

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

1. Instalação e versão (a utilizar) do Protégé (para “Desktop”)

Instalação do Protégé versão 5.2.0 no Windows 10.

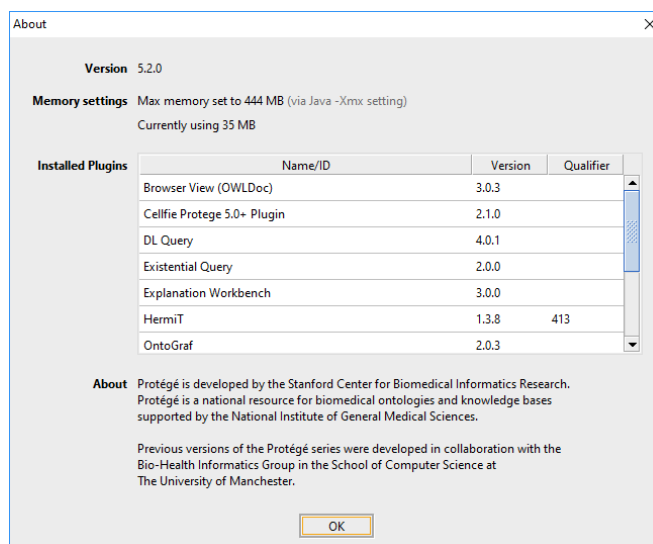


Figura 1- Versão do Protégé instalada.

2. Verificar Protégé e instalar o GraphViz

- Lançar o Protégé através da execução do comando “run.bat” numa janela de consola, por forma a permitir a visualização de eventuais erros.
- Adicionar a janela separadora OWLViz na interface gráfica do Protégé, através do menu “Windows\Tabs\OWLViz”. Como esta operação não gerou qualquer erro passámos para o próximo exercício, garantindo que a instalação do Protégé foi bem-sucedida.

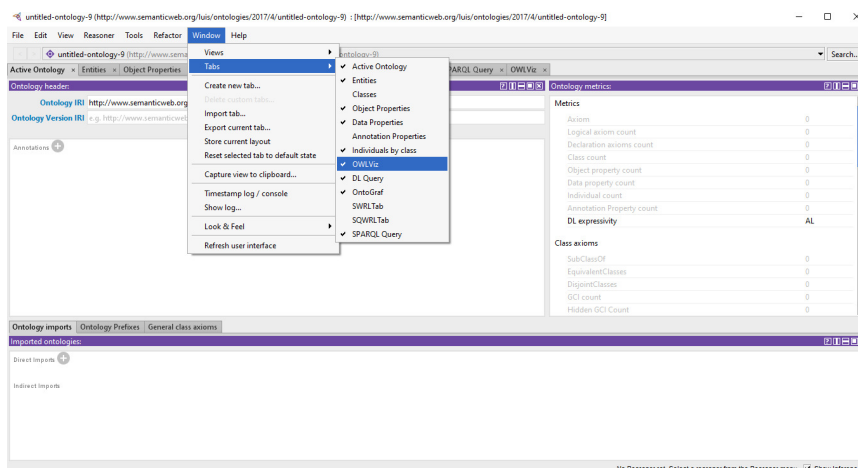


Figura 2- Activação do OWLViz.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

3. Adicionar separadores (“tab”) e “fazer-experiência com SPARQL”

a) Adicionar as seguintes janelas separadores na interface gráfica do Protégé, através do menu “Windows\Tabs\”:

\“Object Properties”
 \“Data Properties”
 \“OntoGraf”
 \“SPARQL Query”.

b) Na janela separadora “Entities”, adicionar a classe “A” e a classe “B” com uma sub-classe de “A”.

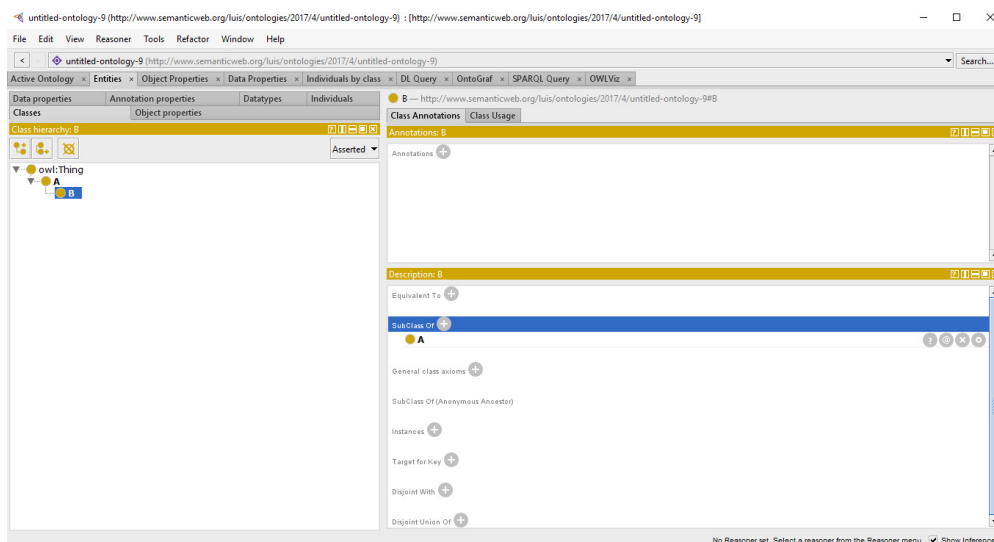


Figura 3- Classe “A” e sub-classe “B”.

c) Seleccionar a janela separadora “SPARQL Query” para executar a interrogação por omissão. Como a janela estava vazia, em consequência do erro que surgiu na janela de consola, foi necessário instalar os “plugins” “OWLAPI RDF Library” e “Protege SPARQL Plugin”, através da selecção do menu “File \ Check for plugins...”.

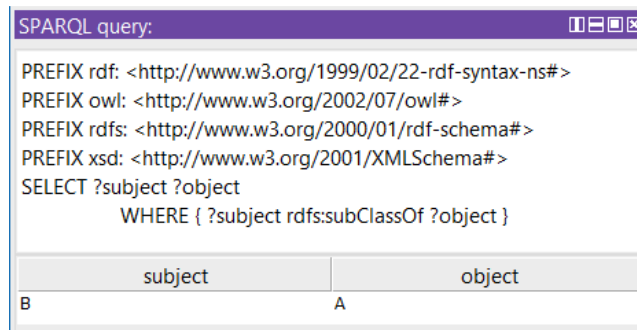
O resultado da interrogação por omissão, mostra todos os pares sujeito-objecto, no qual o sujeito seja sub-classe (*rdfs:subClassOf*) do objecto.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }
```

Figura 4- Interrogação efectuada.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela



SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }
```

subject	object
B	A

Figura 5- Resultado da interrogação.

- d) Gravar a ontologia no formato “RDF/XML Syntax” através do menu “File\Save as...” no ficheiro “myFirstTBox.owl”.

O ficheiro gerado contém o grafo RDF em XML, com os recursos “A” e “B” do tipo classe (owl:Class), com a relação de “B” como sendo uma sub-classe de “A”. Adicionalmente tem o recurso “http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2” como sendo do tipo (“rdf:type”) owl:Ontology.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2#"
  xml:base="http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2"/>

  <!--
  ////////////////////////////////////////////////////
  //
  // Classes
  //
  ////////////////////////////////////////////////////
  -->

  <!-- http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2#A -->
  <owl:Class rdf:about="http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2#A"/>

  <!-- http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2#B -->
  <owl:Class rdf:about="http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2#B">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/nuno/ontologies/2017/4/untitled-ontology-2#A"/>
  </owl:Class>
</rdf:RDF>
```

Figura 6- Ficheiro “myFirstTBox.owl”.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

- e) Executar uma directiva SPARQL no Protégé para obter os sujeitos do tipo “owl:Class”. No resultado aparecem 2 “A”s e 2 “B”s porque A e B são classes e o facto de B ser sub-classe de A também faz com que ambos sejam classes, como se pode ver na representação N-Triples (Figura 9) e no gráfico (Figura 10).

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject
      WHERE { ?subject rdf:type owl:Class }

```

Figura 7- Directiva SPARQL para obter os sujeitos do tipo owl: Class.

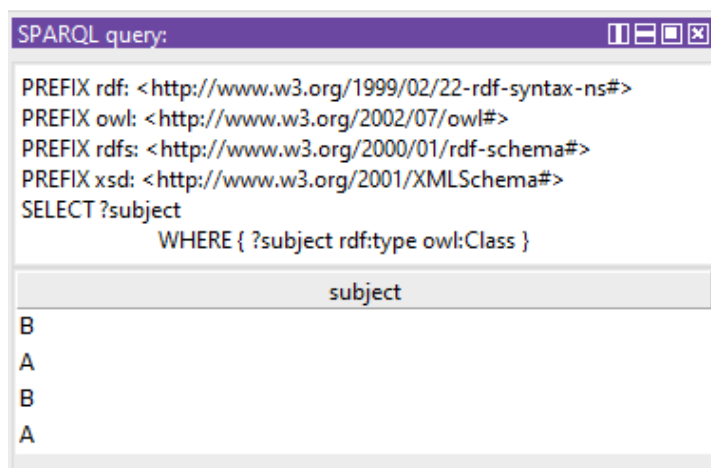


Figura 8- Resultado da execução da directiva.

Number	Subject	Predicate	Object
1	http://.../untitled-ontology-2	http://.../22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
2	http://.../untitled-ontology-2#A	http://.../22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
3	http://.../untitled-ontology-2#B	http://.../22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
4	http://.../untitled-ontology-2#B	http://.../rdf-schema#subClassOf	http://.../untitled-ontology-2#A

Figura 9- Representação em N-triples.

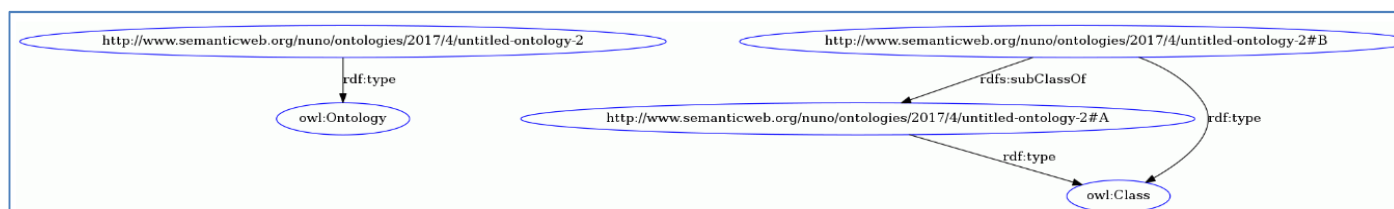


Figura 10- Representação gráfica.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

- f) Retirar a classe “B” como sendo uma sub-classe de “A”, passando a ser “irmã” de “A” e executar a interrogação anterior. Verificamos agora que o resultado da interrogação apresenta o recurso “A” e “B” uma única vez. Esta diferença de resultados relativamente à alínea anterior, deve-se ao facto de, agora, o recurso “B” não ser uma sub-classe de “A”.

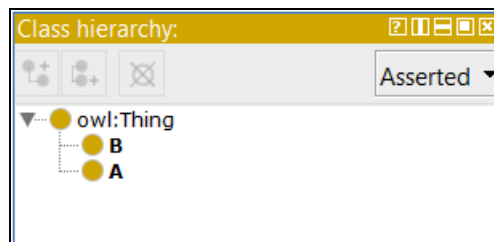


Figura 11- Classes “A” e “B” como irmãs.

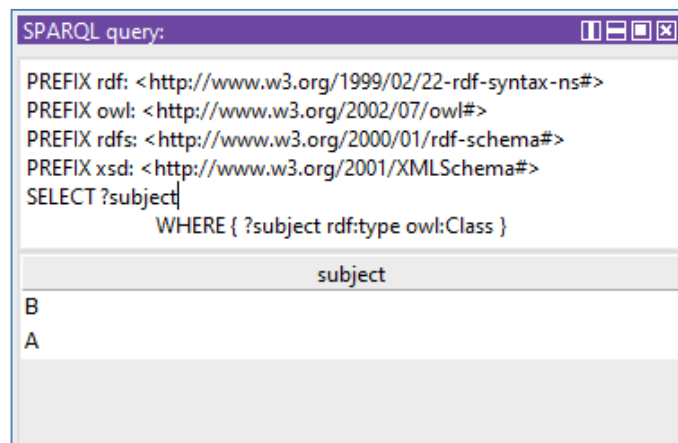


Figura 12- Resultado da execução da directiva

- g) Guardar a ontologia no ficheiro “Protege_ontology_alinea_3g.owl” e criar uma nova para o próximo exercício.

```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/luis/ontologies/2017/4/untitled-ontology-9#"
  xml:base="http://www.semanticweb.org/luis/ontologies/2017/4/untitled-ontology-9"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <owl:Ontology rdf:about="http://www.semanticweb.org/luis/ontologies/2017/4/untitled-ontology-9"/>

  <!--
  //////////////////////////////////////
  //
  
```

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

```
// Classes
//
//
//
-->

<!-- http://www.semanticweb.org/luis/ontologies/2017/4/untitled-ontology-9#A -->

<owl:Class rdf:about="http://www.semanticweb.org/luis/ontologies/2017/4/untitled-ontology-9#A"/>

<!-- http://www.semanticweb.org/luis/ontologies/2017/4/untitled-ontology-9#B -->

<owl:Class rdf:about="http://www.semanticweb.org/luis/ontologies/2017/4/untitled-ontology-9#B"/>
</rdf:RDF>

<!-- Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi -->
```

Figura 13- Ficheiro Protege_ontology_alinea_3g.owl.

4. Criar uma nova Ontologia e atribuir Comentário (usando guião)

O objetivo desta alínea é criar uma nova Ontologia e atribuir-lhe um comentário.

- Iniciar o Protégé através da execução do comando “run.bat”.
- Na janela separadora “Active Ontology”, atribuir em “Ontology IRI” o URL <http://www.pizza.com/ontologies/pizza.owl>.
- Gravar a Ontologia no formato “RDF/XML Syntax”, no ficheiro “pizza.owl”.

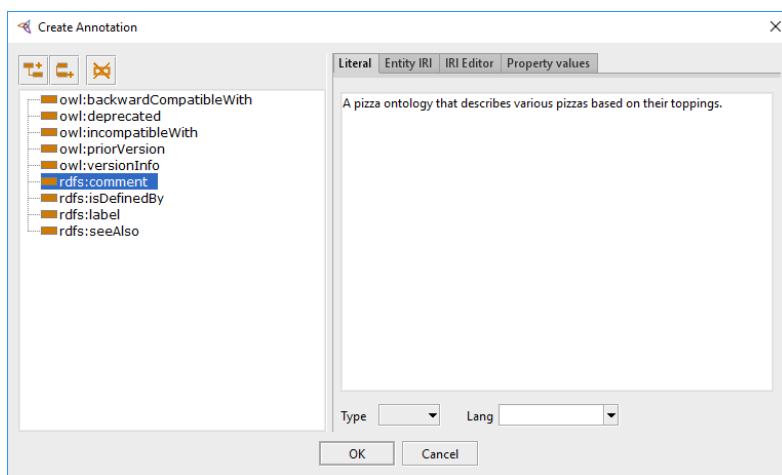
```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.pizza.com/ontologies/pizza.owl#"
  xml:base="http://www.pizza.com/ontologies/pizza.owl"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://www.pizza.com/ontologies/pizza.owl"/>
</rdf:RDF>

<!-- Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi -->
```

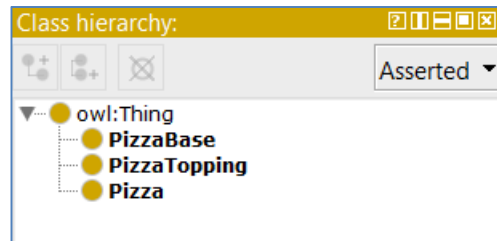
Figura 14- Ficheiro pizza.owl.

Representação e Processamento de Conhecimento (RPC)**Grupo 1 :** N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

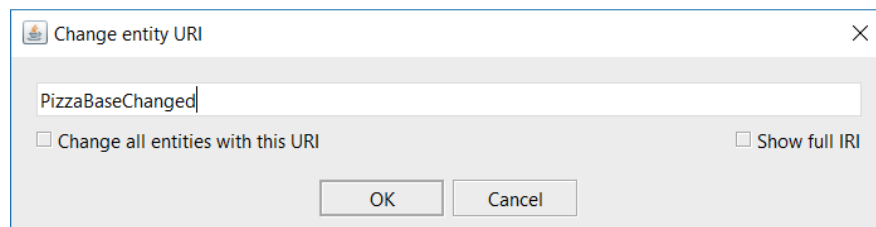
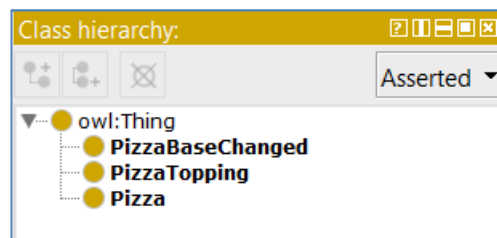
- d) Na janela do separador “Active Ontology”, adicionar “Annotations” atribuindo ao “rdfs:comment” o valor “A pizza ontology that describes various pizzas based on their toppings”.

*Figura 15- Atribuição do comentário.**Figura 16- Atribuição do comentário***5. Construir Conceitos e Axiomas de Inclusão (usando guião)**

- a) Criação de três classes: “Pizza”, “PizzaTopping” e “PizzaBase” como sub-classes de “owl:Thing”, de acordo com o tutorial *01_ProtegeOWLTutorialP4_v1_3.pdf*.

Representação e Processamento de Conhecimento (RPC)**Grupo 1 :** N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela*Figura 17- Criação das classes.*

b) Alterar o nome da classe recorrendo ao “Refactor \ Rename Entity”.

*Figura 18- “Refactor Entity” PizzaBase para PizzaBaseChanged.**Figura 19- Resultado do “Refactor Entity”.*

No final, voltámos a repôr o nome da classe “PizzaBase”.

c) Aplicar o conceito de “classes disjuntas” às classes criadas, por forma a que um objecto não possa ser uma instância de mais do que uma das três classes.

Seleccionámos a classe “Pizza”, na janela de Classes, e utilizámos o botão “Disjoint With” na janela “Description”. Indicámos, então, que pretendíamos tornar as classes “PizzaTopping” e “PizzaBase” disjuntas da classe “Pizza”.

Verificámos, depois, que as outras classes também tinham reflectido a operação de disjunção nas suas propriedades.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

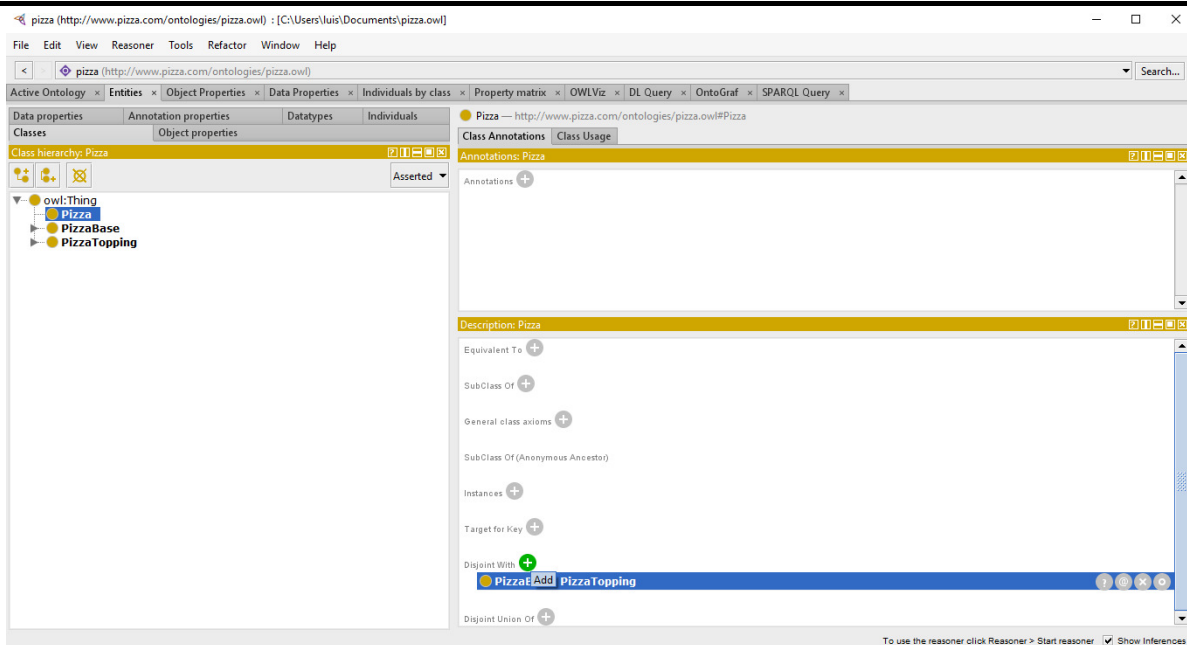


Figura 20- Indicação de classes disjuntas.

Nota: Também se pode fazer esta operação através do menu “Edit\Make primitive siblings disjoint”. O resultado final pode igualmente ser verificado no separador “Class Usage”, seleccionando todas as classes (Figura 21).

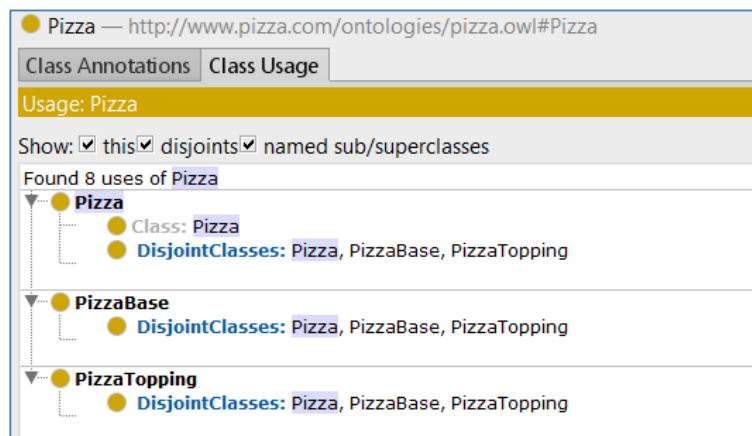


Figura 21- Separador “Class Usage” .

d) Gravar a ontologia, com o formato “RDF/XML syntax”, no ficheiro “pizza.owl”.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.pizza.com/ontologies/pizza.owl#"
  xml:base="http://www.pizza.com/ontologies/pizza.owl"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://www.pizza.com/ontologies/pizza.owl">
    <rdfs:comment>A pizza ontology that describes various pizzas based on their toppings. </rdfs:comment>
  </owl:Ontology>

  <!--
  ////////////////////////////////////////////////////
  //
  // Classes
  //
  ////////////////////////////////////////////////////
  -->

  <!-- http://www.pizza.com/ontologies/pizza.owl#Pizza -->
  <owl:Class rdf:about="http://www.pizza.com/ontologies/pizza.owl#Pizza"/>

  <!-- http://www.pizza.com/ontologies/pizza.owl#PizzaBase -->
  <owl:Class rdf:about="http://www.pizza.com/ontologies/pizza.owl#PizzaBase"/>

  <!-- http://www.pizza.com/ontologies/pizza.owl#PizzaTopping -->
  <owl:Class rdf:about="http://www.pizza.com/ontologies/pizza.owl#PizzaTopping"/>

  <!--
  ////////////////////////////////////////////////////
  //
  // General axioms
  //
  ////////////////////////////////////////////////////
  -->

  <rdf:Description>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
    <owl:members rdf:parseType="Collection">
      <rdf:Description rdf:about="http://www.pizza.com/ontologies/pizza.owl#Pizza"/>
      <rdf:Description rdf:about="http://www.pizza.com/ontologies/pizza.owl#PizzaBase"/>
      <rdf:Description rdf:about="http://www.pizza.com/ontologies/pizza.owl#PizzaTopping"/>
    </owl:members>
  </rdf:Description>
</rdf:RDF>

<!-- Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi -->

```

Figura 22- Ficheiro RDF/XML “pizza.owl” .

No ficheiro *RDF/XML* resultante, podemos verificar que existe um nó anónimo do tipo “owl: AllDisjointClasses” que agrega, através de *rdf: parseType=“Collection”*, três nós anónimos do tipo “Pizza”, “PizzaBase” e “PizzaTopping” (Figura 23).

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

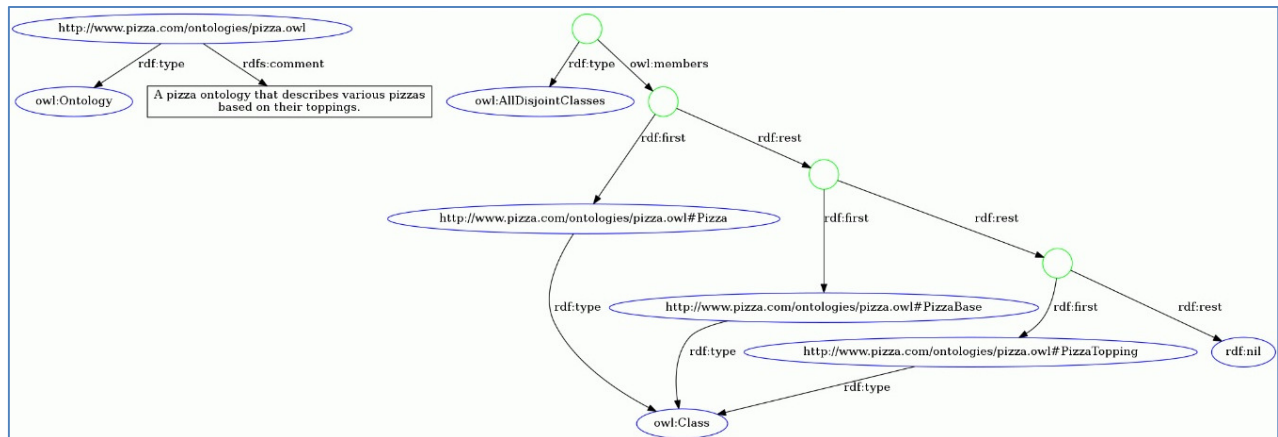


Figura 23- Grafo do ficheiro “pizza.owl”.

- e) Criação das classes “ThinAndCrispyBase” e “DeepPanBase”, sub-classes da “PizzaBase” e disjuntas.

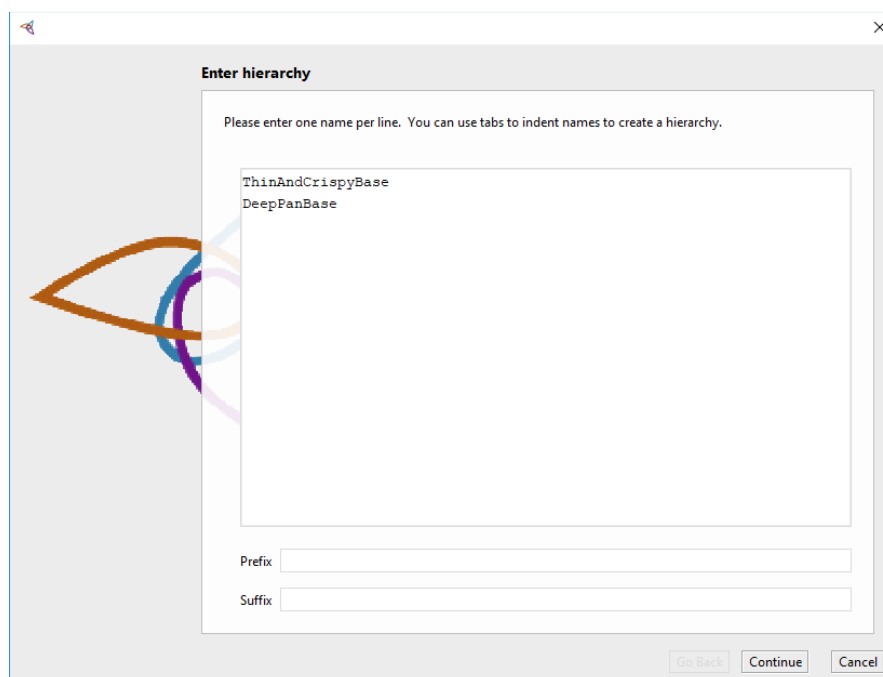
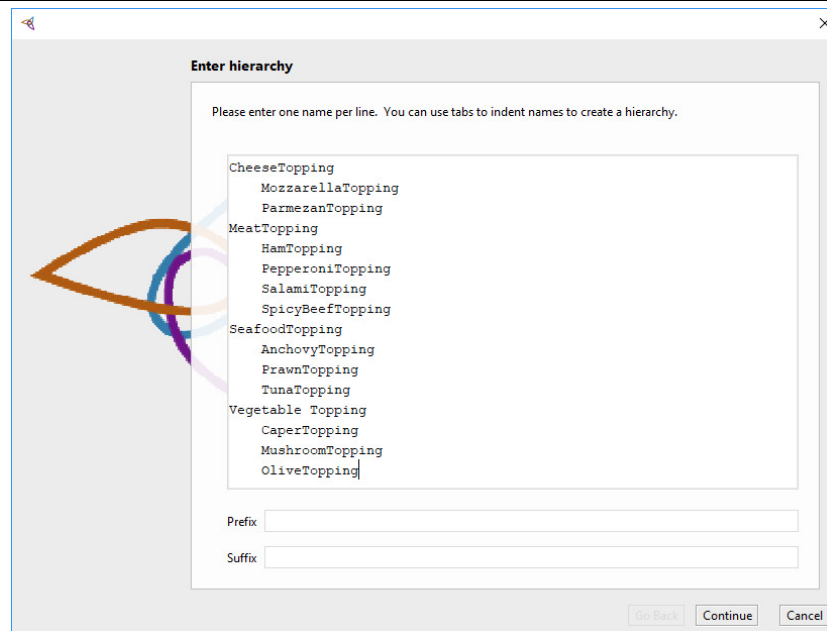
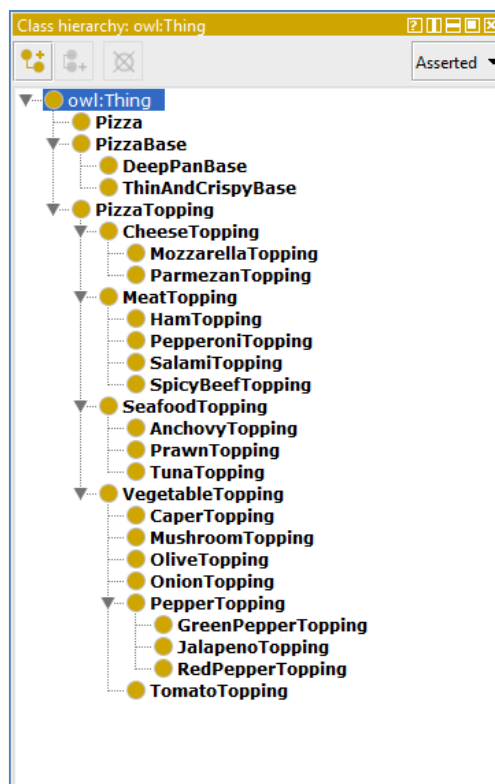


Figura 24- Criação das sub-classes, através da ferramenta
“Create Class Hierarchy...”.

- f) Criação da hierarquia de sub-classes da classe “PizzaTopping” de uma só vez, utilizando a ferramenta “Create Class Hierarchy...”.

Representação e Processamento de Conhecimento (RPC)**Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela***Figura 25- Ferramenta "Create Class Hierarchy..."**Figura 26- Hierarquia de classes criada.*

Representação e Processamento de Conhecimento (RPC)**Grupo 1 :** N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

g) Gravar a ontologia, com o formato “RDF/XML syntax”, no ficheiro “pizza.owl”.

6. Construir Axiomas de Igualdade (sem usar guião)

a) Iniciar uma nova ontologia com “\File\New...”.

b) Dar o nome “http://FAMILIA” à nova ontologia e gravá-la como “familia.owl”.

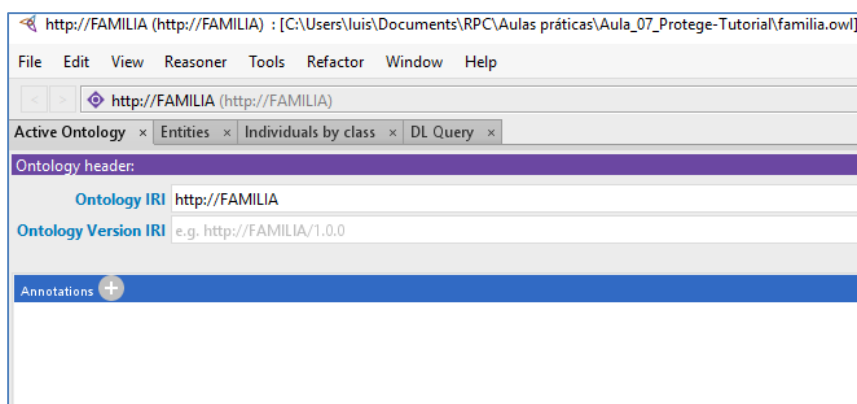


Figura 27- Ontologia “http://FAMILIA” criada e gravada.

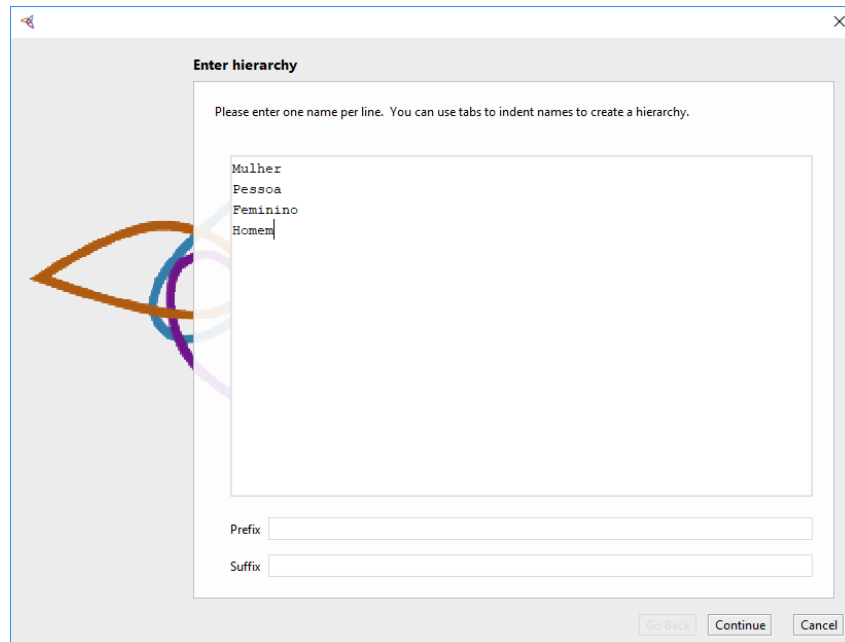
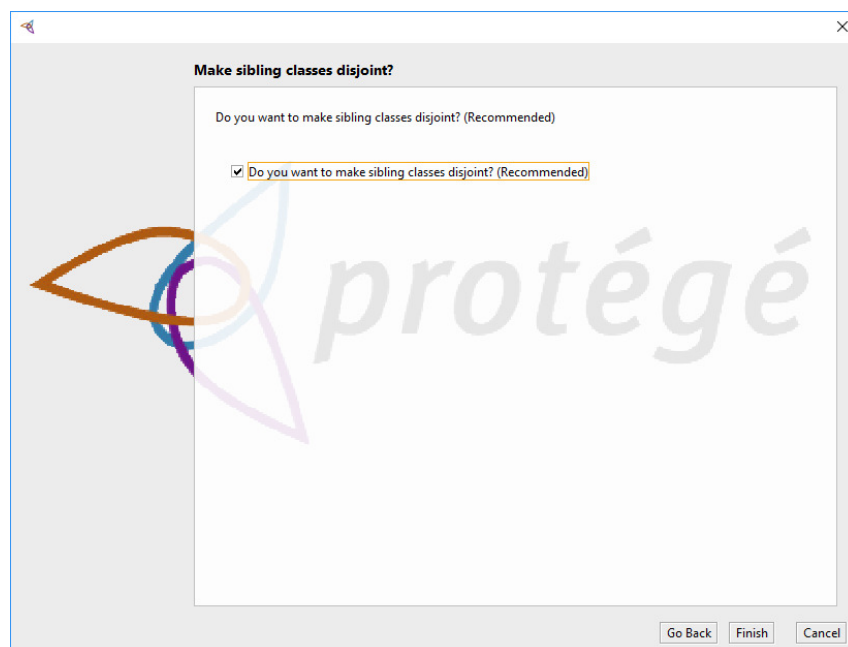
c) Considerar a TBox em “a04_logicaDeDescricao.pdf”, página 25 (relações de parentesco).

- i. $Mulher \equiv Pessoa \sqcap Feminino$

ii. $Homem \equiv Pessoa \sqcap \neg Mulher$

Figura 28- Axiomas de igualdade da TBox.

d) Construir os conceitos (classes): Pessoa, Feminino, Mulher, Homem, todas de uma vez através de “\Tools\Create class hierarchy...”. Notar a indicação sobre se são ou não disjuntas.

Representação e Processamento de Conhecimento (RPC)**Grupo 1 :** N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela*Figura 29- Construção das classes.**Figura 30- Indicação para que sejam criadas classes disjuntas.*

e) Construir os dois primeiros axiomas de igualdade da TBox.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

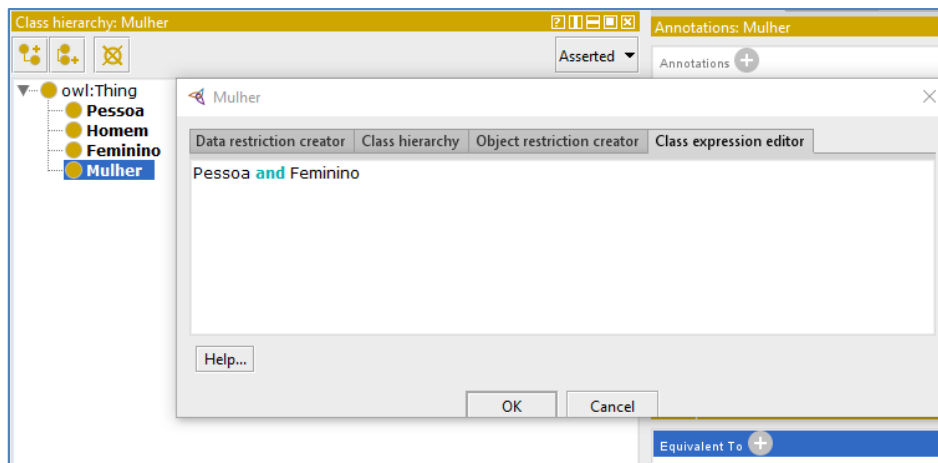


Figura 31- Construção do axioma de igualdade com o bloco “Equivalent To”.

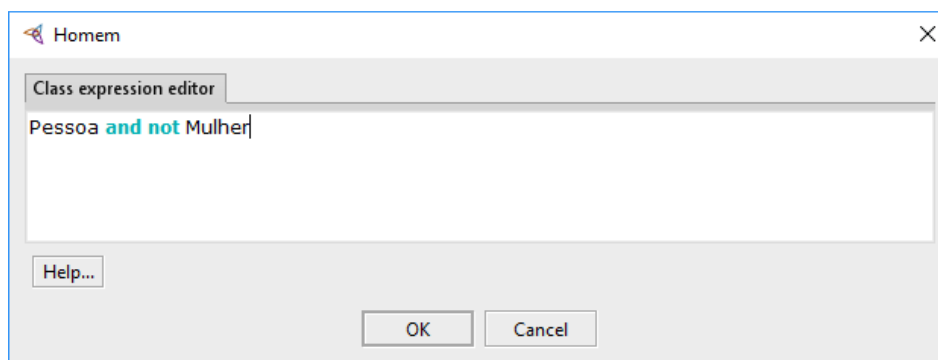


Figura 32- Construção do segundo axioma de igualdade.

f) Ver os seguintes sítios, com informação adicional sobre o “Class expression editor”:

<https://protegewiki.stanford.edu/wiki/Protege4ExpressionEditor>

https://protegewiki.stanford.edu/wiki/Protege4UserDocs#Editor_features/

g) Gravar e ler o ficheiro “família.owl”.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://FAMILIA#"
  xml:base="http://FAMILIA#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://FAMILIA"/>

  <!--
  //////////////////////////////////////
  //
```

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

```

// Classes
//
////////////////////////////////////
-->

<!-- http://FAMILIA#Feminino -->
<owl:Class rdf:about="http://FAMILIA#Feminino"/>

<!-- http://FAMILIA#Homem -->
<owl:Class rdf:about="http://FAMILIA#Homem">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://FAMILIA#Pessoa"/>
        <owl:Class>
          <owl:complementOf rdf:resource="http://FAMILIA#Mulher"/>
        </owl:Class>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<!-- http://FAMILIA#Mulher -->
<owl:Class rdf:about="http://FAMILIA#Mulher">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://FAMILIA#Feminino"/>
        <rdf:Description rdf:about="http://FAMILIA#Pessoa"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<!-- http://FAMILIA#Pessoa -->
<owl:Class rdf:about="http://FAMILIA#Pessoa"/>

<!--
////////////////////////////////////
//
// General axioms
//
////////////////////////////////////
-->

<rdf:Description>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://FAMILIA#Feminino"/>
    <rdf:Description rdf:about="http://FAMILIA#Homem"/>
    <rdf:Description rdf:about="http://FAMILIA#Mulher"/>
    <rdf:Description rdf:about="http://FAMILIA#Pessoa"/>
  </owl:members>
</rdf:Description>
</rdf:RDF>

```


Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

```
<!-- Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi
-->
```

Figura 33- Serialização do grafo RDF(S) em XML, do ficheiro “família.owl”.

7. Construir Relações, ou Propriedades, ou Papéis (usando guião)

Regressar à ontologia da piza. Abrir o “01_ProtegeOWLTutorialP4_v1_3.pdf” na secção 4.4 (pág. 23).

- a) Seguir o tutorial para construir as propriedades `hasTopping` e `hasBase`, como sub-propriedades de `hasIngredient`.

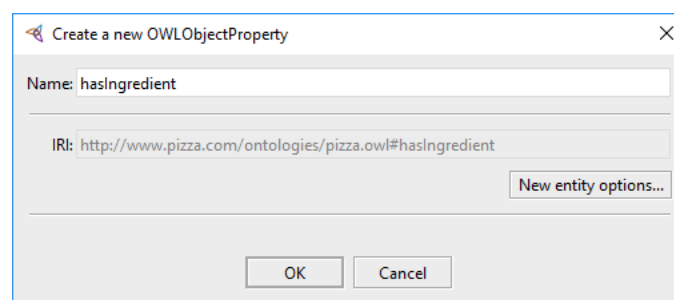


Figura 34- Criação de “OWL object property”.

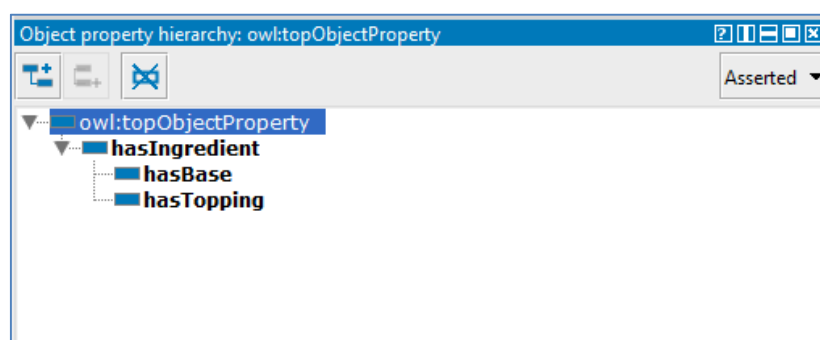


Figura 35- Propriedade `hasIngredient` com as sub-propriedades `hasBase` e `hasTopping`.

- b) Construir as propriedades e propriedades inversas indicadas na secção 4.5.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

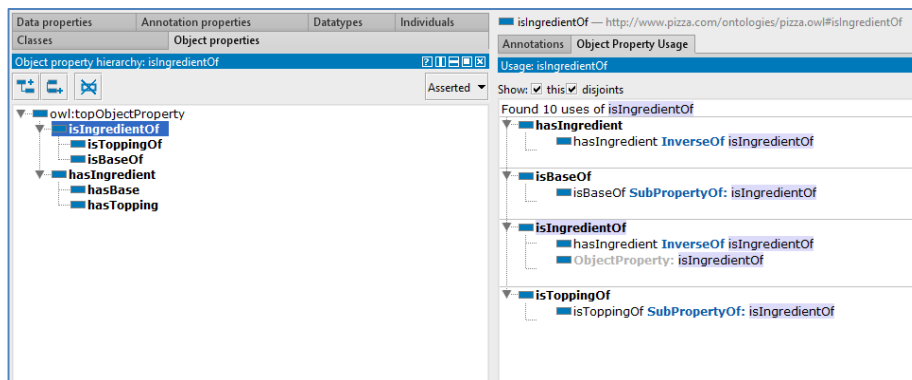


Figura 36- Propriedade *isIngredientOf* com sub-propriedades e propriedades inversas.

c) Instalar o “plug-in” “Matrix View” e activar em “\Windows\Tabs” a “Property Matrix”.

Object Property	Func	Sym	Inv Func	Trans	ASym	Refl	Irrefl	Domain	Range	Inverse
owl:topObjectProperty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
hasIngredient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			isIngredientOf
hasBase	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			isBaseOf
hasTopping	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			isToppingOf
isIngredientOf	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			hasIngredient
isBaseOf	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			hasBase
isToppingOf	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			hasTopping

Figura 37- Visualização e manipulação simplificada das propriedades, através da “Property Matrix”.

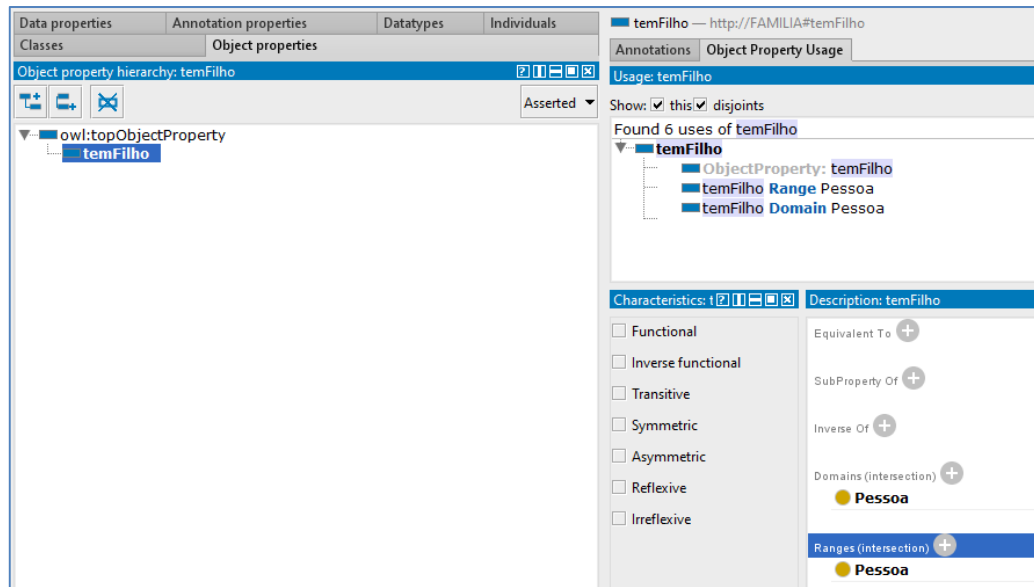
8. Construir e “Configurar” Relações (sem usar guião)

Regressar à ontologia “http://FAMILIA”.

- a) Construir a relação *temFilho* (em “Object Properties”) e indicar que *tem* como domínio e contradomínio *Pessoa*, nos blocos “Domain” e “Range”.

Representação e Processamento de Conhecimento (RPC)

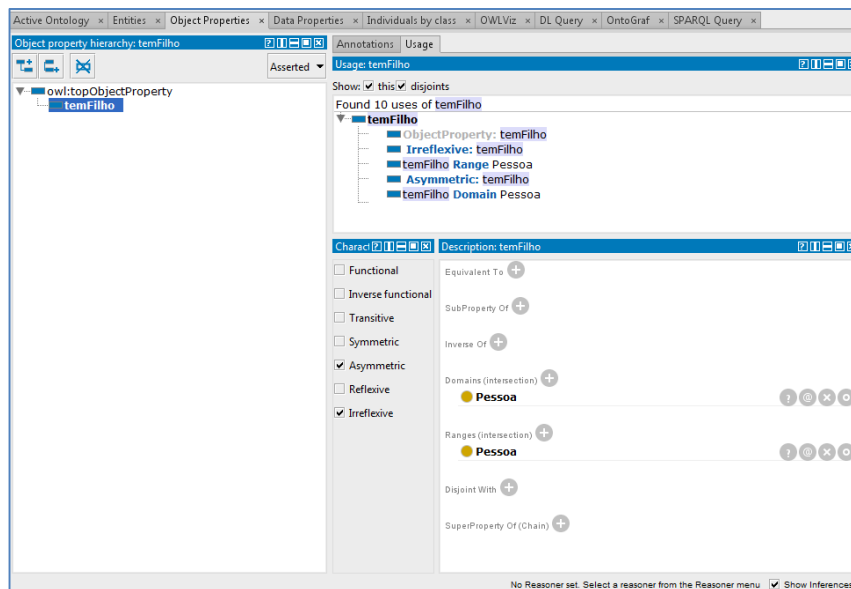
Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

Figura 38- Relação *temFilho* com domínio e contradomínio *Pessoa*.

- b) Ver com atenção a secção 4.6 do “01_ProtegeOWLTutorialP4_v1_3.pdf” e decidir se a relação “temFilho” deve ser:
- *Functional* – Não. Pode haver mais do que um objecto relacionado com o mesmo sujeito através da propriedade “temFilho”, i.e. um indivíduo pode ter mais do que um filho;
 - *Inverse Functional* – Não. A propriedade inversa de “temFilho”, “éFilhoDe” não é *functional*, portanto “temFilho” não é *inverse functional*;
 - *Transitive* – Não. Se ‘a’ temFilho ‘b’ e ‘b’ temFilho ‘c’, não se verifica ‘a’ temFilho ‘c’, i.e. ‘c’ será neto de ‘a’ e não filho;
 - *Symmetric* – Não. Se ‘a’ temFilho ‘b’, não se verifica ‘b’ temFilho ‘a’, ‘a’ será pai e não filho de ‘b’;
 - *Assymmetric* – Sim. Concluímos do indicado anteriormente que nunca se pode verificar ‘b’ temFilho ‘a’ se ‘a’ temFilho ‘b’, por isso a propriedade é assimétrica;
 - *Reflexive* – Não. Um indivíduo não pode relacionar-se consigo próprio através da propriedade *temFilho*, i.e. ninguém pode ser filho de si próprio;
 - *Irreflexive* – Sim. Do indicado imediatamente acima, conclui-se que se ‘a’ temFilho ‘b’, ‘a’ e ‘b’ nunca podem ser o mesmo indivíduo.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

Figura 39- Definição de características da propriedade *temFilho*.

c) Gravar e ler o ficheiro “familia.owl”.

```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://FAMILIA#"
  xml:base="http://FAMILIA"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <owl:Ontology rdf:about="http://FAMILIA"/>

  <!--
  ///////////////////////////////////////////////////
  //
  // Object Properties
  //
  ///////////////////////////////////////////////////
  -->

  <!-- http://FAMILIA#temFilho -->
  <owl:ObjectProperty rdf:about="http://FAMILIA#temFilho">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AsymmetricProperty"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#IrreflexiveProperty"/>
    <rdfs:domain rdf:resource="http://FAMILIA#Pessoa"/>
    <rdfs:range rdf:resource="http://FAMILIA#Pessoa"/>
  </owl:ObjectProperty>

  <!--
  ///////////////////////////////////////////////////

```

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

```

//
// Classes
//
////////////////////////////////////
-->

<!-- http://FAMILIA#Feminino -->
<owl:Class rdf:about="http://FAMILIA#Feminino"/>

<!-- http://FAMILIA#Homem -->
<owl:Class rdf:about="http://FAMILIA#Homem">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://FAMILIA#Pessoa"/>
        <owl:Class>
          <owl:complementOf rdf:resource="http://FAMILIA#Mulher"/>
        </owl:Class>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<!-- http://FAMILIA#Mulher -->
<owl:Class rdf:about="http://FAMILIA#Mulher">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://FAMILIA#Feminino"/>
        <rdf:Description rdf:about="http://FAMILIA#Pessoa"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<!-- http://FAMILIA#Pessoa -->
<owl:Class rdf:about="http://FAMILIA#Pessoa"/>

<!--
////////////////////////////////////
//
// General axioms
//
////////////////////////////////////
-->

<rdf:Description>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://FAMILIA#Feminino"/>
    <rdf:Description rdf:about="http://FAMILIA#Homem"/>
    <rdf:Description rdf:about="http://FAMILIA#Mulher"/>
    <rdf:Description rdf:about="http://FAMILIA#Pessoa"/>
  </owl:members>
</rdf:Description>
</rdf:RDF>

```

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

```
<!-- Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi
-->
```

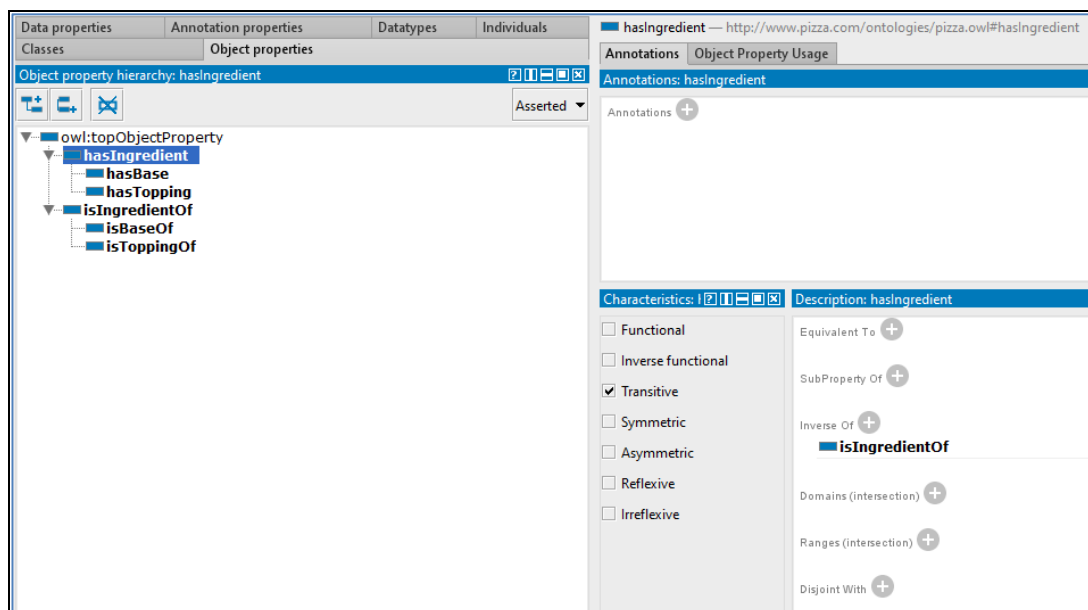
Figura 40- Ficheiro “família.owl”.

9. Construir e “Configurar” Relações (usando guião)

Regressar à ontologia da piza.

a) Para sistematizar, fazer os exercícios das secções 4.6 e 4.7.

Tornar a propriedade `hasIngredient` transitiva.

Figura 41- Propriedade `hasIngredient` Transitive.

Se uma propriedade é transitiva, a sua propriedade inversa é também transitiva. Na nossa versão do Protégé, é necessário assinalarmos também como transitiva a propriedade inversa, pois isso não acontece automaticamente.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

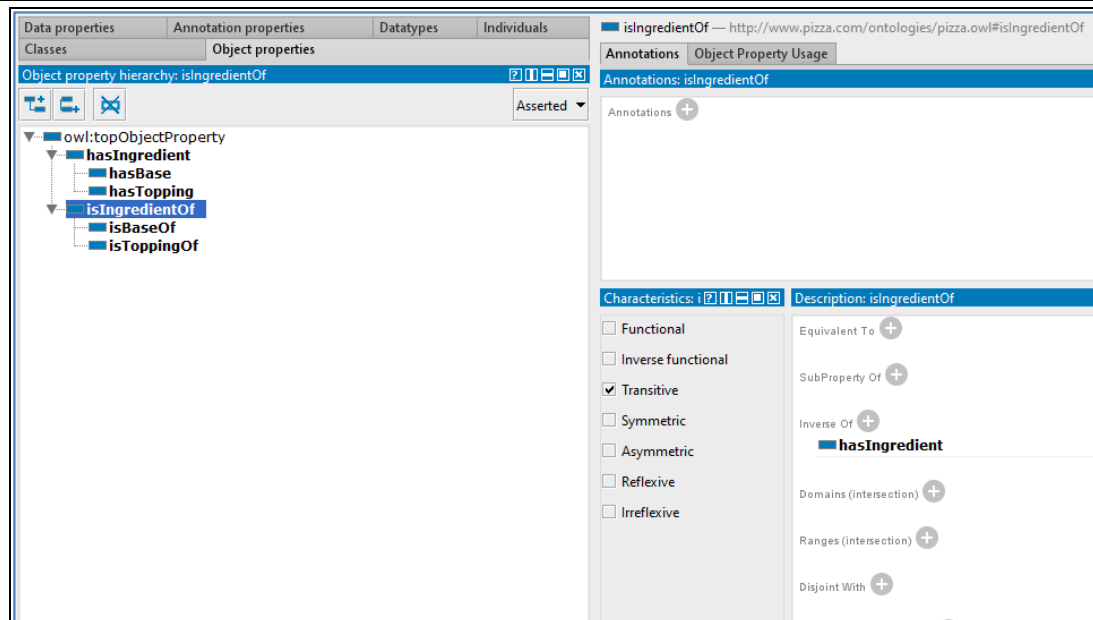


Figura 42- Propriedade isIngredientOf Transitive.

Tornar a propriedade hasBase 'functional'.

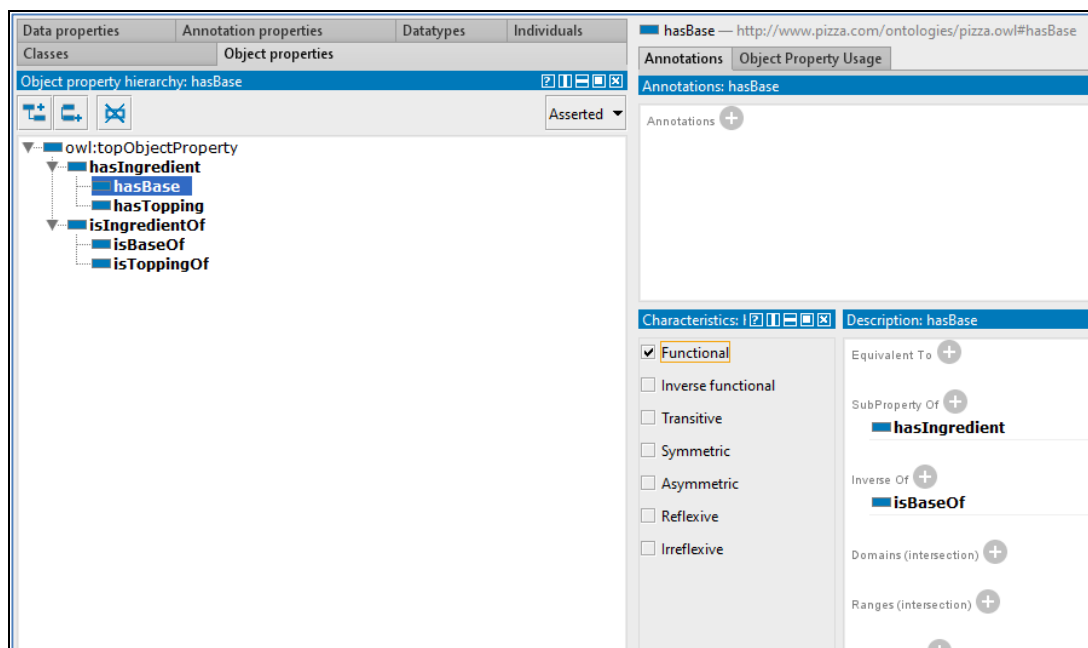
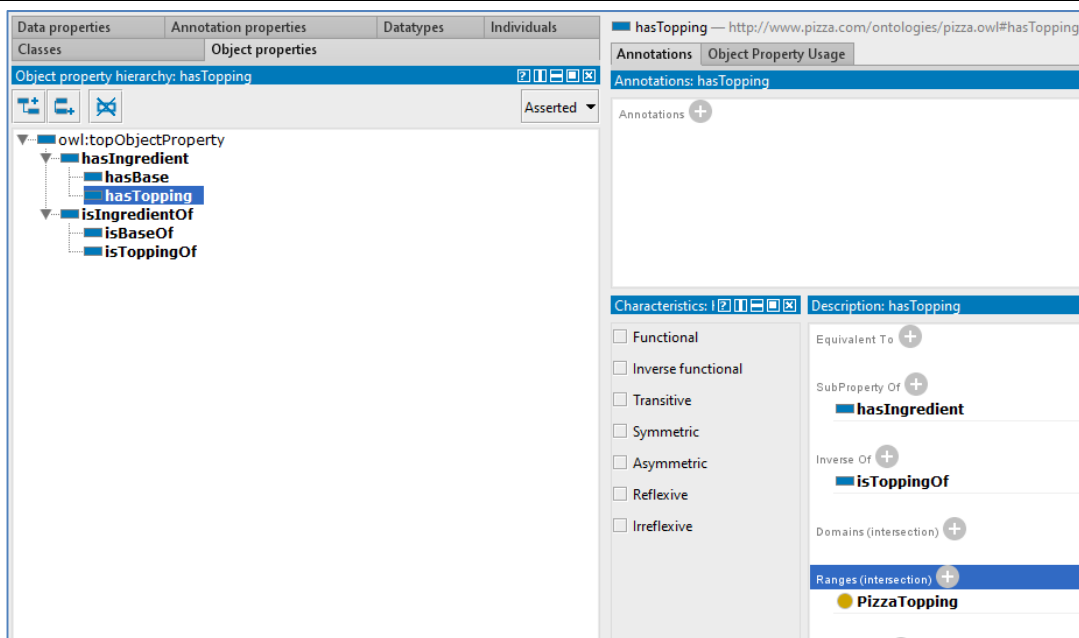
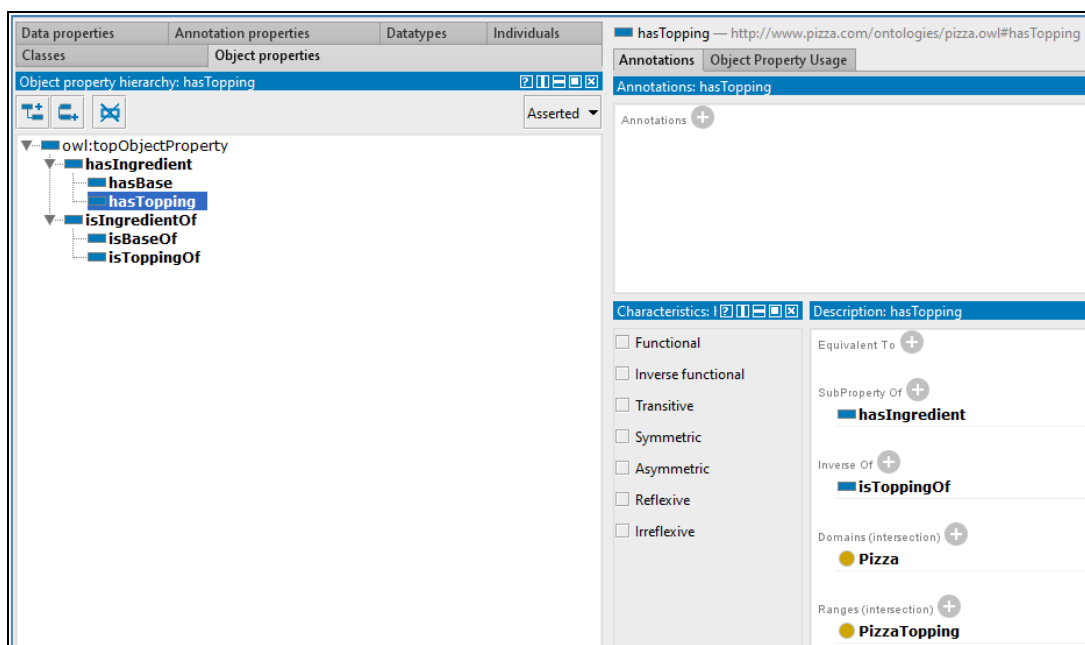


Figura 43- Propriedade hasBase Functional.

Definir o contradomínio de hasTopping.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

Figura 44- Contradomínio de *hasTopping*: *PizzaTopping*.Definir o domínio de *hasTopping*.Figura 45- Domínio de *hasTopping*: *Pizza*.

O domínio de uma propriedade é o contradomínio da sua propriedade inversa e o contradomínio de uma propriedade é o domínio da sua propriedade inversa. Assim, sendo

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

`isToppingOf` a propriedade inversa de `hasTopping`, tem como domínio `PizzaTopping` e como contradomínio `Pizza`.

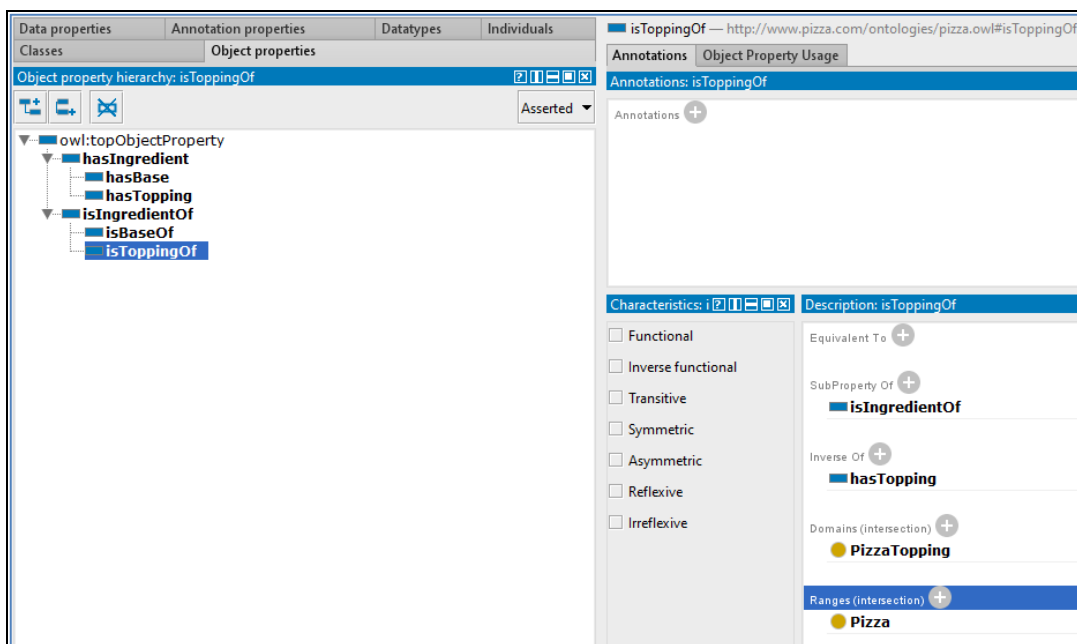


Figura 46- Domínio e contradomínio de `isToppingOf`.

Definir o domínio e contradomínio da propriedade `hasBase` e da sua propriedade inversa `isBaseOf`.

The screenshot shows the 'Object property matrix' for `isBaseOf`. The matrix has columns for 'Func', 'Sym', 'Inv Func', 'Trans', 'ASym', 'Refl', 'Irrefl', 'Domain', 'Range', and 'Inverse'. The rows show the domain and range for the property and its inverse.

	Func	Sym	Inv Func	Trans	ASym	Refl	Irrefl	Domain	Range	Inverse
<code>owl:topObjectProperty</code>										
<code>hasIngredient</code>										<code>isIngredientOf</code>
<code>hasBase</code>								<code>Pizza</code>	<code>PizzaBase</code>	<code>isBaseOf</code>
<code>hasTopping</code>								<code>Pizza</code>	<code>PizzaTopping</code>	<code>isToppingOf</code>
<code>isIngredientOf</code>										<code>hasIngredient</code>
<code>isBaseOf</code>								<code>PizzaBase</code>	<code>Pizza</code>	<code>hasBase</code>
<code>isToppingOf</code>								<code>PizzaTopping</code>	<code>Pizza</code>	<code>hasTopping</code>

Figura 47- Domínio e contradomínio de `hasBase` e de `isBaseOf`.

b) Gravar o projecto.

10. Usar o Motor de Inferência Pré-Configurado (FaCT++ e Hermit)

a) Seguir o tutorial na secção 4.9.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

Para demonstração da utilização de um motor de inferência, detectar inconsistências na ontologia através de uma classe de teste (*ProbeInconsistentTopping*) que é sub-classe de *CheeseTopping* e de *VegetableTopping*, simultaneamente.

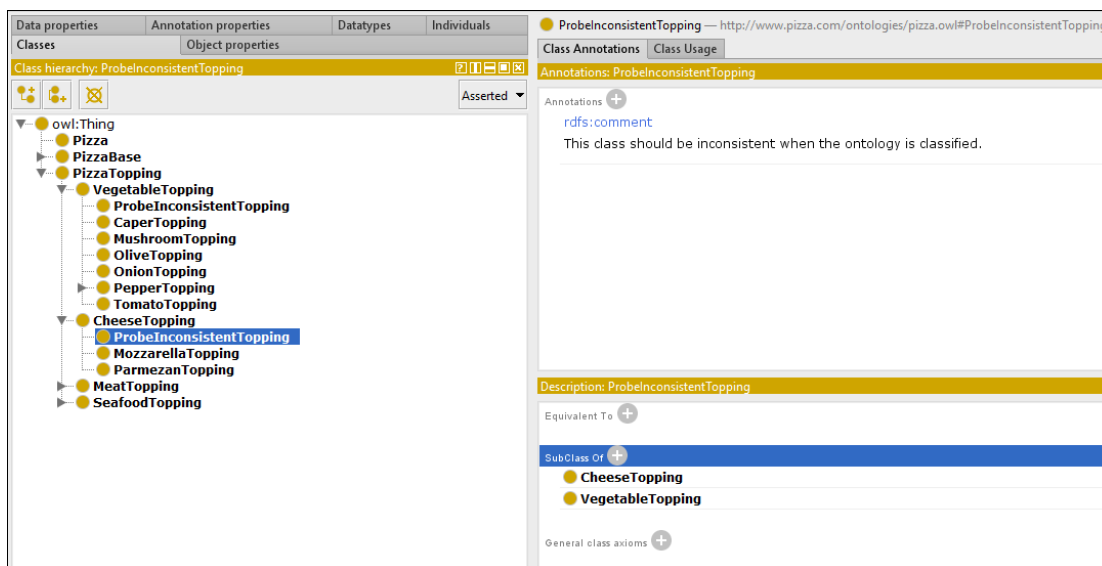


Figura 48- *ProbeInconsistentTopping* sub-classe de *CheeseTopping* e de *VegetableTopping*.

b) Seguir a secção 4.9 de modo a gerar a inconsistência aí descrita.

Executada a opção do menu “\Reasoner\Start reasoner”, a classe “*ProbeInconsistentTopping*” aparece a vermelho, indicando que existe uma inconsistência na ontologia.

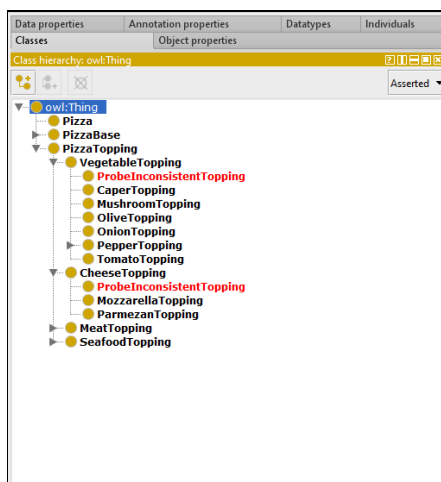


Figura 49- Inconsistência assinalada.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

- c) Qual a hierarquia de subsunção que se infere da TBox construída? Para obter a resposta, seleccionar “Class hierarchy (inferred)” e observar a taxonomia aí apresentada.

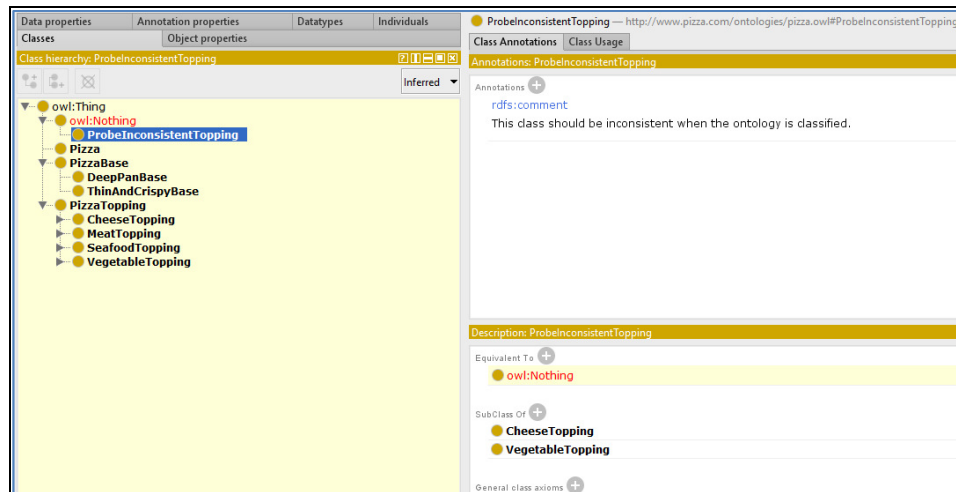


Figura 50- Hierarquia de subsunção que se infere da TBox construída.

Observamos que a classe *ProbeInconsistentTopping* aparece como sub-classe da classe *owl:Nothing* que representa um conjunto vazio e desaparece como sub-classe de *CheeseTopping* e *VegetableTopping*. Isto acontece porque, com as propriedades definidas na ontologia, não é consistente que a classe *ProbeInconsistentTopping* seja simultaneamente sub-classe de *CheeseTopping* e *VegetableTopping* (já que estas são classes disjuntas).

- d) Eliminar a inconsistência anterior editando “Disjoint Union Of”.

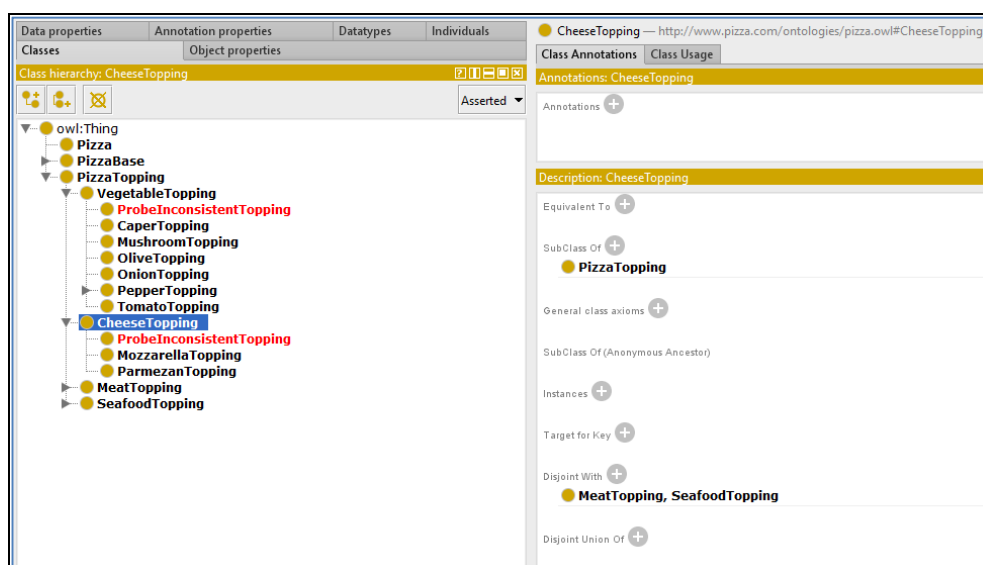


Figura 51- Disjunção entre as classes *CheeseTopping* e *VegetableTopping* removida.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

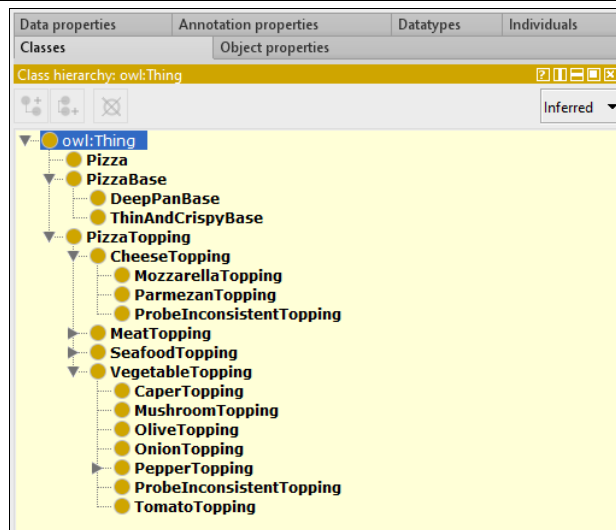


Figura 52- Desapareceu a inconsistência em *ProbeInconsistentTopping*.

A inconsistência assinalada a vermelho na classe *ProbeInconsistentTopping* desapareceu com a alteração efectuada porque, se as classes *CheeseTopping* e *VegetableTopping* já não são disjuntas, podem existir elementos comuns às duas classes. Nesse caso, a forma como foi definida a sub-classe *ProbeInconsistentTopping*, já é consistente com a ontologia definida.

11. Adicionar um Novo Motor de Inferência (FaCT++ e Pellet)

- Seleccionar “File\Check for plugins...”.
- Escolher “Pellet Reasoner” e, depois, escolher “Install”.
- Reiniciar o Protégé.
- Refazer o exercício anterior com “Reasoner” Pellet.

Os resultados obtidos com os três motores de inferência – *Hermit*, *Fact++* e *Pellet* – são exactamente os mesmos.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

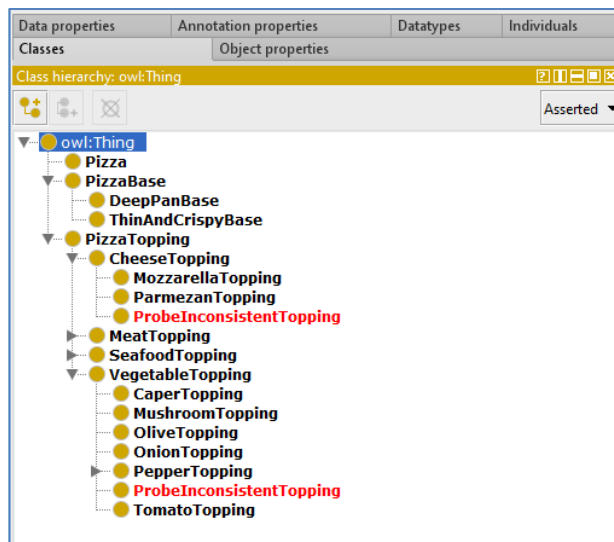


Figura 53- Execução do motor de inferência Pellet.

12. Especificar Restrições com Quantificadores (sem usar guião)

Regressar à ontologia “http://FAMILIA”.

a) Construir o axioma iii (define Mae) da TBox.

- i. $Mulher \equiv Pessoa \sqcap Feminino$
- ii. $Homem \equiv Pessoa \sqcap \neg Mulher$
- iii. $Mae \equiv Mulher \sqcap \exists temFilho.Pessoa$
- iv. $Pai \equiv Homem \sqcap \exists temFilho.Pessoa$
- v. $Progenitor \equiv Pai \sqcup Mae$
- vi. $Avo' \equiv Mae \sqcap \exists temFilho.Progenitor$

Figura 54- Axiomas da TBox.

Os quantificadores são escritos em notação infixa (na lógica de descrição usamos escrita prefixa). Mais informação sobre esta sintaxe em:

http://www.co-ode.org/resources/reference/manchester_syntax/

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

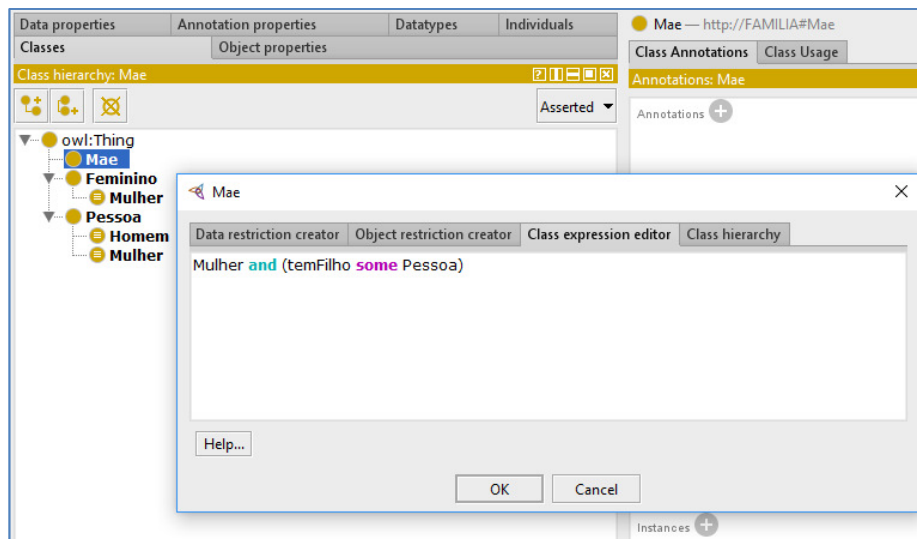


Figura 55- Construção do axioma que define Mae.

b) Construir o axioma iv (define Pai) da TBox.

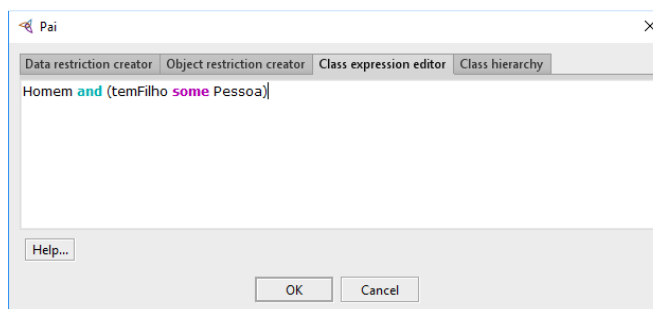


Figura 56- Construção do axioma que define Pai.

c) Construir o axioma v (define Progenitor) da TBox.

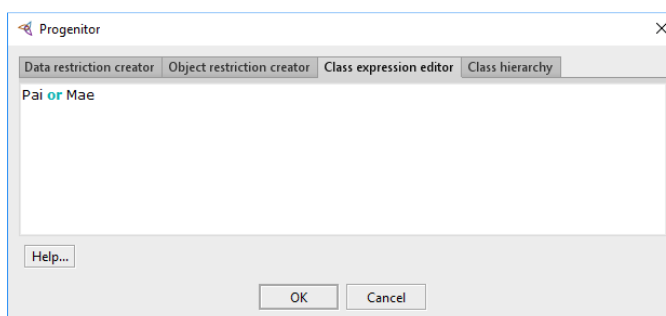
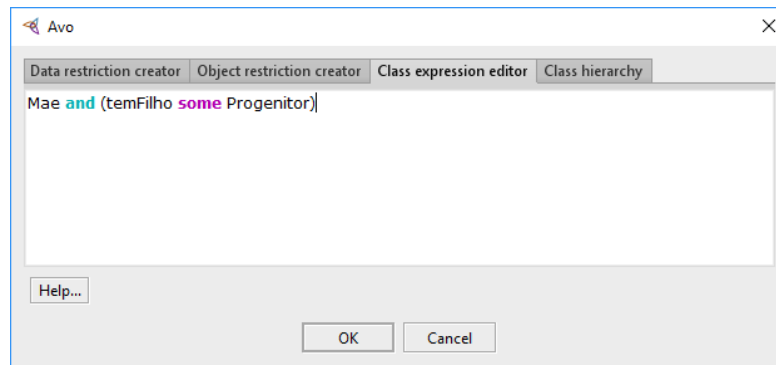


Figura 57- Construção do axioma que define Progenitor.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

d) Construir o axioma vi (define *Avo*) da TBox.Figura 58- Construção do axioma que define *Avo*.

e) Gravar e ler o ficheiro “família.owl”.

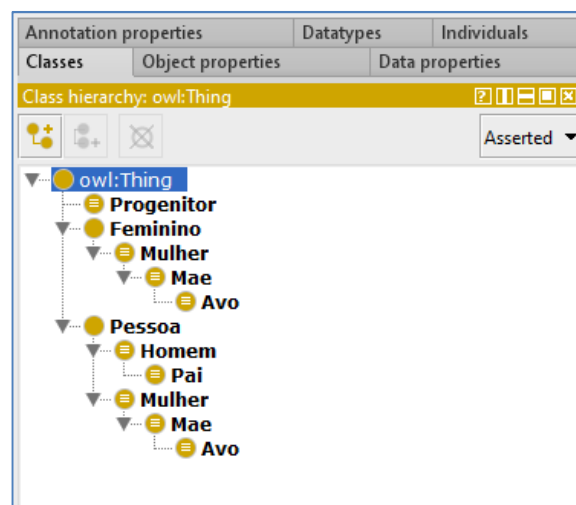


Figura 59- Hierarquia de classes da ontologia “família.owl”.

13. Especificar Restrições com Quantificadores (usando guião)

Regressar à ontologia da pizza.

a) Ler as ideias apresentadas na secção 4.8.2.

Representação e Processamento de Conhecimento (RPC)
--

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

b) Construir as quantificações existenciais indicadas na secção 4.8.2.

Acrescentar uma quantificação a `Pizza` que indique que uma `Pizza` tenha que ter uma `PizzaBase`.

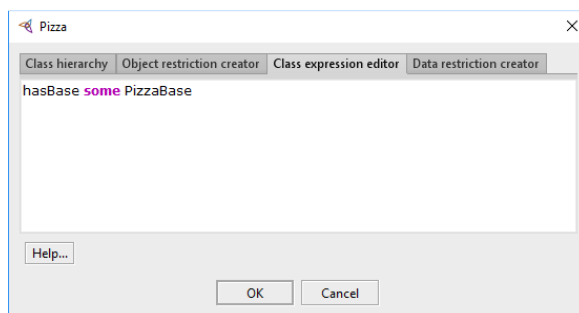


Figura 60- Quantificação acrescentada a `Pizza`.

Criar uma sub-classe de `Pizza` designada `NamedPizza` e uma sub-classe de `NamedPizza` designada `MargheritaPizza`.

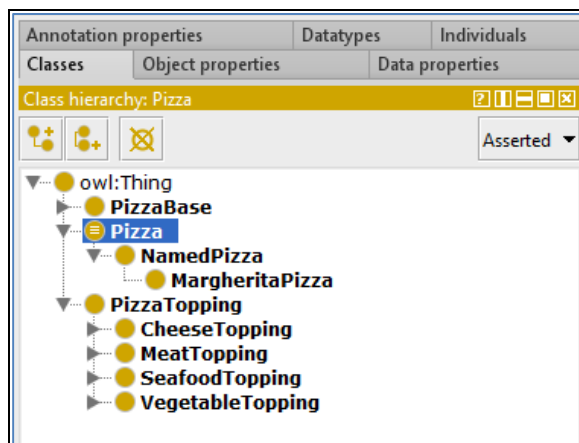


Figura 61- Criadas as sub-classes `NamedPizza` e `MargheritaPizza`.

Criar uma quantificação existencial (*some*) na `MargheritaPizza` que utilize a propriedade `hasTopping` para especificar que a `MargheritaPizza` tem pelo menos uma `MozzarellaTopping`.

Criar novamente uma quantificação existencial que especifique que a `MargheritaPizza` tem pelo menos uma `TomatoTopping`.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

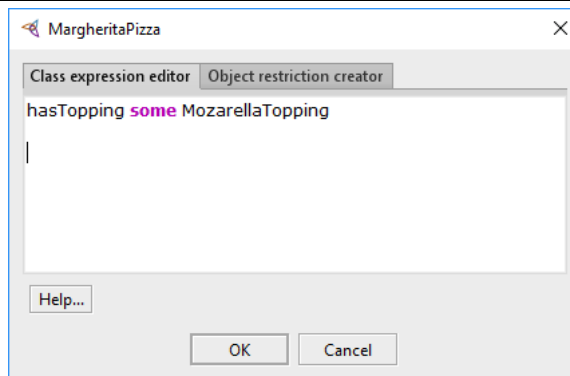


Figura 62- Quantificação especificada em MargheritaPizza.

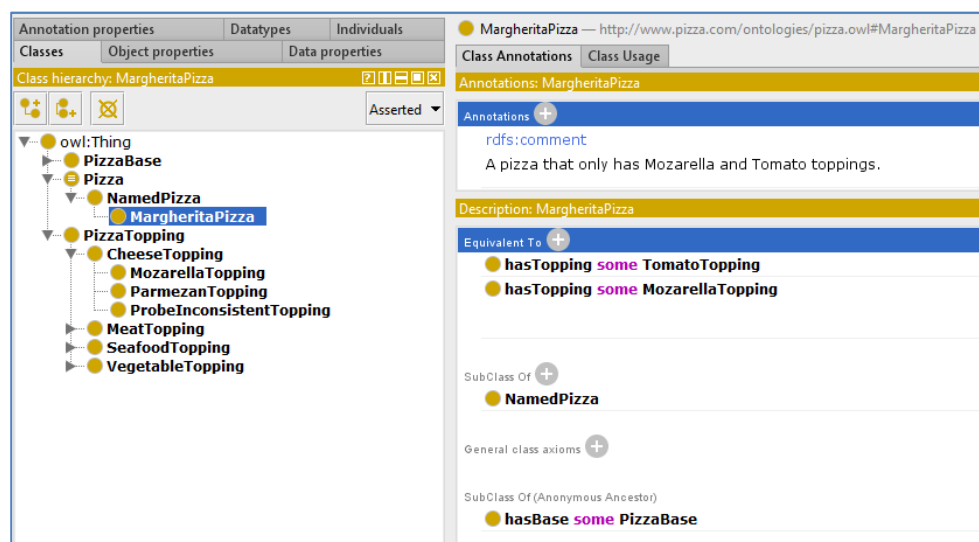
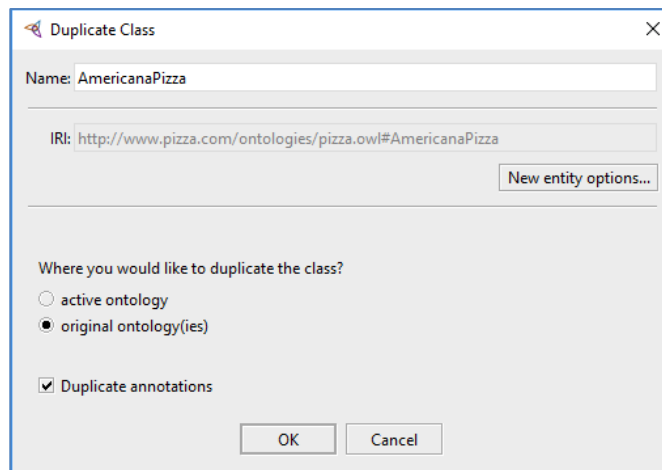
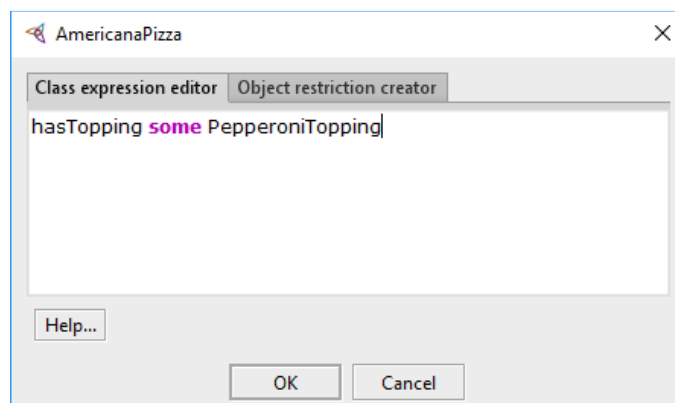


Figura 63- 'Description view' da sub-classe MargheritaPizza.

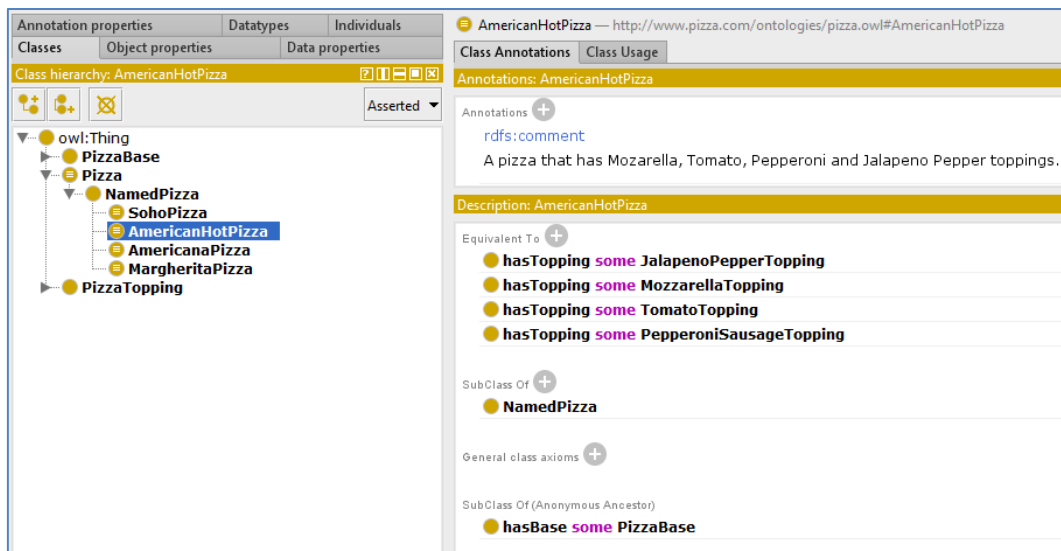
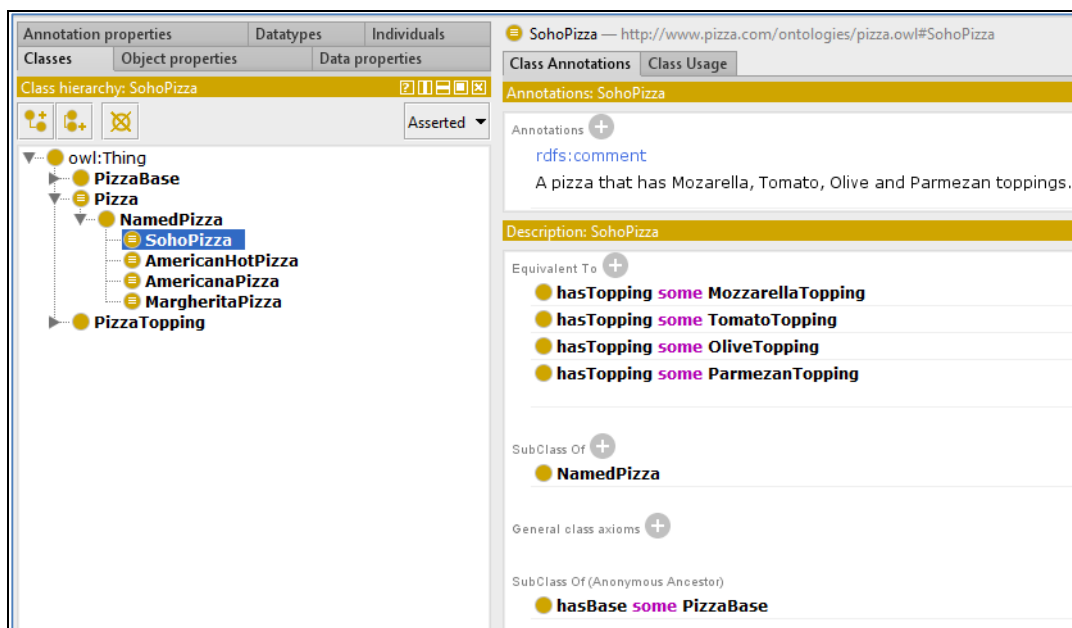
Criar uma classe que represente uma piza americana (AmericanaPizza), com coberturas de *pepperoni*, *mozzarella* e tomate. Como a classe AmericanaPizza é muito semelhante à classe MargheritaPizza, podemos duplicar esta última e acrescentar a quantificação adicional que indica que tem que ter cobertura de *pepperoni*. Isso faz-se com a opção do menu "Edit\Duplicate selected class..."

Representação e Processamento de Conhecimento (RPC)**Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela***Figura 64- Duplicação da classe MargheritaPizza.**Figura 65- Quantificação adicional para a definição da AmericanaPizza.*

Criar uma classe `AmericanHotPizza` e uma classe `SohoPizza`. A `AmericanHotPizza` é semelhante à `AmericanaPizza` mas tem ainda a cobertura `JalapenoPepperTopping`. A `SohoPizza` é semelhante à `MargheritaPizza` mas tem coberturas adicionais `OliveTopping` e `ParmesanTopping`.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

Figura 66- Class description da sub-classe *AmericanHotPizza*.Figura 67- Class description da sub-classe *SohoPizza*.

Tornar as sub-classes de *NamedPizza* disjuntas entre elas, seleccionando a sub-classe *MargheritaPizza* e utilizando a opção do menu “Edit\Make primitive siblings disjoint”.

c) Gravar o projecto

Representação e Processamento de Conhecimento (RPC)**Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela****14. Construir Indivíduos (usando guião)**

Construir uma nova ontologia <http://PAISES>.

Abrir o “01_ProtegeOWLTutorialP4_v1_3.pdf” no capítulo 7 (pág. 89).

a) Tal como indicado na secção 7.1, construir o conceito *Country*.

Criada a classe *Country* e 5 instâncias dessa classe: *America*, *England*, *France*, *Germany* e *Italy*.

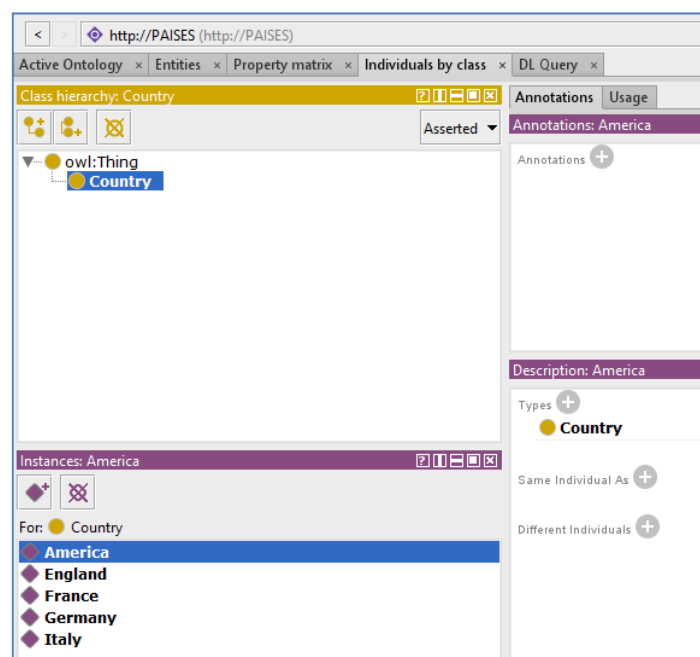


Figura 68- Classe e instâncias criados.

b) Seguir as indicações da secção 7.2 e construir a restrição *hasValue*.

Criar uma restrição *hasValue* para especificar que *MozzarellaTopping* tem a Itália como país de origem.

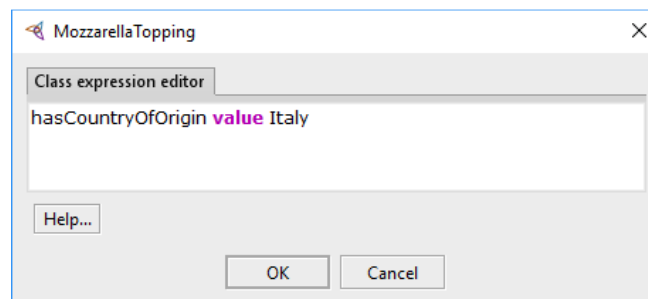


Figura 69- Restrição *hasValue* criada.

Representação e Processamento de Conhecimento (RPC)**Grupo 1 :** N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

- c) Seguir as indicações da secção 7.3 e construir uma “classe enumerada”.
Converter a classe `Country` numa “classe enumerada”.

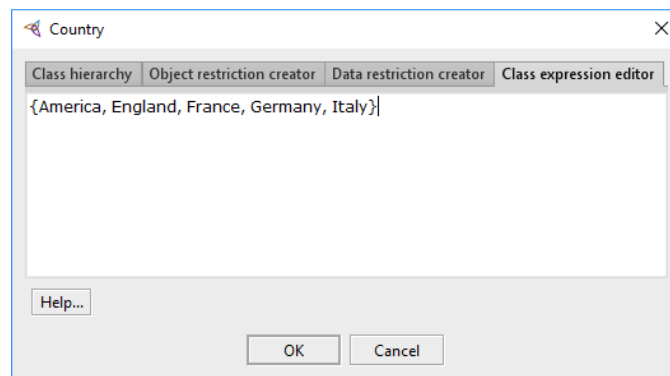


Figura 70- Enumeração da classe `Country`.

- d) Gravar o projecto.

15. Construir Indivíduos (sem usar guião)

Regressar à ontologia “`http://FAMILIA`”.

- a) Definir Pedro e Joana como instâncias de `Pessoa`.

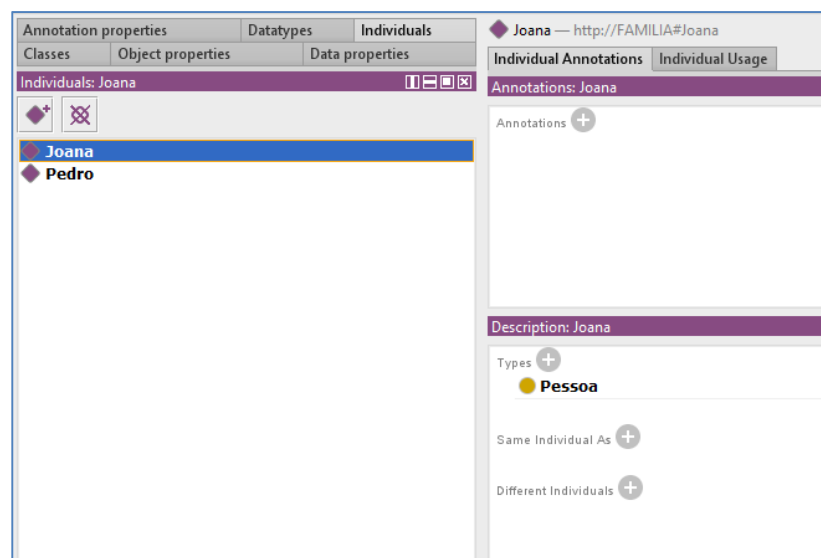


Figura 71- Pedro e Joana definidos como instâncias de `Pessoa`.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

b) Definir Joana também como Feminino.

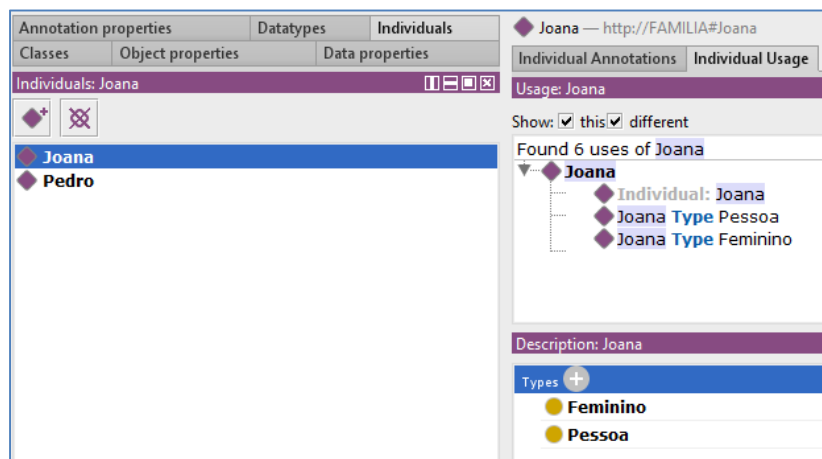


Figura 72- Joana a pertencer aos conceitos Pessoa e Feminino.

c) Indicar que Joana tem como filho Pedro.

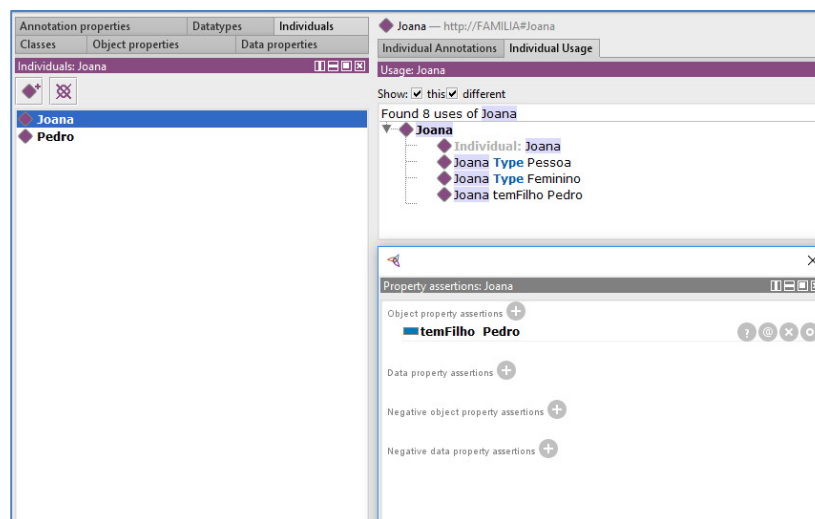


Figura 73- Joana “Property assertions” temFilho Pedro.

d) Fazer inferência (“Reasoner\Start reasoner ou Synchronize reasoner”).

Quando executámos o motor de inferência, surgiu um erro por inconsistência na ontologia. Utilizámos então a opção do menu “Reasoner\Explain inconsistente ontology” para determinarmos a origem da inconsistência.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

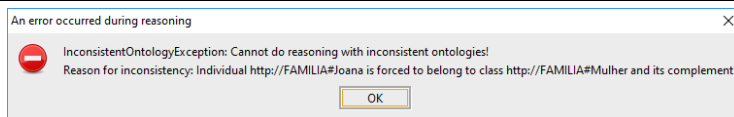


Figura 74- Na primeira execução do “Reasoner” foi encontrada uma inconsistência na ontologia.

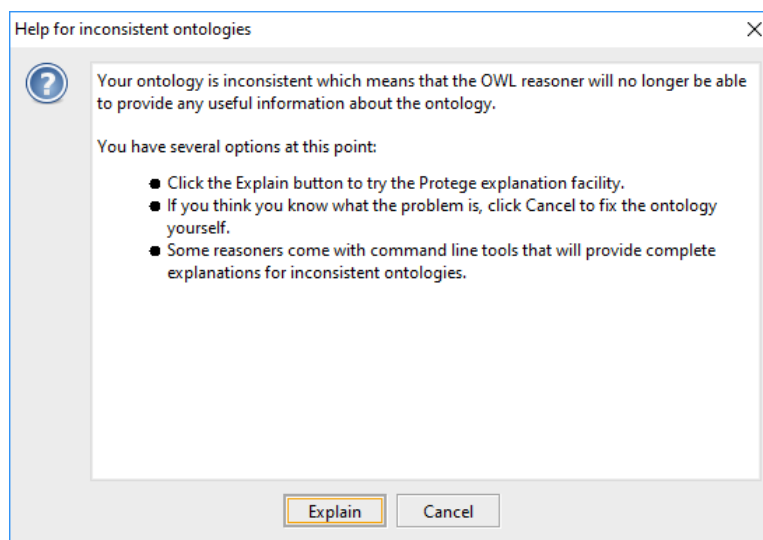


Figura 75- “Reasoner: Explain inconsistent ontology”.

Na explicação, está explícito que a classe `Feminino` é disjunta da classe `Mulher`. Sendo `Mulher` uma sub-classe de `Feminino`, a execução do “Reasoner” marca as classes a vermelho e indica que `Mulher` terá que ser um conjunto vazio “`owl:Nothing`”. Quando definimos a instância `Joana` com sendo do tipo `Feminino` (logo também `Mulher`) surgiu o erro de inconsistência indicado acima.

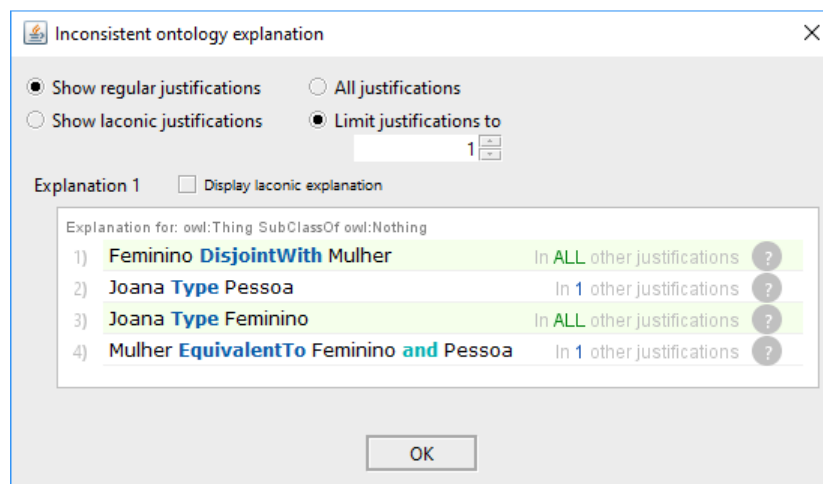


Figura 76- Explicação da inconsistência.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

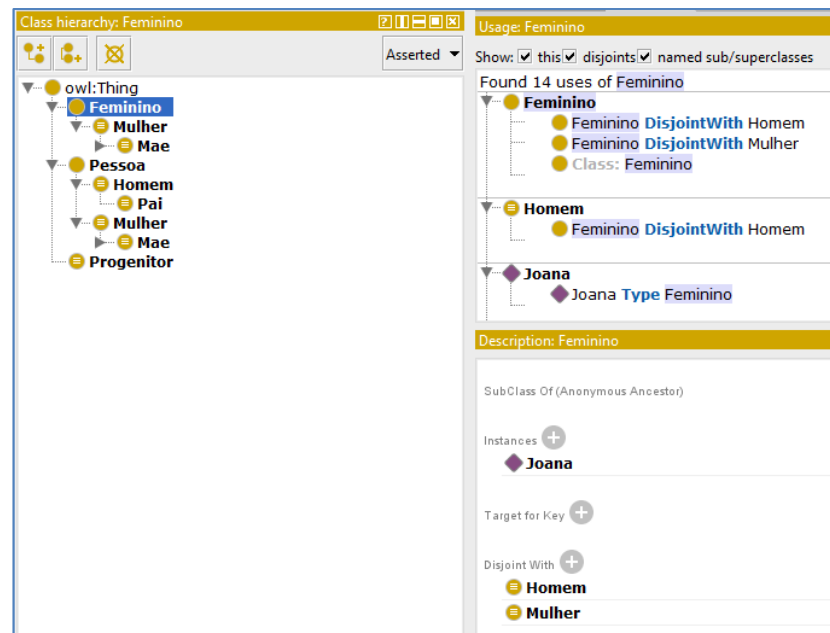


Figura 77- A classe *Feminino* é disjunta das sub-classes *Homem* e *Mulher*.

Para corrigirmos, retirámos a disjunção entre *Feminino* e *Mulher* (tendo permanecido a disjunção correcta entre *Feminino* e *Homem*).

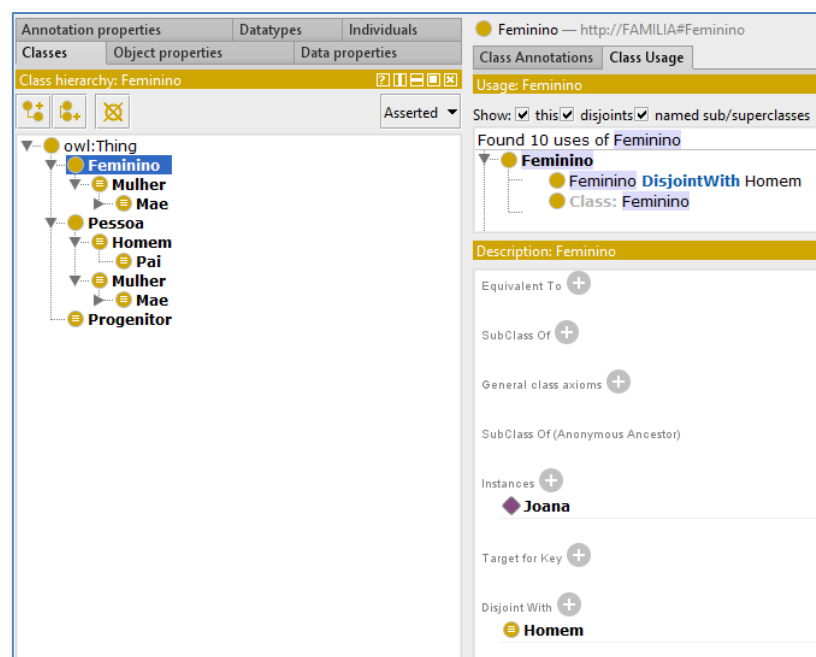


Figura 78- Correção: a classe *Feminino* é disjunta unicamente da sub-classe *Homem*.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

e) Notar o resultado da inferência sobre os indivíduos.

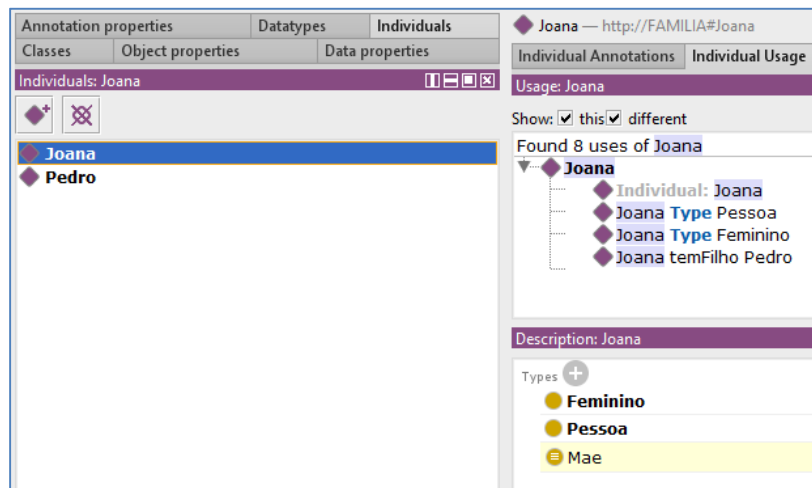


Figura 79- Resultado da inferência: o indivíduo Joana é classificado como Mae.

f) Gravar e ler o ficheiro “família.owl”.

16. Um Problema para Modelar

Construir a ontologia “http://SUPERMERCADO”

a) Construir o modelo que descreve o “mundo”.

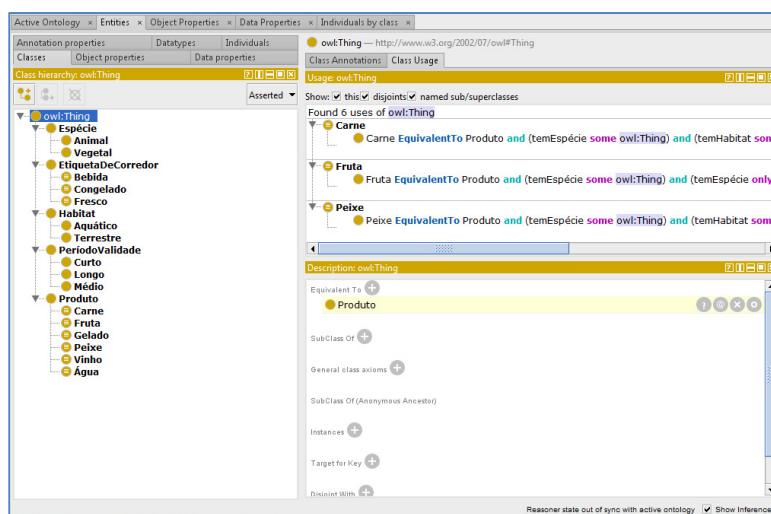


Figura 80- Classes do modelo da ontologia http://SUPERMERCADO.

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

Object Property	Func	Sym	Inv Func	Trans	ASym	Refl	Irrefl	Domain	Range	Inverse
owl:topObjectProperty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
temHabitat	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Produto	Habitat	
temPeríodoValidade	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Produto	PeríodoValidade	
temEspécie	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Produto	Espécie	

Figura 81- Relações do tipo objecto do modelo.

Data Property	Func	Domain	Range
owl:topDataProperty	<input type="checkbox"/>		
temAlcool	<input type="checkbox"/>	Produto	{false, true}
éDoce	<input type="checkbox"/>	Produto	{false, true}
temSabor	<input type="checkbox"/>	Produto	{false, true}

Figura 82- Relações de tipo de dados do modelo.

- b) Adicionar uma “sardinha” como instância de Produto e dizer que ela é “animal que vive na água”. Fazer a inferência “Reasoner\Start reasoner” e verificar a inferência de “sardinha é um Peixe”.

Criámos os seguintes indivíduos:

- Marítimo: instância de “Aquático”;
- Clupeidae: instância de “Animal”;
- Sardinha: instância de “Produto”.

Para a instância “Sardinha”, foram ainda definidas as seguintes ‘objecto property assertions’:

- temHabitat Marítimo;
- temEspécie Clupeidae.

Executado o motor de inferência, verificámos que “Sardinha” passou a estar listado como instância do Produto “Peixe” (Figura 84).

Representação e Processamento de Conhecimento (RPC)

Grupo 1 : N.º 21348 - Nuno Abreu, N.º 8600 - Luís Campanela

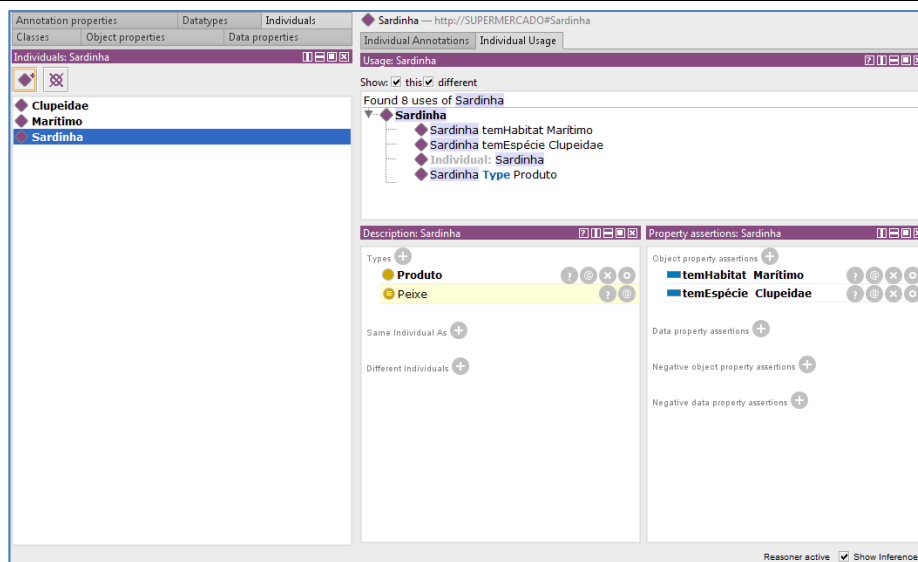


Figura 83- Definição do indivíduo “Sardinha”, como instância de Produto.

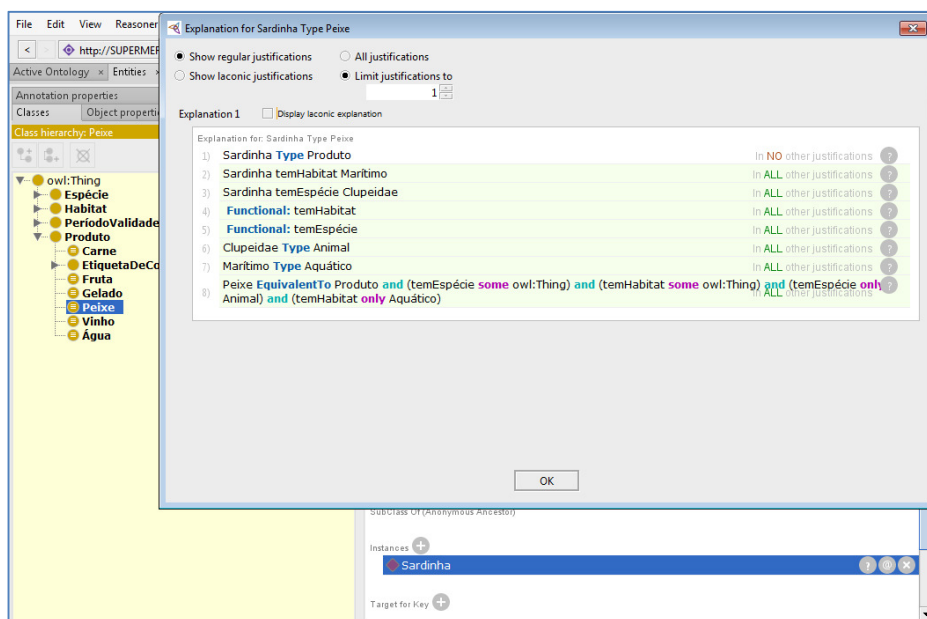


Figura 84- “Sardinha” é um Peixe, por inferência.

- c) Construir 5 indivíduos e mostrar a inferência para concluir “em que corredor está cada um deles”.