Datos en Abierto Enlazados

Jordi Casas Roma, Jordi Conesa i Caralt

1 créditos
PX/XXXXX
Módulo 1

Índice

Introducción

En este módulo didáctico trataremos los conceptos básicos de los datos abiertos (en inglés, *Open Data*) y enlazados (en inglés *Linked Data*). Estas propuestas permiten proveer una gran cantidad de datos con licencia libre y enlazados nivel global. Como tal, se convierte en una potencial fuente de datos a considerar en nuestros análisis.

Las tecnologías y conceptos relacionados con *linked data* son complejos y amplios. En este módulo los introduciremos brevemente para dar una visión general de qué es *linked data*, como se representan datos en *linked Data*, donde encontrar los datos enlazados y cómo consultarlos.

Iniciaremos el módulo reflexionando sobre las definiciones implícitas en los conceptos de "datos" y de "abiertos". A partir de estas reflexiones, en el siguiente capítulo intentaremos definir lo que se entiende por *Open Data*, aunque como veremos la definición no es única. Discutiremos también los beneficios que aportan los datos abiertos y veremos los elementos más importantes que deben ser considerados en el proceso de publicación de datos.

Veremos que con abrir los datos no es suficiente ya que, si no vamos más allá, tendremos datos en abierto, pero totalmente desconectados con datos de otros dominios de aplicación. Recordemos que la potencia de los análisis que podamos hacer no depende tanto de datos aislados sino de las interrelaciones que podamos establecer entre datos de distintos ámbitos. En el tercer capítulo veremos como pretende abordar esta conexión de datos la aproximación de datos enlazados (o *Linked Data* en inglés). En él introduciremos los datos enlazados, hablaremos sobre la definición de los cinco niveles de datos enlazados según Tim Berners-Lee, de los beneficios que aportan, presentaremos ejemplos de conjuntos de datos enlazados, y veremos un ejemplo sobre como, paso a paso, podemos enlazar un conjunto de datos en los distintos niveles del modelo de cinco niveles.

En el siguiente capítulo, presentaremos brevemente las tecnologías asociadas con los datos enlazados con el objetivo de que el lector conozca sus principios básicos y sea capaz de interpretar datos enlazados. En particular, estudiaremos cómo representar datos enlazados mediante RDF, qué son los vocabularios y porqué deben usarse, como representar datos RDF, introduciremos el concepto de *Web de Datos* y veremos qué papel juegan los datos enlazados en la web de datos.

Finalizaremos este módulo didáctico mostrando cómo utilizar SPARQL para consultar datos enlazados en RDF.

Objetivos

En los materiales didácticos de este módulo encontraremos las herramientas indispensables para asimilar los siguientes objetivos:

1. Entender el significado y el potencial de *Open Data*.

2. Conocer los principales beneficios que puede aportar la publicación de datos abiertos.

3. Comprender los distintos aspectos relevantes que deben ser considerados en el proceso de publicación de datos abiertos.

4. Entender el significado de *Linked Data*, las diferencias respecto al *Open Data* y los beneficios que puede aportar.

5. Conocer los principales beneficios que puede aportar la publicación de datos enlazados.

6. Conocer los principales repositorios de datos enlazados y ser capaz de buscar datos enlazados disponibles para un dominio de aplicación dado.

7. Conocer las tecnologías asociadas a los datos enlazados, básicamente RDF, SPARQL.

8. Ser capaz de interpretar datos enlazados.

9. Ser capaz de consultar datos enlazados utilizando SPARQL.

1. Conceptos preliminares

.

Iniciaremos este módulo didáctico reflexionando sobre qué significan los términos que definen los datos abiertos u *Open Data*. En primer lugar reflexionaremos sobre el concepto de "datos", y a continuación veremos qué es los que entendemos por "abierto". Parece un tema obvio, pero la verdad es que también es un concepto muy rico, que admite algunas sutiles deferencias y que es conveniente definir correctamente.

1.1. El concepto de "datos"

En primer lugar, vamos a hablar un poco de lo que significa el concepto de "dato". Supongamos que alguien nos transmite el siguiente dato:

42

Inmediatamente, nos aparece la pregunta "¿42 qué?".

Con este sencillo ejemplo sólo pretendemos mostrar, de momento, dos cosas:

1) Un dato, sin su contexto, carece de significado.

2) El formato (en un sentido amplio) de representación del dato es importante.

Supongamos que ahora nos dicen "la temperatura del paciente es de 42 grados". Hemos dotado de significado en el 42, cuando el **dato** (42) es la respuesta a una pregunta ("¿cuál es la temperatura del paciente?"). Hemos avanzado un nivel y ya podemos hablar de **información**. Pero aún no somos lo suficientemente precisos. ¿Qué quiere decir que "la temperatura del paciente es de 42 grados"? Pues cosas bien distintas:

- Si son grados Celsius, el paciente tiene fiebre (42ºC).
- Si son grados Fahrenheit, el paciente es un cadáver frío (5ºC).
- Si son grados Kelvin, el paciente es un cadáver congelado a -231ºC.

Así pues, para poder hablar de información con propiedad, necesitamos un tercer punto:

3) Los datos tienen unidades y un rango asociado.

Por tanto, este contexto del que acabamos de hablar (formato, unidades, rangos) es lo que da significado a un dato y lo convierte en información que responde a una pregunta. Cuando tenemos mucha información (y, por tanto, muchos datos), podemos inferir nueva información combinando diferentes fuentes, creando **conocimiento**. Así, por ejemplo, sabemos que "si la temperatura del paciente llega a 42 grados Celsius, la fiebre puede causar daños cerebrales irreversibles". Es este conocimiento el que perseguimos, construyéndolo combinando muchas fuentes de información, en este caso las medidas de temperatura de los pacientes con las observaciones de los médicos, experimentos en el laboratorio, el análisis de las muestras de tejido cerebral, etc. Esta estructura se conoce como la Jerarquía DIKW (*Data-Information-Knowledge-Wisdom*).

En nuestro contexto, pero, por "datos" entendemos normalmente tablas de dos dimensiones que representan, para cada fila, un elemento, y para cada columna un valor asociado a un atributo del elemento, de forma que cada elemento queda descrito mediante sus atributos.

> Por ejemplo, mediante el formulario de inscripción a un curso *online* podemos recoger información sobre los participantes en el mismo, de forma que por cada participante sabemos cuando se registró, su edad, su sexo, el ámbito y sector profesional que desarrolla su actividad y si había participado o no en la edición anterior de este curso.

Pero los datos (abiertos o no) no son sólo tablas. Hay muchos otros tipos de datos que tenemos disponibles. Por ejemplo, una entrada en un blog es un tipo de dato de "información textual", con sus particularidades: falta de estructura clara, formato y contenido entrelazados, etc.

1.2. El concepto de "abierto"

A continuación vamos a reflexionar, brevemente, sobre qué significa "abierto". Seguramente todos tenemos una idea intuitiva de lo que significa abierto. Seguramente, la mayoría de nosotros por abierto entendemos "accesible, utilizable, libre". Esta acepción nos da pie a relacionarse "abierto" con "libertad", como veremos a continuación.

En el artículo "El futuro abierto", de David Wiley, se define qué significa abierto pero en un contexto de recursos educativos. Aunque es fácil extrapolar entre contenidos y datos. De forma resumida, el autor define un contenido como abierto cuando tenemos la posibilidad de realizar diferentes acciones (las 4 Rs) sobre el mismo:

Lecturas complementarias

Wiley, David. "The Open Future: Openness as Catalyst for an Educational Reformation". EDUCAUSE Review, vol. 45 (4), pp. 14–16, Jul-Aug 2010.

- **Reutilizar**: el derecho a reutilizar su contenido en forma inalterada o literal (por ejemplo, usándolo tal cual).

- **Revisar**: el derecho a adaptar, ajustar, modificar o alterar el contenido (por ejemplo, traduciendo a otro idioma).

- **Remezclar**: el derecho a combinar el contenido original o revisado con otro contenido para crear un producto nuevo (por ejemplo, integrar el contenido en una remezcla).

- **Redistribuir**: el derecho a compartir copias del contenido original, de las revisiones o de las remezclas con otros (por ejemplo, dar una copia del contenido a un conocido).

El propio autor, recientemente[1], ha ampliado esta definición de abierto con una 5ª R:

- **Retener**: el derecho a hacer copias y poseer una copia del contenido original.

Resumiendo, un contenido (en este caso unos datos) será más o menos abierto en función de más o menos Rs que podemos ejecutar sobre el mismo.

1.3. Reflexión sobre los datos abiertos

En esta sección reflexionaremos sobre la definición que acabamos de ver, basada en los recursos abiertos, aplicándola a un ejemplo concreto. Veremos que en las ambigüedades de esta definición es donde pueden aparecen los posibles problemas.

Supongamos el siguiente contexto:

Cuatro amigos que estudian Periodismo y/o Comunicación Audiovisual quieren hacer una recopilación de noticias que ellos consideran importantes sobre su ciudad, según sus intereses personales. Así, cada vez que uno de ellos encuentra una noticia interesante, crea una pequeña ficha en la que se indica, de momento, el medio donde se ha publicado la noticia (puede ser un periódico, un canal de radio o televisión, una página web, etc.), la fecha y hora de la noticia, de qué temas trata la noticia (usando palabras clave que ellos mismos deciden, por ejemplo "política", "economía" o "deportes") y, finalmente, el enlace a la noticia. Todavía no saben cómo gestionar estas pequeñas fichas, así que de momento han creado una carpeta compartida entre los cuatro y han decidido ponerlas allí. Esta carpeta, donde sólo ellos pueden escribir, la hacen pública, es decir, otras personas pueden ver el resumen de noticias que ellos crean. Los cuatro amigos creen que otros se pueden aprovechar de su esfuerzo, y además, algún profesor ya los ha animado a hacerlo.

Entonces,

[1] Artículo disponible en `http://opencontent.org/blog/archives/3221`

- ¿Cuáles de las 5 Rs realizan los cuatro amigos con las noticias?

  La pregunta misma ya presenta una ambigüedad respecto al escenario planteado: ¿estamos hablando de noticias como contenido o del enlace a la noticia? En función de lo que uno entienda por contenido, los amigos hacen unas cosas u otras. Supondremos que el contenido que utilizan es la noticia y su enlace.

  Entonces, nuestros amigos **reutilizan** la noticia, porque básicamente acceden para decidir si es lo suficientemente interesante o no. Estamos de acuerdo que no enlazarán una noticia que no han visto, ¿verdad? Del mismo modo, si la noticia no hubiera sido accesible mediante un enlace no podrían haberla seleccionado por su recopilación. De hecho, si un contenido es abierto, el simple hecho de acceder y hacer cualquier cosa con él ya implica ejecutar la primera R.

  Nuestros amigos **revisan** el contenido de la noticia porque extraen (computan ellos mismos) unas palabras clave, aparte de usar su fecha y hora, el enlace, etc. Imaginaos que en lugar de palabras clave hubiéramos hablado de un resumen, quizás entonces sería más claro que están "transformando" la noticia en otra cosa, pero la idea detrás de revisar es la misma. De hecho, esta R incluye tantas posibilidades (traducir, adaptar el formato, calcular, etc) que es también una de las operaciones habituales.

  Por otro lado, los cuatro amigos **no remezclan** dos noticias para hacer una nueva, las procesan una a una. Ahora, si imaginamos que publican pequeñas recopilaciones en paquetes temáticos (por ejemplo, noticias de política, deporte, etc), sí que podríamos entender que remezclan el contenido original (cada una de las noticias) creando uno nuevo (el paquete). Esta R junto con el anterior van de la mano muchas veces, porque son dos operaciones complementarias. Es decir, si los cuatro amigos utilizaran imágenes para ilustrar las noticias, podrían tomar una imagen de un lugar y usarla para una noticia (como hacen los periodistas con las imágenes de archivo).

  Llegamos a la R más complicada; tal y como hemos comentado, no es lo mismo hacer un "cortar y pegar" de la noticia que sólo compartir su enlace. Nuestros amigos redistribuyen enlaces (y algo más), pero realmente no redistribuyen las noticias.

  Finalmente, con la quinta R pasa algo parecido a la anterior. Los cuatro amigos retienen el enlace, pero no la noticia. Recordemos que Google (que se supone que gestiona contenidos accesibles mediante enlaces) ofrecía poder ver una página aunque ésta ya no existiera mediante el uso de una copia. Y eso les causó problemas porque los "enlazados" se quejaban de que Google duplicaba contenidos sin permiso.

- ¿Cuál es el rol o roles que adoptan los cuatro estudiantes con respecto a las noticias: son productores, consumidores y/o mediadores?

Nuestros amigos **no son productores** de noticias, producen recopilaciones de noticias. En este sentido, son **mediadores**, ya que facilitan el acceso a ciertos datos. ¿Se puede ser mediador sin consumir? En este caso podemos pensar que si los amigos hacen unas recopilaciones de noticias que ellos aprovechan para otras tareas, entonces también **son consumidores**, pero si sólo lo hacen para terceros, entonces no lo son.

- En lugar de compartir los datos con todo el mundo libremente a través de la carpeta compartida, ¿"podrían" (en un sentido amplio) cobrar por facilitar la información que ellos recogen a terceros?

  Podrían hacerlo, pero es muy improbable que si no añaden valor alguien decida pagar por el servicio ofrecido. De hecho este es el cambio de paradigma que hay detrás de los modelos de negocio: no es fácil vender datos (abiertos o no) como producto, es necesario convertirlo en un servicio que sea interesante para los usuarios finales.

- ¿Podrían los propietarios de los canales tradicionales de comunicación que ellos usan como fuente quejarse de la actividad de los estudiantes por ilícita o competencia desleal?

  La respuesta a esta pregunta es más complicada, porque podemos encontrar las dos posturas: o bien pensar que los cuatro amigos se aprovechan del trabajo de los demás y sacan un beneficio; o bien, pensar en esta iniciativa como un nuevo recurso que aporta visibilidad a contenido ya existente. Otra vez Google ha tenido problemas con proveedores de contenidos que lo acusan de beneficiarse sin hacer nada, cuando por otra parte Google genera un considerable tráfico en muchas páginas web que, sin Google, serían invisibles. Encontrar el equilibrio pasa una vez más por diferenciarse aportando un valor añadido.

2. *Open Data*

.

¿Sabemos exactamente qué parte de nuestro dinero de los impuestos se gasta en iluminación de las calles o en la investigación del cáncer? ¿Sabemos cuál es la ruta en bicicleta más corta, más segura y más pintoresca desde nuestra casa al trabajo? ¿Y la calidad del aire que respiramos de camino al trabajo? ¿Y en qué región de la ciudad se encuentran las mejores oportunidades de trabajo o el mayor número de árboles por cápita?

Lectura complementaria

Open Knowledge Foundation, *Open Data Handbook*. Disponible en `http://opendatahandbook.org/`

Las nuevas tecnologías de la información hacen posible el desarrollo de servicios que permiten responder a estas preguntas de forma automática. Gran parte de los datos necesarios para responder a estas preguntas son generados por los organismos públicos. Sin embargo, a menudo estos datos no están disponibles de forma que sean sencillos de utilizar para responder este tipo de preguntas. En este capítulo veremos cómo podemos liberar el potencial de datos relacionados gracias a los datos abiertos.

2.1. ¿Qué es *Open Data*?

La definición de *Open Data* o datos abiertos no es única en la actualidad. Existen diferentes aproximaciones con ligeros matices, aunque en esencia no presentan diferencias importantes entre las principales definiciones. Por ejemplo, en el capítulo anterior hemos visto la definición que proviene de los recursos educativos en abierto. La organización *Open Knowledge Foundation*[1] define los datos abiertos de forma muy similar, pero no igual, a como la acabamos de definir en el capítulo anterior:

En este módulo utilizaremos los términos Open Data y datos abiertos indistintamente.

> "Los datos abiertos son datos que pueden ser utilizados, reutilizados y redistribuidos libremente por cualquier persona, y que se encuentran sujetos, cuando más, al requisito de atribución y de compartirse de la misma manera en que aparecen."
>
> **Open Knowledge Foundation**, *Open Data Handbook*. `http://opendatahandbook.org/`

La definición de apertura completa da detalles precisos de lo que significa. Según la *Open Knowledge Foundation* los detalles más importantes en relación al concepto de datos abiertos se resumen en los siguientes tres puntos:

- **Disponibilidad y acceso**: la información debe estar disponible como un todo y a un costo razonable de reproducción, preferiblemente descargándola de Internet. Además, la información debe estar disponible en una forma conveniente y modificable.

[1] `https://okfn.org/`

- **Reutilización y redistribución**: los datos deben ser provistos bajo términos que permitan reutilizarlos y redistribuirlos, e incluso integrarlos con otros conjuntos de datos.

- **Participación universal**: todos deben poder utilizar, reutilizar y redistribuir la información. No debe haber discriminación alguna en términos de esfuerzo, personas o grupos. Restricciones "no comerciales" que prevendrían el uso comercial de los datos o restricciones de uso para ciertos propósitos (por ejemplo sólo para educación) no son permitidos.

Ambas definiciones nos llevan al concepto de "interoperabilidad" de los datos. La interoperabilidad denota la habilidad de diversos sistemas y organizaciones para trabajar juntos. En este caso, es la habilidad para interoperar o integrar diferentes fuentes de datos. La interoperabilidad es importante porque permite que distintos componentes trabajen juntos. Esta habilidad de integrar componentes es esencial para construir sistemas complejos y grandes.

Interoperabilidad

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) define interoperabilidad como la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

La esencia de los datos compartidos es que una parte del material abierto pueda a partir de ahí ser mezclado con otro material abierto. Esta interoperabilidad es absolutamente fundamental para entender los principales beneficios prácticos de la apertura: el incremento dramático de la habilidad de combinar distintas fuentes de datos o conjuntos de datos y así desarrollar más y mejores productos y servicios.

Proveer una definición clara de apertura garantiza que cuando se trabaje con conjuntos de datos abiertos de fuentes diferentes, se los pueda combinar, asegurando la posibilidad de combinarlos en sistemas más grandes, donde se encuentra el verdadero valor.

2.2. Beneficios de los datos abiertos

En la sociedad actual existen multitud de individuos, organizaciones y, especialmente, administraciones públicas que colectan y gestionan una gran cantidad y variedad de datos para llevar a cabo sus tareas cotidianas. En este escenario, las administraciones públicas, como por ejemplo ayuntamientos o gobiernos regionales, son especialmente importantes por la cantidad de datos que manejan, pero también porque una parte importante de esta información es abierta y se pone a disposición de cualquier individuo o institución que desee utilizarla.

Existen muchas áreas donde podemos ver que los datos abiertos han creado valor añadido a la sociedad. A modo de ejemplo, podemos enumerar algunas de las más relevantes:

- Transparencia y control democrático

- Participación

- Creación de nuevos productos y servicios
- Innovación
- Mejoras en la eficiencia y eficacia de los servicios ofrecidos por el ayuntamiento
- Medición del impacto de políticas

Category	Open Data Readiness score	Number of countries 2015	Number of countries 2016
Not ready	0-20%	4	2
Partly ready	21-40%	7	4
Almost ready	41-60%	13	13
Ready	61-80%	5	10
Very ready	81-100%	2	2

Figura 21. Madurez de los datos abiertos en los distintos gobiernos europeos.

Los datos en abierto no son una moda pasajera que vaya a quedar descuidada a corto/medio plazo sino una estrategia que nos acompañará durante los próximos años. Prueba de ello es el hecho de que los gobiernos, ya sean locales, autonómicos, nacionales o supranacionales, están apostando fuerte por la abertura de datos. Ejemplo de ello son las medidas tomadas por la Unión Europea para adoptar una política de datos abiertos a nivel europeo y ofrecerlos desde un portal integrado, llamado **European Data Portal**. Gracias a estas medidas todos los países de la unión están impulsando portales de datos abiertos, poblándolos periódicamente con datos y generando políticas que potencien la abertura de datos a nivel gubernamental.

European Data Portal

Portal de la Union Europea sobre datos en abierto. Accesible a traves de `https://www.europeandataportal.eu/`

Actualmente, a principios del 2017, la mayoría de los países de la unión están listos o cuasi listos para ofrecer datos en abierto, periódicamente y formateados para un consumo automático, como se puede ver en la figura 21. Como curiosidad, comentar que a principios del 2017, España (ES) es uno de los países más avanzado en cuanto a disponibilidad de datos en abierto y funcionalidad de su portal de datos en abierto, como puede verse en la figura 2. Podemos encontrar más información sobre el estado de *open data* en Europa en el informe anual de abertura de datos abiertos realizado por la unión europea.

Informe anual sobre la madurez de los datos en abierto en Europa

El de 2016 se puede obtener desde `https://www.europeandataportal.eu/sites/default/files/edp_landscaping_insight_report_n2_2016.pdf`.

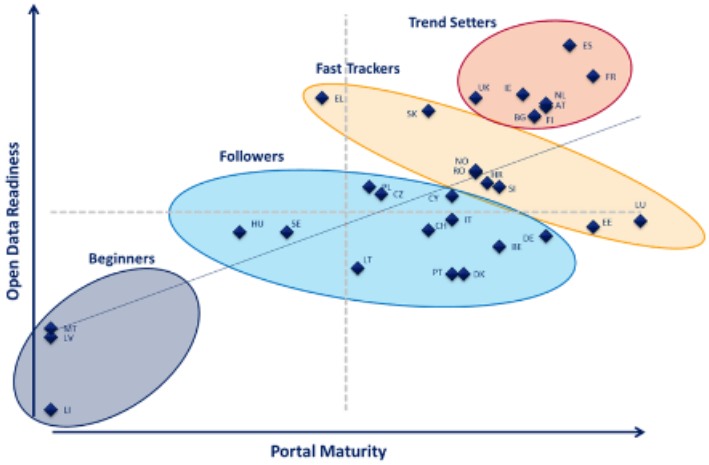


Figura 2. Madurez de los distintos gobiernos europeos en datos en abierto.

Es muy difícil predecir con exactitud cómo, cuando y dónde se creará el valor a partir de los datos en el futuro. En muchas ocasiones, este valor añadido a la sociedad proviene de los lugares más inverosímiles.

En términos de transparencia, proyectos como el *Open Spending*[2] muestran cómo gobierno locales y regionales de múltiples países del mundo está gastando el dinero de los impuestos de los ciudadanos. En términos económicos, los datos abiertos son también de gran importancia. Numerosos estudios estimaron el valor económico de los datos abiertos en varias decenas de billones de Euros al año, sólo en la Unión Europea. Nuevos productos y compañías están reutilizando datos públicos. Por ejemplo, el Traductor de Google[3] usa el enorme volumen de documentos de la Unión Europea que aparecen en todos los idiomas europeos para entrenar sus algoritmos de traducción y así mejorar la calidad de su servicio. Los datos abiertos también pueden contribuir a facilitar las acciones más elementales y cotidianas en la vida de los ciudadanos. Un ejemplo interesante se puede encontrar en la página web FindToilet[4], que muestra todos los baños públicos daneses que puedes encontrar a un cierto radio de ubicación actual usando datos abiertos publicados por el propio gobierno. Este portal fue creado especialmente para personas con problemas de vejiga, aunque se puede beneficiar cualquier ciudadano o visitante.

Ejemplo de transparencia en el Ayuntamiento de Barcelona

El proyecto *Open Spending* recolecta información sobre cómo los gobiernos locales y regionales de más de 70 países gastan el dinero público de los contribuyentes. En este portal de información podemos encontrar, por ejemplo, el gasto de algunos de los partidos políticos en la campaña de los elecciones municipales del año 2015.

[2] `https://openspending.org`
[3] `https://translate.google.com`
[4] `http://www.findtoilet.dk`

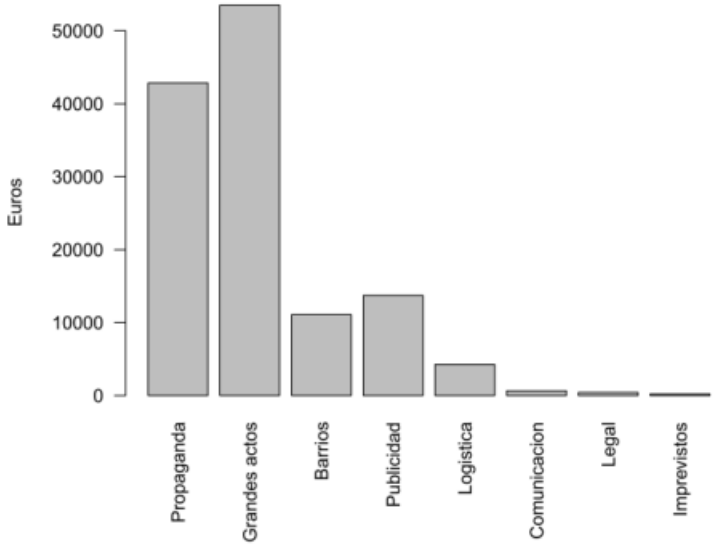


Figura 3. Gastos de la campaña electoral de las municipales 2015 de "Barcelona en Comú".

El partido "Barcelona en Comú" fue uno de los partidos políticos que publicaron de forma abierta los gastos de su campaña electoral. Si descargamos el fichero de datos, vemos que presenta la información en forma de fichero separado por comas (CSV):

```
ID;Fecha;Concepto;Detalle;Descripción;Importe
1;2015-09-30;Propaganda;Propaganda;Carteles 50x70 (15.000);951.06
```

Fichero CSV

Un fichero CSV (del inglés *Comma-Separated Values*) es un tipo de documento en formato abierto que permite representar datos en forma de tabla, donde las columnas se separan por comas y las filas por saltos de línea.

La primera línea indica los campos que vamos a encontrar, que corresponden a los siguientes valores: identificador de la línea, fecha del gasto, concepto asociado al gasto, detalle del gasto, descripción e importe. A partir de la segunda línea se presentan todos los gastos asociados, uno por línea, mostrando la información en el orden indicado anteriormente en la primera fila. En el ejemplo sólo hemos incluido el detalle del primer gasto, pero lógicamente en el fichero original se pueden encontrar todos los demás.

A partir de estos datos es muy sencillo poder extraer información agregada. Nos podemos preguntar, por ejemplo, cuánto gastó el partido "Barcelona en Comú" en cada una de las categorías en toda la campaña. Para obtener los datos deseados, sólo debemos de agrupar los gastos según su categoría y calcular el importe total de cada categoría. La Figura 3 muestra el resultado de ejecutar este proceso y generar un gráfico con los resultados. Como se puede apreciar, un volumen muy importante de gastos están asociados a las categorías "propaganda" y "grandes actos".

Aplicaciones *mashup*

Un claro ejemplo de integración y reutilización lo encontramos en las llamadas *mashup* o aplicaciones web híbridas. Estas aplicaciones son creadas a partir de la reutilización de contenidos y/o funcionalidad de sitios web de terceros que permiten el acceso a datos abiertos. A partir de distintas fuentes de datos, estas aplicaciones permiten crear nuevos servicios simples, visualizado en una única interfaz gráfica la combinación de los datos extraídos. Por ejemplo, se pueden combinar las direcciones y fotografías de una biblioteca con un mapa de Google Maps para crear un *mashup* de mapa.

API

Las interfaces de programación de aplicaciones, en inglés *Application Programming Interface* (API), son el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por un programa externo para extraer datos.

El término implica integración fácil y rápida, a menudo usando varias API abiertas y fuentes de datos para producir resultados enriquecidos, que no fueron necesariamente el motivo original de producir la fuente primaria de datos.

La arquitectura de los *mashups* está compuesta de tres partes principales:

- El proveedor de contenidos o fuente de los datos proporciona los datos abiertos que servirán como información base para crear el servicio que se desea implementar. Generalmente se accede a los datos a través de una API.

- El sitio *mashup* es la nueva aplicación web que provee un nuevo servicio utilizando diferente información mezclada a partir de diferentes fuentes abiertas.

- El navegador web cliente es la interfaz de usuario del *mashup*.

Por ejemplo, el departamento de policía de Chicago tiene un *mashup* llamado Chicago Crime[5] que integra la base de datos del departamento de crímenes reportados con Google Maps con el objetivo de ayudar a detener crímenes en ciertas áreas y avisar a los ciudadanos de áreas potencialmente peligrosas.

2.3. Publicación de datos en abierto

Cuando una organización o institución desea publicar datos en abierto, debe plantearse cinco pasos principales que le conducirán a una correcta publicación de los datos en abierto.

1) En primer lugar, debemos identificar el conjunto o conjuntos de datos que deseamos publicar.

2) A continuación, debemos seleccionar uno o más formatos de datos en los que queremos publicar los datos en abierto.

3) En tercer lugar, escogeremos una licencia abierta adecuada a los datos que deseamos publicar.

4) En cuarto lugar, daremos accesibilidad a los datos en un formato que resulte útil para el resto de la comunidad.

5) Finalmente, es conveniente dar visibilidad a los conjuntos de datos que hemos publicado en abierto.

A continuación veremos con un poco más de detalle cada uno de los pasos comentados.

2.3.1. Identificar los conjuntos de datos

El primer paso en el proceso de publicación de datos en abierto es escoger el conjunto o los conjuntos de datos que planee abrir. Este proceso es iterativo y podemos incluir

[5] `http://gis.chicagopolice.org/`

nuevos conjuntos de datos en el futuro. En general, no hay requisitos para crear una lista completa de conjuntos de datos que sean candidatos a ser publicados. Existen dos puntos principales a tener en cuenta:

1) En primer lugar, debemos asegurarnos de que es factible publicar todos (o una parte) de los datos.

2) En segundo lugar, y no menos importante, es asegurarse de que no haya datos personales o privados de personas individuales en el conjunto de datos que deseamos publicar. Generalmente se publican conjuntos de datos que no contienen datos de carácter personal. En caso contrario, deberemos aplicar ciertos procesos de anonimización y preservación de la privacidad que nos garanticen que los datos personales estarán correctamente protegidos en el conjunto de datos abiertos.

2.3.2. Seleccionar un formato para los datos

En segundo lugar, es importante escoger un formato adecuado para la publicación de los datos abiertos. El formato escogido dependerá de varios factores, pero en primer lugar veremos los tipos básicos de estructura o formato de datos que podemos utilizar en la publicación de los mismos.

La estructura de datos se define como la forma en que se encuentran organizados un conjunto de datos. Existen diferentes formatos y estructuras en que podemos representar un mismo conjunto de datos. Clasificaremos los datos según su nivel de estructuración en:

- **Datos estructurados**. La información viene representada por un conjunto o agrupación de datos atómicos elementales, es decir, datos simples que no están compuestos de otras estructuras. Se conoce de antemano la organización de los datos, la estructura y el tipo de cada dato elemental, su posición y las posibles relaciones entre los datos. Los datos estructurados son de fácil interpretación y manipulación.

  Los ficheros con una estructura fija en forma de tabla, como los ficheros CSV o las hojas de cálculo, son claros ejemplos de orígenes de datos estructurados.

- **Datos semiestructurados**. La información viene representada por un conjunto de datos elementales, pero a diferencia de los datos estructurados no tienen una estructura fija, aunque tienen algún tipo de estructura implícita o autodefinida.

  Ejemplos de este tipo de datos son, por ejemplo, los documentos XML o las páginas web. En ambos casos los documentos siguen ciertas pautas comunes, pero sin llegar a un nivel de estructuración fija.

- **Datos no estructurados**. La información no aparece representada por datos elementales, sino por una composición cohesionada de unidades estructurales de

nivel superior. La interpretación y manipulación de estos datos resulta mucho más compleja que el de los estructurados o semiestructurados.

Ejemplos de orígenes de datos no estructurados son textos, audios, imágenes o vídeos.

A continuación veremos algunos de los tipos de archivos más utilizados en la publicación de datos:

- Fichero PDF (formato de documento portátil o en inglés, *Portable Document Format*). Es un formato no estructurado de almacenamiento para documentos digitales multiplataformas que pueden incorporar texto, imágenes vectoriales y mapas de bits.

- Fichero XLS o XSLX. Es un formato estructurado propietario de Microsoft Office para la hoja de cálculo Microsoft Excel utilizado en tareas financieras y contables.

- Fichero de valores separados por comas (CSV, del inglés *Comma-Separated Values*). Es un tipo de documento estructurado en formato abierto que permite representar datos en forma de tabla, donde las columnas se separan por comas y las filas por saltos de línea. Existen variaciones del mismo formato en donde las columnas se separan utilizando otros caracteres, por ejemplo, tabulaciones (TSV, del inglés *Tab-Separated Values*).

- Fichero XML. Un fichero XML (del inglés *eXtensible Markup Language*) es un tipo de documento semiestructurado, compuesto por datos elementales pero de definición no previamente conocida, que incluye etiquetas para describir su propia definición.

- Fichero JSON (*JavaScript Object Notation*, en inglés). Es un estándar abierto basado en texto diseñado para el intercambio de datos legible por humanos, que permite representar estructuras de datos simples y listas asociativas.

- Fichero RDF (del inglés *Resource Description Framework*). Es una especificación que propone un modelo de datos para describir vocabularios y enlazar datos de distintos ámbitos. Los datos se relacionan mediante tripletas. Veremos qué es, cómo funciona y como consultarlo en los siguientes capítulos de este material. Es el lenguaje utilizado para enlazar datos mediante *linked data*.

- Fichero KML (del acrónimo en inglés *Keyhole Markup Language*). Es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones.

El tipo de fichero que debemos utilizar para generar el conjunto de datos abiertos depende, en gran medida, del tipo de datos que deseamos publicar. Por ejemplo, si deseamos publicar datos en formato de tabla, entonces es aconsejable emplear

un tipo de fichero estructurado, como por ejemplo XLS o CSV. Por el contrario, si deseamos publicar datos con una cierta estructura pero que nos permitan flexibilidad, una opción interesante pueden ser los formatos semiestructurados, como por ejemplo XML o JSON. Finalmente, si los datos que deseamos publicar se basan en la localización o puntos de coordenadas GPS podemos emplear el formato KML. Obviamente, se pueden utilizar distintos formatos para publicar unos mismos datos. Sea cual sea el formato elegido, es siempre aconsejable utilizar también el formato RDF. Así podremos enlazar nuestros datos con otros datos y permitir que sean interpretables por programas informáticos.

Por otro lado, siempre es aconsejable utilizar formatos de ficheros abiertos. En caso contrario, sólo los usuarios que dispongan de la plataforma propietaria podrán cargar y utilizar el conjunto de datos.

> Por ejemplo, el servicio estadístico Eurostat[6] ofrece más de 4 000 conjuntos de datos abiertos, actualizados regularmente y se obtienen, generalmente, en el formato estructurado y abierto TSV.
>
> Otro ejemplo podría ser el catálogo de datos del Distrito de Columbia[7] (*District of Columbia Data Catalog*), que ofrece conjuntos de datos abiertos en multitud de formatos, como por ejemplo CSV, XLS o KML, y que además ofrece también acceso a los mismos datos a través de una API.

2.3.3. Escoger una licencia abierta

Una vez hayamos seleccionado el conjunto o conjuntos de datos y hayamos escogido el formato de publicación de los mismos, debemos escoger una licencia abierta para la publicación de dichos datos. Es importante seleccionar y aplicar una licencia que establezca de forma clara los usos posibles de los datos publicados.

En este contexto hay dos atributos de las licencias abiertas que son de especial importancia para seleccionar aquella licencia que mejor se adapte a nuestras necesidades:

- **Atribución** (BY, *attribution*): indica que el conjunto de datos sólo puede ser reutilizado si se reconoce la autoría original en la nueva publicación.

- **Compartir igual** (SA, *share-alike*): indica que el conjunto de datos sólo puede ser reproducido o utilizado como base para la creación de un nuevo conjunto de datos si también se hace bajo una licencia abierta.

Existen multitud de licencias aplicables a datos o conjuntos de datos. A continuación veremos algunas recomendaciones de licencias aptas para ser aplicadas a conjuntos de datos. Para cada una de ellas se indica si sólo es aplicable a datos o también a otro tipo de contenidos, como por ejemplo software.

[6] `http://ec.europa.eu/eurostat`
[7] `http://opendata.dc.gov/`

Tabla 1. Ejemplos de licencias abiertas para conjuntos de datos. Fuente: `http://opendefinition.org/licenses/`.

Creative Commons CCZero(CC0)	Contenido y Datos	No	No	Dominio público
Open Data Commons PublicDomain Dedication andLicence (PDDL)	Datos	No	No	Dominio público
Creative Commons Attribution4.0 (CC-BY-4.0)	Contenido y Datos	Si	No	Atribución de autoría
Open Data Commons AttributionLicense (ODC-BY)	Datos	Si	No	Atribución de autoría
Creative Commons AttributionShare-Alike 4.0 (CC-BY-SA-4.0)	Contenido y Datos	Si	Si	Atribución y compartir igual
Open Data Commons OpenDatabase License (ODbL)	Datos	Si	Si	Atribución y compartir igual

Podemos encontrar una lista más exhaustiva en, por ejemplo, los siguientes sitios web:

- Recomendaciones de la *Open Definition*, disponible en `http://opendefinition.org/licenses/`.
- Breve guía del sitio *Open Data Commons* disponible en `http://opendatacommons.org/guide/`.

OpenData BCN: el Portal *Open Data* del Ayuntamiento de Barcelona

El Ayuntamiento de Barcelona publica un catálogo de más de 300 conjuntos de datos abiertos, clasificados en distintas categorías, como por ejemplo, administración, ciudad y servicios, economía y empresa, población y territorio. Todos los conjuntos de datos que se ofrecen en el servicio OpenData BCN[8] indican qué licencia y condiciones de uso tienen.

La mayoría de conjuntos de datos se publican bajo los términos de la licencia *Creative Commons* - Reconocimiento (CC-BY 3.0), que permite:

- que se puedan copiar, distribuir y divulgar públicamente.
- que puedan servir de base para obras derivadas como resultado de su análisis o estudio.
- que puedan ser utilizadas con fines comerciales o no comerciales, siempre que este uso no constituya una actividad administrativa pública.
- que se puedan modificar, transformar y adaptar.
- que se deba mencionar la autoría del Ayuntamiento de Barcelona (tal como se indica más adelante).

Sin embargo, en el caso de tipo de datos en los que hay participación de terceros, la reutilización se vehicula a través de la licencia *Creative Commons* - Reconocimiento - SinObraDerivada (CC BY-ND 3.0) , que permite:

- que se puedan copiar, distribuir y divulgar públicamente.
- que puedan ser utilizadas con fines comerciales o no comerciales, siempre que este uso no constituya una actividad administrativa pública.
- que se puedan modificar, transformar y adaptar.
- que se deba mencionar la autoría de los datos.

Además, de acuerdo con el artículo 8 de la Ley 37/2007, la reutilización de la información contenida en los conjuntos de datos está sometida a las condiciones generales siguientes:

[8] `http://opendata.bcn.cat/`

- que el contenido de la información no sea alterado.
- que no se desnaturalice el sentido de la información.
- que se mencione la fuente.
- que se mencione la fecha de la última actualización.

Se puede consultar las condiciones de uso completas del portal OpenData BCN en la dirección `http://opendata.bcn.cat/opendata/es/data-using`.

2.3.4. Asegurar la accesibilidad

Para cerciorarnos que los datos abiertos son realmente "abiertos" hay que asegurarse que enfocamos correctamente dos puntos clave. En primer lugar, los datos deben ser abiertos desde un punto de vista legal. Este punto lo hemos trabajado en el apartado anterior. Pero además, en segundo lugar, los datos deben ser abiertos desde un punto de vista técnico. Es decir, debemos facilitar que sean accesibles de forma sencilla y, preferiblemente, que sean datos legibles por una máquina.

Datos legibles en formato máquina

Los datos legibles en formato máquina (en inglés, *machine-readable format*) son aquellos datos que pueden ser leídos y procesados de forma automática por un computador. Por ejemplo, los documentos de hojas de cálculo son legibles en formato máquina, mientras que una imagen o documento PDF necesita de un proceso específico para que una máquina pueda interpretar la información contenida en él.

Para ilustrar la importancia de la accesibilidad de los datos, supongamos que nuestra organización publica ciertos datos estadísticos en abierto como un documento PDF. Es cierto que este formato favorece la lectura por parte de personas, pero dificulta tremendamente que una máquina pueda leer y entender los datos contenidos en el documento. Por lo tanto, limita en gran medida la reutilización de los datos por parte de otras organizaciones.

Existen muchas alternativas para hacer que nuestros datos sean disponibles para otras organizaciones de forma rápida y eficiente. La forma más natural es la publicación en línea a través de Internet. En su forma más básica, las instituciones o organizaciones pueden publicar en abierto sus datos a través de sus propios sitios web. Sin embargo, cuando la conectividad es limitada o el tamaño de los datos es extremadamente grande, la distribución a través de otros formatos puede presentar múltiples ventajas. A continuación veremos algunas de las alternativas más relevantes y habituales:

- A través del sitio web de la institución u organización. Como hemos comentado, esta es la opción más elemental y sencilla de implementar en muchos contextos. Generalmente los costes del almacenamiento de los datos y del tráfico generado por las descargas de los usuarios son muy bajos, por lo que esta es una opción muy interesante para la publicación de datos de un tamaño razonable.

- A través del sitio web de terceras partes. Existen repositorios de datos generalistas y también repositorios especializados en distintos campos. Los sitios web de terceras partes pueden ser muy útiles, dado que generalmente facilitan el acceso a una comunidad de personas interesadas y ponen en común diversos conjuntos de datos similares o complementarios. Además, este tipo de plataformas proporcionan: (1) una infraestructura adecuada que puede soportar un volumen de descargas importante; (2) a menudo ofrecen análisis e información de uso; y (3)

para las agencias del sector público suelen ser gratuitas.

- A través de servidores FTP. Otra alternativa, aunque en cierto desuso actualmente, es proporcionar acceso a los archivos a través del protocolo de transferencia de archivos (FTP). Esta alternativa puede ser viable si el conjunto de datos que deseamos publicar va dirigido a un colectivo de carácter técnico, como desarrolladores de software o científicos.

FTP

El protocolo de transferencia de archivos (FTP) (en inglés, *File Transfer Protocol*) es un protocolo de red para la transferencia de archivos entre computadoras conectadas a una red, generalmente, a Internet. Desde un equipo cliente se puede conectar a un servidor para descargar archivos, independientemente del sistema operativo utilizado en cada equipo.

- A través de las redes P2P. Las redes punto-a-punto (P2P) trabajan dividiendo el coste de distribución de los archivos entre todas las computadoras que acceden a estos archivos. Es decir, en lugar de que los servidores sean los responsables de enviar la información a todos los clientes que desean acceder a ella, es la propia comunidad conectada a la red P2P quién distribuye partes de la información entre todos los usuarios. Es una alternativa eficiente para la distribución de volúmenes muy grandes de datos.

- A través de una API. Los datos pueden ser publicados a través de una interfaz de programación de aplicaciones (API). Estas interfaces han ganado mucha popularidad recientemente. Algunas de las principales redes sociales, como por ejemplo Twitter o Facebook, ofrecen acceso a los datos a través de este tipo de interfaces. Éstas permiten a los programadores seleccionar qué tipo o parte de los datos desean descargar, en lugar de proporcionar todos los datos en un archivo de grandes dimensiones. Una ventaja importante es que las APIs suelen estar conectadas a una base de datos que está siendo actualizada en tiempo real. Esto significa que el suministro de información a través de una API puede proporcionar datos actualizados de forma automática. Es decir, si queremos publicar datos en un intervalo de tiempo corto, el uso de APIs nos puede facilitar tremendamente el trabajo, evitando el coste de generar y actualizar grandes ficheros de forma continua. Por el contrario, hay que considerar que el coste de desarrollo de una API es muy superior al coste de generar un fichero y distribuirlo a través de las opciones anteriores. Por otro lado, también hay que tener en cuenta que en el caso de utilizar una API, toda la descarga de los datos se realizará a través de este sistema, de forma que los usuarios que hayan obtenido los datos difícilmente podrán redistribuirlos a terceros.

- A través de un punto de acceso SPARQL. Como hemos comentado antes, RDF es un modelo de datos que permite representar datos y la relación que existe entre ellos. SPARQL es un protocolo (y lenguaje) de consulta propuesto por el **World Wide Web Consortium**[9] que permite consultar datos en formato RDF. Los puntos de acceso SPARQL (o *SPARQL endpoints* en inglés) son servicios web que permiten consultar datos de un determinado conjunto de datos abiertos en formato RDF. Por tanto, si ofrecemos un punto de acceso SPARQL para proporcionar acceso a nuestros datos estos serán consultables para cualquiera de forma interactiva. Por ejemplo, podríamos ejecutar la siguiente consulta SPARQL en el punto de acceso SPARQL de la wikipedia (ubicado en

[9] `http://www.w3c.org`

http://dbpedia-live.openlinksw.com/sparql/). La consulta nos devolvería la lista de recursos de la wikipedia que tienen asociada una URL que contenga el texto "disney".

```
SELECT *
WHERE {
      ?s foaf:homepage ?homepage
      FILTER REGEX(?homepage, ".*disney.*", "i")
}
```

Ejemplos de publicación de datos en abierto

En la siguiente URL https://www.w3.org/wiki/SparqlEndpoints podéis encontrar una lista de puntos de acceso SPARQL.

Veamos, brevemente, algunos ejemplos de publicación de datos utilizando los distintos métodos que hemos comentado anteriormente.

En primer lugar, podemos encontrar muchísimas instituciones y organizaciones que publican sus datos en abierto a través de sus propios sitios web. Por ejemplo, el Ayuntamiento de Barcelona publica los datos a través de su OpenData BCN[10], el Gobierno de Aragón a través del portal Aragón Open Data[11] y el Gobierno de España a través del portal Aporta[12].

En segundo lugar, existen una gran cantidad de portales destinados a la publicación de datos de terceras partes. Por ejemplo, el portal AWS Amazon[13] ofrece un repositorio público multidisciplinar con numerosos conjuntos de datos, el proyecto SNAP[14] de la Universidad de Stanford proporciona acceso a conjuntos de datos de todo tipo de redes (sociales, de comunicación, de transportes, etc.), o el portal Research Data Australia[15], que agrupa multitud de recursos abiertos sobre humanidades, ciencias sociales, medicina, tecnología, agricultura y un largo etcétera del gobierno e instituciones Australianas.

Como hemos comentado anteriormente, el acceso a través de servidores FTP se encuentra en cierto desuso. Aún así, algunos recursos, como por ejemplo el *Astronomical Data Archives Center*[16] del *National Astronomical Observatory of Japan* aún son accesibles a través de esta tecnología.

Finalmente, podemos encontrar multitud de ejemplos de instituciones u organizaciones que publican sus datos de forma abierta a través de una interfaz de programación de aplicaciones (API). Por ejemplo, el sitio web Socrata[17] alberga más de cien conjuntos de datos de gobiernos y organizaciones no lucrativas de todo el mundo a través de una API. El mismo *World Bank*[18] proporciona una gran cantidad de conjuntos de datos abiertos sobre desarrollo en distintos países del mundo a través de su API.

2.3.5. Facilitar el descubrimiento de los datos

Para cerrar de forma satisfactoria el círculo de publicación de datos en abierto, es imprescindible que consigamos conectar los potenciales usuarios con los datos abiertos. En el caso contrario, los datos publicados no tendrán la utilidad que se les supone. Por lo tanto, debemos conseguir que los datos abiertos puedan ser descubiertos por toda la comunidad de usuarios potenciales.

Actualmente, podemos encontrar una serie de herramientas o sitios web que están diseñados específicamente para dar visibilidad a los datos abiertos. La misma *Open*

[10] http://opendata.bcn.cat
[11] http://opendata.aragon.es
[12] http://datos.gob.es
[13] http://aws.amazon.com/datasets/
[14] https://snap.stanford.edu/data/
[15] https://researchdata.ands.org.au/
[16] ftp://dbc.nao.ac.jp/
[17] https://dev.socrata.com
[18] http://data.worldbank.org/

Knowledge Foundation nos ofrece dos herramientas que nos permiten dar visibilidad a nuestros datos.

Por un lado, CKAN[19] es una herramienta para la gestión y publicación de colecciones de datos. Esta herramienta ha sido utilizada por distintos gobiernos nacionales y locales, instituciones de investigación y otras organizaciones que recogen una gran cantidad de datos. Los usuarios, sean ciudadanos, desarrolladores, periodistas o investigadores entre otros, pueden buscar datos, registrar conjuntos de datos publicados, crear y administrar grupos de conjuntos de datos, y obtener actualizaciones de bases de datos y los grupos que les interesan.

Por otro lado, podemos obtener libre acceso a muchas de las funciones básicas CKAN a través del sitio web DataHub[20]. Este sitio web facilita que instituciones y organizaciones puedan publicar el material y que los usuarios de datos puedan encontrar el material que necesitan.

Adicionalmente, existen decenas de catálogos especializados en diferentes territorios. El más relevante a nivel europeo quizá es **European Data Portal** (accesible desde `https://www.europeandataportal.eu/`. Por otro lado, existen catálogos especializados para distintos sectores. Como por ejemplo **TourPedia**[21], que ofrece datos en abierto sobre turismo o muchas comunidades científicas, que han creado un sistema de catálogo de sus campos, ya que compartir los datos que usan en sus publicaciones puede ser un requerimiento o puede ayudarlos en la comparación de nuevos métodos y técnicas.

2.4. Decálogo de buenas prácticas

Dentro del mundo de Internet, el *World Wide Web Consortium* (W3C)[22] es el organismo que se encarga de velar por el desarrollo de estándares abiertos, libres e interoperables que aseguren el crecimiento de la Web a largo plazo. Esta organización ha realizado una guía de publicación con pautas sobre cómo han de publicar datos los gobiernos[23]. Igualmente existen otras iniciativas generadoras de manuales de buenas prácticas o de concienciación alrededor de los datos abiertos como las aportadas por ejemplo por la *Sunlight Foundation*[24] o por la *Open Knowledge Foundation*[25].

[19] `http://ckan.org/`
[20] `http://datahub.io/`
[21] `http://tour-pedia.org/about/`
[22] `http://www.w3c.es/`
[23] `http://www.w3.org/TR/gov-data/`
[24] `http://sunlightfoundation.com/policy/opendata/`
[25] `http://okfn.org/opendata/`

Figura 4. Buenas prácticas para la publicación de datos abiertos. Fuente: `http://opendata.aragon.es/portal/open-data`.

Además, dentro del ámbito estatal, la "Comunidad *Open Data* - Reutilización de Información del Sector Público en España" está trabajando en la sensibilización hacia las políticas de *Open Data* y ha generado algunos documentos de interés. Entre ellos destaca el Decálogo *Open Data*[26], que es un resumen de buenas prácticas a la hora de afrontar políticas *Open Data*, que transcribimos a continuación y podemos ver en la Figura 4.

0) Armonización entre administraciones. Todos los puntos del decálogo se basan en la premisa de que debe existir una armonización entre todas las Administraciones. Todas las iniciativas *Open Data* deben compartir los mismos principios y definiciones que se listan en el decálogo. Este punto 0 es básico para la interoperabilidad y aprovechamiento eficiente de las sinergias llevadas a cabo por todos los actores *Open Data* - RISP.

1) Publicar datos en formatos abiertos y estándares. Cualquier iniciativa *Open Data* debería publicar sus conjuntos de datos en formatos abiertos (no-propietarios) y que sean adecuados para permitir la reutilización de los mismos por parte del colectivo reutilizador destinatario.

2) Usar esquemas y vocabularios consensuados. Además de los formatos abiertos y estándar, la estructura de los datos debería seguir un convenio o unos esquemas definidos, si existieran. Si se crean vocabularios o esquemas de representación de la información específicos, éstos se deberían exponer públicamente para que el colectivo reutilizador pueda interpretar correctamente la información.

3) Inventario en un catálogo de datos estructurado. Cualquier iniciativa *Open Data* debe tener un punto de consulta donde se incluya un inventario con información descriptiva y técnica sobre los conjuntos de datos que se exponen. Los metadatos que informan sobre cada conjunto de datos deberían seguir una estructura común y estándar. Asimismo, se deberían compartir las taxonomías de temáticas u otras necesarias -p.e., toponimia- para clasificar los conjuntos de datos dentro de los catálogos.

4) Datos accesibles desde direcciones web persistentes y amigables. Tanto las fichas de los conjuntos de datos, como la distribución de la propia información (volcado en un archivo, API de consulta, RSS, etc.) deberían de estar accesibles desde URLs (direcciones web) que persistan en el tiempo y así evitar que se pierdan las referencias en el futuro. Además deben seguir una estructura homogénea y bien definida, con información legible para que los reutilizadores conozcan o "intuyan" el contenido referido por dichas direcciones web.

5) Exponer un mínimo conjunto de datos relativos al nivel de competencias del organismo y su estrategia de exposición de datos. Cada Administración que impulse una iniciativa *Open Data* debería crear una hoja de ruta donde especifique la estrategia de exposición de los conjuntos de datos y sus prioridades. Inicialmente, debería publicar los conjuntos de mayor interés según las competencias del propio organismo.

[26] `http://red.gnoss.com/comunidad/OpenData/recurso/Decalogo-Open-Data/58581882-63aa-4bc5-9033-90cf81f78793`

6) Compromiso de servicio, actualización y calidad del dato, manteniendo un canal eficiente de comunicación reutilizador. La Administración debe mantener un mínimo de calidad y servicio en su iniciativa *Open Data*, manteniendo lo expuesto en la estrategia de publicación y comprometiéndose con su colectivo reutilizador. Debe establecer un canal eficiente de comunicación que permita la interacción bidireccional organismo público - reutilizadores.

7) Monitorizar y evaluar el uso y servicio mediante métricas. La Administración debe crear métricas y evaluar sus indicadores de uso y servicio de la iniciativa *Open Data*. De esta forma puede monitorizar el funcionamiento y uso, y así analizar si se está cumpliendo el compromiso con la comunidad de reutilizadores y cuales son las potenciales carencias del sistema o de la estrategia.

8) Datos bajo condiciones de uso no restrictivas y comunes. Las condiciones de uso deberían ser lo menos restrictivas posible y permitir la reutilización libre, incluso para fines comerciales. Se recomienda la creación y uso de licencias tipo, autodocumentadas y que sean comunes entre distintas administraciones.

9) Evangelizar y educar en el uso de datos. Es necesario educar en el uso de los datos, tanto a los colectivos de reutilización específicos (sector TIC, periodismo, investigación, etc.) como a la sociedad en general y así fomentar el conocimiento y la inquietud por procesar información de una forma autónoma. Evitar el "disgusto" por los datos.

10) Recopilar aplicaciones, herramientas y manuales para motivar y facilitar la reutilización. Cualquier iniciativa *Open Data* debería recopilar ejemplos de uso y herramientas que faciliten y motiven la reutilización de los datos que se publican.

2.5. Ejemplos de *Open Data*

Para finalizar este módulo didáctico sobre los datos abiertos, veremos en este capítulo algunos ejemplos interesantes de datos abiertos en distintos ámbitos de la sociedad.

2.5.1. Administraciones locales

En la actualidad la mayoría de las administraciones o ayuntamientos de las grandes ciudades disponen de sitios web en donde ofrecen datos abiertos.

El Ayuntamiento de Barcelona

El Ayuntamiento de Barcelona dispone del portal OpenData BCN[27] de datos abiertos relacionados con la ciudad.

[27] `http://opendata.bcn.cat/`

El propio ayuntamiento define los datos abiertos como:

> La apertura de datos públicos – también conocido como *Open Data* – consiste en poner la información que posee el sector público al alcance de todo el mundo en formatos digitales, estandarizados y abiertos, siguiendo una estructura clara que permita la comprensión. Al mismo tiempo se facilita el acceso a esta información con el fin de fomentar la reutilización.
>
> **Ayuntamiento de Barcelona**, *¿Qué es Open Data?*. Disponible en `http://opendata.bcn.cat/opendata/es/what-is-open-data`

Su catálogo de datos abiertos dispone de más de 300 conjuntos de datos, agrupados en las siguientes categorías:

- **Administración**. Incluye conjuntos de datos relacionados con temas de legislación, justicia y sector público.

- **Ciudad y servicios**. Dentro de esta categoría se incluyen todos los conjuntos de datos abiertos que hacen referencia a cultura, ocio, deporte, medio ambiente, seguridad, transporte y turismo de la ciudad.

- **Economía y empresa**. Integra los datos relacionados con ciencia, tecnología, comercio y empleo de la ciudad.

- **Población**. Agrupa los conjuntos de datos relacionados con demografía, educación, sociedad y bienestar.

- **Territorio**. Finalmente, esta categoría contiene datos sobre temas de urbanismo, infraestructuras y vivienda.

Los datos se publican en distintos formatos, la mayoría de los cuales han sido comentados en este módulo didáctico. Los más utilizados corresponden a datos estructurados en formato de tablas, como por ejemplo CSV, XLS, XLSX. Aún así, también encontramos múltiples documentos en formatos semiestructurados, como JSON o XML; formatos para uso específico de geolocalización, como GEO y KML; y también formatos no estructurados como ficheros de texto plano (TXT) o documentos PDF.

Además, el sitio web dispone también de una herramienta que permite la visualización de algunos de los conjuntos de datos sobre un mapa de la ciudad. Por ejemplo, la Figura 5 muestra todas las fuentes de agua públicas en la ciudad de Barcelona.

Figura 5. Fuentes públicas de Barcelona[28].

También es relevante notar que la mayoría de los conjuntos publicados en este portal se acogen a la licencia Creative Commons – Attribution 3.0[29]. Según esta licencia, se permite cualquier explotación de la obra, incluyendo una finalidad comercial, así como la creación de obras derivadas, la distribución de las cuales también está permitida sin ninguna restricción.

El Ayuntamiento de Madrid

De forma similar al caso visto anteriormente, el Ayuntamiento de Madrid también dispone de un portal de datos abiertos[30].

En el mismo portal podemos encontrar la definición de datos abiertos que nos brinda el ayuntamiento:

> Datos Abiertos (*Open Data* en inglés) es una iniciativa global, ligada a las políticas de Gobierno Abierto, que persigue que los datos y la información, especialmente las que poseen las administraciones públicas, se publiquen de forma abierta, regular y reutilizable para todo el mundo, sin restricciones de acceso, copyright, patentes u otros mecanismos de control.
>
> La filosofía origen de estas iniciativas es fomentar la transparencia, la eficiencia, la participación ciudadana y el desarrollo económico.
>
> **Ayuntamiento de Madrid**, *¿Qué son Datos Abiertos?*.

El portal de datos abiertos del Ayuntamiento de Madrid publica los datos en distintos formatos. Preferiblemente se utilizan formatos no propietarios, aunque también se genera información en algún formato propietario que sea considerado un estándar de facto por la mayor parte de la ciudadanía. Destacamos algunos que no hemos visto durante el desarrollo de este módulo didáctico:

[29] `https://creativecommons.org/licenses/by/3.0/legalcode`
[30] `http://datos.madrid.es/portal/site/egob/`

- RSS (en inglés *Really Symple Sindication*): Es un formato XML para la distribución de contenidos de páginas web. Facilita la publicación de información actualizada a los usuarios suscritos a la fuente RSS sin necesidad de usar un navegador, utilizando un software especializado en este formato.

- SHP (*Shapefile*): Es un formato propietario estándar de datos espaciales, desarrollado por la compañía ESRI, que almacena tanto la geometría como la información alfanumérica. Este formato no está preparado para almacenar información topológica. Actualmente se ha convertido en formato estándar de facto para el intercambio de información geográfica entre Sistemas de Información Geográfica por la importancia que los productos ESRI tienen en el mercado de sistemas de información geográfica y por estar muy bien documentado.

- GPX o GPS (Formato de Intercambio GPS, en inglés *eXchange Format*): Es un esquema XML pensado para representar información para navegadores. Se puede usar para describir puntos (*waypoints*), recorridos (*tracks*), y rutas (*routes*).

- MDB: Extensión de archivo utilizado en ciertas versiones de bases de datos Microsoft Access.

- ZIP: es un formato de compresión sin pérdida, muy utilizado para la compresión de documentos, imágenes o programas.

El portal dispone de más de 200 conjuntos de datos de muy diversas temáticas, como por ejemplo ciencia, comercio, cultura, educación, empleo o energía.

Ayuntamiento de Londres

Para finalizar esta sección, en la que hemos visto diferentes ejemplos de datos municipales en abierto, veremos el caso de London Datastore[31], el portal de datos abiertos del Ayuntamiento de Londres.

En este portal podremos encontrar más de 600 conjuntos de datos abiertos relacionados multitud de categorías, como arte, cultura, crimen y seguridad, educación, medio ambiente, transparencia y transporte. Al igual que en los casos anteriores, los conjuntos de datos pueden ser descargados en distintos formatos de datos estructurados, como XLS o CSV; datos semiestructurados, como XML o HTML; y en formatos no estructurados, como PDF, Microsoft Word o Microsoft Powerpoint.

Pero además, este portal ofrece una API basada en la plataforma CKAN[32] e incluye una sección dedicada al desarrollo de aplicaciones utilizando el lenguaje estadístico R, con interesantes ejemplos y código fuente para iniciarse en la ciencia de los datos.

[31] `http://data.london.gov.uk/`
[32] `http://ckan.org/`

2.5.2. Administraciones regionales

En esta sección veremos algunos ejemplos de administraciones públicas de ámbito nacional y regional.

Generalitat de Catalunya

El portal de datos abiertos de la Generalitat de Catalunya[33] ofrece conjuntos de datos abiertos del ámbito autonómico catalán.

El el mismo portal podemos encontrar otra definición del significado de datos abiertos, aunque muy similar a las demás definiciones que hemos ido presentando a lo largo de este módulo didáctico. Según la Generalitat, los datos abiertos son:

> El *Open data* es una filosofía y una práctica que requiere que ciertos datos sean de libre acceso para todo el mundo, sin limitaciones técnicas o legales. En el sector público, tener acceso a los datos de la Administración garantiza la transparencia, la eficiencia y la igualdad de oportunidades, a la vez que se crea valor. La transparencia porque se pueden consultar y tratar datos que vienen directamente de las fuentes oficiales, las eficiencia porque ciudadanos y organizaciones pueden crear servicios de forma más ajustada en colaboración con la administración; y la igualdad de oportunidades porque el acceso es el mismo para todo el mundo.
>
> **Generalitat de Catalunya**, *¿Qué es el Open Data?*. Disponible en el portal web[34]

La Generalitat dispone de un catálogo de casi 3 000 conjuntos de datos abiertos, que al igual que en los casos vistos anteriormente, contempla conjuntos de datos de multitud de temáticas, como cartografía, territorio, urbanismo, agricultura o movilidad entre muchas otras.

Gobierno Vasco

El portal Open Data Euskadi[35] dispone de cerca de 500 conjuntos de datos abiertos con información relacionada con la comunidad autónoma del País Vasco.

El sitio web ofrece una sección en donde se presentan distintas ideas y ejemplos de uso de los datos abiertos publicados en el mismo portal. Podemos ver proyectos en distintos ámbitos, como por ejemplo:

- Calidad de las aguas de Euskadi: Existen en Euskadi docenas de estaciones que analizan periódicamente la calidad del aire y del agua. Los datos se recopilan y son públicos, en este mismo portal web. Con los datos de calidad del agua de consumo en Euskadi, se ha creado una aplicación con un mapa interactivo de consulta.

[33] `http://dadesobertes.gencat.cat/es/`
[34] `http://dadesobertes.gencat.cat/es/que_es_l_open_data/concepte/index.html`
[35] `http://opendata.euskadi.eus/`

- Euskadi costa y olas. Una aplicación móvil que ofrece información detallada sobre la costa de Euskadi a nivel geográfico y de oleaje.

- Euskalmeteo. Una aplicación móvil de predicción meteorológica de varios días. Datos en tiempo real obtenidos directamente de Euskalmet y de la agencia World Weather Online.

2.5.3. A nivel nacional/supranacional

En esta sección veremos algunos ejemplos de datos abiertos de ámbito nacional y supanacional.

Comunidad Europea

La Unión Europea hace tiempo que apuesta por una política de apertura de datos. En este sentido ha potenciado políticas que promulguen la publicación de datos abiertos en Europa a todos los niveles.

Actualmente, el portal de datos abiertos de la unión europea[36] ofrece más de 500.000 conjuntos de datos en abierto. Estos conjuntos de datos se extraen automáticamente de 73 portales web de datos abiertos pertenecientes al sector público (a nivel nacional y regional).

Los datos pueden buscarse en función de su origen (de que portal se han extraido o a qué país se refieren), de su idioma, de su categoría temática (de qué tratan) o de su contenido (mediante una búsqueda por palabras clave). Las categorías permitidas en el portal son las siguientes:

1) Agricultura, pesca, bosques y alimentación

2) Energía

3) Regiones y ciudades

4) Economía y finanzas

5) Internacional

6) Govierno y sector público

7) Justicia, sistema legal y seguridad pública

8) Entorno

9) Educación, cultura y deporte

[36] El *European Data Portal* es accesible desde el siguiente enlace `https://www.europeandataportal.eu/`

10) Salud

11) Población y sociedad

12) Ciencia y tecnología

El portal pretende no sólo un punto de acceso a datos en abierto, sino también fomentar una cultura más propensa al uso de datos abiertos. Para ello, fomenta la accesibilidad a los datos en abierto que ofrece, analiza el valor que aportan sus datos y provee información, mediante cursos de *eLearning*, sobre qué son los datos abiertos y cómo usarlos.

Respecto a la accesibilidad, el portal permite descargar los conjuntos de datos para su posterior tratamiento y acceder a los datos en línea mediante consultas SPARQL. Obviamente, no todos los datos podrán ser consultados mediante SPARQL, sólo aquellos que se hayan guardado utilizando la filosofía de *linked data* explicada en el siguiente capítulo.

Estados Unidos

El portal de datos abiertos de Estados Unidos de América[37] contiene actualmente más de 190.000 conjuntos de datos. Al igual que en el caso europeo, este portal contiene conjuntos de datos generados por las organizaciones públicas del país.

Los datos pueden buscarse en función de su origen (que organismo los ha generado), de su categoría temática (de qué tratan) o de su contenido (mediante una búsqueda por palabras clave). Los datos se clasifican en las siguientes categorías:

1) Agricultura

2) Clima

3) Consumidores

4) Ecosistemas

5) Educación

6) Energía

7) Finanzas

8) Salud

9) Gobierno local

10) Manufactura

11) Maritima

[37] Accesible desde `https://www.data.gov/`

12) Oceano

13) Seguridad Pública

14) Ciencia y investigación

A efectos de gestion de los datos, el portal ofrece información sobre la última actualización de datos de cada conjunto de datos y el organismo encargado de hacer dicha actualización. Al igual que en su homologo europeo, el portal provee información sobre el impacto que han tenido los datos en abierto en Estados Unidos y sobre las aplicaciones que usan sus datos en abierto. También ofrece información para desarrolladores, mediante tutoriales y guías que explican cómo acceder a los datos.

2.5.4. Otros ejemplos

Existen multitud de servicios que se han creado a partir de iniciativas parecidas de publicación de datos por parte de administraciones públicas. A continuación veremos, a modo de ejemplo, algunos servicios relacionados con distintas temáticas, como:

- servicios relacionados con la educación. Un ejemplo es "Schoolscope"[38], un sencillo servicio que informa sobre la calidad de la educación en cada a una de las escuelas inglesas y permite contrastar la información;

- sobre la salud. Una interesante iniciativa americana es el Apps for Healthy Kids[39], un concurso de servicios pensados para promover una vida saludable entre los niños;

- sobre la seguridad ciudadana. Una aplicación americana para móvil muy conocida es "AreYouSafe?"[40] que muestra el grado de delincuencia en función del lugar en el que uno se encuentra;

- la transparencia en el sector público. El servicio "¿Dónde van mis impuestos?"[41] explica, de forma muy visual, los gastos anuales de la Administración central del Estado y la Seguridad Social, tal como aparecen recogidos en los presupuestos generales, tal y como podemos ver en la Figura 6.

[38] `http://www.schooloscope.com/`
[39] `http://www.appsforhealthykids.com/`
[40] `http://areyousafedc.com/`
[41] `http://dondevanmisimpuestos.es/`

Figura 6. España: Gastos (por programa) según el portal "¿Dónde van mis impuestos?".
Fuente: `http://dondevanmisimpuestos.es/`.

3. *Linked Data*

.

Publicar los datos en abierto es el primer paso para permitir que terceras personas saquen provecho de nuestros datos para crear aplicaciones y servicios de valor añadido o generen nuevo conocimiento. No obstante, con hacer públicos los datos no es suficiente. Vivimos en un mundo interconectado, con distintas aplicaciones o sistemas que recogen datos de forma autónoma. Si no relacionamos nuestros datos con los datos de otras aplicaciones estaremos creando un silo de información, que no podrá ser contextualizado y por tanto tendrá una aplicabilidad y un valor menor.

Silo de información

Se denomina silo (o isla de información) a los conjuntos de datos aislados que no pueden relacionarse con otros conjuntos de datos.

Pensemos por ejemplo, en un conjunto de datos con información de colegios en nuestra ciudad que estamos analizando para ver a que colegio llevar a nuestros hijos. En ese caso, los datos proporcionados nos permiten conocer la oferta formativa y, como mucho, las características de cada colegio y el número de estudiantes que soportan. Integrando estos datos con la cartografía de la ciudad, nos permitiría hacer un ranking teniendo en cuenta la distancia de los colegios a nuestra vivienda. Añadiendo un conjunto de datos sobre la criminalidad, podríamos descartar colegios en zonas con un alto índice de criminalidad. Además, añadiendo un conjunto de datos sobre el tráfico en horario escolar en nuestra ciudad, nos permitiría estimar el tiempo que necesitamos para llevar a nuestros hijos al colegio, la disponibilidad de aparcamiento, saber si es mejor ir a pie o en coche, etc. Como podemos ver, cuantos mas conjuntos de datos integremos, más rica será la información que podremos sacar. La utilidad aquí estriba más en la interrelación de los datos que en los datos en sí.

En este documento utilizaremos los términos linked data y datos enlazados indistintamente.

En el ejemplo los cuatro conjuntos de datos son independientes, pero seria conveniente que pudieran definirse de manera que podamos enlazar fácilmente los datos entre sí. Se deberían enlazar los datos de la escuela con los datos de la cartografía de la ciudad, geoposicionando la ubicación de cada centro y de sus puertas de entrada, y los datos sobre el tráfico y criminalidad deberían también estar enlazados con la cartografía de la ciudad. Así podríamos ver los cuatro conjuntos de datos como un sólo conjunto de datos. Esta manera de conectar los datos entre sí es lo que pretende la propuesta de datos enlazados (o *Linked Data*).

Tim Berners-Lee

Científico británico en el área de la computación, conocido por ser el padre de la Web. Estableció la primera comunicación entre un cliente y un servidor usando el protocolo HTTP en noviembre de 1989. En octubre de 1994 fundó el Consorcio de la World Wide Web (W3C) para supervisar y estandarizar el desarrollo de las tecnologías sobre las que se fundamenta la Web y que permiten el funcionamiento de Internet.

Por tanto, el objetivo de los datos enlazados es unir datos de distintos orígenes para poder navegar por datos de distintos dominios y evitar así crear silos (o islas) de información. Hay otras formas de enlazar datos, pero en estos materiales nos centramos en la más popular a día de hoy: el *Linked data* propuesto por Tim Berners Lee.

Tal y como comenta el W3C, el *linked data* es la base sobre la que se sienta la web

de datos y la web semántica.

> Para hacer que la web semántica o denominada asimismo web de datos sea una realidad, es necesario disponer de un gran volumen de estos y tenerlos accesibles en un formato estándar y manejable. Además, las relaciones entre los datos también tienen que estar representadas. A toda esta colección de datos relacionados entre sí en la web se los denomina linked data. También visto como el corazón de la web semántica: la integración y razonamiento a gran escala sobre los datos en la web.
>
> *Linked data*, W3C

En este módulo veremos qué es el modelo de *linked data* y cómo saber hasta que punto los datos estan enlazados. También se presentarán los principios sobre los que se basan los datos enlazados, se apuntará brevemente como publicar datos enlazados y se mostrará un ejemplo de cómo convertir datos en abierto a datos enlazados.

3.1. ¿Qué es *Linked Data*?

Los *datos enlazados* presentan un método para interconectar datos de distintas fuentes de datos. El *linked data* Se basa en tecnologías Web estándar, tales como HTTP, RDF y los URI. Estas tecnologías (con excepción de RDF) se han utilizado en la web de documentos (la web primigenia - la que usamos diariamente -) para servir páginas web a lectores humanos. En este caso, se pretende utilizar estas tecnologías para que también los programas informáticos puedan acceder, enlazar e interpretar los datos. Esto permite que datos de distintas fuentes sean conectados, consultados y analizados.

Principios de diseño de *Linked Data*

Podeis encontrar más información sobre los principios de diseño en la siguiente dirección `https://www.w3.org/DesignIssues/LinkedData.html` para más información.

Los *datos enlazados* se basan en cuatro principios básicos enunciados por Tim Berners-Lee:

1) **Identificación:** Usar URIs para nombrar e identificar las *cosas* de la web (o las cosas a enlazar). Un identificador de recursos uniforme o URI (del inglés *Uniform Resource Identifier*) es una cadena de caracteres que identifica los recursos de una red de forma unívoca. La estructura básica de un URI es `esquema://máquina/directorio/archivo#fragmento`. Por tanto, según este principio, los elementos a compartir deberán tener una dirección web (URI) que los identifique. En el caso del país de Vietnam la URI que lo idenfitique podría ser por ejemplo `http://dbpedia.org/resource/Vietnam`. Como veremos más adelante, esta es la URI que identifica Vietnam en el conjunto de datos de la wikipedia.

2) **Consulta:** Aprovechar el HTTP de la URI para que la gente pueda localizar y saber más de la semántica del recurso que identifica. Por ejemplo, en el caso de Vietnam, la URI que hemos indicado antes podría ser desreferenciada para obtener más información de él (lo que denomina una descripción del mismo). Para hacerlo,

simplemente se accedería a la URI del recurso `http://dbpedia.org/resource/Vietnam`.

3) **Descripción:** Proporcionar información útil sobre la URI utilizando estandares web, como por ejemplo RDF y SPARQL. En el caso de Vietnam, por ejemplo, se podría incluir información sobre sus habitantes, sus provincias, su bandera, etc. Dicha información deberá ser representada mediante lenguaje RDF.

4) **Enlace:** Incluir enlaces a otras URIs para que se pueda navegar por los datos y descubrir información relacionada. En el caso de ejemplo, entre otros, se enlazaría el recurso `http://dbpedia.org/resource/Hanoi` para indicar que Hanoi es la capital de Vietnam y se utilizaría la URI `http://www.w3.org/2003/01/geo/wgs84_pos#Point` para representar los 4 puntos de latitud y longitud de las esquinas superior izquierda (16.166666666666668, 107.83333333333333) e inferior derecha (21.033333333333335, 105.85) que delimitan la caja que contiene el país. Añadiendo estos enlaces un programa sería capaz de identificar *Hanoi* como capital, obtener más información sobre *Hanoi* y saber la ubicación exacta de Vietnam en el mapa.

RDF

El marco de descripción de recursos o RDF (del inglés *Resource Description Framework*) es un método general para el modelado de información. Permite describir metadatos en sitios web, proveyendo interoperatibilidad entre las aplicaciones que intercambian información en lenguaje máquina por la web. Permite detallar información como las fechas de actualizaciones de páginas, palabras claves, derechos de autor, etc. SPARQL es el lenguaje de consulta utilizado para consultar datos en RDF.

3.2. El modelo de cinco estrellas de Tim Berners-Lee

Tim Berners-Lee sugirió un esquema de desarrollo de 5 estrellas para datos abiertos. Según esta clasificación, que presentamos en la Figura 7, los datos abiertos pueden convertirse en *datos enlazados* si se interrelacionan entre ellos. Los datos enlazados son la base técnica para crear la denominada web semántica, un estándar en el que cada dato contiene información asociada que lo relaciona automáticamente con otros.

Metadatos

Los metadatos (en inglés, *metadata*) son datos que describen otros datos. Por ejemplo, en una biblioteca se usan fichas que especifican autores, títulos, casas editoriales y lugares para buscar libros. Así, los metadatos permiten definir información de interés sobre los datos y ayudan a su gestión (ubicar datos, contextualizar datos, conocer su última actualización, etc.).

Podemos considerar que los datos abiertos, como los hemos presentado hasta ahora, llegan al nivel de 3 estrellas del esquema de Tim Berners-Lee. A partir del nivel 4 y 5, es necesario añadir la interconexión entre distintos conjuntos para dotarlos de significado.

Los datos enlazados no tienen porque tener licencia abierta. En caso de que los datos enlazados tengan licencia abierta hablaremos de *Open Linked Data*, en caso de que no sea así hablaremos simplemente de *Linked data.*

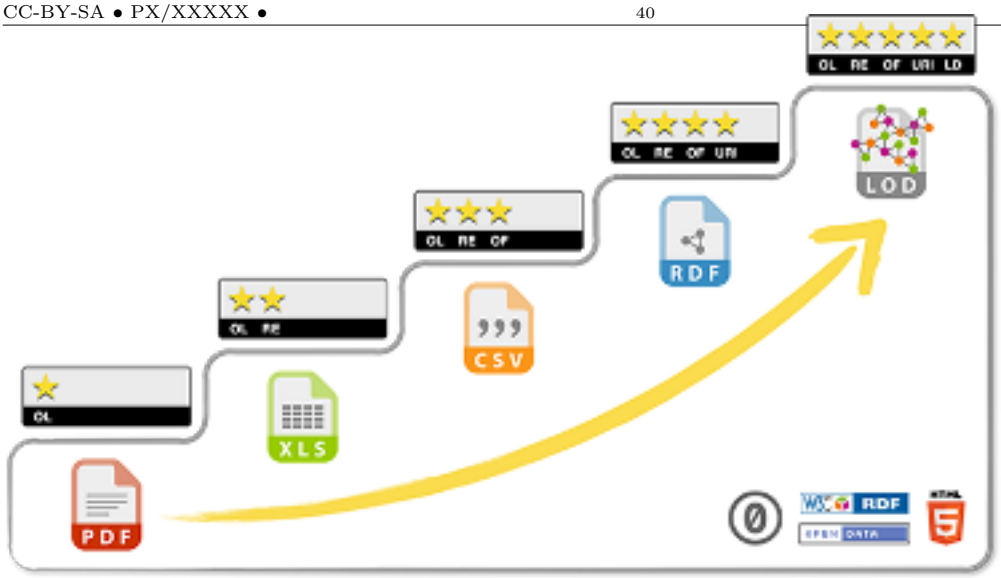


Figura 7. El modelo de cinco estrellas de Tim Berners-Lee. Fuente: `http://5stardata.info`

A continuación presentamos cada uno de los niveles del esquema de 5 estrellas y comentaremos los principales beneficios involucrados en cada caso.

Nivel ★

Publicar los datos en la Web (en cualquier formato).

Ventajas de utilizar datos en publicados en el nivel de 1 ★:

- Podemos verlos.
- Podemos imprimirlos.
- Podemos guardarlos localmente (en el disco duro o en una memoria USB).
- Podemos ingresar los datos en cualquier otro sistema.
- Podemos compartir los datos.

Los datos en esta categoría son datos que no pueden ser interpretados automáticamente por un programa informático. Por tanto, su reusabilidad es limitada.

Nivel ★★

Publicar los datos como datos estructurados (por ejemplo: Formato de Microsoft Excel en vez de una imagen de una tabla escaneada).

Como consumidor, las ventajas de utilizar datos publicados en el nivel de 2 ★ son las mismas que en el nivel anterior, más las siguientes:

- Podemos procesarlos directamente con software propietario para agregarlos, hacer cálculos, visualizarlos, etc.
- Podemos editarlos y/o refinarlos (siempre y cuando tengamos permiso para ello).
- Podemos exportarlos a otro formato (estructurado).

En esta categoría los datos son más reusables, ya que al estar estructurados son procesables por un programa informático.

Nivel ★★★

Utilizar formatos no propietarios (por ejemplo: CSV en vez de Microsoft Excel).

De forma similar, las ventajas de utilizar datos publicados en el nivel 3 ★ son las ventajas del nivel 2 ★, y además podemos añadir las siguientes:

- Podemos manipular los datos de cualquier forma que queramos, sin limitación de características o de uso de algún tipo de software en particular.

Es interesante notar que la persona o institución que desee publicar datos en abierto de nivel 3 ★ podría necesitar convertidores o *plug-ins* para exportar los datos desde formatos propietarios.

Nivel ★★★★

Usar estándares abiertos del W3C (URI, RDF y SPARQL) para identificar cosas, así la gente puede apuntar a éstas.

Como consumidor de datos abiertos, podemos hacer todo lo que hacemos en el nivel 3 ★ y además:

- Podemos enlazarlos desde cualquier otro sitio (Web o local).
- Podemos marcarlos como favoritos.
- Podemos reutilizar partes de los datos.

- Podríamos reutilizar herramientas y librerías disponibles, incluso si éstas sólo entienden parte de los patrones que utilizó quien los publicó.

- Podemos combinar nuestros datos con otros conjuntos de datos. Las URIs son un esquema global, por lo que si dos cosas tienen la misma URI, entonces representan el mismo concepto.

Por el contrario, es relevante destacar que, para un humano, entender la estructura de un documento de datos RDF puede requerir más esfuerzo que el necesario para entender datos estructurados tipo XLS o CSV.

Como persona o institución que desee publicar datos en abierto, debemos considerar los siguientes aspectos:

- Otros conjuntos de datos podrán enlazarse a nuestros datos.

- Deberemos asignar URIs a los datos y pensar en cómo representarlos.

Nivel ★★★★★

Enlazar nuestros datos a otros datos para proveer contexto.

Como consumidor de datos, podemos hacer todo lo que hacemos en el nivel 4 ★ y además:

- Podemos descubrir más datos (relacionados) mientras consumimos los datos.

- Podemos aprender directamente acerca del esquema de datos.

Este esquema es el que ofrece más reusabilidad y una mayor interpretabilidad de los datos para los programas informáticos. No obstante, al publicar datos en esta categoría habrá que tener en cuenta los problemas que puedan surgir por los enlaces rotos a los datos relacionados, como por ejemplo los errores a URIs no encontradas.

Como persona o institución que desee publicar datos según esta categoría, deberíamos considerar los siguientes aspectos:

- Podemos hacer que nuestros datos sean descubiertos a partir de sus relaciones.

- Podemos incrementar el valor de nuestros datos relacionandolos con datos de terceros.

- Terceras personas podran relacionar sus datos con los nuestros.

- Nuestros datos podrán ganar más relevancia y utilidad para la comunidad a

medida que vayan siendo referenciados (y por tanto relacionados) con datos de otros conjuntos de datos.

3.3. Beneficios de los datos enlazados

Un buen uso global de los datos enlazados permitiría transformar los muchos conjuntos de datos individuales que tenemos en la actualidad por un sólo conjunto de datos global y integrado que tiene la información de todas las fuentes de datos y donde los datos están relacionados entre sí con independencia del conjunto de datos al que pertenecían originalmente. Conceptualmente sería como pasar de tener N bases de datos individuales (una para cada conjunto de datos) a tener una sola base de datos distribuida con información de todas las bases de datos individuales integradas.

La integración de datos podría realizarse de distintas maneras. El uso de la filosofía de *Linked Data* que hemos presentado provee una forma genérica y flexible de publicar datos relacionados y facilita la búsqueda de datos, su consumo y su integración con datos de otras fuentes de datos. De hecho, el uso de *Linked Data* provee de las siguientes ventajas:

1) **RDF como modelo de datos unificado:** RDF ha estado especialmente diseñado para compartir datos de forma global y para minimizar los problemas integración de datos.

2) **Mecanismo de acceso a los datos unificado:** utilizar el protocolo HTTP permite que se accedan a los distintos conjuntos de datos utilizando las tecnologías y herramientas estándares que utilizamos diariamente, como por ejemplo un navegador web. Además, al accederse mediante HTTP, el método de acceso es independiente de la fuente de datos que queramos consultar. Eso no pasa cuando accedemos a APIs para recolectar datos, donde cada API tiene interfaces y formatos de datos distintos.

3) **Inferencia de nuevos datos navegando por sus enlaces:** Utilizar URIs para identificar los datos permite enlazar fácilmente datos de distintos origenes. Esos enlaces permiten ver los distintos repositorios de datos como un único repositorio enlazado, en el cual se pueden seguir los enlaces para inferir nuevas relaciones o información sobre los datos de interés.

4) **Datos auto-descriptivos:** Se puede consultar la descripción de cualquier dato simplemente con desreferenciar su URI. Esta descripción estará escrita usando el modelo de datos de RDF, permitiendo que los programas informáticos sean capaces de interpretar también su descripción y inferir nuevas relaciones a partir de la misma.

Previsión meteorológica de Barcelona

Día	Temp. Mín.	Temp. Máx.	Lluvia	Cielo
1/01/2016	7	16	No	Cubierto
2/01/2016	8	18	No	Soleado
3/01/2016	3	11	Si	Cubierto

Figura 8. Ejemplo de publicación de datos abiertos en formato PDF.

3.4. Publicación en el modelo de cinco estrellas

La publicación de datos según el modelo de cinco estrellas de *linked data* difiere en gran medida en función de si queremos enlazar los datos que tenemos en una página web o si queremos generar un conjunto de datos a partir de una base de datos propia. El caso de hacer que los datos que hay en una página web sean accesibles y enlazados se tratará más adelante en estos materiales, en concreto en la sección 4.5. El caso de publicar un conjunto de datos enlazados a partir de una base de datos queda fuera del propósito del curso y no se tratará en estos materiales.

A continuación veremos cómo, sobre un mismo ejemplo, podemos ir "subiendo peldaños" en el modelo de cinco estrellas que acabamos de presentar.

Lectura recomendada

El capítulo 5 del libro *Linked Data: Evolving the Web into a Global Data Space*, accesible desde `http://linkeddatabook.com/editions/1.0/#htoc61` trata con detalle la publicación de datos enlazados a partir de una base de datos.

Supongamos que el Ayuntamiento de Barcelona desea publicar en abierto datos sobre la previsión meteorológica de los próximos días en la ciudad.

Nivel ★

En un primer momento, el Ayuntamiento decide publicar en su sitio web un documento PDF con la previsión meteorológica de los próximos días bajo una licencia abierta, por ejemplo la Creative Commons Attribution 4.0 (CC-BY-4.0)[1]. El resultado puede verse en la Figura 8.

En este momento el Ayuntamiento ha conseguido la primera estrella del modelo de Tim Berners-Lee. Es decir, los datos son accesibles a través de Internet.

Nivel ★ ★

El principal problema que tenemos en este momento es la dificultad que tendrán las terceras partes en procesar esta información. Hemos publicado la información utilizando un formato no estructurado de datos, lo que dificulta su posterior tratamiento. Si el usuario que accede a la información sólo desea consultarla, no debe

[1] `https://creativecommons.org/licenses/by/4.0/`

	A	B	C	D	E
1	Día	Temp. Mín.	Temp. Máx.	Lluvia	Cielo
2	01/01/2016	7	16	No	Cubierto
3	02/01/2016	8	18	No	Soleado
4	03/01/2016	3	11	Si	Cubierto

Figura 9. Ejemplo de publicación de datos abiertos en formato Microsoft Excel (XLS).

haber ningún problema, pero en caso de que desee realizar algún tipo de transformación (por ejemplo, incluirla en otro documento o insertarla en una base de datos) u operación (por ejemplo, analizar o agregar la información) deberá copiar e introducir manualmente los datos en la aplicación donde desee utilizarlos. Si el conjunto de datos es grande, este proceso puede ser tedioso y consumir mucho esfuerzo.

Para conseguir que nuestros datos alcancen el nivel de la segunda estrella, deberemos publicar los datos abiertos en un formato estructurado. En este caso, podemos escoger el formato propietario de Microsoft Excel, conocido como XLS, en lugar de un documento PDF. La Figura 9 muestra el resultado de este cambio de formato.

En este punto los datos que hemos publicados pueden ser cargados, utilizando un programa propietario, y analizados o procesados. Por ejemplo, mediante Microsoft Excel los usuarios o terceras partes pueden calcular la temperatura mínima media en los próximos días o contar los días con posibilidad de lluvia.

Nivel ★ ★ ★

En el paso anterior hemos publicado los datos bajo una licencia abierta y en un formato estructurado. Aún así, es necesario que los usuarios o terceras partes que quieran cargar nuestros datos dispongan de un programa propietario. En caso contrario, no podrán cargar los datos publicados.

El objetivo de este tercer nivel del modelo de Tim Berners-Lee es evitar este problema, utilizando un formato estructurado y no propietario que permita que cualquier usuario o institución pueda cargar los datos sin tener que asumir el coste de la compra de un programa propietario.

En nuestro ejemplo, podemos publicar los datos en formato de fichero separado por comas (CSV) en lugar del formato propietario de Microsoft Excel y habremos conseguido que nuestros datos cumplan con la tercera estrella del modelo. Podemos ver el formato de los datos a continuación.

```
Dia,Temp. Min.,Temp. Max.,Lluvia,Cielo
1/1/16,7,16,No,Cubierto
2/1/16,8,18,No,Soleado
3/1/16,3,11,Si,Cubierto
```

En este nivel, nuestros datos pueden ser descargados por cualquier ciudadano o institución, cargados en un programa no propietario y analizados o procesados.

Nivel ★ ★ ★ ★

A partir de este nivel, hablaremos de datos abiertos y enlazados. El objetivo principal de este nivel es utilizar URIs para identificar los conceptos que aparecen en los datos abiertos.

En primer lugar, es importante notar que este conjunto de datos será publicado en formato RDF. No incluiremos el código completo del ejemplo, ya que es largo, complejo y escapa a los objetivos de este capítulo. Aún así, veremos algunas partes que nos pueden ayudar a entender el contexto semántico que estamos añadiendo en este nivel.

En este ejemplo, utilizaremos el vocabulario de Meteo[2] para dar contexto a ciertos conceptos que estamos utilizando en este conjunto de datos. La declaración del vocabulario se realiza en la línea 3 del código. Esto nos permite contextualizar ciertos conceptos, como por ejemplo el lugar de donde estamos dando la previsión (podemos ver la declaración *typeof="meteo:Place"* y *about="#Barcelona"* en la línea 8 que indica la localización de la previsión) o el tipo de datos que mostramos en cada parte (la línea 25 indica que utilizaremos los grados Celsius para la temperatura *property="meteo:celsius"*).

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/
       xhtml-rdfa-1.dtd">
3  <html xmlns:meteo="http://purl.org/ns/meteo#">
4  <head>
5    <title>Previsión meteorológica de Barcelona</title>
6  </head>
7  <body>
8    <div id="data" about="#Barcelona" typeof="meteo:Place">
9      <table border="1px">
10       <tr>
11         <th>Dia</th>
12         <th>Temp. Min.</th>
13         <th>Temp. Max.</th>
14         <th>Lluvia</th>
15         <th>Cielo</th>
16       </tr>
17       <tr rel="meteo:forecast" resource="#prevision01012016">
18         <td>
19           <div about="#prevision01012016">
20             <span property="meteo:predicted" datatype="xsd:dateTime">01/01/2016</span>
21           </div>
22         </td>
23         <td rel="meteo:temperature">
24           <div about="#temp01012016">
25             <span property="meteo:celsius" datatype="xsd:decimal">7</span>
26           </div>
27         </td>
28         ...
29       </tr>
30       ...
31     </table>
32   </div>
33 </body>
```

[2] http://inamidst.com/sw/ont/meteo

Además, asignaremos una URI distinta a la previsión de cada día de forma independiente. Así, datos externos podrán hacer referencia a la previsión que hemos realizado de un día concreto. Por ejemplo, en la línea 17 declaramos que haremos una previsión (*rel="meteo:forecast"*) y le asignamos un identificador (*resource="#prevision01012016"*), indicando que es la previsión para el día 1 de enero de 2016 (01-01-2016). Si un conjunto externo quiere referenciar esta previsión, deberá utilizar una URI de la forma siguiente:

http://www.ejemplo.com/datosenlazados/4star/#prevision01012016

Donde suponemos que publicaremos estos datos en el dominio *www.ejemplo.com* y en la ruta de acceso *datosenlazados/4star/*.

Aunque el fichero RDF presente una estructura jerárquica, debemos entenderlo como un conjunto de declaraciones o hechos simples. Por ejemplo, en el código HTML hemos declarado que "#Barcelona" es "un lugar" y que "#Barcelona" tiene un conjunto de previsiones, entre ellas "#prevision01012016". Veremos más sobre este tema en el siguiente capítulo.

Nivel ★ ★ ★ ★ ★

En el quinto nivel del modelo de Tim Berners-Lee, añadiremos enlaces a nuestros datos para darles contexto y enlazarlos con otros sitios de Internet.

DBpedia

DBpedia es un proyecto para la representación de datos de la *wikipedia* en un formato más semático, interpretable por programas informáticos. Puede accederse desde la URL `http://dbpedia.org`.

El siguiente código presenta unos datos similares al nivel anterior, donde hemos añadido enlaces a sitios externos que dan contexto a ciertos conceptos que en el nivel anterior no estaban del todo definidos. Veamos un par de ejemplos que nos pueden ayudar a comprender el concepto de "contextualización" del que hemos estado hablando. En el nivel anterior hemos definido que "Barcelona" es "un lugar", pero en este nivel añadimos (línea 9) una referencia a la página que trata sobre Barcelona en la DBPedia. Esta página contiene información estructurada sobre Barcelona, de forma que programa informático sea capaz de procesar esta información y enlazar a otros sitios que ayudarán a definir y contextualizar la información de Barcelona.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/
       xhtml-rdfa-1.dtd">
3  <html xmlns:meteo="http://purl.org/ns/meteo#">
4  <head>
5    <title>Previsión meteorológica de Barcelona</title>
6  </head>
7  <body>
8    <div id="data" about="#Barcelona" typeof="meteo:Place">
9      <span rel="owl:sameAs" resource="http://dbpedia.org/resource/Barcelona"></span>
10     <table border="1px">
11       <tr>
12         <th>Dia</th>
13         <th>
14           <a rel="rdfs:seeAlso" href="https://es.wikipedia.org/wiki/Temperatura" resource
                 ="http://dbpedia.org/resource/Temperature">Temp.</a> Min.
15           (<span rel="owl:sameAs" resource="http://dbpedia.org/resource/Celsius">ºC</span
                 >)
16         </th>
17         <th>Temp. Max.</th>
```

```
18        <th>Lluvia</th>
19        <th>Cielo</th>
20      </tr>
21      <tr rel="meteo:forecast" resource="#prevision01012016">
22        <td>
23          <div about="#prevision01012016">
24            <span property="meteo:predicted" datatype="xsd:dateTime">01/01/2016</span>
25          </div>
26        </td>
27        <td rel="meteo:temperature">
28          <div about="#temp01012016">
29            <span property="meteo:celsius" datatype="xsd:decimal">7</span>
30          </div>
31        </td>
32        ...
33      </tr>
34      ...
35    </table>
36  </div>
37 </body>
```

De forma similar, la línea 14 enlaza el concepto "Temperatura" con recurso *Temperatura* de *Wikipedia*[3] y *DBpedia*, para proporcionar contextualización del concepto para lectura humana y máquina, respectivamente. También hemos añadido información sobre la escala que estamos utilizando para medir la la temperatura, indicando que este dato se ha facilitado en grados Celsius (°C) y enlazando con su contexto de *DBpedia*.

3.5. Visualización de datos enlazados

Quizá una de las tareas más difíciles cuando se accede a los datos enlazados es identificar que enlaces se utilizan en cada conjunto de datos, tanto los que se utilizan para definir los propios datos, como los enlaces externos utilizados para contextualizarlos. Dicha información puede obtenerse consultando los datos mediante SPARQL o leyendo e interpretando el fichero RDF que define el conjunto de datos. Consultar

[3] `https://www.wikipedia.org`

Figura 10. Subtipos de la clase *Work* de la DBPedia en función de su numero de instáncias usando LOD Visualization.

Figura 11. Grafo de los elementos de la DBPedia relacionados con la URI que identifica la Universitat Oberta de Catalunya generado utilizando *LodLive*.

el fichero RDF no es una tarea cómoda y aún menos en nuestros primeros pasos en *linked data* y SPARQL puede ser difícil de utilizar si no sabemos los tipos de enlaces que usa un conjunto de datos. Por ello, puede ser interesante, y aún más en nuestros primeros pasos en *linked data*, utilizar programas que nos permitan visualizar los datos enlazados de una forma más cómoda y usable, adaptada para el consumo humano.

Una opción para visualizar las relaciones entre los datos de interes puede ser bajar su *dataset* y cargarlo en programas que permiten la visualización y análisis de grafos, como por ejemplo gephi[4].

No obstante, en los últimos años han aparecido algunas aplicaciones web para visualizar datos enlazados de forma gráfica y usable para los humanos. Quizá las aplicaciones más representativas son *LOD Visualization*[5] (ver figura 10), que permite navegar gráficamente por distintas fuentes de datos mostrando, por ejemplo, los diagramas de clases mediante mapas de calor, y *LodLive*[6] (ver figura 11), que permite visualizar mediante un grafo distintos datos representados en RDF.

Otras aplicaciones de interés

Quick and Dirty RDF browser permite visualizar documentos RDF desde la web de forma rápida y usable (accesible desde `http://graphite.ecs.soton.ac.uk/browser/`. Por otro lado, la web `http://www.visualdataweb.org/tools.php` contiene diversas herramientas para visualizar, consultar y exportar datos enlazados.

3.6. Ejemplos de *Linked Data*

Tal y como hemos comentado, los conjuntos de datos enlazados pueden tener licencia abierta o no. En caso de que tengan licencia abierta hablaremos de *Open Linked Data (LOD)* y en caso contrario de *Linked Data (LD)*. En la comunidad de linked data se denomina *LOD dataset* (o *LD Dataset*) a un conjunto de datos enlazados que tratan un dominio concreto. Estos conjuntos de datos normalmente comparten un prefijo RDF desde el cual cuelga el resto de recursos descritos. Por ejemplo, en el caso de DBpedia, los elementos RDF cuelgan de `http://dbpedia.org/resource/`.

El acceso a los *datasets* puede realizarse mediante distintos sistemas: *crawling* (que

[4] Para más información ver http://gephi.github.io
[5] Accesible desde `http://lodvisualization.appspot.com/`
[6] Accesible desde `http://en.lodlive.it/`

permite extraer contenidos estructurados escritos en RDF y sus datos relacionados), descarga (consiste en descargar - normalmente en formato comprimido - el fichero RDF que contiene el *dataset* a explorar), o puntos de acceso SPARQL (servicios web que permiten consultar los datos del *dataset* interactivamente mediante SPARQL).

La publicación de conjuntos de datos enlazados ha permitido crear un ecosistema de datos interrelacionados que ofrecen información de distintos dominios de forma unificada. En 2007 empezó a fraguarse la web de *OpenLinkedData*[7], como un espacio que mostraba los contenidos enlazados en la comunidad de *linked data* y que tenía como objetivo reducir las barreras al acceso a los datos de la web. El espacio web mostraba los distintos *datasets* publicados que utilizaban las tecnologías comentadas (URI, RDF y SPARQL) y sus interrelaciones en un diagrama denominado *la nube de linked data* (o *the linked data cloud* en inglés)[8]. En la figura 12 podemos ver la nube de datos enlazados a fecha de 2014[9].

Como se puede ver en la figura 12, en el centro de la nube de datos está un *dataset* llamado *DBpedia*, que representa los datos de la wikipedia en formato RDF. Debido a las características de sus datos (son genericos, no dependen de ningún dominio, se actualizan periódicamente, son multilingües, contienen gran cantidad de información, etc.) es un candidato perfecto para ser usado como *dataset* central que permita interrelacionar información de distintos dominios.

Esta nube de datos se suele denominar también *the open linked data cloud*. No obstante, el termino *open* no se utiliza para indicar que los datos de la nube sean abiertos, sino para indicar que los datos están accesibles desde la web. De hecho, según datos del último estudio, menos del 8 % de sus datos tienen información sobre su licencia[10]. Por tanto, para más del 90 % de los datos de la nube se desconoce si son abiertos o no.

Actualmente la nube de *linked data* integra más de 1000 *datasets*. En la tabla 2 podeis ver como se distribuyen estos *datasets* entre distintos dominios de aplicación.

A continuación, describiremos brevemente, a título de ejemplo, dos de los *datasets* más representativos de datos enlazados.

3.6.1. *DBpedia*

El objetivo de la DBpedia[11] es extraer datos estructurados de la Wikipedia de modo que estén disponibles en la web. Esto permite utilizar DBpedia para responder

[7] Accesible desde `http://linkeddata.org/home`

[8] Hay que tener en cuenta que en esta representación sólo tienen cabida los *datasets* que tengan más de 1000 tripletas RDF.

[9] Podemos encontrar más información en la siguiente URL: `http://lod-cloud.net/`

[10] Más información en `http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/`

[11] `http://dbpedia.org`

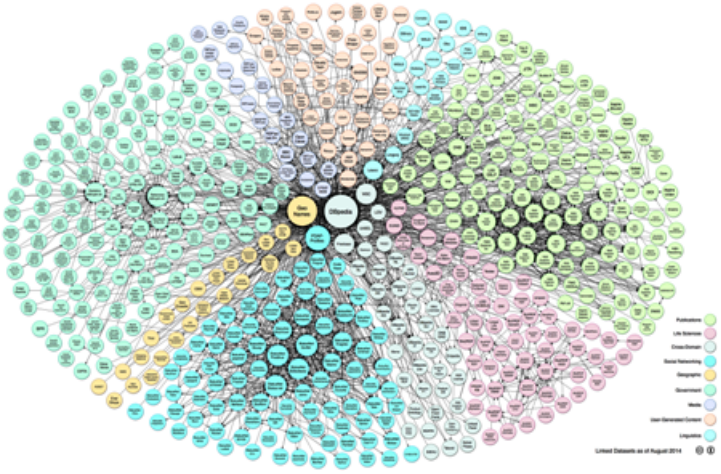


Figura 12. *The linked data cloud*: *Datasets* enlazados a fecha 2014. Las flechas entre dos *datasets* indican relaciones entre sus datos y el tamaño de la burbuja de un *dataset* es proporcional al número de datos que contiene. Fuente: Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. `http://lod-cloud.net/`

Tabla 2. Número de conjuntos de datos por tema. Fuente: `http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/`.

Gobierno	183	18.05 %
Publicaciones	96	9.47 %
Ciencias de la vida	83	8.19 %
Contenido generado por usuarios	48	4.73 %
Multi-dominio	41	4.04 %
Media	22	2.17 %
Geográfica	21	2.07 %
Redes sociales	520	51.28 %

consultas complejas y enlazar datos de la Wikipedia con otros conjuntos de datos.

Actualmente la DBpedia describe más de 3.500.000 de *cosas* distintas. Entre las que destacan más de 400 mil personas, más de 500 mil lugares, más de 100 mil álbumes de música, más de 160 mil organizaciones y más de 180 mil especies. Los datos son multilingües y ofrecen información en 97 idiomas distintos. Respecto a los enlaces que contiene, DBpedia enlaza con cerca de 3 millones de imágenes, con más de 6 millones de páginas web externas y más de 6 millones de recursos RDF externos (ubicados en otros *datasets*). Los datos se clasifican en más de 740 mil categorías distintas. Respecto al tamaño total, actualmente contiene más de 1000 millones de piezas de información RDF (tripletas RDF).

Los recursos de la DBpedia tienen la forma http://dbpedia.org/resource/*name*, donde *name* indica el nombre del recurso a describir. Se puede acceder a la página principal de la wikipedia que describe un recuso utilizando la URI http://wikipedia.org/wiki/*name*. DBpedia permite el consumo de los datos tanto por humanos como por programas.

Para ello, al consultar una URI, como por ejemplo http://dbpedia.org/resource/Vietnam, DBPedia nos retornará un resultado u otro en función de si este debe ser visualizado por un humano (se consulta desde un navegador web por ejemplo) o por un programa. La versión visible para un humano del recurso especificado sería `http://dbpedia.org/page/Vietnam`, mientras que la versión `http://dbpedia.org/data/Vietnam` devolvería directamente un fichero RDF con la descripción de Vietnam.

3.6.2. GeoNames

GeoNames[12] es una base de datos que contiene información geográfica sobre más de 10 millones de lugares de todo el mundo. Para cada lugar se describe su nombre (en distintos idiomas), su ubicación (latitud, longitud y elevación), su población, su codigo postal y su categoria. La categoría permite indicar el tipo de elemento que se esta describiendo (área, carretera, limite administrativo, etc.). La ubicación se representa mediante el vocabulario WGS84[13], que permite representar la geoposicion de los elementos definidos utilizando RDF.

Las URIs de *GeoNames* tienen la siguiente forma http://sws.geonames.org/*codigo*, donde código representa el código que identifica el elemento a describir. Por ejemplo, en el caso del pueblo de Arenys de Mar, podemos acceder a su URI `http://sws.geonames.org/6533985`. Geonames utiliza URIs distintas para distinguir los conceptos (Arenys de Mar por ejemplo `http://sws.geonames.org/6533985`) del documento RDF que describe sus características (en el caso de Arenys de Mar sería `http://sws.geonames.org/6533985/about.rdf`). En el segundo caso, se añade simplemente "/about.rdf" al final de la URI que identifica el objeto.

[12] Accesible desde `http://www.geonames.org/`

[13] La definición del vocabulario *WGS84* puede descargarse de la siguiente url `https://www.w3.org/2003/01/geo/`

4. Tecnologías de representación de datos en RDF

.

Tal y como se ha comentado en el capítulo anterior, *linked data* aboga por una política de datos enlazados usando el modelo de datos RDF para representar el significado de los datos y sus interrelaciones. En este capítulo presentaremos el modelo de datos RDF y mostraremos cómo usarlo, junto con URIs y HTTP, para representar datos y enlazarlos con datos externos. RDF no es un lenguaje, sinó un modelo de datos y como tal puede tener diversas representaciones. En el capítulo veremos las formas más habituales de representar datos en RDF. Continuaremos viendo como relacionar los datos que aparecen en las páginas web utilizando RDF. Finalmente, discutiremos sobre el concepto de web de datos y cómo el *linked data* pueden acercarnos a una web más semántica.

4.1. El modelo de datos RDF

Resource Description Framework (RDF) es un estándar del W3C para el intercambio de datos y la descripción de su semántica.

RDF no propociona un lenguaje de modelado sinó un modelo de datos basado en grafos que permite la integración de datos de distitas fuentes, incluso cuando estos tengan esquemas distintos. Aunque inicialmente RDF se concibio para representar metadatos de páginas web, acualmente es un paradigma de propósito general que se puede utilizar para añadir datos a cualquier tipo de información.

Especificación RDF

Podemos encontrar la especificación de RDF en `http://www.w3.org/TR/REC-rdf-syntax/`

En esta sección vamos a centrarnos en el modelo de datos de RDF sin entrar en la manera en que se representa. Basicamente por dos razones: 1) Es importante entender a fondo el modelo que propone RDF antes de entrar en detalles técnicos de representación (el modelo de datos es fácil de entender pero su representación puede ser confusa al principio), y 2)los datos en RDF se pueden representar utilizando distintos formatos.

El modelo RDF se basa en relacionar entidades, también denominadas recursos, por medio de relaciones binarias, también denominadas enunciados. Cada uno de estos enunciados es una tripleta formada por tres elementos:

1) un **sujeto**, la entidad origen de la relación;

2) un **predicado**, también denominado propiedad;

3) y un **objeto**, entidad de destino de la relación.

Enunciado y tripleta

Los términos enunciado y tripleta (triple) son intercambiables.

El predicado indica el tipo de relación que hay entre el sujeto y el objeto de la tripleta.

En muchas ocasiones las tripletas RDF se representan mediante un grafo dirigido como se muestra en la figura 13.

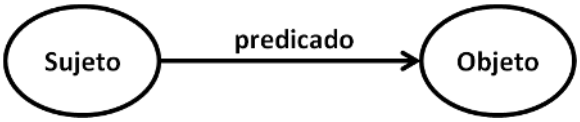


Figura 13. Tripleta RDF.

Por ejemplo, que Juan Valdez vive en Colombia se podría representar mediante la siguiente tripleta (ver figura 14).

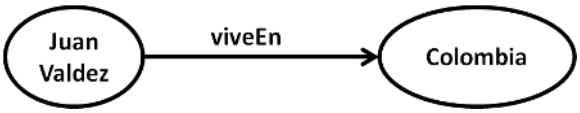


Figura 14. Tripleta RDF indicando que Juan Valdez vive en colómbia.

En las tripletas RDF los predicados pueden ser de dos tipos en función de si representan relaciones entre recursos (en este caso se denominan enlaces) o propiedades de un recurso (en este caso se denominan literales).

- En el caso de los enlaces, el nombre de la propiedad indica el tipo de relación entre los recursos. Por tanto, si dos recursos participan en un predicado con el mismo nombre, quiere decir que ambas tripletas indican el mismo tipo de información. Estos enlaces RDF son fundamentales ya que nos permiten enlazar datos de distintos origenes. En el ejemplo siguiente definimos dos tripetas con el mismo predicado *Vive En*. Eso indica que ambas tripletas definen la misma información, en este caso una relación que indica el lugar donde vive un ser vivo. En el ejemplo, como ambas tripletas tienen además el mismo objeto (*Salento*) podríamos decir que Juan y Conchita son vecinos.

```
Juan Valdez - Vive En -> Salento
Conchita  - Vive En -> Salento
```

- En el caso de los literales nos permiten definir propiedades del recurso sujeto. En este caso el recuso objeto no será una URI, sino un literal (una cadena de caracteres, un número o una fecha). Ejemplos de literales los tendríamos en las siguientes tripletas:

```
Juan Valdez - Nombre -> Juan
Juan Valdez - Apellido -> Valdez
Juan Valdez - edad -> 33
```

Lógicamente, no todo el conocimiento se puede describir con una única tripleta. Cuando estamos describiendo múltiples datos o información compleja puede ser necesario descomponer la información que queremos representar en un conjunto de tripletas. Por ejemplo, si quisiéramos describir la información completa de la dirección donde vive Juan Valdez (Calle Mayor número 5, 631020 Salento (Colombia)) podríamos hacerlo mediante las tripletas que se indican en la figura 15. Podemos ver que la representación en RDF ha requerido que creemos 5 tripletas. Una para indicar que Juan Valdez tiene una dirección. La dirección de Juan Valdez participa en tres tripletas más como sujeto para indicar su calle, su código postal y su población. Además, se ha añadido una tripleta para indicar que la población de Salento pertenece a Colombia.

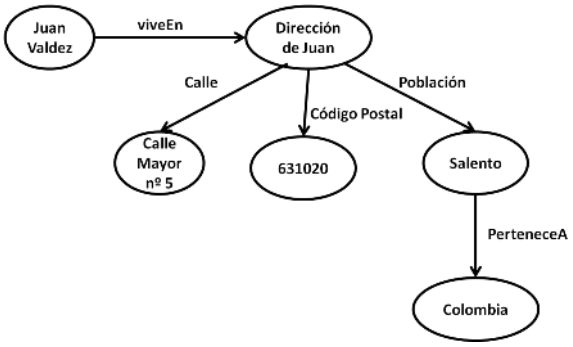


Figura 15. Representación de la dirección completa de Juan Valdez en Colombia usando tripletas RDF.

Los conjuntos de tripletas RDF se denominan **grafo RDF**, puesto que el documento RDF se puede ver como un grafo dirigido etiquetado donde las entidades son vértices del grafo, las tripletas representan los arcos del grafo y los predicados sus etiquetas.

Este es el funcionamiento básico de RDF. No obstante, cabe recordar que en *linked data* y en RDF los recursos se deben identificar mediante URIs. Por tanto, cada una de las etiquetas que hemos visto hasta ahora en las tripletas (para el objeto, el sujeto y el predicado) deberán ser substituidas por URIs que las identifiquen. Con la excepción de cuando se utilicen literales para representar las propiedades de un sujeto.

Al crear tripletas RDF deberemos reaprovechar URIs existentes siempre que sea posible. Eso nos permitirá enlazar datos con otras fuentes de datos, facilitará su acceso y la posibilidad de que nuestros datos sean encontrados y evitará que nuestros datos queden aislados. Teniendo en cuenta esto, podríamos reescribir el grafo RDF de la figura 15 aprovechando la URI de Juan Valdez en DbPedia (`http://dbpedia.org/resource/Juan_Valdez`), la URI de Salento en DBPedia (`http://dbpedia.`

`org/resource/Salento,_Quindío`) y la URI de Colombia en GeoNames (`http://www.geonames.org/3686110/`). Como no hemos encontrado ninguna URI que identifique la dirección de Juan, en este caso deberemos crear nuestra própia URI (`http://nuestroejemplo.org/DireccionDeJuan/`). En la figura 16 puede verse el nuevo grafo.

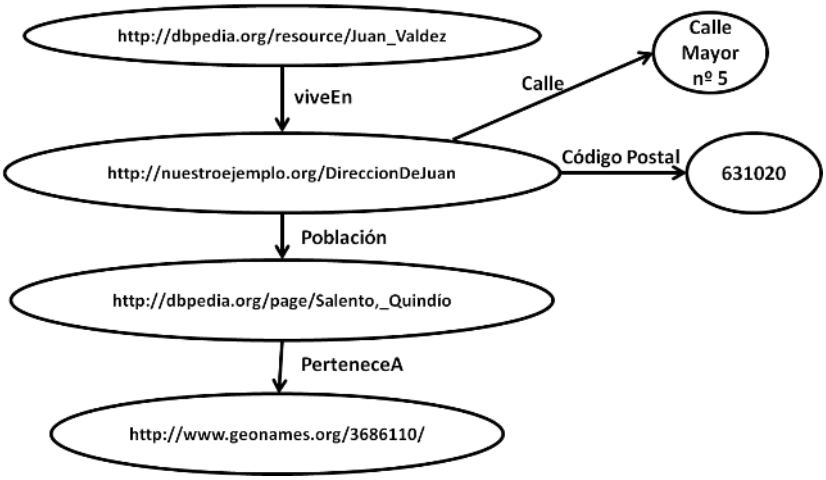


Figura 16. Representación de la dirección completa de Juan Valdez en Colombia usando tripletas RDF y recursos externos de DBpedia y GeoNames.

Normalmente, cuando distintos recursos RDF tienen el mismo prefijo (URL de base) en un grafo RDF, se utilizan abreviaciones para reducir la redundancia y simplificar la escritura y lectura de los grafos. Estas abreviaciones se definen como prefijos al principio del grafo de la siguiente forma: *<prefijo:fragmento_URI>*. Así, por ejemplo, para definir un prefijo para las URIs de la DBpedia de la figura 16 podriamos definir un prefijo llamado *dbp* para la URI *http://dbpedia.org/resource/*, de la siguiente forma: *dbp:http://dbpedia.org/resource/*. Una vez hecho esto, podríamos representar todas las URIs de la DBpedia indicando su nombre concatenado el prefijo *dbp*. Por ejemplo, podriamos definir la URI `http://dbpedia.org/resource/Salento,_Quindío` como *dbp:Salento,_Quindío*. En la figura 17 podemos ver como se ha reescrito el grafo de la figura 16 utilizando prefijos.

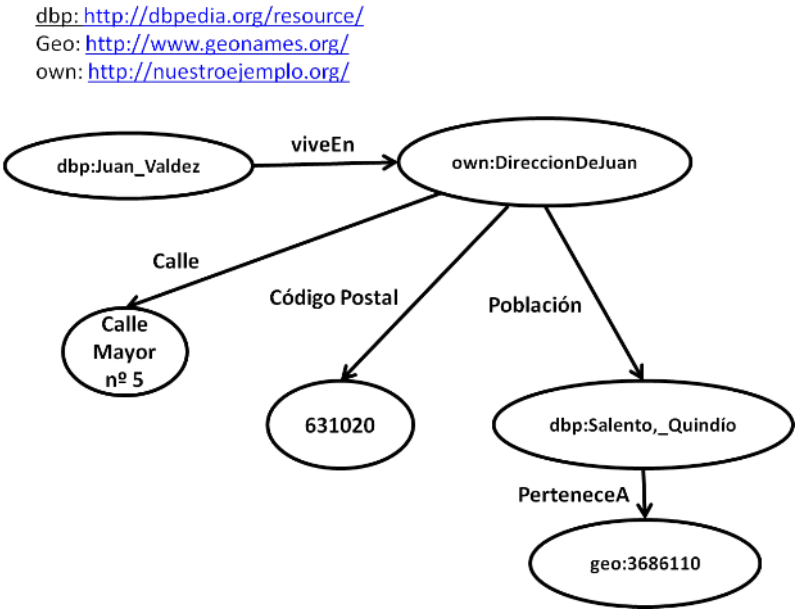


Figura 17. Representación de la dirección completa de Juan Valdez en Colombia usando prefijos.

En algunos casos puede ser que un recurso que queramos etiquetar tenga distintas URIs en función de la fuente de datos que tengamos en cuenta. Por ejemplo, Colombia como país tiene una URI en DBpedia (`http://dbpedia.org/resource/Colombia`) pero también en GeoNames (`http://dbpedia.org/page/Colombia`). Normalmente, eso puede suceder cuando al crear un recurso se desconoce la existencia de otros conjuntos de datos puedan tenerlo definido. Las diferentes URI que se refieren al mismo recuso se denominan **alias URI**.

Las alias URI no son especialmente problemáticas, siempre y cuando se indique que son equivalentes. Para indicar que dos URIs son equivalentes existe un predicado, denominado `http://www.w3.org/2002/07/owl#sameAs`. Por ejemplo, si consultamos la descripción de la URI de Colombia en la DBpedia podremos observar las URI equivalentes buscando aquellas relacionadas con el predicado *owl:sameAs* (fijaos que aquí *owl* es el prefijo de *http://www.w3.org/2002/07/owl#*. En la figura 18 podemos ver algunas de las URI equivalentes.

4.2. Tipos de recursos

Los recursos que nos podemos podemos encontrar en RDF son de dos tipos: recursos de información y *otros tipos de recursos*. Los recursos de información son aquellos recursos que se encuentran en la web: documentos, imágenes, vídeos, etc. Los otros tipos de recursos son conceptos u objetos del mundo real, como por ejemplo perso-

Figura 18. URIs equivalentes para el recurso de Colombia.

nas, perros, coches, lugares, etc. Estos objetos no existen en la web, aunque la web pueda contener información sobre ellos.

En *linked data* ambos recursos se identifican mediante URIs. No obstante, se diferencian en la manera en que se describen. Como habíamos comentado la descripción de un recurso RDF se puede obtener consultando la URI que lo identifica, proceso denominado desreferenciación. En este caso, cuando se desreferencie la URI de un recurso de información obtendremos su descripción junto con un código HTTP de respuesta de *200 - OK*, que indica que todo ha ido bien. En el caso de que se desreferencie otro tipo de recurso se suele obtener un la URI del recurso que describe el objeto junto con un código de respuesta de *303 - See other*. Para el objetivo de este curso no deberíamos preocuparnos en exceso sobre el tema, pero es importante conocerlo.

Lectura complementaria

Cool URIs for the Semantic Web accesible desde `https://www.w3.org/TR/cooluris/`

4.3. Vocabularios

Hasta ahora hemos visto como usar tripletas RDF para representar información y como utilizar URIs externas (tanto en el sujeto como en el objeto) para enlazar nuestros datos con datos de otras fuentes de datos. El hecho de reutilizar recursos

externos para representar cosas da un valor añadido a nuestros datos, permitiendo enlazar datos de distintos dominios. Ahora bién, ¿Que pasa con los predicados de nuestras tripletas? Las relaciones RDF son relaciones tipadas, lo que indica que su nombre (su URI) representa su tipo. Bajo esta premisa seria interesante pensar que distintos predicados que representen la misma información deberían tener el mismo nombre. En caso contrario será muy difícil, sino imposible, identificar cuando dos predicados representan información equivalente. Veamos algunos ejemplos:

```
http://dbpedia.org/resource/Juan_Valdez - http://nuestroejemplo.org/Nombre -> "Juan"
http://es.dbpedia.org/resource/Juan_el_Apóstol  - http://dbpedia.org/property/name -> "
     Juan"
```

En este caso tenemos dos tripletas que nos indican información sobre dos recursos: Juan Valdez y el apostol Juan. Nosotros como humanos podríamos inferir que dichas tripletas están indicando el nombre de estas personas, pero un programa informático difícilmente podrá llegar a esa conclusión, ya que las URI de los predicados son distintas y no hay ninguna tripleta que nos diga que son equivalentes. Por tanto, si preguntáramos qué recursos tienen como nombre "Juan", un programa informático sería capaz de inferir que ambos recursos (Juan Valdez y el apostol Juan) se llaman "Juan", ya que al tener nombres distintos, los dos predicados se interpretarían como cosas distintas.

Esta pequeña diferencia, escalada a nivel global, con miles de conjuntos de datos distintos creados por distintas personas, puede derivar en infinidad de URIs distintas para representar la misma información y eso puede ser inmanejable.

Para resolver estos problemas existen los vocabularios. Los vocabularios proveen colecciones de URIs que se pueden utilizar para representar la información de un determinado dominio. Los vocabularios pueden proveer URIs para definir tanto predicados como recursos.

Vocabularios más utilizados

Podéis encontrar una lista de los vocabularios más utilizados en los siguientes enlaces:

- `http://lov.okfn.org/dataset/lov/`
- `https://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies` .

Por ejemplo, existe un vocabulario ampliamente extendido llamado FOAF (acrónimo de *Friend of a Friend*). Este vocabulario provee predicados para describir personas y recursos para indicar algunos tipos de datos (Personas, Organizaciones, Projectos, etc.). Uno de los predicados que ofrece FOAF es *name* y se representa mediante la siguiente URI (`http://xmlns.com/foaf/0.1/name`). Si utilizaramos dicho vocabulario para representar los predicados obtendríamos un resultado como el siguiente:

```
http://dbpedia.org/resource/Juan_Valdez - http://xmlns.com/foaf/0.1/name - "Juan Valdez"

http://es.dbpedia.org/resource/Juan_el_Apóstol - http://xmlns.com/foaf/0.1/name - "Juan"
```

En este caso un programa informático sería capaz de identificar que las dos tripletas representan el mismo tipo de información y, si supiera que la URI http://xmlns.com/foaf/0.1/name indica el nombre de una persona, sería capaz de identificar que ambos recusos tienen como nombre *Juan*.

Por todo ello, es importante que, antes de definir los predicados de nuestras tripletas, indentifiquemos si hay vocabularios que permitan representar las relaciones entre los datos de forma más estándar. Hacerlo facilitará que la gente pueda navegar e inferir mayor información de nuestros datos.

Actualmente podemos encontrar una gran cantidad de vocabularios, algunos de los más usados son:

1) **Friend-of-a-Friend (FOAF)**[1]: Nos permite describir personas. Algunos de sus predicados más utilizados son *name, familyname, givenname*, que permiten indicar el nombre y apellidos de las personas, *knows*, que permite relacionar personas conocidas, *age*, que permite indicar la edad de las personas y *homepage y mbox*, que permiten indicar la página web y la dirección de correo electrónico de una persona.

2) **Dublin Core (DB)**[2]: nos permite definir atributos generales sobre los metadatos. Algunos de sus predicados más utilizados son *abstract* y *description*, que permiten definir una pequeña descripción de un recurso, *title* que permite definir el título de un recurso (de un libro por ejemplo), *creator* el creador/autor del recurso y *rights* que permiten informar sobre la licencia del recurso descrito.

3) **Semantically-Interlinked Online Communities (SIOC)**[3]: permite representar información sobre redes sociales y comunidades en línea. Algunos predicados de ejemplo son *name* para indicar el nombre de usuario de una persona y *member_of* para indicar que una persona es miembro de un grupo de usuarios o aplicación. Este vocabulario también ofrece URIs para indicar que un recurso es de un tipo determinado en el contexto de redes sociales (un foro, un post, un sitio web, etc.).

4) **RDF Schema (RDFS)**[4]: permite modelar datos en RDF, permitiendo indicar las clases de recursos que se pueden definir, los tipos de propiedades y las restricciones que deben cumplir. Quizá el predicado más utilizado en las busquedas de información es *type*, que indica que un recurso es de un tipo determinado (Juan Valdez es una persona por ejemplo).

5) **schema.org** [5]: vocabulario de amplio espectro que permite modelar datos de básicamente cualquier dominio. Fue creado de forma colaborativa y esponsorizado por Google, Microsoft, Yahoo y Yandex. En su creación se tuvo en cuenta muchos de los vocabularios más conocidos hasta el momento. Permite modelar no sólo los predicados sino también los tipos de los recursos sujeto y objeto. Representa entidades de distintos dominios, relaciones entre entidades y acciones. Actualmente ya hay más de 10 millones de sitios web que lo utilizan.

Teniendo en cuenta lo que hemos aprendido hasta ahora, podemos representar de nuevo el ejemplo de Juan Valdez teniendo en cuenta predicados de vocabularios externos (ver figura 19). En el ejemplo planteado se han utilizado los vocabularios

[1] Su especificación se encuentra en `http://xmlns.com/foaf/spec/`
[2] Su especificación se encuentra en `http://dublincore.org/documents/dcmi-terms/`
[3] Su especificación se encuentra en `http://rdfs.org/sioc/spec/`
[4] Accesible des de `https://www.w3.org/TR/rdf-schema/`
[5] Accesible desde `http://schema.org`

de la DBpedia para representar los predicados *residence* (para indicar la dirección de Juan Valdez), *address* (para indicar su calle), *postalCode* (para indicar el código postal del lugar) y *location* (para indicar la población). De *Geonames* se ha utilizado el predicado *locatedIn* para indicar que *Salento* está ubicada en *Colombia*.

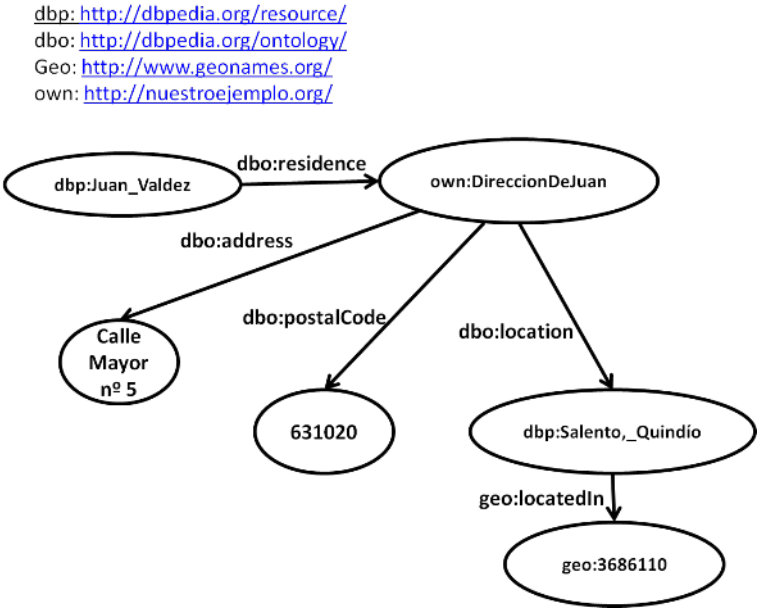


Figura 19. Representación de la dirección completa de Juan Valdez en Colombia usando vocabularios.

4.4. Serialización RDF

Como ya hemos visto, RDF es un modelo de datos y no un formato en sí mismo, de manera que está desvinculado de una sintaxis particular. El concepto serialización se refiere a la representación de datos en un formato para almacenarlos, transportarlos o cualquier otro motivo. En este contexto hablaremos de serializaciones en formato textual, puesto que el objetivo es el procesamiento automático por parte de aplicaciones y no hace falta que sean legibles para los seres humanos. También se podría plantear una serialización en formato binario para reducir su tamaño.

En este subapartado veremos dos sintaxis para serializar datos RDF: RDF/XML y Notation3. Hay que decir, sin embargo, que existen otros, como por ejemplo Turtle[6], que es un subconjunto de Notation3 o JSON-LD, que permite serializar RDF utilizando JSON [7].

[6] De Terse RDF Triple Language.
[7] La especificación de JSON-LD puede accederse desde `http://json-ld.org/`

4.4.1. RDF/XML

La forma más común de representar tripletas RDF es mediante lenguaje XML. XML es un metalenguaje extensible de etiquetas desarrollado por el W3C con el objetivo de repesentar información de forma fácil y de ser altamente reutilizable en el contexto de Internet.RDF provee una sintaxis sobre XML para describir la representación de las tripletas, conocida como RDF/XML.

XML

Para más información sobre XML, consultar `http://www.w3.org/TR/xml/`.

A continuación vamos a ver, paso a paso, como crear un documento RDF/XML.

RDF/XML

En `https://www.w3.org/TR/rdf-syntax-grammar/` se puede consultar la especificación completa de la serialización de RDF sobre XML.

Identificación del documento

En primer lugar hay que añadir un nodo raíz RDF. Cabe destacar que cada documento RDF podrá tener un solo nodo raiz. Para simplificar la escritura del documento, el nodo raiz indicará el espacio de nombres de RDF. Este espacio de nombres contiene información sobre las estructuras disponibles en el lenguaje RDF.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <!-- Body Code Omitted -->
</rdf:RDF>
```

Como se ha definido el espacio de nombres de rdf con la etiqueta “xmlns:rdf”, en el documento se podrá utilizar la cadena *rdf:Elemento* para hacer referencia a cualquier elemento del lenguaje RDF.

Añadir tripletas al documento

Un documento RDF puede contener más de un enunciado (tripleta). Cada enunciado se definirá sobre un sujeto, definido en la etiqueta *rdf:Description*. Podemos añadir tantos enunciados como haga falta sobre el mismo sujeto y por tanto dentro de una misma etiqueta *description*.

La etiqueta *rdf:Description* utiliza el atributo *rdf:about* para indicar la URI del sujeto. En el código siguiente se añade un enunciado sobre el sujeto que representa a Juan Valdez.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <rdf:Description rdf:about="http://dbpedia.org/resource/Juan_Valdez">
      <!-- Statement Code Omitted -->
   </rdf:Description>
</rdf:RDF>
```

Una vez identificado el sujeto de la tripleta, tenemos que indicar el predicado y el objeto del mismo.

Los predicados (o propiedades) se añaden como nuevas etiquetas que cuelgan de la etiqueta *rdf:Description* del sujeto. Para identificar la propiedad, se utilizará la

URI del predicado como nombre de la etiqueta. Para facilitar su lectura se utilizará la forma abreviada de XML por medio del espacio de nombres. Así, en el nodo raíz se indicará qué espacio de nombres se utilizará para abreviar la *URIref*.

Tal y como se ha comentado anteriormente, un predicado puede relacionar dos tipos de objetos: literales o enlaces. En caso de que el objeto sea un enlace, se indicará su URI mediante el atributo *rdf:resource* de la etiqueta de la propiedad. En caso de que el objeto sea un literal se indicará su valor como el valor de la etiqueta de la propiedad.

En el siguiente código podemos ver un ejemplo de representación RDF para un fragmento de los datos de Juan Valdez.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:foaf="http://xmlns.com/foaf/0.1/">
   <rdf:Description rdf:about="http://dbpedia.org/data/Juan_Valdez">
        <foaf:name> Juan Valdez </foaf:name>
        <foaf:homepage rdf:resource="http://www.juanvaldez.com/"/>
   </rdf:Description>
</rdf:RDF>
```

En el ejemplo creamos dos tripletas, una que indica el nombre de Juan Valdez y otra que indica su página web. Fijaos que hemos añadido el vocabulario FOAF en el espacio de nombres del documento porque los dos predicados a utilizar serán los predicados *name* y *homepage* de dicho vocabulario. Fijaos también que las dos tripletas cuelgan del mismo nodo XML ya que comparten el mismo sujeto. En la representación XML una etiqueta de descripción de recurso puede contener múltiples etiquetas de predicado, siempre y cuando sean sobre el sujeto indicado en la etiqueta. En el ejemplo se puede observar también como representar propiedades que enlazan literales (como el caso de la propiedad *name*) o enlaces (como en el caso de la propiedad *homepage*).

A continuación mostramos cómo quedaría la serialización del ejemplo de Juan Valdez completo. En este ejemplo hemos usado otro vocabulario diferente (vcard) para ejemplificar el uso de otros vocabularios.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:vcard="http://www.w3.org/2006/vcard/ns"
     xmlns:foaf="http://xmlns.com/foaf/0.1/">

   <rdf:Description rdf:about="http://dbpedia.org/data/Juan_Valdez">
      <foaf:name> Juan Valdez </foaf:name>
      <foaf:homepage rdf:resource="http://www.juanvaldez.com/"/>
      <vcard:adr>
         <vcard:street-address>Calle Mayor, 5</vcard:street-address>
         <vcard:postal-code>631020</vcard:postal-code>
         <vcard:locality> Salento</vcard:locality>
      </vcard:adr>
   </rdf:Description>
</rdf:RDF>
```

En el ejemplo se puede observar que todas las tripletas cuelgan de el recurso de *Juan Valdez*, definido en la etiqueta de tipo *description*. La primera tripleta permite definir el nombre de Juan Valdez utilizando el vocabulario *FOAF*. La segunda tripleta define el enlace de la página web de Juan Valdez. Posteriormente, se des-

cribe la dirección de Juan Valdez. Se puede observar que uno de los nodos que representa la propiedad de dirección vcard:adr se describe mediante un conjunto de nodos hijos. Es decir, en la serialización de unos datos RDF pueden anidarse nodos que dependen de otros nodos tantas veces como sea necesario.

4.4.2. Notation3

Notation3 (también conocido como N3)[8] es otra forma de serializar datos RDF que no utiliza XML. Este formato tiene la finalidad de facilitar la lectura humana y la compactación de tripletas, de modo que los documentos serializados tienen un aspecto muy diferente al que tendrían con RDF/XML.

La notación también permite trabajar con espacios de nombres, que hay que indicar al inicio del documento. Para declararlos, se utilizará la siguiente distribución en tres elementos (como si fuera una tripleta):

Hay otros formatos de serialización de RDF pero su explicación queda fuera del ámbito de este curso. Para más información consultad el capítulo 2 del libro Linked Data: Evolving the Web into a Global Data Space: http://linkeddatabook.com/editions/1.0/#htoc8.

```
@prefix namespace: <URI #> .
```

A diferencia del formato XML, en N3 todas las tripletas se especifican completamente. Por tanto, aunque distintas tripletas compartan el mismo sujeto, el sujeto estará definido para cada una de ellas explícitamente. Eso genera cierta redundancia en su representación, pero permite que los programas interpreten sus datos más fácilmente y facilita su compresión.

Los enunciados RDF (tripleta) se representan de la siguiente forma:

```
Sujeto Predicado Objeto.
```

A continuación mostramos cómo quedaría la serialización del ejemplo de Juan Valdez con esta notación:

```
@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vcard <http://www.w3.org/2006/vcard/ns#> .
@prefix foaf <http://xmlns.com/foaf/0.1/>.

<http://dbpedia.org/data/Juan_Valdez> <foaf:name> Juan Valdez .
<http://dbpedia.org/data/Juan_Valdez> <foaf:homepage> <http://www.juanvaldez.com/> .
<http://dbpedia.org/data/Juan_Valdez> <vcard:adr> <#adrJuan_Valdez> .
<#adrJuan_Valdez> <vcard:street-address> Calle Mayor, 5 .
<#adrJuan_Valdez> <vcard:postal-code> 631020 .
<#adrJuan_Valdez> <vcard:locality> Salento .
```

4.5. Enlazar datos desde páginas Web

Hasta ahora hemos presentado los principios de *linked data*, las tecnologías asociadas (RDF) y su representación (XML/RDF o Notation3). No obstante, no hemos hablado de donde se encuentran los datos enlazados: si en bases de datos (*datasets*) o incrustados en documentos web. La respuesta a esta pregunta es en ambos sitios.

[8] Más información en http://www.w3.org/TeamSubmission/n3/

Podemos definir información utilizando una filosofía de *linked data*, generando un gran fichero RDF/XML y publicándolo (junto quizá con un punto de acceso SPARQL) para que otros puedan acceder a sus datos. No obstante, si nos limitáramos sólo a hacerlo así estaríamos desaprovechando en gran medida las posibilidades que nos ofrece Internet y continuaríamos con la deriva de mantener una Web orientada a documentos.

De hecho, las páginas web están llenas de información válida que espera ser identificada como relevante, clasificada y enlazada. Esto requiere identificar de qué trata el documento web y qué fragmentos de información contiene. Una forma de hacerlo es enriqueciendo la semántica de los documentos web, identificando la información que contienen y sus interrelaciones, para que los buscadores, los *web crawlers* y los navegadores web puedan comprender su contenido y aprovecharlo para ofrecer nuevos servicios.

Podríamos utilizar lo que hemos aprendido para etiquetar parte de una página web con información RDF. Eso permitiría identificar la información relevante de la página y sus relaciones con otros datos en un formato comprensible para una máquina. Hay distintos formatos que permiten llevar a cabo este enriquecimiento semántico de páginas web y que son soportados por los grandes buscadores web: RDFa, Microdatos y Microformatos. En estos materiales describiremos brevemente como usar RDFa y Microdatos. Los microformatos son menos utilizados a día de hoy, ya que tienen más limitaciones en el modelado de cierto tipo de datos.

Tecnológicamente, estos formatos se componen de un conjunto de atributos de marcas que amplían la sintaxis de marcaje de documentos web. La finalidad de estos atributos es doble: por un lado se amplía el contenido visual de la web con datos legibles por máquinas y, por el otro, se facilita la conectividad entre los datos de la web. Así pues, estamos ante una de las formas de publicar *linked data*.

Lectura complementaria

En `http://manu.sporny.org/2011/uber-comparison-rdfa-md-uf/` se puede leer un artículo interesante donde se comparan los tres tipos de representación.

4.5.1. RDFa

Resource Description Framework in attributes (RDFa) es un mecanismo que permite definir tripletas RDF dentro de documentos HTML de forma integrada con la sintaxis de la página web. Este formato es quizá el más versatil, ya que al estar basado en XML, se puede aplicar a versiones antiguas de HTML y a otros formatos XML, como pro ejemplo XML y SVG.

La esencia de RDFa es utilizar un conjunto de atributos para añadir metadatos a los lenguajes de marcas que permitan incorporar información RDF. Para hacerlo, se marcará lo que se denomina entidad o ítem con un tipo de elemento RDF. Este tipo de elemento tomará el rol de sujeto en las tripletas RDF a representar. Posteriormente, se definirán sus propiedades, que equivaldrán a los predicados RDF, y los valores de las propiedades, que equivaldrán al objeto RDF.

Lectura complementaria

Para consultar toda la información relacionada con el uso y la definición del RDFa (herramientas, sintaxis, desarrollos, etc.), podéis consultar el siguiente enlace: `http://rdfa.info/`.

Los atributos básicos para el marcaje de datos RDF son: *vocab, typeof, property* y *resource*, que constituyen un subconjunto de los atributos que proporciona el *RDFa*. Estos atributos se utilizan de la siguiente forma:

Podeis consultar la especificación completa de RDFa en `http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014/`.

- *vocab*: sirve para indicar el vocabulario a utilizar para el etiquetaje RDF. Alternativamente a este atributo, también se puede utilizar la marca de espacio de nombres *xmlns* para indicar el vocabulario a emplear.

- *typeof*: Este atributo indica el tipo del sujeto que se declara, es decir, de qué se habla.

- *about*: este atributo se utiliza normalmente para indicar la URI del objeto a describir, es decir el ítem del cual se quiere describir información.

- *property*, *rel* o *rev*: Estos atributos permiten especificar las propiedades del sujeto. Es decir, podemos utilizarlas para definir el predicado de las tripletas RDF.

- *href*, *resource*, *src*: se utilizan normalmente para representar la URI de los recursos relacionados, es decir los objetos de las tripletas RDF. En caso de que el objeto sea un literal, su contenido estará dentro del contenido de la etiqueta.

A continuación podemos ver un ejemplo donde se usa RDFa para extender una página web con contenido en RDF. La página original es la siguiente:

```
<div>
   <strong>
      <span>Juan Valdez</span>
   </strong>
   <ul>
      <li>Teléfono:
         <span>548735489</span>
      </li>
      <li>Dirección:
         <span>C/ Mayor, 5</span>
         <span>631020</span>
     <span>Salento (Colombia)</span>
      </li>
   </ul>
   <a href="http://www.juanvaldez.com/">
      http://www.juanvaldez.com/
   </a>
</div>
```

La página se ha etiquetado utilizando RDFa y usando como vocabulario `http://schema.org`. En este caso se ha elegido un vocabulario distinto a los anteriores para mostrar otras alternativas de representación. En el ejemplo podeis ver como se define el sujeto como un recurso de tipo persona (*Person*). Posteriormente se indica su nombre (propiedad *name*), su página web (propiedad *url*) y su dirección. Su dirección podría haberse definido como una cadena de caracteres o como un recurso de tipo *PostalAdress*. En el ejemplo hemos optado por la segunda opción. Una vez creado el recurso de la dirección postal de Juan Valdez, se indica su calle (mediante la propiedad *streetAddress*), código postal (mediante la propiedad *postalCode*), localidad (mediante la propiedad *addressLocality*) y país (mediante la

propiedad *addressCountry*). Todas estas propiedades se cuelgan directamente del nodo donde se ha definido el recurso de la dirección de *Juan Valdez*, indicando que son propiedades de la dirección postal de Juan Valdez y no de la persona Juan Valdez.

```
<div vocab="http://schema.org/" typeof="Person">
   <strong>
      <span property="name">Juan Valdez</span>
   </strong>
   <ul>
      <li>Teléfono:
         <span property="telephone">548735489</span>
      </li>
      <li property="address" typeof="PostalAddress" vocab="http://schema.org/">
          Dirección:
          <span property="streetAddress">C/ Mayor, 5</span>
          <span property="postalCode">631020</span>
          <span property="addressLocality">Salento</span>
          <span property="addressCountry">Colombia</span>
      </li>
   </ul>
   <a property="url" href="http://www.juanvaldez.com/">
        http://www.juanvaldez.com/
     </a>
   </div>
</div>
```

4.5.2. Microdatos

Los microdatos (o *microdata* en inglés) permiten enriquecer documentos HTML con información semántica por medio del uso de los propios atributos de las etiquetas HTML. El marcaje consiste en agrupar conjuntos de propiedades (parejas nombre-valor) llamados ítems.

Los atributos utilizados por los microformatos son los siguientes:

Para conocer más sobre las especificaciones W3C de los microdatos, podéis consultar el siguiente enlace: `https://www.w3.org/TR/microdata/`

1) *itemscope*: atributo booleano que sirve para indicar que el elemento actual es un item sobre el que se representará información (el sujeto de las tripletas).

2) *itemtype*: indica el vocabulario a utilizar en el contexto del item.

3) *itemid*: permite definir un identificador único para un ítem.

4) *itemref*: permite referenciar a un ítem dentro del propio documento.

5) *itemprop*: permite indicar el atributo del ítem (el predicado de las tripletas).

A continuación, a modo de ejemplo, etiquetaremos el codigo anterior usando el formato de microdatos:

```
<div itemscope itemtype="http://schema.org/Person">
   <strong>
      <span itemprop="name">Juan Valdez</span>
   </strong>
   <ul>
      <li>Teléfono:
         <span itemprop="telephone">548735489</span>
      </li>
      <li property="address" itemscope itemtype="http://schema.org/PostalAddress">
           Dirección:
```

```
        <span itemprop="streetAddress">C/ Mayor, 5</span>
        <span itemprop="postalCode">631020</span>
        <span itemprop="addressLocality">Salento</span>
        <span itemprop="addressCountry">Colombia</span>
      </li>
    </ul>
    <a itemprop="url" href="http://www.juanvaldez.com/">
        http://www.juanvaldez.com/
    </a>
</div>
```

4.6. La Web de datos

Se dice que actualmente[9] hay siete zettabytes de información disponible, que este volumen se duplica cada dos años y que, en un solo día, se produce el doble de información de la que contenía Internet hace veinte años. Mucha de esta información se encuentra disponible en Internet, accesible a todo el mundo. No obstante, la web actual es una web de documentos pensada y implementada para que su contenido sea consumido por seres humanos.

La Web de los datos parte de una evolución de la Web original y el primer paso para crear una web semántica, en la que los programas informáticos sean también capaces de entender el contenido de la web. En este contexto los programas informáticos tendrían acceso a la totalidad de la información de la web y podrían encontrar respuestas para preguntas que actualmente son incapaces de responder. Para que esto sea posible hay que representar la información de la web en un formato que sea interpretable por programas informáticos. Eso nos llevaría a una web de datos, que podría ser percibida como una gran base de datos distribuida que contiene la información más relevante de la web textual.

En los últimos años se han hecho ciertos avances para que los programas informáticos tengan acceso a los datos. Uno de los más prolíficos ha sido el uso de APIs para acceder a los datos de un sitio web. Ejemplos de ello son las APIs de Twitter[10] o de Facebook[11]. Las APIs pueden ser útiles para analizar datos de un sitio web, pero no son la solución para analizar información de la web de un contexto más amplio, básicamente porque hay demasiadas y son muy heterogeneas. Por un lado, cada sitio web tiene su propia API totalmente distinta a las otras. Por otro lado, el número de APIs disponibles es enorme y crece rápidamente. Por ejemplo, en el sitio Web ProgramableWeb[12] podemos encontrar más de 16.000 API web para obtener datos de distintos sitios web. Evidentemente, el uso de APIs no parece el camino a la web de los datos.

Otra alternativa es el uso de *linked data*, que permite publicar una gran cantidad de datos en un formato procesable para las máquinas y que además puede interrelacionar datos de distintos dominios. Esta alternativa puede parecer más adecuada

Lectura complementaria

Algunos documentos interesantes sobre la web semántica:

- Diez años construyendo una web semántica de Marco Schorlemmer (`http://www.fgcsic.es/lychnos/es_es/articulos/construyendo_una_web_semantica`).
- La web semántica de Pablo Castells (`http://arantxa.ii.uam.es/~castells/publications/castells-uclm03.pdf`).
- Guía breve de la web semántica (`http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica`).

[9] A principios del 2017.
[10] Más información en `http://dev.twitter.com/overview/api`
[11] Más información en `https://developers.facebook.com/docs/graph-api`
[12] Ver `https://www.programmableweb.com/`

Figura 20. Ejemplo de información extraída de la web de datos por un buscador web.

para acercarnos a una web de datos. No obstante, hasta no hace mucho, la gran mayoría de repositorios de datos enlazados eran publicados por organizaciones y no por el colectivo de creadores de páginas web. Las tecnologías asociadas a los datos enlazados eran percibidas (y en cierto modo aún lo son) como algo complejo, sólo al alcance de los informáticos. Afortunadamente, la tendencia está cambiando, quizá debido a los últimos avances en los datos enlazados, a la aparición de aplicaciones más usables y automáticas para la publicación de datos enlazados, a la propuesta de *schema.org* y a las ventajas que ofrece etiquetar información mediante el mismo. El hecho es que cada vez más creadores de páginas web incrustan tripletas RDF en sus páginas.

Este incremento de datos enlazados en la web esta permitiendo que los buscadores web nos ofrezcan información más contexualizada y útil en algunas consultas. Sólo tenéis que buscar algún termino en cualquier buscador y ver que datos nos está ofreciendo. Por ejemplo, si buscamos podemos *café* en *Google*, el buscador nos ofrece, además de los típicos enlaces, información contextualizada sobre café que sólo es posible obtener cuando el buscador es capaz de interpretar la información que

se encuentra en las páginas web. La figura 20 muestra la caja de información que Google mostró al realizar la consulta de café. Como se puede ver la caja contiene información contextualizada y interactiva (podemos cambiar el *café* por *café expreso* i nos indicará las propiedades nutricionales de el café seleccionado). En teoría los datos obtenidos para crear esta caja provienen de datos etiquetados usando tecnologías de *linked data*.

Si esta tendencia se expande en un futuro y se democratiza, es probable que tengamos de forma natural y en poco tiempo una nueva web de datos. Una web que para los humanos se perciba como la actual, pero que tenga una capa de datos transparente a nuestra vista que permita que los programas informáticos accedan a su información para realizar tareas de análisis complejos y generar servicios valor añadido.

5. Consulta de datos en RDF (SPARQL)

.

SPARQL (pronunciado *esparkel*) es un lenguaje propuesto por el W3C para realizar consultas sobre datos en formato RDF. El lenguaje se basa en dos especificaciones: la de 2008 (SPARQL 1.0 [1]) y la del 2013 (SPARQL 1.1 [2]). Desde la versión 1.1 el lenguaje permite no solo realizar consultas sobre los datos, sinó también modificar o insertar nuevos datos RDF. En este documento nos basaremos en SPARQL 1.1 pero sólo abordaremos la consulta de datos.

El lenguaje de consulta sintácticamente es parecido a SQL, no obstante el modelo de datos subyacente (RDF) difiere en gran medida del modelo relacional. Eso hace, que aunque los dos lenguajes sean parecidos en sintaxis sean bastante distintos en su funcionamiento. El modelo de RDF es un modelo en grafo, por tanto las consultas realizadas en SPARQL serán consultas basadas en patrones de grafo (más adelante veremos lo qué son).

En este documento haremos una breve introducción de los aspectos más relevantes de la sintaxis de SPARQL y presentaremos algunos ejemplos de consultas básicas sobre este lenguaje. Cabe destacar que los materiales son introductorios y pretenden dar las bases principales a complementar a través de los manuales y especificaciones de SPARQL que podemos encontrar en la red.

5.1. Puntos de acceso SPARQL

Para poder ejecutar consultas SPARQL necesitamos un punto de acceso (o *end-point* en inglés). Estos puntos de acceso serían el equivalente al sistema gestor de bases de datos en un modelo relacional. Podemos encontrar distintos puntos de acceso accesibles desde Internet que nos permiten consultar conjuntos de datos RDF localizados. Una opción será utilizar estos puntos de acceso. Otra opción sería descargarse en local el conjunto de datos de interés y realizar las consultas localmente. Existen distintas herramientas, como por ejemplo Protege[3], que pueden actuar de punto de acceso SPARQL local.

Desde la versión 1.1, los puntos de acceso sparql permiten obtener los resultados de una consulta en distintos formatos (XML, JSON, CSV, etc.) o crear un nuevo grafo RDF como resultado de una consulta. En SPARQL también se pueden consultar

[1] SPARQL 1.0 Query Language for RDF: `https://www.w3.org/TR/rdf-sparql-query/`
[2] SPARQL 1.0 Query Language for RDF: `https://www.w3.org/TR/sparql11-query/`
[3] Protegé es un editor de ontologías y permite editar datos en RDF y consultarlos mediante SPARQL. Puede obtenerse desde `http://protege.stanford.edu/`

datos RDF de más de un *dataset* mediante lo que se denominan *federated query*. No obstante, esta funcionalidad no se explicará en estos materiales.

Algunos de los puntos de acceso más populares son los siguientes:

- Para consultar repositorios de datos en RDF:

  – Datahub/CKAN (`http://semantic.ckan.net/sparql`) permite consultar la información sobre los conjuntos de datos existentes en la web datahub.io.

  – Linked Open Data Cloud (`http://lod.openlinksw.com/sparql`) permite consultar la información sobre los conjuntos de datos existentes en la nube de *open linked data*.

- Para consultar datos de carácter general:

  – DBpedia (`http://dbpedia.org/sparql`) permite acceder a los datos de la DBpedia.

  – Wikidata (`https://query.wikidata.org/`) permite acceder a los datos de `www.wikidata.org`.

- Para consultar datos geográficos:

  – Geonames(`http://geosparql.org/`) permite acceder a los datos geograficos disponibles en Geonames.

  – LinkedGeoData (`http://linkedgeodata.org/sparql`) permite acceder a los datos geograficos de la web de OpenStreetMaps (`https://www.openstreetmap.org/`).

- Para consultar otros tipos de datos:

  – LinkedCommerce (`http://linkedopencommerce.com/sparql/`) permite acceder a datos de productos, servicios y ofertas comerciales.

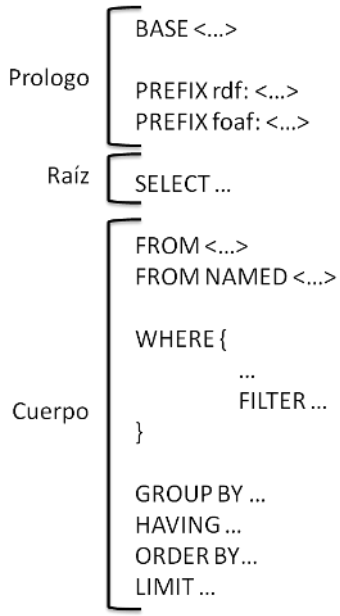
  – *Gene Ontology* (`http://sparql.bioontology.org/`) permite acceder a datos genómicos.

Podeis encontrar en `https://www.w3.org/wiki/SparqlEndpoints` una lista de los puntos de acceso disponibles más relevantes. En `http://labs.mondeca.com/sparqlEndpointsStatus.html` podeis encontrar información sobre el estado de los puntos de acceso SPARQL públicos.

5.2. Sintaxis básica de las consultas SPARQL

SPARQL utiliza notation3 para definir las tripletas que aparezcan en la consulta, indicando las URIs entre los caracteres <> y permitiendo la definición de prefijos para abreviar la definición de URIs.

La sintaxis básica de una consulta SPARQL es la siguiente:



Toda consulta SPARQL está formada por una raíz, un cuerpo y opcionalmente un prólogo. A continuación vamos a entrar en más detalle en cada una de estas partes.

5.2.1. Prólogo

El prologo permite definir los espacios de nombres, es decir las abreviaciones, de los vocabularios que queramos utilizar en la consulta. Se pueden añadir tantos espacios de nombres como sea necesario mediante prefijos. Para hacerlo deberemos utilizar la palabra reservada *PREFIX*.

Por ejemplo, podríamos definir prefijos para los vocabularios de *foaf* y de *skos* de la siguiente forma:

```
dc: http://purl.org/dc/elements/1.1/
foaf: http://xmlns.com/foaf/0.1/
```

Una vez definidos estos vocabularios, podríamos referirnos a las propiedades *knows* del vocabulario *friend of a friend* y *tittle* del vocabulario de *dublin core* de la siguiente forma:

```
foaf:knows
```

```
dc:title
```

El termino *BASE* se puede utilizar una sola vez en una consulta SPARQL y permite definir el vocabulario por defecto de la consulta. Por tanto, si hay definido un vocabulario mediante el término *BASE*, se entenderá que todas las URIs relativas que no utilicen ningún espacio de nombres pertenecerán al vocabulario base.

Prefijos SPARQL

El buscador ubicado en `http://prefix.cc/` permite encontrar las URI de los vocabularios más extendidos.

5.2.2. Raíz

La raíz permite indicar la operación a ejecutar (*SELECT*) y el resultado esperado de la consulta. Es decir, qué valores debe devolver la consulta SPARQL. La raíz es un elemento obligatorio en cualquier consulta SPARQL.

Al igual que en SQL podemos poner un * en la raíz para indicar que queremos que la consulta nos devuelva todas los datos de interés. También podemos indicar una lista de expresiones para especificar los datos a obtener. Al igual que en SQL se podrá usar la cláusula *DISTINCT* para indicar que no se deben devolver valores duplicados.

En el ejemplo siguiente se obtendrían todos los valores que satisfagan la consulta 1. Para la consulta 2 se obtendrían sólo los datos de las variables *nombre* y *edad*. Más adelante explicaremos el concepto de variable en SPARQL y veremos en detalle qué significan y como se obtienen sus valores.

```
//Consulta 1
SELECT * ...

// Consulta 2
SELECT ?nombre, ?edad ...
```

5.2.3. Cuerpo

El cuerpo de la consulta permitirá determinar qué elementos del grafo deben ser seleccionados y en qué formato (orden y agrupación) deberán ser retornados. Para ello el cuerpo de la consulta está dividido en 3 partes:

- *Origen de la consulta*: permite definir el conjunto de datos que se deben consultar. Hace referencia al grafo RDF (o al fragmento del mismo) que debe ser tenido en cuenta como origen de datos. Esta cláusula es opcional. Para indicar el conjunto de datos a utilizar se utilizan las palabras clave *FROM* y *FROM NAMED*.

  Dado que cada punto de acceso SPARQL suele tener un conjunto de datos de referencia, la cláusula *FROM* suele estar vacía cuando trabajamos con ellos. Este será el caso general en el tipo de consultas con las que trabajaremos durante el curso.

- *Patrón de consulta*: indica las características que deben cumplir los datos del grafo para poder ser seleccionados por la consulta. Permitirá seleccionar un conjunto de datos que satisfacen la estructura y los valores indicados en el patrón. El patrón estará contenido dentro de la cláusula *WHERE*. Esta parte podrá ser tan compleja como sea necesario, permitiendo conjunciones de patrones, disyunciones, partes opcionales variables y restricciones de valores.

- *Modificadores*: son operaciones que se ejecutarán sobre los datos seleccionados, ya sea para cambiar su orden (la cláusula *ORDER BY* permite ordenar los datos de acuerdo a un conjunto de propiedades), su nivel de agregación (la cláusula *GROUP BY* permite agrupar los datos de acuerdo a una o más propiedades) o limitar su número (la cláusula *LIMIT X* permite limitar el número de valores devueltos a los X primeros, y la cláusula *OFSET Y* permite empezar a mostrar los datos a partir del elemento Y-esimo devuelto por la consulta).

5.3. Patrones de consultas

Los patrones de consulta son los elementos clave para entender el funcionamiento de las consultas SPARQL. Un patrón de consulta es una condición que deben satisfacen los datos del grafo RDF para ser seleccionados por la consulta. Existen distintos tipos de patrones en SPARQL, uno de los más utilizados es el patrón de tripleta.

Un patrón de tripleta es una tripleta RDF (por tanto compuesto por un objeto, un sujeto y un predicado) en la que uno o más de sus componentes son una variable.

En las consultas SPARQL las variables se representan mediante un símbolo de interrogación (o una almohadilla) y el nombre de la variable. Ejemplos de variables serian *?nombre* y *?email*.

Suponed que estamos realizando una consulta en el punto de acceso SPARQL de la dbpedia (y por tanto consultando el conjunto de datos de la Wikipedia) y que utilizamos el siguiente patrón de tripleta en el cuerpo de una consulta SPARQL:

```
<http://dbpedia.org/resource/Juan_Valdez>  <http://www.w3.org/2000/01/rdf-schema#label>
     ?nombre
```

La consulta nos retornaría todas aquellas tripletas RDF de la DBpedia (ya que es el grafo RDF que se está consultando) que conforman el patrón. Una tripleta conformará (o hará *matching* con) un patrón si cambiando el valor de las variables por los valores de la tripleta que están en su lugar conseguimos que el patrón sea cierto (es decir, tiene una ocurrencia en la los datos consultados). En el caso que nos ocupa, el patrón prefija el sujeto (recurso de *Juan Valdez*) y el predicado (la propiedad *label* de *rdfs* que indica la etiqueta o nombre del recurso). Por tanto, todas las tripletas con sujeto *Juan Valdez* y predicado *rdfs:label* satisfarían el patrón planteado. Para comprobarlo os animamos a que ejecutéis la siguiente consulta SPARQL que utiliza el patrón presentado.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?nombre
WHERE {
  dbr:Juan_Valdez rdfs:label ?nombre
}
```

El resultado obtenido será el siguiente:

```
nombre
-----------------
"Juan Valdez"@es
"Juan Valdez"@it
"????????"@ja
"Juan Valdez"@fr
"Juan Valdez"@en
```

El resultado es el nombre del recurso Juan Valdez en distintos idiomas. Más adelante hablaremos del significado de las cadenas *@es*, *@it*, *@ja*, *@fr* y *@en*. Fijaos que hemos simplificado el patrón de consulta utilizando un prefijo para definir una abreviación de la URI del vocabulario de rdfs y otro para definir el prefijo de donde se encuentran los recursos en la DBpedia.

Podéis ver también que la clausula select contiene la variable *nombre*. La consulta planteada buscará tripletas del grafo que concuerden con el patrón planteado. Para ello se buscaran todas las tripletas que tienen *http://dbpedia.org/resource/Juan_Valdez* como sujeto y *http://www.w3.org/2000/01/rdf-schema#label* como predicado. Hay 5 tripletas en el conjunto de datos consultado que cumplen esta condición:

```
<http://dbpedia.org/resource/Juan_Valdez> rdfs:label ?nombre "Juan Valdez"@es
<http://dbpedia.org/resource/Juan_Valdez> rdfs:label ?nombre "Juan Valdez"@it
<http://dbpedia.org/resource/Juan_Valdez> rdfs:label ?nombre "????????"@ja
<http://dbpedia.org/resource/Juan_Valdez> rdfs:label ?nombre "Juan Valdez"@fr
<http://dbpedia.org/resource/Juan_Valdez> rdfs:label ?nombre "Juan Valdez"@en
```

Para cada una de ellas, se sustituirán las variables del patrón por el valor asociado de la tripleta. Por tanto, tendremos tantas ocurrencias de *nombre*, como tripletas en el grafo satisfagan la condición, en este caso el nombre de Juan Valdez en 5 idiomas distintos. En caso de que esta sustitución satisfaga todas las condiciones indicadas en la consulta, diremos que esta será una solución válida para dicha consulta y la tripleta será seleccionada. En la consulta de ejemplo, añadiendo la variable *nombre* a la cláusula *SELECT* estamos pidiendo que se devuelvan los posibles valores de la variable nombre.

Como hemos comentado anteriormente, en la cláusula nombre podemos poner una expresión, como por ejemplo *count(*)* para saber cuantos valores de la DBpedia satisfacen el patrón especificado. A continuación podemos ver la consulta modificada utilizando *count(*)*:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT count(*)
WHERE {
    dbr:Juan_Valdez rdfs:label ?nombre
}
```

En el caso anterior hemos visto el patrón de tripleta más simple posible. Podríamos tener patrones de tripleta con más de una variable, como por ejemplo el siguiente:

```
<http://dbpedia.org/resource/Juan_Valdez>  ?predicado ?objeto
```

En este caso estamos definiendo un patrón de dos variables, todas las tripletas con sujeto *http://dbpedia.org/resource/Juan_Valdez* serían soluciones válidas de este patrón de consulta, como podemos comprobar ejecutando la siguiente consulta en el punto de acceso:

```
SELECT *
WHERE {
    <http://dbpedia.org/resource/Juan_Valdez>  ?predicado ?objeto.
}
```

La consulta anterior nos permitirá identificar todos aquellos recursos relacionados con el recurso de Juan Valdez y los predicados que los relacionan.

En una consulta SPARQL pueden definirse patrones simples (de una sola tripleta) o conjuntos de patrones (de más de una tripleta). Un conjunto de patrones se compone de distintos patrones simples concatenados mediante un punto. Por tanto, el siguiente conjunto de patrones:

```
?university <http://dbpedia.org/ontology/city> <http://dbpedia.org/resource/Cambridge,
     _Massachusetts>.
?university rdf:type <http://dbpedia.org/ontology/University>
```

Nos permitiría seleccionar aquel subgrafo RDF que identifica las universidades ubicadas en la ciudad de Cambridge. Para entender mejor como funciona un conjunto de patrones podemos analizar los patrones de tripleta por separado para conocer los valores que satisfacen cada patrón simple y luego realizar una Y lógica para identificar los datos que satisfacen ambos patrones. En este caso:

- El primer patrón identificará aquellos recursos ubicados (relacionados mediante la propiedad *http://dbpedia.org/ontology/city*) en la ciudad de Cambridge (representada por la URI *http://dbpedia.org/resource/Cambridge,_Massachusetts*). Por tanto, la solución a este patrón incluiría cualquier tipo de recurso: gasolineras de Cambridge, tiendas de golosinas de Cambridge, etc. Estos recursos se almacenaran en la variable *university*.

- El segundo patrón identificará las universidades definidas en la DBpedia, para ello identificamos aquellos recursos que son de tipo (*rdf:type*) universidad (*http://dbpedia.org/ontology/University*). Esta lista contendrá universidades de todo el mundo. La lista de universidades se debería guardar en la variable *university*. No obstante, como esta variable ya tiene identificados un conjunto de valores posibles (los recursos de Cambridge), las únicas universidades que garantizarán el grupo de patrón serán aquellas ubicadas en Cambridge. Por tanto, al final de la consulta en la variable *university* habría las URIs de la universidades ubicadas en Cambridge.

Podemos ver los resultados del conjunto de patrones ejecutando la consulta siguiente sobre el punto de acceso SPARQL:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT *
WHERE {
    ?university dbo:city <http://dbpedia.org/resource/Cambridge,_Massachusetts>.
    ?university rdf:type dbo:University
}
```

Podríamos complicar el patrón añadiendo más patrones de tripletas al mismo. Por ejemplo, en la siguiente consulta añadimos una nueva restricción para indicar que sólo estamos interesados en universidades privadas.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
SELECT *
WHERE {
  ?university dbo:city <http://dbpedia.org/resource/Cambridge,_Massachusetts>.
  ?university rdf:type dbo:University.
  ?university dbp:type <http://dbpedia.org/resource/Private_school>
}
```

Cuando varios patrones de tripleta consecutivos comparten el mismo sujeto, podemos simplificar su escritura escribiendo su sujeto una sola vez. Para ello, deberemos poner un ';' en vez de un '.' al final de la tripleta para indicar a SPARQL que la tripleta siguiente comparte el mismo sujeto. Teniendo en cuenta esta simplificación podríamos reescribir la consulta anterior de la siguiente forma:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
SELECT *
WHERE {
    ?university dbo:city <http://dbpedia.org/resource/Cambridge,_Massachusetts>;
                rdf:type dbo:University;
                dbp:type <http://dbpedia.org/resource/Private_school>
}
```

5.4. Otros tipos de consultas SPARQL

Hasta ahora hemos visto como realizar consultas simples que devuelven los datos de interés de un *dataset* RDF a partir de un conjunto de variables asociadas a patrones de consulta. Tal y como hemos visto, este tipo de operación se realiza utilizando la cláusula *SELECT* en la raíz de la consulta. No obstante, hay otros tipos de consulta que pueden realizarse mediante SPARQL.

En particular existen 4 tipos de consultas que se pueden realizar. Una de ellas es la que hemos visto hasta ahora (*SELECT*). Los otros tipos de consultas que pueden realizarse son:

- *CONSTRUCT*: Permite crear un nuevo grafo a partir de los resultados de una consulta SPARQL. Un ejemplo clásico de uso de la cláusula *CONSTRUCT* es

la creación de un grafo que contenga información sobre abuelos y nietos a partir de un conjunto de datos que tenga sólo información sobre padres e hijos.

Suponiendo que tenemos un conjunto de datos RDF con información sobre padres y hijos, la siguiente consulta permitiría obtener un nuevo grafo con información de abuelos y nietos:

```
CONSTRUCT {
  ?nieto  :esNietoDe      ?abuelo .
  ?abuelo :esAbueloDe     ?nieto
}
WHERE {
  ?nieto  :esDescendienteDe ?padre.
  ?padre  :esDescendienteDe ?abuelo.
  ?abuelo foaf:gender     :male .
}
```

- *ASK*: Permite ejecutar una consulta para comprobar si una determinada condición es satisfecha en el conjunto de datos consultados. *ASK* devuelve un valor booleano indicando si el patrón de consulta se satisface o no. No devuelve los datos del resultado. Por ejemplo, la siguiente consulta nos devolvería cierto si hay universidades privadas en la ciudad de Cambridge:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
ASK {
    ?university dbo:city <http://dbpedia.org/resource/Cambridge,_Massachusetts>;
                rdf:type dbo:University;
                dbp:type <http://dbpedia.org/resource/Private_school>
}
```

- *DESCRIBE*: Devuelve un subgrafo RDF que describe un recurso RDF. El recurso puede ser indicado mediante una URI o mediante una variable resultante de un patrón de consulta. Cabe tener en cuenta que se puede preguntar por más de una URI/variable a la vez. El ejemplo siguiente pide toda la información del recurso Juan Valdez de la DBpedia:

```
DESCRIBE <http://dbpedia.org/resource/Juan_Valdez>
```

Estos materiales se centrarán en el uso de la consulta de tipo *SELECT*. Para más información sobre el resto de consultas se recomienda consultar la especificación de SPARQL.

5.5. Trabajando con tipos de datos y literales

Hasta ahora hemos visto como realizar consultas SPARQL filtrando los datos en función del esquema que siguen (en qué propiedades participa cada recurso) y de sus URIs (con que recursos se relaciona). No obstante, no hemos visto como filtrar datos en función de literales (de valores). Por ejemplo, identificar los recursos que tengan la subcadena de caracterés 'Juan' en su nombre y que tengan una edad superior a 30 años. Para poder realizar filtros como estos debemos saber cómo se representan los distintos tipos de datos en SPARQL y qué tipo comparaciones están

permitidas.

La sintaxis general para literales es una cadena de caracteres (entre comillas dobles, "...", o comillas simples, '...') y el nombre de su tipo antecedido por ^^ opcionales. Así por ejemplo podríamos utilizar:

- *"1"^^xsd:integer* para indicar que 1 es un entero.
- *"1.3"^^xsd:decimal* para indicar que 1.3 es un decimal.
- *"1.0e6"^^xsd:double* para indicar un real.
- *"true"^^xsd:boolean* para indicar el valor cierto de tipo booleano.

Como podéis ver esta representación no es muy amigable. Siempre que queramos podemos utilizarla pero, por suerte, SPARQL nos permite simplificar la representación de los literales. En particular, SPARQL analiza los valores literales para identificar su tipo automáticamente. Así pues podemos utilizar:

- 1 para indicar el número 1.
- 1.3 para indicar el decimal 1.3.
- 1.0e6 para indicar que 1.0e6 es un valor real.
- true para indicar el valor booleano cierto.

Ahora que ya sabemos como expresar literales en SPARQL, ya podríamos abordar una consulta simple del tipo: obtener los países con 45 habitantes. La consulta SPARQL que resuelve la pregunta anterior es la siguiente:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT *
WHERE {
     ?country rdf:type            dbo:Country;
              dbo:populationTotal 45.
}
```

Para hacerlo se ha creado un patrón formado por dos patrones de tripletas: uno que identifica los recursos que son de tipo país y otro que identifica aquellos que tienen una población de 45 habitantes. El resultado de la consulta es un sólo país: las Islas Kerguelen (ver `http://dbpedia.org/resource/Kerguelen_Islands`).

Esta manera de filtrar datos por valor tiene muchas limitaciones, ya que no permite utilizar operadores de comparación (obtener los países con menos de un millón de habitantes por ejemplo). Para añadir patrones que utilizen comparaciones sobre posibles valores literales de las tripletas podemos utilizar la cláusula *FILTER*. Esta función nos permite filtrar sólo aquellas tripletas que satisfacen una determinada

condición. La condición se indica mediante un conjunto de expresiones booleanas, al igual que en SQL. Por ejemplo, imaginad que queremos conocer los países que tienen menos de 200 habitantes. Para hacerlo podríamos realizar la siguiente consulta:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?country
WHERE {
    ?country rdf:type            dbo:Country;
             dbo:populationTotal ?persons.
    FILTER(?persons <200).
}
```

Fijaos que para resolver la consulta hemos utilizado tres patrones de tripletas: uno para identificar los recursos de tipo país, otro para identificar el número de habitantes de dichos recursos y otro para seleccionar aquellos con menos de 200 habitantes. En el segundo patrón de tripleta se ha creado una variable, llamada *persons*, para almacenar la población de los países. Al hacer un *FILTER* sobre esta variable estamos pidiendo que se seleccione sólo aquellos valores que cumplan la condición indicada (un valor de *persons* inferior a 200 en el ejemplo).

La cláusula *FILTER* se trata como un patrón de tripleta más. En consecuéncia, podemos utilizar la cláusula *FILTER* distintas veces en una misma consulta, como podemos ver en el ejemplo siguiente:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?country, ?persons
WHERE {
    ?country rdf:type            dbo:Country;
             dbo:populationTotal ?persons.
    FILTER(?persons <200).
    FILTER(?persons >100).
}
```

La consulta anterior nos permite permite obtener los países que tienen entre 100 y 200 habitantes. Si ejecutamos la consulta sobre le punto de acceso obtendremos dos resultados: *http://dbpedia.org/resource/French_Southern_and_Antarctic_Lands* y *http://dbpedia.org/resource/Zhedna*.

Las expresiones de la cláusula *FILTER* pueden utilizar distintos operadores. Los más comunes son:

- los operadores boleanos (! para representar el no lógico, & & para representar un y lógico y || para representar un o lógico)
- los operadores de comparación (=, >=, <, <=, >y >=)
- los operadores matemáticos (+, -, * y /).

Utilizando los operadores boleanos podemos simplificar la consulta anterior de la siguiente forma:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT ?country, ?persons
WHERE {
    ?country rdf:type            dbo:Country;
             dbo:populationTotal ?persons.
    FILTER(?persons <200 && ?persons >100).
}
```

Hasta ahora hemos trabajado con tipos de datos numéricos y booleanos pero no con cadenas de caracteres. Las cadenas de caracteres en RDF pueden tener asociado un idioma. Eso permite definir una misma propiedad en distintos idiomas, por ejemplo decir que el nombre del País Guinea Ecuatorial es *Equatorial Guinea* en inglés o *Guinée équatoriale* en francés. Para indicar esto, en RDF se añade un sufijo a la cadena de caracteres que contiene una @ y el código del idioma. Así, si consultaramos el nombre de Guinea Ecuatorial en DBpedia, obtendríamos 'Guinea Ecuatorial'@es, 'Equatorial Guinea'@en y 'Guinée équatoriale'@fr entre otros valores. Los tres valores representan el nombre del país en castellano, inglés y francés respectivamente.

Por ejemplo, si ejecutamos la siguiente consulta en el punto de acceso de la DBpedia obtendremos el nombre de las Islas Kerguelen.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?name
WHERE {
    ?country rdf:type            dbo:Country;
             dbo:populationTotal 45;
             rdfs:label          ?name
}
```

El resultado nos muestra el nombre de las Islas Kerguelen en doce idiomas distintos, como podemos ver a continuación:

```
name
------------------------
"Kerguelen Islands"@en
"??? ????????"@ar
"Kerguelen"@de
"Kerguelen"@es
"Archipel des Kerguelen"@fr
"Isole Kerguelen"@it
"???????"@ja
"Kerguelen"@nl
"Wyspy Kerguelena"@pl
"???????? (?????????)"@ru
"Ilhas Kerguelen"@pt
"??????"@zh
```

Obtener una misma cadena de caracteres en distintos idiomas puede ser innecesario y confuso en algunas situaciones. En caso de que queramos obtener sus valores en un idioma determinado podemos utilizar la función *lang*. A continuación podemos ver la consulta modificada para que devuelva sólo el nombre del país en castellano (idioma 'es'):

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?name
WHERE {
```

```
    ?country rdf:type            dbo:Country;
             dbo:populationTotal 45;
             rdfs:label          ?name.
  FILTER (lang(?name)="es")
}
```

Si queremos que nuestras consultas filtren los datos en función del valor de las variables de tipo de cadena de caracteres podemos utilizar la cláusula *FILTER* en combinación con la función *regex*. La función *regex* se ejecuta sobre una variable y nos permite definir la expresión regular que los valores de la variable deberán satisfacer (algo parecido al operador *LIKE* en *SQL*).

La función *regex* permite utilizar caracteres comodín, como por ejemplo el ^ para indicar el inicio de una cadena de caracteres, el $ para indicar el final y el * para indicar una cadena de cero, uno o más caracteres.

Por ejemplo, la siguiente consulta SPARQL devolvería los recursos de la DBpedia que contienen la cadena 'Matrix' en su nombre.

Más información sobre la función *regex*

Podeis encontrar más información sobre los caracteres comodín y el funcionamiento de redex en `http://skos.um.es/TR/rdf-sparql-query/#funcex-regex` y en `https://www.w3.org/TR/xpath-functions/#regex-syntax`.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?resource
WHERE{
  ?resource rdfs:label ?name.
  FILTER regex(?name, "Matrix", "i").
}
```

El resultado de la consulta sería el siguiente:

```
resource
-------------------------------------------------------------------------
http://dbpedia.org/resource/Category:Matrix_theory
http://dbpedia.org/resource/Pauli_matrices
http://dbpedia.org/resource/Toeplitz_matrix
http://dbpedia.org/resource/Category:Wikipedia_sockpuppets_of_Matrix17
http://dbpedia.org/resource/Hessian_matrix
http://dbpedia.org/resource/Architect_(The_Matrix)
http://dbpedia.org/resource/Augmented_matrix
...
http://dbpedia.org/resource/Rotate_matrix
http://dbpedia.org/resource/Y-matrix
http://dbpedia.org/resource/Y_matrix
http://dbpedia.org/resource/Eigenmatrix
```

Fijaos que la función *regex* toma la variable a comprobar como primer parámetro y la expresión regular en el segundo parámetro. En este caso, al no haber usado caracteres comodín, la función retornaría aquellos recursos cuya etiqueta contenga la palabra indicada en cualquier parte (en nuestro caso los que contengan la palabra 'Matrix'). El tercer parámetro 'i' lo utilizamos para indicar que no queremos que la función distinga entre mayúsculas y minúsculas.

Además de las funciones *regex* existen otras funciones que podemos utilizar para filtrar los valores a devolver en una consulta SPARQL. Algunas de ellas son *datatype*, que devuelve el tipo de datos de un elemento, *str*, que convierte a texto un literal, *isUri*, que indica si un recurso es una URI, y *lang*, que indica el lenguaje asociado a una cadena de texto. En `https://www.w3.org/TR/sparql11-query/#SparqlOps`

se puede encontrar la lista de funciones permitidas en SPARQL, junto con su descripción y algunos ejemplos.

5.6. Definición de patrones opcionales

Las consultas vistas hasta ahora requieren que todos los patrones se satisfagan para obtener los datos de interés. En algunos casos eso puede ser demasiado restrictivo. Imaginemos, por ejemplo, que queremos saber los miembros fundadores de la banda de los *the Beatles*, junto con su fecha de nacimiento y de defunción (en caso de que hayan muerto). Obviamente, querríamos obtener la información tanto de los miembros vivos como de los muertos. La consulta que obtenga estos resultados de DBpedia podría ser la siguiente:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?nombre, ?nacimiento, ?defuncion
WHERE {
    dbr:The_Beatles dbo:formerBandMember ?miembro.
    ?miembro        dbo:birthDate        ?nacimiento;
                    dbo:deathDate        ?defuncion;
                    rdfs:label           ?nombre.
    FILTER (lang(?nombre) = "en")
}
```

Para obtener los valores esperados la consulta empieza identificando los miembros fundadores de la banda de los *Beatles*. Esto se hace en el primer patrón de tripletas, los 4 miembros fundadores se asignarían a la variable *miembro*. Posteriormente en los dos patrones siguientes se obtiene la fecha de nacimiento de cada miembro (en la variable *nacimiento*) y la fecha de defunción de cada miembro (en la variable *defuncion*). También se obtendía el nombre en inglés de los miembros seleccionados.

En las siguientes lineas podemos ver el resultado de la consulta:

```
nombre          nacimiento  defuncion
-----------------------------------------------
"George Harrison"@en    1943-02-25  2001-11-29
"John Lennon"@en        1940-10-09  1980-12-08
```

Fijaos que aunque los fundadores de los *Beatles* eran 4 (los que ha devuelto la consulta más Ringo Starr y Paul McCartney), la consulta sólo ha devuelto 2 resultados. Los antiguos miembros de los *Beatles* que continuan vivos, y por tanto no tienen fecha de defunción, no han sido seleccionados. El motivo es que al no tener fecha de defunción, no han superado el patrón de tripletas *?miembro dbo:deathDate ?defuncion*.

En este caso nos interesaría calcular la fecha de defunción de los fundadores de los *Beatles* pero sin que por ello se descarten los que aún no la tienen. Para resolver este problema podemos utilizar patrones opcionales. Los patrones opcionales se definen mediante una cláusula *OPTIONAL*.

Tanto las tripletas que satisfagan el patrón opcional como las que no se seleccionaran en la consulta. Por tanto, los resultados de la búsqueda seran aquellos donde el patrón opcional se cumpla, pero también los datos donde no se cumpla. Las variables ligadas al patrón opcional no tendrán valor para las tripletas donde el patrón no se ha cumplido. Utilizando esta cláusula se podría reescribir la consulta como sigue.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?nombre, ?nacimiento, ?defuncion
WHERE {
    dbr:The_Beatles dbo:formerBandMember ?miembro.
    ?miembro        dbo:birthDate        ?nacimiento;
                    rdfs:label           ?nombre.
    OPTIONAL {?miembro dbo:deathDate ?defuncion}.
    FILTER (lang(?nombre) = "en")
}
```

En este caso hemos definido como opcional sólo el patrón de tripletas que calcula la fecha de defunción. Los otros patrones de tripletas seran obligatorios. Eso garantizará que la consulta devolverá el nombre de los miembros fundadores de los *Beatles*, su fecha de nacimiento y su fecha de defunción (sólo en caso de que hayan muerto).

A continuación podemos ver el resultado de la consulta, que ahora devuelve los 4 miembros originales de la banda. Para *Paul McCarney* y *Ringo Starr* la fecha de defunción está vacía ya que, en su caso, el patrón de tripleta donde se calculaba no se satisfizo.

```
nombre                  nacimiento  defuncion
-----------------------------------------------
"George Harrison"@en    1943-02-25  2001-11-29
"John Lennon"@en        1940-10-09  1980-12-08
"Ringo Starr"@en        1940-07-07
"Paul McCartney"@en     1942-06-18
```

5.7. Definición de patrones disjuntos

Hasta ahora hemos visto como utilizar combinaciones de patrones de tripletas de forma que todos los patrones se satisfagan a la vez. En algunas ocasiones será necesario utilizar patrones de forma que se garantice que se satisfaga un patrón u otro, como en una OR lógica.

Este tipo de patrones se denominan patrones alternativos en SPARQL. Los patrones alternativos se especifican enmarcando los dos patrones disjuntos entre los simbolos *{ }* y uniendolos mediante la clausula *UNION*. Vamos a ver como funcionan mediante un ejemplo.

Supongamos que continuamos con la consulta anterior, pero que no sólo estamos interesados en los miembros fundadores de los *Beatles* sinó también en los miembros fundadores del grupo *ABBA*. Por tanto, querremos crear una consulta que nos

devuelva a la vez los fundadores de ambos grupos. Una manera de hacerlo (no la única) es utilizar patrones alternativos.

Para seleccionar los miembros de los *Beatles* utilizariamos el siguiente conjunto de patrones:

```
    dbr:The_Beatles dbo:formerBandMember ?miembro;
                    rdfs:label           ?banda.
    ?miembro        dbo:birthDate        ?nacimiento;
                    rdfs:label           ?nombre.
    OPTIONAL { ?miembro dbo:deathDate ?defuncion}.
    FILTER (lang(?nombre) = "en" && lang(?banda) = "en" )
```

Fijaos que es el mismo patrón que antes, con la salvedad que se ha añadido una variable para almacenar el nombre del grupo (*banda*).

El conjunto de patrones necesarios para identificar los miembros de ABBA sería igual que el anterior, pero cambiando la URI del recurso de los *Beatles* por la URI del recurso de *ABBA*. El resultado es el siguiente:

```
    dbr:ABBA dbo:formerBandMember ?miembro;
             rdfs:label           ?banda.
    ?miembro dbo:birthDate        ?nacimiento;
             rdfs:label           ?nombre.
    OPTIONAL { ?miembro dbo:deathDate ?defuncion}.
    FILTER (lang(?nombre) = "en" && lang(?banda) = "en" )
```

Para integrar ambos conjuntos de patrones de forma alternativa en una consulta, se engloba cada patrón entre las marcas *{ }* y se añade la clausula *UNION* en medio. Podemos ver la consulta SPARQL resultado a continuación:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?banda, ?nombre, ?nacimiento, ?defuncion
WHERE {
{
  dbr:The_Beatles dbo:formerBandMember ?miembro;
                  rdfs:label ?banda.
  ?miembro dbo:birthDate ?nacimiento;
      rdfs:label ?nombre.
  OPTIONAL { ?miembro dbo:deathDate ?defuncion}.
  FILTER (lang(?nombre) = "en" && lang(?banda) = "en" )
}
UNION
{
    dbr:ABBA dbo:formerBandMember ?miembro;
                  rdfs:label ?banda.
  ?miembro dbo:birthDate ?nacimiento;
      rdfs:label ?nombre.
  OPTIONAL { ?miembro dbo:deathDate ?defuncion}.
  FILTER (lang(?nombre) = "en" && lang(?banda) = "en" )
}
}
```

Fijaos que la consulta devuelve el valor de las variables *banda*, *nombre*, *nacimiento* y *defunción*. Dentro de cada conjunto de patrones las variables son locales, por tanto, ambos conjuntos de patrones pueden tener los mismos nombres de variables sin que haya interferencia. El resultado de la consulta devuelve el resultado esperado, como podemos ver a continuación:

```
banda               nombre                  nacimiento  defuncion
```

```
-----------------------------------------------------------------
"The Beatles"@en    "George Harrison"@en    1943-02-25  2001-11-29
"The Beatles"@en    "John Lennon"@en        1940-10-09  1980-12-08
"The Beatles"@en    "Ringo Starr"@en        1940-07-07
"The Beatles"@en    "Paul McCartney"@en     1942-06-18
"ABBA"@en           "Björn Ulvaeus"@en      1945-04-25
"ABBA"@en           "Anni-Frid Lyngstad"@en 1945-11-15
"ABBA"@en           "Benny Andersson"@en    1946-12-16
"ABBA"@en           "Agnetha Fältskog"@en   1950-04-05
```

Y hasta aquí esta breve introducción de SPARQL. Tal y como hemos comentado, no pretende ser una guia detallada de SPARQL sinó ofrecer las herramientas principales para los primeros pasos en este lenguaje. Se recomienda tener a mano las especificaciones mientras se trabaja con el lenguaje para complementar lo presentado.

Resumen

En este módulo didáctico hemos tratado el papel actual de los datos abiertos (*Open Data*) y los datos enlazados (*Linked Data*).

Hemos iniciado el módulo discutiendo los conceptos relacionados con los datos abiertos, en concreto la definición de *datos* y de *abiertos*, y viendo un ejemplo en un posible escenario concreto. Posteriormente, hemos introducido una definición formal de datos abiertos. A continuación hemos hablado sobre los beneficios de los datos abiertos y sobre los elementos clave en el proceso de publicación de datos abiertos. Hemos finalizado el capítulo con un decálogo de buenas prácticas en relación a la publicación de datos abiertos.

El tercer capítulo nos ha introducido en el concepto de datos enlazados. Como hemos visto, el modelo de las cinco estrellas de Tim Berners-Lee puede ser visto como el siguiente paso en la publicación de datos abiertos, donde además, los datos están enlazados entre sí para dotarlos de semántica. La propuesta de *linked data* propone utilizar especificaciones web (HTTP, RDF y URI) para enlazar datos de distintos orígenes y dominios, transformando los conjuntos de datos individuales en un gran conjunto de datos enlazado que contiene información relacionada de distintos ámbitos de aplicación. Al final del capítulo se han mostrado algunos de los ejemplos actuales de datos enlazados y se indica cómo y donde buscar conjuntos de datos enlazados.

En el cuarto capítulo se tratan las tecnologías necesarias para llevar a cabo el enlace de datos según una filosofía *linked data*. Se presenta el modelo de datos RDF y se muestra cómo utilizar este modelo para crear conjuntos de datos enlazados, reutilizando recursos externos disponibles en vocabularios. Posteriormente, se explica como representar datos RDF utilizando dos tipos de representación distintas: XML/RDF y Notation3, y se muestra como enriquecer las páginas web con información RDF. Finalmente, se discute brevemente qué es la web de datos, qué necesita y cómo la propuesta de datos enlazados puede acercarnos un poco más a ella.

En el último capítulo se ha introducido el concepto de SPARQL, se ha explicado cómo puede ejecutarse, ya sea mediante puntos de acceso remotos o localmente y se han introducido los aspectos básicos de su lenguaje. A lo largo del capítulo se han proporcionado ejemplos basados en datos de la DBpedia para ejemplificar los conceptos presentados y facilitar su comprensión.

Bibliografía

Dietrich, D., Gray, J., McNamara, T., Poikola, A., Pollock, R., Tait, J., Zijlstra, T. (2012). *Open Data Handbook*. Disponible en `http://opendatahandbook.org/`

Gurin, J. (2014). *Open Data Now*. USA: MacGraw-Hill Education

Wiley, D. (2010). *The Open Future: Openness as Catalyst for an Educational Reformation*. EDUCAUSE Review, vol. 45 (4), pp. 14–16, Jul-Aug 2010.

Open Knowledge. `https://okfn.org/opendata/`

Sikos, L. F.(2015). *Mastering Structured Data on the Semantic Web*. Apress. Disponible en versión electrónica desde la biblioteca de la UOC.

Heath, T., Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool Publishers. Disponible en `http://linkeddatabook.com/editions/1.0/`

Miller, P. (2010). *Linked Data Horizon Scan*. Joint Information Systems Committee (January).

XHTML . `http://www.w3c.es/Divulgacion/GuiasBreves/XHTML`

Microdata . `https://www.w3.org/TR/microdata/`

RDF (Resource Description Framework). `https://www.w3.org/RDF/`

RDFSchema . `https://www.w3.org/TR/rdf-schema/`

Proveer de semántica a los documentos Web: RDFa (Resource Description Framework with Attributes). `https://www.w3.org/TR/2008/REC-rdfa-syntax-20081014/`

SPARQL 1.1 . `https://www.w3.org/TR/sparql11-overview/`

Consulta de RDF: SPARQL (en castellano). `http://skos.um.es/TR/rdf-sparql-query/`

SPARQL by Example. `https://www.cambridgesemantics.com/semantic-university/sparql-by-example`