



**UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA**

---

Relatórios Técnicos  
do Departamento de Informática Aplicada  
da UNIRIO  
n°0004/2009

# **Um estudo para representação da semântica de diagramas entidade-relacionamento em OWL**

**Mauro Lopes**  
**Jairo Souza**  
**Fernanda Baião**  
**Vanessa Nunes**  
**Claudia Cappelli**

Departamento de Informática Aplicada

---

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
Av. Pasteur, 458, Urca - CEP 22290-240  
RIO DE JANEIRO – BRASIL

# Projeto de Pesquisa

## Grupo de Pesquisa Participante



## Patrocínio



***PETROBRAS***

## Um estudo para representação da semântica de diagramas entidade-relacionamento em OWL \*

Mauro Lopes<sup>2</sup>, Jairo Souza<sup>1,2</sup>, Fernanda Baião<sup>2,3</sup>, Vanessa Nunes<sup>2</sup>, Claudia Cappelli<sup>1,2</sup>

<sup>1</sup>Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO)

<sup>2</sup>Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec)

<sup>3</sup>Departamento de Informática Aplicada (DIA) – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

mauro.lopes@uniriotec.br, jsouza@inf.puc-rio.br, fernanda.baiao@uniriotec.br,  
vanessa.nunes@uniriotec.br, claudia.cappelli@uniriotec.br

**Abstract.** Ontologies and entity-relationship models are used to describe concepts from specific domain. However, these models have constructs with distinct meanings. This work shows a study made for comparing constructs from entity-relationship models and constructs from the ontology language OWL. This work contributes to ontology creation task from entity-relationship models.

**Keywords:** Ontology, entity-relationship model, modeling.

**Resumo.** Ontologias e modelos entidade-relacionamento são usados para descrever os conceitos de um domínio. Contudo, estes modelos possuem construtos com semântica distinta. Este trabalho apresenta o estudo realizado para comparação dos construtos usados em modelos entidade-relacionamento e os construtos da linguagem OWL para descrição de ontologias. O trabalho contribui para a tarefa de criação de ontologias à partir de modelos entidade-relacionamento.

**Palavras-chave:** Ontologia, modelo entidade-relacionamento, modelagem.

---

\* Trabalho patrocinado pela Petrobras.



## Sumário

1	Introdução	8
2	Expressividade do modelo lógico do ERWin	8
2.1	Entidades	9
2.2	Atributos	10
2.2.1	Atributo: Normal	10
2.2.2	Atributo: Chave Primária	11
2.2.3	Atributo: Chave Estrangeira	12
2.2.4	Atributo: Não-nulo	13
2.3	Relacionamentos	14
2.3.1	Relacionamento do tipo categoria	14
2.3.2	Relacionamento do tipo muitos para muitos	15
2.3.3	Relacionamento do tipo um para muitos	16
2.3.4	Relacionamento não identificador	18
3	Expressividade do diagrama entidade-relacionamento	20
3.1	Entidades	20
3.1.1	Entidades primárias	20
3.1.2	Entidade dependente	21
3.1.3	Entidade associativa	23
3.2	Atributos	25
3.2.1	Atributo único ou não-único	25
3.2.2	Atributo obrigatório ou opcional	27
3.2.3	Atributo simples ou composto	28
3.2.4	Atributo univalorado ou multivalorado	30
3.2.5	Atributo relacionante ou descritivo	32
3.2.6	Atributo derivado	33
3.2.7	Atributo identificador	34
3.3	Relacionamento	35
3.3.1	Relacionamento do tipo normal	35
3.3.2	Relacionamento do tipo dependência	36
3.3.3	Relacionamento do tipo associativo	36
3.3.4	Relacionamento do tipo categoria	37
3.3.5	Relacionamento do tipo papel	38
4	Expressividade adicional da linguagem OWL	39
4.1	Operação de conjunto: União	39
4.2	Operações de conjunto: Interseção	40
4.3	Operação de conjunto: Complemento	41
4.4	Operação de conjunto: Classes disjuntas	42
4.5	Operações de conjunto: Classes enumeradas (OneOf)	43
4.6	Relacionamento transitivo	43
4.7	Relacionamento simétrico	44

4.8	Relacionamento funcional	45
4.9	Relacionamento inverso funcional	45
4.10	Relacionamento inverso	45
4.11	Instâncias	45
4.12	Semelhança entre instâncias: sameAs	45
4.13	Semelhança entre propriedades: equivalentProperty	46
4.14	Semelhança entre classes: equivalentClass	47
4.15	Restrição de range “Todos os valores de”: allValuesFrom	48
4.16	Restrição de range “Alguns valores de”: someValuesFrom	49
4.17	Restrição de range “Tem valor”: hasValue	50
5	Conclusão	50
	Referências Bibliográficas	50

## Tabela

Tabela 1 - Construtos primitivos do modelo lógico do ERWin	9
Tabela 2 - Entidade no ERWin	9
Tabela 3 - Atributo: Normal	10
Tabela 4 - Atributo: Chave Primária	11
Tabela 5 - Atributo: Chave Estrangeira	12
Tabela 6 - Atributo não-nulo	13
Tabela 7 - Relacionamento do tipo categoria	14
Tabela 8 - Relacionamento do tipos muitos para muitos	16
Tabela 9 - Relacionamento do tipo um para muitos	17
Tabela 10 - Relacionamento não identificador	19
Tabela 11 - Entidade Primária	21
Tabela 12 - Entidade Dependente	22
Tabela 13 - Entidade Associativa	23
Tabela 14 - Atributo Único	26
Tabela 15 - Atributo Obrigatório	27
Tabela 16 - Atributo Opcional	28
Tabela 17 - Atributo Composto	29
Tabela 18 - Atributo Univalorado	30
Tabela 19 - Atributo Multivalorado	31
Tabela 20 - Atributo Relacionante	32
Tabela 21 - Atributo Descritivo	33
Tabela 22 - Atributo Identificador	34
Tabela 23 - Relacionamento do tipo normal	36
Tabela 24 - Relacionamento do tipo categoria	37
Tabela 25 - Relacionamento do tipo papel	38
Tabela 26 - União	39
Tabela 27 - Interseção	40
Tabela 28 - Complemento	41
Tabela 29 - Classes disjuntas	42
Tabela 30 - Classes enumeradas	43
Tabela 31 - Relacionamento transitivo	44
Tabela 32 - Relacionamento simétrico	44
Tabela 33 - Semelhança entre instâncias	46
Tabela 34 - Semelhança entre propriedades	47
Tabela 35 - Semelhança entre classes	48
Tabela 36 - Restrição allValuesFrom	48
Tabela 37 - Restrição someValuesFrom	49

# 1 Introdução

Este documento tem como objetivo apresentar o estudo realizado sobre modelagem conceitual, comparando com duas formas de representação: diagramas Entidade-Relacionamento [Chen, 1976] e ontologias utilizando a linguagem OWL [Dean *et al.*, 2006], como parte da frente de estudo sobre Utilização de Ontologias em Ambiente Corporativo para Realização de Consultas Semânticas do Projeto de Pesquisa, dentro do Termo de Cooperação em Modelagem de Processos, estabelecido entre a TIC-E&P/GDIEP e o NP2Tec/UNIRIO.

Como explicitado em [Pompilho, 2002], modelagem conceitual de dados objetiva representar uma realidade do mundo, sendo a mais aderente possível a esta realidade proposta. Antes da implementação dos dados, uma descrição da realidade fielmente retratada deve ser desenvolvida para facilitar a percepção do mundo. Diagrama Entidade-Relacionamento é uma forma mais antiga para representação de modelos conceituais utilizada principalmente por pessoal com conhecimento em banco de dados. Ontologias são estruturas para representação de conhecimento fortemente baseado em lógica de descrição [Badder *et al.*, 2003] e que também possuem expressividade suficiente para a construção de modelos conceituais.

Esse relatório pretende analisar as capacidades de expressividade de cada uma dessas formas de representação.

Para tal, a linguagem OWL é comparada com os construtos disponíveis pela ferramenta de modelagem ERWin [ERWin, 2009], atualmente utilizada pela Petrobras, no capítulo 2.

Em seguida, para demonstrar que linguagem OWL possui uma capacidade de expressividade capaz de representar os construtos presentes em um diagrama Entidade-Relacionamento (ou DER), o capítulo 3 apresenta comparações das estruturas de um diagrama ER com um correlato OWL.

Ainda, os construtos da OWL que não são abordados por diagramas entidade-relacionamento são apresentados no capítulo 4.

Para finalizar o documento, são apresentadas as conclusões no capítulo 5, seguido das referências bibliográficas.

## 2 Expressividade do modelo lógico do ERWin

A utilização da ferramenta ERWin para modelagem conceitual se dá através da criação de um modelo que utiliza os mesmos construtos utilizados para modelagem lógica de dados. O modelo lógico do ERWin possui um conjunto de construtos primitivos semelhantes a um modelo conceitual de dados: entidades, relacionamentos e atributos.

Esse conjunto de elementos é representado de forma gráfica pela ferramenta e possuem a mesma semântica dos elementos básicos da linguagem OWL. A Tabela 1 apresenta a comparação destes construtos básicos.



**Tabela 1 - Construtos primitivos do modelo lógico do ERWin**

Construtos primitivos do modelo lógico do ERWin	
ERWin	OWL
Entidade	Class
Relacionamento	ObjectProperty
Atributo	DataTypeProperty

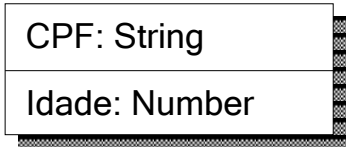

A tabela acima mostra que os construtos básicos do modelo lógico da ferramenta ERWin possuem correspondente na linguagem OWL. As seções seguintes apresentam comparações na expressividade possível ao utilizar os construtos básicos do ERWin e sua correspondência em OWL. As figuras geradas nestas seções foram criadas na ferramenta ERWin. As figuras apresentadas como representação OWL foram geradas pela ferramenta Protégé [PROTÉGÉ, 2009]. Como a linguagem OWL não é gráfica, o código referente a cada figura é apresentado nas tabelas seguintes, uma vez que as ferramentas de visualização OWL não são capazes de representar graficamente todos os construtos OWL.

## 2.1 Entidades

Entidade é uma classificação de algo na realidade, apresentando um conjunto de objetos semelhantes e que podem ser agrupados. As entidades possuem relacionamentos e atributos. Em OWL, essa mesma semântica é expressa com a criação de uma classe.

A Tabela 2 apresenta uma comparação entre uma entidade no ERWin e uma classe em OWL.

**Tabela 2 - Entidade no ERWin**

Entidade
ERWin
<p>Pessoa</p> 
OWL

<p><b>Código correspondente:</b></p> <pre>&lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt; &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="CPF"&gt;</pre>

```

    <rdfs:domain rdf:resource="#Pessoa"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Idade">
    <rdfs:domain rdf:resource="#Pessoa"/>
    <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

```

## 2.2 Atributos

Atributos são características que identificam uma entidade. Os dados de uma entidade são armazenados em forma de atributos, que pode assumir vários formatos, como cadeias de caracteres ou números. No ERWin, os atributos possuem as seguintes categorias: atributo normal, chave primária, chave estrangeira e atributo não-nulo.

### 2.2.1 Atributo: Normal

Atributos do tipo normal são características que identificam uma entidade, mas que não possuem nenhuma informação adicional identificando-a. São atributos que descrevem uma entidade, sem nenhuma restrição a essa descrição. Em OWL, essa mesma semântica é expressa com a criação de uma propriedade de tipo de dado (DatatypeProperty).

A Tabela 3 apresenta uma comparação entre atributo normal do ERWin e uma propriedade de tipo de dado em OWL. No exemplo, a cardinalidade máxima 1 de um atributo do tipo normal é representado a partir do construto "maxCardinality".

**Tabela 3 - Atributo: Normal**

Atributo: Normal						
ERWin						
<div style="text-align: center;"> <p>Pessoa</p> <table border="1"> <tr> <td>CPF: String</td> </tr> <tr> <td>Idade: Number</td> </tr> </table> </div>	CPF: String	Idade: Number				
CPF: String						
Idade: Number						
OWL						
<div style="text-align: center;"> <table border="1"> <tr> <td colspan="2">Pessoa</td> </tr> <tr> <td>CPF</td> <td>String</td> </tr> <tr> <td>Idade</td> <td>Integer</td> </tr> </table> </div>	Pessoa		CPF	String	Idade	Integer
Pessoa						
CPF	String					
Idade	Integer					
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf&gt;   &lt;owl:Restriction&gt;   &lt;owl:onProperty rdf:resource="#Idade"/&gt;   &lt;owl:maxCardinality rdf:datatype="&amp;xsd:int"&gt;1&lt;/owl:maxCardinality&gt; </pre>						

```

    </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="CPF">
    <rdfs:domain rdf:resource="#Pessoa"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Idade">
    <rdfs:domain rdf:resource="#Pessoa"/>
    <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

```

## 2.2.2 Atributo: Chave Primária

Atributos do tipo chave primária identificam uma instância da entidade, sendo um valor único e não-nulo que identifica cada uma das instâncias de uma entidade. Em OWL, essa mesma semântica é expressa com a criação de uma propriedade de tipo de dado (DatatypeProperty) do tipo funcional, que identifica que um valor atribuído a essa propriedade é único.

A Tabela 4 apresenta uma comparação entre um atributo chave primária no ERWin e uma propriedade de tipo de dado do tipo funcional em OWL.

**Tabela 4 - Atributo: Chave Primária**

Atributo: Chave Primária						
ERWin						
<p>Pessoa</p> <table border="1"> <tr> <td>CPF: String</td> </tr> <tr> <td>Idade: Number</td> </tr> </table>	CPF: String	Idade: Number				
CPF: String						
Idade: Number						
OWL						
<table border="1"> <tr> <td colspan="2">Pessoa</td> </tr> <tr> <td>CPF</td> <td>String</td> </tr> <tr> <td>Idade</td> <td>Integer</td> </tr> </table>	Pessoa		CPF	String	Idade	Integer
Pessoa						
CPF	String					
Idade	Integer					
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;     &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt; &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="CPF"&gt;     &lt;rdfs:type rdf:resource="&amp;owl;FunctionalProperty"/&gt;     &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;     &lt;rdfs:range rdf:resource="&amp;xsd:string"/&gt; &lt;/owl:DatatypeProperty&gt; &lt;owl:DatatypeProperty rdf:ID="Idade"&gt;     &lt;rdfs:domain rdf:resource="#Pessoa"/&gt; </pre>						

```
<rdfs:range rdf:resource="&xsd:int" />
</owl:DatatypeProperty>
```

### 2.2.3 Atributo: Chave Estrangeira

Atributos do tipo chave estrangeira acrescentam informações sobre os relacionamentos entre as classes, através da presença do(s) atributo(s) do tipo chave primária da entidade (no exemplo, "Consumidor") que se relaciona com a entidade participante do relacionamento (no exemplo, "Compra"). Em OWL, essa mesma semântica é expressa com a criação de uma propriedade de objeto (ObjectProperty), que não apresenta uma propriedade de tipo de dado (DatatypeProperty) de forma semelhante aos exemplos de outros tipos de atributos, pois um relacionamento em OWL não necessita da criação deste tipo de propriedade. Como o relacionamento no ERWin é bidirecional, um relacionamento inverso é necessário para representar por completo as cardinalidades dos relacionamentos entre as classes.

A Tabela 5 apresenta uma comparação entre chave estrangeira do ERWin e propriedades de objeto da OWL.

**Tabela 5 - Atributo: Chave Estrangeira**

Atributo: Chave Estrangeira
ERWin
OWL
<p><b>Código correspondente:</b></p> <pre>&lt;owl:ObjectProperty rdf:ID="Realiza_Compra"&gt;   &lt;rdf:type rdf:resource="&amp;owl;InverseFunctionalProperty" /&gt;   &lt;rdfs:domain rdf:resource="&amp;#Consumidor" /&gt;   &lt;rdfs:range rdf:resource="&amp;#Compra" /&gt;   &lt;owl:inverseOf rdf:resource="&amp;#E_feita_pelo_consumidor" /&gt;</pre>

```

</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="E_feita_pelo_consumidor">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Compra" />
  <rdfs:range rdf:resource="#Consumidor" />
  <owl:inverseOf rdf:resource="#Realiza_Compra" />
</owl:ObjectProperty>

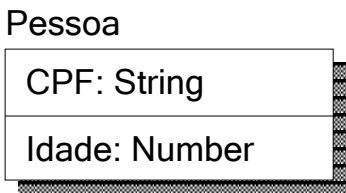

```

## 2.2.4 Atributo: Não-nulo

Atributo do tipo não-nulo são características da entidade que devem existir e não podem estar em branco. Em OWL, essa mesma semântica é expressa com a criação de uma propriedade de tipo de dado (DatatypeProperty) com cardinalidade mínima 1. Foi feita uma alteração no exemplo já apresentado.

A Tabela 6 apresenta a comparação entre um atributo não-nulo em ERWin e uma propriedade de tipo de dado em OWL (no exemplo, "Idade").

**Tabela 6 - Atributo não-nulo**

Atributo: Não-nulo
ERWin
<div style="text-align: center;">  </div>
OWL
<div style="text-align: center;">  </div>
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="#Idade" /&gt;       &lt;owl:minCardinality rdf:datatype="&amp;xsd;int"&gt;1&lt;/owl:minCardinality&gt;     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing" /&gt; &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="CPF"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa" /&gt;   &lt;rdfs:range rdf:resource="&amp;xsd;string" /&gt; &lt;/owl:DatatypeProperty&gt; &lt;owl:DatatypeProperty rdf:ID="Idade"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa" /&gt; </pre>

```
<rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>
```

## 2.3 Relacionamentos

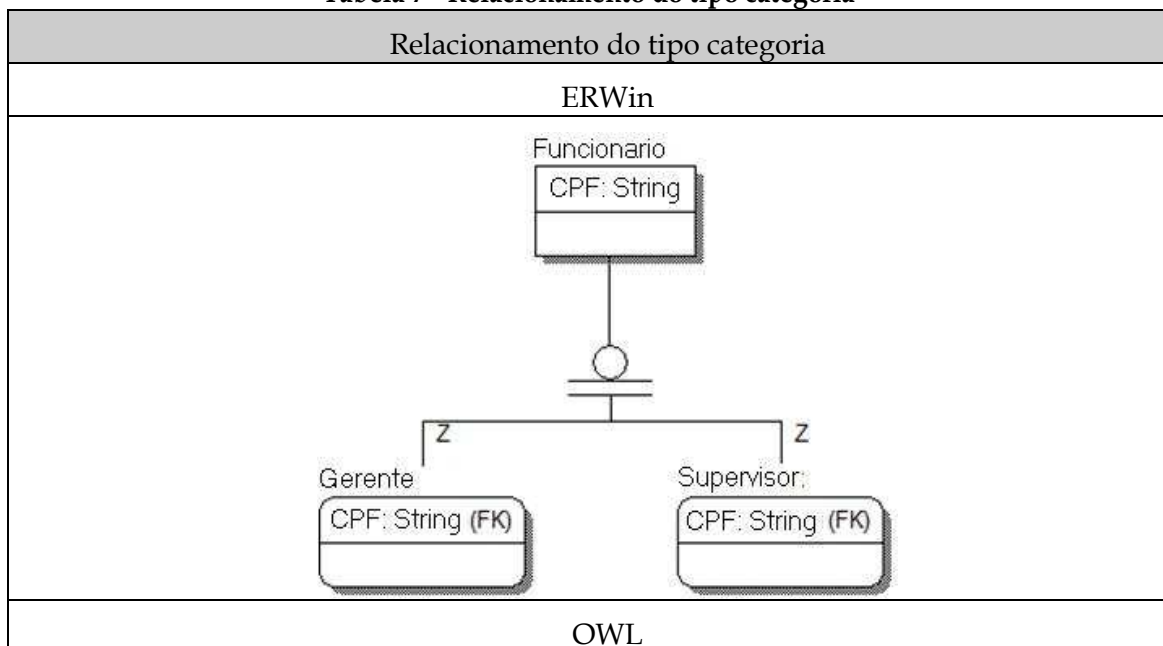
Relacionamento é uma associação entre duas e somente duas entidades. Em modelagem conceitual, é comum a presença de um atributo compartilhado entre duas entidades. No modelo lógico do ERWin, os seguintes tipos de relacionamentos estão disponíveis: relacionamento do tipo categoria, relacionamento do tipo um para muitos, relacionamento do tipo muitos para muitos, e relacionamento não identificador.

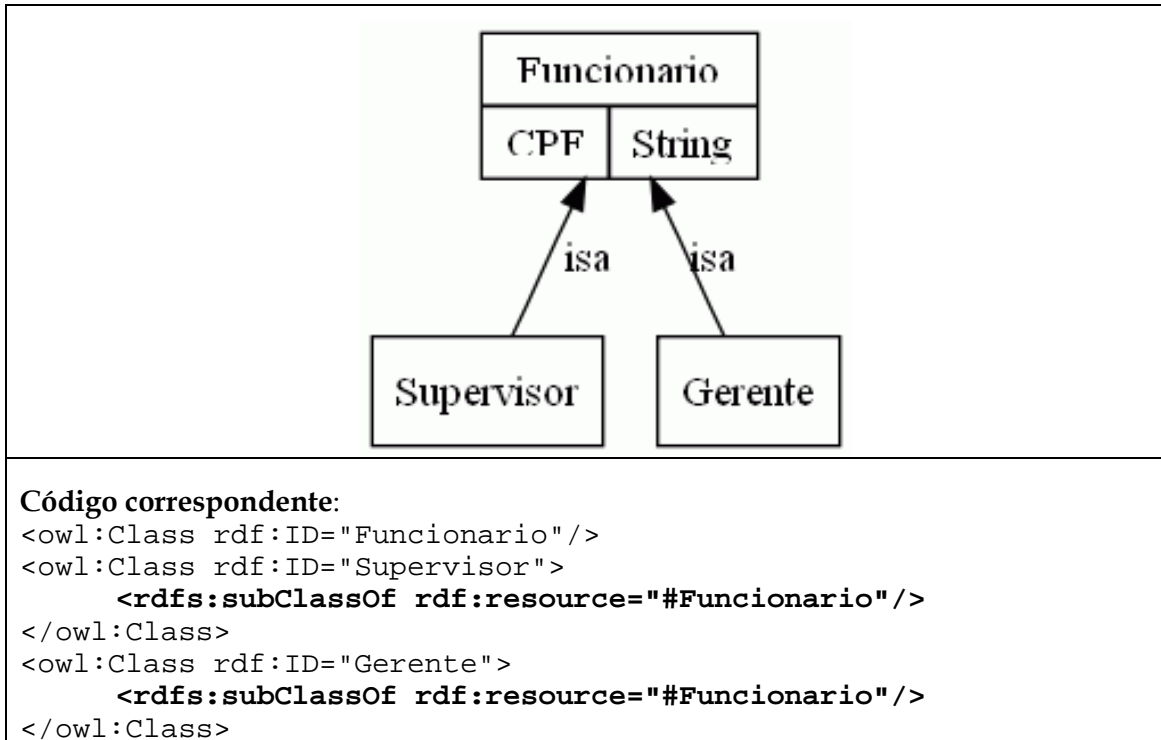
### 2.3.1 Relacionamento do tipo categoria

Relacionamento do tipo categoria é um tipo de relacionamento que identifica uma entidade que possui características referentes à outra entidade, especializando essas características, com novos atributos que a identificam. Essa nova entidade herda as características da entidade que a identifica. Em OWL, essa mesma semântica é expressa com a criação de subclasses.

A Tabela 7 apresenta uma comparação entre o tipo de relacionamento categoria do ERWin e a subclasse da OWL. Na figura do OWL gerada, o relacionamento “isa” representa a idéia de hierarquia, mas não é um construto OWL.

Tabela 7 - Relacionamento do tipo categoria





### 2.3.2 Relacionamento do tipo muitos para muitos

Relacionamento do tipo muitos para muitos representa uma ligação entre uma entidade a outra em que as cardinalidades do relacionamento representam que muitas entidades participam de cada lado do relacionamento. Em OWL, essa mesma semântica é expressa com a criação de uma propriedade de objeto (ObjectProperty) sem nenhuma alteração pois as propriedades de objeto já possuem cardinalidades maior que um. Como o relacionamento no ERWin é bidirecional, um relacionamento inverso é necessário para representar por completo as cardinalidades dos relacionamentos entre as classes. Este tipo de relacionamento também identifica a idéia de entidade fraca, conceito não presente em OWL.

A Tabela 8 apresenta uma comparação entre o relacionamento do tipo muitos para muitos do ERWin e de uma propriedade de objeto em OWL.

**Tabela 8 - Relacionamento do tipos muitos para muitos**

Relacionamento do tipo muitos para muitos
ERWin
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:ObjectProperty rdf:ID="Participa_de_Disciplina"&gt;   &lt;rdfs:domain rdf:resource="#Estudante"/&gt;   &lt;rdfs:range rdf:resource="#Disciplina"/&gt;   &lt;owl:inverseOf rdf:resource="#Possui_Estudante"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:ObjectProperty rdf:ID="Possui_Estudante"&gt;   &lt;rdfs:domain rdf:resource="#Disciplina"/&gt;   &lt;rdfs:range rdf:resource="#Estudante"/&gt;   &lt;owl:inverseOf rdf:resource="#Participa_de_Disciplina"/&gt; &lt;/owl:ObjectProperty&gt; </pre>

### 2.3.3 Relacionamento do tipo um para muitos

Relacionamento do tipo um para muitos representa uma ligação entre uma entidade a outra em que as cardinalidades do relacionamento representam que uma entidade participam de um relacionamento com muitas outras entidades. Esse tipo de ligação acrescenta identidade à entidade que possui a cardinalidade maior que um (no exemplo, “Compra”). Isto ocorre devido à criação de um atributo do tipo chave estrangeira como chave primária desta entidade. Em OWL, essa mesma semântica é expressa com a criação de uma propriedade de objeto (ObjectProperty) em que o relacionamento que



liga a entidade que se relaciona com somente um de outra entidade possui cardinalidade um. Como o relacionamento no ERWin é bidirecional, um relacionamento inverso é necessário para representar por completo as cardinalidades dos relacionamentos entre as classes. Este tipo de relacionamento também identifica a idéia de entidade fraca, conceito não presente em OWL.

A Tabela 9 apresenta a comparação entre o relacionamento do tipo um para muitos do ERWin com a propriedade de objeto da OWL.

**Tabela 9 - Relacionamento do tipo um para muitos**

Relacionamento do tipo um para muitos
ERWin
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Compra"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt;   &lt;rdfs:subClassOf&gt;   &lt;owl:Restriction&gt;     &lt;owl:onProperty <b>rdf:resource="#E_feita_pelo_consumidor"/&gt;</b>     &lt;owl:maxCardinality <b>rdf:datatype="&amp;xsd;int"&gt;1&lt;/owl:maxCardinality&gt;</b>     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt; &lt;owl:ObjectProperty rdf:ID="Realiza_Compra"&gt;   &lt;rdf:type rdf:resource="&amp;owl;InverseFunctionalProperty"/&gt; </pre>

```
<rdfs:domain rdf:resource="#Consumidor"/>
<rdfs:range rdf:resource="#Compra"/>
<owl:inverseOf rdf:resource="#E_feita_pelo_consumidor"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="E_feita_pelo_consumidor">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Compra"/>
  <rdfs:range rdf:resource="#Consumidor"/>
  <owl:inverseOf rdf:resource="#Realiza_Compra"/>
</owl:ObjectProperty>
```

### 2.3.4 Relacionamento não identificador

Relacionamento não identificador representa uma ligação entre duas entidades em que as cardinalidades do relacionamento representam que uma entidade participa de um relacionamento com muitas outras entidades. Esse tipo de ligação acrescenta informações à entidade que possui a cardinalidade maior que um (no exemplo, "Compra"). Isto ocorre devido à criação de um atributo do tipo chave estrangeira. Em OWL, essa mesma semântica é expressa com a criação de uma propriedade de objeto (ObjectProperty) em que o relacionamento que liga a entidade que se relaciona com somente um de outra entidade possui cardinalidade um. Como o relacionamento no ERWin é bidirecional, um relacionamento inverso é necessário para representar por completo as cardinalidades dos relacionamentos entre as classes.

A Tabela 10 apresenta a comparação entre o relacionamento não identificador do ERWin com a propriedade de objeto da OWL.

Tabela 10 - Relacionamento não identificador

Relacionamento não identificador
ERWin
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Carro"&gt;   &lt;rdfs:subClassOf rdf:resource="#owl:Thing"/&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="#E_possuido_por_Pessoa"/&gt;       &lt;owl:maxCardinality rdf:datatype="&amp;xsd:int"&gt;1&lt;/owl:maxCardinality&gt;     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt; &lt;owl:ObjectProperty rdf:ID="Possui_Carro"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="#Carro"/&gt;   &lt;owl:inverseOf rdf:resource="#E_possuido_por_Pessoa"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:ObjectProperty rdf:ID="E_possuido_por_Pessoa"&gt;   &lt;rdfs:domain rdf:resource="#Carro"/&gt;   &lt;rdfs:range rdf:resource="#Pessoa"/&gt;   &lt;owl:inverseOf rdf:resource="#Possui_Carro"/&gt; &lt;/owl:ObjectProperty&gt; </pre>

### 3 Expressividade do diagrama entidade-relacionamento

O diagrama Entidade-Relacionamento (DER) possui todos os construtos existentes no modelo lógico do ERWin, apresentado na seção anterior. Porém, esse tipo de diagrama possui construtos que não estão disponíveis na ferramenta ERWin, os quais adicionam maior poder de expressividade ao DER.

As seções seguintes apresentam as capacidades de expressão do DER, como apresentado em [Chen, 1976] e [Pompilho, 2002]. Como notação para o DER, foi utilizada a notação de Peter Chen [Chen, 1976] com a adição de algumas notações presentes no livro de Pompilho [Pompilho, 2002] e as figuras foram geradas na ferramenta Gliffy [Gliffy, 2009].

#### 3.1 Entidades

As entidades no DER podem ser classificadas levando em consideração os atributos que as identificam. Assim, entidades podem ser classificadas em entidades primárias, dependentes ou associativas.

##### 3.1.1 Entidades primárias

Uma entidade é entidade primária quando suas características a identificam por si só, sem depender de qualquer outra coisa, sendo identificado simplesmente pelos seus atributos. Esta classificação de entidade primária é equivalente a classes em OWL, que também são identificadas simplesmente pelos seus atributos.

A Tabela 11 apresenta a comparação entre entidade primária do DER e classes da OWL.

Tabela 11 - Entidade Primária

Entidade Primária						
DER						
<pre> classDiagram     class Pessoa     Pessoa --&gt; CPF     Pessoa --&gt; Idade         </pre>						
OWL						
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="2" style="text-align: center;"><b>Pessoa</b></td> </tr> <tr> <td style="text-align: center;">CPF</td> <td style="text-align: center;">String</td> </tr> <tr> <td style="text-align: center;">Idade</td> <td style="text-align: center;">Integer</td> </tr> </table>	<b>Pessoa</b>		CPF	String	Idade	Integer
<b>Pessoa</b>						
CPF	String					
Idade	Integer					
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf rdf:resource="#owl:Thing"/&gt; &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="CPF"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="#xsd:string"/&gt; &lt;/owl:DatatypeProperty&gt; &lt;owl:DatatypeProperty rdf:ID="Idade"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="#xsd:int"/&gt; &lt;/owl:DatatypeProperty&gt;         </pre>						

### 3.1.2 Entidade dependente

Uma entidade é entidade dependente quando sua existência é identificada pela existência de outra entidade, da qual é dependente. Sua identidade é completa através de um relacionamento com a entidade da qual é dependente.

Esse tipo de construto não possui um construto equivalente na linguagem OWL e a linguagem não permite expressar a semântica de dependência. Entretanto, caso seja imprescindível expressar essa semântica na linguagem OWL, pode-se inserir essa semântica na linguagem OWL através de uma notação textual, no caso nomeando o relacionamento entre a classe dependente e a entidade da qual esta depende de uma forma específica (no exemplo, o texto "Dependencia\_de\_" em conjunto com o nome da classe

dependente é usado), acrescentando a característica funcional e inverso funcional aos relacionamentos envolvidos.

A Tabela 12 apresenta a comparação entre entidade dependente do DER e a estrutura proposta em OWL.

**Tabela 12 - Entidade Dependente**

Entidade Dependente
DER
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Funcionario"/&gt; &lt;owl:Class rdf:ID="Dependente"/&gt; &lt;owl:ObjectProperty rdf:ID="Inverso_de_Dependencia_de_Dependente"&gt;   &lt;rdf:type rdf:resource="&amp;owl;FunctionalProperty"/&gt;   &lt;rdf:type rdf:resource="&amp;owl;InverseFunctionalProperty"/&gt;   &lt;rdfs:domain rdf:resource="#Funcionario"/&gt;   &lt;rdfs:range rdf:resource="#Dependente"/&gt;   &lt;owl:inverseOf rdf:resource="#Dependencia_de_Dependente"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:ObjectProperty rdf:ID="Dependencia_de_Dependente"&gt;   &lt;rdf:type rdf:resource="&amp;owl;FunctionalProperty"/&gt; </pre>

```

<rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
<rdfs:domain rdf:resource="#Dependente"/>
<rdfs:range rdf:resource="#Funcionario"/>
<owl:inverseOf
rdf:resource="#Inverso_de_Dependencia_de_Dependente"/>
</owl:ObjectProperty>

```

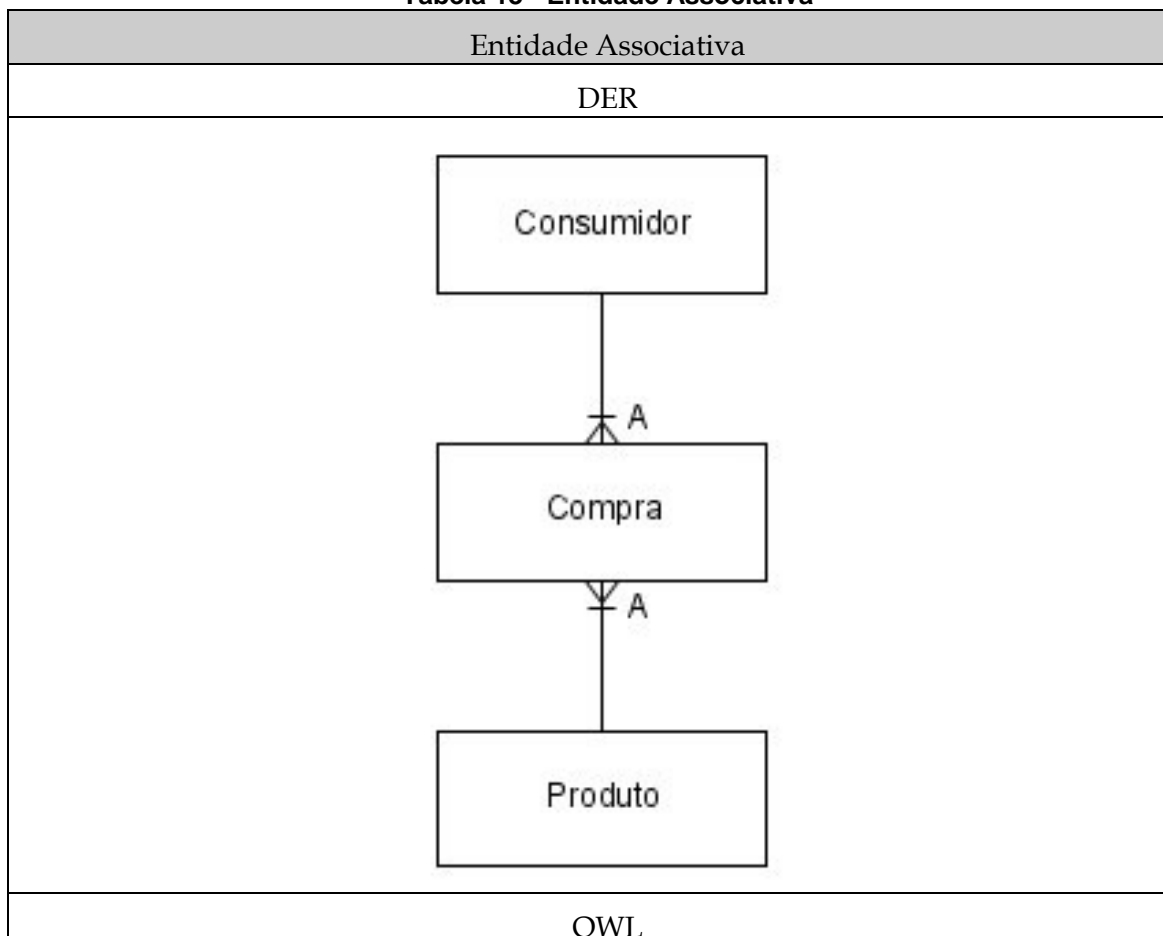
### 3.1.3 Entidade associativa

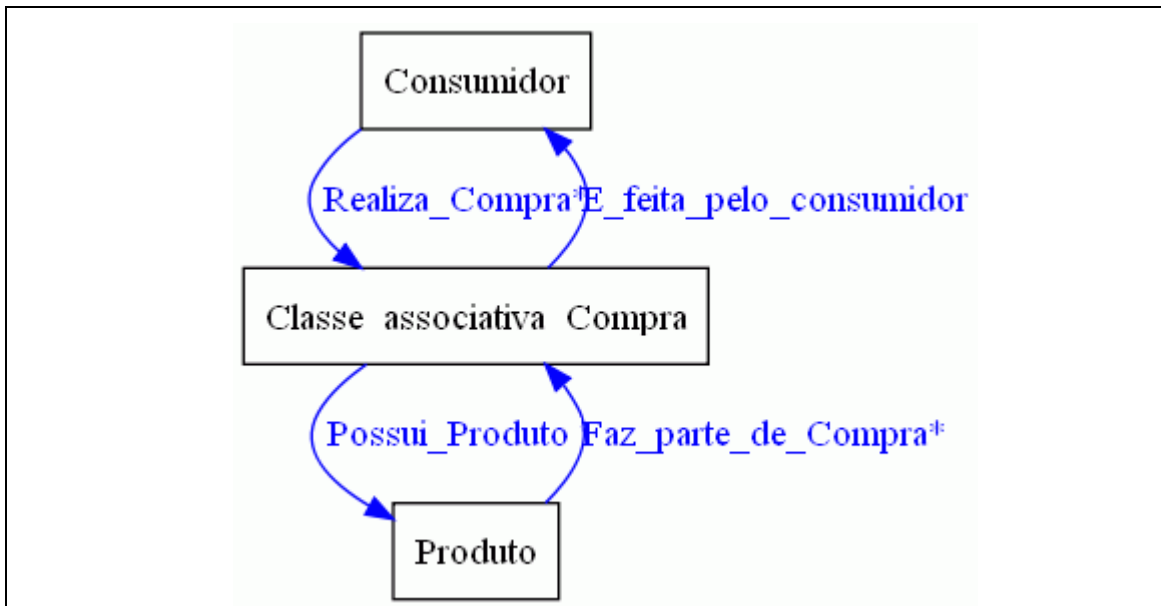
Uma entidade é entidade associativa quando sua existência é identificada pela existência de duas ou mais entidades, sendo concebidas a partir destas. A entidade associativa possui relacionamentos do tipo associativo com as outras entidades. Este tipo de relacionamento será abordado em 3.3.3 .

Esse tipo de construto não possui um construto equivalente na linguagem OWL e a linguagem não permite expressar a semântica de relacionamentos associativos. Entretanto, caso seja imprescindível expressar essa semântica na linguagem OWL, pode-se inserir essa semântica na linguagem OWL através de uma notação textual, no caso nomeando a classe de forma específica (no exemplo, o texto “Classe\_associativa\_” com o acréscimo do nome da classe é usado para representar a classe associativa).

A Tabela 13 apresenta a comparação entre entidade associativa do DER e a estrutura proposta em OWL.

**Tabela 13 - Entidade Associativa**





**Código correspondente:**

```

<owl:Class rdf:ID="Classe_associativa_Compra">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#E_feita_pelo_consumidor"/>
        <owl:maxCardinality
rdf:datatype="&xsd:int">1</owl:maxCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#owl:Thing"/>
  </owl:Class>
<owl:Class rdf:ID="Consumidor"/>
<owl:Class rdf:ID="Produto"/>
<owl:ObjectProperty rdf:ID="Faz_parte_de_Compra">
  <rdfs:domain rdf:resource="#Produto"/>
  <rdfs:range rdf:resource="#Classe_associativa_Compra"/>
  <owl:inverseOf rdf:resource="#Possui_Produto"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Possui_Produto">
  <rdfs:domain rdf:resource="#Classe_associativa_Compra"/>
  <rdfs:range rdf:resource="#Produto"/>
  <owl:inverseOf rdf:resource="#Faz_parte_de_Compra"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Faz_parte_de_Compra">
  <rdfs:domain rdf:resource="#Produto"/>
  <rdfs:range rdf:resource="#Classe_associativa_Compra"/>
  <owl:inverseOf rdf:resource="#Possui_Produto"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Possui_Produto">
  <rdfs:domain rdf:resource="#Classe_associativa_Compra"/>
  <rdfs:range rdf:resource="#Produto"/>
  <owl:inverseOf rdf:resource="#Faz_parte_de_Compra"/>
</owl:ObjectProperty>
  
```



## 3.2 Atributos

Atributos são características e propriedades que identificam uma entidade. Os dados de uma entidade são armazenados em forma de atributos, que pode assumir vários formatos, como cadeias de caracteres ou números. Atributos podem ser classificados nas seguintes categorias: único ou não único, obrigatório ou opcional, simples ou composto, univalorado ou multivalorado, relacionante ou descritivo, derivado ou não-derivado, e identificador ou não-identificador.

### 3.2.1 Atributo único ou não-único

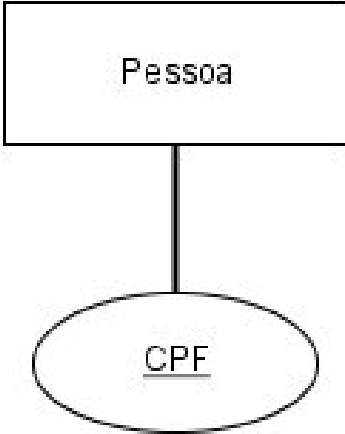
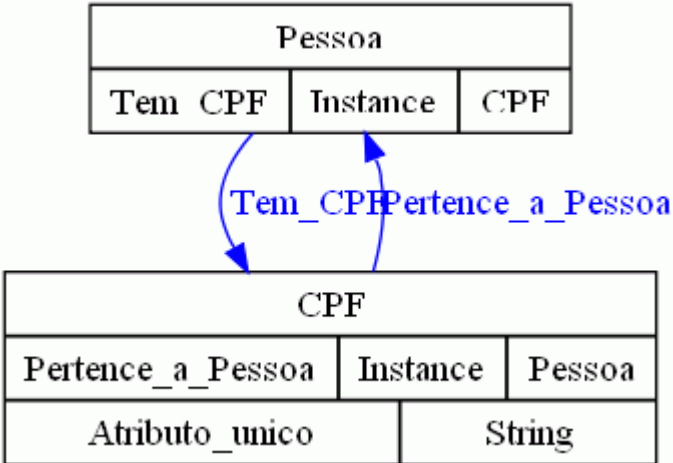
Um atributo é único quando cada instância da entidade descrita possui um valor único para o atributo. O oposto desse tipo de atributo é o não-único, indicando que os valores podem ser repetidos em instâncias diferentes.

O atributo único não possui um construto equivalente em OWL, mas a semântica deste pode ser representada com a reificação do atributo único em uma nova classe, que terá propriedades de objeto com a classe que a possui (funcional e inverso funcional) e uma propriedades de tipo de dado funcional, com o nome "Atributo\_Unico", como apresentado no exemplo.

Este tipo de atributo pode ser representado em OWL de outra forma, caso este esteja no formato OWL-Full, permitindo que um `datatypeProperty` seja funcional inverso, ou seja, o valor (seja texto ou número) estará relacionado somente a uma única instância de uma classe.

A Tabela 14 apresenta a comparação entre atributo único do DER e estrutura reificada proposta em OWL.

**Tabela 14 - Atributo Único**

Atributo Único
DER
 <p>The diagram shows a rectangular box labeled 'Pessoa' at the top. A vertical line connects the bottom center of this box to the top center of an oval labeled 'CPF' below it.</p>
OWL
 <p>The diagram shows two class boxes. The top box is labeled 'Pessoa' and has three columns: 'Tem CPF', 'Instance', and 'CPF'. The bottom box is labeled 'CPF' and has three columns: 'Pertence_a_Pessoa', 'Instance', and 'Pessoa'. Below the 'CPF' box, there are two columns: 'Atributo_unico' and 'String'. Two blue curved arrows connect the 'Tem CPF' column of the 'Pessoa' box to the 'Pertence_a_Pessoa' column of the 'CPF' box, and vice versa. The text 'Tem_CPF' and 'Pertence_a_Pessoa' is written in blue between the arrows.</p>
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"/&gt; &lt;owl:ObjectProperty rdf:ID="Pertence_a_Pessoa"&gt;   &lt;rdf:type rdf:resource="&amp;owl;FunctionalProperty"/&gt;   &lt;rdf:type rdf:resource="&amp;owl;InverseFunctionalProperty"/&gt;   &lt;rdfs:domain rdf:resource="#CPF"/&gt;   &lt;rdfs:range rdf:resource="#Pessoa"/&gt;   &lt;owl:inverseOf rdf:resource="#Tem_CPF"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:ObjectProperty rdf:ID="Tem_CPF"&gt;   &lt;rdf:type rdf:resource="&amp;owl;FunctionalProperty"/&gt;   &lt;rdf:type rdf:resource="&amp;owl;InverseFunctionalProperty"/&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="#CPF"/&gt;   &lt;owl:inverseOf rdf:resource="#Pertence_a_Pessoa"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:Class rdf:ID="CPF"/&gt; &lt;owl:DatatypeProperty rdf:ID="Atributo_unico"&gt;   &lt;rdf:type rdf:resource="&amp;owl;FunctionalProperty"/&gt; </pre>

```

<rdfs:domain rdf:resource="#CPF"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

### 3.2.2 Atributo obrigatório ou opcional

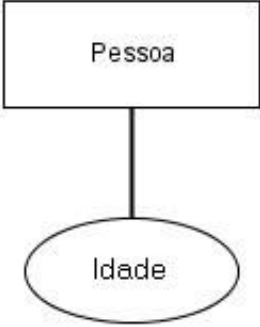
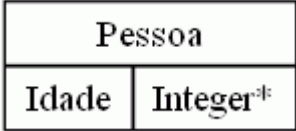
Um atributo é obrigatório quando deve ser sempre preenchido, diferente de nulo ou em branco. O oposto desse tipo de atributo é opcional, indicando que pode ser preenchido ou não.

Em OWL, o atributo obrigatório é equivalente a um `datatypeProperty` com cardinalidade mínima 1, representada pelo “Nome” no exemplo da Tabela 15. Sem essa regra, o `datatypeProperty` é equivalente ao tipo de atributo opcional, apresentado na Tabela 16. A ferramenta utilizada para gerar as figuras do DER das tabelas abaixo não possui uma notação para atributos obrigatórios ou opcionais.

**Tabela 15 - Atributo Obrigatório**

Atributo Obrigatório
DER
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="#Nome"/&gt;       &lt;owl:minCardinality rdf:datatype="&amp;xsd:int"&gt;1&lt;/owl:minCardinality&gt;       &lt;/owl:Restriction&gt;     &lt;/rdfs:subClassOf&gt;   &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="Nome"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="&amp;xsd:string"/&gt; &lt;/owl:DatatypeProperty&gt; </pre>

**Tabela 16 - Atributo Opcional**

Atributo Opcional
DER

OWL

<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt; &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="Idade"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="&amp;xsd;int"/&gt; &lt;/owl:DatatypeProperty&gt; </pre>

### 3.2.3 Atributo simples ou composto

Um atributo é simples quando seu domínio é composto por valores simples, contendo somente uma informação. O oposto desse tipo de atributo é composto, indicando que várias informações estão presentes em um atributo.

O atributo composto não possui um construto equivalente em OWL, mas a semântica deste pode ser representada com a reificação do atributo composto em uma nova classe, que terá propriedades de tipo de dado com o nome equivalente ao original, como apresentado no exemplo.

A Tabela 17 apresenta a comparação de atributo composto do DER e o `datatypeProperty` em OWL.

**Tabela 17 - Atributo Composto**

Atributo Composto
DER
<pre> classDiagram     class Pessoa     class Endereco     class Rua     class Complemento     class Bairro     Pessoa -- Endereco     Endereco -- Rua     Endereco -- Complemento     Endereco -- Bairro         </pre>
OWL
<pre> classDiagram     class Pessoa     class Endereco     class Possui_Endereco     class Rua     class Complemento     class Bairro     Pessoa -- Possui_Endereco     Possui_Endereco -- Endereco     Endereco -- Rua     Endereco -- Complemento     Endereco -- Bairro         </pre>
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"/&gt; &lt;owl:Class rdf:ID="Endereco"/&gt; &lt;owl:ObjectProperty rdf:ID="Possui_Endereco"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="#Endereco"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:DatatypeProperty rdf:ID="Bairro"&gt;   &lt;rdf:type rdf:resource="&amp;owl;FunctionalProperty"/&gt;   &lt;rdfs:domain rdf:resource="#Endereco"/&gt;   &lt;rdfs:range rdf:resource="&amp;xsd:string"/&gt;         </pre>

```

</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Complemento">
  <rdfs:domain rdf:resource="#Endereco"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Rua">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Endereco"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```


### 3.2.4 Atributo univalorado ou multivalorado

Um atributo é univalorado quando esse atributo só pode ter um valor preenchido. O oposto desse tipo de atributo é multivalorado, indicando que pode ter mais de um valor preenchido.

Em OWL, um atributo univalorado é equivalente a um `datatypeProperty` com cardinalidade máxima 1, enquanto que um atributo multivalorado é equivalente a um `datatypeProperty` sem nenhuma regra o restringindo.

A Tabela 18 apresenta o atributo univalorado no DER e o `datatypeProperty` em OWL. A Tabela 19 apresenta o atributo multivalorado no DER e o `datatypeProperty` em OWL.

**Tabela 18 - Atributo Univalorado**


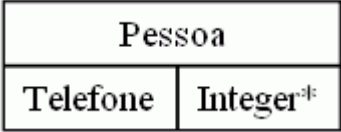
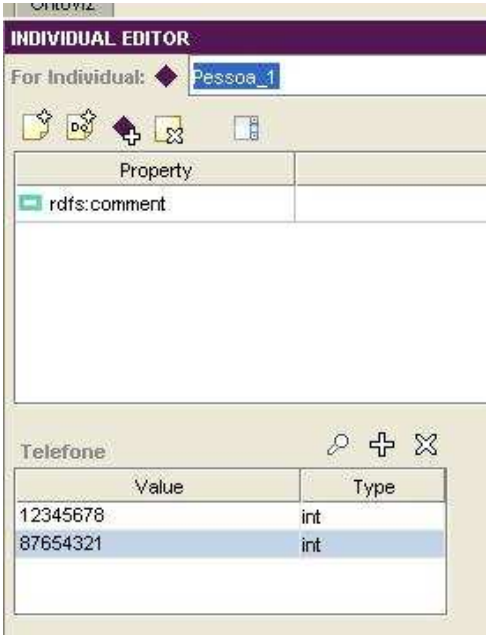
Atributo Univalorado
DER

OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="#Idade"/&gt;       &lt;owl:maxCardinality rdf:datatype="&amp;xsd:int"&gt;1&lt;/owl:maxCardinality&gt;     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="Idade"&gt; </pre>

```

<rdfs:domain rdf:resource="#Pessoa"/>
<rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

```

**Tabela 19 - Atributo Multivalorado**

Atributo Multivalorado						
DER						
						
OWL						
						
 <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Value</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>12345678</td> <td>int</td> </tr> <tr> <td>87654321</td> <td>int</td> </tr> </tbody> </table>	Value	Type	12345678	int	87654321	int
Value	Type					
12345678	int					
87654321	int					
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"/&gt; &lt;owl:DatatypeProperty rdf:ID="Telefone"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="&amp;xsd:int"/&gt; &lt;/owl:DatatypeProperty&gt; </pre>						

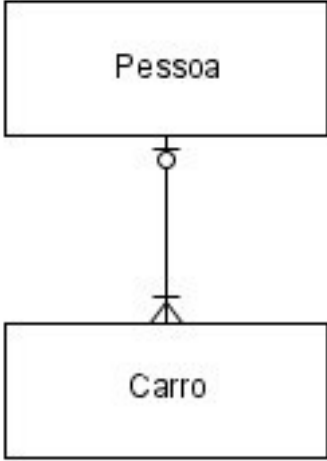

### 3.2.5 Atributo relacionante ou descritivo

Um atributo é relacionante quando se é o elo entre duas entidades, que efetiva o relacionamento entre duas entidades. O oposto desse tipo de atributo é descritivo, apresentando dados que descrevem a entidade, indiferente aos seus relacionamentos.

Em OWL, um atributo do tipo relacionante é equivalente ao `objectProperty` e um atributo do tipo descritivo é equivalente ao `dataTypeProperty`.

A Tabela 20 apresenta a comparação entre atributo relacionante de DER e `objectProperty` do OWL. A Tabela 21 apresenta a comparação entre atributo descritivo de DER e `dataTypeProperty`.

**Tabela 20 - Atributo Relacionante**

Atributo Relacionante
DER

OWL

<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Carro"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt;   &lt;rdfs:subClassOf&gt;   &lt;owl:Restriction&gt;     &lt;owl:onProperty rdf:resource="#E_possuido_por_Pessoa"/&gt;       &lt;owl:maxCardinality rdf:datatype="&amp;xsd:int"&gt;1&lt;/owl:maxCardinality&gt;     &lt;/owl:Restriction&gt;   &lt;/rdfs:subClassOf&gt; </pre>

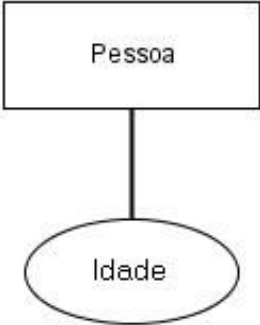
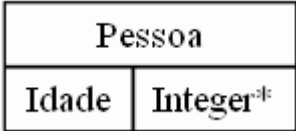


```

</owl:Class>
<owl:ObjectProperty rdf:ID="Possui_Carro">
  <rdfs:domain rdf:resource="#Pessoa"/>
  <rdfs:range rdf:resource="#Carro"/>
  <owl:inverseOf rdf:resource="#E_possuido_por_Pessoa"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="E_possuido_por_Pessoa">
  <rdfs:domain rdf:resource="#Carro"/>
  <rdfs:range rdf:resource="#Pessoa"/>
  <owl:inverseOf rdf:resource="#Possui_Carro"/>
</owl:ObjectProperty>

```

**Tabela 21 - Atributo Descritivo**

Atributo Descritivo
DER

OWL

<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"&gt;   &lt;rdfs:subClassOf rdf:resource="&amp;owl;Thing"/&gt; &lt;/owl:Class&gt; &lt;owl:DatatypeProperty rdf:ID="Idade"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="&amp;xsd:int"/&gt; &lt;/owl:DatatypeProperty&gt; </pre>

### 3.2.6 Atributo derivado

Um atributo é derivado quando seu valor é obtido através de manipulação de outros atributos. Um exemplo desse tipo de atributo: o salário de um funcionário é calculado a partir de um salário base (um valor definido a partir de seu cargo) somado ao tempo de serviço em anos multiplicado pelo aumento anual (também referente ao seu cargo), resultando na fórmula: “Salário base + (Tempo de serviço anual X Valor de aumento anual)”. A partir disto, o valor desse atributo é derivado desse cálculo.

Esse tipo de atributo não possui equivalente em OWL.

### 3.2.7 Atributo identificador

Um atributo é identificador quando seu valor é usado para reconhecer inequivocamente cada instância de uma entidade.

O atributo identificador não possui um construto equivalente em OWL, mas a semântica deste pode ser representada com a reificação do atributo único em uma nova classe, que terá propriedades de objeto com a classe que a possui (funcional e inverso funcional) e uma propriedades de tipo de dado funcional, com o nome "Atributo\_Unico", como apresentado no exemplo.

Este tipo de atributo pode ser representado em OWL de outra forma, caso este esteja no formato OWL-Full, permitindo que um dataTypeProperty seja funcional inverso, ou seja, o valor (seja texto ou número) estará relacionado somente a uma única instância de uma classe.

A Tabela 22 apresenta o atributo identificador do DER e a propriedade de objeto do OWL.

**Tabela 22 - Atributo Identificador**

Atributo Identificador																					
DER																					
<p>Diagrama de uma entidade Pessoa (retângulo) com um atributo CPF (oval) que é o identificador. Uma linha vertical conecta a entidade ao atributo.</p>																					
OWL																					
<p>Diagrama OWL mostrando a reificação do atributo CPF em uma classe Pessoa e a criação de uma classe CPF com propriedades de objeto.</p> <p>Classe Pessoa:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td colspan="4">Pessoa</td></tr> <tr><td>Tem CPF</td><td>Instance</td><td>CPF</td><td></td></tr> </table> <p>Classe CPF:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td colspan="4">CPF</td></tr> <tr><td>Pertence_a_Pessoa</td><td>Instance</td><td>Pessoa</td><td></td></tr> <tr><td colspan="2">Atributo_unico</td><td colspan="2">String</td></tr> </table> <p>Propriedades de objeto: Tem_CPF (de Pessoa para CPF) e Pertence_a_Pessoa (de CPF para Pessoa).</p>		Pessoa				Tem CPF	Instance	CPF		CPF				Pertence_a_Pessoa	Instance	Pessoa		Atributo_unico		String	
Pessoa																					
Tem CPF	Instance	CPF																			
CPF																					
Pertence_a_Pessoa	Instance	Pessoa																			
Atributo_unico		String																			
<p><b>Código correspondente:</b></p> <pre>&lt;owl:Class rdf:ID="Pessoa"/&gt; &lt;owl:ObjectProperty rdf:ID="Pertence_a_Pessoa"&gt;</pre>																					

```

    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <rdfs:domain rdf:resource="#CPF"/>
    <rdfs:range rdf:resource="#Pessoa"/>
    <owl:inverseOf rdf:resource="#Tem_CPF"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Tem_CPF">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <rdfs:domain rdf:resource="#Pessoa"/>
    <rdfs:range rdf:resource="#CPF"/>
    <owl:inverseOf rdf:resource="#Pertence_a_Pessoa"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="CPF"/>
<owl:DatatypeProperty rdf:ID="Atributo_unico">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#CPF"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

### 3.3 Relacionamento

Relacionamento é uma associação entre duas e somente duas entidades. Em modelagem conceitual, é comum a presença de um atributo compartilhado entre duas entidades. Relacionamentos podem ser classificados nas seguintes categorias: normal, dependência, associativo, categoria, e papel.

#### 3.3.1 Relacionamento do tipo normal

Relacionamento do tipo normal é um relacionamento simples, que associa entidades que não são especializadas.

Em OWL, o relacionamento do tipo normal é equivalente ao `objectProperty`, idem a seção 2.3.4. A Tabela 23 apresenta o relacionamento do tipo normal em DER e o `objectProperty` em OWL.

Tabela 23 - Relacionamento do tipo normal

Relacionamento do tipo normal
<p>DER</p>
<p>OWL</p>
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Carro" /&gt; &lt;owl:ObjectProperty rdf:ID="Possui_Carro"&gt;   &lt;rdfs:domain rdf:resource="#Pessoa"/&gt;   &lt;rdfs:range rdf:resource="#Carro"/&gt;   &lt;owl:inverseOf rdf:resource="#E_possuido_por_Pessoa"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:ObjectProperty rdf:ID="E_possuido_por_Pessoa"&gt;   &lt;rdfs:domain rdf:resource="#Carro"/&gt;   &lt;rdfs:range rdf:resource="#Pessoa"/&gt;   &lt;owl:inverseOf rdf:resource="#Possui_Carro"/&gt; &lt;/owl:ObjectProperty&gt; </pre>

### 3.3.2 Relacionamento do tipo dependência

Relacionamento do tipo dependência identifica a existência de uma entidade dependente de outra entidade, ligadas por esse relacionamento.

Ver seção 3.1.2.

### 3.3.3 Relacionamento do tipo associativo

Relacionamento do tipo associativo identifica a existência de uma entidade associativa associada a uma entidade, ligadas por esse relacionamento

Ver seção 3.1.3 .

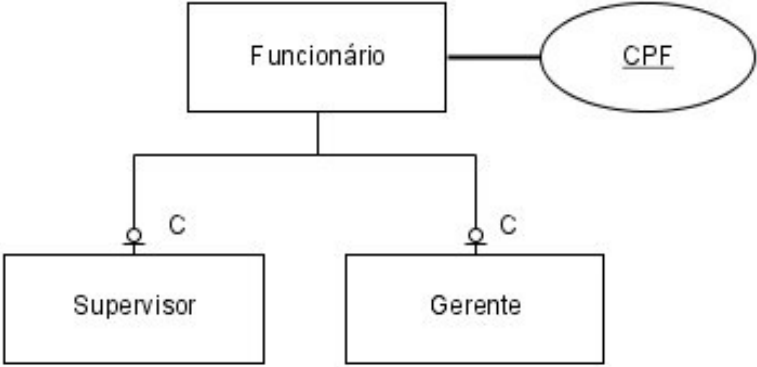
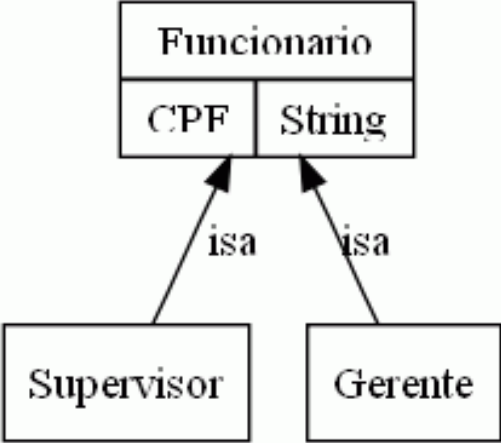
### 3.3.4 Relacionamento do tipo categoria

Relacionamento do tipo categoria identifica a existência de uma generalização, em que uma entidade é supertipo de outras entidades, compartilhando atributos comuns a todas elas e pertencente à superclasse. Essas subclasses provêm identidade a um conjunto de instâncias diferentes entre cada subclasse.

De forma semelhante à apresentada na comparação com o modelo lógico do ER-Win, em OWL, esse tipo de relacionamento é equivalente ao relacionamento existente entre uma classe e sua subclasse chamado subClassOf, que na figura tem o nome de "isa".

A Tabela 24 apresenta a comparação entre relacionamento do tipo categoria de DER e subclasse de OWL.

**Tabela 24 - Relacionamento do tipo categoria**

Relacionamento do tipo categoria
DER

OWL

<b>Código correspondente:</b> <pre>&lt;owl:Class rdf:ID="Funcionario"/&gt; &lt;owl:Class rdf:ID="Gerente"&gt;   &lt;rdfs:subClassOf rdf:resource="#Funcionario"/&gt;</pre>

```

</owl:Class>
<owl:Class rdf:ID="Supervisor">
  <rdfs:subClassOf rdf:resource="#Funcionario"/>
</owl:Class>

```

### 3.3.5 Relacionamento do tipo papel

Relacionamento do tipo papel identifica uma entidade que é subclasse de uma entidade, porém, não é idêntico ao relacionamento do tipo categoria. Uma entidade que participa de um relacionamento do tipo papel identifica instâncias da sua superclasse durante um evento específico. A instância de uma superclasse pode ser identificada em mais de um papel ao mesmo tempo ou em nenhum.

Em OWL, esse tipo de construto não possui um equivalente. Entretanto, pode ser representando com a seguinte estrutura: as subclasses equivalentes a papéis com nome específico (no exemplo, o texto "Papel\_" seguido pelo nome da classe).

A Tabela 25 apresenta a comparação do relacionamento do tipo papel de DER e subclasse de OWL.

**Tabela 25 - Relacionamento do tipo papel**

Relacionamento do tipo papel
DER
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"/&gt; &lt;owl:Class rdf:ID="Papel_Estudante"&gt; </pre>

```

    <rdfs:subClassOf rdf:resource="#Pessoa"/>
</owl:Class>
<owl:Class rdf:ID="Papel_Professor">
    <rdfs:subClassOf rdf:resource="#Pessoa"/>
</owl:Class>

```

## 4 Expressividade adicional da linguagem OWL

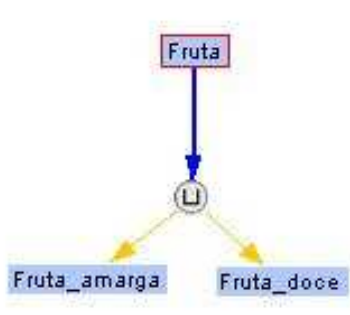
Com a comparação entre os construtos do modelo lógico do ERWin e o diagrama entidade-relacionamento com os construtos OWL, alguns construtos da OWL não possuem equivalente. Estes são as operações de conjunto (união, interseção, classes disjuntas, complemento, classes enumeradas), relacionamentos transitivo, simétrico, funcional, inverso funcional e inverso, instância, semelhança entre instâncias, entre propriedades e entre classes e restrições de range (allValuesFrom, someValuesFrom, hasValue, oneOf). As figuras seguintes foram gerada na ferramenta GrOWL [GROWL, 2009].

### 4.1 Operação de conjunto: União

Este construto permite expressar a existência de uma classe que representa a união de outras duas ou mais classes (ou definições), ou seja,  $X$  é uma classe que é a união das classes  $Y$  e  $Z$ . Tal afirmação denota que todos os indivíduos pertencentes a  $X$  ou são pertencentes a  $Y$  ou a  $Z$ .

A Tabela 26 apresenta um exemplo de união de OWL. Neste exemplo, a classe fruta é definida pela união das classes fruta amarga e fruta doce, ou seja, as instâncias pertencentes à fruta são o conjunto de instâncias da classe fruta amarga somado com o conjunto de instâncias da classe fruta doce.

Tabela 26 - União

União
<p>OWL</p> 
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Fruta"&gt;   &lt;rdfs:subClassOf rdf:resource="#owl:Thing"/&gt;   &lt;rdfs:subClassOf&gt;   &lt;owl:Class&gt;   &lt;owl:unionOf rdf:parseType="Collection"&gt;     &lt;owl:Class rdf:about="#Fruta_amarga"/&gt;     &lt;owl:Class rdf:about="#Fruta_doce"/&gt;   &lt;/owl:unionOf&gt; &lt;/owl:Class&gt; </pre>

```

    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Fruta_doce">
<owl:Class rdf:ID="Fruta_amarga">

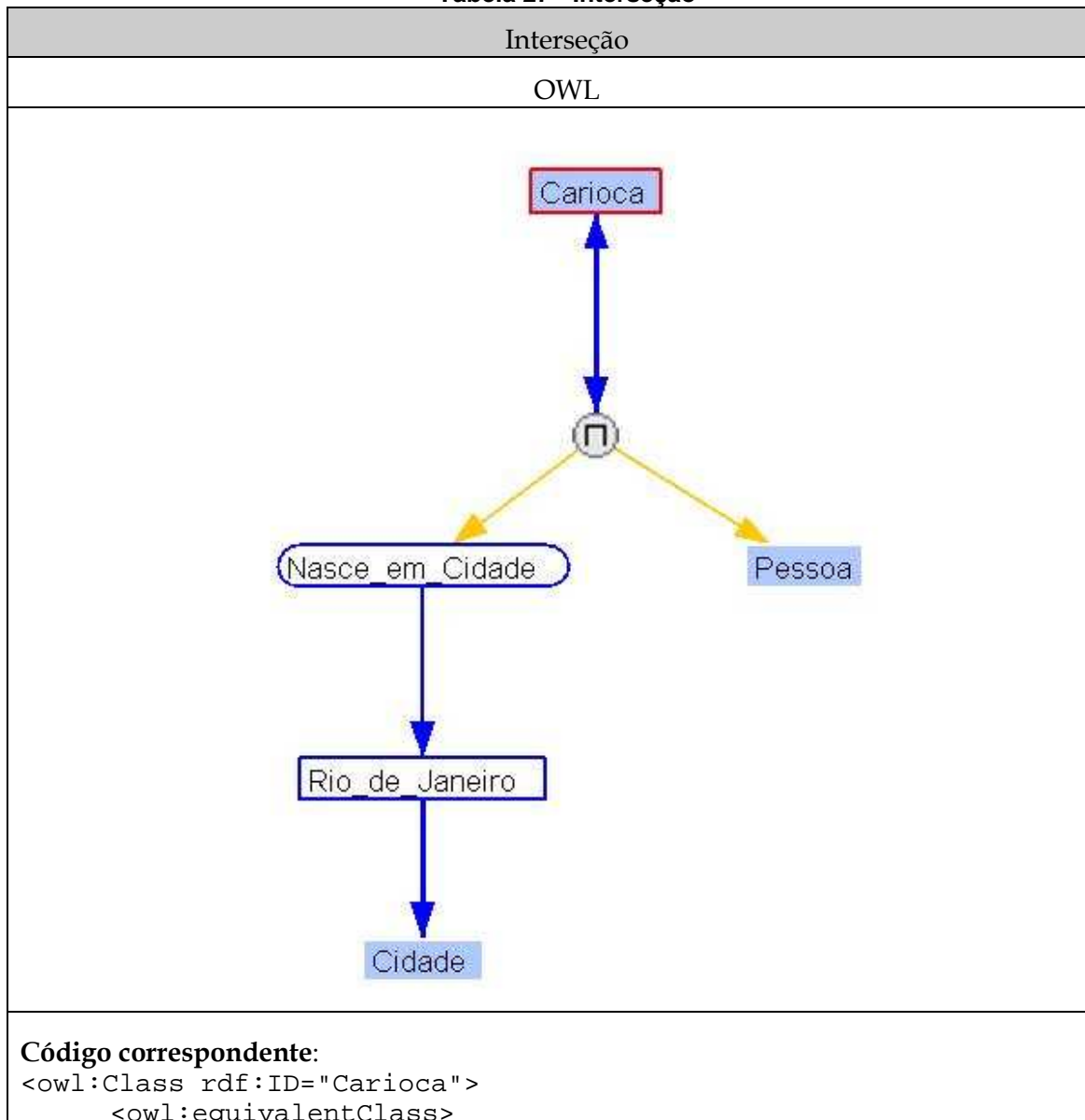
```

## 4.2 Operações de conjunto: Interseção

Este construto permite expressar a existência de uma classe que representa a interseção de outras duas ou mais classes (ou interseções), ou seja,  $X$  é uma classe que é a interseção das classes  $Y$  e  $Z$ . Tal afirmação denota que todos os indivíduos pertencentes a  $X$  ou são pertencentes a  $Y$  e a  $Z$ .

A Tabela 27 apresenta um exemplo de interseção de OWL. Neste exemplo, a classe carioca é definida pela interseção da classe pessoa e instâncias que nascem na cidade do Rio de Janeiro, ou seja, as instâncias pertencentes à carioca são o conjunto de instâncias da classe pessoa e que possuem o relacionamento "Nascer\_em\_Cidade" ligado com a cidade "Rio\_de\_Janeiro".

**Tabela 27 - Interseção**





```

<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Nasce_em_Cidade"/>
      <owl:hasValue rdf:resource="#Rio_de_Janeiro"/>
    </owl:Restriction>
    <owl:Class rdf:about="#Pessoa"/>
  </owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
<rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>

```

### 4.3 Operação de conjunto: Complemento

Este construto permite expressar a existência de uma classe que é complemento de outra(s) classe(s), ou seja, a classe X é complemento da classe Y. Tal afirmação denota que todos os indivíduos pertencentes a X é o conjunto de indivíduos que não pertencem a Y.

A Tabela 28 apresenta um exemplo de complemento em OWL. Neste exemplo, a classe homem é complemento da classe mulher e vice-versa, ou seja, o conjunto de instâncias pertencentes a homem é complementado pelo conjunto de instâncias pertencentes à mulher e vice-versa.

**Tabela 28 - Complemento**

Complemento
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"/&gt; &lt;owl:Class rdf:ID="Homem"&gt;   &lt;rdfs:subClassOf rdf:resource="#Pessoa"/&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Class&gt;       &lt;owl:complementOf rdf:resource="#Mulher"/&gt;     &lt;/owl:Class&gt;   &lt;/rdfs:subClassOf&gt; </pre>

```

</owl:Class>
<owl:Class rdf:ID="Mulher">
  <rdfs:subClassOf rdf:resource="#Pessoa"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf rdf:resource="#Homem"/>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>

```

#### 4.4 Operação de conjunto: Classes disjuntas

Este construto permite expressar a existência de uma classe que é disjunta de outra(s) classe(s), ou seja,  $X$  é classe disjunta de  $Y$ . Tal afirmação denota que todos os indivíduos pertencentes a  $X$  não podem pertencer a  $Y$ .

A Tabela 29 apresenta um exemplo de classes disjuntas de OWL. Neste exemplo, a classe homem é disjunta da classe mulher e vice-versa, ou seja, nenhuma instância de homem pode pertencer ao conjunto de instâncias de mulher.

**Tabela 29 - Classes disjuntas**

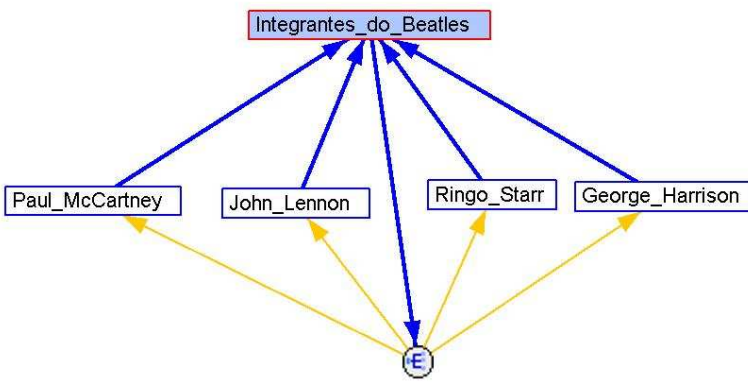
Classes disjuntas
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Pessoa"/&gt; &lt;owl:Class rdf:ID="Homem"&gt;   &lt;rdfs:subClassOf rdf:resource="#Pessoa"/&gt;   &lt;owl:disjointWith rdf:resource="#Mulher"/&gt; &lt;/owl:Class&gt; &lt;owl:Class rdf:ID="Mulher"&gt;   &lt;rdfs:subClassOf rdf:resource="#Pessoa"/&gt;   &lt;owl:disjointWith rdf:resource="#Homem"/&gt; &lt;/owl:Class&gt; </pre>

## 4.5 Operações de conjunto: Classes enumeradas (OneOf)

Este construto permite expressar a existência de um conjunto de instâncias que podem pertencem a uma classe, ou seja, as instâncias  $P$ ,  $T$  e  $W$  são um conjunto de instâncias da classe enumerada  $X$ . Tal afirmação denota que todos os indivíduos pertencentes a  $X$  não podem diferentes de  $P$ ,  $T$  e  $W$ .

A Tabela 30 apresenta um exemplo de classe enumerada de OWL. Neste exemplo, a classe enumerada dos integrantes do Beatles é o conjunto de instâncias “Paul\_McCartney”, “John\_Lennon”, “George\_Harrison” e “Ringo\_Star”, ou seja, a classe de integrantes do Beatles não pode possuir instâncias diferentes das presentes neste conjunto.

Tabela 30 - Classes enumeradas

Classes enumeradas
OWL

<b>Código correspondente:</b> <pre data-bbox="226 1238 1252 1657">&lt;owl:Class rdf:ID="Integrantes_do_Beatles"&gt;   &lt;rdfs:subClassOf rdf:resource="#owl;Thing"/&gt;   &lt;rdfs:subClassOf&gt;   &lt;owl:Class&gt;     &lt;owl:oneOf rdf:parseType="Collection"&gt;       &lt;rdf:Description rdf:about="#George_Harrison"/&gt;       &lt;rdf:Description rdf:about="#John_Lennon"/&gt;       &lt;rdf:Description rdf:about="#Paul_McCartney"/&gt;       &lt;rdf:Description rdf:about="#Ringo_Starr"/&gt;     &lt;/owl:oneOf&gt;   &lt;/owl:Class&gt; &lt;/rdfs:subClassOf&gt; &lt;/owl:Class&gt;</pre>


## 4.6 Relacionamento transitivo

Este tipo de relacionamento permite expressar a existência de um relacionamento transitivo, ou seja, que em um relacionamento transitivo  $relacionamento_T$  entre os pares  $(x, y)$  e  $(y, z)$  é inferido que  $relacionamento_T(x, z)$ .

A Tabela 31 apresenta um exemplo de relacionamento transitivo em OWL. Neste exemplo, a classe região geográfica possui um relacionamento transitivo explicitando que contém outra região geográfica, ou seja, se a região geográfica do Brasil contém a região geográfica do estado do Rio de Janeiro e a região geográfica do estado do Rio de

Janeiro contém a região geográfica da cidade do Rio de Janeiro, então a região geográfica do Brasil contém a região geográfica da cidade do Rio de Janeiro.

**Tabela 31 - Relacionamento transitivo**


Relacionamento transitivo
OWL

<p><b>Código correspondente:</b></p> <pre>&lt;owl:Class rdf:ID="Regiao_geografica"/&gt; &lt;owl:ObjectProperty rdf:ID="Contem_Regiao_geografica"&gt;   &lt;rdf:type rdf:resource="#owl:TransitiveProperty"/&gt;   &lt;rdfs:domain rdf:resource="#Regiao_geografica"/&gt;   &lt;rdfs:range rdf:resource="#Regiao_geografica"/&gt; &lt;/owl:ObjectProperty&gt;</pre>

#### 4.7 Relacionamento simétrico

Este tipo de relacionamento permite expressar a existência de um relacionamento simétrico, ou seja, que em um relacionamento simétrico *relacionamento<sub>S</sub>* entre o par  $(x, y)$  é inferido que *relacionamento<sub>S</sub>*  $(y, x)$ .

A Tabela 32 apresenta um exemplo de relacionamento simétrico em OWL. Neste exemplo, a classe região geográfica possui um relacionamento simétrico explicitando que é adjacente a outra região geográfica, ou seja, se a região geográfica do estado do Rio de Janeiro é adjacente a região geográfica do estado de São Paulo, então a região geográfica do estado de São Paulo é adjacente a região geográfica do estado do Rio de Janeiro.

**Tabela 32 - Relacionamento simétrico**

Relacionamento simétrico
OWL

<p><b>Código correspondente:</b></p> <pre>&lt;owl:Class rdf:ID="Regiao_geografica"/&gt; &lt;owl:ObjectProperty rdf:ID="E_adjacente_a_Regiao_geografica"&gt;   &lt;rdf:type rdf:resource="#owl:SymmetricProperty"/&gt;   &lt;rdfs:domain rdf:resource="#Regiao_geografica"/&gt;   &lt;rdfs:range rdf:resource="#Regiao_geografica"/&gt;   &lt;owl:inverseOf rdf:resource="#E_adjacente_a_Regiao_geografica"/&gt; &lt;/owl:ObjectProperty&gt;</pre>

## 4.8 Relacionamento funcional

Este tipo de relacionamento define que em um relacionamento funcional *relacionamento\_F* entre os pares  $(x, y)$  e  $(x, z)$  é inferido que  $y$  é igual a  $z$  e a qualquer outra instância que assumir o papel de *range* em um relacionamento funcional.

Em propriedades de tipos de dados, define que em um relacionamento funcional *relacionamentoDeTipoDeDado\_F* entre os pares  $(x, y)$  e  $(x, z)$ , sendo  $x$  uma classe e  $y$  e  $z$  tipos de dados, é inferido que  $y$  é igual a  $z$  e a qualquer outro valor que assumir o papel de *range* em um relacionamento funcional.

A Tabela 22 apresenta um exemplo de relacionamento funcional em OWL.

## 4.9 Relacionamento inverso funcional

Este tipo de relacionamento define que em um relacionamento inverso funcional *relacionamento\_IF* entre os pares  $(x, y)$  e  $(z, y)$  é inferido que  $x$  é igual a  $z$  e a qualquer outra instância que assumir o papel de *range* em um relacionamento inverso funcional.

Em propriedades de tipos de dados, define que em um relacionamento inverso funcional *relacionamentoDeTipoDeDado\_IF* entre os pares  $(x, y)$  e  $(z, y)$ , sendo  $x$  e  $z$  classes e  $y$  tipos de dados, é inferido que  $x$  é igual a  $z$  e a qualquer outra instância que assumir o papel de *range* em um relacionamento inverso funcional.

A Tabela 22 apresenta um exemplo de relacionamento funcional em OWL.

## 4.10 Relacionamento inverso

Este tipo de relacionamento define que um relacionamento é inverso do outro; se *relacionamento\_A\_inverso\_de\_B* entre o par  $(x, y)$  é um relacionamento inverso de *relacionamento\_B\_inverso\_de\_A*, é inferido que existe o *relacionamento\_B\_inverso\_de\_A* entre o par  $(y, x)$ .

A Tabela 8, abordada anteriormente, apresenta um exemplo de relacionamento inverso em OWL.

## 4.11 Instâncias

Instâncias representam algo no mundo real que se enquadra em uma classe específica, um objeto no mundo real.

A Tabela 19 apresenta um exemplo de instância em OWL.

## 4.12 Semelhança entre instâncias: sameAs

Este construto permite expressar que dois indivíduos são semelhantes, identificados como a mesma instância, ou seja, caso a instância  $P$  seja equivalente a  $T$ , então  $P$  igual a  $T$ .

A Tabela 33 apresenta um exemplo de semelhança entre instâncias em OWL. Neste exemplo, a instância Rio de Janeiro é semelhante à instância Cidade Maravilhosa, ou seja, as duas instâncias representam o mesmo objeto da realidade, a cidade do Rio de Janeiro.

**Tabela 33 - Semelhança entre instâncias**


Semelhança entre instâncias
OWL
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Cidade"/&gt; &lt;Cidade rdf:ID="Cidade_Maravilhosa"&gt;   &lt;owl:sameAs rdf:resource="#Rio_de_Janeiro"/&gt; &lt;/Cidade&gt; &lt;Cidade rdf:ID="Rio_de_Janeiro"&gt;   &lt;owl:sameAs rdf:resource="#Cidade_Maravilhosa"/&gt; &lt;/Cidade&gt; </pre>

#### 4.13 Semelhança entre propriedades: `equivalentProperty`

Este construto permite expressar que duas propriedades são semelhantes, identificadas como o mesmo conceito, ou seja, caso a propriedade *R* seja equivalente a *S*, então *R* igual a *S*.

A Tabela 34 apresenta um exemplo de semelhança entre propriedades de OWL. Neste exemplo, a propriedade de objeto extrai óleo é semelhante à propriedade extrai petróleo, ou seja, os dois relacionamentos representam o mesmo relacionamento na realidade e seus domínios e *ranges* devem ser iguais.

**Tabela 34 - Semelhança entre propriedades**

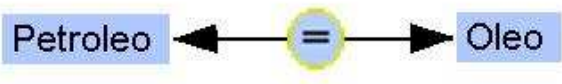
Semelhança entre propriedades	
OWL	
	
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Oleo"/&gt; &lt;owl:Class rdf:ID="Petroleo"/&gt; &lt;owl:Class rdf:ID="Poco"/&gt; &lt;owl:ObjectProperty rdf:ID="Extrai_Oleo"&gt;   &lt;rdfs:domain rdf:resource="#Poco"/&gt;   &lt;owl:equivalentProperty rdf:resource="#Extrai_Petroleo"/&gt;   &lt;rdfs:range rdf:resource="#Oleo"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:ObjectProperty rdf:ID="Extrai_Petroleo"&gt;   &lt;rdfs:domain rdf:resource="#Poco"/&gt;   &lt;owl:equivalentProperty rdf:resource="#Extrai_Oleo"/&gt;   &lt;rdfs:range rdf:resource="#Petroleo"/&gt; &lt;/owl:ObjectProperty&gt; </pre>	

#### 4.14 Semelhança entre classes: equivalentClass

Este construto permite expressar que duas classes são semelhantes, identificadas como o mesmo conceito, ou seja, caso a classe X seja equivalente a Y, então X igual a Y.

A Tabela 35 apresenta um exemplo de semelhança entre classes de OWL. Neste exemplo, a classe petróleo é semelhante a classe óleo, ou seja, as duas classes representam o mesmo conceito na realidade e seus conjuntos de instâncias são iguais.

Tabela 35 - Semelhança entre classes

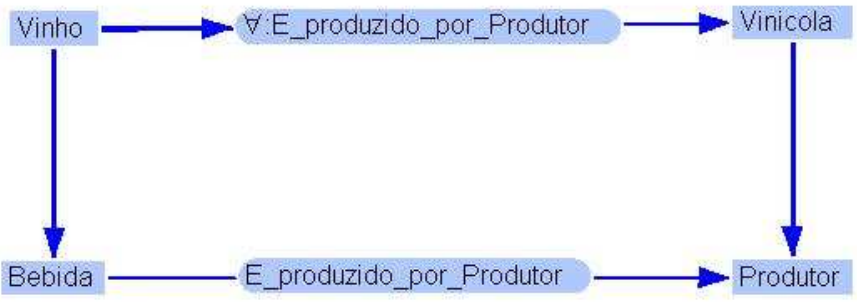
Semelhança entre classes
OWL

<p><b>Código correspondente:</b></p> <pre>&lt;owl:Class rdf:ID="Oleo"&gt;   &lt;owl:equivalentClass rdf:resource="#Petroleo"/&gt; &lt;/owl:Class&gt; &lt;owl:Class rdf:ID="Petroleo"&gt;   &lt;owl:equivalentClass rdf:resource="#Oleo"/&gt; &lt;/owl:Class&gt;</pre>

#### 4.15 Restrição de range “Todos os valores de”: allValuesFrom

Este construto restringe a uma classe *X* o *range* de um relacionamento, proibindo a instanciação desses relacionamentos com instância de classe que não sejam da classe *X*.

A Tabela 36 apresenta um exemplo de restrição *allValuesFrom* de OWL. Neste exemplo, a classe *vinho* é subclasse de *bebida* e *vinícola* é subclasse de *produtor*. *Bebida* e *produtor* possuem um relacionamento indicando que uma bebida é produzida por um produtor, e esse relacionamento é herdado por *vinho*, porém existe uma restrição *allValuesFrom* com a classe *vinícola*, ou seja, para que o relacionamento seja usado por *vinho*, todas as instâncias no *range* devem ser de *vinícola*.

Tabela 36 - Restrição *allValuesFrom*

Restrição <i>allValuesFrom</i>
OWL

<p><b>Código correspondente:</b></p> <pre>&lt;owl:Class rdf:ID="Bebida"/&gt; &lt;owl:ObjectProperty rdf:ID="E_produzido_por_Produtor"&gt;   &lt;rdfs:domain rdf:resource="#Bebida"/&gt;   &lt;rdfs:range rdf:resource="#Produtor"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:Class rdf:ID="Produtor"/&gt;</pre>



```

<owl:Class rdf:ID="Vinho">
  <rdfs:subClassOf rdf:resource="#Bebida"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
rdf:resource="#E_produzido_por_Produtor"/>
      <owl:allValuesFrom rdf:resource="#Vinicola"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Vinicola">
  <rdfs:subClassOf rdf:resource="#Produtor"/>
</owl:Class>

```

#### 4.16 Restrição de range “Alguns valores de”: someValuesFrom

Este construto restringe a instanciação de um relacionamento com essa regra a uma classe *X*, obrigando o uso de pelo menos uma instância de uma classe *X* no *range*.

A Tabela 37 apresenta um exemplo de restrição *someValuesFrom* da OWL. Neste exemplo, a classe vinho é subclasse de bebida e vinícola é subclasse de produtor. Bebida e produtor possuem um relacionamento indicando que uma bebida é produzida por um produtor, e esse relacionamento é herdado por vinho, porém existe uma restrição *someValuesFrom* com a classe vinícola, ou seja, para que o relacionamento seja usado por vinho, pelo menos uma instância no *range* deve ser de vinícola.

Tabela 37 - Restrição *someValuesFrom*

Restrição <i>someValuesFrom</i>	
OWL	
<pre> graph TD     Vinho --&gt; PropRest[∃:E_produzido_por_Produtor]     PropRest --&gt; Vinicola     Vinho --&gt; Bebida     Vinicola --&gt; Produtor     Bebida --&gt; PropGen[E_produzido_por_Produtor]     PropGen --&gt; Produtor </pre>	
<p><b>Código correspondente:</b></p> <pre> &lt;owl:Class rdf:ID="Bebida"/&gt; &lt;owl:ObjectProperty rdf:ID="E_produzido_por_Produtor"&gt;   &lt;rdfs:domain rdf:resource="#Bebida"/&gt;   &lt;rdfs:range rdf:resource="#Produtor"/&gt; &lt;/owl:ObjectProperty&gt; &lt;owl:Class rdf:ID="Produtor"/&gt; &lt;owl:Class rdf:ID="Vinho"&gt;   &lt;rdfs:subClassOf rdf:resource="#Bebida"/&gt;   &lt;rdfs:subClassOf&gt;     &lt;owl:Restriction&gt;       &lt;owl:onProperty rdf:resource="#E_produzido_por_Produtor"/&gt; </pre>	

```

        <owl:someValuesFrom rdf:resource="#Vinicola"/>
    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Vinicola">
    <rdfs:subClassOf rdf:resource="#Produtor"/>
</owl:Class>

```

#### 4.17 Restrição de range “Tem valor”: hasValue

Este construto permite especificar classes a partir das instâncias definidas no *range* de seus relacionamentos: caso uma classe  $x$  possua o relacionamento *relacionamento\_tem\_valor\_z* com a restrição *hasValue(z)*, todo indivíduo presente na ontologia que possuir o *relacionamento\_tem\_valor\_z* com  $(z)$  será inferido como instância da classe  $x$ .

A Tabela 27 apresenta um exemplo da restrição *hasValue* de OWL.

## 5 Conclusão

Modelagem conceitual tem como objetivo representar uma realidade (ou parte). Existem várias formas de representação, algumas voltadas para a modelagem de dados, como Diagramas Entidade-Relacionamento, outras voltadas para representação de um domínio independente de dados, como UML [OMG, 2007] e ontologias. Ontologias criadas utilizando a linguagem OWL, além de serem um modelo conceitual de um domínio, são construídas sobre uma base de lógica matemática que as tornam capazes de serem processadas e inferir novo conhecimento [Dean *et al.*, 2006], sendo úteis tanto para entendimento do domínio quanto para adicionar interoperabilidade em sistemas de informação ou melhorar capacidades pesquisa nestes sistemas.

Apesar de seus objetivos diferentes, um modelo conceitual em DER pode ser representado em OWL, respeitando as características da linguagem e fazendo adaptações quando for necessário para o modelador. Foi também apresentado que o modelo DER apresentado por Chen [1976] possui construtores que não estão disponíveis na ferramenta ERWin.

## Referências Bibliográficas

- BAADER, F.; CALVANESE, D.; MCGUINNESS, D. L.; NARDI, D.; PATEL-SCHNEIDER, P. F. **The Description Logic Handbook: Theory, Implementation, Applications**. Cambridge University Press, Cambridge, UK, 2003.
- ERWIN, CA. **ERWin® Data Modeler**. Disponível em: <<http://www.ca.com/us/data-modeling.aspx>>. Acesso em: 21 jan. 2009.
- CHEN, P.P. **The Entity-Relationship Model-Toward a Unified View of Data**. ACM Transactions on Database Systems, v. 1, n. 1, p. 9-36, 1976.
- DEAN, M.S., G.; BECHHOFFER, S.; HARMELEN, F. V.; ENDLER, J.; HORROCKS, I.; MCGUINNES, D. L.; PATEL-SCHNEIDER, P. F.; STEIN, L. A. **OWL: Web Ontology Language Reference**, 2006. Disponível em: <<http://www.w3.org/TR/owl-ref/>>. Acesso em: 18 dez. 2008.

GLIFFY, Gliffy Online. Disponível em: <<http://www.gliffy.com/>>. Acesso em: 15 fev. 2009.

GROWL, GrOWL. Disponível em: <<http://www.uvm.edu/~skrivov/growl/>>. Acesso em: 10 mar. 2009.

OMG, **OMG Unified Modeling Language (OMG UML), Infrastructure**, v. 2.1.2, 2007. Disponível em: <<http://www.omg.org/spec/UML/2.1.2/>>. Acesso em: 05 fev. 2009.

POMPILHO, S. **Análise Essencial – guia prático de Análise de Sistemas**. Rio de Janeiro: Ciência Moderna Ltda, 2002.

PROTÉGÉ, **The Protégé Ontology Editor and Knowledge Acquisition System**. Universidade de Stanford, Estados Unidos. Disponível em: <<http://protege.stanford.edu/>>. Acesso em: 03 jan. 2009.