

Ontologías

Motivación

- El agente cometió un error



¿Cómo lo interpreto?



Problemas con el manejo de la información

- Inconsistencia
- Incompatibilidad
- Falta de completitud
- Falta de límites



Web semántica versus Web actual

- Web actual representa la información utilizando documentos en lenguaje natural con poca estructura
 - Fácil comprensión por humanos (html sólo define presentación).
 - Difícil soportar el procesamiento automático.

Web semántica versus Web actual

- Alternativas para facilitar el procesamiento de la información en la web
 - Máquinas más inteligentes comprender el significado de la información que hay en la web
 - Procesamiento de lenguaje natural, reconocimiento de imágenes, etc.
 - Información más inteligente -> representar la información de modo que sea sencilla de comprender a las máquinas
 - Expresar contenidos en un formato procesable automáticamente.
 - Uso de metainformación (metadatos = datos sobre los datos)

¿Por qué ontologías?

- Hay un incremento en la necesidad de:
 - Capturar el conocimiento.
 - Definir vocabulario común (consensuado).
 - Compartir entendimiento común (reutilizable).
 - Interpretación y manipulación automática.



ONTOLOGÍAS

Orígenes de las ontologías

- En Filosofía
 - **Ontología.** Parte de la metafísica que trata del ser en general y de sus propiedades trascendentales.
- En Ciencias de la Computación
 - Algo existe si puede ser representado, descripto, definido (formalmente) para ser interpretado por una máquina.

Definición

« Una especificación de una conceptualización... una descripción de los conceptos y relaciones que pueden existir para un agente o una comunidad de agentes.»

T.R. Gruber

«Una ontología es un catalogo de los tipos de cosas que, se asume, existen en un dominio de interés D desde la perspectiva de una persona, la cual usa un lenguaje L para hablar sobre D.»

John F. Sowa

« Un ontología trata sobre la exacta descripción de las cosas y sus relaciones. »

World Wide Web Consortium (W3C)

Componentes de una ontología

- Una ontología está compuesta por:
 - Conceptos o clases
 - Instancias o individuos
 - Propiedades o relaciones
 - Axiomas

Ejemplo

Representa mediante una **Ontología** la siguiente información acerca de la organización de un Congreso:

-En dicho Congreso se debe poder almacenar información acerca de las presentaciones que se van a realizar que serán bien artículos aceptados, conferencias invitadas o posters. De cada una de estas presentaciones se desea conocer su título, numero de referencia, autor/es, su lista de descriptores y si está confirmada su presentación en el Congreso.

-Se desea también almacenar información de los diferentes autores con datos como nombre, apellidos, universidad o centro donde trabajan y numero de artículos presentados.

-Por otro lado se debe mantener una lista de las personas inscritas, indicando su nombre, cantidad abonada, numero de tarjeta de crédito y si es estudiante o no. En el caso de ser estudiante se deberá guardar información acerca de la universidad donde está estudiando.

-Se quiere disponer de una estructura que refleje las sesiones del Congreso por días. El Congreso dura 3 dias (Miércoles, Jueves y Viernes) y hay 3 sesiones diarias (MAÑANA1, MAÑANA2 y TARDE1) donde en cada sesión puede haber o bien 3 artículos o 1 conferencia invitada o un número indeterminado de posters (no puede haber mezclas de presentaciones diferentes)

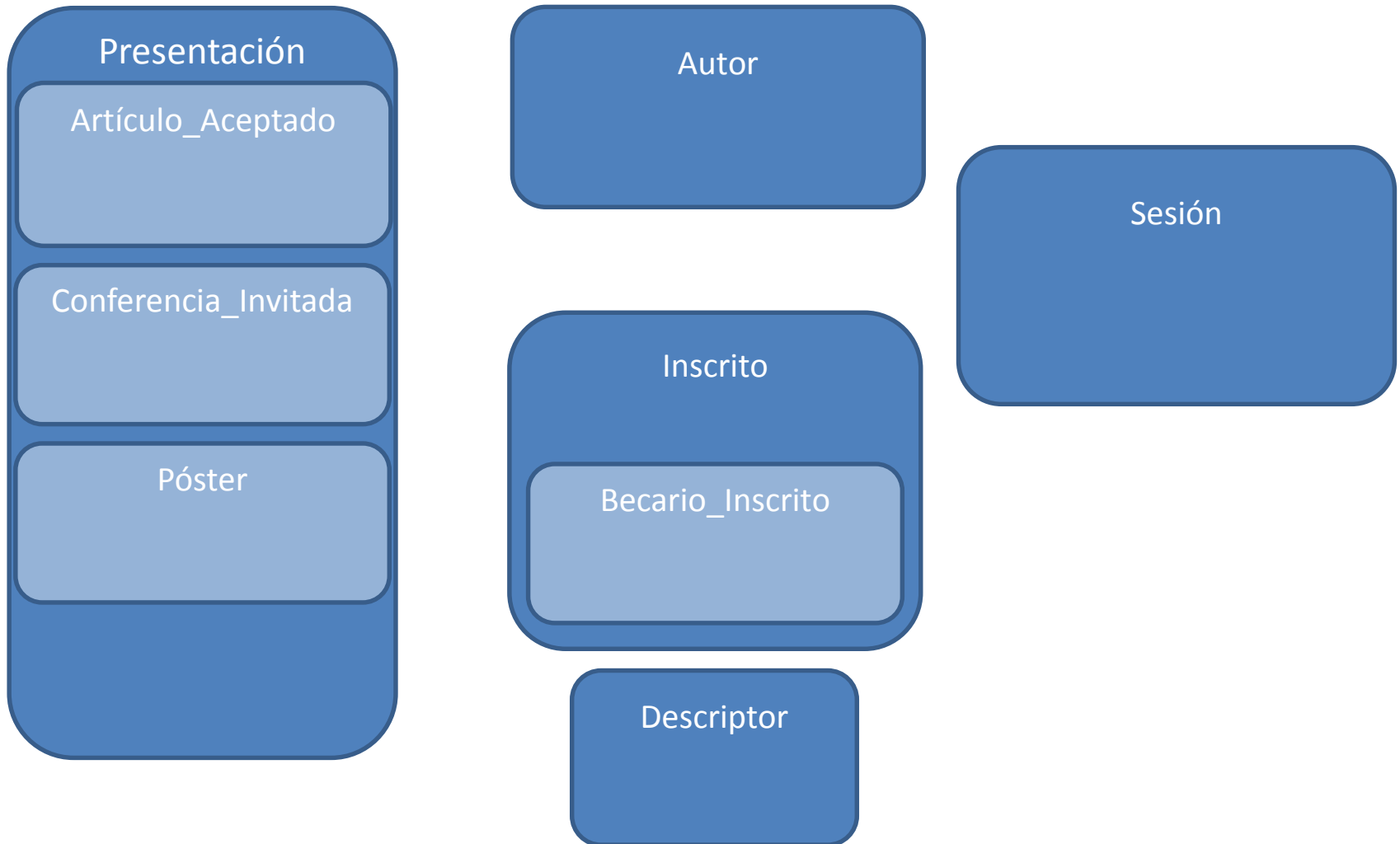
Cada uno de los descriptores del Congreso debe asociarse a una descripción del mismo que explique el significado del descriptor.

Conceptos

- Colección de individuos (p.e. persona, árbol ..)

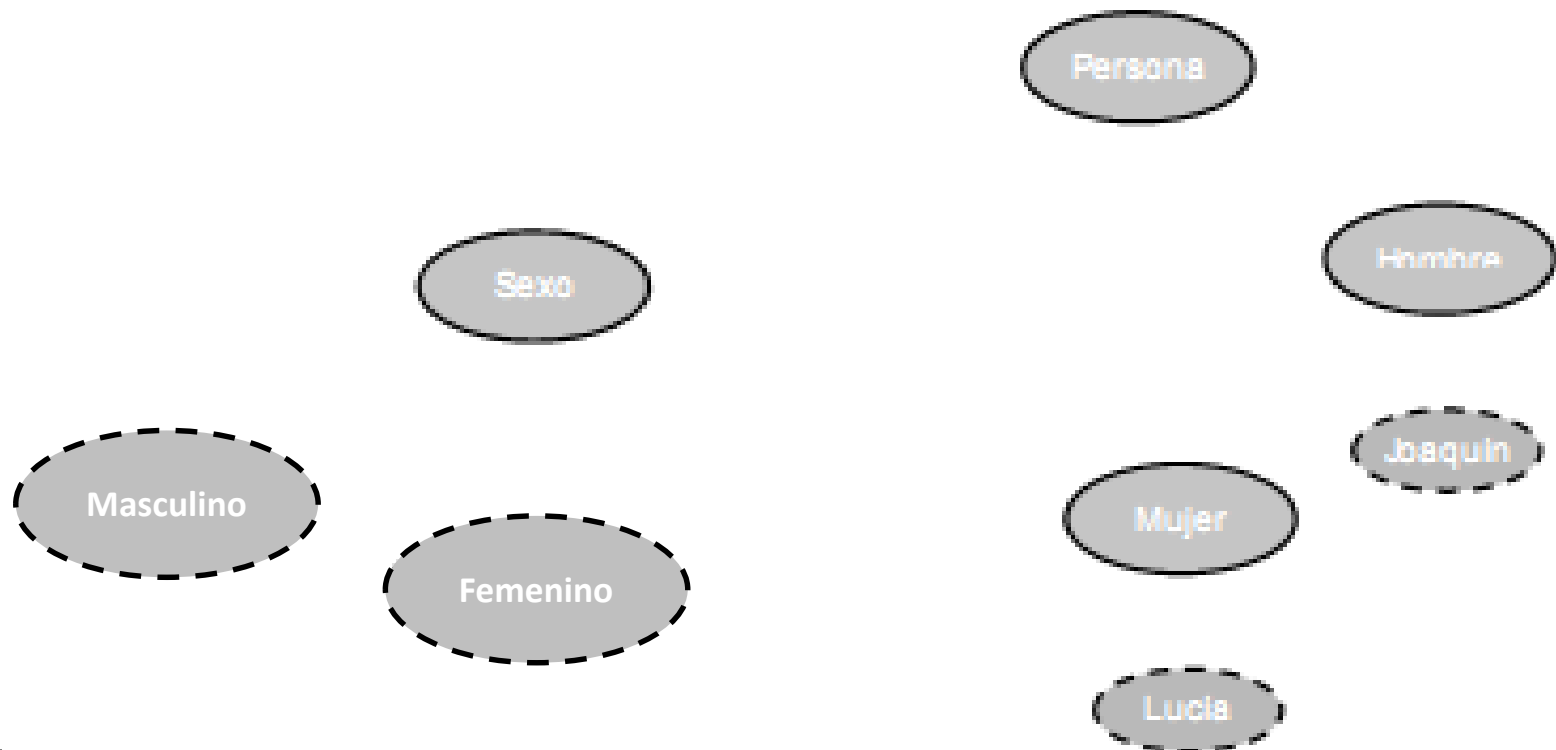


Conceptos o clases: conjuntos de objetos del dominio

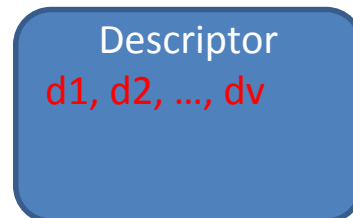
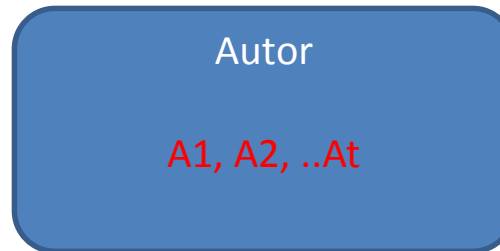
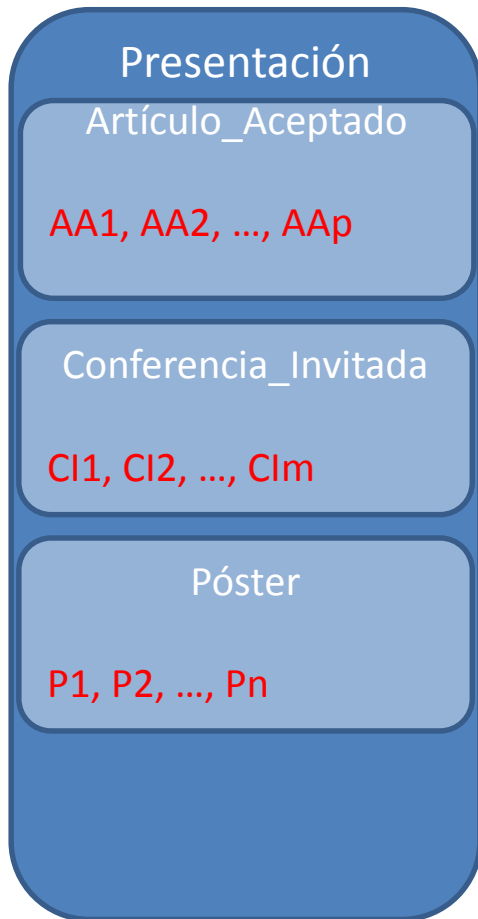


Instancias

- Objetos en el mundo (p.e. Joaquín, Lucía)
- Valores (un integer, un string, etc..)

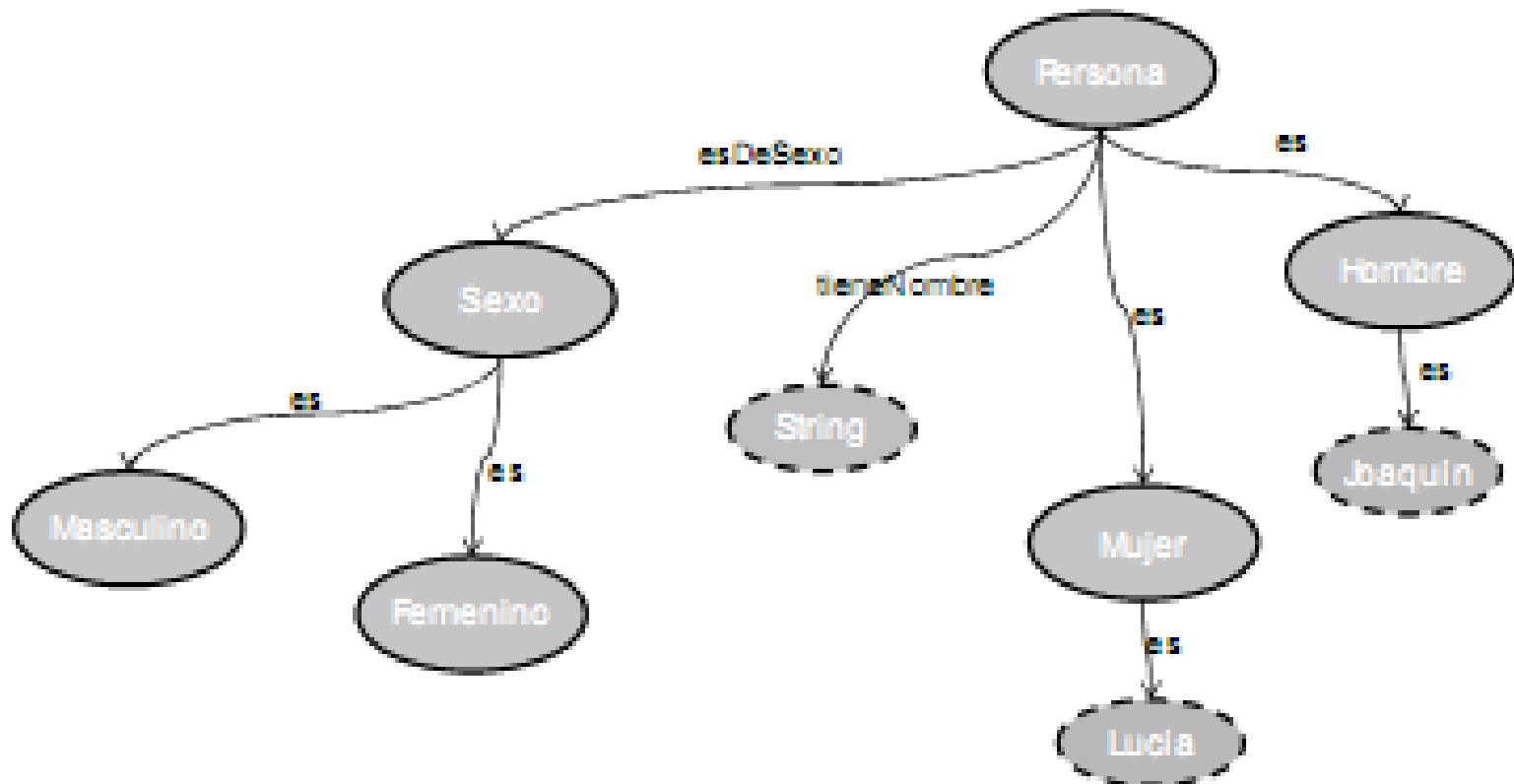


Individuos

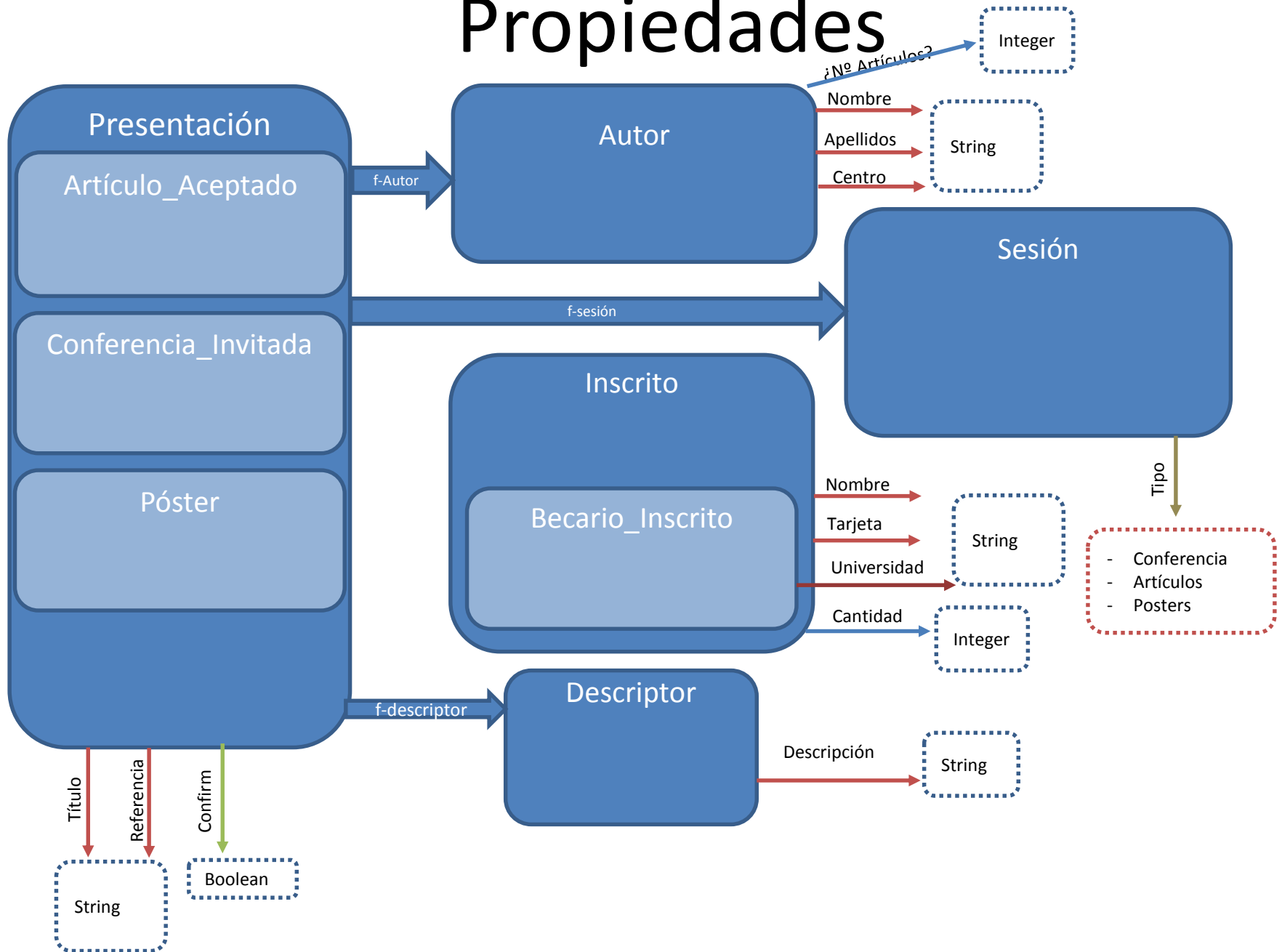


Propiedades

- Describen las relaciones entre los conceptos
- Tipos de propiedades: a) de objeto, b) de datos

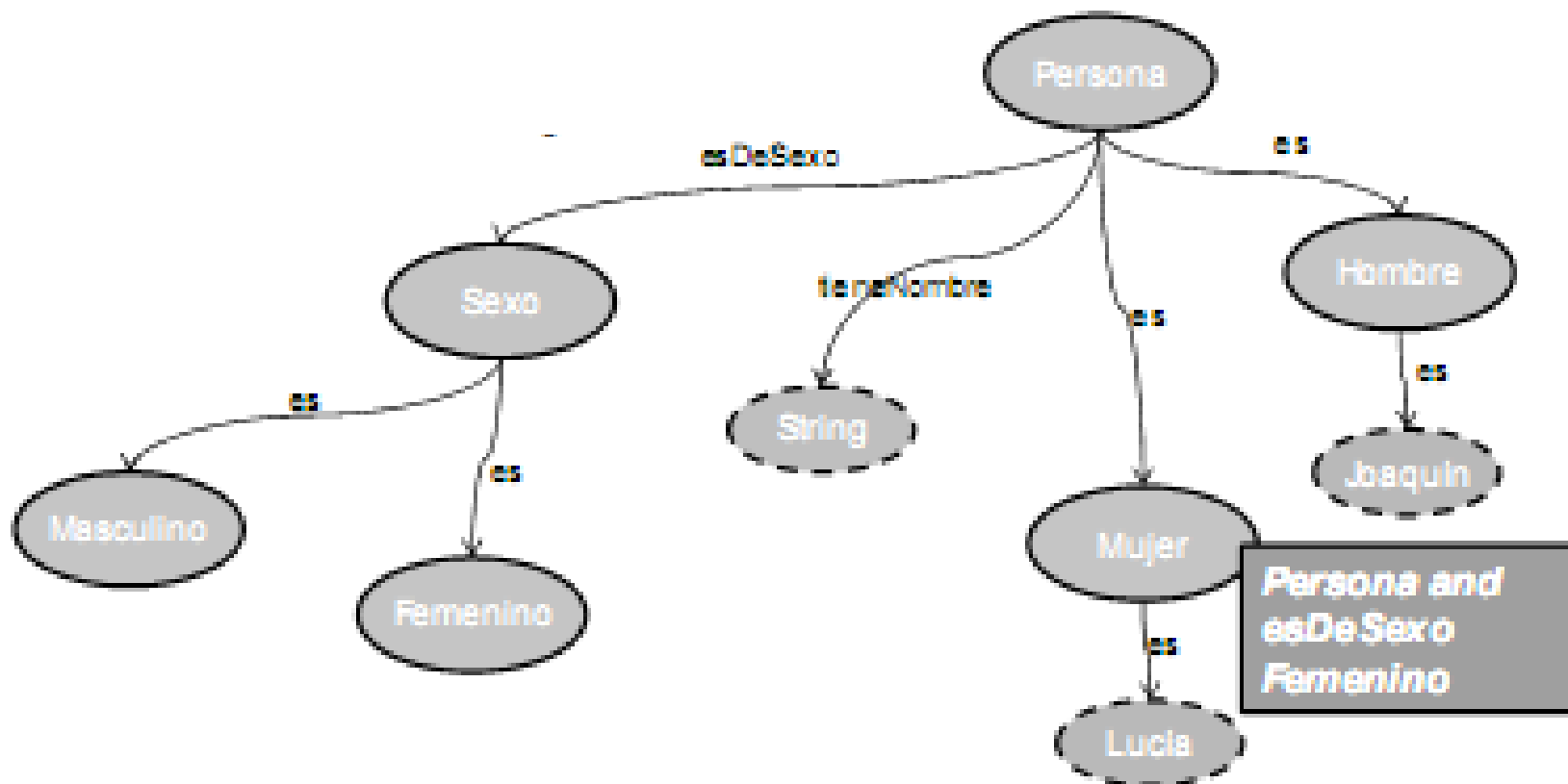


Propiedades

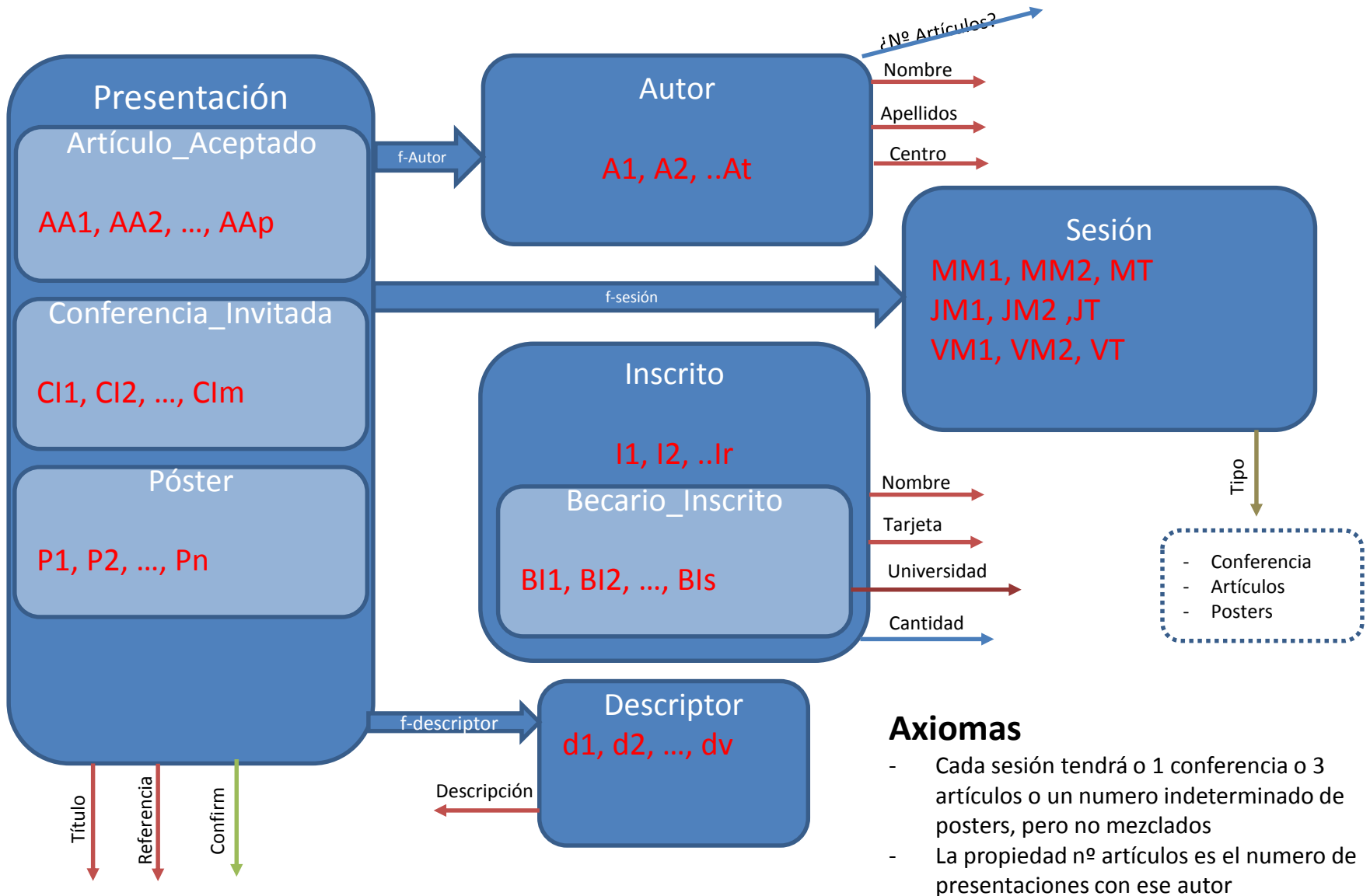


Axiomas

- Restricciones y meta-información sobre las relaciones. Definen el significado y permiten razonar con la ontología



Ontología Congreso



¿Qué aportan las Ontologías?

- Clasificación y consultas automáticas
 - Localizar un concepto o un conjunto de conceptos basándose en la descripción y/o las relaciones.
 - Intercambio de vocabulario entre dominios.
- Legible por:
 - Humanos
 - Computadoras
- Chequeo de consistencia
- Razonamiento automático:
 - Subsumpción: Inferir que la clase A es más general que la clase B.
 - Reconocimiento: Inferir que la instancia X debe ser un hijo de la clase B.

Clasificación de Ontologías

- Ontologías genéricas
 - Conceptos comunes de alto nivel. p.e. “Individuo”, “Conjunto”, “Sustancia”
 - Útil para la reutilización.
 - Importantes cuando generamos o analizamos expresiones de LN.
- Ontologías de dominio
 - Conocimiento específico de dominio.
 - Generalizaciones del dominio.
- Ontologías orientadas a tareas
 - Conocimiento específico de tareas.
 - Generalización de tareas
- Ontologías de aplicación
 - Conceptos comunes de bajo nivel.
 - Combina, integra y extiende todas las sub-ontologías para una aplicación.

¿Dónde usar las ontologías?

- Semantic web
- Inteligencia artificial / IA distribuida
- Sistemas expertos / KBS
- Especificación formal de requerimientos
- Estándares

Lenguajes para la definición de ontologías -Estándares

- RDF
- RDF Schema
- OWL

Estándares básicos

- UNICODE estándar que proporciona el medio por el cual codificar un texto en cualquier forma e idioma
 - IRI international resource identifier cadena caracteres que identifica inequívocamente un recurso (servicio, página, documento, etc.) físico o abstracto
 - identifica el recurso, pero no tiene por que localizar su ubicación(URL)
 - XML meta-lenguaje extensible de etiquetas usado para el intercambio de datos en la web
 - uso de etiquetas con significado intuitivo para humanos, pero no para las máquinas
 - XML estandariza formato no significados
 - nombre de las etiquetas XML no ofrece semántica por si mismo

Representación de conceptos y relaciones (RDF)

- RDF (Resource Definition Format) estándar W3C para describir recursos (cualquier concepto que tenga una URI) en la web
 - Formato común para describir información que pueda ser leída y entendida por una aplicación informática.
 - Permite representar conceptos y relaciones mediante un conjunto de tripletas.
 - tripleta: describe propiedades de un recurso identificado por una IRI:
 - representa un documento o parte de él, o de una colección de documentos, un objeto, e
 - Propiedad: es siempre una IRI (predefinida y con significado preestablecido)
 - Cada tripleta combina: un recurso (Sujeto), una propiedad (Predicado) y un valor para la propiedad (Objeto)



Representación de conceptos y relaciones (RDF)

- RDF permite usar vocabularios semánticos definidos por expertos para describir recursos:
 - Dublin Core: descripción de recursos digitales (páginas HTML, libros, etc) [<http://dublincore.org/>]
 - FOAF (friend of a friend): ontología para descripción de personas [<http://www.foaf-project.org/>]
- Representable en forma de documentos XML [serialización RDF/XML]
- Posibilidad de usar lenguajes de consulta sobre tripletas RDF
 - SPARQL (<http://www.w3.org/TR/rdf-sparql-query/>): sintaxis tipo SQL sobre bases de datos de tripletas
 - RDF. DBPEDIA (<http://wiki.dbpedia.org/>): versión estructurada (tuplas RDF) de la Wikipedia.

RDF Schema

- RDFS (RDF Schema) lenguaje extensible que proporciona los elementos básicos para crear ontologías (vocabularios semánticos RDF)
 - Permite definir clases, relaciones entre clases, restricciones sobre propiedades, etc.
 - `rdfs:Class` declarar recursos como clases para otros recursos
 - `rdfs:subClassOf` definir jerarquías (relaciona clase con superclases)
 - `rdf:property` definir subconjunto de recursos RDF que son propiedades
 - `rdfs:subPropertyOf` definir jerarquías de propiedades
 - `Rdfs:domain` dominio de una propiedad (clase de recursos que aparecen como sujetos en las tripletas de ese predicado)
 - `rdfs:range` rango de una propiedad (clase de recursos que aparecen como objetos en las tripletas de ese predicado)
- RDF Schema define el significado de los términos usados en las tripletas RDF

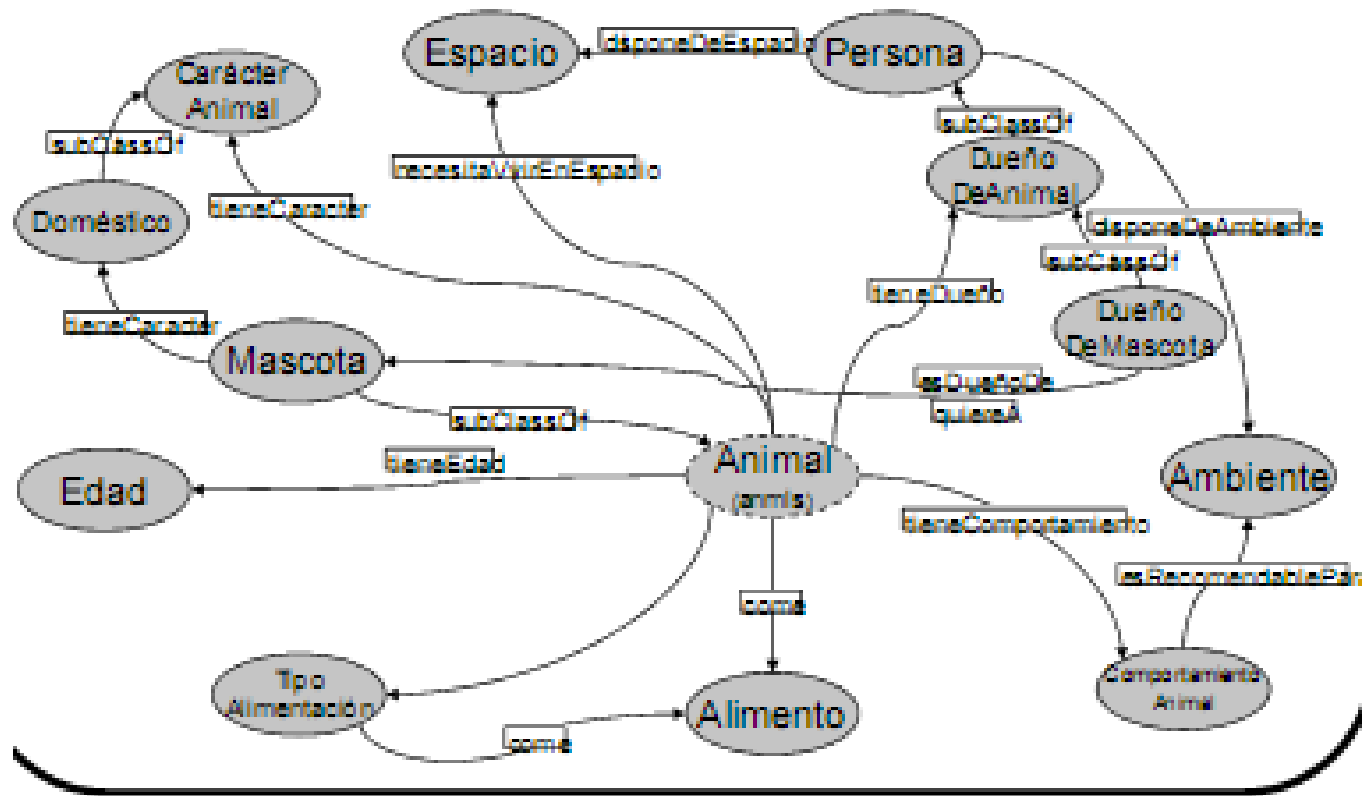
OWL

- OWL (ontology web language) extiende RDFS para permitir la expresión de relaciones complejas entre clases RDFS, y mayor precisión en las restricciones de clases y de propiedades.
 - Derivado de la fusión de los lenguajes de ontologías DAML y OIL.
 - Permite:
 - expresar relaciones entre clases
 - expresar y restringir clases (rango, dominio)
 - expresar y restringir propiedades (cardinalidad)

OWL

- Tres variantes/sublenguajes (menor a mayor potencia expresiva)
 - OWL-lite: versión simplificada (representación de jerarquías simples)
 - OWL-DL: incluye constructores tomados de Description Logics (DL) - Busca compromiso entre máxima expresividad y eficiencia computacional (sólo constructores decidibles de DL)
 - OWL-full: soporte completo de constructores DL

Ontología ejemplo



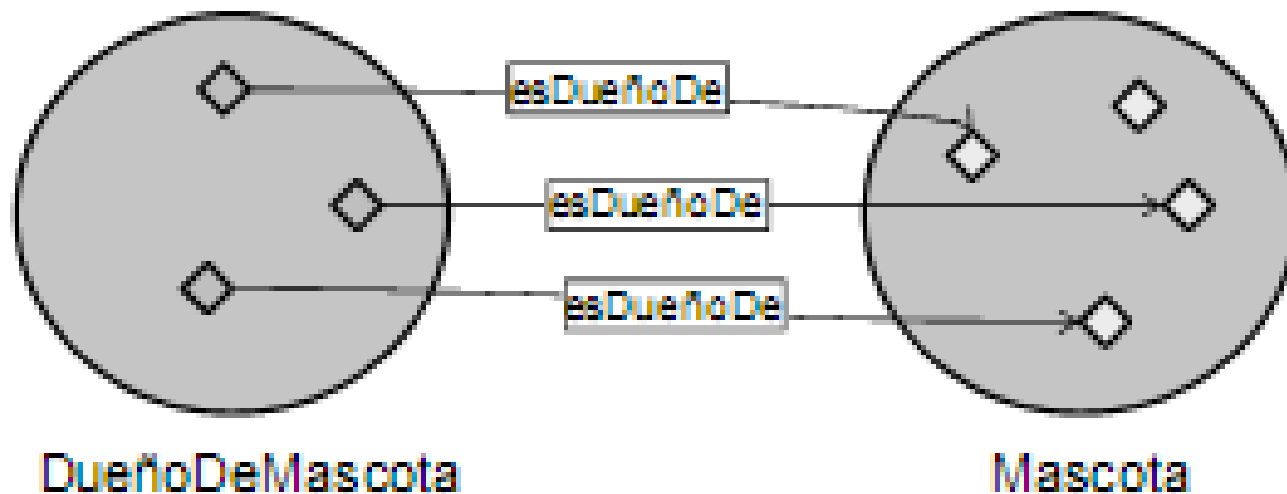
Tipo de propiedades

- Propiedades de tipo de dato
 - Relaciones entre instancias de clases y literales RDF y XML. Esquema de tipo de datos.
- P.e. la clase “Mascota” tiene la propiedad de datos “Nombre”.
- Propiedades de objeto
 - Relaciones entre instancias de dos clases.
 - P.e. la clase “DueñoDeMascota” tiene una propiedad objeto “esDueñoDe”.

Dominio y rango de las propiedades

- Las propiedades ligan individuos de un dominio a individuos de un rango

Dominio \rightarrow esDueñoDe \rightarrow Rango

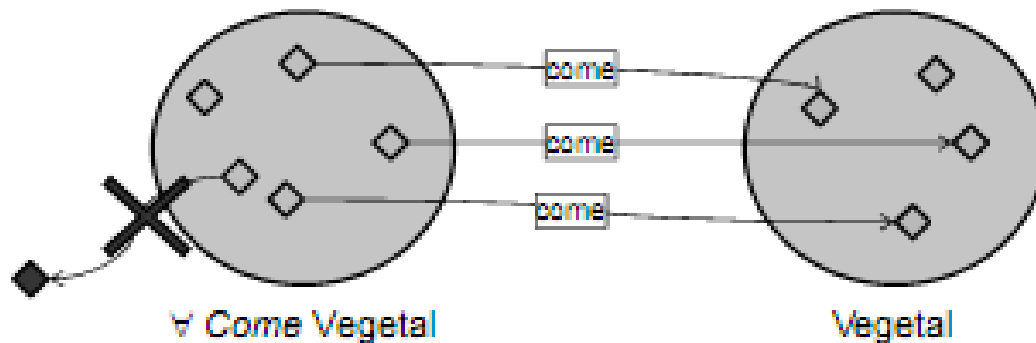


Constructores OWL

Constructor	DL Syntax	Example	Modal Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male	$C_1 \wedge \dots \wedge C_n$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer	$C_1 \vee \dots \vee C_n$
complementOf	$\neg C$	\neg Male	$\neg C$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} \sqcup {mary}	$x_1 \vee \dots \vee x_n$
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor	$[P]C$
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer	$\langle P \rangle C$
maxCardinality	$\leq nP$	≤ 1 hasChild	$[P]_{n+1}$
minCardinality	$\geq nP$	≥ 2 hasChild	$\langle P \rangle_n$

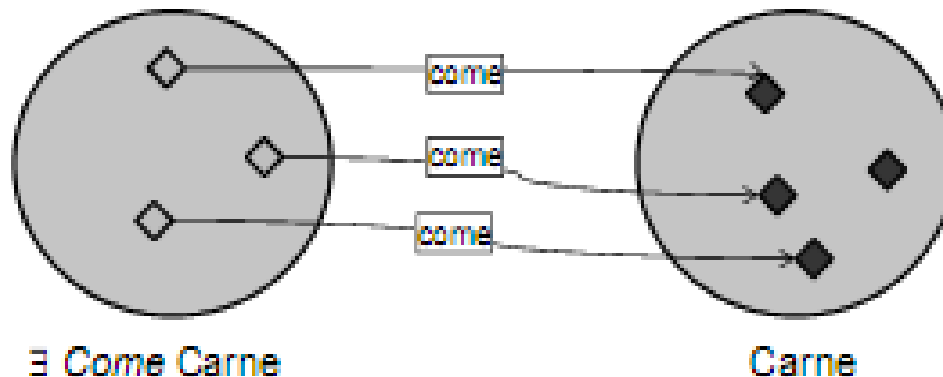
Restricciones de cuantificación

- allValuesFrom (only) – Cuantificador universal
 - Para cada instancia de la clase que tiene instancias de una determinada propiedad, los valores de la propiedad son todos miembros de la clase indicada por la clausula allValuesFrom



Restricción de cuantificadores

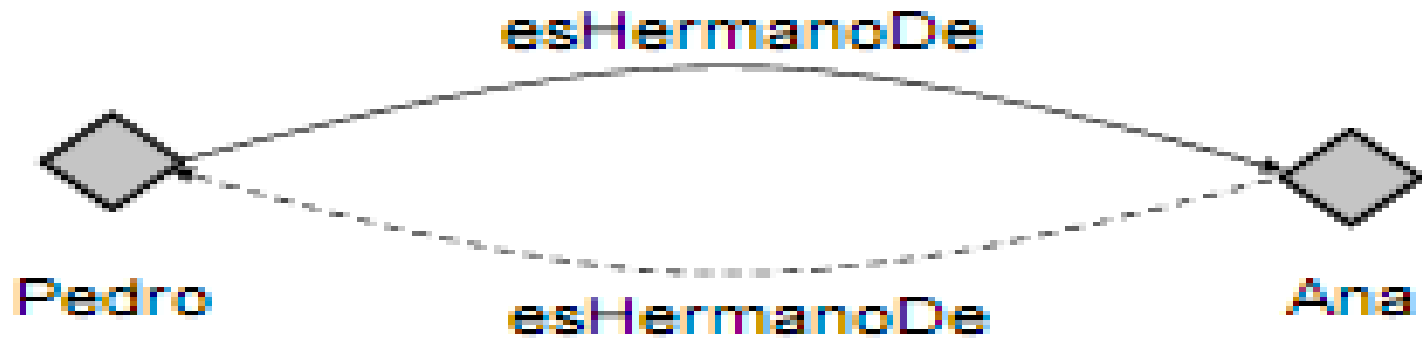
- someValuesFrom (some) – Cuantificador existencial
 - Al menos uno de los valores de la propiedad debe ser miembro de la clase indicada por la cláusula someValuesFrom.



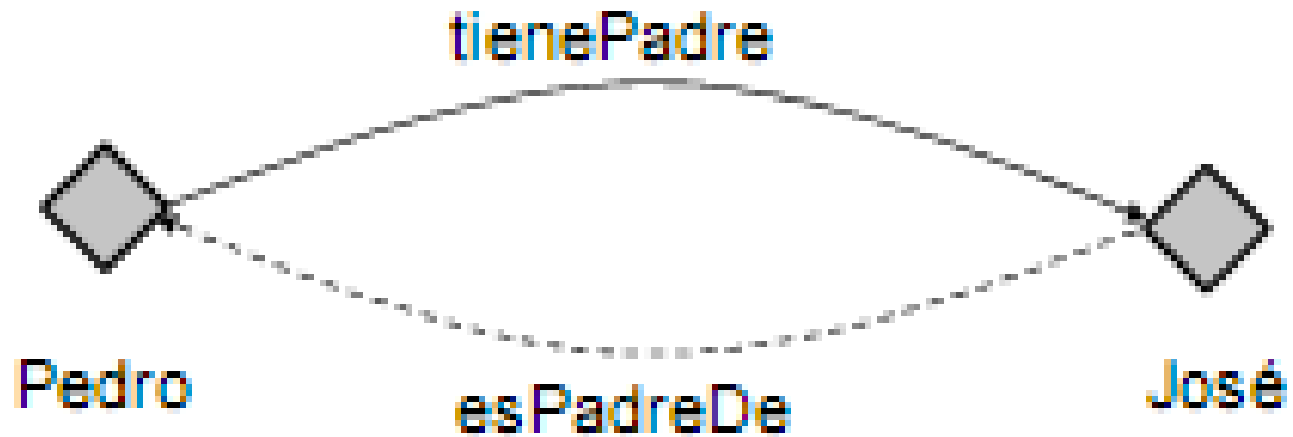
Axiomas OWL

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President.Bush} \equiv {G.W.Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
inverseOf	$P_1 \equiv P_2^-$	hasChild \equiv hasParent ⁻
transitiveProperty	$P^+ \sqsubseteq P$	ancestor ⁺ \sqsubseteq ancestor
functionalProperty	$T \sqsubseteq \leq 1P$	T $\sqsubseteq \leq 1$ hasMother
inverseFunctionalProperty	$T \sqsubseteq \leq 1P^-$	T $\sqsubseteq \leq 1$ hasSSN ⁻

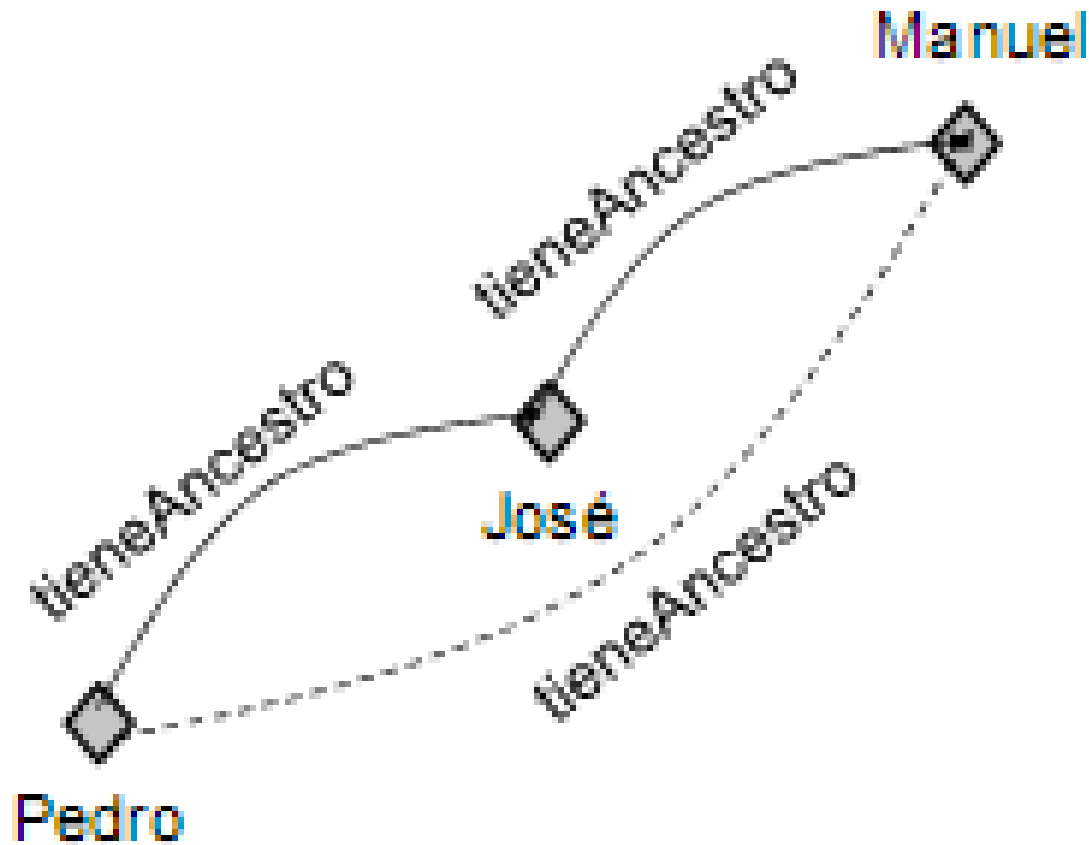
Propiedades simétricas



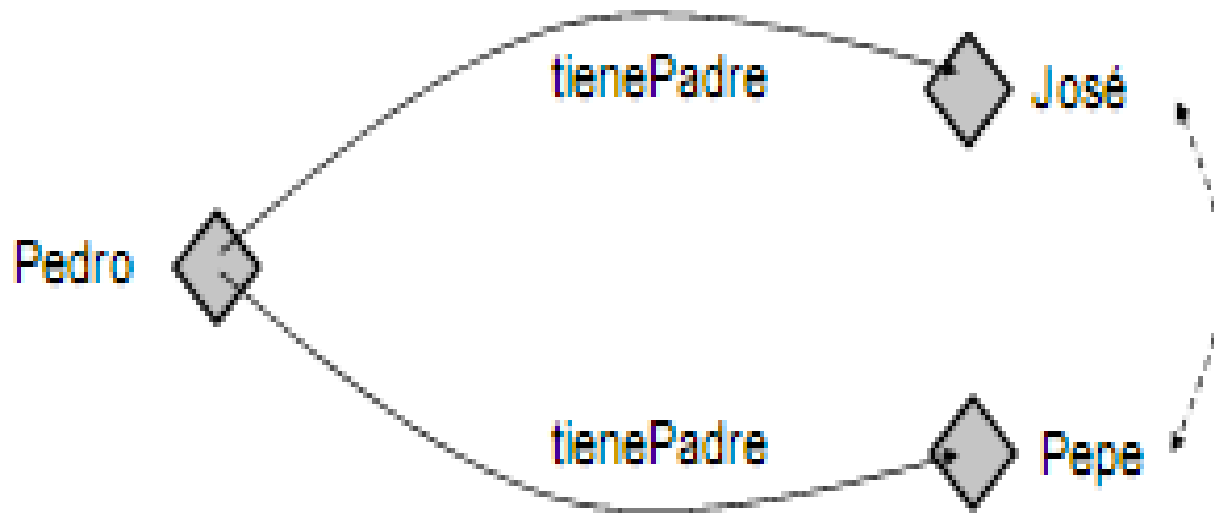
Propiedades inversas



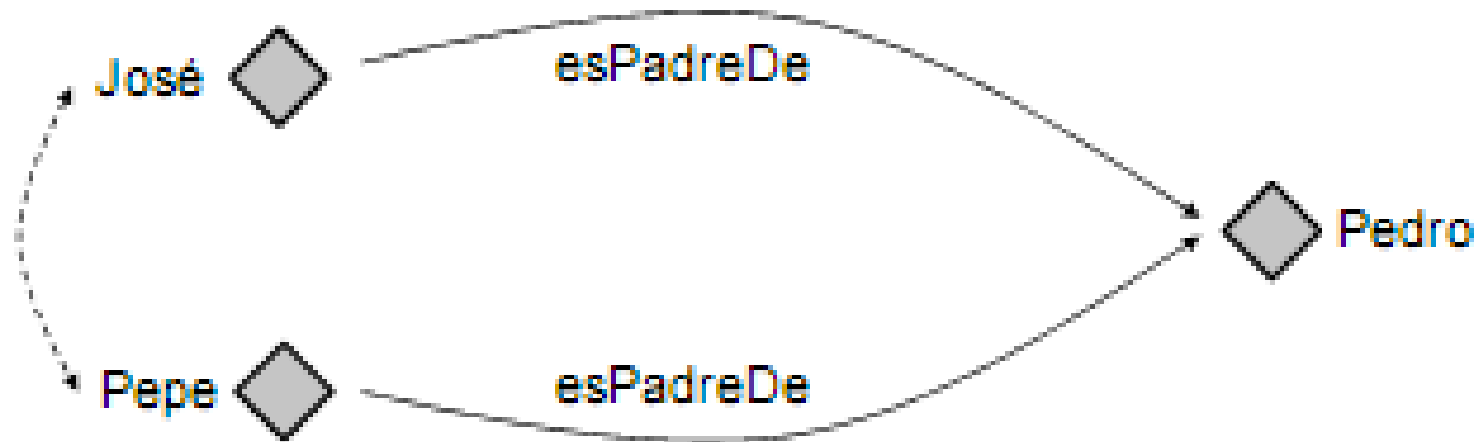
Propiedades transitivas



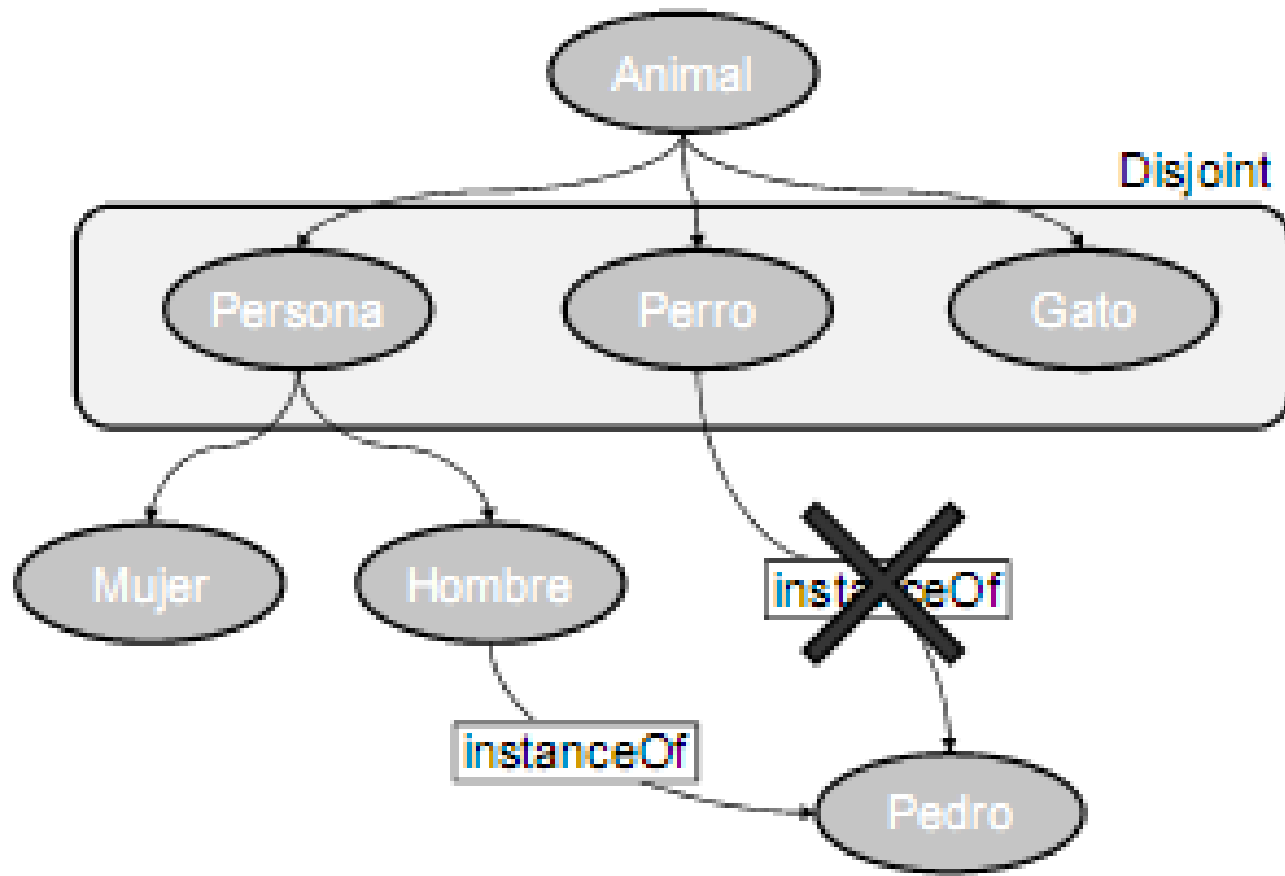
Propiedades funcionales



Propiedades funcionales inversas



Disjunción



Razonadores

- FaCT++

<http://owl.man.ac.uk/factplusplus/>

- Pellet

<http://clarkparsia.com/pellet/download>

- Racer

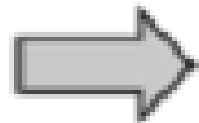
<http://www.racer-systems.com/products/download/index.phtml>

Razonamiento con ontologías

- Clasificación automática
- Clasificación de instancias
- Detección de redundancia
- Chequeo de consistencia
 - Disjoint
 - Restricciones

Clasificación automática

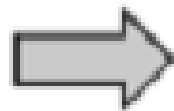
- $\text{DueñoDeMascota} \equiv \text{Persona} \wedge \exists \text{esDueñoDe some Mascota} \wedge \exists \text{quiereA some Mascota}$
- $\text{Mascota} \equiv \text{Animal} \wedge \exists \text{esQueridoPor some DueñoDeAnimal} \wedge \exists \text{tieneDueño some DueñoDeAnimal}$



$\text{DueñoDeMascota} \subseteq \text{DueñoDeAnimal}$

Clasificación automática

- $Gato \equiv Animal \cap \forall \text{ come only } AlimentoCarnivoro$
- $AnimalCarnivoro \equiv \forall \text{ come only } AlimentoCarnivoro$



$Gato \subseteq AnimalCarnivoro$

Clasificación de instancias

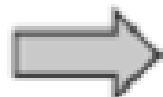
- { Carlos, Ariel, Virginia, Vicky } ∈ Persona
- { Criollo_1, Criollo_2, PerroTom, GatoBom } ∈ Mamifero
- esDueñoDe(Carlos, Criollo_1)
- esDueñoDe(Carlos, Criollo_2)
- esDueñoDe(Ariel, PerroTom)
- esDueñoDe(Vicky, GatoBom)
- quiereA(Ariel, PerroTom)
- quiereA(Vicky, GatoBom)



Carlos ∈ DueñoDeAnimal
{ Ariel, Vicky } ∈ DueñoDeMascota

Clasificación de instancias

- *Mascota \equiv Animal $\wedge \exists$ tieneDueño some DueñoDeAnimal $\wedge \exists$ esQueridoPor some DueñoDeAnimal*
- *Vicky \in Persona*
- *GatoBom \in Gato*
- *tieneDueño(GatoBom, Vicky)*
- *esQueridoPor(GatoBom, Vicky)*



GatoBom \in Mascota

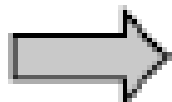
Detección de redundancia

- $\{Ariel, Vicky\} \in \text{DueñoDeMascota}$
- $\text{DueñoDeMascota} \subseteq \text{Persona}$

 $\{Ariel, Vicky\} \in \text{Persona (REDUNDANT)}$

Chequeo de consistencia

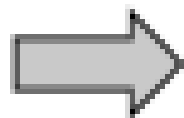
- $\text{DueñoDeMascota} \subseteq \text{Persona}$
- $\text{Mascota} \equiv \text{Animal} \cap \exists \text{ tieneDueño some DueñoDeAnimal} \cap \exists \text{ esQueridoPor some DueñoDeAnimal} \cap \forall \text{ tieneCaracter only Domestico}$
- $\text{Ñandu} = \text{Ave} \cap \forall \text{ tieneCaracter only Salvaje}$
- $\text{Vicky} \in \text{DueñoDeMascota}$
- $\text{ÑanduAnahi} \in \text{Ñandu}$
- $\text{ÑanduAnahi} \in \text{Mascota}$
- $\text{tieneDueño}(\text{ÑanduAnahi}, \text{Vicky})$
- $\text{esQueridoPor}(\text{ÑanduAnahi}, \text{Vicky})$



$\text{ÑanduAnahi} \in \text{Mascota}$ (INCONSISTENT)

Chequeo de consistencia

- $Carnivoro \equiv Animal \sqcap \forall \text{ come only } AlimentoCarnivoro$
- $PerroTom \in Carnivoro$
- $Arroz \in AlimentoHerbivoro$
- $(AlimentoHerbivoro \sqcap AlimentoCarnivoro) \subseteq \perp$
- $(Herbivoro \sqcap Carnivoro) \subseteq \perp$



$\text{come}(PerroTom, Arroz)$ (INCONSISTENT)

Consultas con RDQL

- ¿Qué mascota tiene una cierta persona?

```
SELECT ?name
WHERE  (?name, <rdf:type>, masc:Mascota),
       (?name, masc:tieneDueno, <" + nameSpace + "Vicky">)
      USING masc for <" + nameSpace + ">"

nameSpace =
  "http://www.exa.unicen.edu.ar/escuelapav/ontologias/OntoMasc
  otas#"
```

Conclusiones

- Las ontologías...
 - Definen vocabulario común.
 - Crean entendimiento compartido.
 - Proveen acceso común al conocimiento.
 - Permiten la extracción de nuevo conocimiento implícito a través de razonamiento automático.
 - Permiten compartir, integrar y re-utilizar conocimiento.
 - Proveen conocimiento entendible por humanos y computadoras.