

Tema-4-Sistemas-de-Representacio...



ParmigianoReg



Ingenieria del Conocimiento



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada

NEW

WUOLAH Print

Lo que faltaba en Wuolah



Imprimir



Tema 4 - Sistemas de Representación Estructurados

Profesor Juan Luis Castro Peña, Curso 20-21

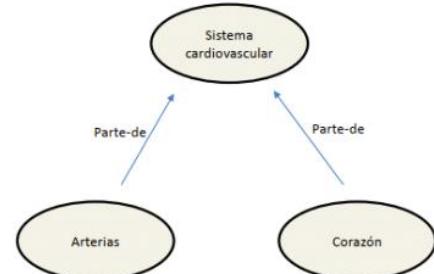
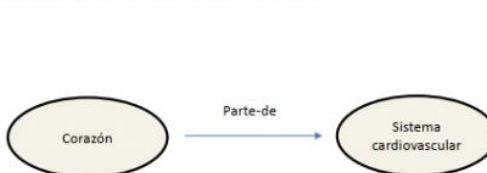
- Los Sistemas de Representación Estructurados tienen el objetivo de representar el conocimiento mediante grafos, ya sean conceptos o relaciones, además, facilitan la representación de este conocimiento humano.
- Permiten por su propia estructura deducir nuevos hechos
- Existen dos modelos clásicos:
 - Redes Semánticas
 - Frames

Redes Semánticas (Quillian, 1968):

- Es el precursor de los frames.
- Pretende representar el lenguaje natural de forma simple y visual, como un grafo.
- Formalismo limitado, solo para dominios simples y forma de inferencia simple, no adecuado para tratar con formas de inferencia más sofisticadas. Inferencia basada en herencia.
- Una red semántica se representa como un grafo dirigido etiquetado, donde se etiquetan nodos y las conexiones.
 - Nodos: representan conceptos, ya sea un objeto individual o una clase de objetos.
 - Arcos: representan relaciones binarias entre los conceptos.
- Se debe de tener cuidado con las deducciones ya que el sistema puede quedarse sin memoria por las deducciones que realiza

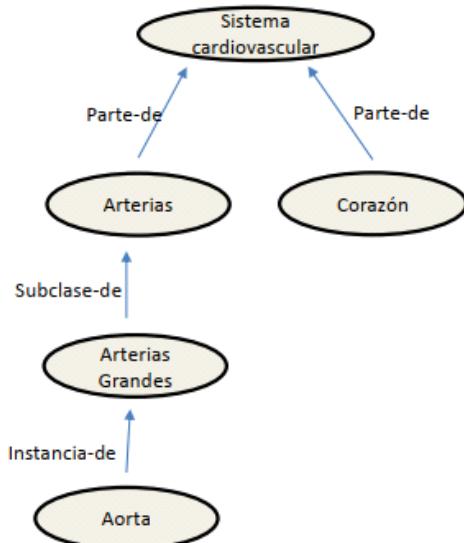
El corazón y las arterias son parte del sistema cardiovascular

El corazón es parte del sistema cardiovascular



El corazón y las arterias son parte del sistema cardiovascular

La aorta es una arteria grande

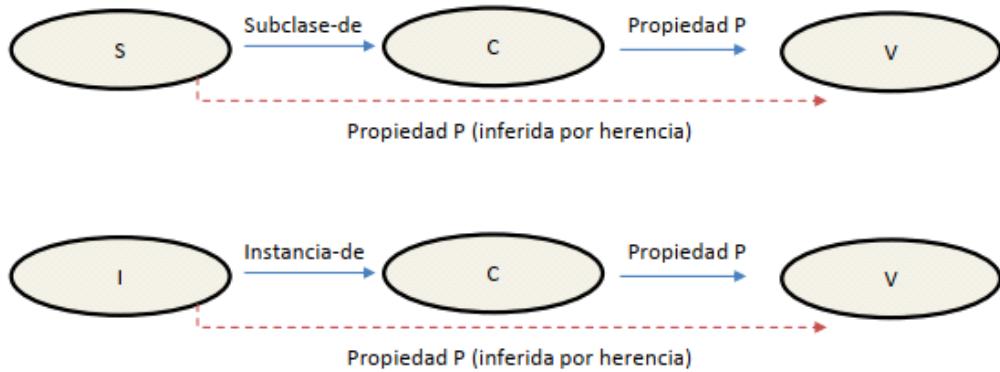


Relaciones Subclase-de e Instancia-de

- Son dos tipos de relaciones que se utilizan en todas las redes semánticas, por medio de ellas se realiza el proceso de herencia, que es la forma de inferencia asociada a las redes semánticas.
- En el lenguaje natural corresponden con las expresiones "es un", "son", ..., etc.

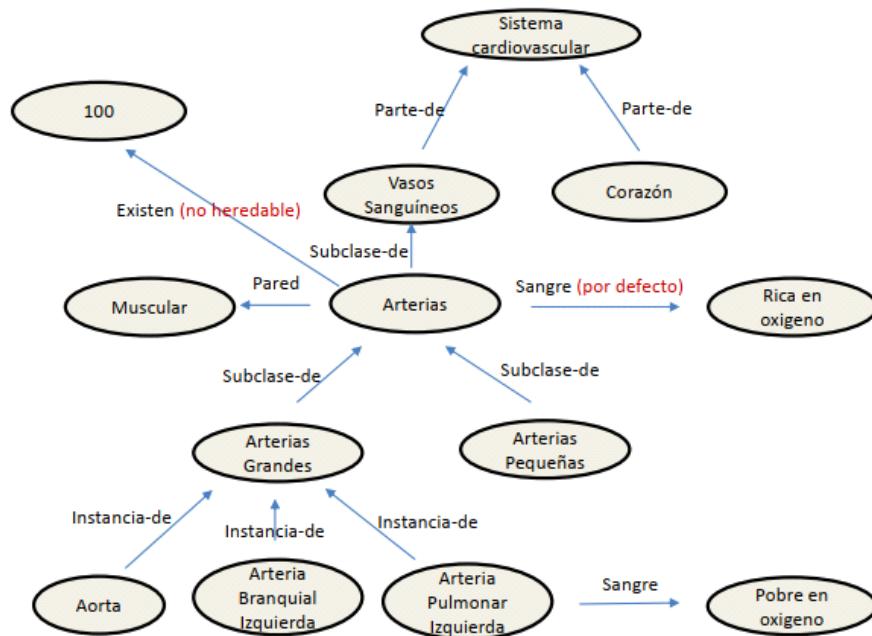
Herencia

- Forma de inferencia asociada a las redes semánticas, corresponde con el siguiente razonamiento:
 - "Las subclases y las instancias heredan las propiedades de las clases más generales"
- Gráficamente se traduce en...



Excepciones a la Herencia

- Puede ocurrir que una propiedad heredada no sea válida.
- Soluciones posibles:
 - Hacer que la propiedad sea por defecto: la propiedad no se hereda si el concepto ya contiene esa propiedad explícitamente en la red
 - Marcar esa propiedad como no heredable.





COMPRAR
ENTRADAS

MARVEL STUDIOS

DOCTOR STRANGE EN EL MULTIVERSO DE LA LOCURA

6 DE MAYO SOLO EN CINES

©2022 MARVEL. PENDIENTE DE CALIFICACIÓN POR EDADES.

Redes Semánticas Extendidas

- Introducidas para representar en una red semántica predicados que no sean binarios, de esta manera se puede representar cláusulas de lógica mediante una red semántica.
- Para establecer una equivalencia semántica entre la forma causal de la lógica y las redes semánticas, se pueden utilizar reglas de inferencia de la lógica para manipular los nodos y arcos de la red.

Representación de predicados unarios

Ejemplo: Traducción predicado unario a binario

Supongamos el siguiente predicado unario:

Arteria(x) = "x es una arteria"

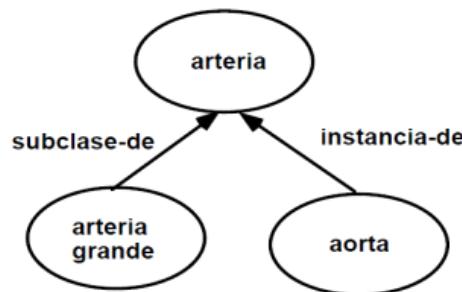
y las cláusulas

Arteria(aorta) y Arteria(arteria-grande)

Estas cláusulas pueden ser reemplazadas por las cláusulas

instancia-de(aorta, arteria)

subclase-de(arteria-grande, arteria)



Representación de un predicado n-ario ($n > 2$)

Ejemplo:

PresiónSangre(x, y, z) = "la presión sanguínea de x varía entre y mmHg y z mmHg"

El predicado

PresiónSangre (arteria, 40, 80)

puede ser reemplazado por la conjunción de predicados binarios:

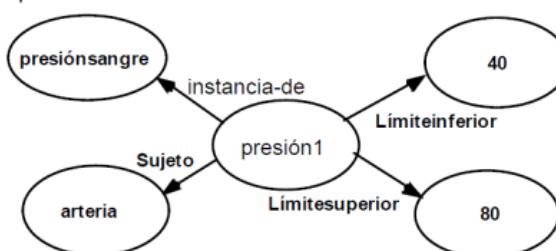
instancia-de(presión1, presiónsangre)

sujeto(presión1, arteria)

Límiteinferior(presión1, 40)

Límitesuperior(presión1, 80)

Red semántica equivalente:



Redes Semánticas en CLIPS:

Definición de conceptos y arcos

```
(deffacts Conceptos
    (Concepto <etiqueta_concepto>
        ...
        (Concepto <etiqueta_concepto>)
    )
    (deffacts Arcos
        (Arco <etiqueta_arco> <etiqueta_concepto_origen> <etiqueta_concepto_destino>)
    )
```

Herencia

```
(defrule Herencia_red_semantica
    (Arco Instancia_de|Subclase_de ?O ?C)
    (Arco ?Propiedad ?C ?V)
    =>
    (assert (Arco ?Propiedad ?O ?V))
    ; Opcionalmente;
    (printout t "He deducido que por herencia de la clase " ?C ", " ?Propiedad " de " ?
    O " es " ?V crlf)
)
```

Limitaciones de las redes semánticas:

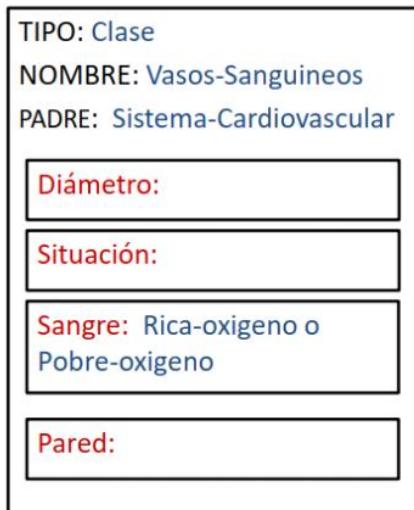
- Resulta poco natural que en las redes semánticas que los conceptos relevantes del dominio como los valores de las propiedades de esos conceptos estén al mismo nivel en la red.
 - No se resaltan conceptos relevantes.
 - No resulta cómodo reconocer las propiedades de un concepto, habría que mirar todas las flechas que salen de un concepto en una red bastante grande.
 - No se resalta la estructura jerárquica entre esos conceptos relevantes.

Frames (Minsky, 1975):

- Se basan en considerar la *resolución de problemas humanos como el proceso de llenar huecos de descripciones parcialmente realizadas*.
- La idea subyacente en un sistema basado en Frames es que el conocimiento concerniente a los conceptos de un dominio (individuos o clases de individuos) incluyendo las relaciones entre los mismos es almacenada en una entidad compleja de representación llamada Frame.
- Un Frame representa un objeto o concepto relevante e incluye propiedades del mismo.
- Un conjunto de Frames que representa el conocimiento de un dominio de interés es organizada jerárquicamente en lo que es llamado una taxonomía.
- El método de razonamiento automático que utilizan los frames es la herencia, un frame (concepto) hereda de los frames de rango superior en la taxonomía.
- El conocimiento relevante de un concepto (objeto individual o clase de objetos) es representado mediante una entidad compleja de representación llamada Frame constituida por un conjunto de propiedades (atributos)
- Proporcionan formalismo para agrupar explícitamente todo el conocimiento concerniente a las propiedades de objetos individuales o clases de objetos.
- Un template de CLIPS es básicamente un Frame.
- Tipos de Frames:
 - Frames Clase: son genéricos, representan el conocimiento de clases de objetos.
 - Frames Instancia: representan conocimiento de objetos individuales.

Traducción

Traducción



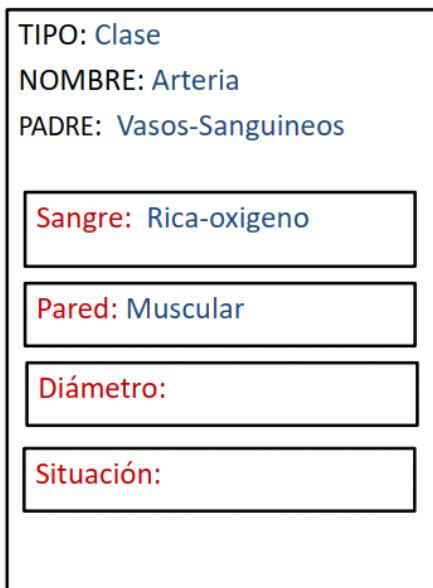
Los vasos sanguíneos forman parte del Sistema Cardiovascular. Tienen como propiedades principales el diámetro, la situación, el tipo de sangre que contienen (puede ser rica o pobre en oxígeno) y el tipo de pared por la que están formados.

Propiedades

- Cada Frame posee tres propiedades fijas:
 - TIPO: Clase o Individuo
 - NOMBRE: Etiqueta única que los identifica
 - PADRE: Nombre del Frame Padre de la jerarquía.
- Tienen otras propiedades, conocidas como slots cuyo valor caracteriza individualmente a los distintos individuos.
- En cada Frame las propiedades podrán tener un valor fijo propio de la clase o del individuo, o un conjunto de posibles valores.
- Los Frames iniciales tendrán un padre que será un Frame llamado "Cosa", para que todos los frames tengan un padre.

Herencia

- Cada Frame hereda las propiedades del Frame Padre.
- Cada Frame hereda los valores de una propiedad (o las restricciones de esos valores) del Frame Padre.



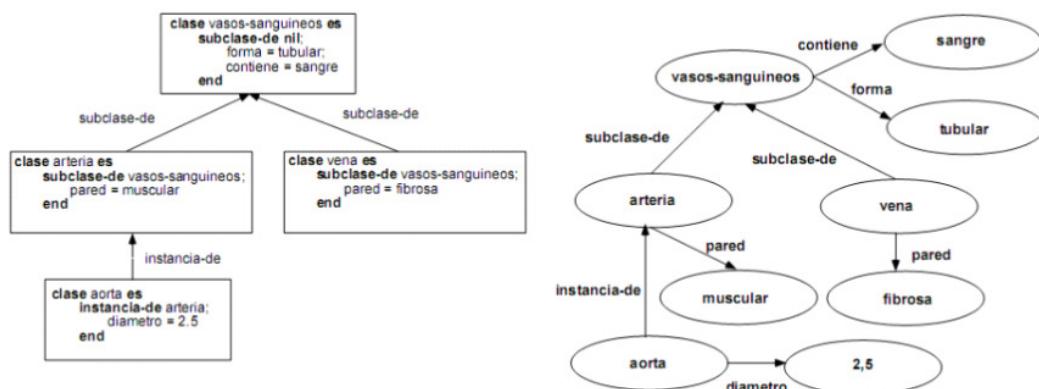
Heredado

Equivalencia Redes Semánticas-Frames

- Una red semántica puede ser representada como un conjunto de Frames
 - Crear un Frame por cada concepto de la red de la que salga una relación (flecha) del tipo Instancia-de o Subclase-de
 - En esos Frames se ponen como propiedades propios del Frame las relaciones que salgan de él y que no sean del tipo Instancia-de o Subclase-de
- Una jerarquía de Frames se puede representar como una red semántica.
 - Para cada Frame, etiquetamos un concepto de la red con el nombre del Frame.
 - Se añade una relación (flecha) de tipo Instancia-de o Subclase-de desde ese concepto hasta el concepto correspondiente a su padre.
 - Para cada valor de cada propiedad del Frame se añade una relación (flecha) desde ese concepto al concepto correspondiente al valor.

Equivalencias Frames / Redes Semánticas

clases, instancias y valores atributos ≡ conceptos (nodos)
atributos ≡ relaciones (arcos)



Herencia Simple (Excepciones)

- Los atributos (slots) se pueden marcar como heredables o no heredables.
- Los valores también pueden tener excepciones, si un individuo posee un valor diferente para un atributo que su padre, el valor del padre no se hereda.
- Las restricciones de valores o tipo de valores también podrán tener excepciones. Si ya un individuo tiene una restricción particular, no heredará la del padre.

```

clase arteria es
  subclase-de vasos-sanguíneos;
    sangre = rica-oxigeno
end

```

```

instancia arteria-pulmonar-izquierda es
  instancia-de arteria;
    sangre = pobre-oxigeno;
end

```

Arteria-pulmonar-izquierda no hereda el valor de sangre de Arteria porque ya tiene un valor explicitado en su frame

Facetas

- Para facilitar la definición de excepciones y de restricciones se extienden los Frames para que cada slot pueda tener facetas. Una faceta es una propiedad del slot.
- Se utilizan para extender los Frames ya que el formalismo que poseen no permiten:
 - Saber si el valor del atributo de una instancia ha sido heredado o ha sido especificado explícitamente.
 - Calcular los valores de un atributo a partir de los valores de otros atributos.
 - Muchos lenguajes de Frames proporcionan constructores especiales del lenguaje llamados facetas que permiten manejar las funcionalidades anteriores.

Tipos de Facetas

faceta valor	La más común y referencia el valor real del atributo.
faceta valor por defecto	Denota el valor inicial del atributo en caso de que no se especifique lo contrario.
faceta tipo valor	Especifica el tipo de datos del valor del atributo.
faceta cardinalidad	Especifica si se trata de un atributo uni o multivaluado.
faceta máxima cardinalidad	Solo válida para atributos multivaluados, especifica el máximo número de valores asociados al atributo
facetas demonio	Permiten la integración de conocimiento declarativo y procedural. Un demonio o valor activo es un procedimiento que es invocado en un momento determinado durante la manipulación del atributo donde ha sido especificado (si-necesario, si-añadido, si-eliminado)
faceta herencia	Especifica el tipo de herencia del atributo

Tipos de Herencia de Valores

- La herencia depende de cómo se recorre la taxonomía para determinar los valores del atributo considerado.



Motivación: Limitaciones de los Frames:

- Los frames no permiten de manera cómoda representar relaciones entre conceptos.
- En los frames, cada subconcepto sobre el que se quiera deducir o razonar cosas se tiene que especificar en forma de Frame.
- En los frames, cada condición que deben satisfacer los conceptos (clases o individuos) hay que representarlo en forma de procedimiento, hay que programarlo y por tanto será difícil modificar el conocimiento si lo escribió otra persona; lo que hace que vaya en contra de la filosofía de los SBCs
- Además, habitualmente los conocimientos que expresamos presentan:
 - Inconsistencia (Afirmaciones contradictorias)
 - Incompatibilidad (Afirmaciones que no tienen sentido, por lo menos en el contexto de quién lo recibe)
 - Falta de completitud (Falta de información sobre algo)
 - Falta de límites (Información que no entendemos porque presupone un conocimiento del que no se dispone)

Objetivo: Representar el conocimiento abstracto con un modelo normalizado independiente del lenguaje

- Permitirá intercambiar el conocimiento entre personas que hablan diferentes idiomas.
- Permite que las máquinas procesen el conocimiento y hagan deducciones.
- Evita la ambigüedad del lenguaje, se resuelve al modelar.

Propiedades deseables del modelo de representación:

- Que incluya la posibilidad de representar
 - Conceptos (clases e individuos) *de manera similar a los Frames.*
 - Propiedades de esos conceptos *similar a los Frames.*
 - Relaciones (entre esos conceptos): "*Cada presentación tiene uno o más autores*"
 - Condiciones que deben verificar los conceptos, propiedades o relaciones: "*Una sesión constará de: o bien de 3 artículos, o bien de un conferencia o bien un número indefinido de posters*"
- Permita reducir
 - Si hay inconsistencia: "*Hay presentaciones sin ningún autor inscrito*"
 - Si hay incompletitud (Falta de información sobre algo): "*Hay autores sin filiación*"
 - Hablar y deducir cosas sobre conceptos construidos con conceptos básicos sin tenerlos que incluir: "*¿Qué país es el que tiene más autores en el congreso?*"
 - No haga falta implementar, solo representar, de forma que cualquiera pueda comprender el conocimiento, completarlo y modificarlo.

Para todo esto se utilizan **Ontologías**.

Otros usos de ontologías: Web Semántica vs Web Actual

- La web actual representa la información usando documentos en lenguaje natural de poca estructura
 - Fácil comprensión por los humanos (HTML sólo define presentación, por ejemplo)
 - Difícil soporte para el procesamiento automático.

Se debe representar el contenido de forma normalizada, de manera que lo pueda manejar y utilizar una máquina.

- Alternativas para facilitar el procesamiento de la información de la web
 - Máquinas más inteligentes para comprender el significado de la información que hay en la web.
 - Procesamiento de lenguaje natural, reconocimiento de imágenes, etc.
 - Información más inteligente, se representa la información de modo que sea sencilla de comprender para las máquinas.
 - Expresar los contenidos en un formato procesable automáticamente.
 - Uso de metainformación

¿Por qué tanto interés actualmente en las ontologías?

- Hay un incremento en necesidad de
 - Capturar el conocimiento
 - Definir un vocabulario común (consensuado)
 - Compartir entendimiento común (reutilizable)

- Interpretación y manipulación automática

Ontologías:

Orígenes de las ontologías

- En Filosofía
 - Ontología es 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, descrito, definido formalmente para ser interpretado por una máquina.

Definición

- Una ontología trata sobre la exacta descripción de las cosas y sus relaciones -- W3C
- 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 catálogo 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

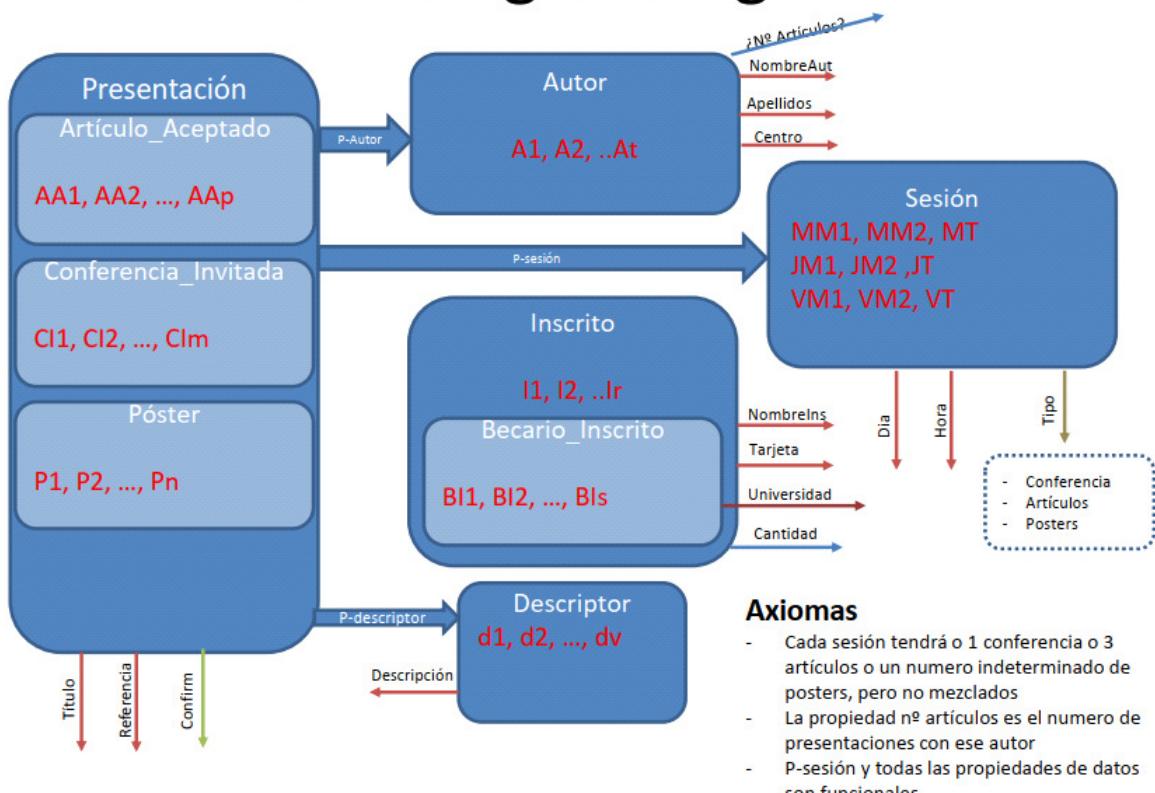
Estructura Jerárquica y de Relaciones

- En una ontología existe una estructura jerárquica entre conceptos del mismo tipo que los Frames (Subclase_de e Instancia_de), se puede ver ejemplificado como los conjuntos y subconjuntos en el ejemplo.
- Hay una estructura entre los conceptos parecida a la de las redes semánticas, donde las flechas o arcos son las propiedades de objeto (En el ejemplo, las **flechas gruesas**) y además, cada concepto tiene las propiedades de datos del mismo tipo que los atributos de los Frames (En el ejemplo, las **flechas finas**)

Componentes de una ontología

- Una ontología está compuesta por:
 - Conceptos o clases: colección de individuos
 - Instancias o individuos: objetos en el mundo y valores
 - Propiedades o relaciones: describen las relaciones entre los conceptos, las propiedades pueden ser de objetos o de datos.
 - Las propiedades de un **objeto** son relaciones entre **conceptos** (Por ejemplo, Autor_de es una propiedad que va de la clase presentación a la clase autor)
 - Las propiedades de **datos** son similares a las propiedades de un Frame, atributos que se rellenan con **valores** de un tipo de dato (Por ejemplo, nombre es una propiedad de la clase Autor de tipo de dato string)
 - Axiomas: Se usan para describir las condiciones que se tienen que verificar, son restricciones y metainformación sobre las relaciones, definen el significado y permiten razonar con la ontología.

Ontología Congreso



Axiomas

- Cada sesión tendrá o 1 conferencia o 3 artículos o un número indeterminado de posters, pero no mezclados
- La propiedad nº artículos es el número de presentaciones con ese autor
- P-sesión y todas las propiedades de datos son funcionales

¿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 y Máquinas
- Chequeo de consistencia
- Razonamiento automático
 - **Subsunción**: Inferir que una 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 las Ontologías

- **Ontologías Genéricas**
 - Conceptos comunes de alto nivel como "Individuo", "Conjunto" o "Substancia".
 - Útil para la reutilización.
 - Importantes cuando generamos o analizamos expresiones de LN.
- **Ontologías de Dominio**
 - Conocimiento específico del 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 utilizar las Ontologías?

- En la Web Semántica
- En IA / IA Distribuida
- Sistemas Expertos / Knowledge-Based Systems
- Especificación formal de requerimientos
- Estándares

Representación Estándar (Formal y Lógica) de las Ontologías

- Las ontologías introducen conceptos y predicados básicos sobre un dominio:
 - Los conceptos se traducen en clases o individuos.
 - **Ejemplo:** El concepto profesor, alumno o asignatura serían clases útiles para describir el conocimiento sobre unos estudios, que tendrían profesores, alumnos o asignaturas concretas como individuos.
 - Los predicados se traducen en relaciones entre los individuos de una clase origen o dominio y los individuos de una clase destino o rango.
 - **Ejemplo:** La relación es_profesor_de es un predicado que relaciona individuos de la clase profesor (dominio) con la clase asignatura (rango)
 - Los conceptos pueden tener una relación jerárquica de subclase
 - **Ejemplo:** Asignatura optativa sería una subclase de asignatura
 - En algunos predicados, el destino o rango es un tipo de dato estándar (Predicado de datos)
 - **Ejemplo:** El valor del predicado nombre tomará como valores una cadena de caracteres.
 - Las clases e individuos junto a sus propiedades de datos y su jerarquía tienen una estructura similar al Frame.
 - Una primera ampliación sobre los Frames es que además hay propiedades que pueden tomar valor en clases definidas dentro de la antología.
 - Otra ampliación es que las ontologías permiten añadir axiomas, que son condiciones que tienen que verificarse y que permitirán integrar conocimiento a muy alto nivel de forma muy simple.

¿Cómo se representan las ontologías formalmente?

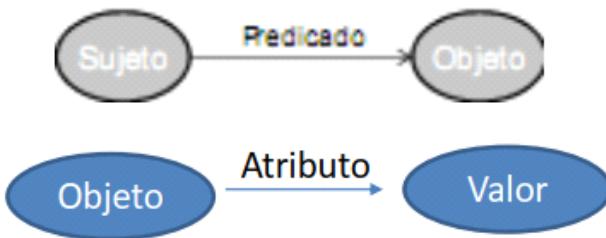
- Mediante lenguajes estándar para la definición de ontologías
 - RDF
 - RDF Schema
 - OWL

Estándares Básicos

- Se utiliza Unicode Estándar para codificar el texto en cualquier forma o idioma.
- Se utiliza el IRI, el International Resource Identifier: Es una cadena de caracteres que identifica inequívocamente un recurso (ya sea un servicio, página, documento, etc) físico o abstracto. Identifica un recurso pero no necesariamente lo tiene porque localizar (a diferencia de una URL)
- Cada objeto va a poseer un IRI que lo identifique.
- Para el intercambio de datos en la web se utiliza el metalenguaje XML
 - Usan etiquetas con significado intuitivo para humanos pero no para las máquinas.
 - XML estandariza formato pero no significados.
- Nota: Un IRI es lo mismo que un URI (Unified Resource Identifier), la diferencia recae que URI utiliza codificación ASCII, mientras que IRI utiliza Unicode.

Representación basada en ternas

- Clásicamente, una afirmación básica se puede representar por medio de una terna
 - **Objeto-Atributo-Valor;** por ejemplo: "Ingeniería de Conocimiento se imparte en el 3º curso" se traduce como (Ingenieria_del_conocimiento, Curso, 3º)
 - Análogamente se puede hablar de la estructura **Sujeto-Predicado-Objeto;** por ejemplo (Juan_Luis_Castro, Profesor, Ingenieria_de_conocimiento)
- Son formas equivalentes, la idea es poder representar el conocimiento como una terna donde el segundo componente es el predicado que relaciona la primera con la tercera.



Representación de Conceptos y Relaciones (RDF)

- RDF (Resource Definition Format) es un estándar del W3C para describir recursos (cualquier recurso que posea una IRI) en la web utilizando una representación basada en ternas.
 - Formato común para describir información que pueda ser leída y entendida por un programa.
 - Permite representar conceptos y relaciones mediante un conjunto de triplets.
 - Triplet: Describe propiedades de un recurso identificado por una IRI
 - Representa un documento o parte de él, o de una colección de documentos.
 - Propiedad: Es siempre una IRI predefinida y con significado preestablecido.
 - Cada triplet combina un recurso (Sujeto), una propiedad (Predicado) y un valor de la propiedad (Objeto)
- El formato se compone de la terna (IRI, IRI, VALOR) donde
 - El primer IRI es el identificador del sujeto
 - El segundo IRI es el identificador del predicado
 - El VALOR es libre, podría ser un cierto tipo de dato, otro IRI.

Uso del Estándar RDF

- RDF permite usar vocabularios semánticos definidos por expertos para describir recursos
 - Dublin Core: Descripción de recursos digitales.
 - FOAF (Friend Of A Friend): Ontología para describir personas.
- Representable en forma de documentos XML (Serialización RDF/XML)
- Posibilidad de usar lenguajes de consulta sobre triplets RDF
 - SPARQL: Sintaxis tipo SQL sobre bases de datos de triplets.
 - RDF DBPEDIA: Versión estructurada en tuplas de la Wikipedia.

RDF Schema

- Si se utiliza RDF para representar el conocimiento de una ontología, habrán predicados que siempre estarán:
 - Que un objeto sea una clase.
 - Que un objeto sea una propiedad.
 - Que una clase sea subclase de otra.
 - Que una propiedad tenga como dominio una clase.
 - Que una propiedad tenga como rango una clase.
 - Que una propiedad sea subpropiedad de otra.
- RDF Schema fija un nombre estándar para cada uno de esos predicados primarios que estarán en cualquier ontología.
- De esta forma, una máquina comprenderá el conocimiento representado en esa ontología, pues la semántica de esos predicados ya están definidos por RDF Schema.
- Es un lenguaje extensible que proporciona los elementos básicos para crear ontologías (Vocabularios semánticos RDF)
- RDF Schema define el significado de los términos usados en triplets RDF
- Permite definir clases, relaciones entre clases, restricciones sobre propiedades, etc. Utilizando los siguientes IRIs de predicado prefijados:

rdfs:Class	Declarar recursos como clases para otros recursos.
rdfs:subClassOf	Define jerarquías (Relaciona clases con subclases)
rdfs:property	Define un subconjunto de RDFs que son propiedades.

rdfs:subPropertyOf	Define jerarquías de propiedades.
rdfs:domain	Dominio de una propiedad, aquellos recursos que aparecen como sujetos en las tripletas de ese predicado.
rdfs:range	Rango de una propiedad, aquellos recursos que aparecen como objetos en las tripletas de ese predicado.
rdfs:individual	Declarar un individuo de esa clase

Ejemplo:

- **Queremos introducir el concepto profesor, alumno, asignatura, y asignatura_optativa como subclase de asignatura**

(vamos a obviar que sería el IRI correspondiente en lugar de los nombres naturales que estamos usando)

```
profesor rdfs:Class ""
alumno rdfs:Class ""
asignatura rdfs:Class ""
asignatura_optativa rdfs:subClassOf asignatura
```

(fijaros que la tercera componente de las tres primeras líneas se deja vacía y lo he marcado con "" para indicar que es la cadena vacía)

- **Queremos introducir la propiedad es_profesor_de que es un predicado que relaciona individuos de la clase profesor con individuos de la clase asignatura**

```
es_profesor_de rdfs:property ""
es_profesor_de rdfs:domain profesor
es_profesor_de rdfs:range asignatura
```

- Ahora queremos introducir que JLC es profesor de la asignatura IC

JLC es_profesor_de IC

Fijaros que ahora el sistema, con lo introducido podría deducir por consistencia de la ontología:

JLC rdfs:individual profesor

IC rdfs:individual asignatura

Introduciendo conceptos, propiedades y relaciones no básicas

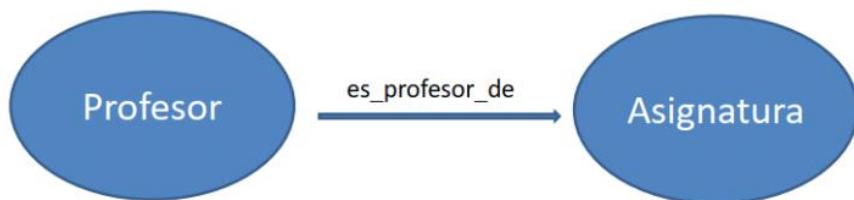
- RDF Schema nos sirve para declarar cosas básicas de las ontologías, y para hacer unas deducciones elementales como hemos visto
- Pero todavía es necesario introducir los axiomas, aquellas condiciones pueden ser complejas.
- Para ello se utilizará el lenguaje de lógica descriptiva OWL

OWL

- El Ontology Web Language extiende RDF Schema para permitir la expresión de relaciones complejas entre clases RDF Schema y mayor precisión en las restricciones de clases y propiedades.
 - Derivado de la fusión de lenguajes de ontologías DAML y OIL
 - Permite:
 - Expresar clases complejas
 - Expresar relaciones entre clases
 - Expresar y restringir clases (dominio o rango)
 - Expresar y restringir propiedades (cardinalidad)
 - Hacer afirmaciones complejas
- Posee tres variantes o sublenguajes que varían en su potencia expresiva
 - OWL-Lite: Versión simplificada, para representación de jerarquías simples.
 - OWL-DL: Incluye constructores tomados de Description Logics (DL), busca el compromiso entre máxima expresividad y eficiencia computacional, solo constructores decibiles de DL.
 - OWL-Full: Soporte completo de DL

Dominio y Rango de las propiedades

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



Constructores OWL

- Permiten nombrar clases no simples

intersectionof:	$C_1 \cap \dots \cap C_n$	Doctor \cap Mujer
unionof:	$C_1 \cup \dots \cup C_n$	Doctor \cup Abogado
complementof:	$\neg C$	\neg Hombre
oneof	$\{x_1\} \cup \dots \cup \{x_n\}$	{Juan} \cup {Maria}
AllValuesFrom	$\forall P.C$	\forall es_profesor_de.CSI
SomeValuesFrom	$\exists P.C$	\exists matriculado.CSI
maxCardinality	$\leq n.P$	≤ 1 .tieneHijo
minCardinality	$\geq n.P$	≥ 2 .tieneHijo

Ejemplos

- \forall es_Profesor_De.CSI → "La clase de los profesores con toda la docencia en la rama CSI o que no tienen docencia"
 - Siempre se restringe a la clase dominio de la propiedad
- \exists matriculado.CSI → "La clase de los alumnos que están matriculados en alguna asignatura de CSI"
- ≤ 1 .TieneHijo → "La clase de personas que tienen 1 hijo o menos"

Propiedades Derivadas

- Propiedad Inversa

inverseOf p^{-1} es_Profesor_De $^{-1}$

- Ejemplo

- \forall es_Profesor_De $^{-1}$.DECSAI → "La clase de asignaturas con todos los profesores que son de DECSAI"

Relaciones y Axiomas OWL

subClassof $C_1 \textcolor{red}{C} C_2$ Humanos $\textcolor{red}{C}$ Animales \cap Bipedos

EquivalentClass $C_1 \equiv C_2$ Doctor \equiv Tesis.{Si}

DisjointWith $C_1 \textcolor{red}{C} \neg C_2$ Matriculado.{TFG} $\textcolor{red}{C} \neg \exists$ Matriculado.cuarto

SameIndividualAs $\{x_1\} \equiv \{x_2\}$ {JLC} = {JuanLuisCastro}

DifferentFrom $\{x_1\} \textcolor{red}{C} \neg \{x_2\}$ {IC} $\textcolor{red}{C} \neg \{TSI\}$

subPropertyOf $P_1 \textcolor{red}{C} P_2$ tiene_hija $\textcolor{red}{C}$ tiene_hijos

equivalentProperty $P_1 \equiv P_2$ coste \equiv precio

inverseof $P_1 \equiv P_2^{-1}$ padre_de \equiv hijo_de $^{-1}$

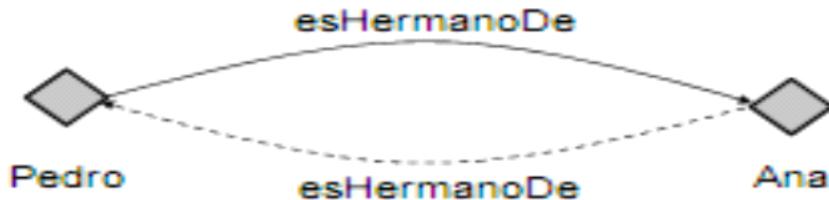
transitiveProperty $P^+ \textcolor{red}{C} P$ antecesor $^+$ $\textcolor{red}{C}$ antecesor

functionalProperty $T \leq 1.P$ $T \leq 1$.tieneMadre

inversefunctionalProperty $T \leq 1.P^-$ $T \leq 1$.tieneDNI $^-$

Propiedades simétricas

- P es simétrica: $P \equiv P^{-1}$



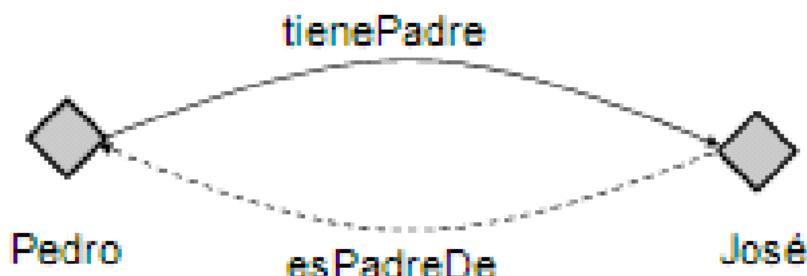
Si declaramos una propiedad P como simétrica, para cada declaración “S P O”, terna

S P O

el sistema añadirá (deducirá) “O P S”, la terna

O P S

Propiedades inversas: $P_1 \equiv P_2^{-1}$



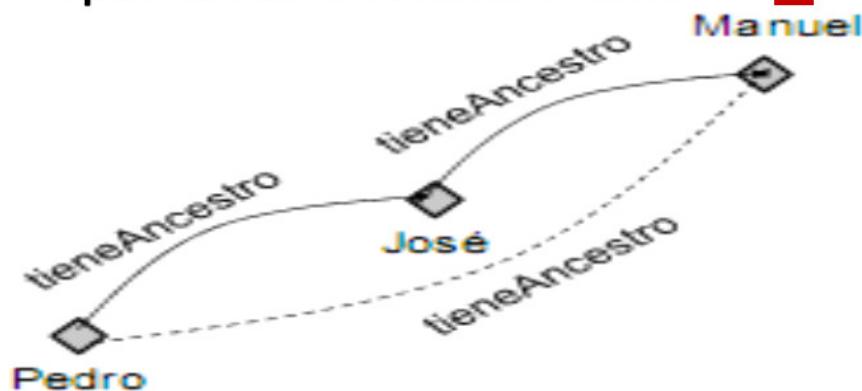
Si declaramos una propiedad P1 y P2 como inversas, para cada declaración “S P1 O”, terna

S P O

el sistema añadirá (deducirá) “O P2 S”, la terna

O P S

Propiedades transitivas: P+ C P



Si declaramos una propiedad P como transitiva, para cada par de declaraciones “S P O” y “O P O1, ternas

S P O

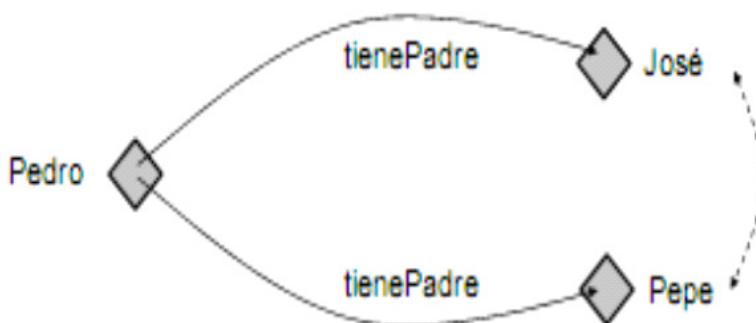
O P O1

el sistema añadirá (deducirá) “S P O1”, la terna

S P O1

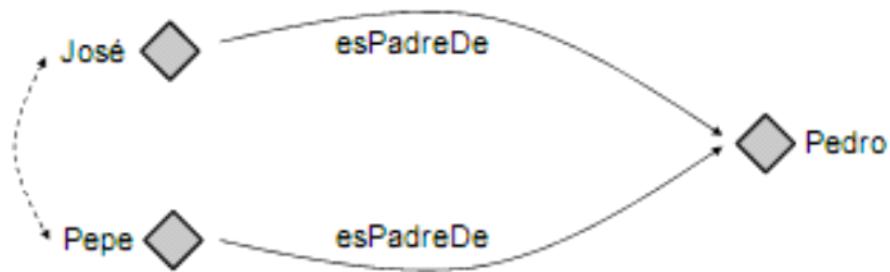
Propiedades funcionales

Cada individuo del dominio tiene esta relacionado con uno y solo uno del rango



Si declaramos una propiedad como funcional, si un individuo del dominio esta relacionado con dos elementos del rango, deduciremos que esos dos elementos son el mismo

Propiedades funcionales inversas



Si declaramos una propiedad como funcional inversa, si un individuo del rango esta relacionado con dos elementos del dominio, deduciremos que esos dos elementos son el mismo

- Se han desarrollado motores que implementan de forma automática el razonamiento sobre el conocimiento de una ontología
 - Deducen y rellenan los componentes que se pueden deducir según los axiomas indicados.
 - Dada una afirmación expresada en OWL, nos indica si ese axioma se verifica en la ontología.
- Algunos razonadores:
 - FaCT++
 - Pellet
 - Racer

Razonamiento con Ontologías

- Clasificación automática: Dado un concepto expresado en OWL, chequea si está incluido en algunas de las clases de ontología
- Clasificación de instancias: Dada una nueva instancia, deduce si pertenece a alguna de las clases de la ontología
- Detección de redundancia: Detecta si dos individuos o dos clases son la misma
- Chequeo de consistencia:
 - Disjoint: Detecta si hay elementos comunes en clases que deben ser distintas.
 - Restricciones: Detecta si algún axioma no se verifica.

Conclusiones

- Las ontologías...
 - Definen un 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 reutilizar el conocimiento.
 - Proveen conocimiento entendible por humanos y computadoras.
 - De una ontología solo nos debemos de preocupar de representar bien sus dominios, no es necesario indicar como razonar, para ello existe razonadores
- Para representar conocimiento y razonar con ontologías:
 1. Seleccionamos los conceptos y propiedades básicas del dominio y del problema.
 2. Establecemos las jerarquías de conceptos y definimos los axiomas relativos a cada una de las propiedades (¿Es funcional? ¿Inversa funcional? ¿Simétrica? ¿Transitiva?)
 3. Añadimos los axiomas más complejos expresados en OWL.
 4. Rellenamos los individuos de la ontología.
 5. Chequeamos la consistencia con el razonador y se deja que deduzca y se rellena lo que pueda

6. Planteamos las tareas a desarrollar en forma de consulta estándar del lenguaje OWL.