

Construcción de una ontología OWL con Protégé 4.3

Flavio E. Spetale

spetale@cifasis-conicet.gov.ar

Basado: T. Rodríguez y J. Aguilar

**A Practical Guide To Building OWL
Ontologies Using Protégé 4**



Construcción de ontología OWL

- Las ontologías son usadas para capturar el conocimiento sobre algún dominio de interés.
- Una ontología describe los conceptos dentro del dominio y la relación que tiene entre esos conceptos.
- Un lenguaje estándar para hacer ontologías es OWL desarrollado por W3C.
- OWL permite describir conceptos y además cuenta con un conjunto de operadores (intersección, unión, y negación).
- OWL esta basado en lógica descriptiva que permite el uso de un razonador.

Componentes de una ontología OWL

Ontologías	OWL	PROTÉGÉ
Instancias	Individuos	Casos (instance)
Relaciones	Propiedades	Slots
Conceptos	Clases	Clases

Individuos de una ontología OWL

Representan objetos del dominio de interés
y son también conocidos como instancias.



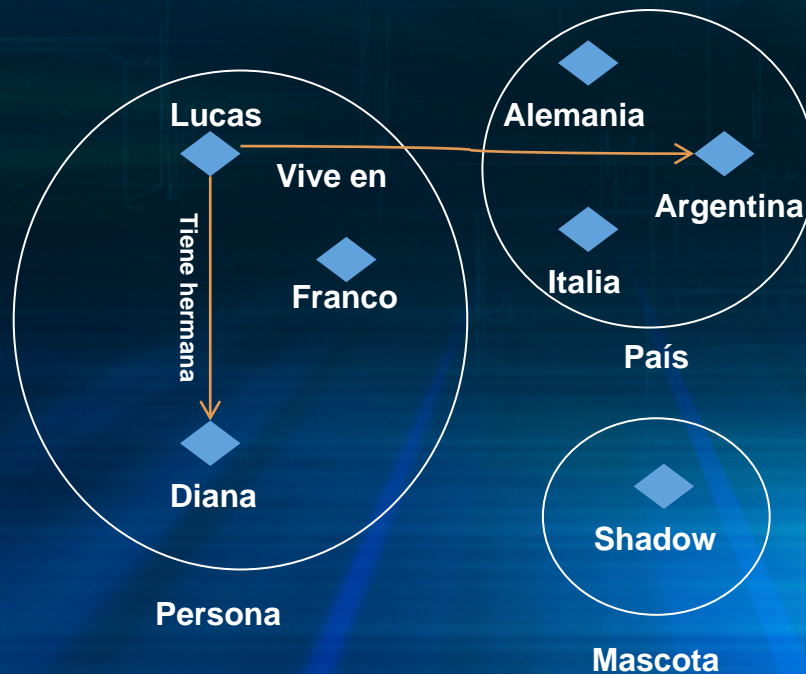
Propiedades de una ontología OWL

Son relaciones binarias sobre los individuos y pueden ser inversas, transitivas o simétricas.



Clases de una ontología OWL

Se entienden como conjuntos que contienen individuos y pueden ser organizadas dentro de una jerarquía de clases y subclases conocida como taxonomía.



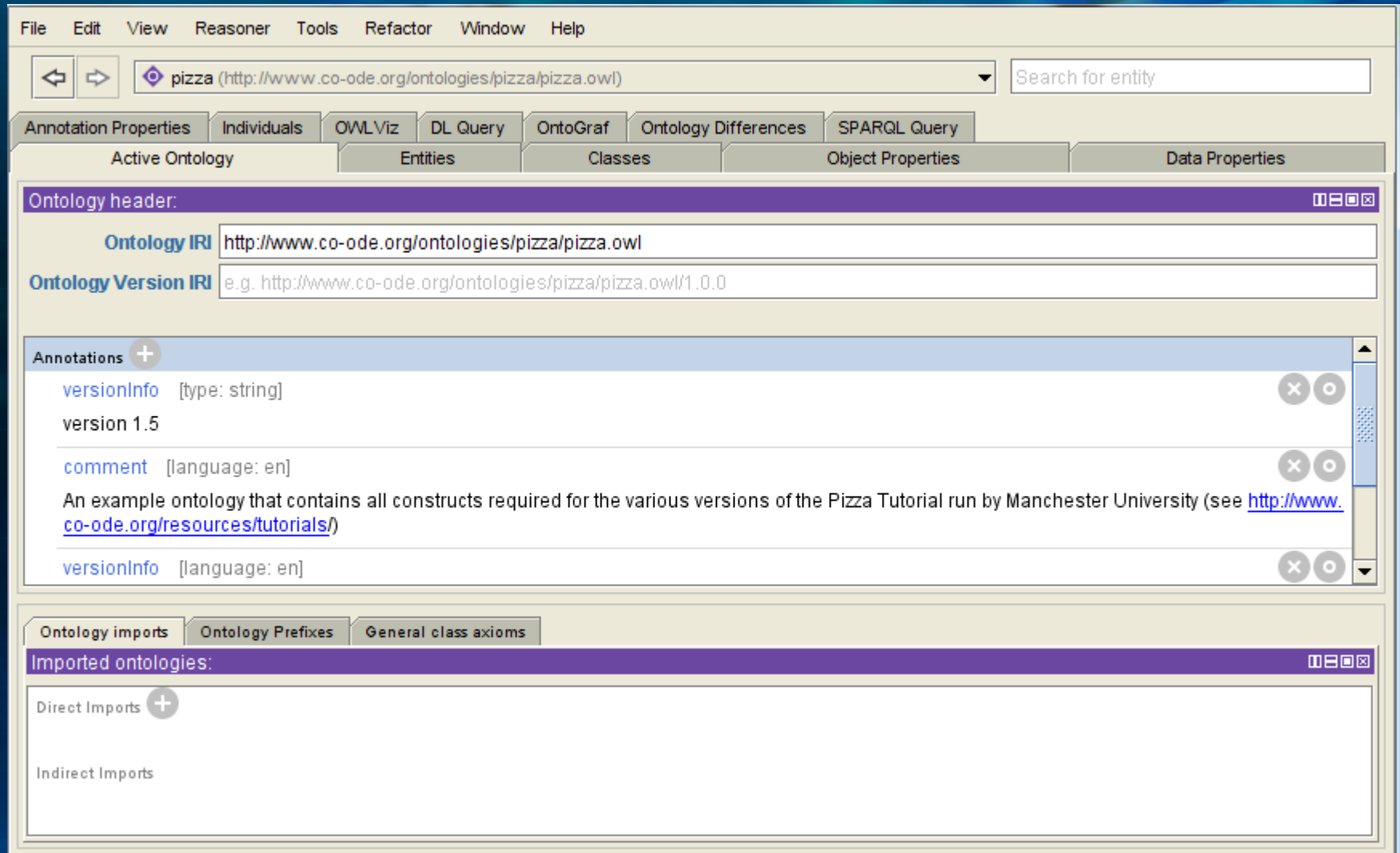
Correspondencia entre OWL y DL

Constructor OWL	Representación DL	Ejemplo
owl:equivalentTo (C,D)	$C \equiv D (C \sqsubseteq D \text{ y } D \sqsubseteq C)$	<i>Persona</i> \equiv <i>Humano</i>
rdfs:subClassOf (C,D)	$C \sqsubseteq D$	<i>Padres</i> \sqsubseteq <i>Persona</i>
owl:complementOf (C,D)	$C \equiv \neg D (\text{negacion})$	<i>Varon</i> $\equiv \neg$ <i>Mujer</i>
owl:disjointWith (C,D)	$C \sqsubseteq \neg D$	<i>Padre</i> $\sqsubseteq \neg$ <i>Madre</i>
owl:intersectionOf (C,D)	$C \sqcap D (\text{conjuncion})$	<i>Padres</i> \sqcap <i>Varon</i>
owl:unionOf (C,D)	$C \sqcup D (\text{disjuncion})$	<i>Padre</i> \sqcup <i>Madre</i>
owl:oneOf (I1, I2)	$\{I_1\} \sqcup \{I_2\}$	$\{Juan\} \sqcup \{Maria\}$
owl:someValuesFrom(P,C)	$\exists P.C (\text{existencial})$	$\exists \text{tieneHijo.Hija}$
owl:allValuesFrom(P,C)	$\forall P.C (\text{universal})$	$\forall \text{tieneHijo.Hijo}$
owl:hasValue (P,I1)	$\exists P.\{I_1\}$	$\exists \text{tieneHijo.}\{Juan\}$
owl:cardinality(P,n)	$= n.P$	$= 2.\text{tienePadres}$
owl:minCardinality(P,n)	$\geq n.P$	$\geq 1.\text{tieneHija}$
owl:maxCardinality(P,n)	$\leq n.P$	$\leq 2.\text{tieneHijos}$

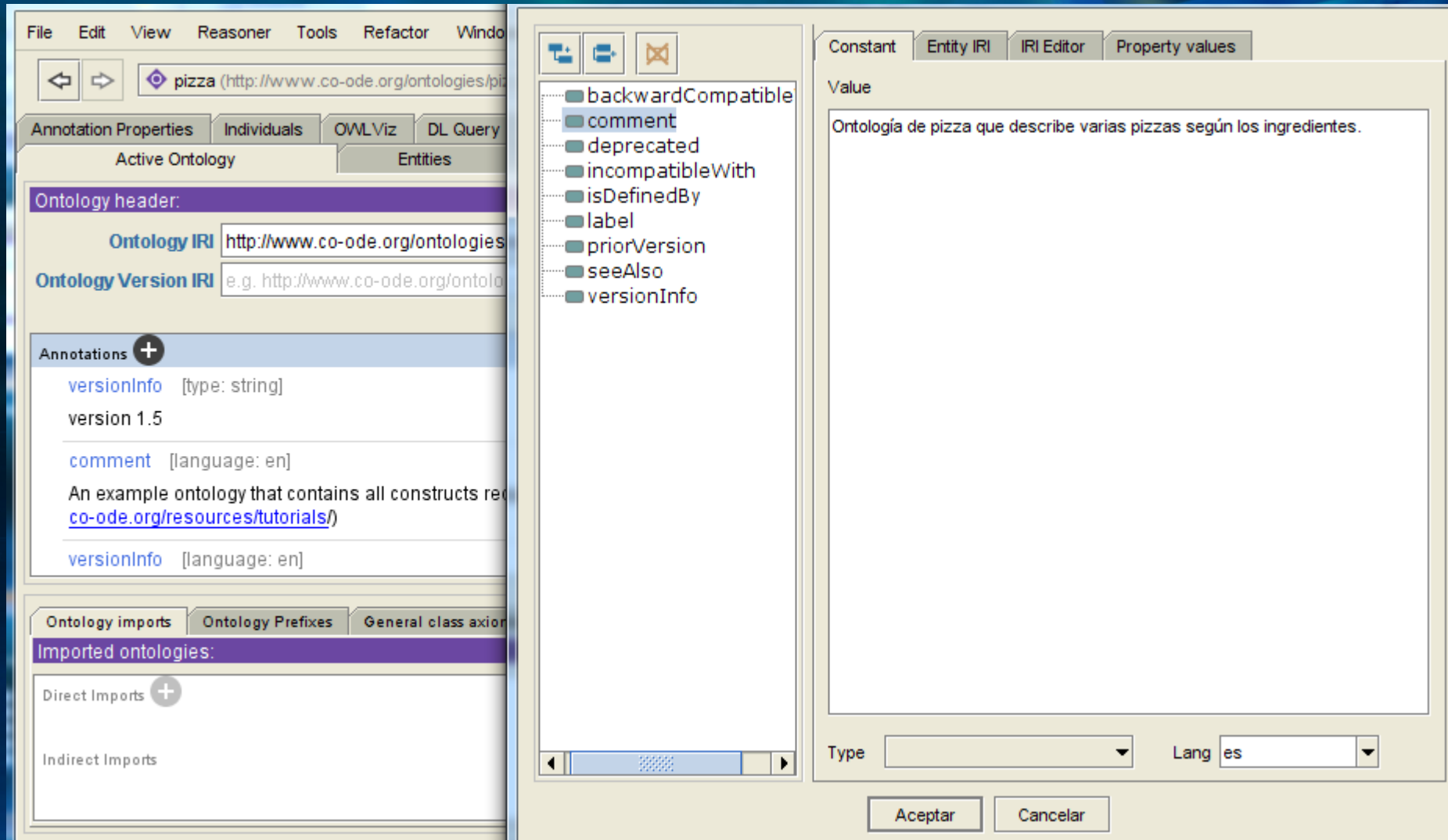
Un concepto en DL se refiere a una clase en OWL.

Un rol en DL es una propiedad en OWL.

Interfaz de Protégé



Comentarios en Protégé



Clases en Protégé

The screenshot displays the Protégé ontology editor interface. At the top, a menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. Below the menu, a toolbar shows navigation icons and a dropdown menu currently set to 'untitled-ontology-7' with the URL <http://www.semanticweb.org/flavio/ontologies/2014/4/untitled-ontology-7>. A search bar labeled 'Search for entity' is positioned to the right of the dropdown.

The main workspace is divided into several panes. On the left, the 'Class hierarchy' pane shows a tree structure with 'Thing' as the root class. The right pane is split into 'Annotations' and 'Usage' tabs, with 'Annotations: Thing' currently active. Below this, the 'Description: Thing' pane lists various class relationships with plus icons for expansion: SubClass Of, SubClass Of (Anonymous), Members, Target for Key, Disjoint With, and Disjoint Union Of.

A modal dialog box titled 'Create a new OWLClass' is open in the foreground. It contains the following fields and controls:

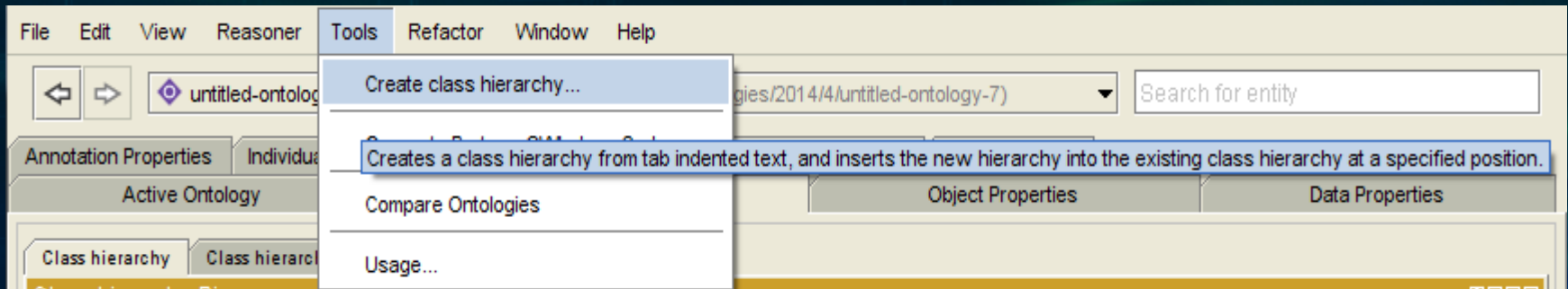
- Name:** A text field containing the word 'Pizza'.
- IRI:** A text field containing the URL <http://www.semanticweb.org/flavio/ontologies/2014/4/untitled-ontology-7#Pizza>.
- New entity options...** A button located to the right of the IRI field.
- Acceptar** and **Cancelar** buttons at the bottom of the dialog.

At the bottom of the Protégé window, a status bar contains the text 'To use the reasoner click Reasoner->Start reasoner' and a checked checkbox labeled 'Show Inferences'.

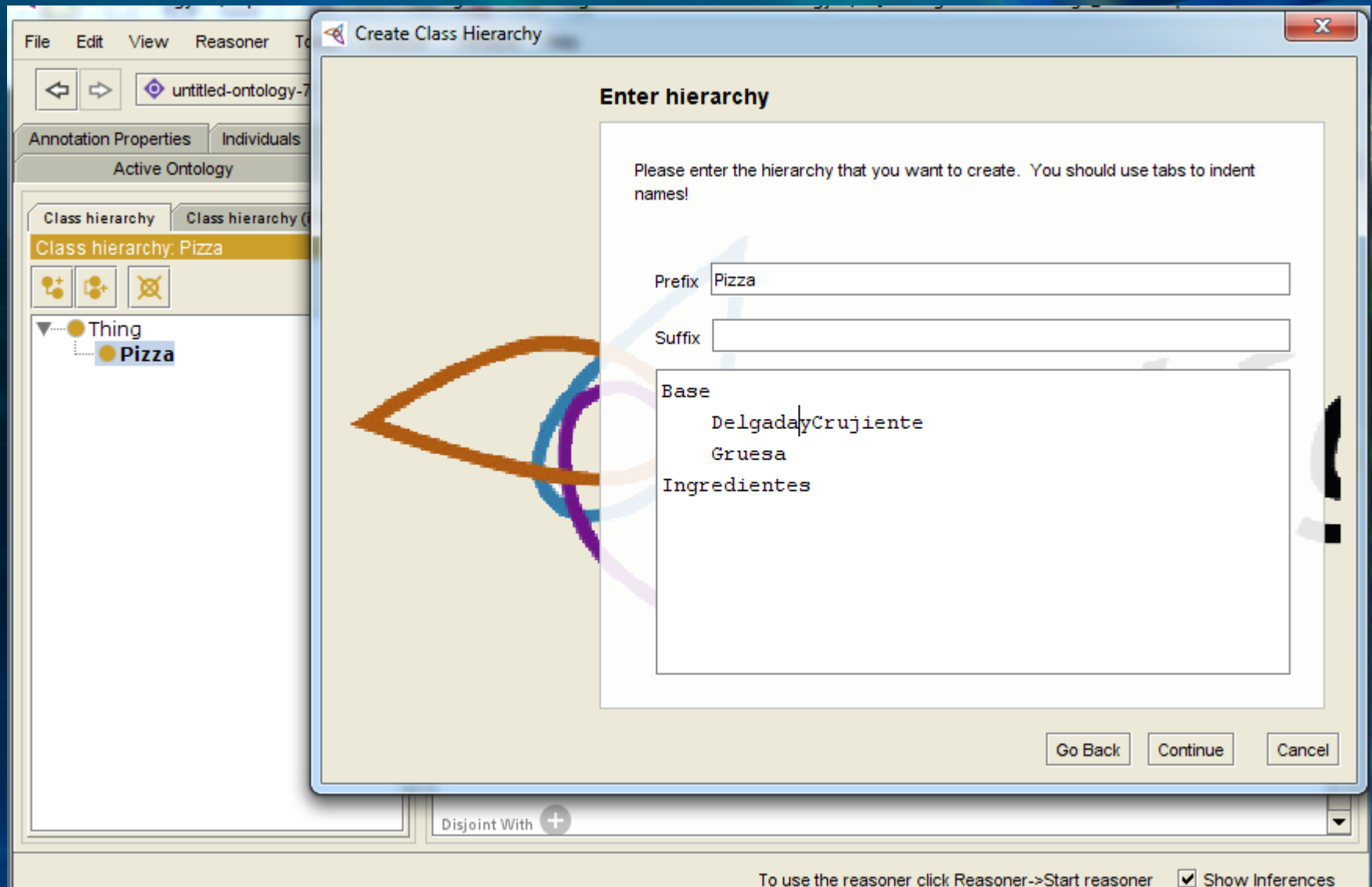
Clases en Protégé

Para generar clases y subclases en un solo paso se utiliza la herramienta:

Create class hierarchy



Clases en Protégé



Clases en Protégé

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. Below the menu is a toolbar with navigation icons and a dropdown menu showing the current ontology: 'untitled-ontology-7 (http://www.semanticweb.org/flavio/ontologies/2014/4/untitled-ontology-7)'. A search bar labeled 'Search for entity' is located to the right of the dropdown.

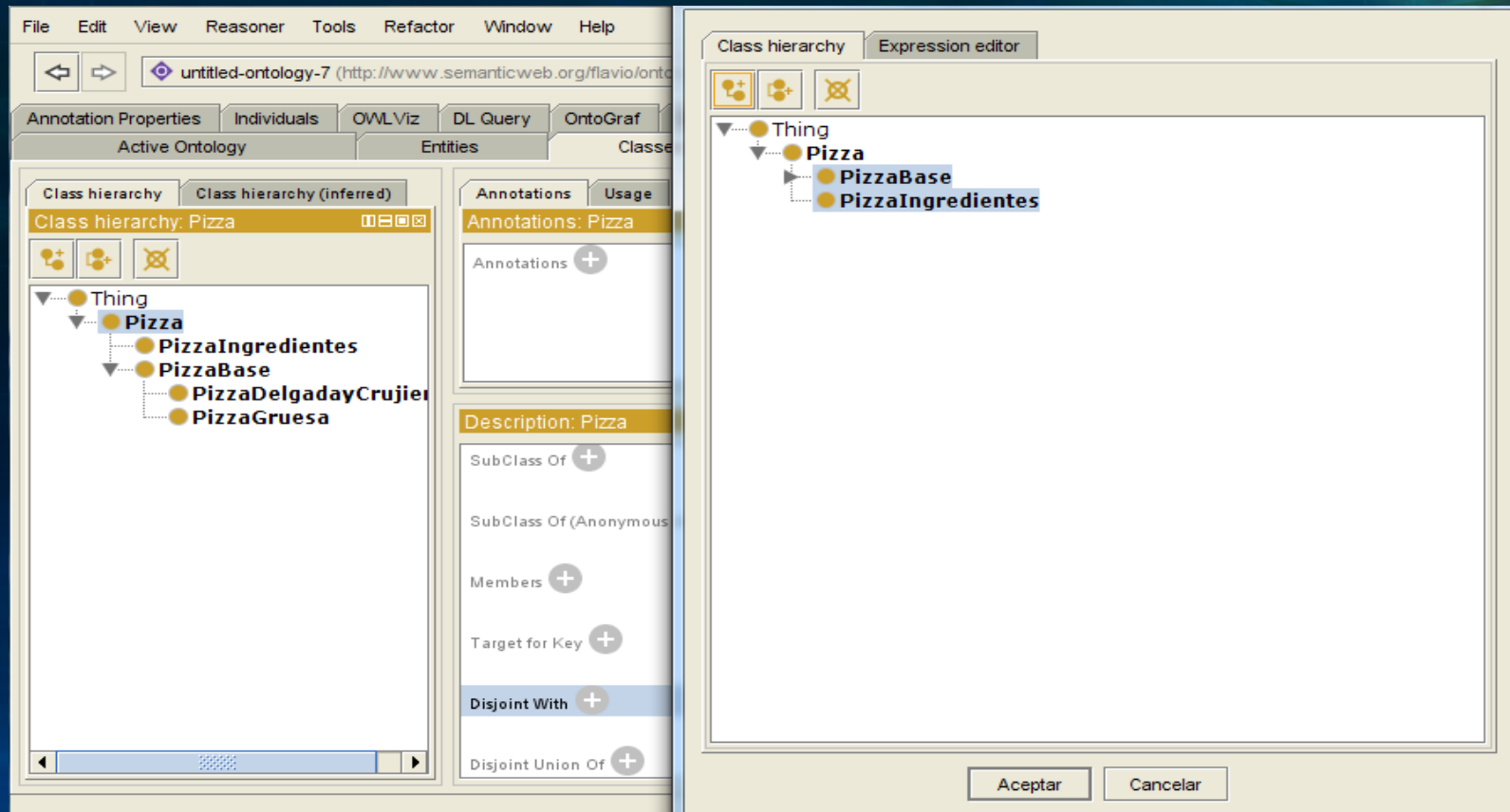
The main workspace is divided into several tabs: Annotation Properties, Individuals, OWL Viz, DL Query, OntoGraf, Ontology Differences, and SPARQL Query. Below these tabs are two rows of buttons: 'Active Ontology', 'Entities', 'Classes', 'Object Properties', and 'Data Properties'.

The left pane shows the 'Class hierarchy' tab, displaying a tree structure under 'Thing'. The hierarchy includes 'Pizza', which is expanded to show its subclasses: 'PizzaIngredientes', 'PizzaBase', 'PizzaDelgadayCrujiente', and 'PizzaGruesa'. The right pane shows the 'Annotations' and 'Usage' tabs for the selected class 'Pizza'. The 'Annotations' tab is active, showing a list of annotations with a '+' button to add new ones. Below this, the 'Description: Pizza' tab is visible, showing a list of properties with '+' buttons to add new ones: 'Equivalent To', 'SubClass Of', 'SubClass Of (Anonymous Ancestor)', 'Members', 'Target for Key', and 'Disjoint With'.

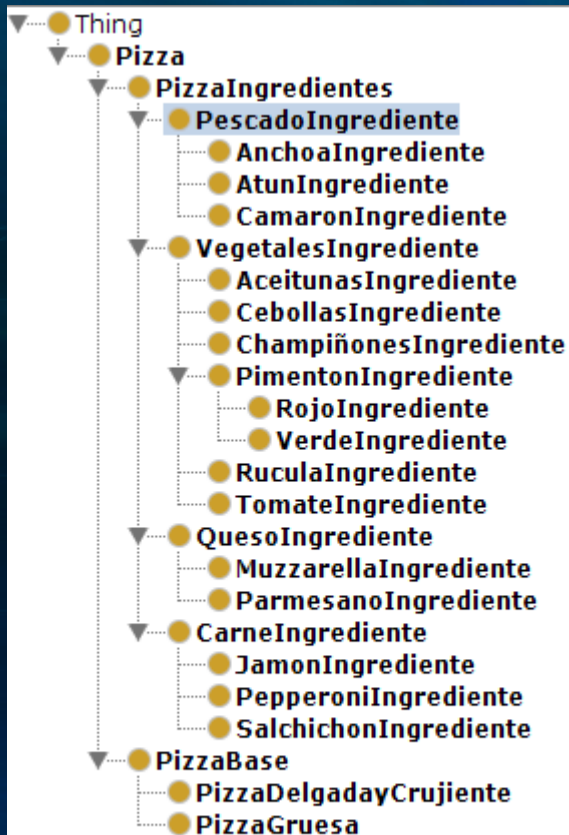
At the bottom of the interface, a status bar contains the text: 'To use the reasoner click Reasoner->Start reasoner' and a checkbox labeled 'Show Inferences' which is currently checked.

Clases Disjuntas en Protégé

Después de adicionar varias clases en la jerarquía de la ontología, se requiere establecer clases disjuntas, que indican que un objeto o individuo no puede ser instancia de más de una de estas clases que se establecieron disjuntas.



Clases Disjuntas en Protégé



Description: PescadoIngrediente

Equivalent To +

SubClass Of +

PizzaIngredientes

SubClass Of (Anonymous Ancestor)

Members +

Target for Key +

Disjoint With +

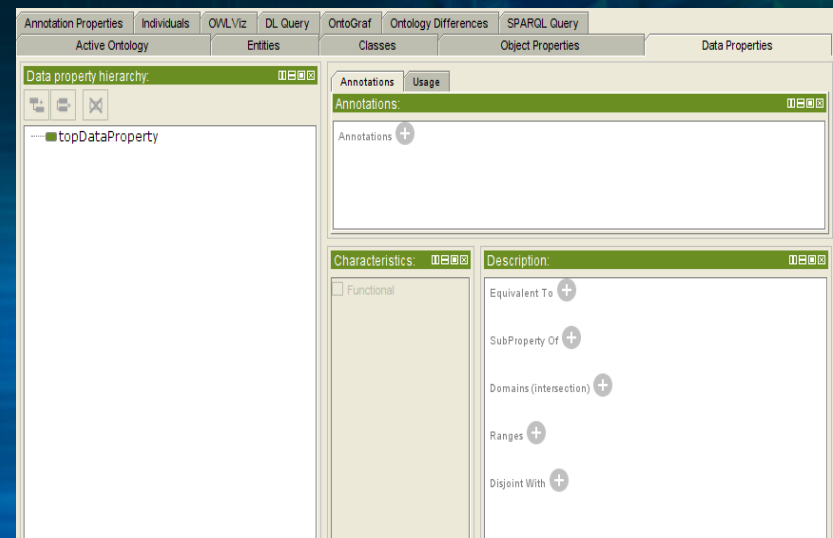
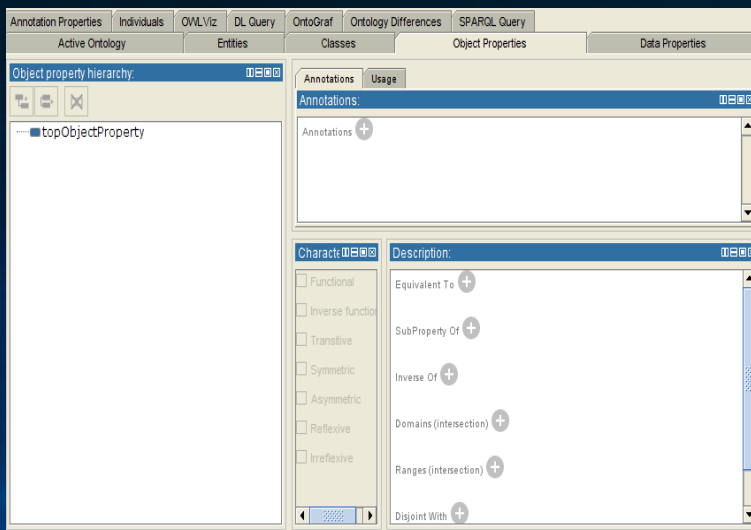
CarneIngrediente, QuesoIngrediente, VegetalesIngrediente

Disjoint Union Of +

Propiedades OWL en Protégé

Existen dos tipos de propiedades en OWL:

- ☛ “ObjectProperties”, que permite relacionar un individuo con otro
- ☛ “DatatypeProperties”, que relaciona un individuo con un XML Schema Datatype value o un literal RDF



Propiedades OWL en Protégé

Los elementos que debe tener un ObjectProperty son:

Nombre

Dominio: hace referencia a la clase o clases iniciales

Rango: hace referencia a la clase o clases finales.

Ejemplo la relación: es profesor

Nombre: es_profesor

Dominio: Docente

Rango: Estudiante

Propiedades OWL en Protégé

The screenshot displays the Protégé OWL editor interface. The top menu bar includes: Annotation Properties, Individuals, OWLViz, DL Query, OntoGraf, Ontology Differences, SPARQL Query, and a tabbed area with Active Ontology, Entities, Classes, Object Properties, and Data Properties.

The left pane, titled "Object property hierarchy: TieneBase", shows a tree structure:

- topObjectProperty
 - Tiene_Ingrediente
 - TieneIngredientes
 - TieneBase

The right pane is divided into two sections:

- Annotations: TieneBase**: Contains a single "Annotations" entry with a plus icon for adding more.
- Description: TieneBase**: Contains several property relationship sections, each with a plus icon for adding a value:
 - Equivalent To
 - SubProperty Of: **Tiene_Ingrediente** (with icons for ? @ x o)
 - Inverse Of
 - Domains (intersection)
 - Ranges (intersection)

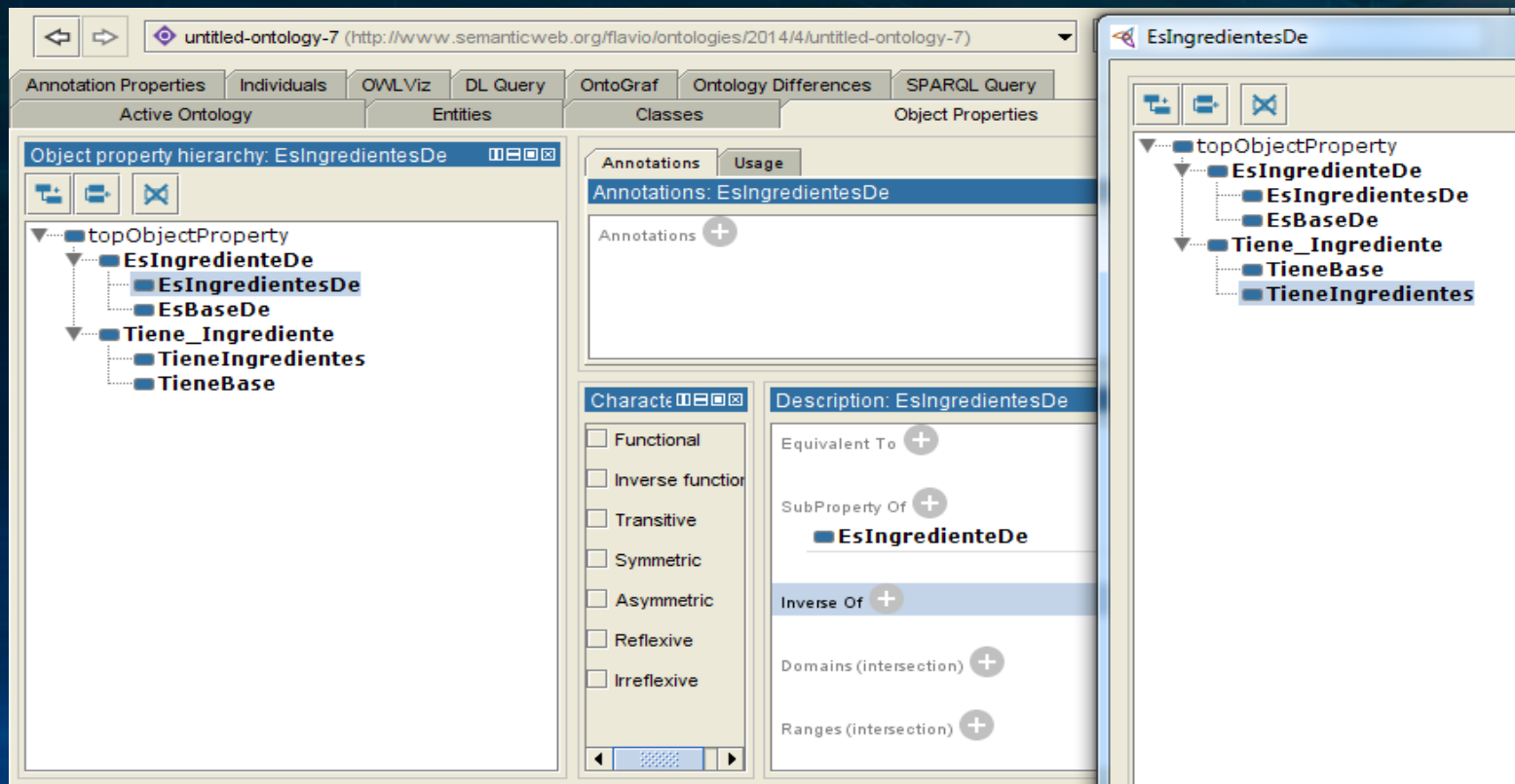
Below the Description pane is a "Characteristics" section with a list of checkboxes:

- ☐ Functional
- ☐ Inverse function
- ☐ Transitive
- ☐ Symmetric
- ☐ Asymmetric
- ☐ Reflexive
- ☐ Irreflexive

Propiedades Inversas en Protégé

Cada *ObjectProperty* debe tener su correspondiente propiedad inversa.

Si una propiedad enlaza un objeto A con otro B, entonces la propiedad inversa enlaza el objeto B con el A.



Características de las propiedades en Protégé

OWL permite que el significado de las propiedades sea enriquecido con características de las propiedades

- ☛ Funcional
- ☛ Funcional Inversa
- ☛ Simétrica
- ☛ Transitiva

Propiedad Funcional en Protégé

Define que a lo sumo un objeto puede estar relacionado con otro objeto.

Ejemplo, si se tienen tres objetos que son A, B y C y se tiene una propiedad funcional *tienePadre*, entonces se podrían asociar los objetos A y B por medio de la propiedad y daría como resultado A *tienePadre* B. Igualmente se podrían asociar los objetos A y C por medio de la propiedad y daría como resultado A *tienePadre* C. Como *tienePadre* es propiedad funcional, *se concluye que B y C son el mismo objeto*. En caso contrario estaríamos en una contradicción.

Propiedad Funcional Inversa en Protégé

Indica que puede estar a lo sumo un objeto relacionado con otro.

Ejemplo, si se tienen tres objetos que son A, B y C y se tiene una propiedad funcional **esPadreDe**, entonces se podría asociar el objeto B y A por medio de la propiedad y daría como resultado B **esPadreDe** A. Igualmente se podrían asociar los objetos C y A por medio de la propiedad y daría como resultado C **esPadreDe** A. Como **esPadreDe** es propiedad funcional inversa, se concluye que B y C son el mismo objeto.

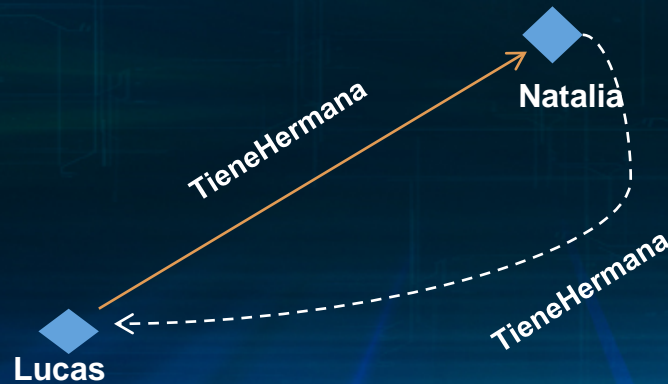
Propiedad Transitiva en Protégé

Relaciona dos objetos A y B, y además hay una propiedad que relaciona al objeto B con otro C, entonces se puede inferir que el objeto A está relacionado con el objeto C mediante la propiedad transitiva.

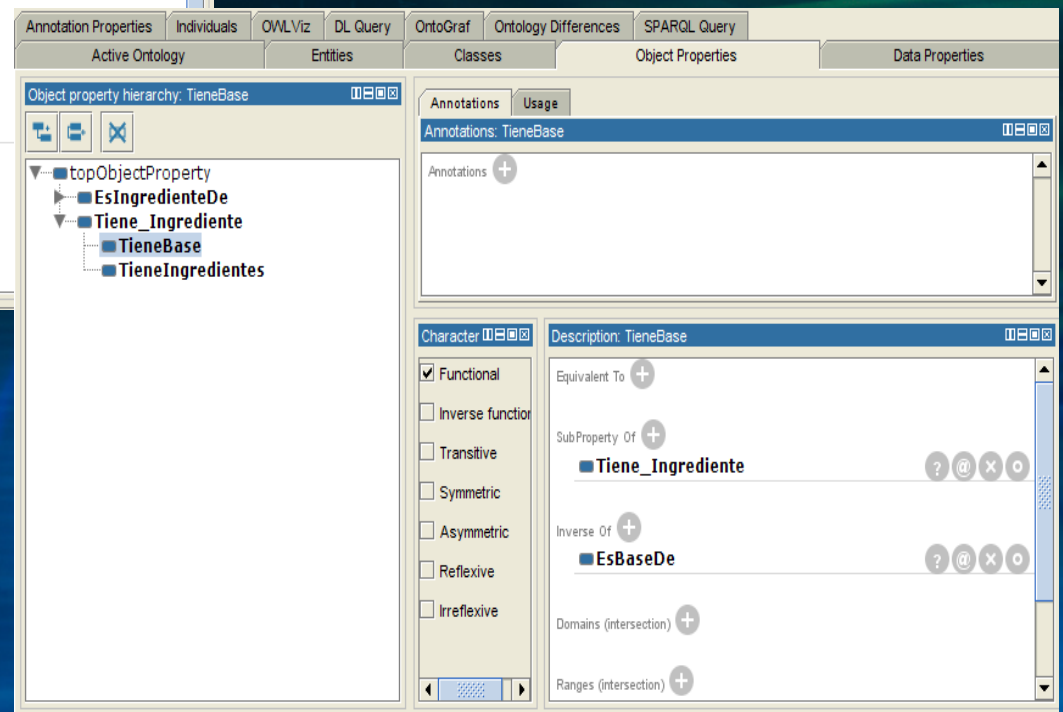
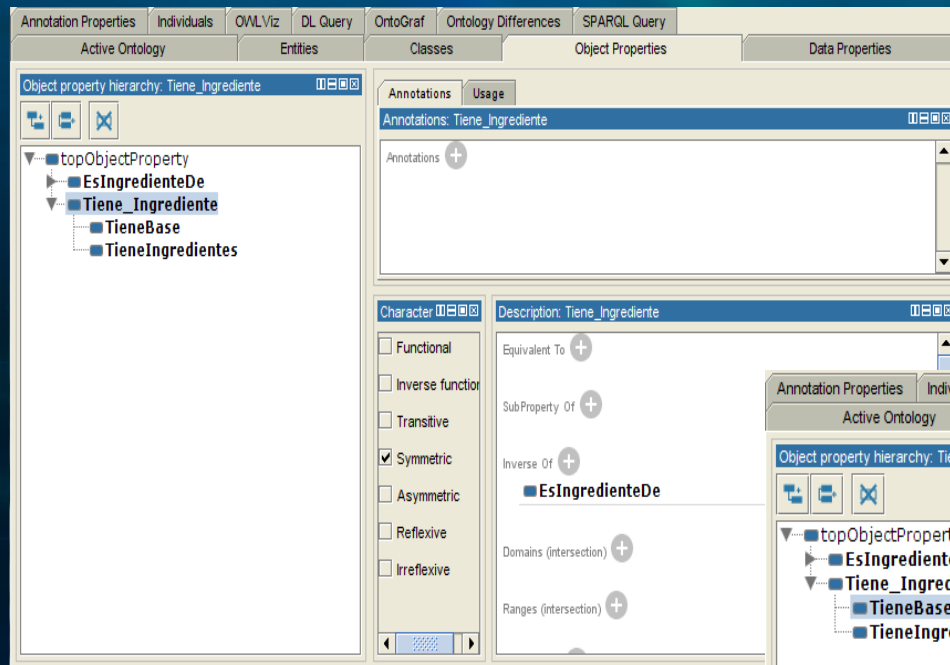


Propiedad Simétrica en Protégé

Relaciona a los objetos A y B, entonces el objeto B es relacionado por medio de la propiedad P con el objeto A.



Propiedades en Protégé



Rango de una Propiedad en Protégé

The screenshot displays the Protégé ontology editor interface. The top menu bar includes 'Annotation Properties', 'Individuals', 'OWL Viz', 'DL Query', 'OntoGraf', 'Ontology Differences', and 'SPARQL Query'. Below this, the 'Active Ontology' section shows 'TieneBase' as the active ontology. The 'Object Properties' tab is selected, showing a hierarchy of object properties: 'topObjectProperty' (expanded) contains 'EsIngredienteDe', which contains 'Tiene_Ingrediente', which contains 'TieneBase' (selected) and 'TieneIngredientes'.

The 'TieneBase' property configuration window is open, showing the 'Object restriction creator' and 'Data restriction creator' tabs. The 'Class expression editor' is active, showing a hierarchy of classes: 'Thing' (expanded) contains 'Pizza', which contains 'PizzaBase' (selected) and 'PizzaIngredientes'.

The 'Character' tab is selected, showing the following properties:

- ☒ Functional
- ☐ Inverse function
- ☐ Transitive
- ☐ Symmetric
- ☐ Asymmetric
- ☐ Reflexive
- ☐ Irreflexive

The 'Description: TieneBase' tab is also visible, showing the following configuration:

- SubProperty Of: **Tiene_Ingrediente**
- Inverse Of: **EsBaseDe**
- Domains (intersection):
- Ranges (intersection):
- Disjoint With:

The 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons are visible at the bottom of the configuration window.

Dominio de una Propiedad en Protégé

The screenshot displays the Protégé ontology editor interface with several panels open:

- Top Panel:** Contains tabs for 'Annotation Properties', 'Individuals', 'OWL Viz', 'DL Query', 'OntoGraf', 'Ontology Differences', and 'SPARQL Query'. Below these are sub-tabs for 'Active Ontology', 'Entities', 'Classes', 'Object Properties', and 'Data Properties'.
- Object property hierarchy: TieneBase:** A tree view showing the hierarchy of object properties. The selected path is: `topObjectProperty` > `EsIngredienteDe` > `Tiene_Ingrediente` > `TieneBase`.
- TieneBase (Dialog Box):** A dialog box for editing the property. It has tabs for 'Class expression editor', 'Class hierarchy', 'Object restriction creator', and 'Data restriction creator'. The 'Class hierarchy' tab is active, showing a tree with `Thing` as the parent and `Pizza` as a child.
- Character:** A panel with checkboxes for property characteristics: ☒ Functional, ☐ Inverse function, ☐ Transitive, ☐ Symmetric, ☐ Asymmetric, ☐ Reflexive, and ☐ Irreflexive.
- Annotations: TieneBase:** A panel for adding annotations to the property.
- Description: TieneBase:** A panel showing the logical description of the property. It includes:
 - Subproperty of:** `Tiene_Ingrediente`
 - Inverse Of:** `EsBaseDe`
 - Domains (intersection):** (Empty)
 - Ranges (intersection):** `PizzaBase`
 - Disjoint With:** (Empty)

At the bottom of the interface, there is a status bar with the text: 'To use the reasoner click Reasoner->Start reasoner' and a checked checkbox for 'Show Inferences'.

Restricciones de una Propiedad en Protégé

Las propiedades son utilizadas para crear restricciones en las clases en una ontología OWL.

Usualmente el nombre de la propiedad debería sugerir las restricciones impuestas a los objetos de la clase.

Las restricciones OWL se presentan en las siguientes tres categorías:

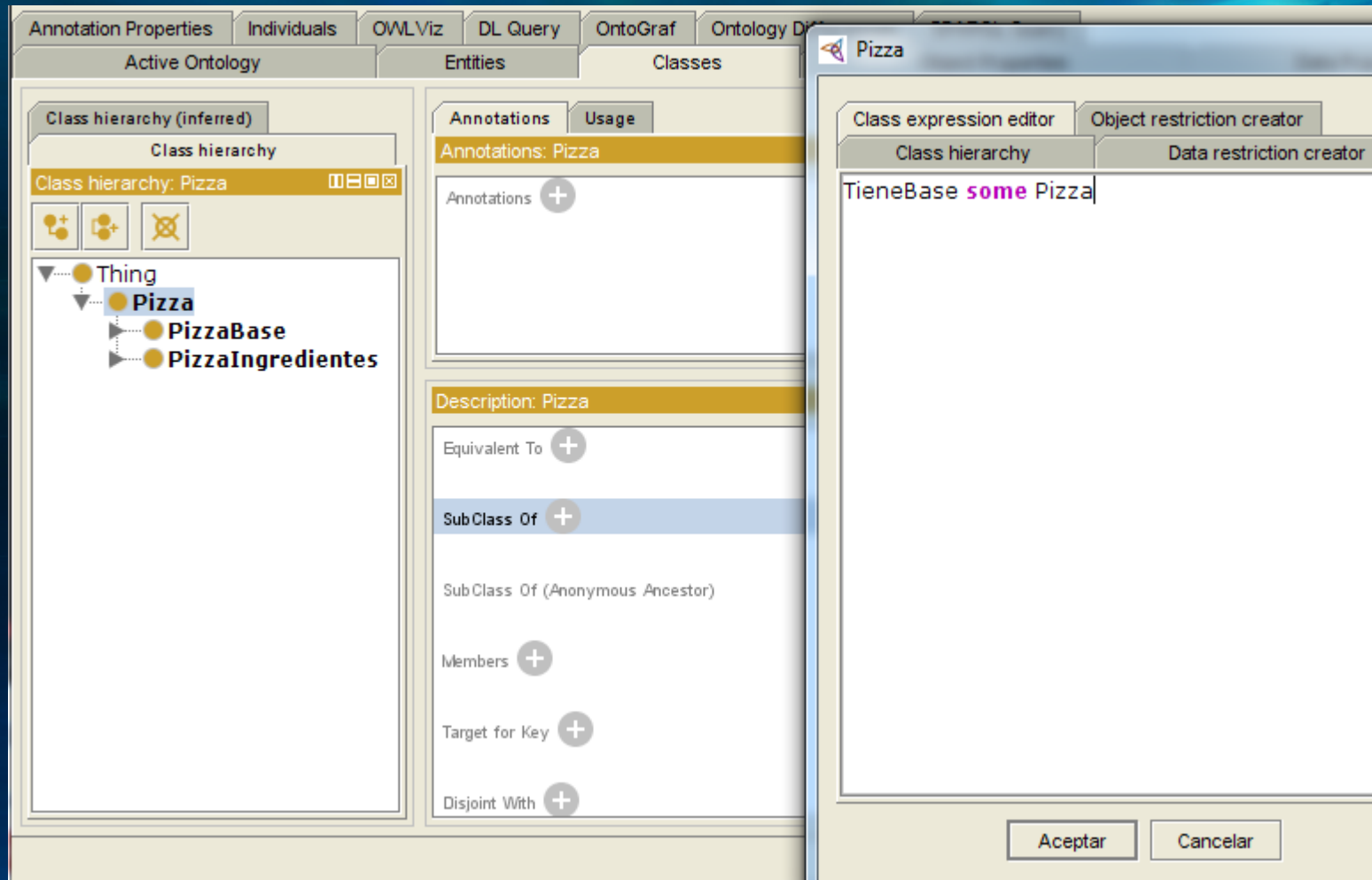
- ☛ Restricciones de cuantificación.
- ☛ Restricciones de cardinalidad.
- ☛ Restricciones de valor.

Restricciones de una Propiedad en Protégé

Cuantificador existencial (\exists), el cual permite indicar la existencia de al menos un objeto. En Protégé 4 la palabra clave **some** es usado para denotar \exists .

Cuantificador universal (\forall), el cual permite indicar la existencia de todos los objetos. En Protégé 4. la palabra clave es **only** es usado para denotar \forall .

Restricciones de una Propiedad en Protégé



Clase inconsistente en Protégé

Clasificando con el razonador Pellet, Fact++ o Hermit

The screenshot displays the Protégé ontology editor interface. The 'Reasoner' menu is open, showing options: 'Start reasoner' (Ctrl-R), 'Synchronize reasoner', 'Stop reasoner', 'Explain inconsistent ontology', 'Configure...', 'FaCT++', 'Hermit 1.3.8', 'Pellet' (selected), 'Pellet (Incremental)', and 'None'. The 'Class hierarchy (inf)' panel on the left shows a tree structure starting with 'Thing', which has a child 'Pizza'. 'Pizza' has children 'PizzaAmericana', 'PizzaMargarita', and 'PizzaTopping'. 'PizzaTopping' has children 'CarneTopping', 'PescadoTopping', and 'QuesoTopping'. 'QuesoTopping' has children 'PruebaIncoTopping' (highlighted in red), 'MozarellaTopping', and 'ParmesanoTopping'. 'VegetablesTopping' is also listed under 'PizzaTopping'. 'PizzaBase' has children 'BaseDelgadayCrujiente' and 'BaseGruesa'. The right panel shows the 'Annotations: Thing' and 'Description: Thing' tabs, with the 'Description: Thing' tab active, displaying sections for 'Equivalent classes', 'Superclasses', 'Inferred anonymous superclasses', 'Members', and 'Disjoint classes'.

Condición necesaria y suficiente en Protégé

The screenshot displays the Protégé ontology editor interface. The top menu bar includes options like Annotation Properties, Individuals, OWL Viz, DL Query, OntoGraf, Ontology Differences, and SPARQL Query. Below this, a tabbed interface shows the 'Active Ontology' with sub-tabs for Entities, Classes, Object Properties, and Data Properties.

On the left, the 'Class hierarchy (inferred)' panel shows a tree structure starting from 'Thing'. Under 'Thing', there is a class 'Pizza', which is a subclass of 'Cantimpalo'. 'Cantimpalo' has several subclasses: 'NombrePizza' (with subclasses 'Americana' and 'Margarita'), 'PizzaBase', 'PizzaIngredientes', 'CarneIngrediente', 'PescadoIngredien', 'QuesoIngrediente' (with subclasses 'Prueba', 'MuzzarellaIngr', and 'ParmesanoIngr'), and 'VegetalesIngredie'.

On the right, the 'Annotations' and 'Usage' tabs are active. The 'Annotations' tab shows 'Annotations: Cantimpalo' with a list of annotations. The 'Usage' tab shows 'Description: Cantimpalo' with a list of descriptions. The first description is 'Equivalent To' followed by 'Pizza and TieneIngredientes some QuesoIngrediente'. The second description is 'Sub Class Of' followed by 'TieneBase some Pizza'.

Condición necesaria y suficiente en Protégé

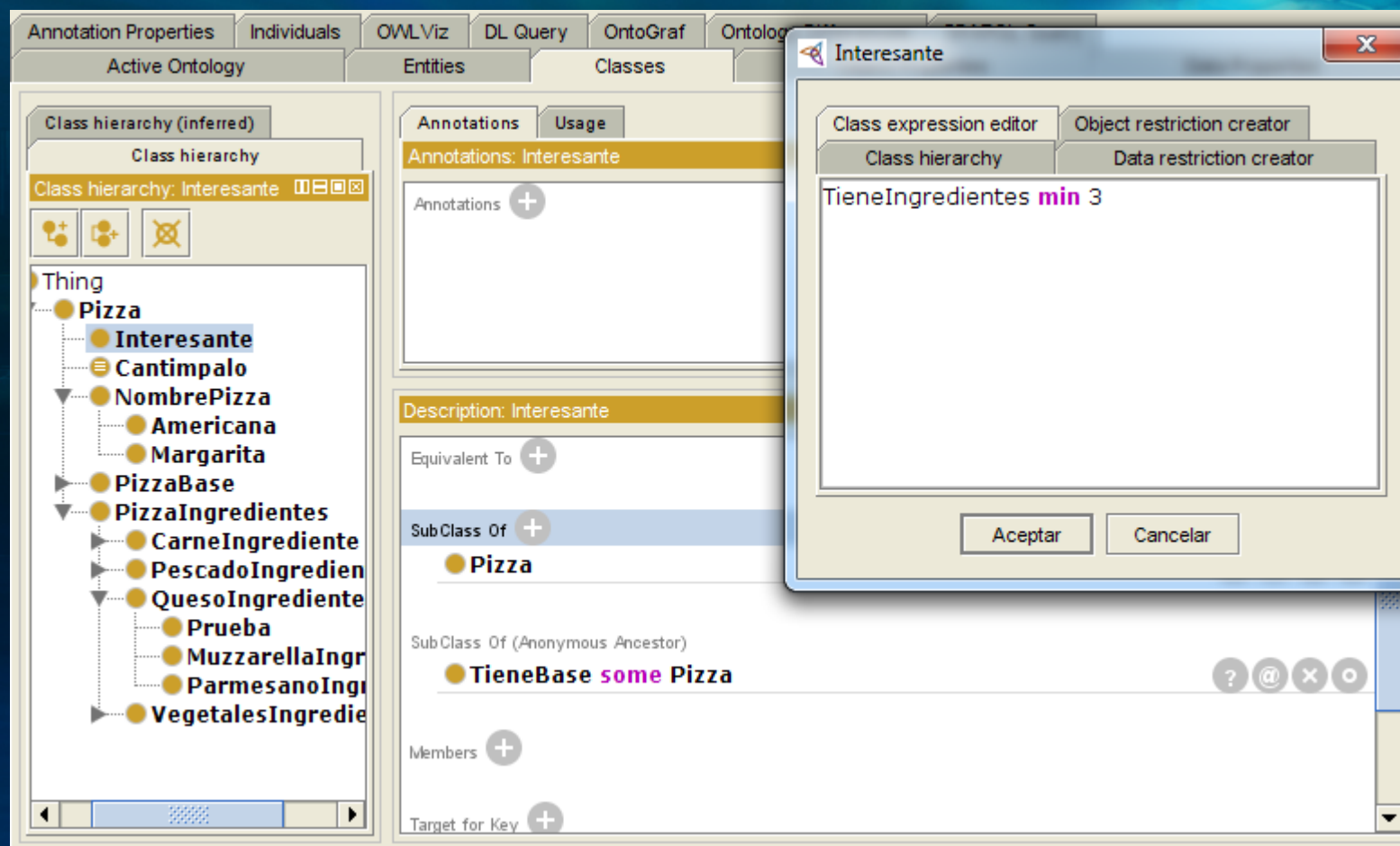
The screenshot displays the Protégé ontology editor interface. The top menu bar includes options like Annotation Properties, Individuals, OWL Viz, DL Query, OntoGraf, Ontology Differences, and SPARQL Query. Below this, a tabbed interface shows the 'Active Ontology' with sub-tabs for Entities, Classes, Object Properties, and Data Properties.

On the left, the 'Class hierarchy (inferred)' panel shows a tree structure starting from 'Thing'. Under 'Thing', there is a class 'Pizza', which is a subclass of 'Cantimpalo'. 'Cantimpalo' has several subclasses: 'NombrePizza' (with subclasses 'Americana' and 'Margarita'), 'PizzaBase', 'PizzaIngredientes' (with subclasses 'CarneIngrediente', 'PescadoIngrediente', 'QuesoIngrediente', 'Prueba', 'MuzzarellaIngr', and 'ParmesanoIngr'), and 'VegetalesIngrediente'.

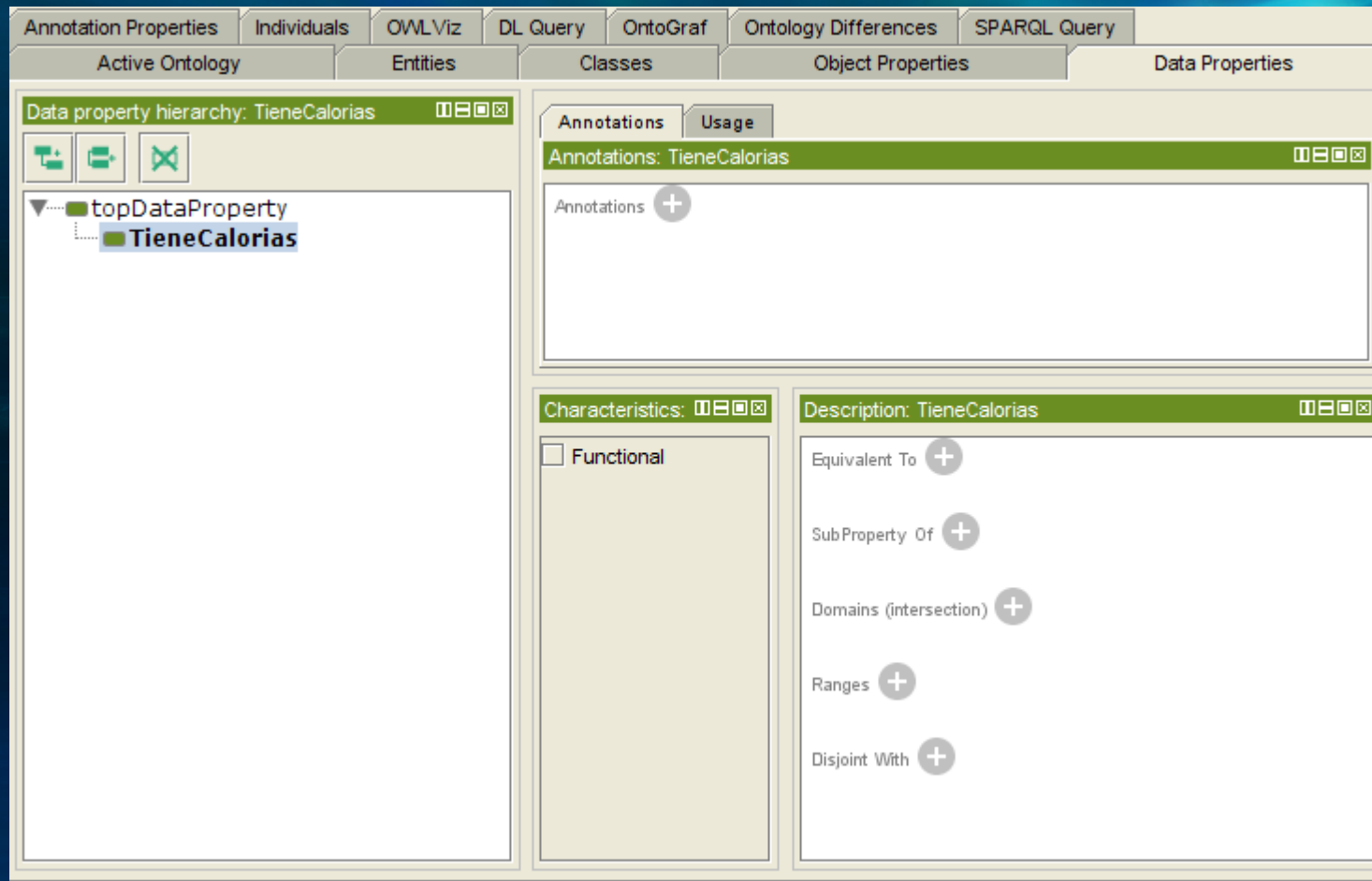
The right panel shows the 'Annotations' and 'Usage' tabs. The 'Annotations' tab is active, displaying 'Annotations: Cantimpalo'. Below this, the 'Description: Cantimpalo' tab is active, showing the following description:

- Equivalent To: **Pizza** and **TieneIngredientes** some **QuesoIngrediente**
- Sub Class Of: **TieneBase** some **Pizza**
- Sub Class Of (Anonymous Ancestor): **TieneBase** some **Pizza**
- Members: (empty)
- Target for Key: (empty)

Restricción de Cardinalidad en Protégé



Propiedad Datatype en Protégé



Propiedad Datatype en Protégé

The screenshot displays the Protégé ontology editor interface. The top menu bar includes 'Annotation Properties', 'Individuals', 'OWL Viz', 'DL Query', 'OntoGraf', 'Ontology Differences', and 'SPARQL Query'. Below this, the 'Active Ontology' tab is selected, showing 'Entities', 'Classes', 'Object Properties', and 'Data Properties'.

On the left, the 'Class hierarchy (inferred)' panel shows a tree structure starting from 'Thing'. The 'Pizza' class is highlighted, with its subclasses listed: 'Interesante', 'Cantimpalo', 'NombrePizza', 'Americana', 'Margarita', 'PizzaBase', 'PizzaIngredientes', 'CarneIngredientes', 'PescadoIngredientes', 'QuesoIngredientes', 'Prueba', 'MuzzarellaIngredientes', 'ParmesanoIngredientes', and 'VegetalesIngredientes'.

The central panel shows the 'Annotations: Pizza' and 'Description: Pizza' tabs. The 'Description: Pizza' tab is active, displaying 'Equivalent To', 'SubClass Of', 'SubClass Of (Anonymous Ancestor)', 'Members', and 'Target for Key'.

On the right, the 'Pizza' class editor is open, showing the 'Class expression editor' and 'Object restriction creator' tabs. The 'Data restriction creator' tab is active, displaying the 'Restricted property' section with 'topDataProperty' and 'TieneCalorias'. The 'Restriction type' is set to 'Some (existential)' and the 'Cardinality' is 1. The 'Restriction filler' list on the right includes 'int', 'integer', 'language', 'Literal', 'long', 'Name', 'NCName', 'negativeInteger', and 'NMTOKEN'.

At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

Propiedad Datatype en Protégé

The screenshot displays the Protégé ontology editor interface. On the left, the 'Class hierarchy (inferred)' panel shows a tree structure starting with 'Thing', followed by 'Pizza', 'Interesante', 'Cantimpalo', 'NombrePizza', 'PizzaBase', 'PizzaIngrediente', and a list of ingredients: 'CarneIngrediente', 'PescadoIngrediente', 'QuesoIngrediente', 'Muzzarela', 'Parmesano', 'Prueba', and 'VegetalesIngrediente'. The 'Active Ontology' panel shows 'Individuals: NewCantimpalo' with a single instance 'NewCantimpalo'.

A dialog box titled 'NewCantimpalo' is open, showing the 'Data Property' tab. The 'Value' field contains '200' and the 'Type' is set to 'integer'. The 'Aceptar' button is highlighted.

Below the dialog, the 'Description: NewCantimpalo' panel shows the 'Types' section with 'Cantimpalo' selected. The 'Property assertions: NewCantimpalo' panel shows sections for 'Object property assertions', 'Data property assertions', 'Negative object property assertions', and 'Negative data property assertions', each with a '+' button to add assertions.