



Por una  
universidad  
de **excelencia**  
y **solidaria**



Universidad  
del Cauca



ISO 9001:2015 SC-CER 450832



IQNet: CO-SC-CER450832

# Linked Data

Linked Data – Linked Open Data

# LINKED RDF



Por una  
universidad  
de **excelencia**  
**y solidaria**



Universidad  
del Cauca

# Linked Data

- La Web semántica no se trata de poner solo datos y metadatos. Se trata de enlazar todos esos datos con enlaces.
- Web Hipertexto vs. Web de Datos. Primera relaciones a anclas, en la segunda a datos arbitrarios.
- Reglas de crecimiento:
  1. URIs como nombres para las cosas.
  2. Use URI HTTP para que las personas puedan buscar esos nombres.
  3. Cuando alguien busca un URI, proporcione información útil, utilizando los estándares (RDF, SPARQL).
  4. Incluir enlaces a otros URI. para que puedan descubrir más cosas.

# Conceptos Básicos de Linked Data

- **Recursos:** Son las cosas cuyas propiedades y relaciones queremos describir en los datos.
- **Tipos de recursos:** recursos de *información* y recursos *no informativos*. Esta distinción es bastante importante en un contexto de Datos Vinculados. Todos los recursos que encontramos en la Web de documentos tradicionales, como documentos, imágenes y otros archivos multimedia, son recursos de información. Pero muchas de las cosas sobre las que queremos compartir datos no lo son: personas, productos físicos, lugares, proteínas, conceptos científicos, etc. Como regla general, todos los "objetos del mundo real" que existen fuera de la Web son recursos no informativos.
- **Identificadores de Recursos:** se usan los URI. En el contexto de los Datos Vinculados, nos limitamos a usar URI HTTP solo y evitamos otros esquemas de URI como URN y DOI

# Conceptos Básicos de Linked Data

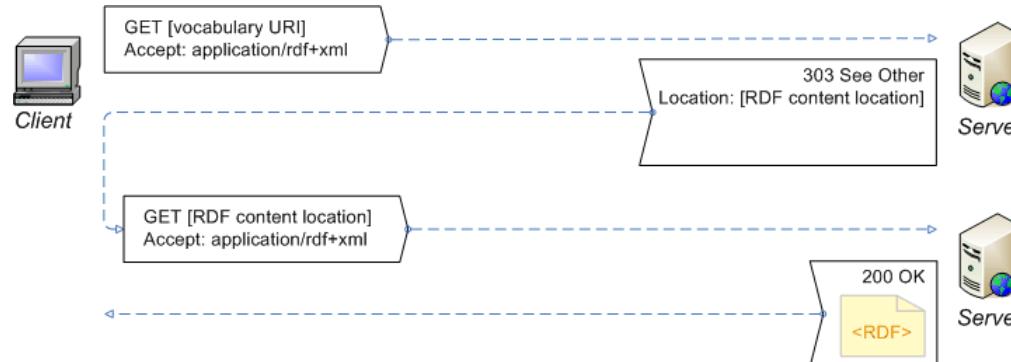
- **Representaciones:** Una representación es un flujo de bytes en un formato determinado, como HTML, RDF/XML o JPEG. Por ejemplo, una factura es un recurso de información. Podría representarse como una página HTML, como un documento PDF imprimible o como un documento RDF. Un solo recurso de información puede tener muchas representaciones diferentes, por ejemplo, en diferentes formatos, cualidades de resolución o lenguajes naturales.

# Conceptos Básicos de Linked Data

- **Desreferenciación de URI HTTP:** es el proceso de buscar un URI en la Web para obtener información sobre el recurso al que se hace referencia. La desreferenciación es diferente dependiendo del tipo de recurso:
  - **Recursos de información:** Cuando se anula la referencia de un URI que identifica un recurso de información, el servidor del propietario del URI suele generar una nueva representación, una nueva instantánea del estado actual del recurso de información y la envía de vuelta al cliente mediante el código de respuesta HTTP.200 OK
  - **Los recursos que no son de información** no se pueden desreferenciar directamente. Por lo tanto, la arquitectura web utiliza un truco para permitir que los URI que identifican recursos no informativos se desreferencian: en lugar de enviar una representación del recurso, el servidor envía al cliente el URI de un recurso de información que describe el recurso no informativo utilizando el código de respuesta HTTP. Esto se denomina redirección 303. En un segundo paso, el cliente elimina la referencia a este nuevo URI y obtiene una representación que describe el recurso original que no es de información.<sup>303 See Other</sup>

# Conceptos Básicos de Linked Data

- **Negociación de contenido:** Los navegadores HTML generalmente muestran representaciones RDF como código RDF sin procesar, o simplemente las descargan como archivos RDF sin mostrarlas. Esto no es muy útil para el usuario promedio. Por lo tanto, servir una representación HTML adecuada además de la representación RDF de un recurso ayuda a los humanos a averiguar a qué se refiere un URI.
- **Solución:** Los clientes HTTP envían encabezados HTTP con cada solicitud para indicar qué tipos de representación prefieren. Los servidores pueden inspeccionar esos encabezados y seleccionar una respuesta adecuada. Si los encabezados indican que el cliente prefiere HTML, entonces el servidor puede generar una representación HTML. Si el cliente prefiere RDF, entonces el servidor puede generar RDF
- **Ejemplo:**
  - <http://www4.wiwiss.fu-berlin.de/factbook/resource/Russia> (URI que identifica el recurso no informativo Rusia)
  - <http://www4.wiwiss.fu-berlin.de/factbook/data/Russia> (recurso de información con una representación RDF/XML que describe Rusia)
  - <http://www4.wiwiss.fu-berlin.de/factbook/page/Russia> (recurso de información con una representación HTML que describe Rusia)



# Conceptos Básicos de Linked Data

- **Alías de URI:** En un entorno abierto como la Web, a menudo sucede que diferentes proveedores de información hablan sobre el mismo recurso no informativo, por ejemplo, una ubicación geográfica o una persona famosa. Como no se conocen entre sí, introducen diferentes URI para identificar el mismo objeto del mundo real. Por ejemplo: DBpedia, una fuente de datos que proporciona información que se ha extraído de Wikipedia, utiliza el <http://dbpedia.org/resource/Berlin> URI para identificar Berlín. Geonames es una fuente de datos que proporciona información sobre millones de ubicaciones geográficas utiliza el URI <http://sws.geonames.org/2950159/> para identificar Berlín. Como ambos URI se refieren al mismo recurso no relacionado con la información, se denominan alias de URI. Los alias de URI son comunes en la Web de Datos, ya que no se puede esperar de manera realista que todos los proveedores de información acuerden los mismos URI para identificar un recurso no relacionado con la información. Los alias URI proporcionan una función social importante a la Web de Datos, ya que se desreferencian a diferentes descripciones del mismo recurso no informativo y, por lo tanto, permiten expresar diferentes puntos de vista y opiniones. Para poder rastrear que diferentes proveedores de información hablan sobre el mismo recurso no informativo, es una práctica común que los proveedores de información establezcan enlaces **owl:sameAs** a alias de URI que conocen

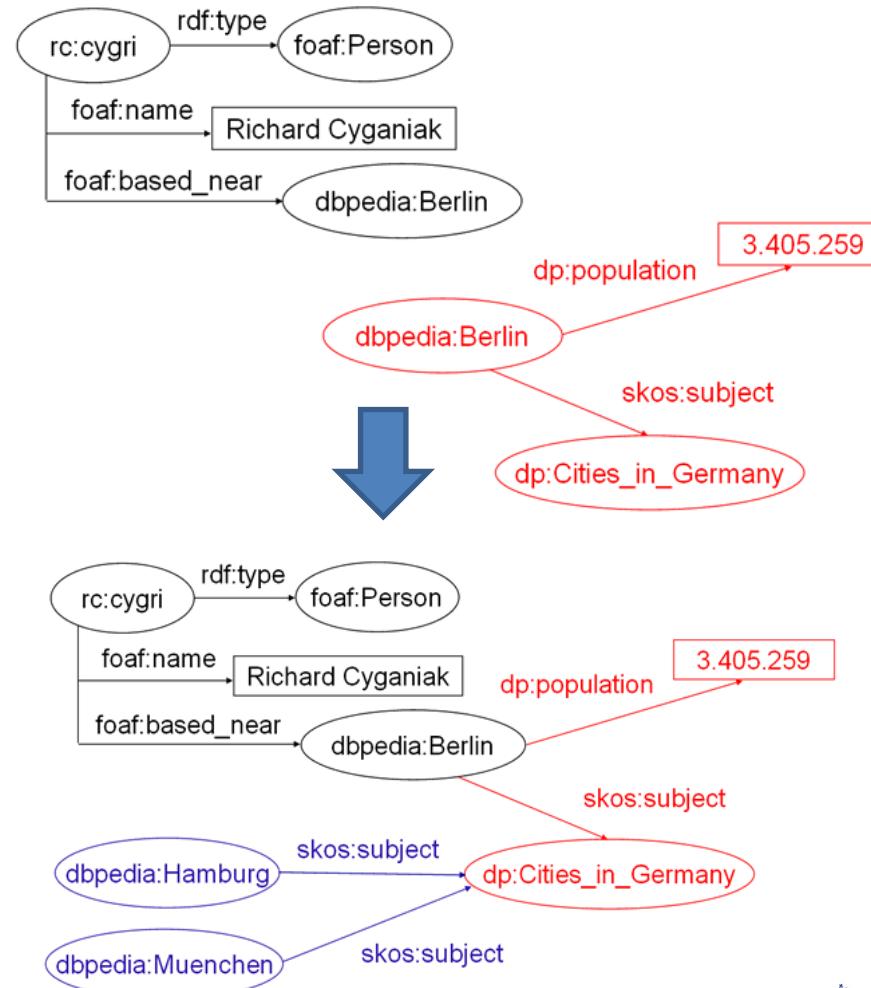
# Enlaces en un RDF

- **Tipos de Enlaces RDF:**

- **Triples literales:** tener un literal RDF como una cadena, número o fecha como objeto. Los triples literales se utilizan para describir las propiedades de los recursos.
- **Enlaces RDF:** representar vínculos tipados entre dos recursos. Los vínculos RDF constan de tres referencias URI.

- Los enlaces RDF son la base de la “Web of Data”. Al desreferenciar el URI que aparece como destino de un vínculo, se obtiene una descripción del recurso vinculado.

<http://richard.cyganiak.de/foaf.rdf#cygri>



# Enlaces RDF Externos

# Two RDF links taken from Dbpedia

```
<http://dbpedia.org/resource/Berlin>
owl:sameAs <http://sws.geonames.org/2950159/> .
<http://dbpedia.org/resource/Tim_Berners-Lee>
owl:sameAs <http://www4.wiwiss.fu-berlin.de/dblp/resource/person/100007> .
```

# RDF links taken from Tim Berners-Lee's FOAF profile

```
<http://www.w3.org/People/Berners-Lee/card#i>
owl:sameAs <http://dbpedia.org/resource/Tim_Berners-Lee> ;
foaf:knows <http://www.w3.org/People/Connolly/#me> .
```

# RDF links taken from Richard Cyganiak's FOAF profile

```
<http://richard.cyganiak.de/foaf.rdf#cygr>
foaf:knows <http://www.w3.org/People/Berners-Lee/card#i> ;
foaf:topic_interest <http://dbpedia.org/resource/Semantic_Web> .
```

# Beneficios del Uso del Modelo de Datos RDF en Linked Data

- Los clientes pueden buscar cada URI en un gráfico RDF a través de la Web para recuperar información adicional.
- La información de diferentes fuentes se fusiona naturalmente.
- El modelo de datos permite establecer vínculos RDF entre datos de diferentes orígenes.
- El modelo de datos permite representar información que se expresa utilizando diferentes esquemas en un solo modelo.
- Combinado con lenguajes de esquema como RDF-S o OWL, el modelo de datos le permite usar tanta o tan poca estructura como necesite, lo que significa que puede representar datos estrechamente estructurados, así como datos semiestructurados.

# Elegir URI

- Los recursos se nombran con referencias URI. Al publicar datos vinculados, debe dedicar un poco de esfuerzo a elegir buenos URI para sus recursos.
  - Por un lado, deben ser buenos nombres que otros editores puedan usar.
  - Por otro lado, tendrá que poner en marcha una infraestructura técnica para que sean desreferenciables.

# ¿Qué vocabularios Usar?

- Debe reutilizar términos de vocabularios conocidos siempre que sea posible. Solo debe definir nuevos términos usted mismo si no puede encontrar los términos requeridos en los vocabularios existentes.
- Lista más usada de vocabularios:
  - [Friend-of-a-Friend \(FOAF\)](#), vocabulario para describir a las personas.
  - [Dublin Core \(DC\)](#) define atributos generales de metadatos. Vea también su nuevo [borrador de dominios y rangos](#).
  - [Semantically-Interlinked Online Communities \(SIOC\)](#), vocabulario para representar comunidades en línea.
  - [Descripción de un proyecto \(DOAP\)](#), vocabulario para describir proyectos.
  - [Sistema de Organización del Conocimiento Simple \(SKOS\)](#), vocabulario para representar taxonomías y conocimientos poco estructurados.
  - [Music Ontology](#) proporciona términos para describir artistas, álbumes y pistas.
  - [Vocabulario de revisión](#), vocabulario para representar revisiones.
  - [Creative Commons \(CC\)](#), vocabulario para describir los términos de la licencia.
- Una lista más extensa [de vocabularios conocidos](#) es mantenida por el proyecto de la comunidad [W3C SWEOLinking Open Data](#).
- Es una práctica común mezclar términos de diferentes vocabularios. Recomendamos especialmente el uso de las propiedades [rdfs:label](#) y [foaf:depiction](#) siempre que sea posible, ya que estos términos son bien compatibles con las aplicaciones cliente.

# ¿Qué Vocabularios Usar?

- Si necesita referencias URI para lugares geográficos, áreas de investigación, temas generales, artistas, libros o CD, debe considerar el uso de URI de fuentes de datos dentro del proyecto de la comunidad [W3C SWEO Linking Open Data](#), por ejemplo, [Geonames](#), [DBpedia](#), [Musicbrainz](#), [dbtune](#) o [RDF Book Mashup](#). Los dos principales beneficios de usar URI de estas fuentes de datos son:
  1. Los URI son desreferenciables, lo que significa que se puede recuperar una descripción del concepto de la Web. Por ejemplo, el uso del URI de DBpedia <http://dbpedia.org/page/Doom> para identificar el juego de computadora Doom le brinda una descripción extensa del juego que incluye resúmenes en 10 idiomas diferentes y varias clasificaciones.
  2. Los URI ya están vinculados a los URI de otras fuentes de datos. Por ejemplo, puede navegar desde el URI de DBpedia <http://dbpedia.org/resource/Berlin> a los datos sobre Berlín proporcionados por [Geonames](#) y [EuroStat](#). Por lo tanto, mediante el uso de URI de concepto de estos conjuntos de datos, se interconectan los datos con una red rica y de rápido crecimiento de otros orígenes de datos.
- Una lista más extensa [de conjuntos de datos con URI desreferenciables](#) es mantenida por el proyecto de la comunidad Linking Open Data en el ESW Wiki.
- Los buenos ejemplos de cómo los términos de diferentes vocabularios conocidos se mezclan en un documento y cómo se reutilizan los URI de concepto existentes son dados por los perfiles FOAF de [Tim Berners-Lee](#) e [Ivan Herman](#).

# ¿Cómo definir Nuevos Términos?

- Puede definir vocabularios utilizando el [lenguaje de descripción de vocabulario RDF 1.0: Esquema RDF](#) o el Lenguaje de [ontología web \(OWL\)](#). Para obtener instrucciones sobre RDFS, consulte la [sección sobre documentación de vocabulario](#) en el tutorial de SWAP y la [sección Esquema RDF](#) muy detallada del Manual RDF. OWL se presenta en la [Descripción general de OWL](#).
- Aquí damos algunas pautas para aquellos que están familiarizados con estos idiomas:
  1. **No defina nuevos vocabularios desde cero**, sino que complemente los vocabularios existentes
  2. **Proveer tanto para humanos como para máquinas**. No olvides añadir prosa, por ejemplo [rdfs:comments](#) para cada término inventado. Proporcione siempre una etiqueta para cada término mediante la propiedad [rdfs:label](#).
  3. **Hacer que los URI de términos sean desreferenciables**. Es esencial que los URI de términos sean desreferenciables para que los clientes puedan buscar la definición de un término. Por lo tanto, debe hacer que los URI de términos sean desreferenciables siguiendo las Recetas de mejores prácticas del [W3C para publicar vocabularios RDF](#).
  4. **Hacer uso de los términos de otras personas**. Las propiedades comunes para proporcionar tales asignaciones son [rdfs:subClassOf](#) o [rdfs:subPropertyOf](#).
  5. **Indique toda la información importante explícitamente**. Por ejemplo, indique todos los rangos y dominios explícitamente.
  6. **No cree modelos demasiado restringidos y frágiles; deje cierta flexibilidad para el crecimiento**. Por lo tanto, a menos que sepa exactamente lo que está haciendo, use RDF-Schema para definir vocabularios.

# Ejemplo de Definición de Nuevo vocabulario:

```
# Definition of the class "Lover"

<http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LoveVocabulary#Lover>

rdf:type rdfs:Class ;

rdfs:label "Lover"@en ;

rdfs:label "Liebender"@de ;

rdfs:comment "A person who loves somebody."@en ;

rdfs:comment "Eine Person die Jemanden liebt."@de ;

rdfs:subClassOf foaf:Person .
```

# Ejemplo de Definición de Nuevo vocabulario:

```
# Definition of the property "loves"

<http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LoveVocabulary#loves>

    rdf:type rdf:Property ;

    rdfs:label "loves"@en ;

    rdfs:label "liebt"@de ;

    rdfs:comment "Relation between a lover and a loved person."@en ;

    rdfs:comment "Beziehung zwischen einem Liebenden und einer geliebten Person."@de ;

    rdfs:subPropertyOf foaf:knows ;

    rdfs:domain <http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LoveVocabulary#Lover> ;

    rdfs:range foaf:Person .
```



Por una  
universidad  
de **excelencia**  
y solidaria



Universidad  
del Cauca

# ¿Cómo establecer enlaces a otras fuentes?

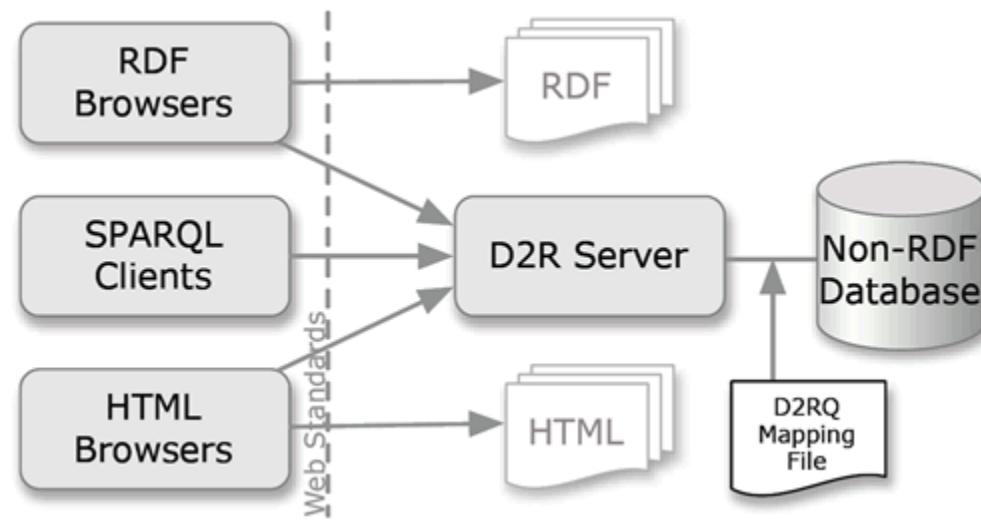
- Los vínculos RDF permiten a los exploradores y rastreadores de datos vinculados navegar entre orígenes de datos y descubrir datos adicionales.
- El dominio de aplicación determinará qué propiedades RDF se utilizan como predicados. Por ejemplo, las propiedades de enlace comúnmente utilizadas en el dominio de la descripción de personas son [foaf:knows](#), [foaf:based\\_near](#) y [foaf:topic\\_interest](#). Ejemplos de combinación de estas propiedades con valores de propiedad de [DBpedia](#), la [bibliografía de DBLP](#) y el [RDF Book Mashup](#) se encuentran en los perfiles FOAF [de Tim Berners-Lee](#) e [Ivan Herman](#).
- Es una práctica común utilizar la propiedad [owl:sameAs](#) para indicar que otra fuente de datos también proporciona información sobre un recurso específico no relacionado con la información. Un vínculo owl:sameAs indica que dos referencias URI en realidad se refieren a la misma cosa. Por lo tanto, owl:sameAs se utiliza para asignar entre diferentes alias de URI

# Consideraciones Finales para Servir Datos

- Servir información con datos vinculados
  - **¿Cuántos datos quieres servir?**
  - **¿Cómo se almacenan actualmente sus datos?**
  - **¿Con qué frecuencia cambian sus datos?**
- Servir archivos RDF estáticos
  - AddType application/rdf+xml .rdf
  - AddType text/rdf+n3; charset=utf-8 .n3
  - AddType application/x-turtle .ttl

# Consideraciones Finales para Servir Datos

- Una herramienta para servir vistas de datos vinculados en bases de datos relacionales es D2R Server. El servidor D2R se basa en una asignación declarativa entre los esquemas de la base de datos y los términos RDF de destino. Sobre la base de esta asignación, D2R Server sirve una vista de datos vinculados en la base de datos y proporciona un extremo SPARQL para la base de datos.



# Consideraciones Finales para Servir Datos

- Si su información está representada actualmente en formatos como CSV, Microsoft Excel o BibTEX y desea servir la información como Datos Vinculados en la Web, generalmente es una buena idea hacer lo siguiente:
  - Convierta sus datos en RDF utilizando una herramienta RDFizing. Hay dos ubicaciones donde se enumeran tales herramientas: [ConverterToRdf](#) mantenido en el ESW Wiki, y [RDFizers](#) mantenido por el equipo de SIMILE.
  - Después de la conversión, almacene los datos en un repositorio RDF. Se mantiene una [lista de repositorios RDF](#) en el ESW Wiki.
  - Idealmente, el repositorio RDF elegido debe venir con una interfaz de datos vinculados que se encargue de hacer que sus datos web sean accesibles. Como muchos repositorios RDF aún no han implementado interfaces de datos vinculados, también puede elegir un repositorio que proporcione un punto de enlace SPARQL y colocar [Pubby](#) como interfaz de datos vinculados frente a su punto de enlace SPARQL.

# Consideraciones Finales para Servir Datos - **wrappers**

- Un gran número de aplicaciones web han comenzado a hacer que sus datos estén disponibles en la web a través de API web. Ejemplos de fuentes de datos que proporcionan tales API incluyen eBay, Amazon, Yahoo, Google y Google Base. Una lista de API más completa se encuentra en Programmable Web. Las diferentes API proporcionan diversas interfaces de consulta y recuperación y devuelven resultados utilizando una serie de formatos diferentes, como XML, JSON o ATOM. Esto lleva a tres limitaciones generales de las API web:
  - los motores de búsqueda no pueden rastrear su contenido
  - No se puede acceder a las API web mediante navegadores de datos genéricos
  - Los mashups se implementan contra un número fijo de fuentes de datos y no pueden aprovechar las nuevas fuentes de datos que aparecen en la Web.
- Estas limitaciones se pueden superar mediante la implementación de contenedores de datos vinculados alrededor de las API. En general, los contenedores de datos vinculados hacen lo siguiente:
  - Asignan URI HTTP a los recursos no informativos sobre los que la API proporciona datos.
  - Cuando se elimina la referencia de uno de estos URI, el contenedor reescribe la solicitud del cliente en una solicitud contra la API subyacente.application/rdf+xml
  - Los resultados de la solicitud de API se transforman en RDF y se envían de vuelta al cliente.

# Bibliografía

- Introducción a los Microformatos de Dublin Core.

<http://www.webpossible.com/microformatos-dublincore/introduccion-microformatos.html>, consultado 23/07/2009.