteres (string), independientemente del tipo de datos subyacentes.



```
{
  "type": "apple",
  "fairTrade": true,
  "amount": 5
}
```

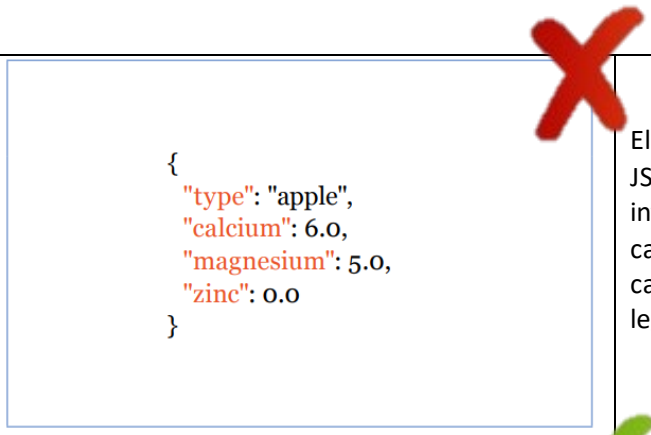
A la izquierda podemos observar el mismo archivo JSON, pero esta vez los datos han sido codificados atendiendo al tipo correcto de dato.

### 4.3.2. Utilizar jerarquías para agrupar datos

**Características de calidad afectadas:** Interoperabilidad, Reutilización por maquinas/Procesabilidad.

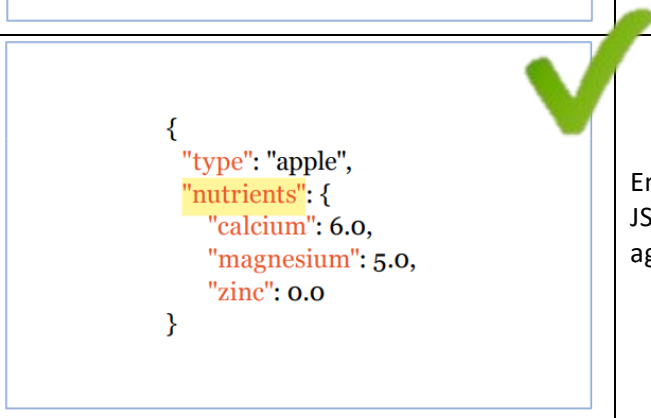
**Recomendaciones:** En lugar de vincular todos los campos al objeto raíz JSON, los datos deben agruparse semánticamente. Esta práctica mejora la legibilidad por parte de las personas y ayuda a mejorar el rendimiento al procesar el archivo. Prácticamente todos los editores de código permiten colapsar objetos y arrays, lo que permite que los usuarios puedan navegar más rápida e intuitivamente por el código.

**Ejemplo:**



```
{
  "type": "apple",
  "calcium": 6.0,
  "magnesium": 5.0,
  "zinc": 0.0
}
```

El código de la izquierda muestra un archivo JSON con datos agrupados. Toda la información está vinculada al objeto raíz. En casos donde exista un gran número de campos, esta forma de codificación reduce su legibilidad.



```
{
  "type": "apple",
  "nutrients": {
    "calcium": 6.0,
    "magnesium": 5.0,
    "zinc": 0.0
}
```

En esta alternativa se muestra el mismo JSON, pero con los datos semánticamente agrupados.

### 4.3.3. Utilizar arrays, solo cuando sea necesario

**Características de calidad afectadas:** Interoperabilidad, Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** Los datos sólo deben codificarse en arrays -colecciones, generalmente ordenadas de elementos-, si el tamaño de la lista es dinámico, es decir, no se conoce de antemano o está sujeto a cambios. Si no es el caso, el uso de campos explícitos facilita el procesamiento posterior. Además, no se puede garantizar que los valores de un array se proporcionen siempre en el mismo orden, lo que hace que los datos sean propensos a una interpretación errónea.

**Ejemplo:**

| | |
|---|---|
| {<br>"type": "apple",<br>"nutrients": [6.0, 5.0, 0.0]<br>} | En esta primera parte del ejemplo se está mostrando una sección de código JSON que utiliza un array, pero no se indican a qué tipo de nutrientes se refieren los valores. La especificación de estos campos, en este caso, habría sido necesaria. |
| {<br>"type": "apple",<br>"nutrients": {<br>"calcium": 6.0,<br>"magnesium": 5.0,<br>"zinc": 0.0<br>}<br>} | A la izquierda podemos ver el mismo ejemplo JSON en el que el uso del array está justificado. |

## 4.4. Formato RDF

RDF es uno de los formatos habituales para publicar datos enlazados (Linked data). En la Guía práctica para la publicación de datos enlazados en RDF encontrarás información muy útil sobre el formato y una forma práctica de transformar datos en CSV a RDF. Además de las indicadas en el apartado de pautas generales, a continuación, te vamos a describir de forma no exhaustiva algunas pautas útiles aplicables a este formato.

### 4.4.1. Utilizar URIs para identificar recursos en la web

**Características de calidad afectadas:** Conformidad/Cumplimiento, Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** Los identificadores de recursos deben ser URIs (Uniform Resource Identifier, por sus siglas en inglés), ya que idealmente esta forma de direccionamiento HTTP permite el acceso directo al recurso en cuestión. También hacen que los recursos sean indexables por los motores de búsqueda, lo que mejora su capacidad de búsqueda. Sin embargo, esto solo se aplica si estos identificadores son persistentes y no contienen información volátil, por ejemplo, las credenciales de una persona denotada por una cadena de texto.

**Ejemplo:**

| | |
|---|---|
| <vcard:hasAddress<br>rdf:resource="myAddress"> | Se muestra un recurso en RDF/XML expresado mediante una cadena de caracteres |
| <vcard:hasAddress<br>rdf:resource="http://www.w3.org/2006/vcard/ns#myAddress"> | En este caso el recurso se expresa mediante una URI univoca y persistente |

## 4.4.2. Utilizar espacios de nombres

**Características de calidad afectadas:** Consistencia.

**Recomendaciones:** Un espacio de nombres es en esencia, un mapeo que vincula un prefijo a un URI de un vocabulario, lo que permite una representación de RDF más legible para las personas. Si bien los espacios de nombres no son necesarios para procesar RDF, reducen su verbosidad y el tamaño del archivo. Por otro lado, y de forma similar a las recomendaciones vistas para XML, los identificadores de clases deben escribirse en PascalCase, mientras que los identificadores de propiedades generalmente se escriben en camelCase.

Una herramienta útil, gratuita y de código abierto, orientada a la transformación de datos y que a la vez permite generar grafos RDF es OpenRefine.

**Ejemplo:**

| | |
|---|---|
| <rdf:rdf><br><rdf:description rdf:about="http://myresource"><br><http://mynamespace#myproperty>Sample<br></http://mynamespace#:myproperty><br></rdf:description><br></rdf:rdf> | El RDF/XML de la izquierda no usa espacios de nombres ni estilos de escritura para clases y propiedades, lo que puede dar lugar a una menor legibilidad |
| <rdf:RDF<br>xmlns:myNamespace="http://myNamespace#"> | Este RDF/XML utiliza espacios de nombres y |

| | |
|---|---|
| <rdf:Description rdf:about="http://myResource"><br><myNamespace:myProperty>Sample</myNamespace:myProperty><br></rdf:Description><br></rdf:rdf> | PascalCase para la clase "Description" y camelCase para la propiedad "myProperty" |

### 4.4.3. Utilizar vocabularios controlados siempre que sea posible

**Características de calidad afectadas:** Conformidad/Cumplimiento, Reutilización por maquinas/Procesabilidad.

**Recomendaciones:** En la pauta general "3.6 estandarizar valores de datos" ya se mencionó la importancia de utilizar vocabularios controlados como mecanismo para normalizar la estructura y los valores de los campos de un dataset. En el contexto de los datos enlazados la reutilización de los vocabularios controlados existentes es un requisito para garantizar la interoperabilidad de datos y deben utilizarse siempre que sea posible. La Oficina de Publicaciones de la Unión Europea proporciona numerosos vocabularios para su uso en el ámbito de las administraciones públicas. DCAT-AP es el vocabulario controlado que se utiliza para la interoperabilidad de catálogos de datos abiertos en Europa. Otras referencias útiles son los vocabularios aplicables en contexto de ciudades abiertas o para un propósito más general, los relacionados en el sitio web Linked Open Vocabularies (LOD).

**Ejemplo:** En el ejemplo se observan dos formas -validas sintácticamente en ambos casos- pero en el segundo caso el uso del vocabulario controlado "license" es más eficiente.

```
<dct:accrualPeriodicity>
    <dct:Frequency rdf:about="http://dataset/Frequency">
        <rdf:value>
            <time:DurationDescription rdf:about="htpp://dataset/DurationDescription">
                <time:years rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"1</time:years>
                    </time:DurationDescription>
        </rdf:value>
    </dct:Frequency>
</dct:accrualPeriodicity>
```

```
<dct:accrualPeriodicity>
        <dct:Frequency rdf:about="http://publications.europa.eu/resource/authority/frequency/DAILY"/>
</dct:accrualPeriodicity>
```

## 4.5. APIs

Los interfaces de programación de aplicaciones (APIs) son los mecanismos de acceso a datos más eficientes para consumir datos de alto valor, dinámicos o con alta frecuencia de actualización. En la Guía práctica para la publicación de datos abiertos usando APIs encontrarás información muy útil para diseñar e implementar APIs basadas en el diseño arquitectónico REST (Representational State Transfer). Entre otras cuestiones relevantes conocerás la especificación OpenAPI, cómo interpretar y gestionar métodos y códigos de estado HTTP, ajustar correctamente cabeceras, usar estrategias de filtrado y paginado para el acceso a grandes cantidades de datos o cómo generar una documentación completa de una API. Además, no olvides que las recomendaciones indicadas en el apartado de pautas generales son aplicables a este mecanismo de acceso. A continuación, te resumimos algunas de las pautas específicas que debes tener en cuenta para garantizar una buena calidad en la puesta a disposición de APIs:

### 4.5.1. Documentar la API

**Características de calidad afectadas:** Comprensibilidad /Interpretabilidad.

**Recomendaciones:** Los principales usuarios de las APIs son desarrolladores de aplicaciones y servicios y la documentación de la API es el primer punto de contacto para conocer la calidad y utilidad de la misma. Si la documentación de la API está completa, actualizada y resulta fácil de entender, favorecerá un uso más eficiente.

El contenido de la documentación debe reflejar inequívocamente cómo se realizan peticiones a la API, cuáles son los parámetros necesarios y cuál será la salida esperada. Asimismo, es fundamental reflejar con claridad que nuevas funcionalidades o resolución de problemas incorpora cada nueva versión disponible de la API.

Igualmente, la documentación de la API debe detallar sin ambigüedad qué recursos estará el usuario autorizado a utilizar y el método de autenticación de usuario utilizado para garantizar un acceso seguro, si éste es requerido.

Al menos, forman parte de la estructura de contenidos de la documentación de referencia de la API, los siguientes epígrafes:

- Si es requerido, **método de autenticación** utilizado para autorizar el acceso y utilización de los recursos provistos.
- Listado completo de las **peticiones que la API puede manejar, incluyendo el propósito de cada una, los parámetros permitidos, y la salida** esperada.
- **Ejemplos de uso** de cada una de las peticiones posibles escritos en diferentes lenguajes de programación preferentes por la comunidad de desarrolladores en cada momento
- **Relación de versiones de la API y las características** incorporadas en cada una.
- Información de **contacto con los promotores de la API** y mecanismo de contacto para proporcionar feedback sobre errores, sugerencias o preguntas sobre cualquier aspecto

- Es recomendable acompañar la documentación de algún mecanismo online que **permita testar las peticiones y comprobar la respuesta de la API.**

Un estándar utilizado para describir API es la especificación OpenAPI. Permite utilizar JSON o YAML para describir las API. Para crear y validar especificaciones OpenAPI se puede utilizar el editor en línea Swagger (comercial/código abierto).

### 4.5.2. Definir interpretaciones comprensibles de códigos de estado

**Características de calidad afectadas:** Comprensibilidad /Interpretabilidad.

**Recomendaciones:** Los códigos de estado devueltos por el protocolo HTTP constituyen una información esencial para el desarrollador dado que sirven para distinguir entre respuestas satisfactorias e insatisfactorias a las invocaciones realizadas a la API. Por lo tanto, las respuestas derivadas de las peticiones que realizan los clientes de la API deben ser informativas, comprensibles por las personas y legibles por las máquinas. Es importante utilizar los códigos de estado del estándar HTTP en la implementación de la gestión de respuesta de la API ya que además son reconocidos por los frameworks de desarrollo de aplicaciones habituales.

Los valores de los códigos de estado se agrupan en torno a cinco categorías principales de información. A continuación, se reproducen los más habituales y su interpretación:

| | | |
|---|---|---|
| **1xx** | Informativo | Solicitud recibida, el proceso de respuesta está en marcha. |
| **2xx** | Éxito | La petición del cliente se ha recibido, entendido y aceptado con éxito. |
| **3xx** | Redirección | Se deben tomar medidas adicionales para completar la solicitud. |
| **4xx** | Error del cliente | La solicitud contiene una sintaxis incorrecta o no se puede cumplir. |
| **5xx** | Error del servidor | El servidor no pudo resolver una solicitud aparentemente válida |

### 4.5.3. Utilizar cabeceras HTTP para el intercambio de información

**Características de calidad afectadas:** Conformidad, Comprensibilidad /Interpretabilidad.

**Recomendaciones:** las cabeceras HTTP son el medio fundamental para establecer una adecuada negociación de contenido, pero además tienen otras funciones relevantes para enviar información adicional en el cuerpo de una petición o respuesta. Esta información adicional no forma parte de la

carga útil real del recurso que se solicita y en ella se puede codificar información de interés para los consumidores de los datos.

Las cabeceras tienen la forma de pares clave-valor y están formadas por su nombre (no sensible a las mayúsculas) seguido de dos puntos ':', y su valor (sin saltos de línea).

Las cabeceras pueden ser agrupadas de acuerdo con sus contextos:

- **Cabecera general:** Cabeceras que se aplican tanto a las peticiones como a las respuestas, pero sin relación con los datos que finalmente se transmiten en el cuerpo. Por ejemplo: "Connection", "Cache-control", etc.
- **Cabecera de consulta:** Cabeceras que contienen más información sobre el recurso que se solicita. Por ejemplo: "Accept", "Accept-charset", etc.
- **Cabecera de respuesta:** Cabeceras que contienen más información sobre el contenido, como su origen o el servidor (nombre, versión, etc.). Por ejemplo: "Content-length", "Content-encoding", etc.
- **Cabecera de entidad:** Cabeceras que contienen más información sobre el cuerpo de los recursos, como el tamaño del contenido o su tipo MIME. Por ejemplo: "Content-type", "Content-Language", etc.

**Ejemplo:** A continuación, se observa una cabecera de contexto de respuesta que indica la lista de métodos de peticiones aceptadas por un servidor. Esta cabecera la envía el servidor si éste responde con un código de estado 405 (Method Not Allowed):

| | |
|---|---|
| Sintaxis: | Allow: <http-methods> |
| Ejemplo de uso: | Allow: GET, HEAD |

### 4.5.4. Utilizar paginado para servir grandes cantidades de datos

**Características de calidad afectadas:** Eficiencia, Comprensibilidad /Interpretabilidad.

**Recomendaciones:** Solicitar grandes cantidades de datos puede generar fácilmente significativas cargas en el servidor. Puede ocurrir que el usuario no requiera todos los datos, o no todos a la vez. Para reducir esta carga y aumentar los tiempos de respuesta, se debe utilizar la paginación cuando corresponda. Esto significa que se sirven porciones de datos en lugar de un conjunto de datos completo. El cliente puede indicar en la solicitud qué porción recuperar, así como su tamaño. Esto generalmente se logra utilizando los parámetros offset (indica el punto donde comienza la porción de datos solicitada) y limit (indica el número de registros de datos a recuperar).

**Ejemplo:** La invocación a una API como la que se muestra a continuación devolverá de la colección de contratos, los registros entre el 51 y el 75.

```
https://datos.ejemplo.com/v1/licitaciones/contratos?offset=50&limit=25&estado=finalizado
```

# 5. Recomendaciones para la estandarización y enriquecimiento de datos

A medida que aumenta el volumen de datos disponibles, la estandarización es cada vez más relevante. La reutilización de datos puede verse perjudicada si previamente a su procesamiento es necesario convertirlos a una estructura o formato común, es decir, llevar a cabo tareas de preprocesamiento consistentes en depuración y transformación de datos. Por tanto, la estandarización aumenta la procesabilidad de los datos.

Por otro lado, el enriquecimiento es el concepto que implica vincular datos de fuentes externas a conjuntos de datos existentes. Estos datos pueden provenir, entre otros, de fuentes públicas o bases de conocimiento abiertas. La vinculación de datos puede aumentar el valor inicial de los datos al crear nuevas relaciones y, por lo tanto, permitir nuevos tipos de análisis. Por ejemplo, si se publica la base de datos de bienes de interés cultural de un determinado territorio se puede enriquecer con datos procedentes de una base de conocimiento abierto como es Wikidata para incluir referencias a los lugares donde se ubican tales elementos.

El objetivo de esta sección es aportar a los publicadores de datos recomendaciones prácticas que les permitan publicar conjuntos de datos con un alto nivel de estandarización y enriquecimiento. A continuación, se detallan recomendaciones sobre cómo reutilizar conceptos de vocabularios controlados, cómo referenciar recursos mediante el uso de identificadores uniformes de recursos (URIs) y cómo sacar partido de ellos resolviendo traducciones de datos, para cerrar la sección sobre el proceso a seguir para vincular datos externos enriqueciendo los datos de partida.

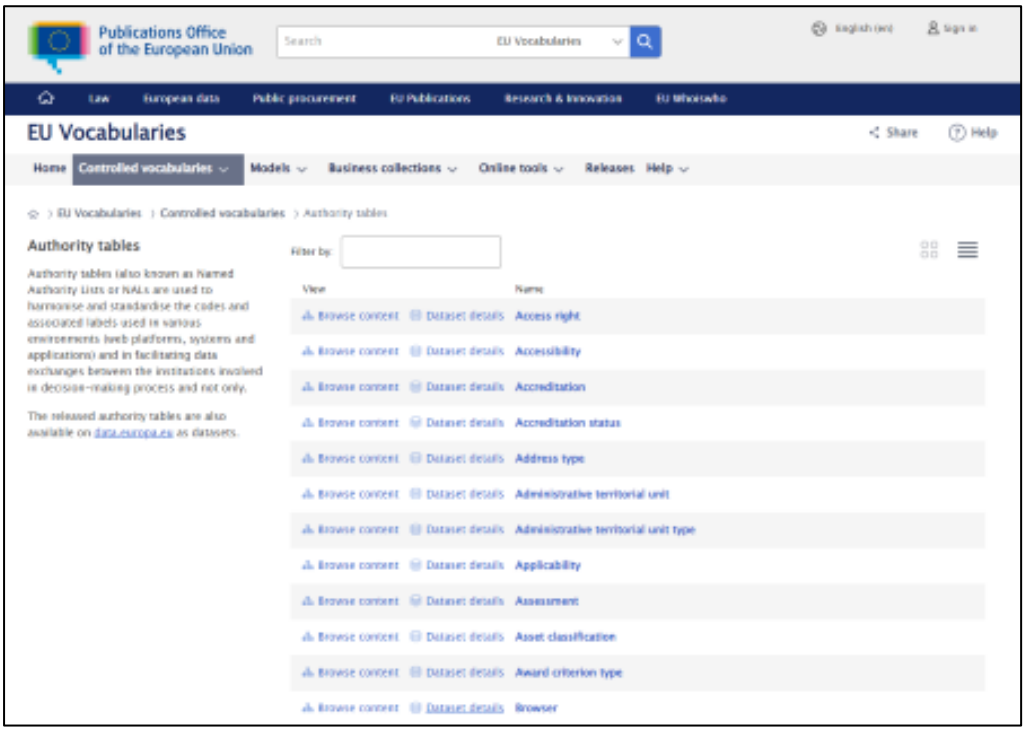
## 5.1. Reutilizar conceptos de vocabularios controlados

Se puede lograr un mayor nivel de estandarización de datos integrando en los datos vocabularios RDF que se expresan como listas de códigos, taxonomías, clasificaciones o terminologías. Estos vocabularios controlados describen, identifican y organizan los conceptos sin ambigüedad en su área de especialización y pueden reutilizarse tanto para armonizar como para enriquecer los datos.

En los vocabularios RDF, cada concepto se identifica mediante un identificador único de recurso (URI), lo que permite a cualquier sistema hacer referencia a él sin ambigüedad. Esta característica es importante, ya que permite referenciar estos conceptos desde cualquier lugar una vez que se han publicado en la web. Estas referencias incluidas en los conjuntos de datos forman una extensa red de datos vinculados, es decir, la conocida como web semántica.

**Ejemplo:** la clasificación multilingüe europea de capacidades, competencias, cualificaciones y ocupaciones (ESCO) funciona como un diccionario que describe, identifica y clasifica las ocupaciones, capacidades y cualificaciones profesionales relevantes para el mercado laboral, la educación y la formación en la UE. Los conceptos referidos en esta clasificación se reutilizan en diferentes plataformas de empleo que asocian solicitantes de empleo con puestos de trabajo en función de las capacidades de los solicitantes o sugieren acciones formativas a personas que deciden mejorar su curriculum profesional.

Como se ha mencionado con anterioridad, la Oficina de Publicaciones mantiene una serie de vocabularios y tablas de datos maestros de la UE útiles para la publicación de datos abiertos del sector público.




Por otro lado, cabe reseñar la iniciativa de Ciudades Abiertas que propone una serie de vocabularios creados para su utilización por parte de los Ayuntamientos participantes y de cualquier otra entidad que los considere de utilidad. A fecha de publicación de esta guía, el listado de vocabularios disponibles abarca los siguientes:

- Censo de locales y terrazas, así como sus actividades económicas y licencias de apertura asociadas
- Agenda Municipal
- Padrón de Habitantes
- Cubos de Datos del Padrón de Habitantes
- Bicicleta pública
- Autobuses urbanos
- Tráfico
- Empleo
- Presupuesto y Ejecución Presupuestaria
- Convenios

- Contaminación Acústica
- Deuda Pública Financiera
- Deuda Pública Comercial
- Subvenciones

Adicionalmente el proyecto ha elaborado dos vídeos que ilustran tanto los beneficios de la utilización de vocabularios para la representación de datos abiertos como la metodología utilizada en el proyecto para la definición de estos.

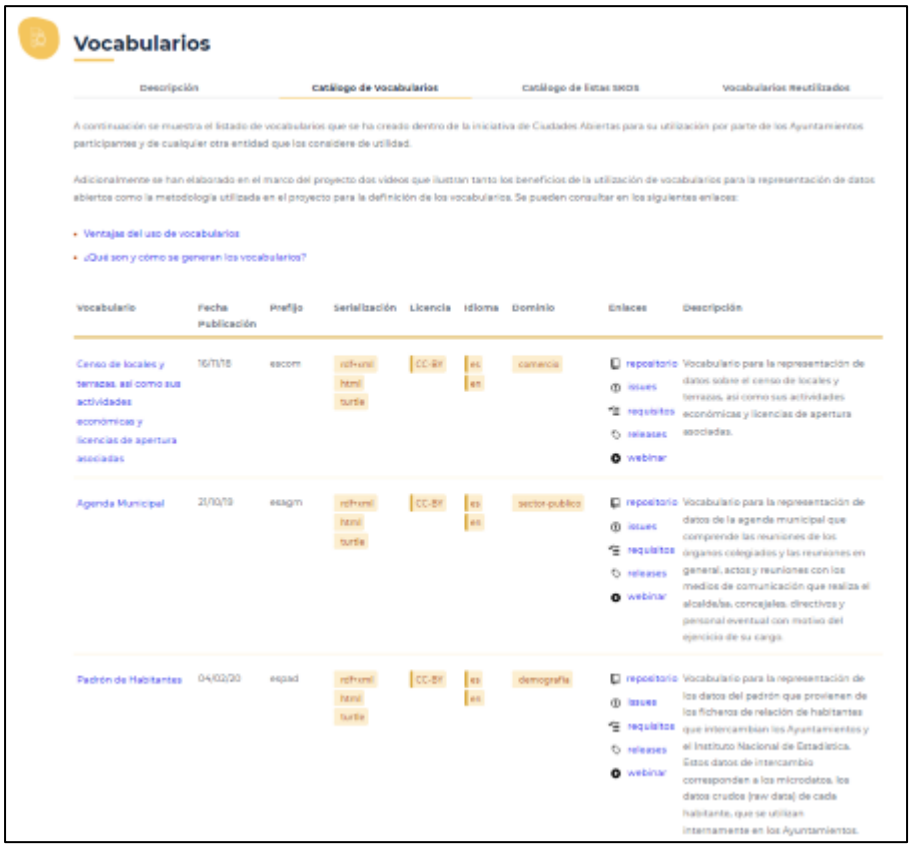



*Ilustración 1.- Catálogo de vocabularios del proyecto de ciudades abiertas [https://ciudades-abiertas.es/]*

En esta línea de acciones, otra actuación relevante es la llevada a cabo por la iniciativa SmartDataModels (SDM), que propone un amplio repositorio de modelos de datos ampliamente adoptados en diferentes dominios. Se basan en escenarios de casos reales o en el mapeo de estándares abiertos adoptados en los diferentes contextos de aplicación. Los modelos de datos juegan un papel crucial porque definen los formatos de representación armonizados y la semántica que utilizarán las aplicaciones para consumir y publicar datos.

SDM clasifica la información por ámbitos de aplicación, creando un repositorio para cada uno de ellos. Cada dominio contiene submódulos con los temas relevantes para ese dominio y, dentro de cada tema, los modelos de datos relacionados. También dispone de elementos transversales compartidos para

todos los dominios. Para facilitar la compartición y el entendimiento común, cada modelo incluye tres elementos:

- las especificaciones e información relacionada en diferentes idiomas
- la definición del esquema específico en JSON, incluyendo la descripción de cada propiedad, el tipo, valores válidos y otros elementos.
- Ejemplos en diferentes formatos.

Además de su carácter público y uso gratuito, se pone a disposición de los usuarios la posibilidad de realizar modificaciones en caso de que lo consideren necesario, así como compartir dichas modificaciones con el resto de los usuarios.

## 5.2. Utilizar identificadores únicos

La web es, por su propia naturaleza, una red descentralizada de datos y documentos que se identifican de forma única mediante identificadores de recursos uniformes (URI) sobre el protocolo de comunicación de la web HTTP. Una buena práctica ya comentada con anterioridad en las pautas generales de calidad consiste en codificar valores de datos utilizando URIs. Pero hay que tener en cuenta que la web es una plataforma dinámica donde los cambios en los identificadores asignados en un sistema de TI podrían afectar negativamente a otros sistemas de TI si se producen cambios en los identificadores originales. Es decir, la web es sensible a los cambios en los identificadores. Por lo tanto, es necesario asegurarse de que los URI se mantienen estables y persistentes para que, cuando se utilicen para codificar valores de datos o realizar búsquedas, siempre apunten al mismo recurso. Para ello, es necesario dotarse de unas pautas para el diseño, uso y mantenimiento de URI estables con el objetivo de mantener una identificación permanente e inalterable de los recursos digitales en un marco público común.

Una excelente referencia para conocer en detalle el diseño y gestión de URIs persistentes se encuentra en este módulo de formación publicado en data.europa.eu.

Uno de los valores que aporta esta práctica es la harmonización de tablas de datos: en lugar de ajustar valores en datos, se puede hacer referencia a tales valores utilizando URIs. De esta forma, cualquier tabla que utilice tales valores lo hará de manera harmonizada. Esto significa que, si esos valores cambian, no es necesario ajustar la referencia en cada tabla, lo que reduce la carga de mantenimiento para los publicadores de datos.

**Ejemplo:** a continuación, se observa una muestra de datos con estadísticas sobre estudiantes Erasmus.

| Estudiante-id | Localidad-institución | País-Institución | Edad-estud | Género-estud | Nacionalidad-estud |
|---|---|---|---|---|---|
| **E1** | E Alicant01 | ES | 21 | F | ES |
| **E5** | D Koln07 | DE | 22 | F | DE |

| Estudiante-id | Localidad-institución | País-Institución | Edad-estud | Género-estud | Nacionalidad-estud |
|---|---|---|---|---|---|
| **E7** | SF Turku01 | FI | 21 | M | FI |

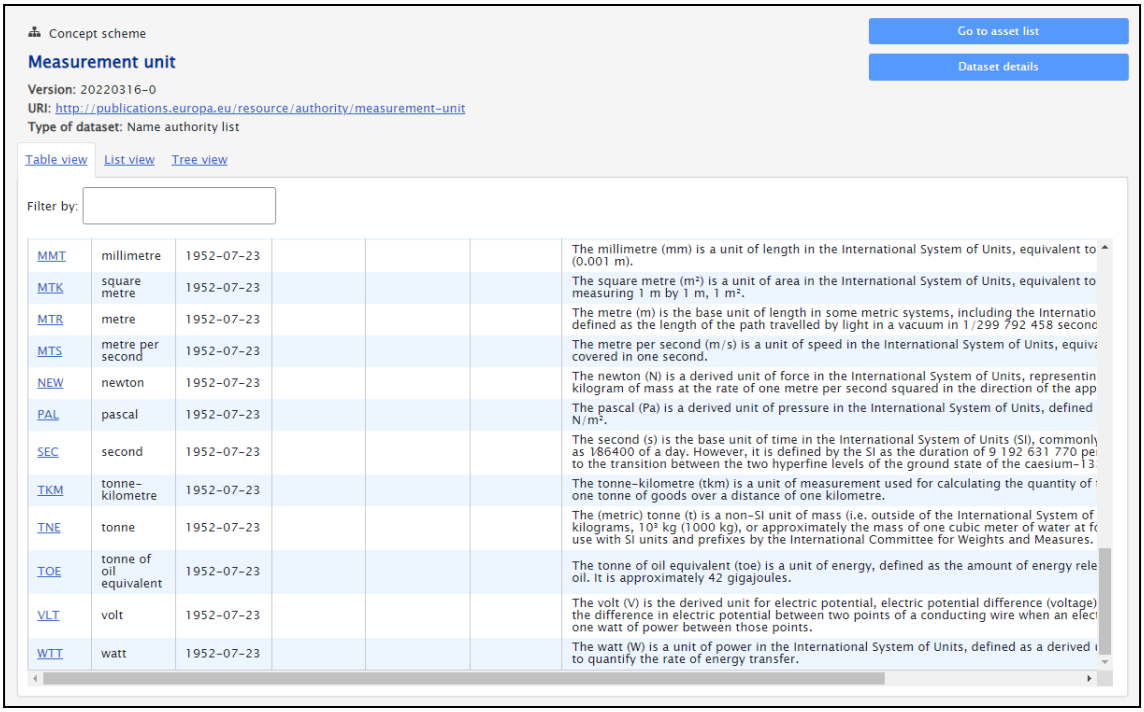
Los valores proporcionados en los campos “país-institución” o "nacionalidad-estud" se pueden estandarizar en función de la tabla País que se muestra a continuación. En lugar de codificar directamente en la tabla el código de país (ES, DE o FI), se pueden proporcionar los identificadores únicos correspondientes para estos países derivándose además, datos adicionales del identificador de cada país, como una etiqueta o el código ISO del país utilizando dos o tres letras.

| Home-inst-ctry | Id-Único | Etiqueta | ISO-31662 | Iso-31663 |
|---|---|---|---|---|
| **ES** | http://publications.europa.eu/resource/authority/country/ESP | Spain | ES | ESP |
| **DE** | http://publications.europa.eu/resource/authority/country/DEU | Germany | DE | GER |
| **FI** | http://publications.europa.eu/resource/authority/country/FIN | Finland | FI | FIN |

## 5.3. Facilitar la traducción de etiquetas de datos

Una vez que los valores de datos se ajustan utilizando identificadores únicos de vocabularios controlados, los URI pueden ser desreferenciados. Que una URI sea desreferenciable significa que la referencia abstracta que representa devuelve un valor concreto del objeto o concepto identificado. Uno de los valores asociados a este principio de los datos enlazados es que la etiqueta asociada con un concepto se resuelva en cualquier idioma compatible con el vocabulario controlado.

Ejemplo: el concepto de "metro" está definido en el vocabulario controlado unidades de medida del repositorio de la Oficina de Publicaciones de la Unión Europea.




Como se ve en la porción de código a continuación, "metro" está representado por diferentes etiquetas (prefLabel) en las diferentes lenguas oficiales de la UE. La asignación de la URI del concepto de "metro" en un conjunto de datos permite la desreferenciación automática de las diferentes versiones lingüísticas ofreciendo un acceso más versátil a los datos. Además, si se procede a realizar cualquier actualización sobre las traducciones contenidas en el vocabulario, no es necesario actualizar el concepto de "metro" en la tabla de datos. El URI desreferenciará automáticamente el valor correcto de la tabla.

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:ns2="http://purl.org/dc/terms/"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:skos="http://www.w3.org/2004/02/skos/core#"
        xmlns:ns5="http://publications.europa.eu/ontology/euvoc#"
        xmlns:ns6="http://lemon-model.net/lemon#"
        xmlns:ns7="http://www.w3.org/2008/05/skos-xl#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:ns9="http://publications.europa.eu/ontology/authority/"
        xmlns:ns10="http://publications.europa.eu/resource/authority/" >
 <rdf:Description rdf:about="http://publications.europa.eu/resource/authority/measurement-unit/MTR">
  <rdf:type rdf:resource="http://publications.europa.eu/ontology/euvoc#MeasurementUnit" />
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept" />
  <ns2:created rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2019-01-10</ns2:created>
  <ns2:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2022-02-17</ns2:modified>
  <owl:versionInfo>20220316-0</owl:versionInfo>
  <skos:inScheme rdf:resource="http://publications.europa.eu/resource/authority/measurement-unit" />
  <skos:prefLabel xml:lang="de">Meter</skos:prefLabel>
  <skos:prefLabel xml:lang="nl">meter</skos:prefLabel>
  <skos:prefLabel xml:lang="en">metre</skos:prefLabel>
  <skos:prefLabel xml:lang="el">μέτρο</skos:prefLabel>
  <skos:prefLabel xml:lang="sk">meter</skos:prefLabel>
```

```
<skos:prefLabel xml:lang="ro">metru</skos:prefLabel>
<skos:prefLabel xml:lang="lv">metri</skos:prefLabel>
<skos:prefLabel xml:lang="fr">mètre</skos:prefLabel>
<skos:prefLabel xml:lang="da">meter</skos:prefLabel>
<skos:prefLabel xml:lang="lt">metras</skos:prefLabel>
<skos:prefLabel xml:lang="cs">metr</skos:prefLabel>
<skos:prefLabel xml:lang="hr">metar</skos:prefLabel>
<skos:prefLabel xml:lang="pl">metr</skos:prefLabel>
<skos:prefLabel xml:lang="bg">метър</skos:prefLabel>
<skos:prefLabel xml:lang="sl">meter</skos:prefLabel>
<skos:prefLabel xml:lang="pt">metro</skos:prefLabel>
<skos:prefLabel xml:lang="sv">meter</skos:prefLabel>
<skos:prefLabel xml:lang="ga">méadar</skos:prefLabel>
<skos:prefLabel xml:lang="hu">méter</skos:prefLabel>
<skos:prefLabel xml:lang="et">meeter</skos:prefLabel>
<skos:prefLabel xml:lang="es">metro</skos:prefLabel>
<skos:prefLabel xml:lang="mt">metru</skos:prefLabel>
<skos:prefLabel xml:lang="fi">metri</skos:prefLabel>
<skos:prefLabel xml:lang="it">metro</skos:prefLabel>
```

## 5.4. Vincular y enriquecer datos

El uso consistente de identificadores únicos también permite la vinculación y el enriquecimiento utilizando datos externos. Esto agrega valor a los datos al vincular nuevos conceptos o aspectos de los datos existentes. El uso óptimo de vocabularios controlados y enriquecimiento de datos se puede lograr utilizando un formato de datos de cuatro o cinco estrellas de la escala de Tim Berners-Lee, como RDF o JSON-LD.

Ejemplo: Uno de los contenidos que se describe en la Guía práctica para la publicación de datos enlazados en RDF es el proceso para enlazado de un archivo CSV con fuentes externas. Para ello, se utiliza la herramienta OpenRefine que permite enlazar recursos que tengamos en el CSV con fuentes externas como Wikidata, lo que permite obtener un conjunto de datos 5 estrellas.

OpenRefine es una poderosa herramienta para la gestión y enriquecimiento de datos (software código abierto). Existe mucha documentación sobre esta herramienta, pero te sugerimos este artículo que describe en detalle cómo descubrir inconsistencias en datos y cómo diagnosticar su precisión. Por otro lado, OntoRefine es una herramienta que, entre otras funcionalidades, también te permitirá llevar a cabo transformaciones para convertir datos tabulares en RDF (software comercial/código abierto).

# 6. Recomendaciones para la documentación de datos

Cada día aumenta la disponibilidad de datos para la reutilización. Una práctica esencial para mejorar la interoperabilidad y facilitar el procesamiento posterior, es que los datos deben documentarse. Una buena documentación o diccionario de datos implica que los usuarios pueden conocer de antemano qué se espera tanto de la sintaxis (es decir, la estructura) como de la semántica (es decir, el contenido) de los datos. Por otro lado, la documentación contribuye a incrementar el valor de los datos aportando contexto y evitando malinterpretaciones que pueden dificultar la reutilización.

El objetivo de esta sección es aportar recomendaciones prácticas que cubran las tareas involucradas en la documentación de datos. Estas tareas incluyen documentar la estructura y la semántica, así como llevar un adecuado control de versiones.

La sección se inicia con una pauta general que explica dónde publicar la documentación. A continuación, se abordan recomendaciones sobre el uso de esquemas como mecanismo fundamental para documentar las estructuras¬ de los datos. Además de la estructura, como se ha indicado, también debe documentarse la semántica de los datos, que se trata posteriormente.

## 6.1. Dónde publicar la documentación de los datos

Los temas tratados en esta sección incluyen principalmente pautas para la descripción de estructuras de datos, es decir, la representación interna de archivos y el seguimiento de los cambios que se producen en los datos a lo largo del tiempo. Aunque no es objeto de este documento, el desarrollo de un plan de gestión de datos (DMP, por sus siglas en inglés) que describa cómo se deben manejar los datos es una práctica necesaria antes de publicar datos abiertos. Un DMP aborda, entre otras, cuestiones del tipo: dónde publicar datos, dónde almacenar metadatos, qué formato usar y qué estándares seguir. Implementar un DMP antes de publicar¬ es importante para definir estructuras de datos coherentes y aspectos sustanciales para un procesamiento eficiente de los datos.

Independientemente del formato o tipo de archivo utilizado para documentar datos, es vital que esta documentación se publique junto con los datos, idealmente en una distribución separada. Esta distribución debe vincularse a los datos en sí a través de la propiedad *dct:conformsTo* de la especificación estándar DCAT-AP a nivel de dataset y de las distribuciones vinculadas, o por la propiedad *dct:references* definida en la NTI-RISP a nivel de conjunto de datos.

**Ejemplo**: Se muestra una distribución que utiliza la propiedad *dct:conformsTo* para vincular otra distribución que contiene un esquema que especifica la estructura de los datos.

```
<rdf:Description rdf:about="http://data.europa.eu/distribution/truck_data">
    <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Distribution"/>
        <dcterms:conformsTo rdf:resource="http://data.europa.eu/distribution/truck_data_schema"/>
        <dcterms:title xml:lang="en">Truck parking static data</dcterms:title>
        <dcterms:format rdf:resource="http://publications.europa.eu/resource/authority/file-type/XML"/>
</rdf:Description>
```

## 6.2. Utilizar esquemas para especificar la estructura de datos

Aunque las especificaciones de los formatos de datos (XML, JSON, etc.) definen directrices para ajustar la estructura interna con respecto a la sintaxis, el uso de determinadas palabras reservadas o identificadores permitidos, son los publicadores de datos quienes deciden la forma en que los datos se escriben o serializan en archivos para su procesamiento posterior. Esta serialización debe ser interpretable por el usuario mediante esquemas o modelos de datos. En lugar de esperar que el usuario descargue y analice los datos, el esquema de serialización o modelo de datos se puede especificar por separado, normalmente utilizando formatos adecuados para la especificación de modelos de datos. Veamos a continuación, cómo especificar esquemas de datos utilizando los lenguajes de modelado más habituales, JSON, XML, CSV y RDF.

## 6.3. Cómo especificar estructuras de datos JSON

El lenguaje de modelado utilizado para los archivos JSON se denomina JSON Schema. Los esquemas son archivos JSON en sí mismos, pero contienen información que describe una estructura de datos que se codifica como JSON. Los publicadores de datos deben publicar un JSON Schema que especifique la estructura JSON junto con sus datos.

**Ejemplo:** Se muestra un ejemplo de documentación de una API, siguiendo la especificación OpenAPI, usando JSON Schema que sirve datos sobre empleados en la que se muestra el tipo de datos del nombre del empleado y las restricciones sobre el rango de edades posibles.

```
{
        "$schema": "http://json-schema.org/draft-07/schema#",
        "description": "Employee data",
        "type": "object",
        "properties": {
                "name": {
                        "type": "string"
                        },
                "age": {
                        "type": "number",
                        "minimum": 0,
                        "maximum": 100
                }
        }
}
```

Algunas herramientas útiles para generar y validar esquemas JSON, son JSON schema y JSON schema generator que permite generar esquemas a partir de datos JSON.

## 6.4. Cómo especificar estructuras de datos XML

Existen varios lenguajes de modelado para especificar la estructura de los archivos XML, por ejemplo, RELAX NG y Schematron. XSD (XML Schema Definition Language) es la especificación recomendada por el W3C. Un archivo XSD es en sí mismo un XML. Se compone de dos partes: estructuras y tipos de datos. Como sus nombres sugieren, el primero define la parte estructural de XSD, mientras que el segundo define los tipos de datos que se pueden usar en XSD. En general, XSD especifica exactamente qué elementos/atributos están permitidos y qué tipo de datos debe tener el contenido. También es posible especificar patrones para comprobar la validez de las estructuras de datos, como por ejemplo, los códigos postales. Los publicadores de datos deben publicar esquemas XSD que especifiquen la estructura XML junto con los datos en XML.

**Ejemplo:** El código que se muestra a continuación contiene una porción de un esquema XSD que especifica la estructura de los datos que describen una fruta. Por ejemplo, establece que el valor "pepita" puede ser verdadero o falso, no desconocido.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
        <xs:element name="fruit">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="type"/>
                                <xs:element ref="description"/>
                                <xs:element ref="pepita"/>
                        </xs:sequence>
                        <xs:attribute name="id" use="required" type="xs:integer"/>
                </xs:complexType>
        </xs:element>
        <xs:element name="type" type="xs:NCName"/>
                <xs:element name="description" type="xs:string"/>
                <xs:element name="pepita" type="xs:boolean"/>
</xs:schema>
```

Ejemplos de herramientas útiles son los editores de esquemas XML: Liquid Studio, que permite la generación de archivos XSD a partir de XML existente (software comercial) y XMLFox, que cuenta con validación XSD (software comercial y código abierto).

## 6.5. Cómo especificar estructuras de datos CSV

Frictionless Data ha desarrollado un esquema de tabla CSV expresable en JSON. Esto significa que la estructura con la que debe ser coherente un archivo CSV se describe en un archivo JSON. Por el momento, la herramienta para crear esquemas de datos Frictionless solo está disponible como librerías para varios lenguajes de programación (Python, Javascript, Java, etc.), no obstante, dado que Frictionless Data se especifica mediante JSON, se puede utilizar cualquier editor de texto compatible con JSON para esta tarea.

Por su parte, el UK National Archives ha desarrollado varias herramientas que están disponibles públicamente bajo varias licencias de código abierto. Entre ellas un lenguaje de esquema CSV y un validador que se pueden utilizar para describir y validar el contenido de archivos CSV. Entre otras

cuestiones, se puede utilizar para especificar el número de columnas por tabla, si los valores son obligatorios u opcionales y qué rango de datos se aplica en cada campo.

Un publicador de datos puede publicar esquemas usando los formatos Frictionless Data o el esquema propuesto por el UK National Archives para especificar tablas de datos CSV junto con los propios datos en CSV.

**Ejemplo:** A continuación, se muestra un esquema Frictionless Data para datos ficticios de empleados. En él se observan diferentes restricciones: los nombres de los departamentos solo pueden contener letras mayúsculas y los números del 1 al 4, y el estado de los empleados es "jubilado" o no.

```
{
        "fields": [
                {
                "name": "name",
                "type": "string",
                "description": "Employee's name"
                },
                {
                "name": "department",
                "type": "string",
                "description": "Department ID"
                        "constraints": {
                                        "pattern": "[A-Z]{1,4}"
                                }
                },
                {
                "name": "retired",
                "type": "boolean",
                "description": "Employee status"
                }
]
```

**Ejemplo:** Se muestra el esquema CSV usando la especificación del UK National Archives para describir una tabla de datos de empleados. Contiene las mismas restricciones que en el ejemplo anterior.

```
version 1.1
@separator ";"
@totalColumns 3
name: unique notEmpty
department: regex("[A-Z]{1,4}")
retired: is("yes") or is("no")
```

## 6.6. Cómo especificar estructuras de datos RDF

La forma natural de definir un modelo de datos en forma de grafo RDF es mediante una ontología (representación explicita del conocimiento sobre un dominio concreto a través de sus entidades y relaciones). Un grafo RDF es en sí mismo un modelo de datos por lo que se puede especificar mediante

esquemas. La especificación de W3C SHACL (Shapes Constraint Language) es un potente marco conceptual que permite la validación contra estos esquemas. SHACL también se expresa utilizando RDF. Especifica una sintaxis en RDF que se puede utilizar para definir las condiciones con las que debe ser coherente un RDF de entrada. Los publicadores de datos pueden publicar archivos SHACL (denominados archivos 'shapes' o de condiciones que debe cumplir el modelo) que especifiquen la estructura RDF además de los datos reales en RDF.

**Ejemplo:**

| | |
|---|---|
| @prefix schema: <http://schema.org/>.<br>@prefix sh: <http://www.w3.org/ns/shacl#>.<br>@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.<br><br>schema:PersonShape<br>a sh:NodeShape ;<br>sh:targetClass schema:Person ;<br>sh:property [<br>sh:path schema:givenName ;<br>sh:datatype xsd:string ;<br>sh:name "given name" ;<br>] ;<br>sh:property [<br>sh:path schema:birthDate ;<br>sh:lessThan schema:deathDate ;<br>sh:maxCount 1 ;<br>]. | Se muestra un archivo de 'shapes' en SHACL que especifica determinadas condiciones sobre datos personales. A modo de ejemplo, observad la restricción de que la fecha de nacimiento debe ser anterior a la fecha de defunción. |
| [<br>a sh:ValidationResult ;<br>sh:resultSeverity sh:Violation ;<br>sh:sourceConstraintComponent sh:LessThanConstraintComponent ;<br>sh:sourceShape _:n703 ;<br>sh:focusNode <http://example.org/ns#Bob> ;<br>sh:resultPath schema:birthDate ;<br>sh:value "1971-07-07" ;<br>sh:resultMessage "Value is not < value of schema:deathDate" ;<br>]. | Si los datos de ejemplo se validan contra el archivo SHACL anterior, se genera el informe de la izquierda. Se puede observar que el desajuste entre la fecha de nacimiento y la fecha de defunción se detecta como una violación. |

Ejemplos de herramientas de validación reseñables son: SHACL playground, disponible online y TopBraid Composer (software comercial).

## 6.7. Cómo especificar datos servidos vía API

Las API permiten el intercambio de información entre aplicaciones. El modelo de arquitectura API-REST es el mecanismo que permite servir datos en la web mediante la invocación de URLs que permiten determinadas operaciones, como por ejemplo, el filtrado o la selección de ciertos datos desde un origen

específico. Para que los usuarios puedan usar fácilmente una API, debe estar completa y detalladamente documentada. Los publicadores de datos deben documentar no solo la estructura de los datos servidos, sino también cómo se puede acceder y descargar estos datos de la web. Dependiendo del protocolo de la API, pueden aplicarse diferentes métodos de documentación. La recomendación para documentar API-REST es utilizar el estándar OpenAPI. Las especificaciones OpenAPI se pueden escribir en JSON o YAML. Contendrá, la estructura y semántica de los datos servidos, además de especificar los siguientes aspectos de la API:

- URL y endpoints;
- los protocolos de los puntos de acceso (por ejemplo: HTTP, FTP);
- métodos de acceso (por ejemplo, métodos HTTP, códigos de estado);
- formas de alterar los resultados (por ejemplo, parámetros de consulta, encabezados HTTP).

Ejemplo: A continuación, se muestra un fragmento de una especificación OpenAPI que describe la llamada a la API del portal data.europa.eu que permite recuperar un conjunto de datos. Además de garantizar una estructura sólida con todos los campos obligatorios establecidos, la especificación debe ser completa y exhaustiva con respecto a los aspectos mencionados anteriormente. Las descripciones ayudan a comprender el significado semántico de los datos servidos.

```
paths:
    '/data set/{data setId}.rdf':
        get:
            summary: Retrieve data set in RDF/XML
            description: >
                Return the full content of the data set in RDF
            operationId: getData set
            parameters:
                - name: data setId
                  in: path
                  description: data set identifier
                  required: true
                  style: simple
                  explode: false
                  schema:
                    type: string
            responses:
                '200':
                  description: OK
                  content:
                    application/rdf+xml:
                      schema:
                        type: string
                '404':
                  description: not found
                  content:
                    text/html:
                      schema:
                        type: string
```

Una herramienta útil que ayuda a editar y validar las especificaciones que siguen el estándar OpenAPI es Swagger (software comercial/código abierto).

## 6.8. Documentar la semántica de los datos

Dependiendo de su complejidad, la publicación de un esquema no siempre es suficiente. Si bien un esquema describe la sintaxis y la estructura, no siempre explica adecuadamente la semántica de los datos. Una descripción de las propiedades individuales de una estructura de datos ayuda a los usuarios a interpretar y reutilizar los datos correctamente y de la manera prevista por el publicador de datos.

**Ejemplo:** A continuación, se muestra un diccionario de datos que de manera sencilla en formato de texto vincula tanto el esquema como la descripción semántica de sus datos. Este archivo de texto se puede disponer fácilmente como una distribución más del conjunto de datos.

Archivo de datos: http://example.org/automoviles.csv
Descripción: Tabla con datos de automóviles clásicos
Publicador: Autor del ejemplo
Columna 1:
Titulo: marca
Descripción: Este campo contiene información sobre la marca y modelo de cada vehículo.
Tipo de datos: string
Columna 2:
Titulo: año
Descripción: Este campo contiene información sobre el año de fabricación de cada vehículo.
Tipo de datos: date
Columna 3:
Titulo: cilindros
Descripción: Este campo contiene información sobre el número de cilindros de cada vehículo.
Tipo de datos: integer
Columna 4:
Título: consumo
Descripción: Este campo contiene información sobre el consumo medio de cada vehículo, medido en litros / 100 kms.
Tipo de datos: decimal
Columna 5:
Título: potencia
Descripción: Este campo contiene información sobre la potencia de cada vehículo, medida en CV.
Tipo de datos: decimal
Columna 6:
Título: aceleración
Descripción: Este campo contiene datos sobre la aceleración de cada vehículo, medida en m/seg2.
Tipo de datos: decimal

Algunas sencillas herramientas pueden ayudar a crear documentación. Entre otras, Sphinx es un generador escrito en Python de documentos en formato HTML, PDF y texto plano, entre otros. (software de código abierto). ReadTheDocs, además de la generación de documentos, ofrecen un servicio de alojamiento para que la documentación generada quede disponible online (software de código abierto)

# 7. Conclusiones

La proliferación de datos de diversa índole en estos últimos años ha puesto el foco en la disponibilidad de los mismos en detrimento de su calidad, a pesar de ser una preocupación general de publicadores y reutilizadores de datos. Los publicadores de datos publican miles de conjuntos de datos con deficiencias de calidad que solo pueden ser detectados después del comienzo del proceso de reutilización, generando una necesidad mayor de recursos para su 'curación', en muchos casos inasumible por el usuario, lo que provoca una pérdida de interés por la reutilización de datos.

En esta guía se ha mostrado una recopilación de los errores más comunes que presentan los conjuntos de datos y a los cuales se enfrentan a diario los reutilizadores. Errores que, en la medida de lo posible, deben minimizarse aplicando las recomendaciones referidas, que están basadas en diferentes estándares cuyo objetivo es aumentar la calidad de los datos.

La guía hace un repaso por las pautas generales y específicas de los formatos de datos abiertos más habituales, que detallan los problemas que es necesario afrontar, las características de calidad afectadas y las recomendaciones para su resolución con ejemplos prácticos que ayudarán a entender cada caso presentado. En la última parte se han tratado dos aspectos importantes que contribuyen a mejorar la calidad general de los datos y que son la esencia del paradigma de la web de datos o Linked Data, como son la estandarización y el enriquecimiento. El último aspecto tratado y no menos importante, es el de la documentación de datos abiertos que -aunque no ha sido objeto de esta guía-debe complementarse con una adecuada gestión de metadatos.

Esta guía es una pequeña pincelada de los cambios que se deben implementar en la disponibilidad de conjuntos de datos para mejorar su calidad y por tanto aumentar su reutilización y con ello su valor. No obstante, el publicador de datos debe valorar el feedback aportado por la comunidad de reutilizadores, dado que su experiencia ayudará a detectar y solucionar deficiencias contenidas en los datos que a priori no es sencillo identificar, por lo que su contribución también permitirá mejorar la calidad general de datos.

Esperamos que os resulte útil esta nueva guía. Seguiremos generando contenidos de interés relacionados con el mundo de los datos abiertos. ¡Hasta pronto!

# 8. Herramientas

A continuación, se incluye una relación de herramientas útiles para trabajar la calidad de los datos:

| | | |
|---|---|---|
| UTF-8 Tools | Colección de utilidades basadas en el navegador para trabajar con la codificación UTF8 | https://onlineutf8tools.com/ |
| CSVLint | Información sobre si los archivos CSV son legibles | https://csvlint.io/ |
| DenCode | Herramienta de ayuda a los publicadores en la conversión de datos tipo fecha al formato ISO 8601 | https://dencode.com/date/iso8601 |
| XML Escape/Unescape | Herramienta online de código abierto utilizada para el escape de los caracteres especiales en XML y realización del proceso inverso | https://www.freeformatter.com/xml-escape.html |
| Title Case | Herramienta para la conversión de frases compuestas por múltiples palabras en varios formatos de mayúsculas y minúsculas | https://titlecase.com/ |
| Jsonlint | Herramienta online para chequear si el código JSON es válido | https://jsonlint.com/ |
| OpenRefine | Herramienta para el tratamiento y enriquecimiento de datos. | https://openrefine.org/ |
| OntoRefine | Herramienta que permite transformaciones para convertir datos tabulares en RDF | https://graphdb.ontotext.com/documentation/9.8/free/loading-data-using-ontorefine.html |
| JSON Schema | Herramienta para generar y validar esquemas JSON | https://json-schema.org/ |
| JSON schema generator | Herramienta para generar y validar esquemas JSON | https://jsonschema.net/ |
| Liquid Studio | Software comercial que permite la generación de archivos XSD a partir de XML existente | https://www.liquid-technologies.com/xml-schema-editor |
| XMLFox | Software comercial y código abierto que cuenta con validación XSD | https://www.xmlfox.com/ |
| SHACL playground | Herramienta online de validación para el lenguaje SHACL | https://shacl.org/playground/ |
| TopBraid Composer | Software comercial de validación para el lenguaje SHACL | https://www.topquadrant.com/products/topbraid-composer/ |

| | | |
|---|---|---|
| Swagger | Herramienta para la edición y validación de especificaciones que siguen el estándar OpenAPI | https://swagger.io/ |
| Sphinx | Software de código abierto para documentar la semántica de los datos | https://www.sphinx-doc.org/en/master/index.html |
| ReadTheDocs | Software de código abierto para alojar y documentar la semántica de los datos | https://readthedocs.org/ |

# 9. Referencias y recursos bibliográficos

1.Oficina del Dato. https://oficinadato.gob.es

2.Data.europa.eu data quality guidelines. Oficina de Publicaciones de la UE. https://op.europa.eu/en/publication-detail/-/publication/023ce8e4-50c8-11ec-91ac-01aa75ed71a1/language-en

3.5 estrellas de datos abiertos. 5 estrellas de datos abiertos. https://5stardata.info/es/

4.The 8 Principles of Open Government Data. Principios de los datos del gobierno abierto. Open Goverment Data. https://opengovdata.org/.

5.Carta Internacional de Datos Abiertos. https://opendatacharter.net/principles-es/

6.Quality Framework and Guidelines for OECD Statistics Activities. Organización de Cooperación y Desarrollo Económico (OCDE). https://www.oecd.org/officialdocuments/publicdisplaydocumentpdf/?cote=std/qfs%282011%291&doclanguage=en.

7.ISO/IEC 25012. Portal ISO 25000. https://iso25000.com/index.php.

8.FAIR Principles. https://www.go-fair.org/fair-principles/

9.Código ASCII. https://elcodigoascii.com.ar/

10.EU Vocabularies. Oficina de Publicaciones de la UE. https://op.europa.eu/en/web/eu-vocabularies/

11.Ciudades Abiertas. https://ciudades-abiertas.es/

12.Linked Open Vocabularies (LOV). https://lov.linkeddata.es/dataset/lov

13.Nomenclatura estadística de actividades económicas de la Comunidad Europea, Rev. 2. Eurostat. https://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=LST_NOM_DTL&StrNom=NACE_REV2&StrLanguageCode=ES&IntPcKey=&StrLayoutCode=HIERARCHIC&IntCurrentPage=1

14.ISO 8601. Date and time format. https://www.iso.org/iso-8601-date-and-time-format.html

15.¿Qué es un diccionario de datos y por qué es importante?. datos.gob.es. https://datos.gob.es/es/blog/que-es-un-diccionario-de-datos-y-por-que-es-importante

16.Guía práctica para la publicación de Datos Espaciales. datos.gob.es. https://datos.gob.es/es/documentacion/guia-practica-para-la-publicacion-de-datos-espaciales.

17.Open Geospatial Consortium. https://www.ogc.org/

18.DCAT. The World Wide Web Consortium (W3C). https://www.w3.org/TR/vocab-dcat/

19.Guía de aplicación de la Norma Técnica de Interoperabilidad de política de firma y sello electrónicos y de certificados de la administración (2ª ed) (2017). Portal de la Administración Electrónica (PAe).

https://administracionelectronica.gob.es/pae_Home/pae_Biblioteca/pae_PublicacionesPropias/Monografias-administracion-electronica/Guias-de-aplicacion-NTI.html.

20. Guía práctica para la publicación de datos tabulares en archivos CSV. datos.gob.es. https://datos.gob.es/es/documentacion/guia-practica-para-la-publicacion-de-datos-tabulares-en-archivos-csv.

21.Extensible Markup Language (XML) 1.1 (Second Edition). The World Wide Web Consortium (W3C) (2006). https://www.w3.org/TR/2006/REC-xml11-20060816/

22.Guía práctica para la publicación de datos enlazados en RDF. datos.gob.es. https://datos.gob.es/es/documentacion/guia-practica-para-la-publicacion-de-datos-enlazados-en-rdf

23.DCAT-AP. Comisión Europea. https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe/releases

24. Linked Open Vocabularies (LOV). https://lov.linkeddata.es/dataset/lov

25.Guía práctica para la publicación de Datos Abiertos usando APIs. datos.gob.es. https://datos.gob.es/es/documentacion/guia-practica-para-la-publicacion-de-datos-abiertos-usando-apis.

26.Wikidata. https://www.wikidata.org/wiki/Wikidata:Main_Page

27.SmartDataModels (SDM). https://smartdatamodels.org/

28.Design & Manage Persistent URIs. Comisión Europea (2014). https://opendatahubs.eu/sites/default/files/presentations/d2.1.2_training_module_2.3_persistent_uri_design_and_management_en_edp.pdf

29.Cleaning Data with OpenRefine. https://doaj.org/article/3ccd075407a4481c85c0d00d65a003c0

30.W3C XML Schema Definition Language (XSD). The World Wide Web Consortium (W3C) (2012). https://www.w3.org/TR/xmlschema11-1/

31.Shapes Constraint Language (SHACL). The World Wide Web Consortium (W3C) (2017). https://www.w3.org/TR/shacl/

32.OpenApi Initiative. https://www.openapis.org/

## ¿QUIERES SABER MÁS SOBRE LA INICIATIVA APORTA?

Visita www.datos.gob.es

Instagram: @datosgob

Twitter: @datosgob

LinkedIn: datos.gob.es

Suscríbete a nuestro boletín

Escribe a contacto@datos.gob.es

Puedes identificar los espacios de **datos abiertos** gracias a este logo

datos abiertos