

- Formatos de datos abiertos
  - JSON
  - CSV
  - Planillas
  - Documentos de texto
  - XML
  - RDF

- Formatos de datos abiertos
  - JSON
    - Es un formato de archivo muy fácil de ser interpretado por cualquier lenguaje de programación
    - En general es más fácil para los computadores procesaren JSON do que otras

- Formatos de datos abiertos
  - CSV (comma separated vallues)
    - Formato compacto e, por lo tanto, adecuado para transferir grandes conjuntos de dados con una misma estructura
    - Es necesario ofrecer documentación para explicitar la estructura de los datos (el significado de los campos)

- Formatos de datos abiertos
  - Planillas
    - Datos pueden ser usados de forma mediata, adicionando-se apenas la descripción correcta de las columnas
    - Para hacer una planilla disponible, puede ser necesario mudar el formato de el archivo, de forma que ella sea visitada por cualquier tipo de software

- Formatos de datos abiertos
  - Documentos de texto
    - Es barato exhibir en esos formatos, una vez que en general es el formato en que los datos fueron generados
    - Es necesario extraer la información que está almacenada en el documento (templates pueden auxiliar el procesamiento del documento)

- XML – Extensible Markup Language
  - Lenguaje de marcación extensible
  - Metalenguaje para creación de otras lenguajes
  - RDF, RDFS e OWL son lenguajes basados en XML
  - XML es el estándar para **troca** de datos en la Web
  - Adecuada para representación de datos semi-estructurados

## Exemplo de documento XML

```
<?xml version="1.0" encoding="utf-8"?>
<livraria>
  <livro id="L01" ano="1936">
    <autor> Jorge Amado </autor>
    <titulo>Mar Morto</titulo>
  </livro>
  <livro id="L04" ano="1930">
    <autor>
      <nome>Rachel</nome >
      <sobrenome>de Queiroz</sobrenome >
    </autor>
    <titulo>O Quinze</titulo>
    <genero> Romance </genero>
  </livro>
</livraria>
```

- Composición de un documento XML – Elementos
  - Los elementos son los bloques principales de la composición de un documento XML
  - Cada elemento tiene un nombre e un contenido
  - El contenido de un elemento es delimitado por marcaciones especiales, conocidas como marcación (*tag*) de inicio y marcación (*tag*) de final

## Las marcaciones de inicio e de final son obligatorias!

```
<livro>  
  <autor>Elmasri</autor>  
  <titulo>Fundamentals of Database Systems</titulo>  
  <preco>R$90.99</preco>  
</livro>
```

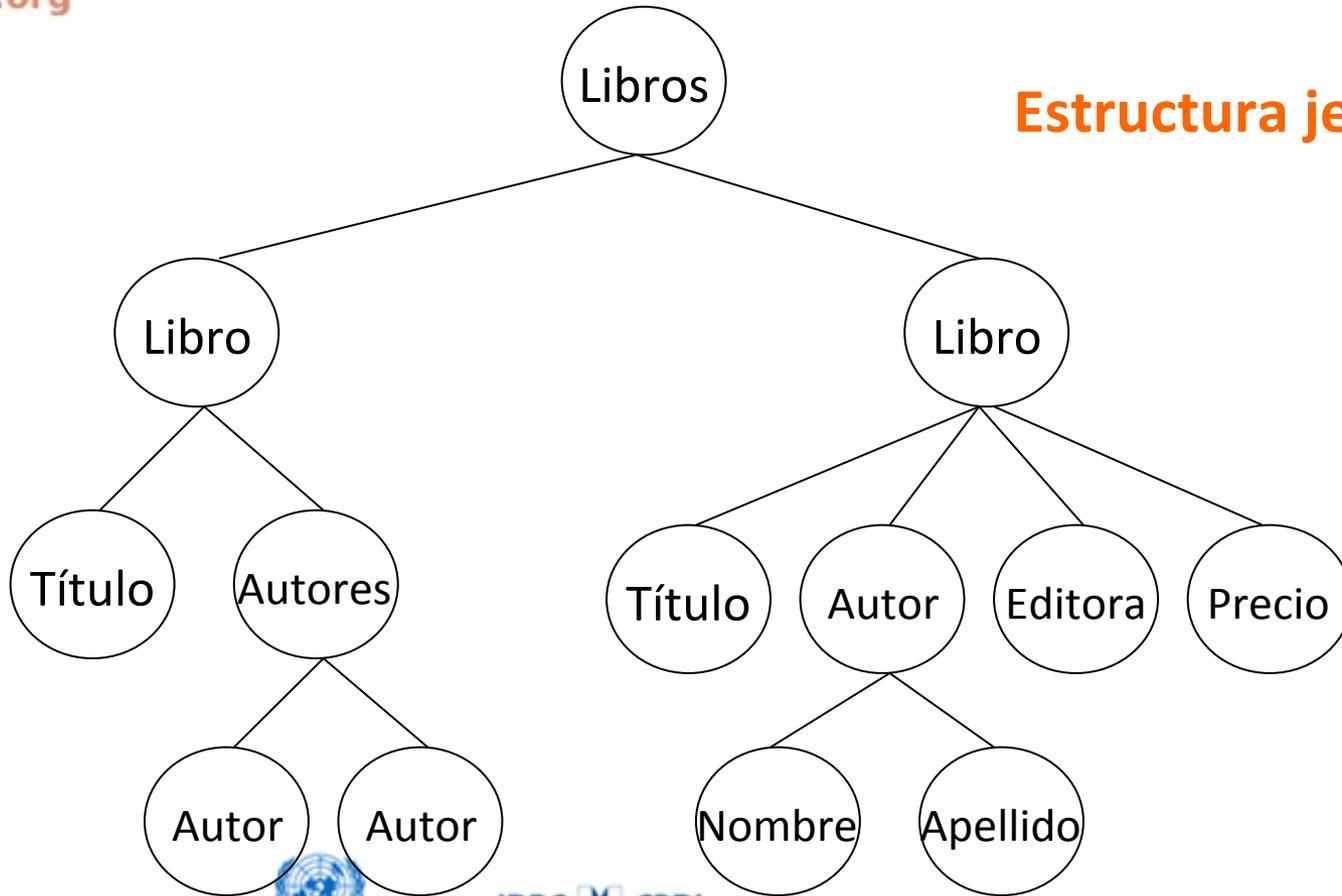
- Recuerde-se:
  - XML no define elementos
  - XML ofrece una sintaxis común para almacenar informaciones de acuerdo con una determinada estructura
  - XML separa contenido de presentación
  - Las marcaciones de un documento XML ofrecen significado a su contenido

- Composición de un documento XML - El contenido de los Elementos
  - Cada elemento puede tener
    - caracteres
    - otros elementos
    - Caracteres y otros elementos

## Un elemento puede tener otros elementos

```
<livros>  
  <libro>  
    <titulo> Fund.... </titulo>  
    <autores>  
      <autor> Navathe </autor>  
      <autor> Elmasri </autor>  
    </autores>  
  </libro>  
</livros>
```

## Estructura jerárquicas



- Composición de un documento XML
  - En la raíz do documento es necesario haber uno y sólo uno elemento
  - Todos los elementos del documento precisan ser hijos de uno elemento

```
<libro>  
....otros elementos  
</libro>
```

- Composición de un documento XML -  
Atributos
  - Es posible anexar informaciones a los elementos en la forma de atributos
  - Los atributos tienen un nombre y un valor
    - Los nombres de los atributos siguen las mismas reglas de los nombres de los elementos
    - Valores de atributos pueden ser delimitados por “ o ‘

Los elementos pueden tener uno o más atributos en el tag de inicio, y el nombre es separado de valor por “=”

```
<libro ISBN = “12345” editora = “Addison Wesley” >  
  <autor>Elmasri</autor>  
  <titulo>Fundamentals of Database Systems</titulo>  
  <precio>R$90.99</precio>  
</libro>
```

## RDF - Resource Description Framework

- RDF es el estándar para **publicación** de datos en la Web
- Modelo de datos basado en grafos
- Recursos son descritos por triplas

sujeto

predicado

objeto

Paris es la capital de Francia

## RDF

[http://dbpedia.org/resource/Gustave\\_Eiffel](http://dbpedia.org/resource/Gustave_Eiffel)



**dbpprop:mainContractor**



[http://dbpedia.org/resource/Eiffel\\_Tower](http://dbpedia.org/resource/Eiffel_Tower)



**dbpedia-owl:capital**

<http://dbpedia.org/resource/Paris>

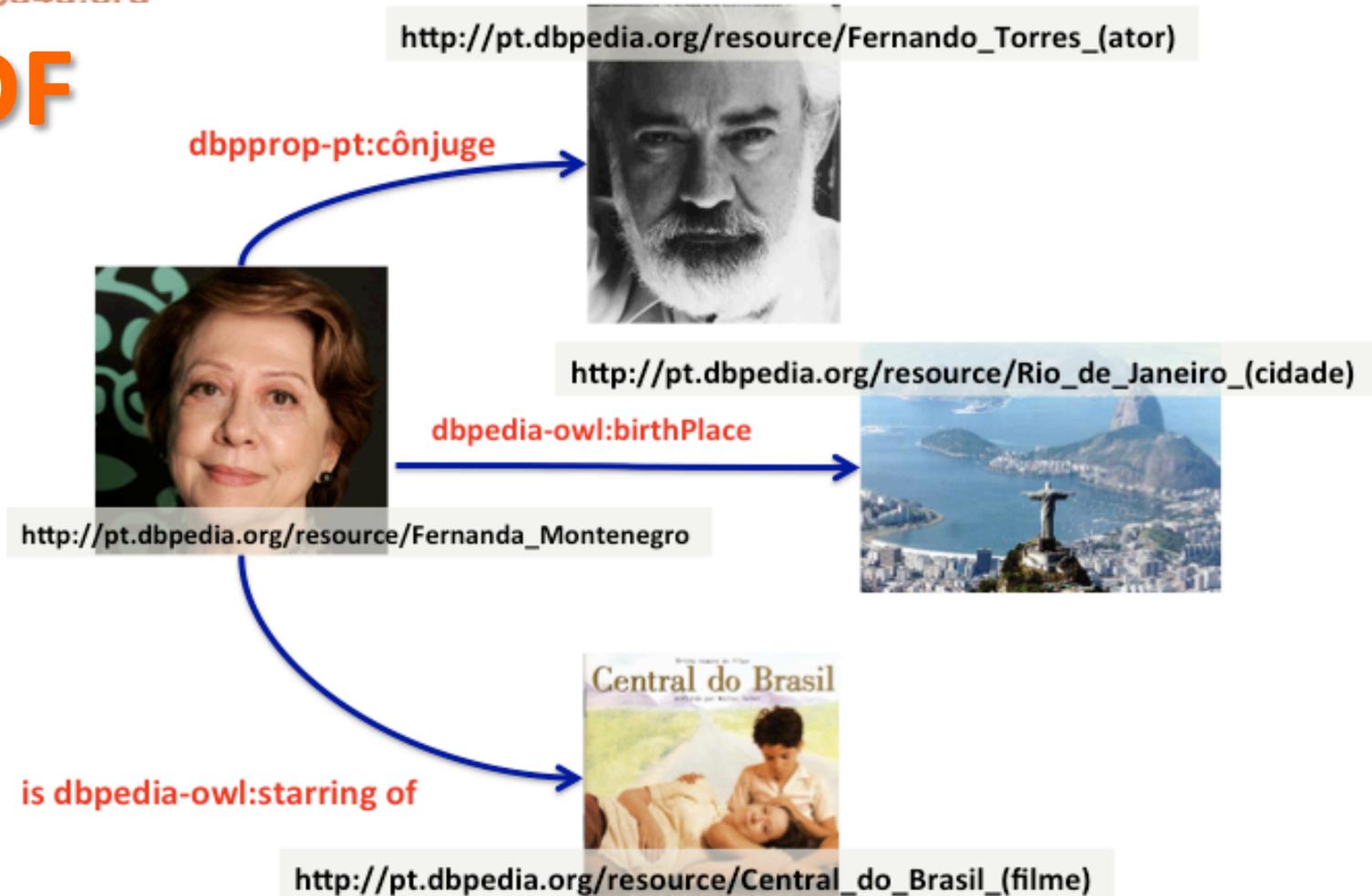
<http://dbpedia.org/resource/France>



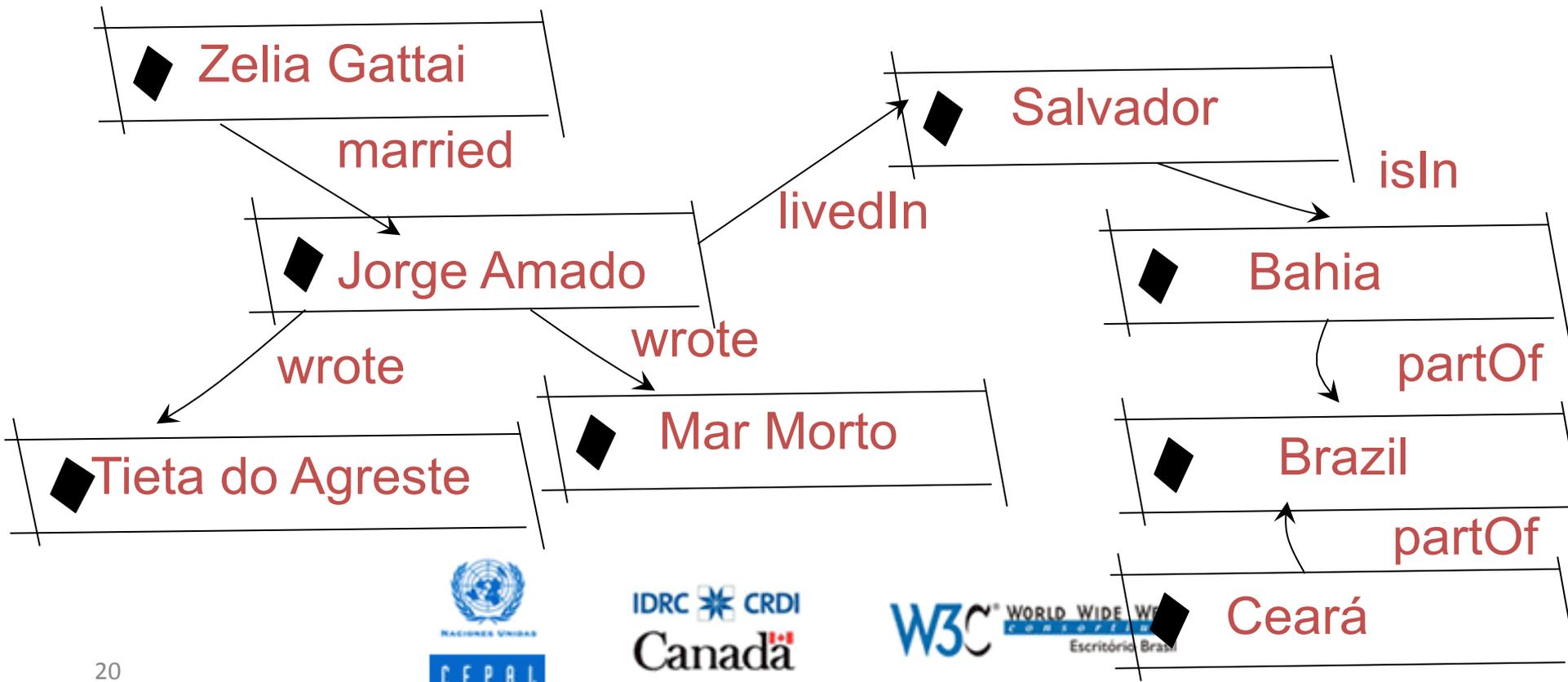
**dbpedia-owl:nationality**

**dbpedia-owl:motto**  
*(Liberty, Equality, Fraternity)*

## RDF



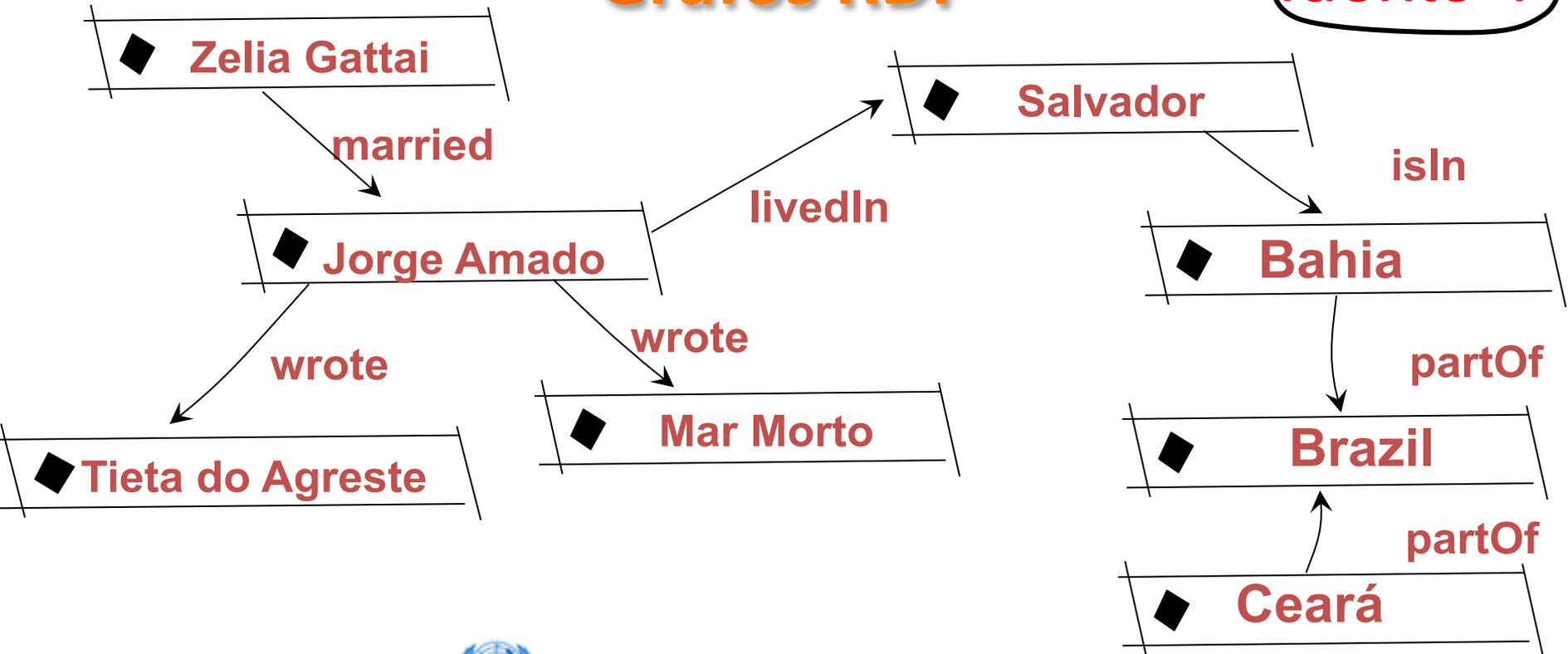
# Grafos RDF



sujeto	predicado	objeto
Jorge Amado	Wrote	Mar Morto
Zelia Gattai	Married	Jorge Amado
Jorge Amado	Lived in	Salvador
Salvador	Is in	Bahia
Bahia	Part of	Brazil
Ceará	Part of	Brazil
Jorge Amado	Wrote	Tieta do Agreste

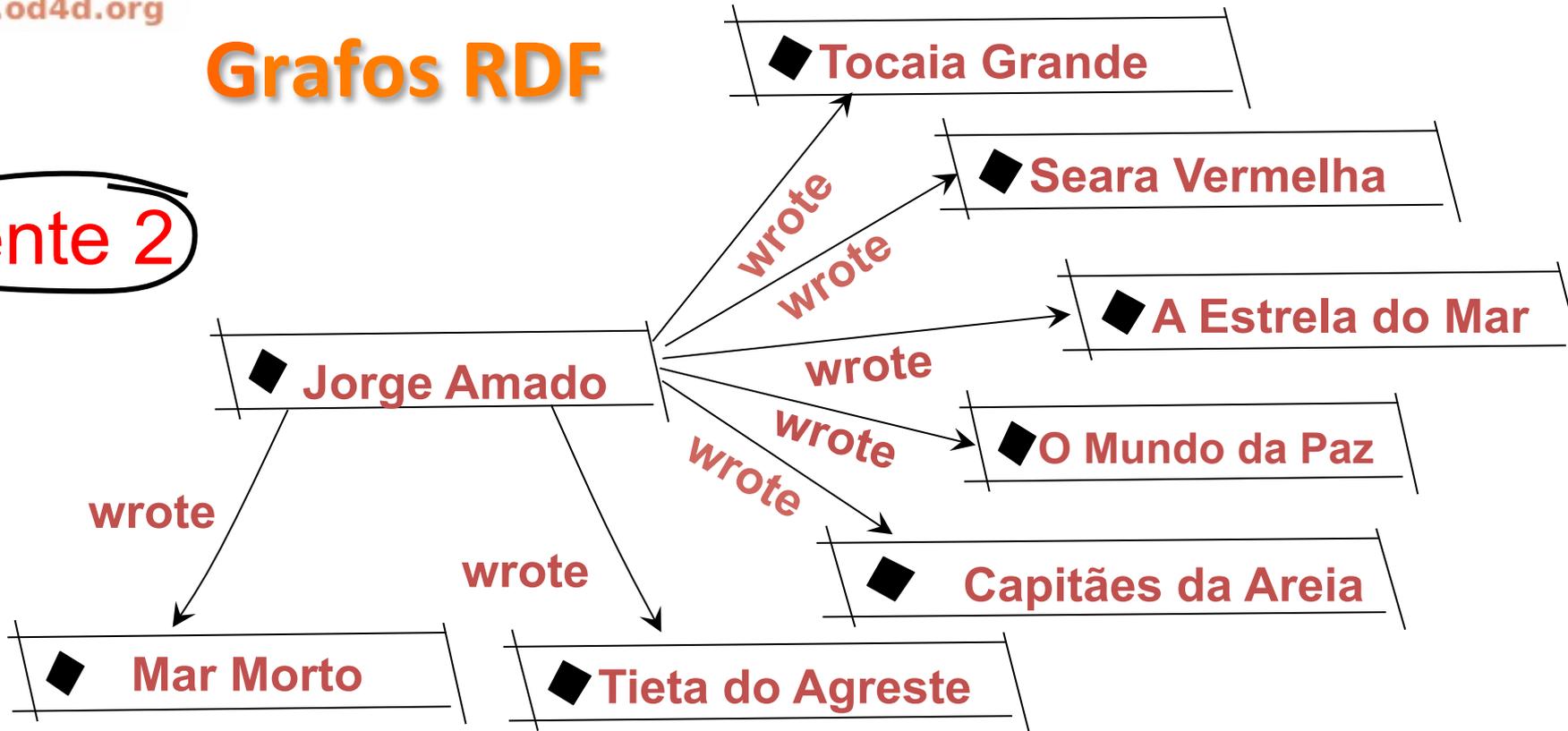
# Grafos RDF

fuelle 1



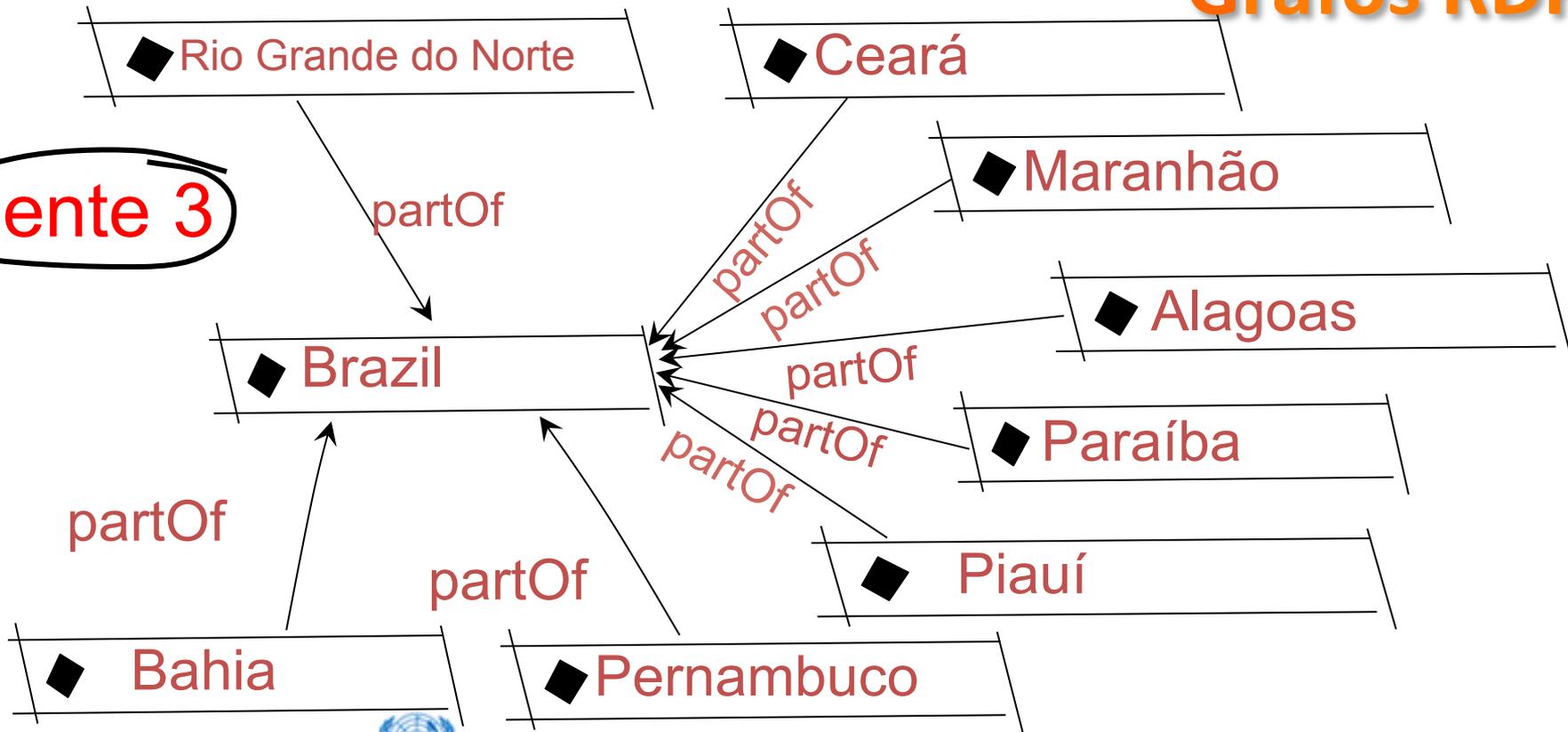
## Grafos RDF

fuelle 2



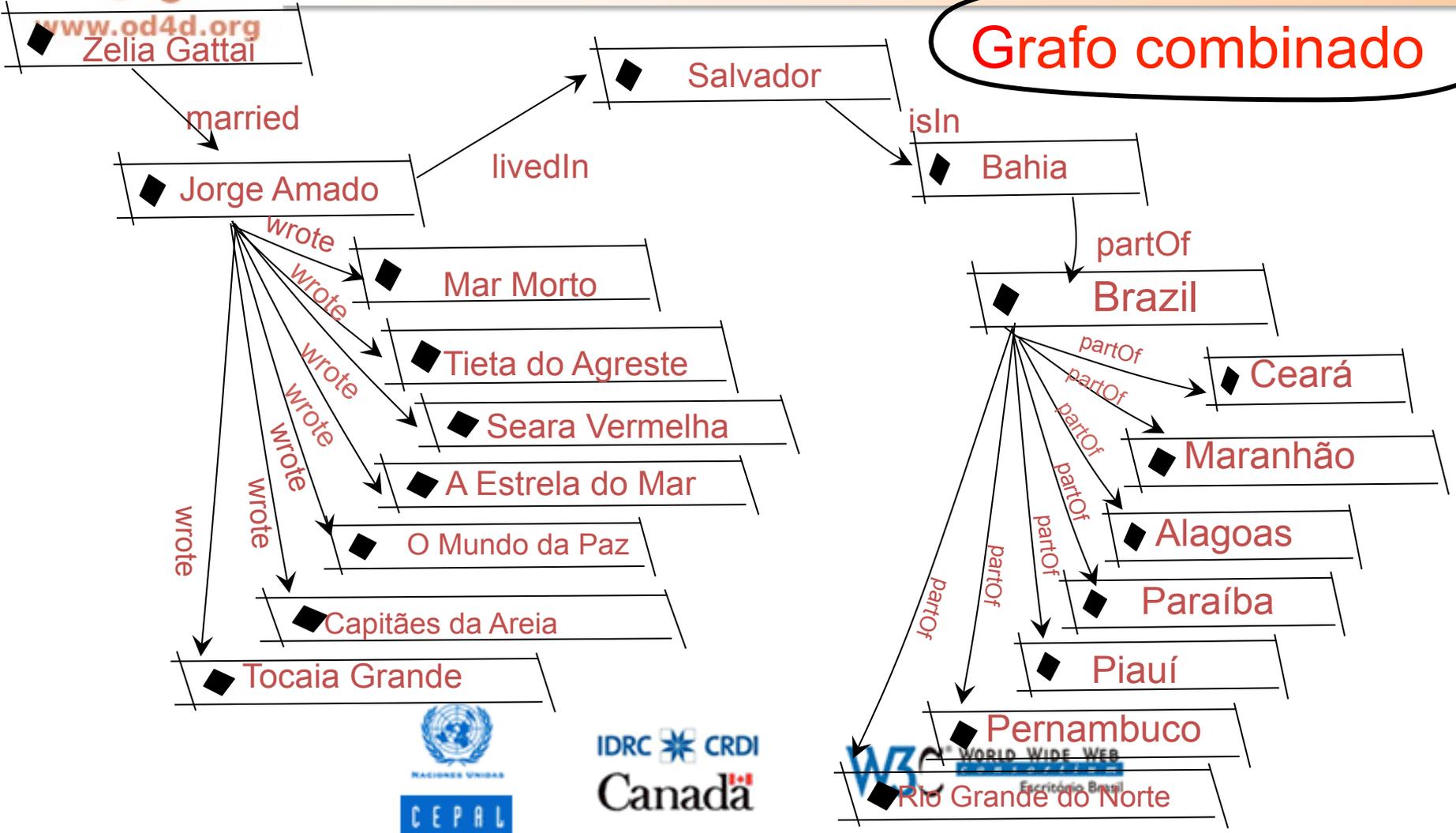
# Grafos RDF

fuelle 3

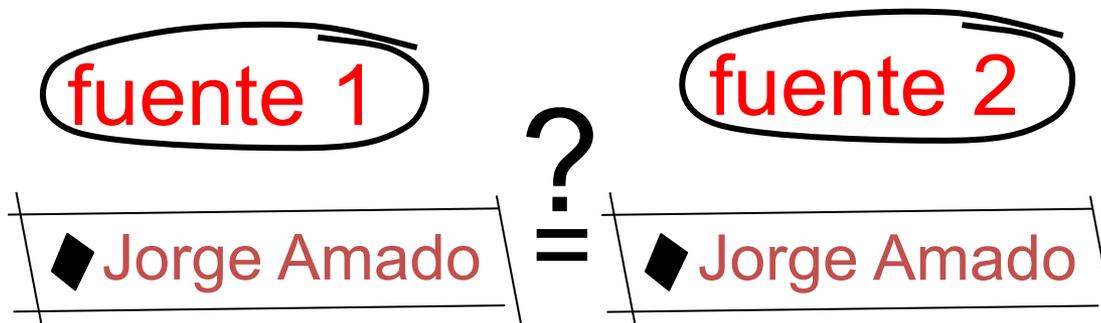




# Grafo combinado



- ¿Pero como grafos son combinados?
- ¿Como saber que los nodos en diferentes bases de datos se corresponden?



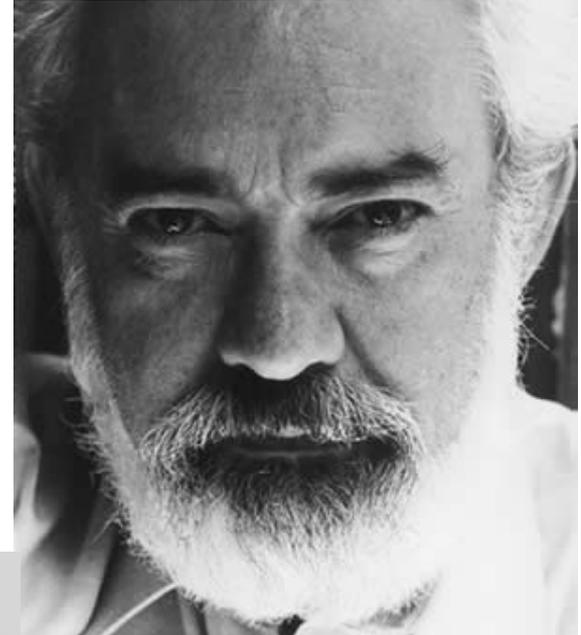
Uniform Resource Identifiers pueden ser usados para identificación única de recursos.

URI es cualquier identificador único (ex: cpf, isbn, URL).

[http://pt.dbpedia.org/resource/Fernando\\_Torres\\_\(ator\)](http://pt.dbpedia.org/resource/Fernando_Torres_(ator))

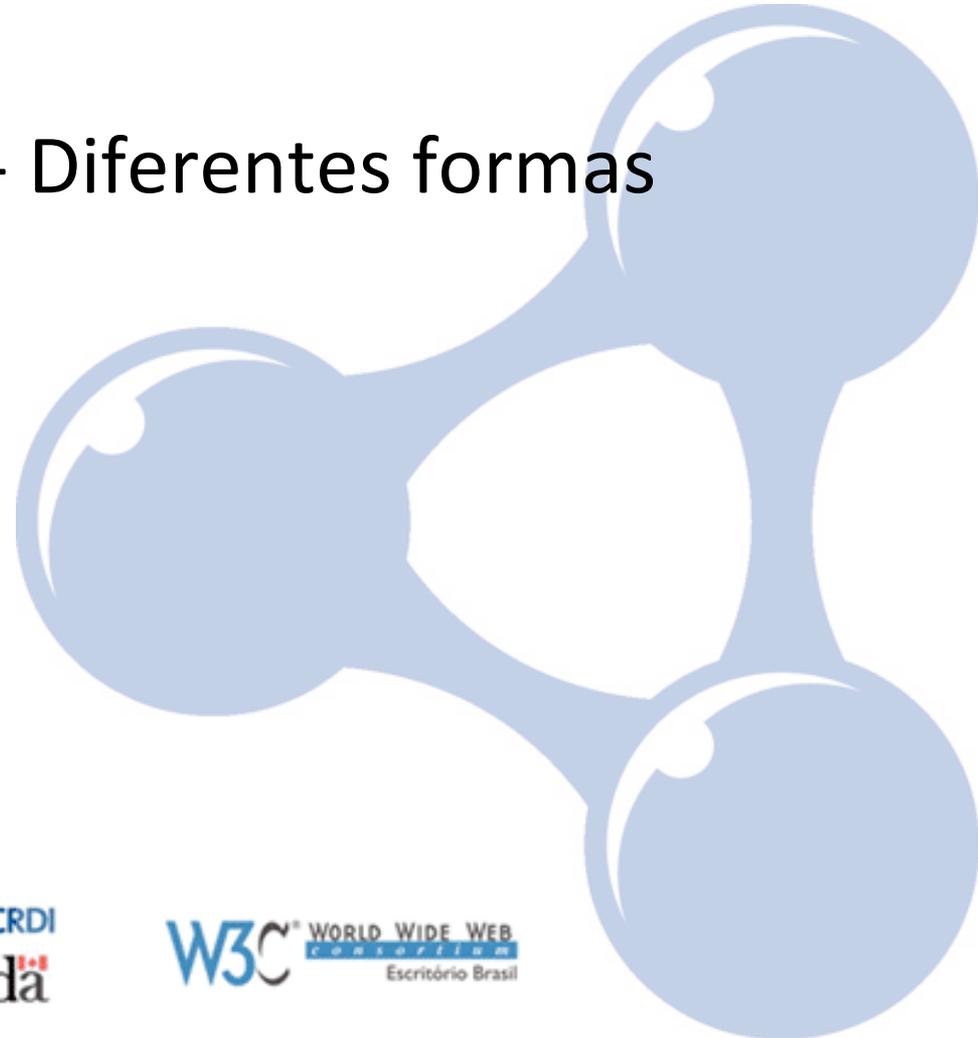


[http://pt.dbpedia.org/resource/Fernando\\_Torres](http://pt.dbpedia.org/resource/Fernando_Torres)



Por medio de las URIs cada recurso puede tener un  
identificador global y único en la Web

- Serialización de RDF - Diferentes formas
  - RDF/XML
  - RDFa
  - Turtle
  - N-Triples



# RDF/XML

```
<?xml version="1.0" encoding = "UTF-8" ?>
```

```
<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://pt.dbpedia.org/resource/Fernanda_Montenegro">
```

```
  <rdf:type rdf:resource= "http://xmlns.com/foaf/0.1/Person/">
```

```
  <foaf:name> Fernanda Montenegro </foaf:name>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

predicado

objeto

sujeto

predicado

objeto

```
<?xml version="1.0" encoding = "UTF-8" ?>
```

```
<html
```

```
  xmlns = "http://www.w3.org/1999/xhtml"
```

```
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
  xmlns:foaf = "http://xmlns.com/foaf/0.1">
```

```
<head>
```

```
  <meta http-equiv="Content-Type"
```

```
    content="application/xhtml+xml; charset=UTF-8"/>
```

```
  <title> Página da Fernanda Montenegro </title>
```

```
</head>
```

...



**RDFa**

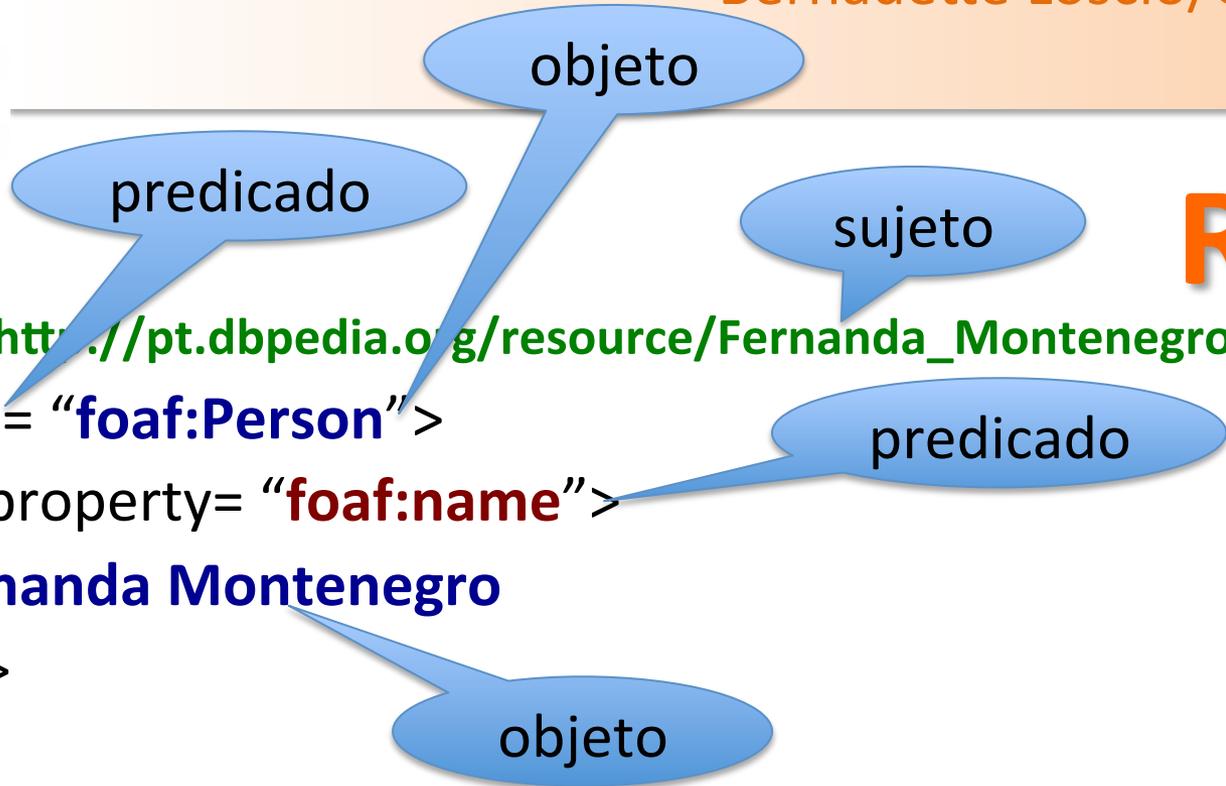


# RDFa

```

...
<body>
<div about = http://pt.dbpedia.org/resource/Fernanda\_Montenegro
  typeof = "foaf:Person">
    <span property= "foaf:name">
      Fernanda Montenegro
    </span>
  </div>
</body>
</html>

```



# Turtle

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

@prefix foaf: <http://xmlns.com/foaf/0.1>

predicado

sujeto

<[http://pt.dbpedia.org/resource/Fernanda\\_Montenegro](http://pt.dbpedia.org/resource/Fernanda_Montenegro)>

**rdf:type foaf:Person;**

**foaf:name "Fernanda Montenegro".**

objeto

# N-triples

<http://pt.dbpedia.org/resource/Fernanda\_Montenegro>

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

sujeto

<http://xmlns.com/foaf/0.1/Person/>

predicado

<http://pt.dbpedia.org/resource/Fernanda\_Montenegro>

objeto

<http://xmlns.com/foaf/0.1/name/>

“Fernanda Montenegro”.

- Serialización de RDF - Cuando usar?
  - RDF/XML: formato ya conocido y con vasta gama de herramientas
  - RDFa: cuando existe a restricción de publicación en html
  - Turtle: formato más fácil para lectura y escrita
  - N-Triples: para procesamiento de grandes volumen de datos

- Vocabularios
  - RDF proporciona medios para describir recursos usando triplas  $\langle s, p, o \rangle$
  - No proporciona medios para describir clases y sus relaciones
  - Son necesarios taxonomías, vocabularios y ontologías

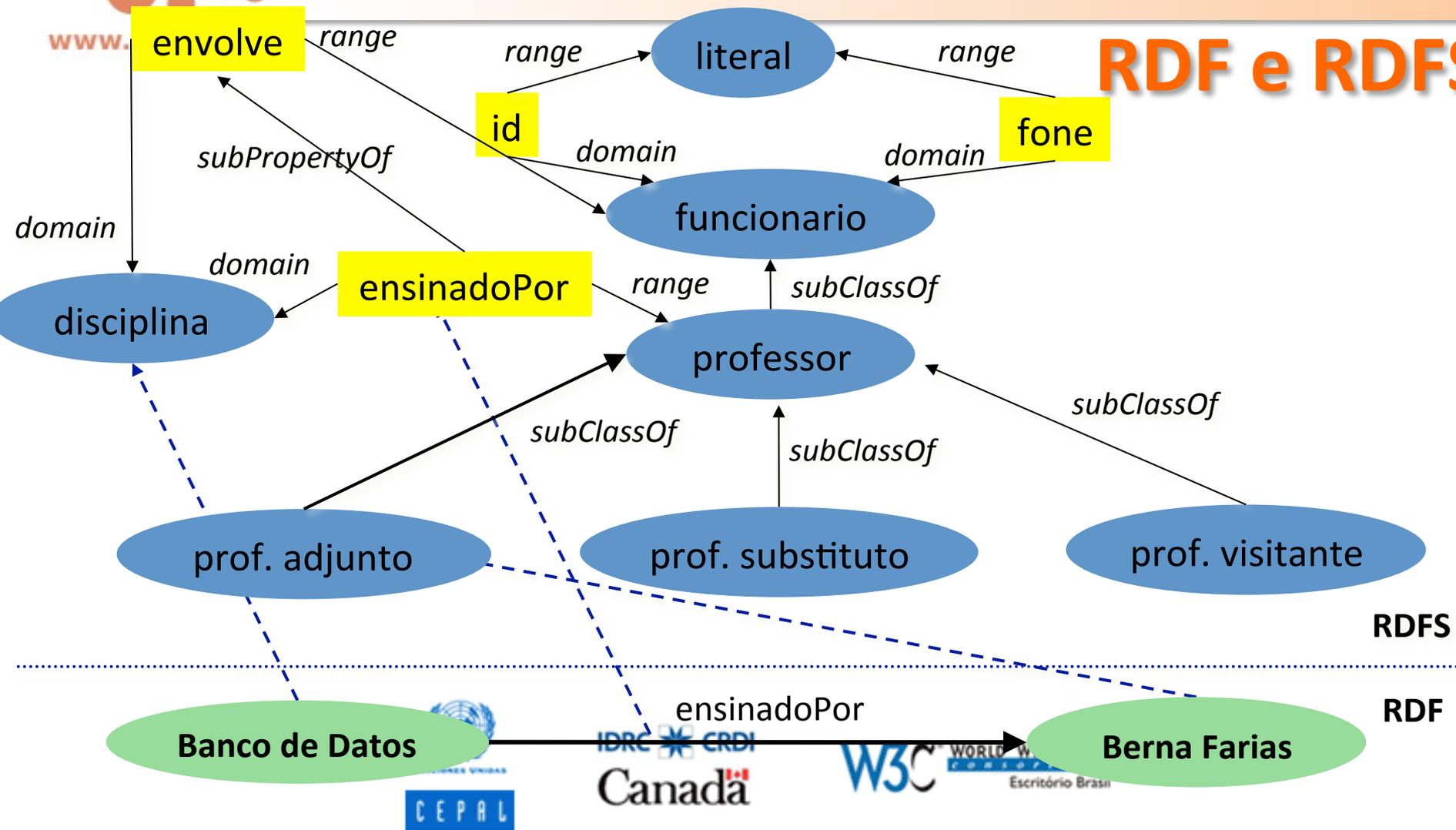
- Vocabularios
  - SKOS es un vocabulario para describir jerarquías conceptuales (taxonomías)
  - RDF y OWL ofrecen un vocabulario para describir modelos conceptuales en términos de clases y sus propiedades
  - RDFS e OWL permiten la inferencia de relacionamientos implícitos

- RDFS
  - Utiliza-se el RDFS en conjunto con el RDF
  - Lenguaje para describir vocabularios
  - Principales conceptos
    - Clases
    - Propiedades
    - jerarquía de clases
    - jerarquía de propiedades



www.

# RDF e RDFS



RDFS

RDF

– RDF

- modelo de datos para describir recursos da Web
- complementa XML ofreciendo significado para o añamiento de tags

– RDFS

- lenguaje para describir vocabularios para dominios específicos
- lenguaje primitiva para descripción de ontologías

## RDFS – Definición de clase

```
<rdf:RDF
```

```
  xmlns:rdf = "http://www.w3.org/1999/02/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#" >
```

```
<rdfs:Class rdf:ID="funcionario" >
```

```
  <rdfs:comment> Classe de todos os funcionarios</rdfs:comment>
```

```
</rdfs:Class>
```

...

## RDFS – Definición de subclase

```
...  
<rdfs:Class rdf:ID="professor">  
  <rdfs:comment>  
    Classe de todos os professores  
  </rdfs:comment>  
  <rdfs:subClassOf rdf:resource="#funcionario"/>  
</rdfs:Class>
```

```
<rdfs:Class rdf:ID="disciplina">  
  <rdfs:comment> Classe de todas as disciplinas</rdfs:comment>  
</rdfs:Class>
```

...

## RDFS – Definición de propiedad

```
...  
<rdf:Property rdf:ID="envolve">  
  <rdfs:comment> Relaciona disciplinas a  
    funcionarios  
  </rdfs:comment>  
  <rdfs:domain rdf:resource="#disciplina"/>  
  <rdfs:range rdf:resource="#funcionario"/>  
</rdf:Property/>  
...
```

... RDFS – Definición de subpropiedade

```
<rdf:Property rdf:ID="ensinadoPor">  
  <rdfs:comment> Herda domain e o range da propriedade  
    envolve  
  </rdfs:comment>  
  <rdfs:subPropertyOf rdf:resource="#envolve"/>  
</rdf:Property/>
```

...  
</rdf:RDF>

- RDFS muy limitada para describir detalles dos recursos
- No permite especificar
  - restricciones de cardinalidad
  - propiedades transitivas, inversas ou simétricas
  - disyunción de clases

# Conclusiones