

Linked Data

Samuel Martín Morales alu0101359526@ull.edu.es
Jorge Domínguez González alu0101330600@ull.edu.es
Juan Diego Rendon Cachafeiro alu0101327747@ull.edu.es

November 15, 2023

Contents

1	Introducción	2
2	Linked Data	3
2.1	Componentes de Linked Data	3
2.1.1	Uniformidad de las URIs	3
2.1.2	Características del protocolo HTTP	4
2.1.3	Características de RDF	4
2.2	Esquematización de los componentes de Linked Data	4
3	URI	6
4	HTTP	7
5	RDF	8
5.1	Introducción a RDF	8
5.2	Modelo de datos RDF	8
5.3	Ejemplos de sentencias RDF	8
6	RDF y XML	11
6.1	comparación entre RDF y XML	11
6.2	Estructura del documento XML	11
7	Tim Berners-Lee	13
7.1	Principios de Linked Data según Tim Berners-Lee	13
8	Linked Data Plataform, LDP	14
9	Proyecto Linking Open Data	15
10	Linked Open Data (LOD)	16
11	Relación entre Linked Data y Open Data	17
12	Linked Data y Web Semántica	18
13	Linked Data y Web 3.0	19
14	Beneficios de Linked Data	20
15	Problemas de Linked Data	21
16	Linked Data en bibliotecas	22
17	Linked Data en la actualidad	23
18	Conclusiones	24

Chapter 1 Introducción

En la última década, la web ha experimentado una transformación fundamental desde una simple red de información hacia lo que hoy se conoce como **Linked Data** o **datos enlazados**. Este cambio, impulsado por la evolución de la web semántica, ha llevado a la adopción de un paradigma que va más allá de la presentación de información en forma de texto. Linked Data propone una visión donde los datos adquieren una estructura que facilita la creación de conexiones y enlaces entre diversos conjuntos de datos, provenientes incluso de fuentes y proveedores distintos.

De manera general, Linked Data representa un conjunto de prácticas sólidas para la publicación y conexión de datos estructurados en la web. Haciendo uso de tecnologías del W3C, como **URIs**, el **protocolo HTTP** y el modelo de datos **RDF** o **Resource Description Framework**, se establece una base que permite la identificación única de entidades, la recuperación de recursos y la descripción detallada de los mismos.

En el presente informe se tiene como objetivo explorar los fundamentos de Linked Data, desde sus principios esenciales hasta su aplicación práctica. En esencia este se centrará en cómo las URIs, el protocolo HTTP y el modelo RDF forman parte de la revolución semántica, permitiendo la interconexión de datos. Además, se examinará el impacto del Linked Open Data (LOD) y cómo este enfoque híbrido entre datos enlazados y datos abiertos está transformando la forma en la que se accede, se utiliza y se comparte la información en un mundo cada vez más interconectado.

Chapter 2 Linked Data

2.1 Componentes de Linked Data

Para comenzar con el estudio de Linked Data, es necesario entender los componentes que lo conforman. En este sentido, se puede decir que Linked Data se basa en tres pilares fundamentales: URIs, HTTP y RDF. A continuación, se describirá cada uno de estos componentes y se explicará su importancia en el contexto de Linked Data.

- **URI (Identificadores de recursos uniformes):** Una URI es una cadena de caracteres que identifica de manera única un recurso en la web. En esta perspectiva, se puede decir que una URI es un identificador de recursos uniforme, ya que, permite la identificación de los distintos recursos en la web de una manera uniforme y consistente. Además, las URIs son utilizadas por los agentes de software para acceder a los recursos de esta. 2.1.1

- **HTTP (Protocolo de Transferencia de Hipertexto):** Se hace uso del protocolo HTTP para que las URIs sean desreferenciables. Esto quiere decir que, al acceder a una URI mediante el protocolo HTTP se debe de obtener información sobre el recurso al que hace referencia la URI. Además, el protocolo HTTP permite la recuperación de recursos a través de la web. En este sentido, se puede decir que el protocolo HTTP es el protocolo de la web, ya que, es el protocolo que permite la recuperación de recursos a través de esta. 2.1.2

- **RDF (Marco de Descripción de Recursos):** RDF es un modelo estándar para describir recursos y sus relaciones haciendo uso de tripletes (sujeto, predicado, objeto). 2.1.3

- **Enlaces entre recursos:** Las URIs deben de incluir enlaces (enlaces de hipertexto) a otras URIs, de esta manera se pueden establecer relaciones entre los recursos y la navegación entre estos. Esto lo que permite es la fomentación de la creación de una red interconectada de datos en la web.

Estos componentes permiten que los datos estén interconectados, facilitando por un lado la navegación y el descubrimiento de información relacionada. Por tanto, cuando se siguen estos principios, se puede decir que los datos están enlazados (**Linked Data**), siendo estos datos fundamentales para la construcción de la web semántica, donde, la información tiene un significado definido y las máquinas pueden entender y procesar los datos de manera efectiva.

2.1.1 Uniformidad de las URIs

- **Unicidad:** Cada recurso debe tener una URI única. La unicidad garantiza que no haya conflictos ni duplicados en la identificación de los recursos. Cada URI debería de ser única en el ámbito global de la web.

- **Consistencia:** Las URIs deben de seguir un formato consistente y estandarizado. Esto permite la facilidad de comprensión y manejo por parte tanto de las máquinas como de las personas. Además, permite el establecimiento de patrones y la simplificación de su uso.

- **Persistencia:** Las URIs deben de ser persistentes. Esto quiere decir que una URI debe de ser válida y accesible en todo momento. De esta manera, se garantiza que los recursos puedan ser accedidos en el momento que se considere.

- **Desreferenciables:** Las URIs deben de ser desreferenciables. Es decir, el acceder a una URI mediante el protocolo HTTP se debe de obtener información sobre el recurso al que hace referencia la

URI. Esto permite que las URIs no solo se traten de identificadores únicos, sino que también sean enlaces a información relevante.

2.1.2 Características del protocolo HTTP

- **Cliente-Servidor:** El protocolo HTTP se basa en un modelo cliente-servidor. Esto quiere decir que, el cliente realiza una petición al servidor y este le responde con la información solicitada. En este sentido, el cliente es el agente de software que realiza la petición y el servidor es el agente de software que responde a la petición.
- **Sin estado:** El protocolo HTTP es un protocolo sin estado. Es decir, cada petición que se realiza al servidor es independiente de las demás. Por tanto, el servidor no guarda información sobre las peticiones anteriores. Esto permite que el protocolo sea simple y fácil de implementar.
- **Protocolo de aplicación:** Se trata de un protocolo de nivel de aplicación utilizado para la transferencia de información en la WWW (World Wide Web). De manera general, opera en la capa de aplicación del modelo OSI (Open Systems Interconnection) [1].
- **Mensajes:** Las comunicaciones se realizan mediante mensajes. Un solicitud de un cliente y una respuesta del servidor consisten en un encabezado y de manera opcional en un cuerpo. El encabezado contiene la información sobre dicha solicitud o respuesta y el cuerpo contiene la información que se quiere transmitir.
- **URIs:** Las distintas solicitudes y respuestas en HTTP hacen uso de identificadores de recursos uniformes para identificar los distintos recursos en la web. Esto lo que permite, es especificar la ubicación y el nombre del recurso.
- **Basado en texto:** Las distintas solicitudes y respuestas se codifican en texto. Esto lo que permite es que se facilita la comprensión y el procesamiento de las solicitudes y respuestas tanto por parte de los humanos como de las propias máquinas.
- **Métodos:** El protocolo define una serie de métodos que se utilizan para poder indicar la acción que el cliente quiere realizar. Algunos de los métodos más comunes son GET, POST, PUT, DELETE, etc.

2.1.3 Características de RDF

- **Reutilización:** RDF hace uso de URIs para identificar los recursos, permitiendo la facilidad en la reutilización de la información RDF en diferentes aplicaciones.
- **Interoperatividad:** Este está estandarizado por el W3C [5], lo que permite que diferentes aplicaciones RDF puedan trabajar juntas más fácilmente.
- **Extensibilidad:** RDF permite la extensión de vocabularios RDF existentes, lo que permite la creación de vocabularios RDF más específicos.
- **Escalabilidad:** Se puede hacer más grande según sea necesario (escalable), permitiendo que se pueda usar para representar grandes cantidades de información o datos.

2.2 Esquematización de los componentes de Linked Data

Teniendo en cuenta todo esto anterior, se puede observar a continuación una esquema de todo lo comentado anteriormente, de tal manera que los conceptos básicos puedan ser comprendidos de manera más sencilla 2.1.



Figure 2.1: Esquema de los principales componentes de Linked Data..

Chapter 3 URI

SAMUEL

Chapter 4 HTTP

SAMUEL

Chapter 5 RDF

5.1 Introducción a RDF

Como se ha comentado de manera previa, RDF es un **modelo** que permite representar propiedades y valores de propiedades. Este, se basa en principios que se encuentran establecidos haciendo uso de varios tipos de representación de datos. Por otro lado, se puede hablar de las **propiedades RDF**, estas, se asemejan a los atributos y de manera general se corresponden con los pares de atributo-valor. Por último, se puede hablar de los **recursos RDF**, estos, se asemejan a los objetos en cuanto a aspectos de la terminología del diseño orientado a objetos, por tanto, los recursos son los objetos y las propiedades son los objetos específicos y variables de una categoría.

5.2 Modelo de datos RDF

En cuanto al modelo de datos básico de **RDF** se basa en los siguientes tres tipos de objetos:

- **Recursos:** Los recursos se tratan de los objetos que se quieren describir, es decir, se enfocan como todos los elementos que son descritos por expresiones RDF. Es decir, por ejemplo, un recurso puede ser una página web completa, una parte de una página web, una colección completa de páginas, etc. Para finalizar, los recursos se deben de designar siempre por URIs más etiquetas de identificación de destino.

- **Propiedades:** Las propiedades se tratan de los atributos que se quieren describir, es decir, se trata de un aspecto específico, característica, atributo o relación utilizado para describir un recurso. Por ejemplo, una propiedad puede ser el título de una página web, el autor de una página web, la fecha de creación de una página web, etc. Cada propiedad tiene un significado en específico, define los valores permitidos, los tipos de recursos que puede describir y las relaciones con otras propiedades.

- **Sentencias:** Las sentencias se tratan de una expresión que relaciona un recurso con una propiedad y un valor. Las tres partes individuales de una sentencia se denominan como **sujeto (recurso)**, **predicado (propiedad)** y **objeto (valor de la propiedad o literal)**, es decir, los denominados **tripletes**. El objeto de una sentencia puede ser otro recurso o un valor literal, a su vez un recurso puede ser especificado por un URI o una cadena simple de caracteres que se denominan como **literales**, además, dicho recurso puede ser a su vez datos primitivos definidos por **XML** (Lenguaje de marcado extensible) [6].

5.3 Ejemplos de sentencias RDF

Para poder comprender todo esto anterior, se tiene el siguiente ejemplo de sentencia RDF:

`http://www.example.es/index.html` tiene una desarrolladora cuyo valor es Andrea López.

Sujeto: `http://www.example.es/index.html`

Predicado: "desarrollador"

Objeto: Andrea López

Tras esto, se puede observar el diagrama de nodo y arco simple que representa el ejemplo adjunto anteriormente:

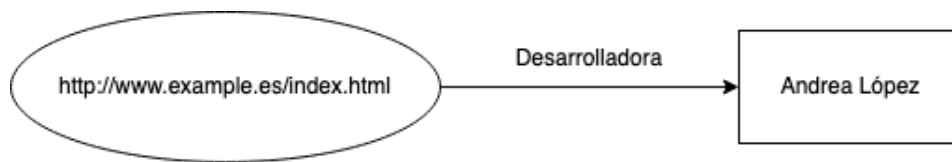


Figure 5.1: Diagrama de nodo y arco simple que representa el ejemplo de sentencia RDF.

Nota: La dirección de la flecha del arco simple es muy importante. El arco siempre debe de empezar en el sujeto y apunta hacia el objeto de la sentencia RDF. De manera general se puede usar la siguiente sentencia:

<Sujeto> TIENE <Predicado> <Objeto>

A partir del ejemplo comprendido de manera previa, se pueden especificar algunas características más para dicho ejemplo como:

El individuo cuyo nombre es Andrea López, correo electrónico <andre@example.es>, es la desarrolladores de la página web <http://www.example.es/index.html>.

Teniendo esto en cuenta, la intención del ejemplo se basa en darle valor a la propiedad **desarrolladora** mediante una entidad estructurada. Por tanto, se puede observar a continuación el diagrama de nodo y arco simple que representa el ejemplo adjunto anteriormente:

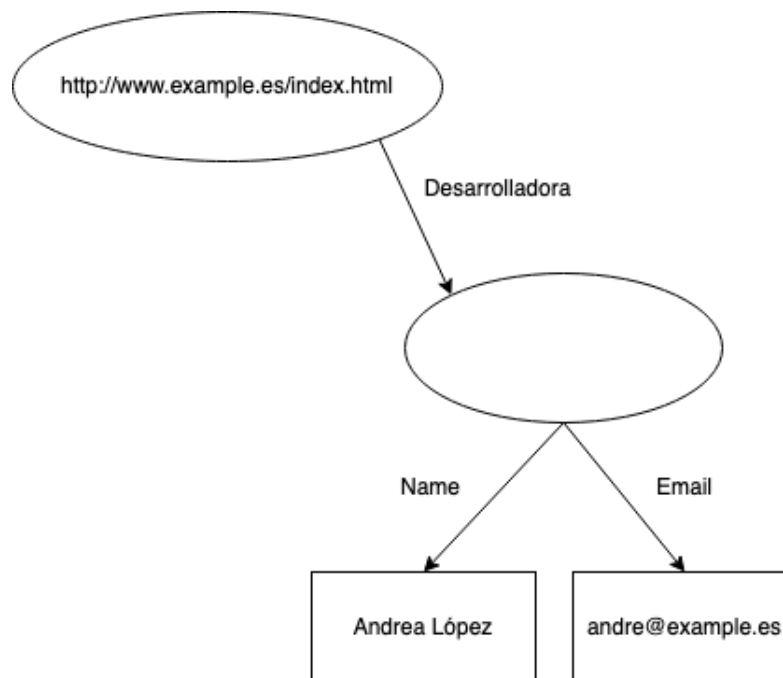


Figure 5.2: Diagrama de nodo y arco simple que representa el ejemplo de sentencia RDF.

Nota: Se puede leer el diagrama anterior como `http://www.example.es/index.html` tiene la desarrolladora cualquiera y este cualquiera tiene el nombre Andrea López y el correo electrónico `andre@example.es`.

Para finalizar, se puede implementar como ejemplo el uso de múltiples frases o sentencias RDF, por tanto, se tiene el siguiente ejemplo para ello:

El individuo al que se refiere el identificador de empleado id 15151 se llama Andrea López y

tiene la dirección de correo electrónico <andre@example.es>. Este individuo creó el recurso <http://www.example.es/index.html> y es la desarrolladora de este recurso.

Con esto en mente, se implementa el siguiente **modelo RDF**:

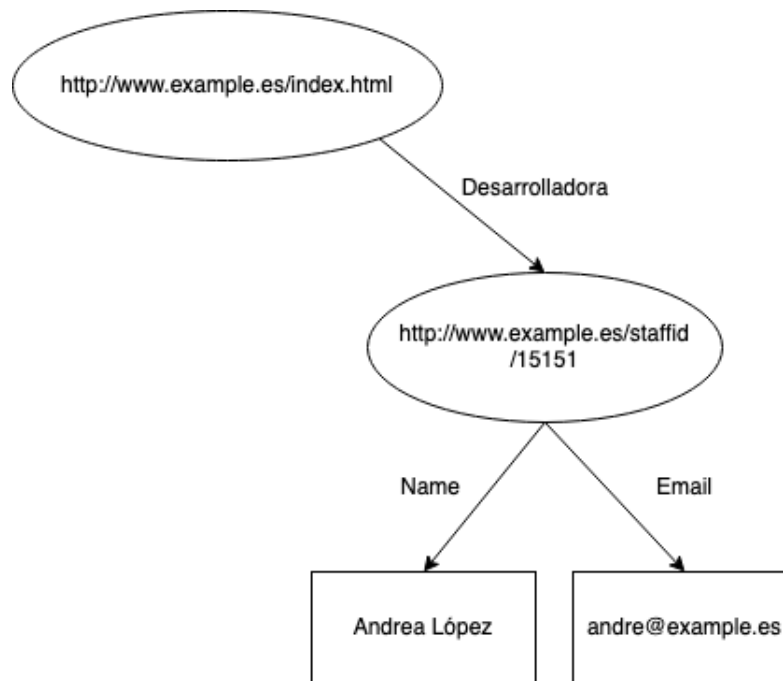


Figure 5.3: Modelo RDF que representa el ejemplo de múltiples sentencias RDF.

Chapter 6 RDF y XML

6.1 comparación entre RDF y XML

Para comenzar con esta sección, se establecerá una comparación entre RDF y XML. RDF hace uso del lenguaje XML (**Extensible Markup Language**) como método para poder representar la información.

XML no se trata de un lenguaje de etiquetado, sino que se trata de un **Metalenguaje**, es decir, de un lenguaje que establece un conjunto de reglas que permiten a su vez la creación de lenguajes de etiquetado. Es decir, XML de manera única muestra las distintas normas que se deben de seguir sobre cómo se combinan las cadenas de caracteres, cómo se tienen de que especificar las propiedad de los elementos, etc.

Es por ello que, tras entender el concepto de XML, se puede comentar que RDF hace uso de una DTD (**Document Type Definition**) de XML para desarrollar las propias etiquetas de RDF, es decir, una DTD se trata de un documento específico de trabajo dónde se establece un esquema el cuál nos permite informar sobre el contenido de cada conjunto de datos, la interpretación de estos y la forma correcta de trabajar con ellos. De esta manera, RDF al hacer uso de XML se favorece a la hora de generar nuevos conjuntos de etiquetas y el mecanismo semántico que ofrece para expresar la descripción de cualquier tipo de recurso RDF.

Por otra parte, XML permite la generación de árboles con los distintos elementos y subelementos que forman el documento, a la vez, permite la representación del documento añadiendo o eliminando etiquetas.

6.2 Estructura del documento XML

Un documento XML debe de incluir un **prólogo** y la **instancia documento**. El prólogo está formado por tres secciones distintas:

- Declaración XML.
- Declaración del tipo de documento.
- Referencia a una hoja de estilo externa.

Por tanto, para la comprensión de las distintas partes en las que se divide el documento XML se puede observar el siguiente ejemplo para ello:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE ProductoElectronico SYSTEM "producto.dtd">
<?xml-stylesheet href="productohtml.xsl" type="text/xsl"?>
<TiendaElectronica>
<ProductoElectronico>
<Nombre>Smartphone Galaxy</Nombre>
<Marca>Samsung</Marca>
<Precio>599.99</Precio>
</ProductoElectronico>
<ProductoElectronico>
<Nombre>Portátil Inspiron</Nombre>
<Marca>Dell</Marca>
```

```
<Precio>799.99</Precio>
</ProductoElectronico>
</TiendaElectronica>
```

Teniendo en cuenta el ejemplo adjunto anteriormente, se puede observar en la siguiente imagen 6.1 las distintas partes en las que se divide el prólogo en XML.

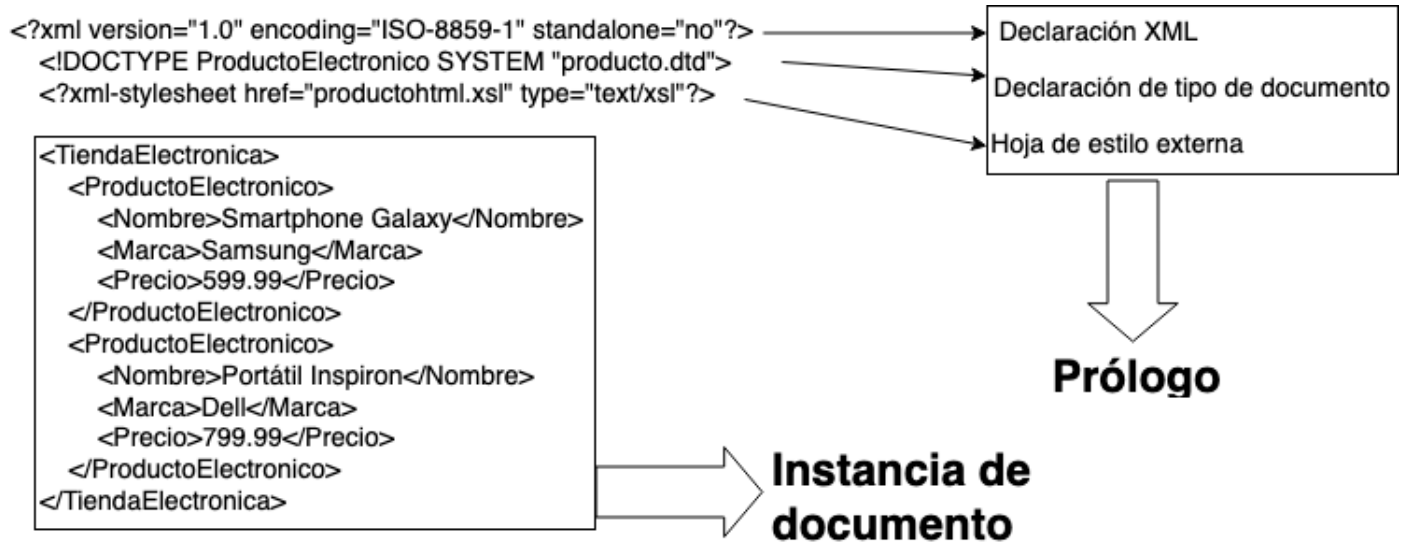


Figure 6.1: Esquema de partes en las que se divide el documento XML.

Chapter 7 Tim Berners-Lee

7.1 Principios de Linked Data según Tim Berners-Lee

Tras comprender un poco más sobre los aspectos básicos de `Linked Data` durante los cuatro capítulos desarrollados anteriormente, podemos hablar de **Tim Berners-Lee** [11], el padre de la web, el cual, en la presentación sobre `Linked Data` en las conferencias TED 2009, explicó los principios de `linked data` mediante tres reglas que permiten entender el concepto de esto de una manera mucho más sencilla:

1. Todos los objetos conceptuales posee denominaciones que se inician con `HTTP`.
2. Si se toma uno de los nombres `HTTP` y se realiza una búsqueda, se recibirá como resultado ciertos datos en un formato estándar que resulta útil para aquel que quiera conocer sobre dicha cosa o evento.
3. Cuando se obtiene información, no sólo se obtiene la información que se buscaba, sino que también se obtiene información sobre otros objetos relacionados con el objeto que se buscaba, es decir, sus relaciones. Y, cuando existen relaciones, a pesar de que no se trata de la información que se buscaba, esos propios datos también tendrán nombre que empiezan por `HTTP`, por tanto, se podrá acceder a ellos de la misma manera que se accedió a la información que se buscaba en un principio.

Con esto, la plataforma de Datos Enlazados (**LDP, Linked Data Platform**) define una serie de reglas para las distintas operaciones `HTTP` sobre los recursos web, algunos de estos están basados en `RDF`, permitiendo promover una arquitectura para leer y escribir Datos Enlazados en la web.

Chapter 8 Linked Data Plataform, LDP

SAMUEL

Chapter 9 Proyecto Linking Open Data

Example....

Chapter 10 Linked Open Data (LOD)

Example....

Chapter 11 Relación entre Linked Data y Open Data

Example....

Chapter 12 **Linked Data y Web Semántica**

Example....

Chapter 13 **Linked Data y Web 3.0**

Example....

Chapter 14 **Beneficios de Linked Data**

Example....

Chapter 15 Problemas de Linked Data

Example....

Chapter 16 **Linked Data en bibliotecas**

Example....

Chapter 17 **Linked Data en la actualidad**

Example....

Chapter 18 Conclusiones

Example....

Bibliography

- [1] CloudFlare. 2023. ¿Qué es el modelo OSI?. CloudFlare. <https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/>
- [2] 12Características. 2023. HTTP (Características, concepto y funciones). 12Características. <https://www.12caracteristicas.com/http/>
- [3] REDTECA. 2023. Protocolo HTTP. REDTECA. <https://redteca.com/hostings/http>
- [4] OpenData Euskadi. 2023. RDF (Resource Description Framework). OpenData. https://opendata.euskadi.eus/contenidos/informacion/opendata_rdf_euskadi/es_info/adjuntos/RDF.pdf
- [5] ARITMETICS. 2023. Qué es W3C. ARITMETICS. <https://www.arimetrics.com/glosario-digital/w3c>
- [6] AWS. 2023. ¿Qué es XML?. Amazon. <https://aws.amazon.com/es/what-is/xml/>
- [7] . Eva Méndez. 2001. Resource Description Framework(RDF) Especificación del Modelo y la Sintaxis. SIDAR. <http://www.sidar.org/recur/desdi/traduc/es/rdf/rdfesp.htm#basic>
- [8] María Jesús Lamarca Lapuente. 2018. Hipertexto, el nuevo concepto de documento en la cultura de la imagen. Hipertexto. <http://www.hipertexto.info/documentos/rdf.htm>
- [9] José A. Senso Ruiz. 2003. RDF: Resource Description Framework. Universidad de Granada. <https://arxiu-web.upf.edu/hipertextnet/numero-1/rdf.html#2.2>
- [10] David Fernández Medina. 2003. Estudio del modelo de representación XML/RDF. Universitat Oberta de Catalunya. <https://openaccess.uoc.edu/bitstream/10609/1024/1/20847tfc.pdf>
- [11] Wikipedia. 2023. Tim Berners-Lee. Wikipedia. https://es.wikipedia.org/wiki/Tim_Berners-Lee
- [12] . Gonzalo Villareal. 2015. Linked Data, recomendaciones de la W3C. SEDICIBlog. <https://blog.sedici.unlp.edu.ar/2015/03/13/linked-data-recomendaciones-de-la-w3c/>