



# Creación de Ontologías mediante OWL

## Taller práctico

Jose Emilio Labra Gayo  
Departamento de Informática  
Universidad de Oviedo

## Persona

Declarar una clase Persona y varias  
instancias

```
Persona  $\sqsubseteq$  owl:Thing  
Persona(federico)  
Persona(juan)
```

Declarar la clase Asignatura y varias  
instancias

```
Asignatura  $\sqsubseteq$  owl:Thing  
Asignatura(logica)  
Asignatura(algebra)  
Asignatura(programacion)
```

## Profesor

- Declarar las clases Profesor y Alumno como subclases de Persona

```
Profesor  $\subseteq$  Persona  
Alumno  $\subseteq$  Persona
```

```
Profesor(labra)  
Profesor(dani)
```

```
Alumno(miguel)  
Alumno(ruben)
```

## Propiedades de Objetos

- Declarar la propiedad “daClaseDe”

```
daClaseDe  $\subseteq$  owl:ObjectProperty  
Domain(daClaseDe, Profesor)  
Range(daClaseDe, Asignatura)
```

```
daClaseDe(labra, logica)  
daClaseDe(dani, logica)
```

- Declarar la propiedad “estaMatriculadoEn”

```
estaMatriculadoEn  $\subseteq$  owl:ObjectProperty  
Domain(estaMatriculadoEn, Alumno)  
Range(estaMatriculadoEn, Asignatura)
```

```
estaMatriculadoEn(miguel, logica)  
estaMatriculadoEn(ruben, logica)
```

## Propiedades de datos

- Declarar las propiedades “nombre”, “apellidos”, “edad”

```
nombre : Persona → xsd:string  
edad : Persona → xsd:integer
```

```
edad(labra,38)  
edad(dani,30)  
edad(miguel,23)  
edad(ruben,24)  
nombre(labra,"Jose Emilio")  
apellidos(labra,"Labra Gayo")  
...
```

## Padre y RecienNacido

```
Padre ≡ Persona ∧ ∃ tieneHijo Persona
```

```
 $\forall x(\text{Padre}(x) \leftrightarrow (\text{Persona}(x) \wedge \exists y(\text{tieneHijo}(x,y) \wedge \text{Persona}(y))))$ 
```

```
tieneHijo(Jose,Sergio)  
Persona(Jose)
```

```
ReciénNacido ⊆ Persona  
ReciénNacido(Sergio)
```

# Orgullosos

- Definir:

$\text{Orgullosos} \equiv \text{Persona} \cap \exists \text{ tieneHijo ReciénNacido}$

Clasificar ontologías:

$\text{Orgullosos} \subseteq \text{Padre}$

Buscar orgullosos:

$\text{Orgullosos}(\text{jose})$

–

## Recién nacido orgullosos?

- Añadir:

$\text{Padre} \subseteq \neg \text{ReciénNacido}$   
 $\text{RecienNacidoOrgullosos} \subseteq \text{Orgullosos}$   
 $\text{RecienNacidoOrgullosos} \subseteq \text{ReciénNacido}$

$\text{RecienNacidoOrgullosos} = \text{Insatisfacible}$

$\text{RecienNacidoOrgullosos}(\text{kiko})$

Kiko = inconsistente

## Ejecución en otros entornos

Guardar en fichero ejemplo.owl

Utilizar Protègè + Pellet-DIG

Línea de comandos

Cargar y mostrar información de la ontología

`pellet --inputFile ejemplo.owl`

Realizar una consulta SPARQL a una ontología

`pellet --inputFile ejemplo.owl --queryFile consulta.sq`

## Propiedades inversas

Añadir y comprobar las inferencias

```
esInversa(tieneHijo, esHijoDe)
Profesor(Jose)
HijodeProfesor  $\equiv \exists$  esHijoDe Profesor
```

## Subpropiedades y transitividad

- Añadir:

tieneHijo  $\subseteq$  tieneDescendiente  
tieneDescendiente: Transitiva  
tieneHijo(Emilio,Jose)  
AntepasadoOrguloso  $\equiv$  Persona  $\cap \exists$  tieneDescendiente RecienNacido

## Propiedades Simétricas

- viveCon

viveCon : Simétrica, Transitiva  
viveCon(Jose,Sergio)  
viveCon(Jose,Ana)

## Propiedades funcionales

- Añadir

```
tienePadre : Functional Property  
tienePadre  $\subseteq$  esHijoDe  
Persona(Pepe)
```

```
tienePadre(Nuria,Jose)  
tienePadre(Nuria,Pepe)
```

## Propiedades func. inversas

- Añadir

```
esMadreDe : InverseFunctional Property  
esMadreDe(Ana,Sergio)  
esMadreDe(Maria,Sergio)
```

# Asignaturas

- Añadir

```
estaMatriculado(Juan,Algebra)
estaMatriculado(Juan,Calculo)
estaMatriculado(Juan,Programacion)

estaMatriculado(Luis,Algebra)
estaMatriculado(Luis,Calculo)

estaMatriculado(Carlos,Programacion)

tieneAlumno : Inversa (estaMatriculado)
```

# Alumno

- Añadir

```
Alumno =  $\exists$  estaMatriculado Asignatura
Disjuntos(Profesor,Alumno)
```



# Cardinalidades

- Añadir

MuyOcupado  $\equiv$  Alumno  $\cap$  estaMatriculado  $\geq 3$   
PocoOcupado  $\equiv$  Alumno  $\cap$  estaMatriculado  $\leq 3$

# Distintos

- Añadir

Programacion  $\neq$  Algebra, Programacion  $\neq$  Calculo, Calculo  $\neq$  Algebra

Ahora infiere que Juan está muy ocupado  
Pero...no infiere que Luis y Carlos estén poco ocupados...  
tampoco infiere nada sobre Jose...

Motivo: Con la información que tiene, sabe que están poco ocupados,  
PERO con la *Hipótesis de Mundo Abierto*, podría encontrar más  
Información en el futuro que invalidase dicha inferencia...

## AsignaturaLimitada

- Añadir

AsignaturaLimitada : tieneAlumno  $\leq$  2

AsignaturaLimitada(Algebra)

AsignaturaLimitada(Calculo)

Juan  $\neq$  Carlos  $\neq$  Luis

## Tipo de Asignatura

- Añadir

AsignaturaInformatica  $\subseteq$  Asignatura

AsignaturaMatematica  $\subseteq$  Asignatura

AsignaturaMatematica(Algebra)

AsignaturaMatematica(Calculo)

AsignaturaInformatica(Programacion)

## Cuantificador Universal

- Añadir:

$\text{SoloCursaInformaticas} \equiv \text{Alumno} \cap \forall \text{ estaMatriculado AsignaturaInformatica}$

$\text{SoloCursaMatematicas} \equiv \text{Alumno} \cap \forall \text{ estaMatriculado AsignaturaMatematicas}$

## Conjuntos mediante Enumeraciones

- Definir:

$\text{AsignaturasFaciles} \equiv \{ \text{Calculo}, \text{Programacion} \}$

$\text{AlumnoContento} \equiv \text{Alumno} \cap \exists \text{ estaMatriculado AsignaturasFaciles}$

## Propiedades de tipos de datos

- Añadir

```
nombre(Juan,"Juan Gomez")
nombre(Luis,"Luis Lopez")
nombre(Carlos, "Carlos Alvarez")

edad(Juan,23)
edad(Luis,24)
edad(Carlos,20)
```

Consultas SPARQL con tipos de datos...

## Lugares

- Añadir información sobre lugares de nacimiento
  - Ciudades
  - Regiones
  - Países
  - Propiedades: nacidoEn, estaEnPais

## Instituciones

- Añadir información sobre instituciones educativas
  - Universidades
    - Campus
    - Centros, Departamentos
    - Cargos de gobierno: rector, decanos, etc

## Importar ontologías

- Importar otra ontología

Ejemplo:

[http://www.geonames.org/ontology/ontology\\_v2.0\\_Lite.rdf](http://www.geonames.org/ontology/ontology_v2.0_Lite.rdf)

- Enlazar diferentes ontologías

Ejemplo:

nacidoEn subPropertyOf <http://www.geonames.org/ontology/inCountry>

- Comprobar inferencias

## Propiedades aridad-n

Añadir:

Juan tiene un 5.3 en Algebra  
Luis tiene un 7 en Fisica  
etc....

## Clases como Individuos

- Declarar:

Propiedad: trataDe  
trataDe(AsignaturaInformática,DesarrolloSoftware)

# Hermano y Tios

- Declarar:

esHermanoDe: Persona  $\rightarrow$  Persona

esHermanoDe(Manuel,Jose)

esTioDe: Persona  $\rightarrow$  Persona

¿Cómo declarar lo siguiente?

$\forall x \forall y \forall z (\text{hermanoDe}(x,y) \wedge \text{tieneHijo}(y,z) \rightarrow \text{esTioDe}(x,z))$