

Ontologias (II)

Linguagens de Anotação de Documentos

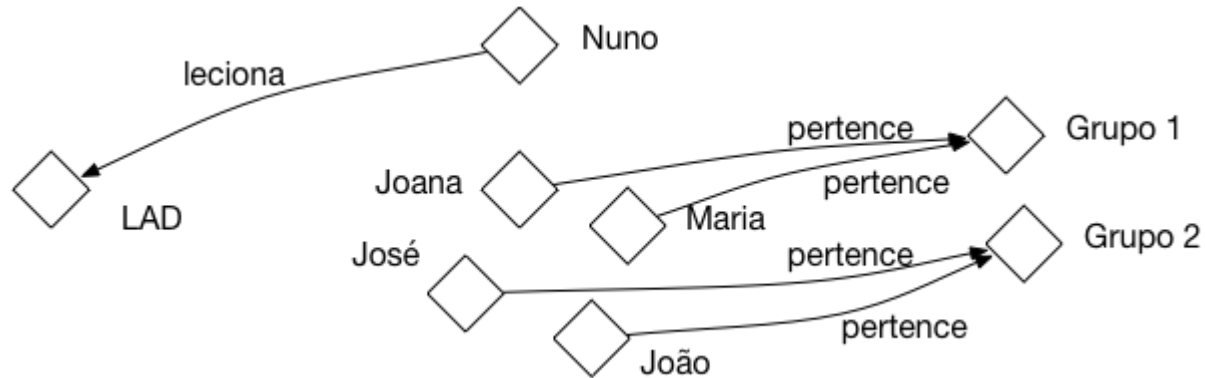
Ontologias

- Uma ontologia é um modelo conceptual da informação relevante para um domínio
- Permite criar sistemas de classificação para atribuir semântica adicional aos recursos
- O conceito tem vindo a ganhar popularidade na *web* visto facilitar a partilha de informação entre vários serviços de modo consistente
- Para isso são necessárias tecnologias que permitam codificar ontologias para que os serviços as possam interpretar

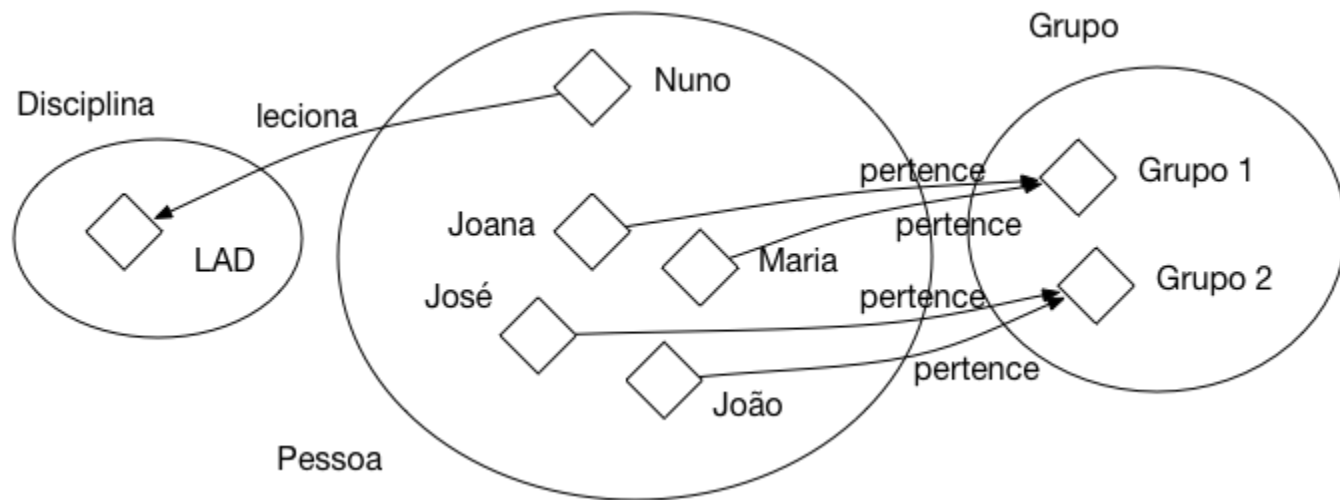
Ontologias: Indivíduos



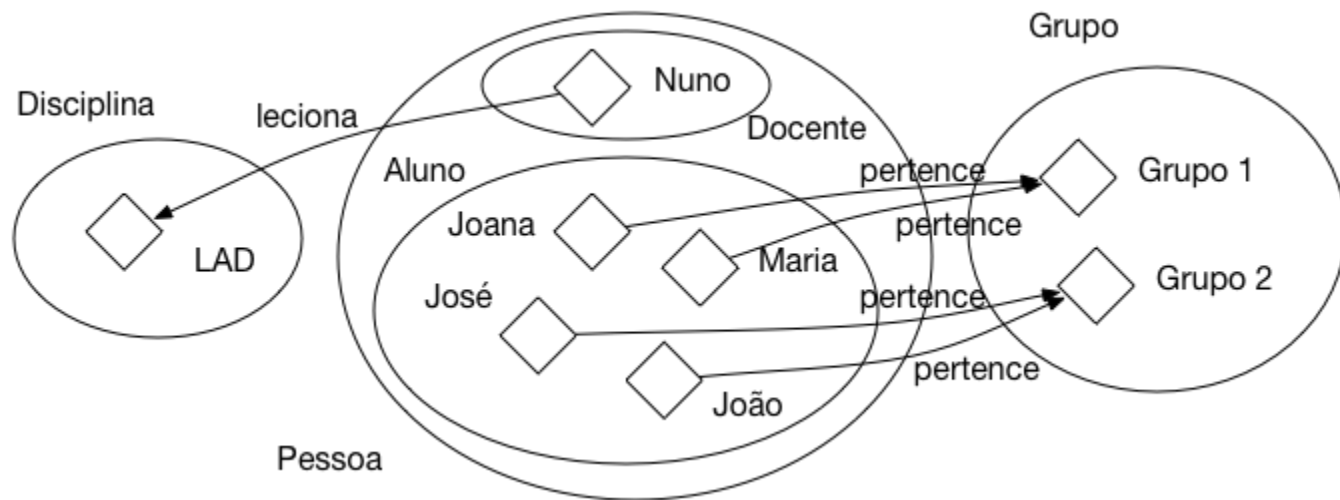
Ontologias: Propriedades



Ontologias: Classes



Ontologias: Axiomas



Web Semântica

- A criação de ontologias necessita de:
 - Identificadores únicos para os recursos (*web*, por isso URIs)
 - Um sistema para expressar os conceitos (RDFs, geralmente codificado em XML)
 - Um vocabulário com semântica associada (RDFS e OWL)

RDF

- Disponibiliza os construtores do grafo
- Afirmações (*statements*) através de triplos sujeito - predicado - objeto
- Os três elementos são recursos unicamente identificados
- Várias syntaxes
 - XML facilmente processável pelas máquinas,
 - Outras (N-triples, Turtle) mais legíveis para humanos

N-triples

- O indivíduo *José* pertence ao *Grupo 1*

```
@prefix lad: <http://www.lad.pt/grupos#>  
lad:José lad:pertence lad:Grupo1 .
```

RDF XML

- O indivíduo *José* pertence ao *Grupo 1*

```
<rdf:RDF xmlns:grupos="http://lad.pt/grupos"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description about="http://lad.pt/Jose">
    <lad:pertence resource="http://lad.pt/Grupo1">
  </rdf:Description>
</rdf:RDF>
```

N-triples

- O indivíduo *José* pertence ao *Grupo 1*
- O indivíduo *José* tem número *99999*

```
@prefix lad: <http://www.lad.pt/lad#>  
lad:José lad:pertence lad:Grupo1 .  
lad:José lad:numero "99999" .
```

RDF XML

- O indivíduo *José* pertence ao *Grupo 1*
- O indivíduo *José* tem número *99999*

```
<rdf:RDF xmlns:lad="http://lad.pt/lad"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description about="http://lad.pt/Jose">
    <lad:pertence resource="http://lad.pt/Grupo1">
      <lad:numero>99999</lad:numero>
    </rdf:Description>
  </rdf:RDF>
```

RDF Schema

- O RDF especifica como construir triplos mas estes não têm significado / semântica associada
- O RDF Schema (RDFS) disponibiliza um vocabulário para descrever a taxonomia básica da ontologia
- Introduz elementos predefinidos que têm semântica associada
- Permite identificar o tipo dos recursos e usar propriedades para os descrever

RDFS: Classes

- Introduz a noção de *classe* e propriedades para as relacionar
 - `rdf:type`: define o tipo ou classe de um recurso
 - `rdfs:Class`: recursos que representam tipos, classes da nossa ontologia
 - `rdfs:subClassOf`: define que os elementos de uma classe estão todos contidos numa outra
 - `rdfs:resource`: a classe de topo, a que todos os recursos pertencem

RDFS: Classes

- *Pessoa, Aluno e Docente* são classes da nossa ontologia, por isso o seu tipo é `Class`

```
lad:Pessoa rdf:type rdfs:Class .
```

```
lad:Aluno rdf:type rdfs:Class .
```

```
lad:Docente rdf:type rdfs:Class .
```

RDFS: Classes

- Pessoas concretas (indivíduos) pertencem às classes definidas em cima

```
lad:Jose rdf:type lad:Aluno .
```

```
lad:Joana rdf:type lad:Aluno .
```

```
lad:Nuno rdf:type lad:Docente .
```

- Se algo é o tipo de um recurso, o sistema assume implicitamente que é uma classe

RDFS: Classes

- *Aluno* e *Docente* são sub-classes de *Pessoa*, por isso estão relacionadas pela propriedade `subClassOf`

```
lad:Aluno rdfs:subClassOf lad:Pessoa .  
lad:Docente rdfs:subClassOf lad:Pessoa .
```

- Todos os indivíduos da sub-classe pertencem à super-classe (e de modo transitivo)
- O sistema assume que *Jose* é uma *Pessoa*
- Cada classe pode ser sub-classe de várias classes

RDFS: Classes

- No topo da hierarquia das classes está o `rdfs:Resource`
 - Todas os recursos são indivíduos de `Resource`
 - Todas as classes são sub-classes de `Resource`

RDFS: Propriedades

- Introduz propriedades predefinidas que permitem caracterizar as propriedades da ontologia
 - `rdf:Property`: a classe de todas as propriedades
 - `rdfs:domain`: define os elementos de origem das propriedades
 - `rdfs:range`: define os elementos de destino das propriedades
 - `rdfs:subPropertyOf`: define uma propriedade com sub-tipo de outra

RDFS: Propriedades

- A propriedade *pertence* atribui um grupo a um aluno, é do tipo `Property`

```
lad:pertence rdf:type rdf:Property .  
lad:Jose lad:pertence lad:Grupo1 .
```

- O sistema assume que todos os predicados dos triplos são propriedades, por isso a primeira afirmação é desnecessária
- As propriedades são também recursos, por isso podemos definir propriedades sobre as propriedades

RDFS: Propriedades

- Podemos definir quais as classes relacionadas pelas propriedades através de `domain` e `range`

```
lad:pertence rdfs:domain rdf:Aluno .
```

```
lad:pertence rdfs:range rdf:Grupo .
```

```
lad:leciona rdfs:domain rdf:Docente .
```

```
lad:leciona rdfs:domain rdf:Disciplina .
```

- O sistema passa a assumir que os elementos relacionados pelas propriedades pertencem às classes atribuídas

RDFS: Propriedades

- Uma propriedade pode também apontar para valores concretos em vez de recursos

```
lad:numero rdfs:domain rdf:Pessoa .  
lad:numero rdfs:range rdf:Datatype .  
  
lad:Jose lad:numero "999999" .
```

RDFS: Utilidades

- Introduz propriedades predefinidas que ajudam na leitura e gestão da ontologia
 - `rdfs:label`: uma etiqueta legível para o recurso
 - `rdfs:comment`: comentários legíveis sobre o recurso
 - `rdfs:seeAlso`: um apontador entre recursos

RDFS: Limitações

- O RDFS já nos permite escrever ontologias relativamente expressivas
- No entanto ainda tem algumas limitações
 - Restrições de cardinalidade sobre as propriedades
 - Definir propriedades como transitivas
 - Definir classes combinando outras classes
 - ...
- O OWL introduz novos termos que permitem fazer isto (e muito mais)

OWL

- Várias versões do OWL, cada uma mais expressiva mas mais complexa de processar
 - OWL Lite
 - OWL DL
 - OWL Full
- O OWL DL é o mais equilibrado, expressivo mas ainda relativamente fácil de processar e interpretar

OWL DL: Classes

- A noção de classe no OWL DL é mais restrita do que no RDFS, para que sejam mais fáceis de interpretar
- Todas as classes são sub-classe de *thing*, e *nothing* é sub-classe de todas as classes

```
lad:Pessoa rdf:type owl:Class .  
lad:Pessoa rdf:subClassOf owl:Thing .
```

- Ao contrário de RDFS, um elemento ou é um indivíduo ou uma classe

OWL DL: Classes

- Em OWL podemos também definir classes à custa de outras, i. e., por *intenção* (em vez de declarar os indivíduos, i. e., por *extensão*)
 - Todos os elementos de outra classe limitados por alguma restrição
 - Definidas como a interseção, união ou complemento de outras classes
- Além de sub-classe, podem-se definir outros axiomas
 - Equivalência, as classes têm exatamente os mesmos elementos
 - Disjunção, as classes não partilham qualquer elemento

OWL DL: Classes

- Os grupos podem conter no máximo 3 alunos

```
_:temp rdf:type owl:Restriction .  
_:temp owl:onProperty lad:elementos .  
_:temp owl:maxCardinality "3" .
```

```
lad:Grupo rdfs:subClassOf _:temp .
```

- Recursos com prefixo `_` são temporários para ajudar na construção de triplos, e não são efetivamente criados na ontologia

OWL DL: Propriedades

- Em OWL DL, as propriedades são divididas entre
 - `ObjectProperty`: as que apontam para indivíduos
 - `DatatypeProperty`: as que apontam para valores literais
- Podem também ser definidas como
 - equivalentes a outras propriedades
 - o inverso de outras propriedades

OWL DL: Propriedades

- Podem ser impostos axiomas adicionais sobre as propriedades
- De cardinalidade
 - funcional (cada recurso de origem só pode estar relacionado com um de destino)
 - inversamente funcional (cada recurso de destino só pode estar relacionado com um de origem)
- Lógicas: simétrica ou transitiva

OWL DL: Classes

- Cada Aluno só pode pertencer a um Grupo, e os elementos do grupo são o seu inverso

```
lad:elementos owl:inverseOf lad:pertence .
```

```
lad:pertence rdf:type owl:FunctionalProperty .
```

OWL DL: Indivíduos

- Um indivíduo pode ser classificados como sendo
 - `sameAs`: equivalente a outro
 - `differentFrom`: diferente de outro

Take-home Lesson

- As ontologias são modelos conceptuais construídos à volta dos conceitos de indivíduos, classes, propriedades e axiomas
- Formam um grafo, definido à custa de triplos sujeito - predicado - objeto
- O RDF é usado para definir esses triplos, mas sem semântica associada
- Tecnologias adicionais (RDFS e OWL) definem recursos com semântica predefinida, que permitem descrever ontologias expressivas