

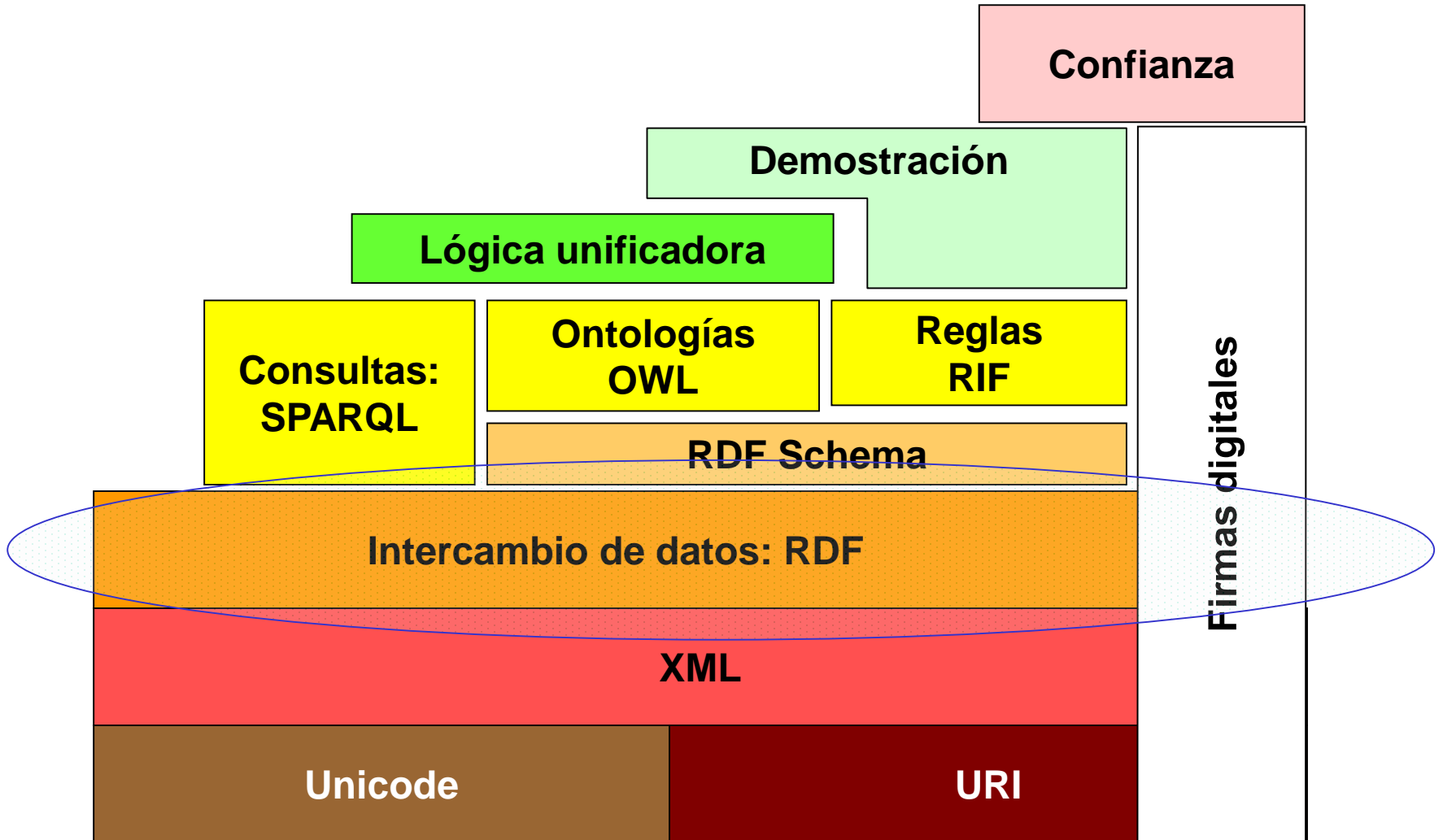


RDF

Jose Emilio Labra Gayo

Departamento de Informática

Universidad de Oviedo



RDF

Recomendación W3C en 1999

Orígenes: MCF de Apple y proyecto PICS (W3C)

Modelo de datos en forma de grafo

Las aristas son URIs

Varios formatos de representación:

RDF/XML, Turtle, N-Triples, etc.



RDF

Resource Description Framework (1998)

Descripción de recursos

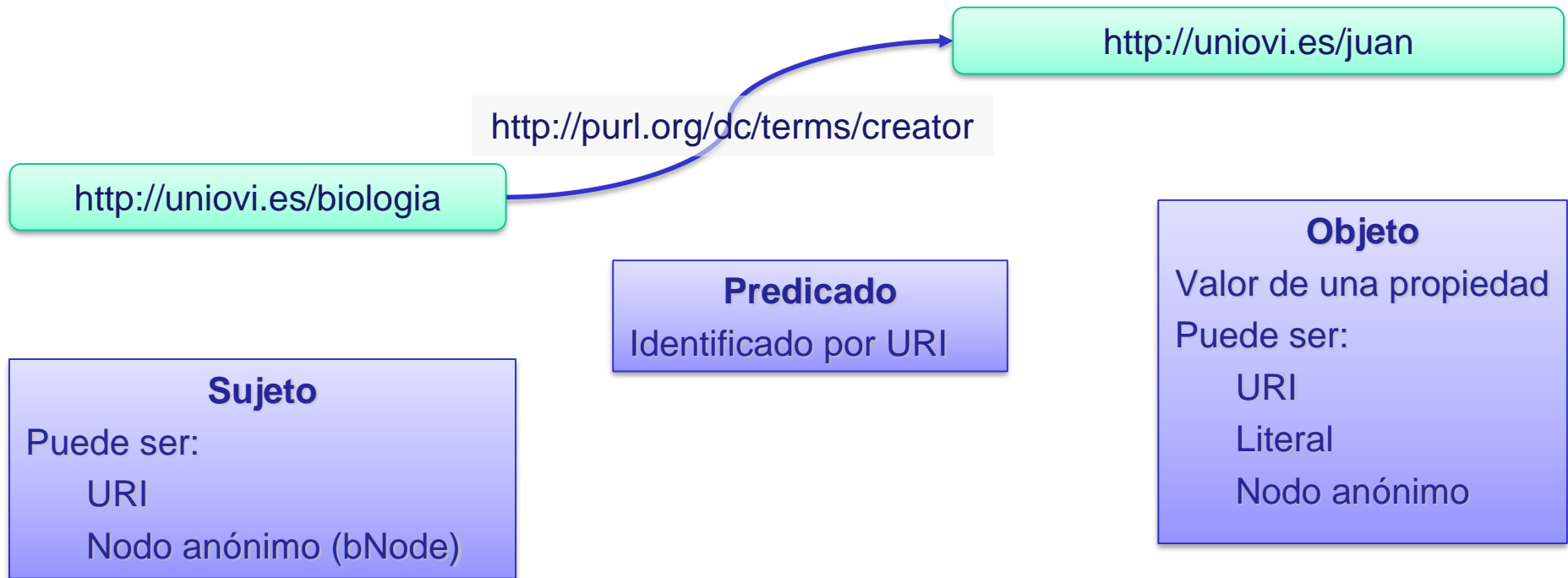
Recurso = se identifica con URI

Tripletas: Sujeto → Predicado → Objeto





Tripletas RDF



RDF en notación Turtle

Ejemplo1.ttl

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix uni: <http://uniovi.es/> .

uni:biologia dc:creator uni:juan .
```

Chequear sintaxis

rdfcats: Cargar ficheros RDF y chequear que están bien formados

```
$ rdfcats -out N3 ejemplo1.ttl
```

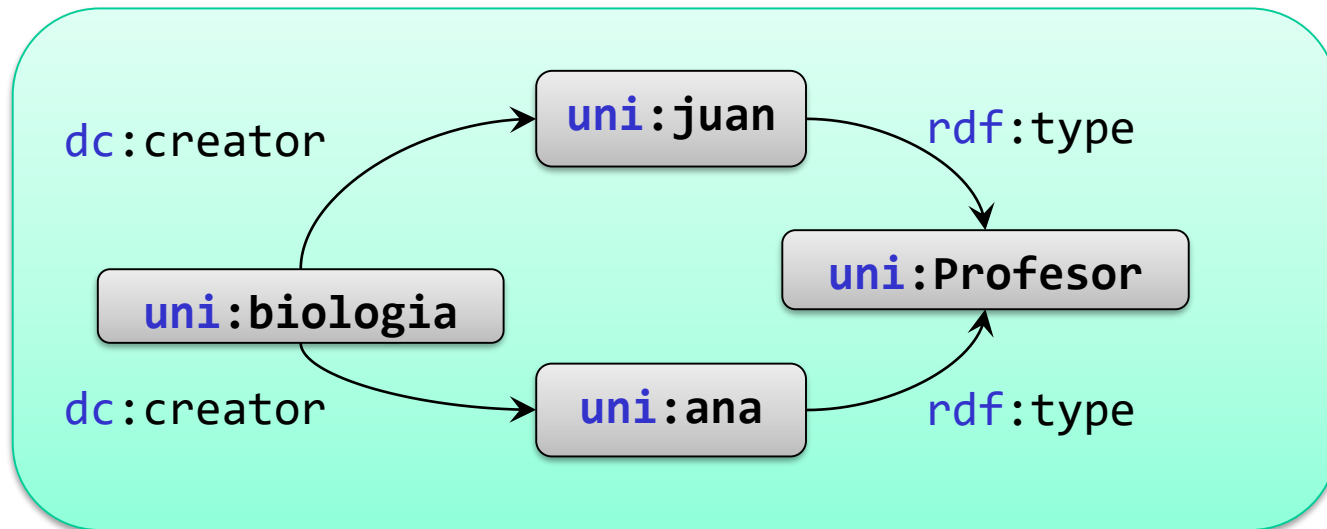
```
$ rdfcats -out RDF/XML ejemplo1.ttl
```

Grafo RDF

grafo1.ttl

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix uni: <http://uniovi.es/> .  
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

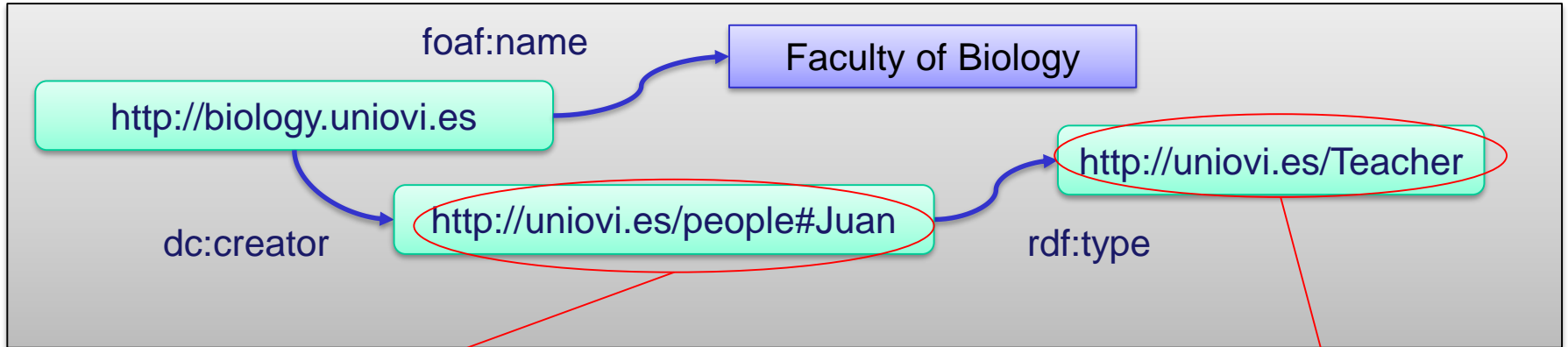
```
uni:biologia    dc:creator    uni:juan .  
uni:biologia    dc:creator    uni:ana .  
uni:juan        rdf:type      uni:Profesor .  
uni:ana         rdf:type      uni:Profesor .
```



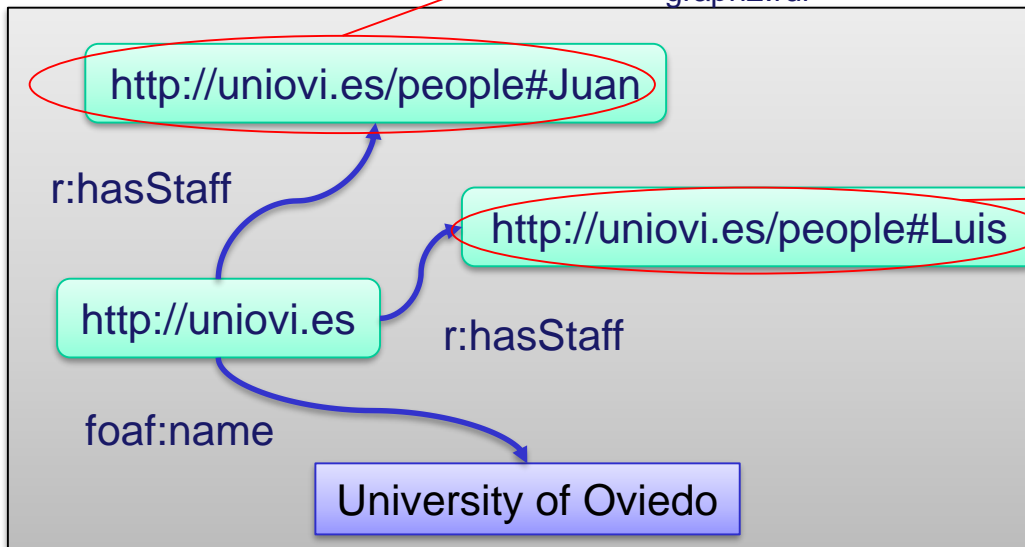


RDF es composicional

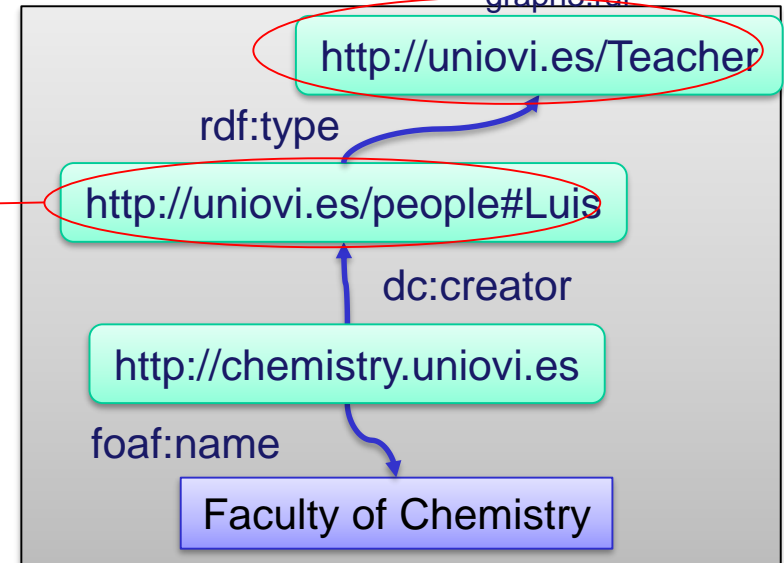
graph1.rdf



graph2.rdf

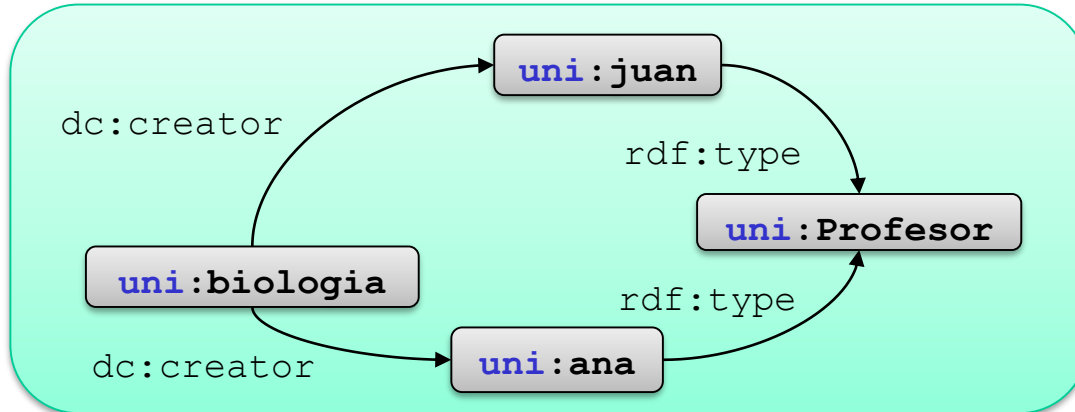


graph3.rdf

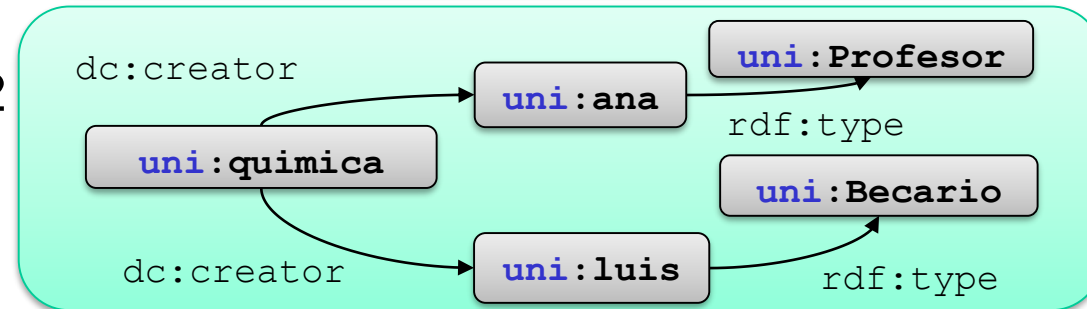


Mezclando grafos RDF

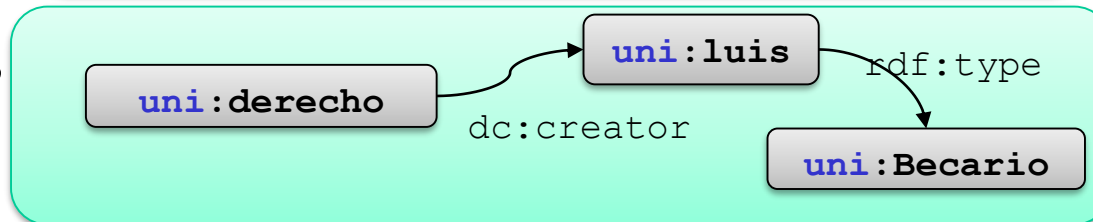
Grafo 1



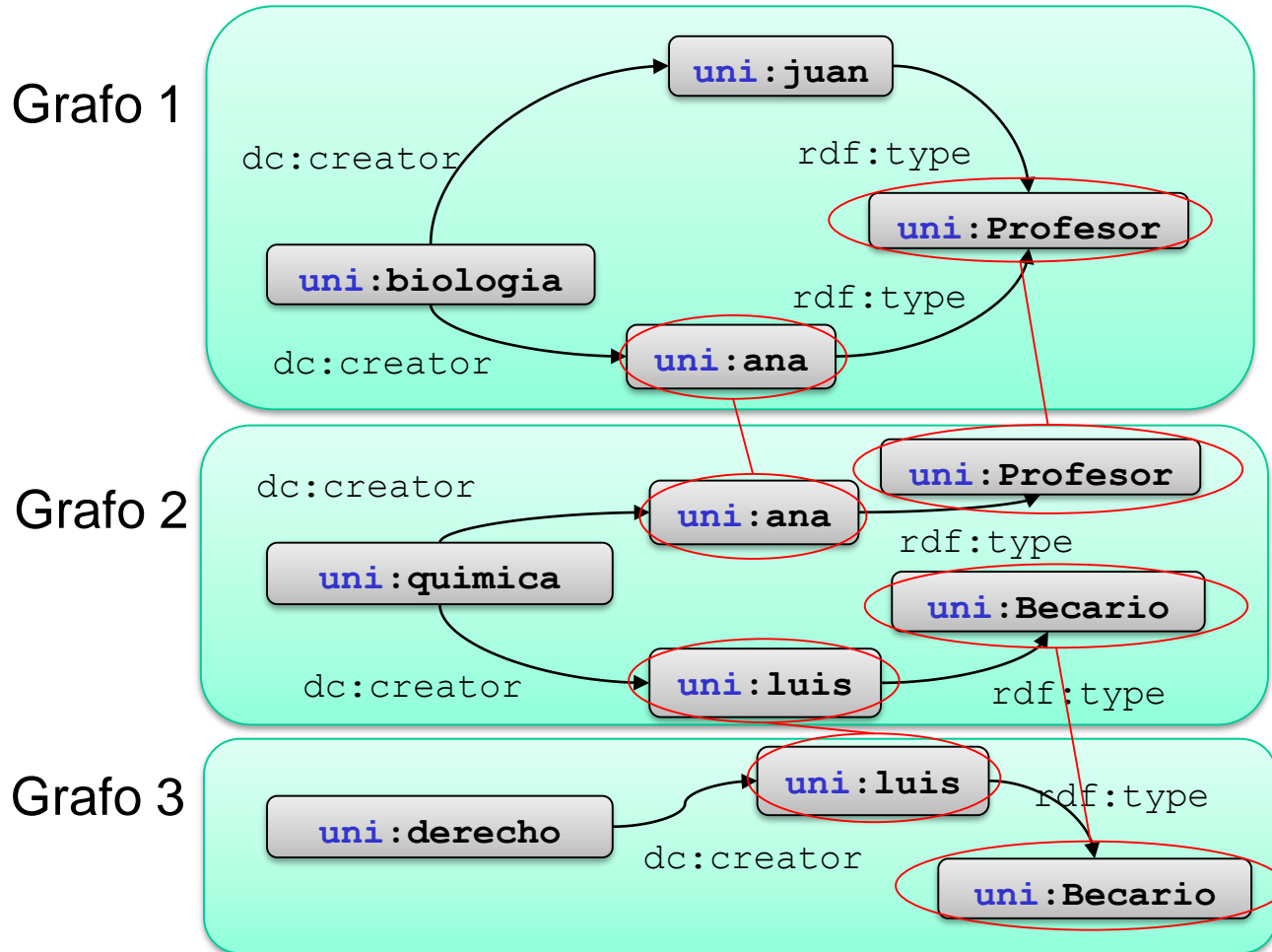
Grafo 2



Grafo 3

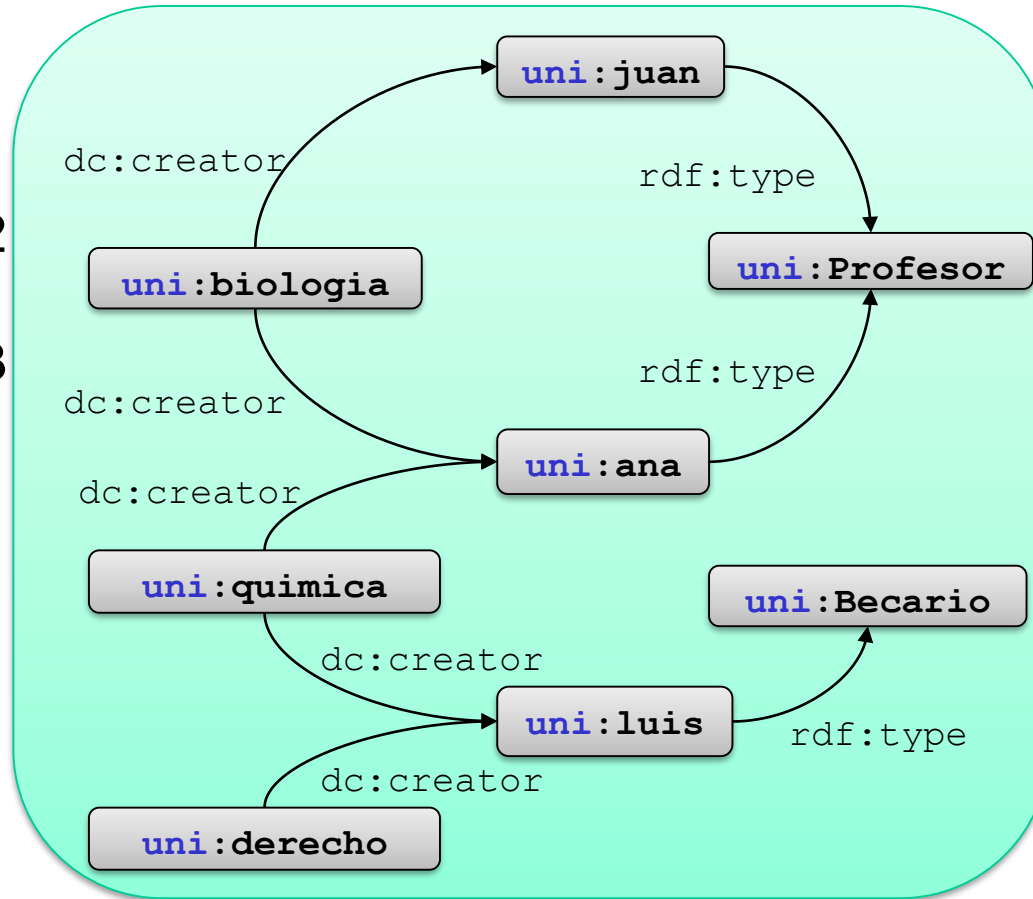


Mezclando grafos RDF



Mezclando grafos RDF

Grafo 1
+
Grafo 2
+
Grafo 3



Mezclando grafos RDF

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix uni: <http://uniovi.es/> .  
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

grafo1.ttl

```
uni:biologia    dc:creator    uni:juan .  
uni:biologia    dc:creator    uni:ana .  
uni:juan        rdf:type      uni:Profesor .  
uni:ana         rdf:type      uni:Profesor .
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix uni: <http://uniovi.es/> .  
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

grafo2.ttl

```
uni:quimica     dc:creator    uni:ana .  
uni:ana         rdf:type      uni:Profesor .  
uni:quimica     dc:creator    uni:luis .  
uni:luis        rdf:type      uni:Becario .
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix uni: <http://uniovi.es/> .  
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

grafo3.ttl

```
uni:derecho     dc:creator    uni:luis .  
uni:luis        rdf:type      uni:Becario .
```

Mezclando grafos RDF

```
$ rdfcat -out N3 grafo1.ttl grafo2.ttl grafo3.ttl
```

```
@prefix dc:      <http://purl.org/dc/elements/1.1/> .  
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix uni:    <http://uniovi.es/> .
```

```
uni:biologia    dc:creator    uni:juan , uni:ana .  
uni:quimica     dc:creator    uni:ana , uni:luis .  
uni:derecho     dc:creator    uni:luis .  
uni:luis        rdf:type      uni:Becario .  
uni:ana         rdf:type      uni:Profesor .  
uni:juan        rdf:type      uni:Profesor .
```

```
grafo1.ttl  
+  
grafo2.ttl  
+  
grafo3.ttl
```

URIs y Espacios de nombres

Espacios de nombres facilitan declaración de URIs

@prefix alias: <http://ejemplo.org/> .

alias representanta <http://ejemplo.org/>

alias:n = <http://ejemplo.org/n>

Espacio de nombres por defecto mediante

@prefix : <uri>

Para referirse al documento actual se puede usar <>

rdfcompare compara si 2 grafos son equivalentes

Propiedad type

La propiedad type

<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

declara el tipo al que pertenece un recurso

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix e:   <http://www.ejemplo.org#> .

e:Jose      rdf:type      e:Persona.
e:Juan      rdf:type      e:Persona.
```

`rdf:type` puede simplificarse como `a`

```
@prefix e:   <http://www.ejemplo.org#> .

e:Jose      a      e:Persona.
e:Juan      a      e:Persona.
```

Otros Espacios de nombres populares

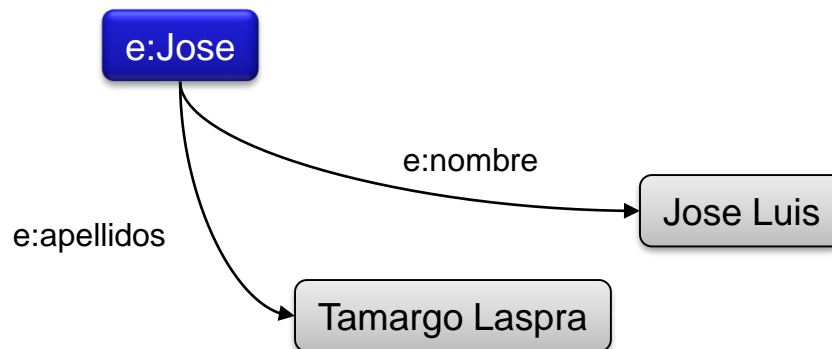
Alias	URL	Nombre	Ejemplos
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF	type, subject, predicate, object,...
rdfs:	http://www.w3.org/2000/01/rdf-schema#	RDF Schema	domain, range Class, Property subclassOf,...
owl:	http://www.w3.org/2002/07/owl#	OWL Ontologías	intersectionOf unionOf, ...
dc:	http://purl.org/dc/elements/1.1/	Dublin Core	author, date, creator, ...
foaf	http://xmlns.com/foaf/0.1/	FOAF Friend of a Friend	name, knows, etc.
skos:	http://www.w3.org/2004/02/skos/core# http://www.w3.org/2008/05/skos#	SKOS Simple Knowledge Organization System	broader, narrower,

La página <http://prefix.cc> permite recuperar la URI del prefijo más habitual

Literales

El valor puede ser una URI o un literal

```
@prefix e: <http://www.ejemplo.org#> .  
e:Jose e:nombre "Jose Luis" .  
e:Jose e:apellidos "Tamargo Laspra" .
```



Literales con idioma

Es posible asociar un idioma al valor del literal

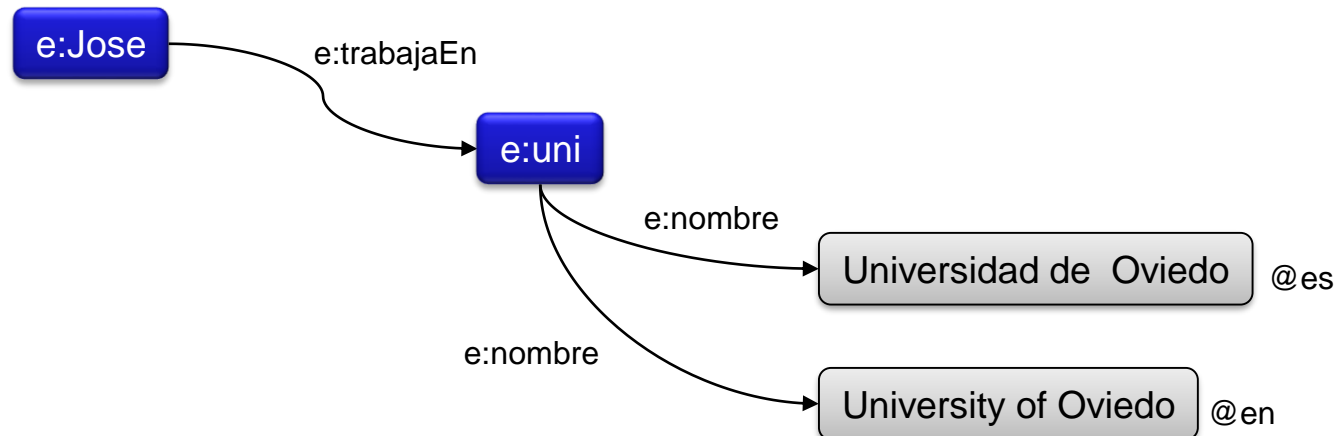
Sintaxis @idioma

```
@prefix e: <http://www.ejemplo.org#> .
```

```
e:Jose e:trabajaEn e:uni .
```

```
e:uni e:nombre "Universidad de Oviedo"@es .
```

```
e:uni e:nombre "University of Oviedo"@en .
```



Literales con tipo

Se puede declarar el tipo de datos de un literal

Permite indicar cómo analizar el valor

```
@prefix e: <http://www.ejemplo.org#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
e:Jose e:edad "23"^^xsd:integer .
```

El grafo anterior es equivalente al siguiente

```
@prefix e: <http://www.ejemplo.org#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
e:Jose e:edad "0023"^^xsd:integer .
```

Simplificaciones en Turtle

```
true = "true"^^xsd:boolean  
3 = "3"^^xsd:integer  
4.2 = "4.2"^^xsd:decimal
```

Notación Turtle

Simplificar descripciones (;)

Mediante ; pueden declararse varias descripciones de propiedades a un recurso

```
e:Jose e:conoceA e:Luis .  
e:Jose e:esPadreDe e:Ana .
```

≡

```
e:Jose e:conoceA e:Luis ;  
e:esPadreDe e:Ana .
```

Notación Turtle

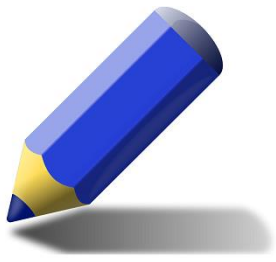
Simplificar descripciones (,)

Mediante `,` pueden declararse varios valores para una propiedad de un recurso

```
e:Pepe e:conoceA e:Juan .  
e:Pepe e:conoceA e:Luis .
```

≡

```
e:Pepe e:conoceA e:Juan , e:Luis .
```



Ejercicio

Simplificar el siguiente documento:

```
<http://www.ejemplo.org#Pepe> <http://www.ejemplo.org#conoceA> <http://www.ejemplo.org#Juan> .  
<http://www.ejemplo.org#Pepe> <http://www.ejemplo.org#conoceA> <http://www.ejemplo.org#Luis> .  
<http://www.ejemplo.org#Pepe> <http://www.ejemplo.org#conoceA> <http://www.ejemplo.org#Ana> .  
<http://www.ejemplo.org#Pepe> <http://www.ejemplo.org#esPadreDe> <http://www.ejemplo.org#Quique> .  
<http://www.ejemplo.org#Pepe> <http://www.ejemplo.org#esPadreDe> <http://www.ejemplo.org#Eva> .  
<http://www.ejemplo.org#Juan> <http://www.ejemplo.org#conoceA> <http://www.ejemplo.org#Luis> .  
<http://www.ejemplo.org#Juan> <http://www.ejemplo.org#esPadreDe> <http://www.ejemplo.org#Mar> .  
<http://www.ejemplo.org#Luis> <http://www.ejemplo.org#conoceA> <http://www.ejemplo.org#Ana> .  
<http://www.ejemplo.org#Luis> <http://www.ejemplo.org#conoceA> <http://www.ejemplo.org#Sandra> .
```

NOTA

rdcompare comparar si 2 grafos son equivalentes

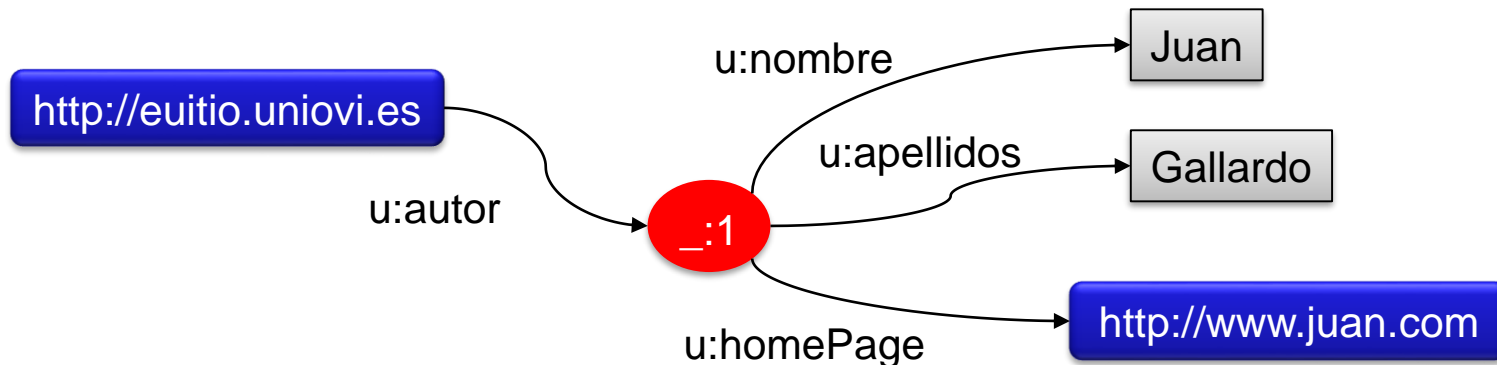
Nodos anónimos (blank nodes)

Los nodos blancos son nodos que no tienen asociada una URI

Permite hacer descripciones sobre elementos de los que no se conoce su URI

En N3 se identifican mediante `_:identificador`

```
<http://euitio.uniovi.es>    u:autor    _:1 .
_:1                          u:nombre   "Juan" .
_:1                          u:apellidos "Gallardo" .
_:1                          u:homePage <http://juan.com> .
```



Nodos anónimos (blank nodes)

Puede haber varios nodos anónimos en una descripción

Cada nodo tendrá su propio identificador

Los identificadores de nodos anónimos son locales al contexto en el que se definen

grafo1.ttl

```
<http://uniovi.es>    u:autor  _:1 .
_:1                   u:nombre "Juan" .
<http://unileon.es>  u:autor  _:2 .
_:2                   u:nombre "Jose" .
```

grafo2.ttl

```
<http://upc.es>    u:autor  _:1 .
_:1                 u:nombre "Santi" .
```




Ejercicio 1 (páginas)

Representar el siguiente conocimiento:

La página <http://www.uniovi.es> ha sido realizada por [Juan Gallardo](#), el cual tiene por correo electrónico juan@uniovi.es y tiene 26 años.

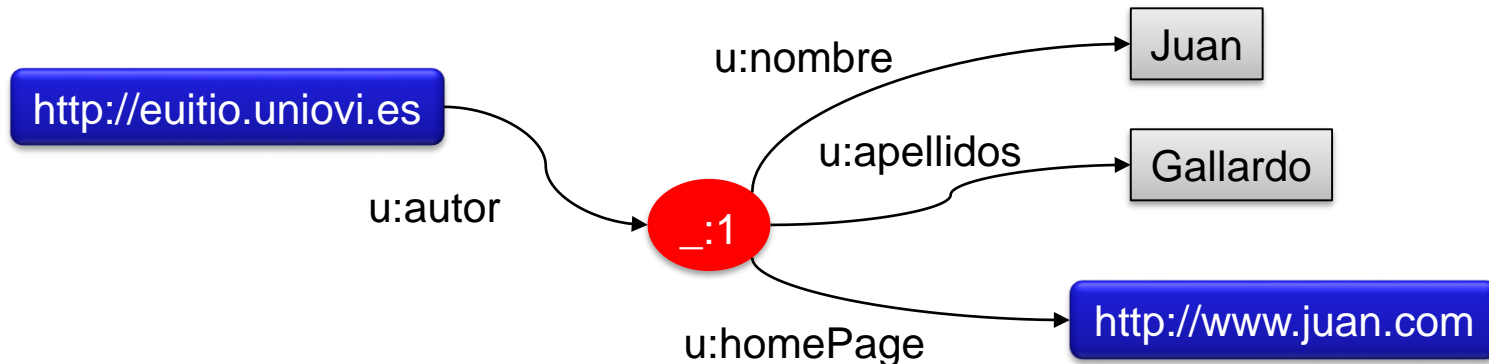
Sin embargo, la página <http://www.euitio.uniovi.es> ha sido realizada por [Isabel Castilla](#) y [Juan Gallardo](#). El correo electrónico de [Isabel](#) es isa@uniovi.es y tiene 25 años.

Nodos anónimos (blank nodes)

[] representa un nodo anónimo

Las declaraciones realizadas dentro de [] hacen referencia a dicho nodo anónimo.

```
<http://uniovi.es> u:autor [ u:nombre "Juan" ;  
u:apellidos "Gallardo";  
u:homePage <http://juan.com> ].
```



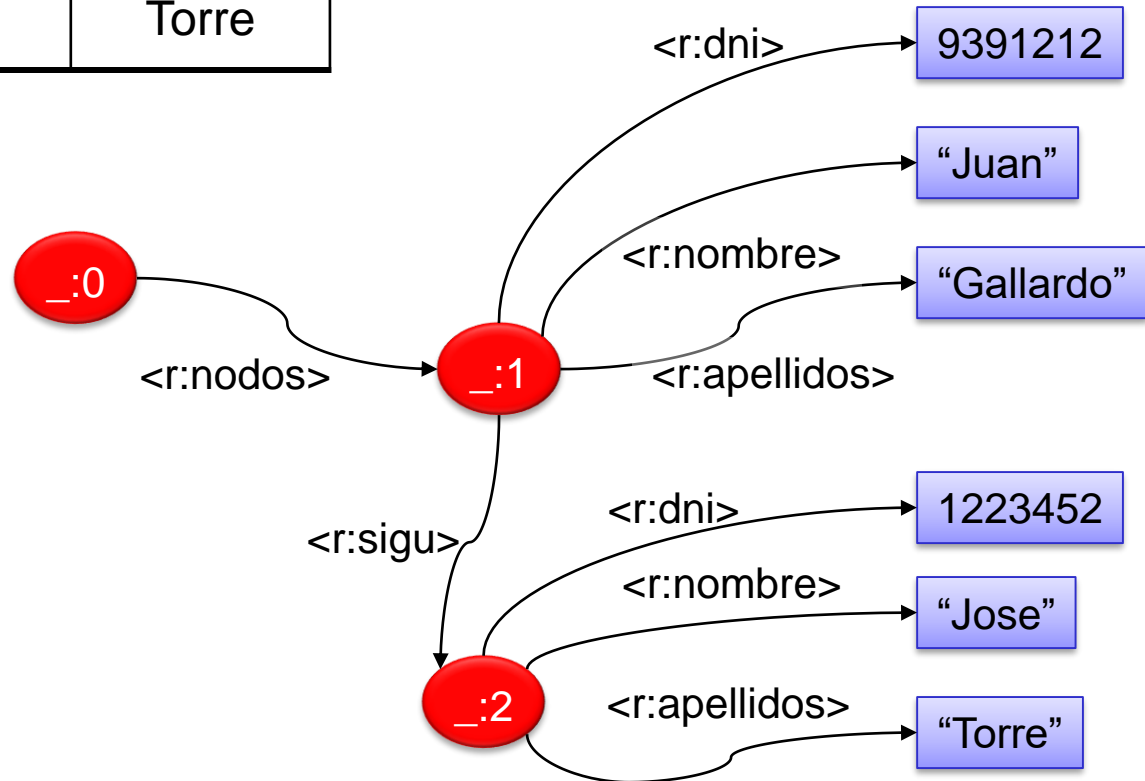
Ejercicio - Tabla

Representar información de una tabla

DNI	Nombre	Apellidos
9391212	Juan	Gallardo
1223452	Jose	Torre

Solución

DNI	Nombre	Apellidos
9391212	Juan	Gallardo
1223452	Jose	Torre



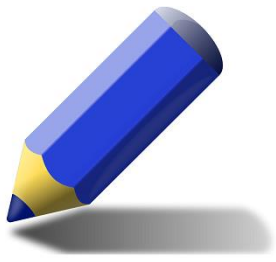
Ejercicio: Tabla con Motes

Añadir motes a la tabla anterior.

Jose Torre también es conocido como "Pepe" y como "Pepín"

NOTA: En una tabla de bases de datos, requeriría celdas con valores múltiples y con valores nulos

DNI	Nombre	Apellidos	Mote
9391212	Juan	Gallardo	?
1223452	Jose	Torre	Pepe, Pepín



Ejercicio

Representar los siguientes grafos por separado y luego mezclarlos

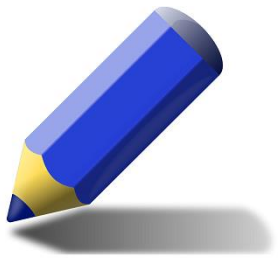
```
@prefix r: <http://ejemplo.org#>.
```

```
_:1 r:dni 9999.  
_:1 r:nombre "Juan" .  
_:1 r:esAmigoDe _:2 .  
  
_:2 r:dni 8888 .  
_:2 r:nombre "Jose" .  
_:2 r:esAmigoDe _:3 .  
  
_:3 r:dni 7777 .
```

```
@prefix r: <http://ejemplo.org#>.
```

```
_:1 r:dni 7777.  
_:1 r:nombre "Isabel" .  
_:1 r:esAmigoDe _:2 .  
  
_:2 r:dni 6666 .  
_:2 r:nombre "Quique" .  
_:2 r:esAmigoDe _:3 .  
  
_:3 r:dni 9999 .
```

NOTA: Los nodos anónimos son locales



Ejercicio

¿Cuál de los siguientes grafos es equivalente a:

```
@prefix : <> .  
  
:a :p "1" .  
:a :p _:1 .  
:a :p _:2 .  
_:1 :q "A" .  
_:2 :r "B" .
```

```
@prefix : <> .
```

```
:a :p "2" .  
:a :p _:1 .  
:a :p _:2 .  
_:1 :q "A" .  
_:2 :r "B" .
```

```
@prefix : <> .
```

```
:a :p "1" .  
:a :p _:2 .  
:a :p _:1 .  
_:2 :q "A" .  
_:1 :r "B" .
```

```
@prefix : <> .
```

```
:a :p "1" .  
:a :p _:1 .  
:a :p _:2 .  
_:1 :q "B" .  
_:2 :r "A" .
```

```
@prefix : <> .
```

```
:a :p "1" .  
:a :p _:2 .  
:a :p _:2 .  
_:1 :q "B" .  
_:1 :r "A" .
```

Listas

Similar a listas enlazadas (sintaxis especial en Turtle)

Turtle

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix e: <http://www.ejemplos.org#>.

e:Logica e:tieneAlumnos _:1 .
_:1 rdf:first e:Juan .
_:1 rdf:rest _:2 .
_:2 rdf:first e:Luis .
_:2 rdf:rest _:3 .
_:3 rdf:first e:Marcos .
_:3 rdf:rest rdf:nil .
```

Puede simplificarse como:

```
e:Logica e:tieneAlumnos
  (e:Juan e:Luis e:Marcos).
```


Sintaxis RDF/XML

RDF/XML = Sintaxis XML para representar grafos RDF

```
@prefix e: <http://www.ejemplo.org#>.
```

```
e:Juan e:nombre "Juan".
```

```
e:Juan e:apellidos "Gallardo".
```

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplo.org#">
  <rdf:Description rdf:about="http://www.ejemplo.org#Juan">
    <e:nombre>Juan</e:nombre>
    <e:apellidos>Gallardo</e:apellidos>
  </rdf:Description>
</rdf:RDF>
```

Sintaxis RDF/XML

rdf:Description captura una o varias tripletas

```
@prefix e: <http://www.ejemplo.org#>.
```

```
e:Juan e:nombre "Juan".
```

```
e:Juan e:conoceA e:Pepe.
```

```
e:Pepe e:nombre "Jose".
```

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplo.org#">
  <rdf:Description rdf:about="http://www.ejemplo.org#Juan">
    <e:conoceA rdf:resource="http://www.ejemplo.org#Pepe" />
    <e:nombre>Juan</e:nombre>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.ejemplo.org#Pepe">
    <e:nombre>Jose</e:nombre>
  </rdf:Description>
</rdf:RDF>
```

Sintaxis RDF/XML

Modelo en cebolla

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplo.org#">
  <rdf:Description rdf:about="http://www.ejemplo.org#Juan">
    <e:conoceA>
      <rdf:Description rdf:about="http://www.ejemplo.org#Pepe">
        <e:nombre>Jose</e:nombre>
      </rdf:Description>
    </e:conoceA>
    <e:nombre>Juan</e:nombre>
  </rdf:Description>
</rdf:RDF>
```

@prefix e: <http://www.ejemplo.org#>.

e:Juan e:nombre "Juan".

e:Juan e:conoceA e:Pepe.

e:Pepe e:nombre "Jose".

Sintaxis RDF/XML

rdf:ID permite hacer referencia a un nodo local

Toma como base la URL del documento base

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplo.org#"
  xml:base="http://www.ejemplo.org#">
  <rdf:Description rdf:ID="Juan">
    <e:conoceA rdf:resource="http://www.ejemplo.org#Pepe" />
    <e:nombre>Juan</e:nombre>
  </rdf:Description>
  <rdf:Description rdf:ID="Pepe">
    <e:nombre>Jose</e:nombre>
  </rdf:Description>
</rdf:RDF>
```

Sintaxis RDF/XML

Varias reglas para simplificar las expresiones

```
<rdf:Description rdf:about="http://www.ejemplo.org#Juan">  
  <rdf:type rdf:resource="http://www.ejemplo.org#Persona" />  
  <e:nombre>Juan</e:nombre>  
</rdf:Description>
```

La declaración de “type” puede incluirse en la etiqueta

```
<e:Persona rdf:about="http://www.ejemplo.org#Juan">  
  <e:nombre>Juan</e:nombre>  
</e:Persona>
```

Si las propiedades no se repiten, pueden incluirse como atributos

```
<e:Persona  
  rdf:about="http://www.ejemplo.org#Juan" e:nombre="Juan" />
```

Nodos anónimos en RDF/XML

```
@prefix u: <http://uniovi.es#>
```

```
<http://euitio.uniovi.es> u:autor      _:1 .  
_:1                       u:nombre   "Juan".  
_:1                       u:apellidos "Gallardo".  
_:1                       u:homePage <http://juan.com>
```

```
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:u="http://uniovi.es#">  
  <rdf:Description rdf:about="http://euitio.uniovi.es">  
    <u:autor rdf:parseType="Resource">  
      <u:homePage rdf:resource="http://juanlopez.com"/>  
      <u:apellidos>Lopez</u:apellidos>  
      <u:nombre>Juan</u:nombre>  
    </u:autor>  
  </rdf:Description>  
</rdf:RDF>
```

RDF: Contenedores

Tipos de contenedores

Bag: Conjunto no ordenado (permite duplicados)

Seq: Lista ordenada (permite duplicados)

Alt: Valor único alternativo (elección de un elemento del contenedor)

Los elementos se indican con `<rdf:_no>` ó con `<rdf:li>`

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://ejemplos.org#"
  <rdf:Description rdf:about="http://ejemplos.org#Logica">
    <e:tieneAlumnos>
      <rdf:Bag>
        <rdf:_1 rdf:resource="http://ejemplos.org#Juan"/>
        <rdf:_2 rdf:resource="http://ejemplos.org#Luis"/>
        <rdf:_3 rdf:resource="http://ejemplos.org#Marcos"/>
      </rdf:Bag>
    </e:tieneAlumnos>
  </rdf:Description>
</rdf:RDF>
```

Turtle

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix e:   <http://ejemplos.org#>.
e:Logica e:tieneAlumnos _:1 .
_:1 a      rdf:Bag .
_:1 rdf:_1 e:Juan .
_:1 rdf:_2 e:Luis .
_:1 rdf:_3 e:Marcos .
```

RDF: Colecciones

Listas de elementos permiten definir colecciones cerradas

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplos.org#">
  <rdf:Description rdf:about="http://www.ejemplos.org#Logica">
    <e:tieneAlumnos rdf:parseType="Collection">
      <rdf:Description rdf:about="http://www.ejemplos.org#Juan" />
      <rdf:Description rdf:about="http://www.ejemplos.org#Luis" />
      <rdf:Description rdf:about="http://www.ejemplos.org#Marcos" />
    </e:tieneAlumnos>
  </rdf:Description>
</rdf:RDF>
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix e: <http://www.ejemplos.org#>.
```

Turtle

```
e:Logica e:tieneAlumnos _:1 .
_:1 rdf:first e:Juan .
_:1 rdf:rest _:2 .
_:2 rdf:first e:Luis .
_:2 rdf:rest _:3 .
_:3 rdf:first e:Marcos .
_:3 rdf:rest rdf:nil .
```

Puede simplificarse como:

```
e:Logica e:tieneAlumnos
  (e:Juan e:Luis e:Marcos).
```


RDF: Reificación

Permite definir sentencias sobre sentencias (orden superior)

Ej. *El sitio Web de Uniovi dice que Labra es el profesor de Lógica*

Las sentencias se representan con el tipo predefinido `rdf:Statement`

Los atributos de `rdf:Statement` son: `rdf:subject`, `rdf:predicate` y `rdf:object`

Es posible añadir otros atributos a las sentencias

```
@prefix e: <http://www.ejemplos.org#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
```

```
e:Uniovi e:dice e:d1 .
e:d1 a rdf:Statement .
e:d1 rdf:subject e:Labra .
e:d1 rdf:predicate e:esProfesorDe .
e:d1 rdf:object e:Logica .
```

RDF: Reificación

Sintaxis RDF/XML

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplos.org#">
  <rdf:Description rdf:about="http://www.ejemplos.org#Uniovi">
    <e:dice>
      <rdf:Statement rdf:about="http://www.ejemplos.org#d1">
        <rdf:subject rdf:resource="http://www.ejemplos.org#Labra"/>
        <rdf:predicate rdf:resource="http://www.ejemplos.org#esProfesorDe"/>
        <rdf:object rdf:resource="http://www.ejemplos.org#Logica"/>
      </rdf:Statement>
    </e:dice>
  </rdf:Description>
</rdf:RDF>
```

e:Uniovi	e:dice	e:d1 .
e:d1	a	rdf:Statement .
e:d1	rdf:subject	e:Labra .
e:d1	rdf:predicate	e:esProfesorDe .
e:d1	rdf:object	e:Logica .

RDF: Reificación

Es posible añadir más información al enunciado reificado

Ejemplo:

La EUITIO declara en 2008 que Labra es profesor de Lógica.

```
e:Uniovi      e:dice      e:d1 .
e:d1          e:fecha    2008.
e:d1          a      rdf:Statement .
e:d1          rdf:subject  e:Labra .
e:d1          rdf:predicate e:esProfesorDe .
e:d1          rdf:object   e:Logica .
```

RDF/XML: Tipos de Datos

RDF/XML permite declarar tipos de datos

En general se utilizan los tipos de XML Schema

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplo.org#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <rdf:Description rdf:about="http://www.ejemplo.org#Pepe">
    <e:edad
      rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">23</e:edad>
  </rdf:Description>
</rdf:RDF>
```

TRUCO: Se puede ahorrar escribir la URI entera declarando una entidad

```
<!DOCTYPE rdf:RDF [ <!ENTITY xsd
  "http://www.w3.org/2001/XMLSchema#" > ] >
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:e="http://www.ejemplo.org#">
  <rdf:Description rdf:about="http://www.ejemplo.org#Pepe">
    <e:edad rdf:datatype="&xsd;integer">23</e:edad>
  </rdf:Description>
</rdf:RDF>
```

JSON/LD

Permite representar linked data en JSON

```
{
  "@context": "http://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John\_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia\_Lennon"
}
```

Herramientas para visualizar RDF

IsaViz (<http://www.w3.org/2001/11/IsaViz>)

Welkin (<http://simile.mit.edu/welkin/>)

The image displays two software interfaces for visualizing RDF data. The top window is the 'IsaViz RDF Editor', which features a menu bar (File, Edit, Views, Help) and a toolbar with icons for selection, creation, zooming, and editing. The main area shows a graph with nodes and edges. One node is a yellow square labeled 'Amarillo', and another is a green oval labeled 'mailto:werner@cartoons.com'. A curved edge connects them, labeled 'dc:Creator'. A third green oval node is labeled 'http://miDireccion/#pIolin'. A zoom control panel is visible in the top right of the graph area.

The bottom window is 'Welkin', which has a 'Load' and 'Clear' button at the top left. It features a tree view on the left with 'Predicates' and 'Resources' sections. The main area shows a dense network graph with red nodes and grey edges. Below the graph are three small plots: 'In Degree', 'Out Degree', and 'Clustering Coefficient'. At the bottom, there are buttons for 'Start', 'Circle', 'Scramble', 'Shake', and checkboxes for 'Nodes', 'Edges', 'Arrows', 'Antialias', and 'Background'. The system tray at the bottom shows the taskbar with various applications and the time 21:25.

Aplicaciones de RDF

Primeras aplicaciones

RSS 1.0, FOAF

2007 Resurgir gracias a Linked data

DBpedia, Wikidata

Dominios específicos

Salud, eGovernment, ...

Validación y descripción de RDF

Hasta 2012...validación de RDF ad-hoc

Algunas propuestas...

- Stardog ICV

- SPIN (TobQuadrant)

- Consultas SPARQL

2012 Workshop sobre validación de RDF

- <https://www.w3.org/2012/12/rdf-val/>

2013 Propuesta Shape Expressions (ShEx)

2015 SHACL (Shapes Constraint Language)

ShEx - Shape Expressions

Lenguaje de dominio específico para validar RDF

Inspirado en Turtle y RelaxNG

Basado en expresiones regulares

Expresiones regulares ampliadas a grafos

SHACL - Shapes Constraint Language

Desarrollado por W3c Data Shapes Working Group

Primer borrador: Diciembre, 2015

Última versión publicada: <http://www.w3.org/TR/shacl/>

Combinación entre SPIN, ShEx, SPARQL

Vocabulario RDF

No tiene sintaxis *human-friendly*

Fin de la Presentación

